Adversarial Attacks and Robust Training for Hypergraph Neural Networks

Naheed Anjum Arafat, Howard University, USA

Debabrota Basu, INRIA, France

Yulia R. Gel, Virginia Tech, USA

Danda B. Rawat, Howard University, USA

Motivation

- Hypergraphs model higher-order relations beyond pairwise edges.
  - **Examples:**
    - recommender systems and item baskets
    - biomedical and multimodal relational data,
    - and many more..

- Hypergraph Neural Networks (HGNNs) are increasingly used for node classification and related prediction tasks.

- Recent works showed that HGNNs are vulnerable to adversarial perturbations.

Contributions

1. **Existing attacks are white-box** (HyperAttack, MGHGA, etc.)
   - unrealistic (e.g. In recommender systems, it is unrealistic to assume that the attacker has access to proprietary model parameters.)
2. **Existing attacks are limited to either node features or hypergraph structure, but not both.**
   - We propose attacks in gray-box setting (attacker knows the training labels, but not the model params)
   - We consider both structural and feature perturbations.
3. **There is no adversarial training mechanism for HGNN's defense.**
   - We propose MeLA-D: an adversarial training mechanism to robustify HGNNs.

Problem Statement & Threat Model

Problem: Semi-Supervised Node Classification task on Hypergraphs

- **Inputs:** incidence matrix $\boldsymbol{H}$, node features $\boldsymbol{X}$, and training labels $\boldsymbol{y}_L$.

- **Attacker goal:** Degrade target HGNN accuracy using budgeted perturbations.
  - Structural perturbations: flip entries of $\boldsymbol{H}$ under budget $\boldsymbol{\Delta_H}$: $\left|\left|\boldsymbol{\Delta_H}\right|\right|_0 \leq \delta_H$.
  - Feature perturbations: modify $\boldsymbol{X}$ under l_∞ budget $\boldsymbol{\Delta_X}$: $||\boldsymbol{\Delta_X}||_\infty \leq \delta_X$.

- **Gray-box:** attacker knows $\boldsymbol{H}, \boldsymbol{X}$ and $\boldsymbol{y}_L$, but not target HGNN parameters.

- **Evaluation:**
  - Poisoning setting: Attacker perturbs $\boldsymbol{H}, \boldsymbol{X}$ before model training.
  - Evasion setting: Attacker perturbs $\boldsymbol{H}, \boldsymbol{X}$ after the model is trained.

MeLA: The Meta-Laplacian Attack Objective

- MeLA trains a surrogate HGNN f_θ on clean hypergraph.

- It then finds perturbations $(\Delta \boldsymbol{H}, \Delta \boldsymbol{X})$ that **maximizes** the following **meta-objective** $(\mathcal{L}_{\boldsymbol{meta}})$.

$$\mathcal{L}_{meta}(\theta^*, \Delta \boldsymbol{H}, \Delta \boldsymbol{X}) = \alpha \mathcal{L}_{smooth}(\theta^*, \Delta \boldsymbol{H}, \Delta \boldsymbol{X}) - \beta \mathcal{L}_{stealth}(\theta^*, \Delta \boldsymbol{H}, \Delta \boldsymbol{X}) + \gamma \mathcal{L}_{train}(\theta^*, \Delta \boldsymbol{H}, \Delta \boldsymbol{X})$$

such that

$$\theta^* = argmin_\theta \, \mathcal{L}_{train}(f_\theta(\theta^*, \boldsymbol{H} + \Delta \boldsymbol{H}, \boldsymbol{X} + \Delta \boldsymbol{X}), \boldsymbol{y}_L), \qquad \alpha, \beta, \gamma > 0$$

- The objective combines:

  - **Laplacian smoothing drift** $(\mathcal{L}_{\boldsymbol{smooth}})$**:** disrupt message passing behavior

$$\mathcal{L}_{smooth} = \left|\left| \boldsymbol{L}_{pert} \boldsymbol{Z}_{pert} - \boldsymbol{L}_{orig} \boldsymbol{Z}_{orig} \right|\right|_2, \qquad \text{where}$$
$$\boldsymbol{L}_{orig} = \text{LAP}(\boldsymbol{H}), \quad \boldsymbol{L}_{pert} = \text{LAP}(\boldsymbol{H} + \Delta \boldsymbol{H})$$
$$\boldsymbol{Z}_{orig} = f_{\theta^*}(\boldsymbol{H}, \boldsymbol{X}), \quad \boldsymbol{Z}_{pert} = f_{\theta^*}(\boldsymbol{H} + \Delta \boldsymbol{H}, \ \boldsymbol{X} + \Delta \boldsymbol{X})$$

  - **Stealth penalty** $(\mathcal{L}_{\boldsymbol{stealth}})$**:** discourage large degree shifts to promote un-noticeability.

$$\mathcal{L}_{stealth} = \frac{1}{n} ||\boldsymbol{d}_{pert} - \boldsymbol{d}_{orig}||_2^2$$

  - **Classification loss** $(\mathcal{L}_{\boldsymbol{train}})$**:** make prediction harder (cross-entropy loss)

Attack variants: MeLA-FGSM vs MeLA-PGD

MeLA-FGSM	MeLA-PGD
Single Gradient-step (T=1)	Multi-step projected ascent (T>1)
one signed step at the full l_∞ budget	iterative signed steps, re-projected to budget each step
Gradient taken once, at the clean Laplacian	Gradient **recomputed** at the **perturbed Laplacian** every step
Best fit: **evasion-** model is frozen, so a one-shot step already matches the setting	Best fit: **poisoning & nonlinear regimes-** re-checking the gradient each step corrects the attack direction as the Laplacian shifts.

MeLA-D: Defense via Adversarial Training

- To robustify target HGNN, MeLA-D repeatedly:
  - generates MeLA-style adversarial perturbations $(\boldsymbol{\Delta H}, \boldsymbol{\Delta X})$ via continuous relaxation.
  - trains the target HGNN on the perturbed hypergraph $(\mathbf{H} + \boldsymbol{\Delta H}, \mathbf{X} + \boldsymbol{\Delta X})$
  - projects perturbations back to the allowed budgets via top-k.

- **Convergence behavior:** Under local smoothness, local concavity, and bounded stochastic-gradient variance assumptions, MeLA-D converges to a first-order stationary point of the relaxed robust-training objective at $\boldsymbol{O(T^{-1/2})}$**.**

Empirical Results

- **Datasets:** Citeseer, Cora, Cora-CA, Large-scale datasets (OGBN-MAG, Trivago, Yelp, Walmart, Flickr)
- **Target HGNNs:** HyperMLP, HGNN, AllsetTransformer
- **Baseline attacks:** RandFeat, RandFlip, GradArgMax, DICE, HyperAttack
- **Attack performance:** MeLA-PGD is the strongest attack.
  - Drop in accuracy:
    - up to 60.1% accuracy drop in poisoning
    - up to 75.9% accuracy drop in evasion

Table 2. Comparison of different adversarial attacks (poisoning and evasion) for Citeseer. Drop = median relative % accuracy drop.

Attack	HyperMLP (72.20 ± 0.40)		AllsetTrans. (72.90 ± 0.25)		HGNN (74.13 ± 0.10)	
	Poison / Drop (P)	Evasion / Drop (E)	Poison / Drop (P)	Evasion / Drop (E)	Poison / Drop (P)	Evasion / Drop (E)
RandFeat	69.42 ± 0.59 / 4.0	70.48 ± 0.92 / 2.3	71.84 ± 1.17 / 2.3	72.56 ± 0.77 / 0.7	71.64 ± 0.52 / 3.3	72.78 ± 0.49 / 2.0
RandFlip	72.05 ± 0.25 / 0.2	72.20 ± 0.40 / 0.0	67.44 ± 1.28 / 6.8	66.74 ± 1.16 / 8.6	64.49 ± 1.04 / 13.2	65.77 ± 0.98 / 11.6
GradArgMax	72.22 ± 0.40 / 0.0	72.20 ± 0.40 / 0.0	71.38 ± 0.76 / 1.8	71.06 ± 1.27 / 2.8	71.26 ± 0.51 / 3.7	71.91 ± 0.25 / 2.9
DICE	72.17 ± 0.45 / 0.0	72.20 ± 0.40 / 0.0	69.49 ± 1.17 / 5.5	70.05 ± 0.90 / 4.3	68.62 ± 1.08 / 7.8	69.28 ± 1.17 / 5.9
HyperAttack	72.22 ± 0.34 / 0.0	72.20 ± 0.40 / 0.0	71.47 ± 0.74 / 2.0	70.82 ± 0.72 / 3.0	63.91 ± 1.94 / 12.9	71.69 ± 0.33 / 3.1
MeLA-FGSM	66.55 ± 1.55 / 7.9	47.61 ± 0.87 / 34.3	61.59 ± 3.78 / 14.6	46.30 ± 0.50 / 36.3	68.41 ± 0.82 / 7.2	46.45 ± 0.82 / 37.9
MeLA-PGD	60.72 ± 3.69 / **15.4**	42.54 ± 1.83 / **41.1**	28.82 ± 0.75 / **60.1**	41.30 ± 1.43 / **43.7**	54.37 ± 1.17 / **26.5**	39.81 ± 3.45 / **44.6**

- **Defense performance:** MeLA-D+HyperMLP gains against MeLA-PGD:
  - Gain in accuracy:
    - +28.12% points on Citeseer
    - +41.60% points on Cora-CA
    - +31.14% points on Cora

Conclusion

- HGNNs are vulnerable under realistic gray-box attacks.
- MeLA gives a hypergraph-specific attack framework using a Laplacian-aware meta-objective.
  - MeLA-FGSM and MeLA-PGD jointly perturb structure and features.
- MeLA-D turns the same adversarial signal into a practical robust-training defense.

Thank You