BOCLOAK: Optimal Transport-Guided Adversarial Attacks on Graph Neural Network-Based Bot Detection

Kunal Mukherjee, Zulfikar Alom, Tran Gia Bao Ngo, Cuneyt Gurcan Akcora, Murat Kantarcioglu

ICML International Conference On Machine Learning

Motivation

- Social media bots threaten public discourse, yet the effectiveness of GNN-based bot detectors remains unclear.
- Existing graph attacks often assume unrealistic powers, such as arbitrary rewiring or full graph access, unlike real bots.
- Real attackers face hard constraints: limited observability, no forced human follow-backs, temporal plausibility, and only local edits around the bot.
- There is a need for an attack framework that reflects these real-world constraints while still being effective.
- Optimal transport offers a principled way to compare neighborhood structures and guide sparse, plausible edits.

Problem Definition

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, and a labelled nodes subset $\mathcal{V}_L \subset \mathcal{V}$ with $y_v \in \{\text{human}, \text{bot}\}$. A bot detector perform binary classification for each node.

The attacker operates at test time by injecting or editing a existing bot under constraints:

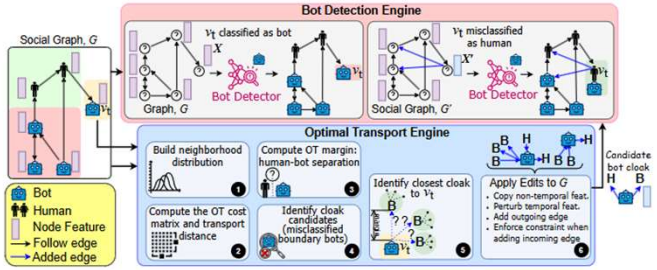
$$\mathcal{F}(B) = \{\Delta\mathcal{E} : |\Delta\mathcal{E}| \leq B,\ \Delta\mathcal{E} \subseteq (\{v_t\} \times \mathcal{V}) \cup (\mathcal{V} \times \{v_t\}),$$
$$\Delta\mathcal{E} \cap (\mathcal{V}_{\text{hum}} \times \{v_t\}) = \emptyset,$$
$$\Psi_{\text{dir}}(\Delta\mathcal{E}) \leq 0,\ \Psi_{\text{time}}(\Delta\mathcal{E}) \leq 0\}.$$

Black box setting:

✅ Training graph + node labels + node features

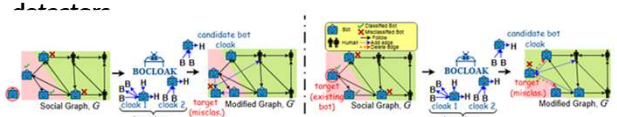
❌ Victim model architecture, parameters, gradients or logit outputs

BOCLOAK Framework



BOCLOAK learns an optimal transport geometry over k-hop neighborhood distributions and uses transport plans to guide sparse, plausible local edge edits for evasion.

1. Build a weighted neighborhood distribution for each node
2. Learn OT-based geometry between bot and human neighborhoods
3. Find boundary bot "cloaks" near the human region
4. Compute transport plans to match target bots to cloak templates
5. Decode high-mass transport matches into feasible edge edit
6. Apply sparse, realistic edits to evade GNN-based bot detectors

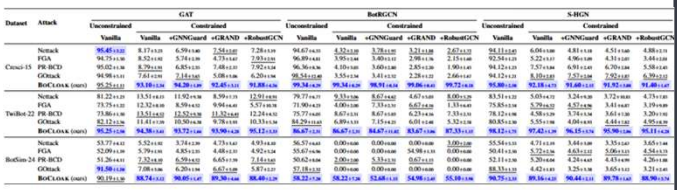


(a) Node injection (b) Node editing

BOCLOAK pipeline: (a) Injection constructs a new bot with a human-like neighborhood (b) Editing transforms an existing bot's neighborhood to induce misclassification

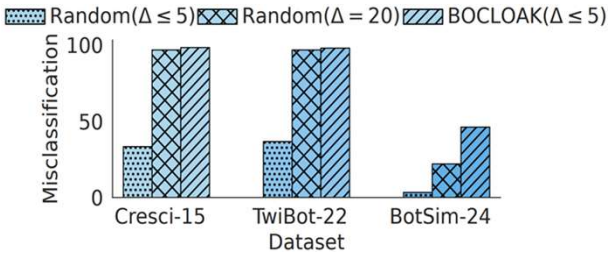
Experiments

Node Editing Attack (Δ = 1): Misclassification rate (%) for flipping 50 correctly classified bots against GAT, BotRGCN, and S-HGN under adversarial defenses



BOCLOAK incurs 99.80% less GPU footprint and runs up to 20 × faster than SOTA attacks.

Attack	BotSim-24		
	Time (sec) (↓)	RAM (MB) (↓)	GPU (MB) (↓)
Nettack	19.7 ± 1.65	2973 ± 628	156 ± 40
FGA	0.3 ± 1.57	1720 ± 13	286 ± 2
PR-BCD	2.1 ± 0.75	1857 ± 21	286 ± 1
GOttack	7.5 ± 1.75	2069 ± 623	240 ± 2
BOCLOAK	0.1 ± 0.13	1450 ± 2	11 ± 1



BOCLOAK achieves injection success with very small budgets

Ablation Study: Misclassification rate for node editing attack

Dataset	$-\lambda_{\text{BCE}}$	$-\lambda_{\text{sp}}$	$-\lambda_{\text{pl}}$	$-\alpha_{\text{deg}}$	$-\alpha_{\text{age}}$	BOCLOAK
Cresci-17	38.21±5.12	96.88±5.89	57.99±3.38	73.23±0.23	87.23±1.78	**99.67±0.20**
TwiBot-22	22.45±6.44	92.12±7.65	55.23±6.77	70.98±4.29	83.88±6.63	**94.00±2.00**
BotSim-24	32.45±5.64	96.55±0.66	60.34±5.09	69.22±2.15	85.24±1.38	**99.33±1.15**

VIRGINIA TECH UCF THE UNIVERSITY OF TOLEDO University of Manitoba