AttackGNN: Red-Teaming GNNs in Hardware Security Using Reinforcement Learning

Vasudev Gohil

Satwik Patnaik

Dileep Kalathil

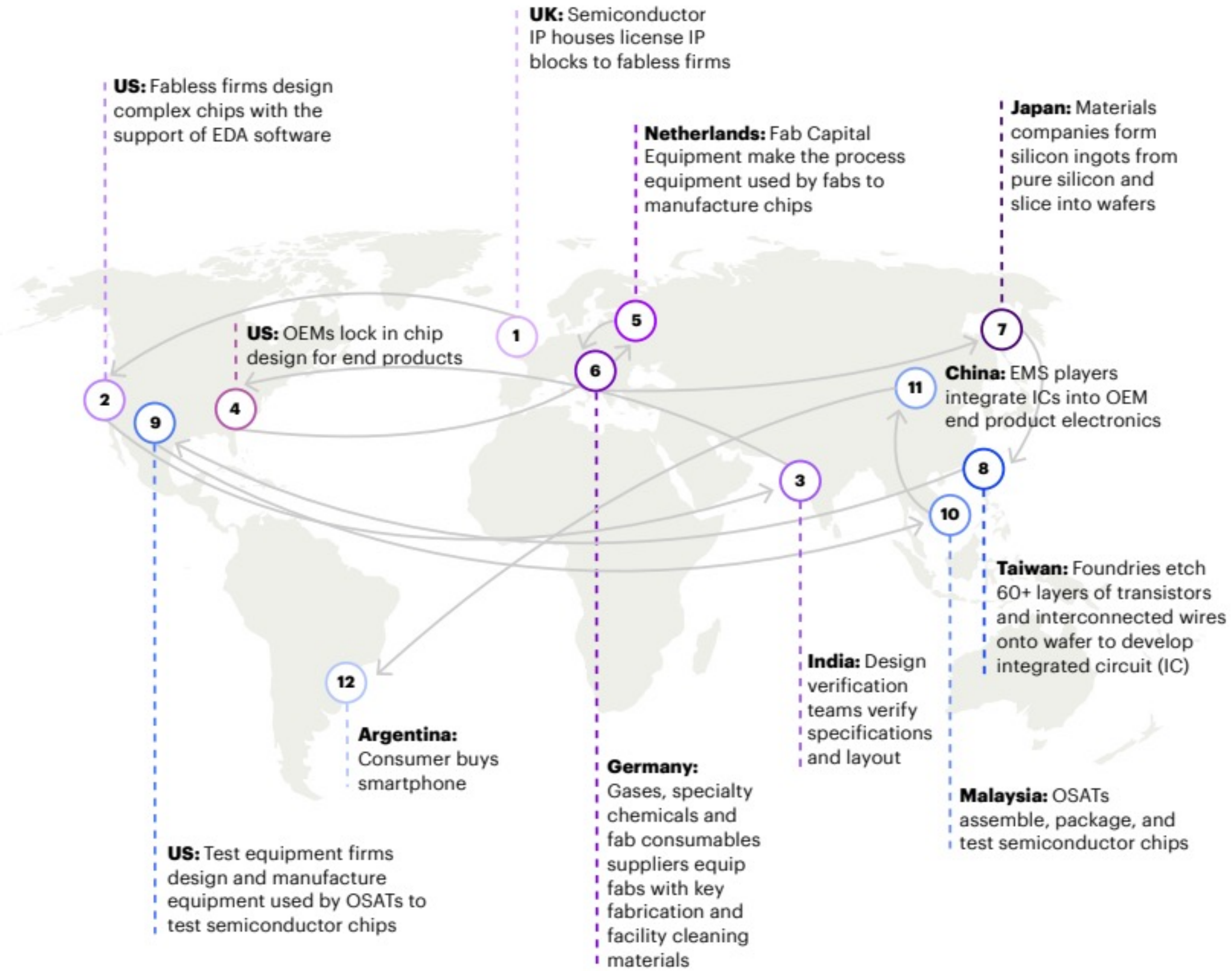
Jeyavijayan “JV” Rajendran

TEXAS A&M UNIVERSITY Engineering

University of Delaware

Hardware-focused Threats to Computing Systems

Due to Globalized Supply Chain

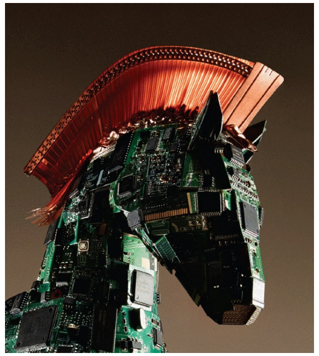


Hardware-focused Threats to Computing Systems

Real

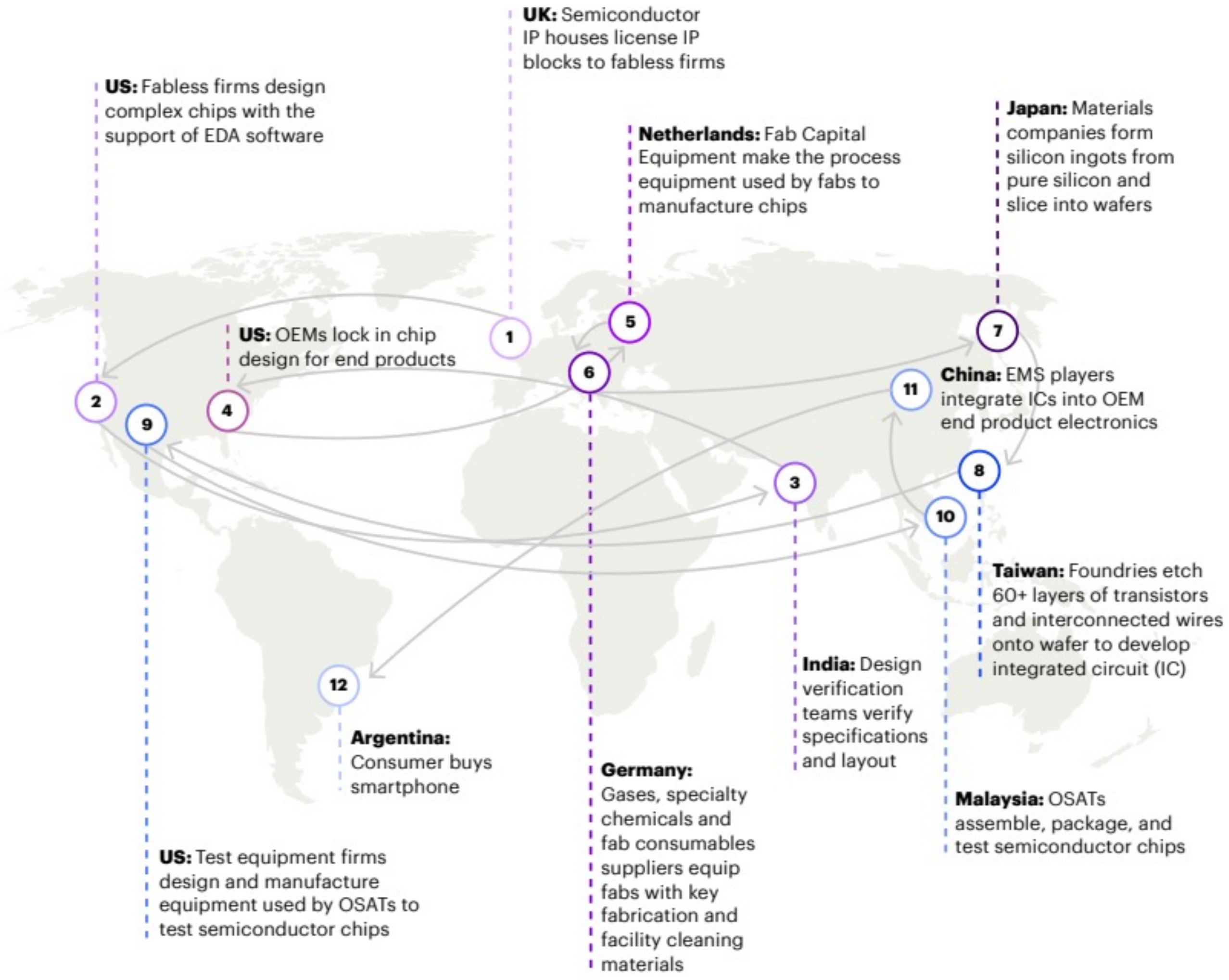
Fake

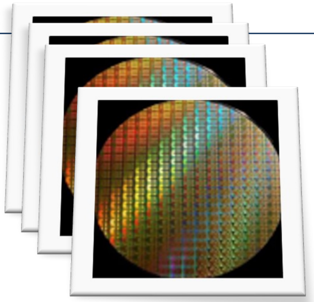
Counterfeiting



Hardware Trojans

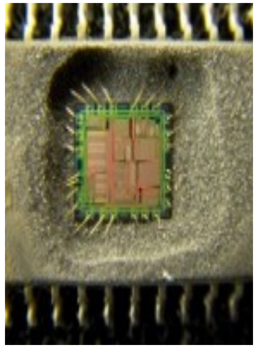
Due to Globalized Supply Chain





Overproduction

IP Piracy



Reverse Engineering

State-of-the-art GNNs in Hardware Security

Technique Type	Security Problem	Technique	GNN Framework	Claimed Efficacy
Defense	Detecting Trojans	GNN4TJ [1]	Attention-based custom GCN	97% TPR
	Locating Trojans	TrojanSAINT [2]	Graph attention network	98% TPR, 96% TNR
	Detecting IP Piracy	GNN4IP [3]	Attention-based custom GCN	94.61% Accuracy
Attack	Reverse Engineering	GNN-RE [4]	Graph attention network	98.87% Accuracy
	Hardware Obfuscation	OMLA [5]	Graph isomorphism network	89.55% Accuracy

State-of-the-art GNNs in Hardware Security

Are Graph Neural Networks (GNNs) Used To Solve Hardware Security Problems Robust?

Technique Type	Security Problem	Technique	GNN Framework	Claimed Efficacy
	Trojans	[2]	network	IR
	Detecting IP Piracy	GNN4IP [3]	Attention-based custom GCN	94.61% Accuracy
Attack	Reverse Engineering	GNN-RE [4]	Graph attention network	98.87% Accuracy
	Hardware Obfuscation	OMLA [5]	Graph isomorphism network	89.55% Accuracy

State-of-the-art GNNs in Hardware Security

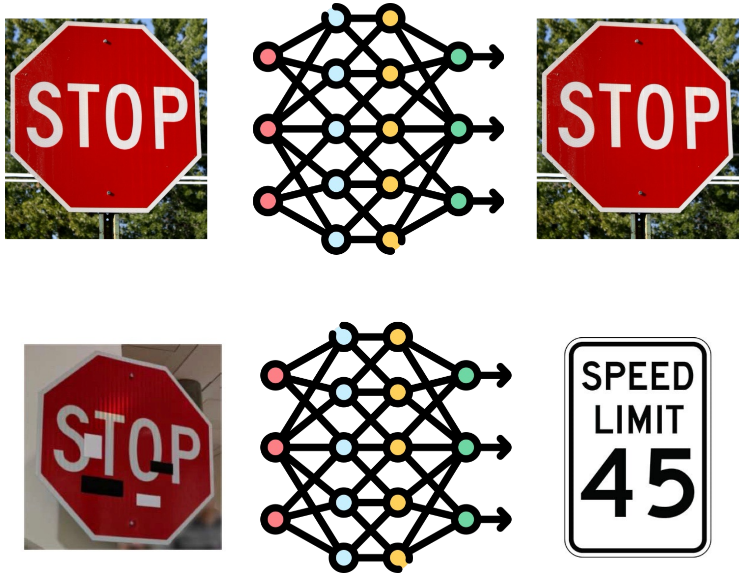
Technique Type	Security Problem	Technique	GNN Framework	Claimed Efficacy
	Trojans	[2]	network	
	Detecting IP	GNN4IP [3]	Attention-based	94.61% Accuracy
	Hardware Obfuscation	OMLA [5]	Graph isomorphism network	89.55% Accuracy

Are Graph Neural Networks (GNNs) Used To Solve Hardware Security Problems Robust?

AttackGNN: Red-Teaming GNNs in Hardware Security Using Reinforcement Learning

Threat Model

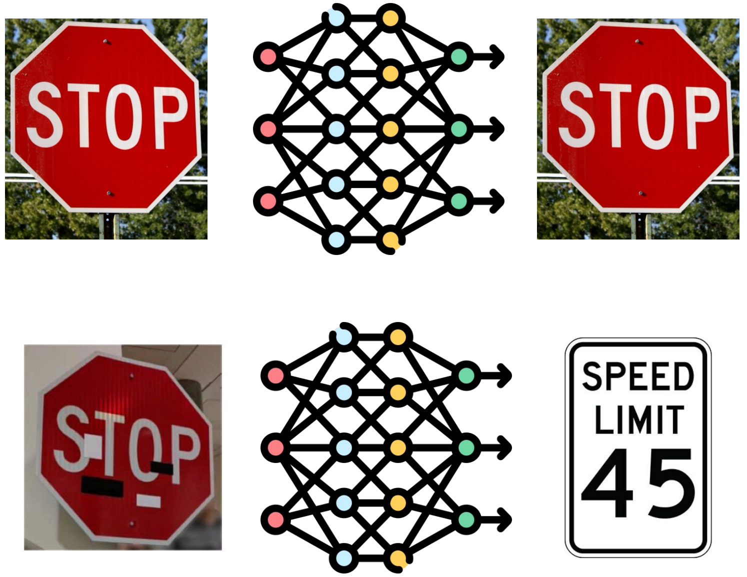
Standard attack model of adversarial attacks



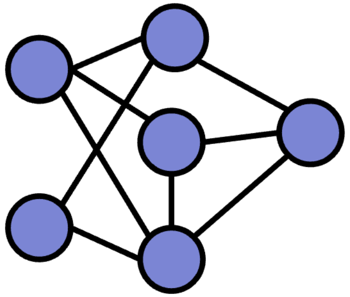
Kevin Eykholt et al., “Robust physical-world attacks on deep learning visual classification,” In Proc. of CVPR, 2018

Threat Model

Standard attack model of adversarial attacks



Kevin Eykholt et al., “Robust physical-world attacks on deep learning visual classification,” In Proc. of CVPR, 2018



Trained GNN

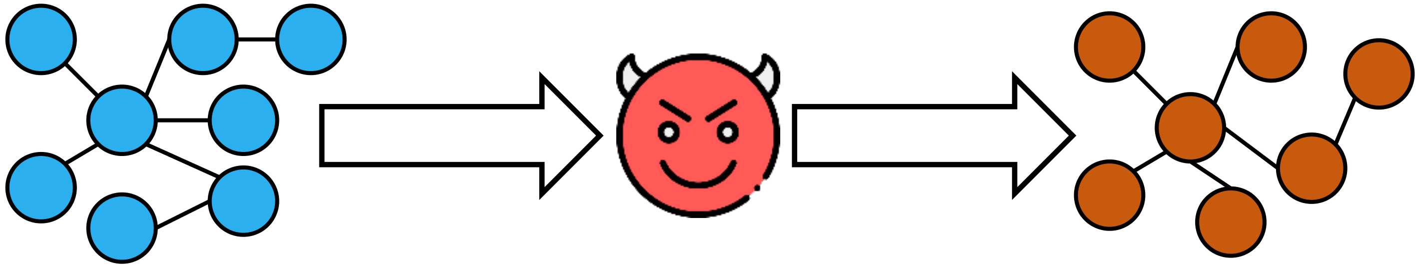
No Modifications

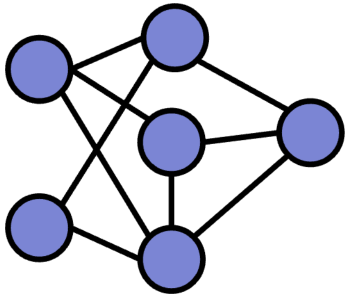
Threat Model

Standard attack model of adversarial attacks

Original Circuit

Perturbed Circuit

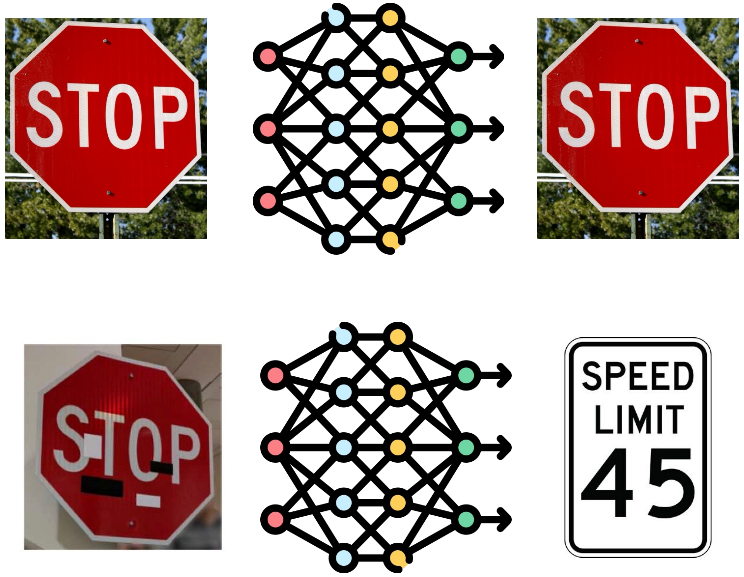




Trained GNN

No Modifications

Perturbations Following Circuit Design Rules



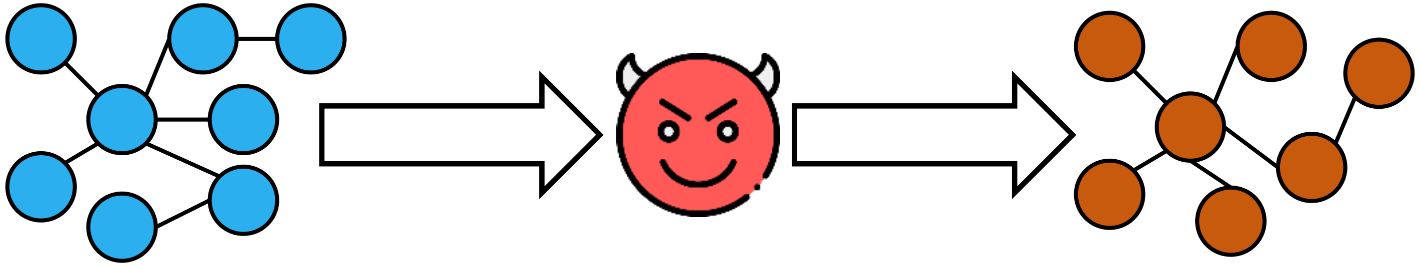
Kevin Eykholt et al., “Robust physical-world attacks on deep learning visual classification,” In Proc. of CVPR, 2018

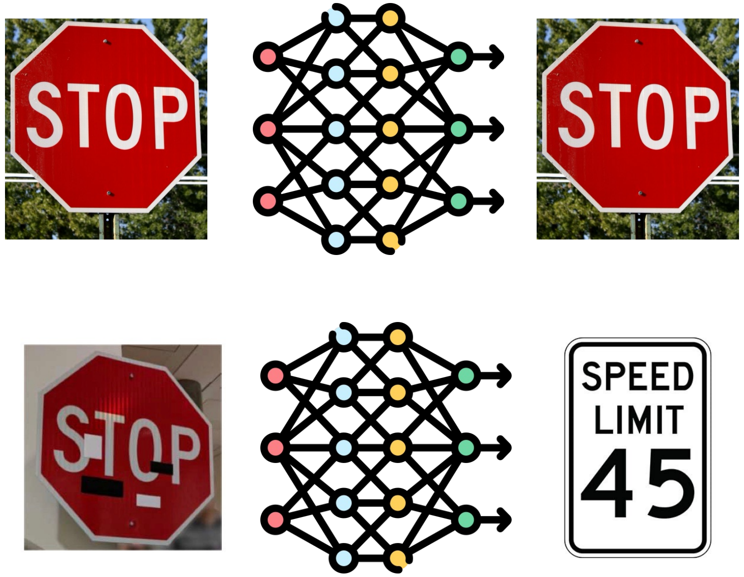
Threat Model

Standard attack model of adversarial attacks

Original Circuit

Perturbed Circuit





Kevin Eykholt et al., "Robust physical-world attacks on deep learning visual classification," In Proc. of CVPR, 2018

Trained GNN

No Modifications

Perturbations Following Circuit Design Rules

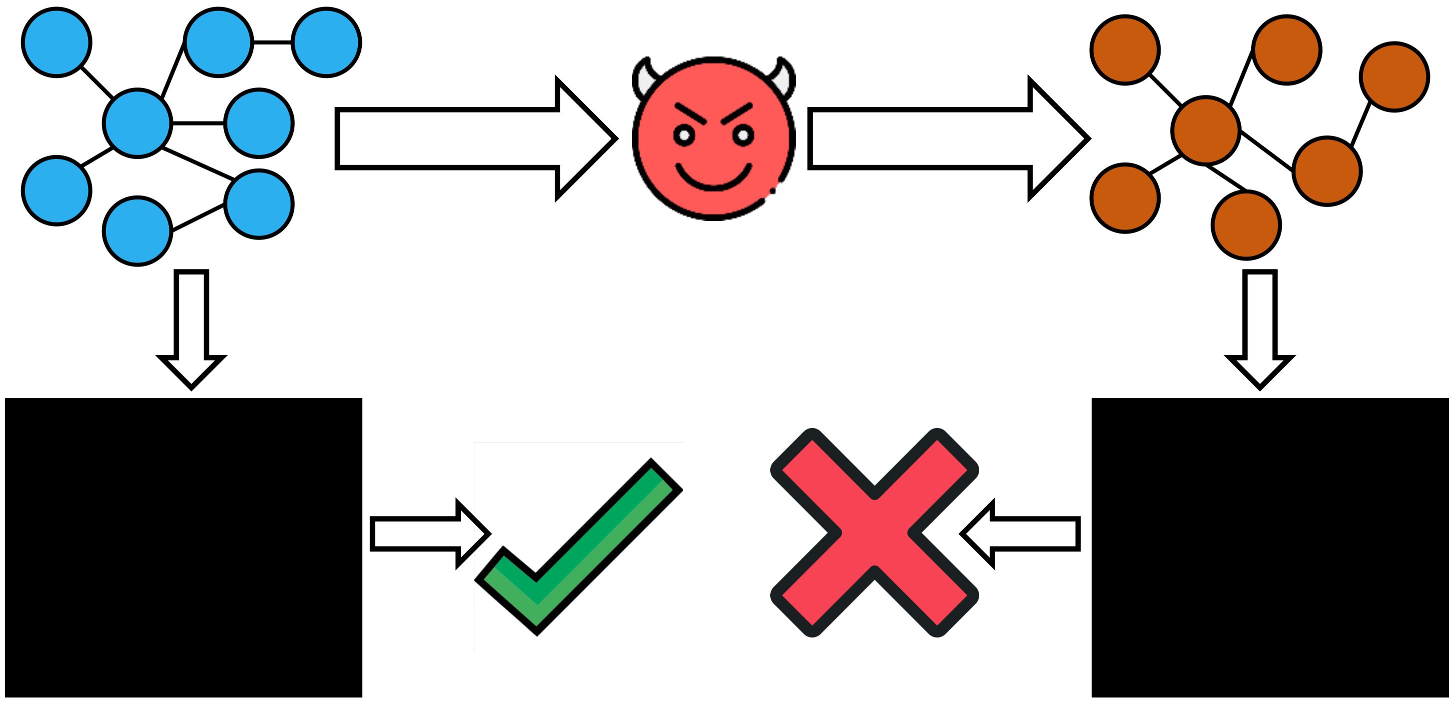
Black-box Access

Threat Model

Standard attack model of adversarial attacks

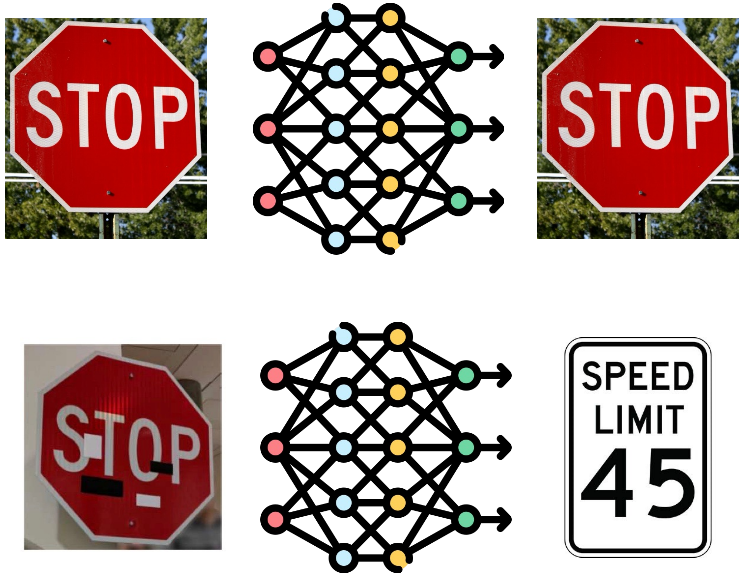
Original Circuit

Perturbed Circuit



Trained GNN

Same GNN



Kevin Eykholt et al., "Robust physical-world attacks on deep learning visual classification," In Proc. of CVPR, 2018

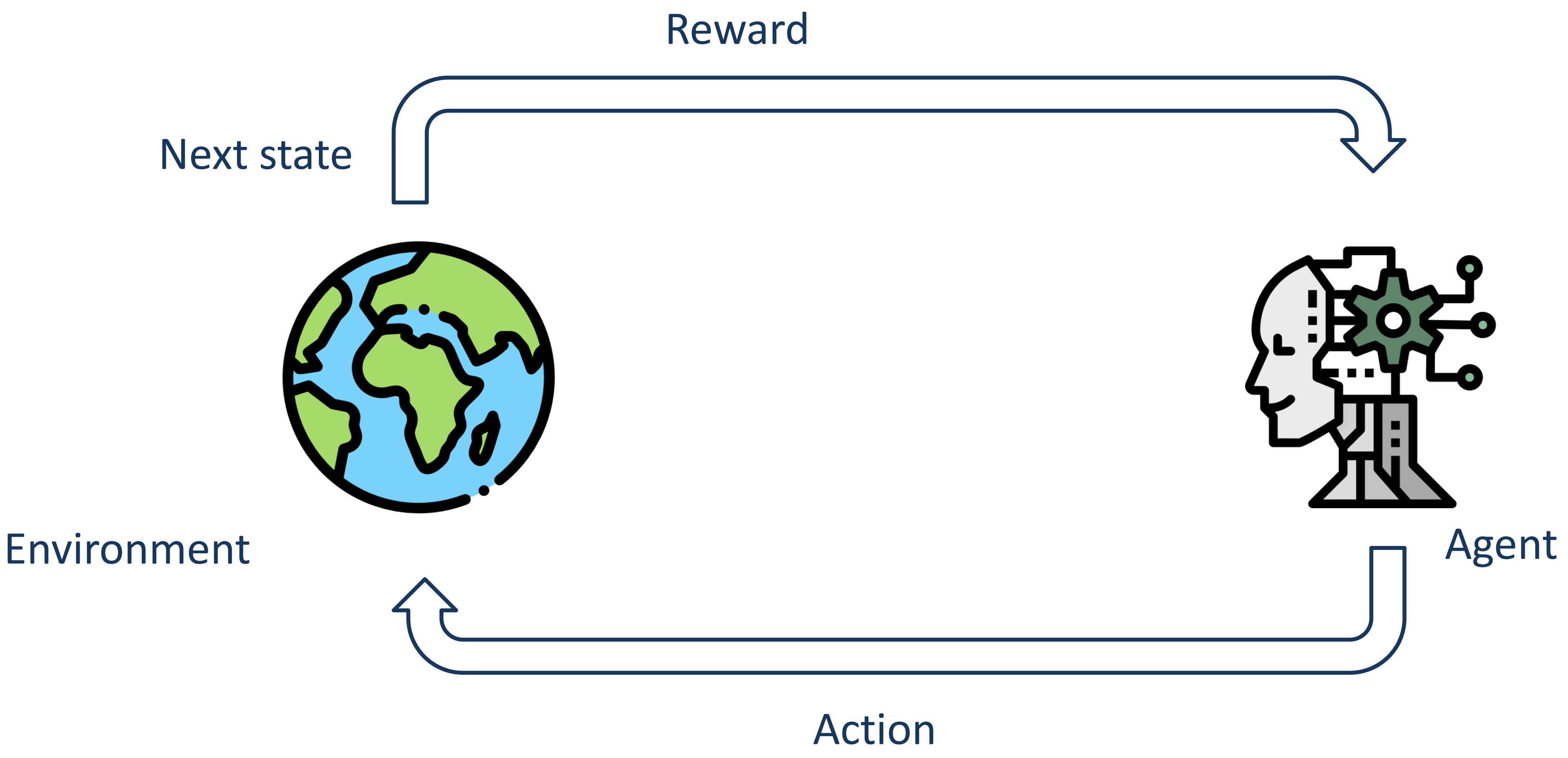
Goal: Misclassification

No Modifications

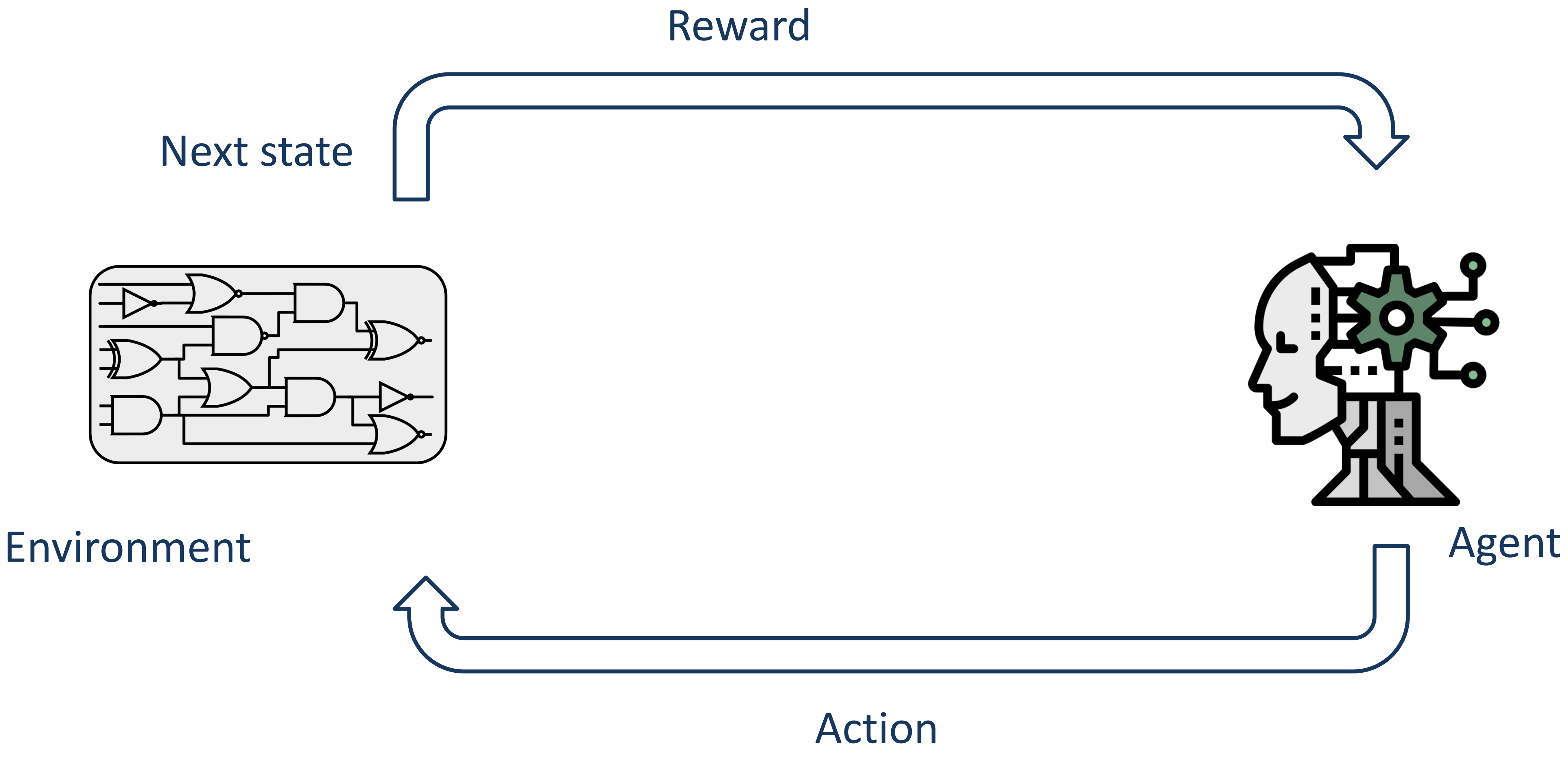
Perturbations Following Circuit Design Rules

Black-box Access

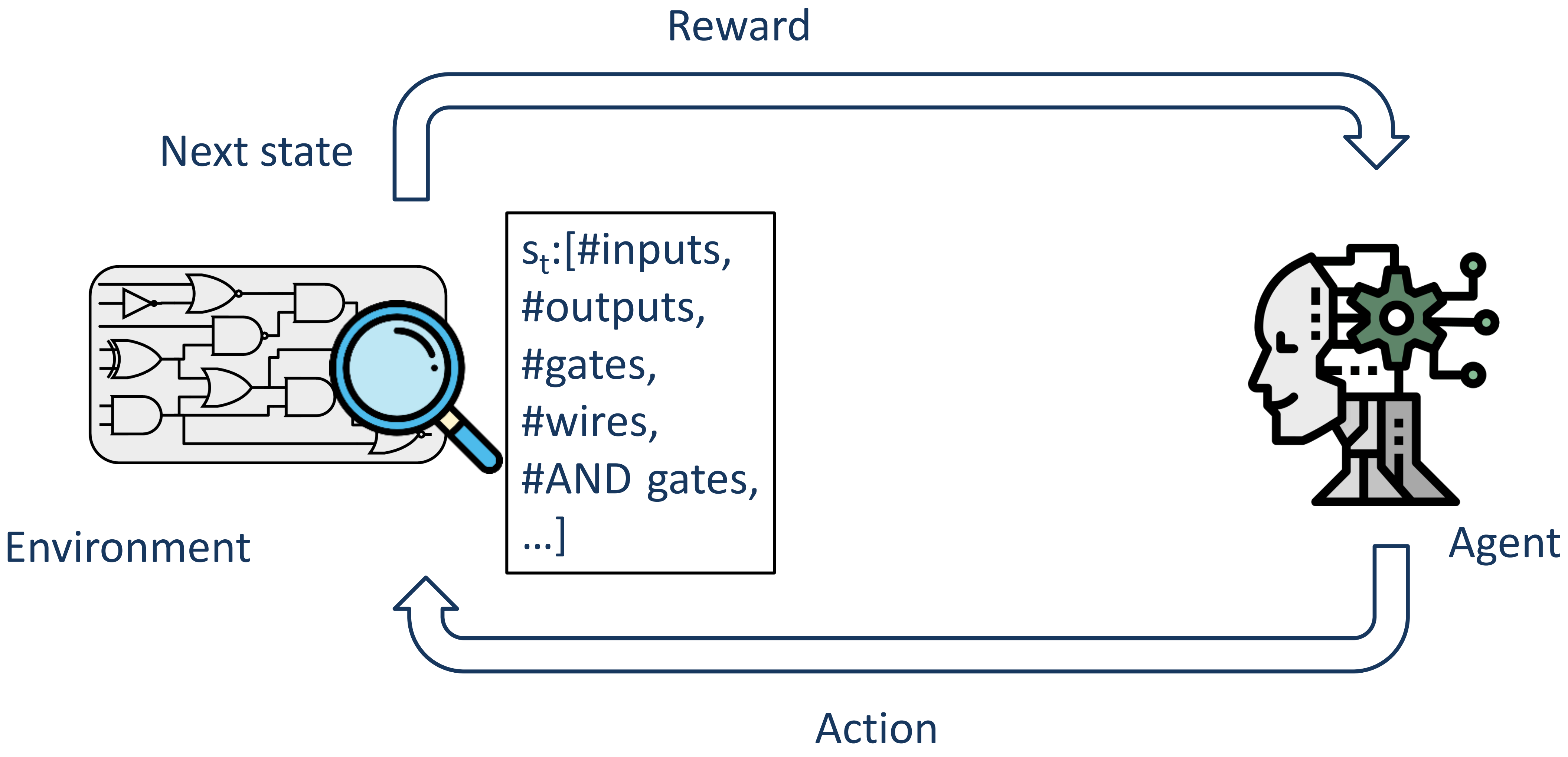
AttackGNN – Preliminary Formulation



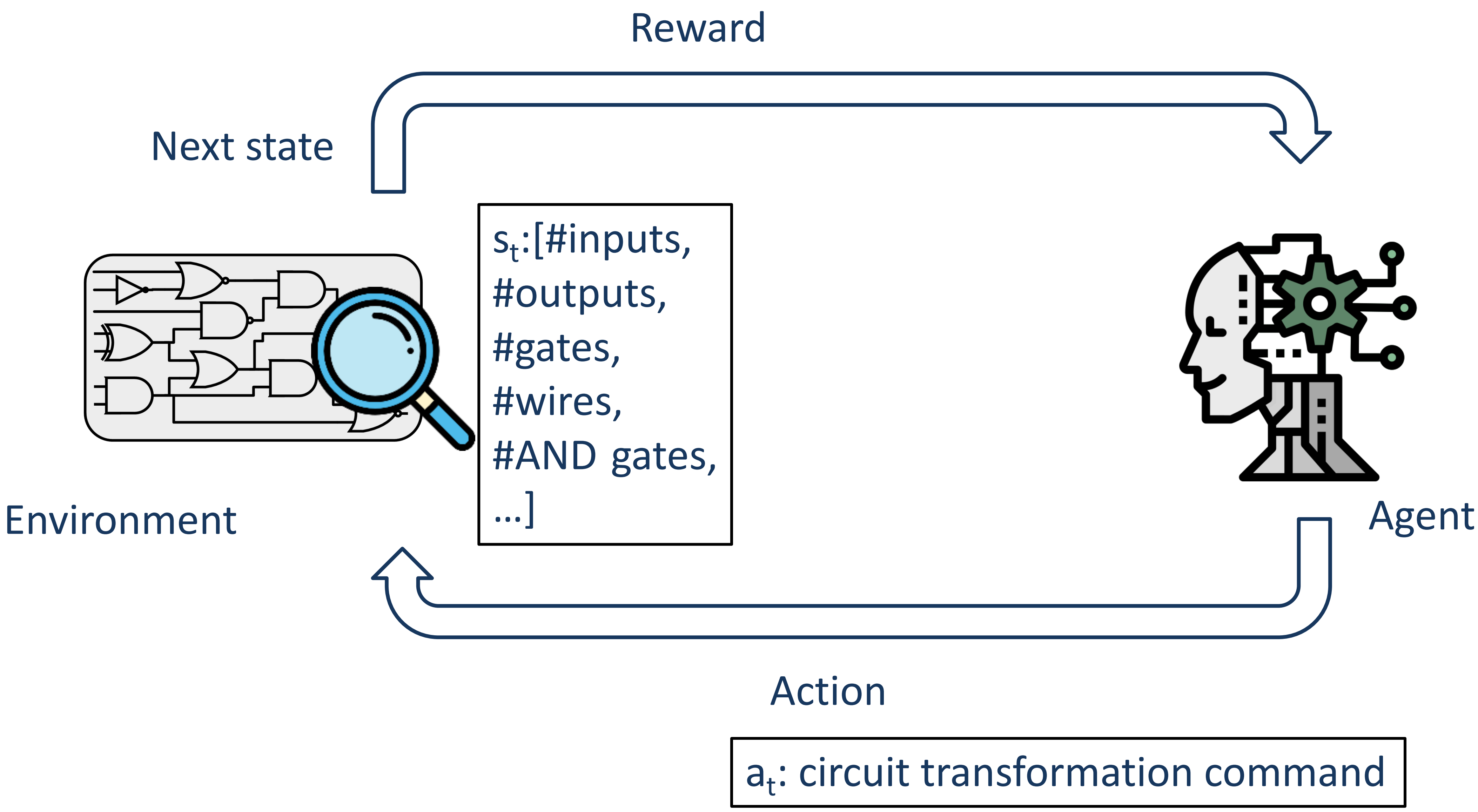
AttackGNN – Preliminary Formulation



AttackGNN – Preliminary Formulation



AttackGNN – Preliminary Formulation

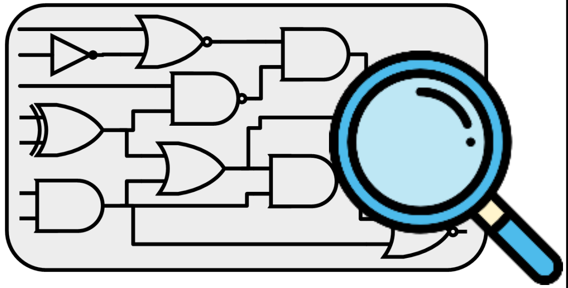


AttackGNN – Preliminary Formulation

$$r_t = \begin{cases} \boldsymbol{\alpha}\ (> \mathbf{0}) & \text{if next state is misclassified} \\ \mathbf{0} & \text{else} \end{cases}$$

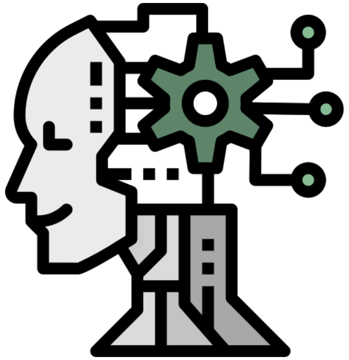
Reward

Next state



Environment

s_t:[#inputs,
#outputs,
#gates,
#wires,
#AND gates,
…]



Agent

Action

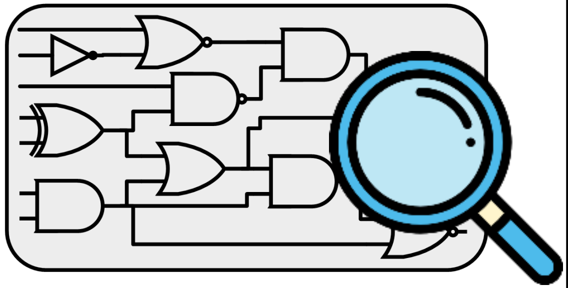
a_t: circuit transformation command

AttackGNN – Preliminary Formulation

$$r_t = \begin{cases} \boldsymbol{\alpha}\ (> \mathbf{0}) & \text{if next state is misclassified} \\ \mathbf{0} & \text{else} \end{cases}$$

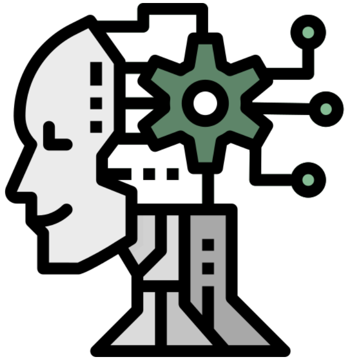
Reward

Next state



Environment

s_t:[#inputs, #outputs, #gates, #wires, #AND gates, …]

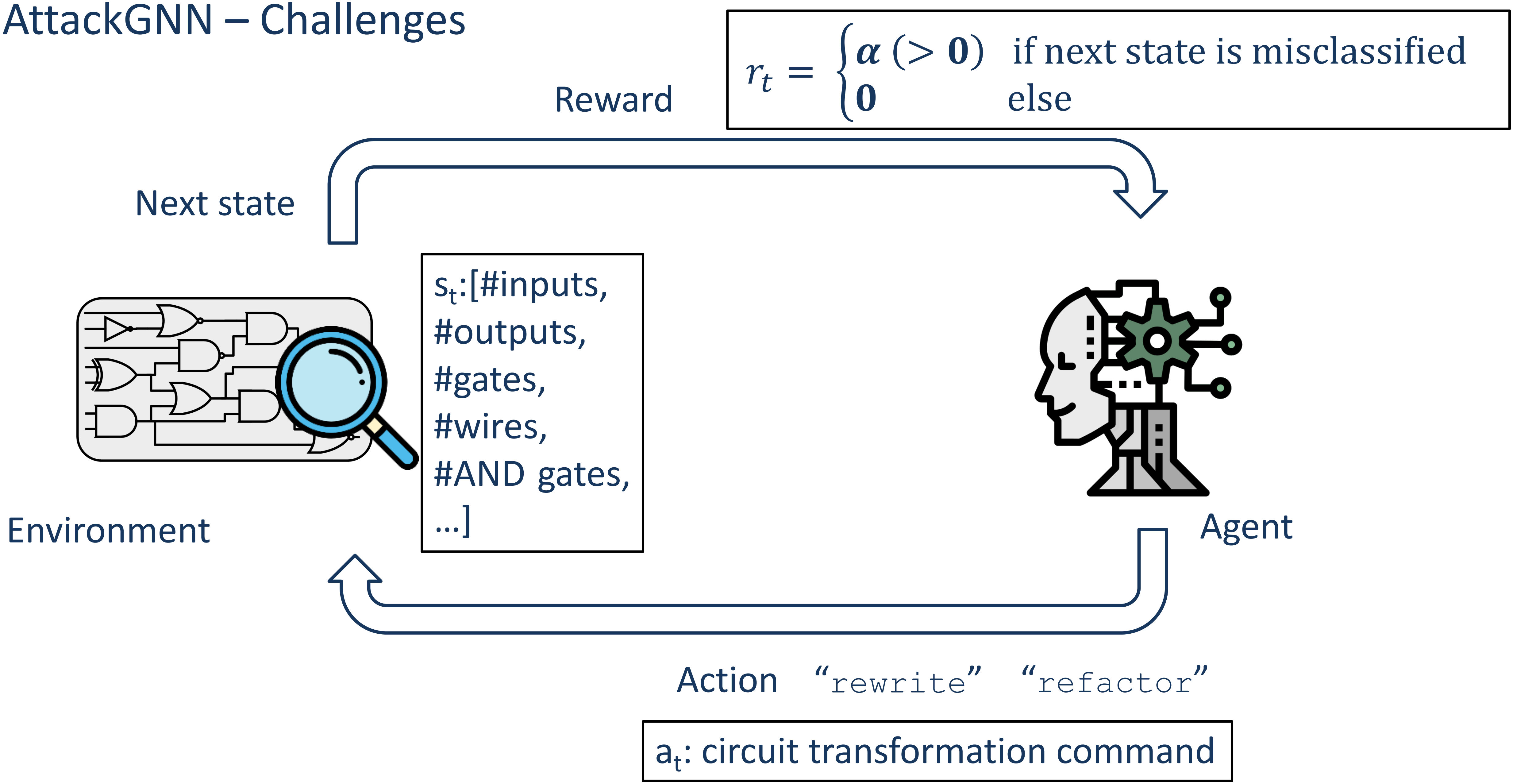


Agent

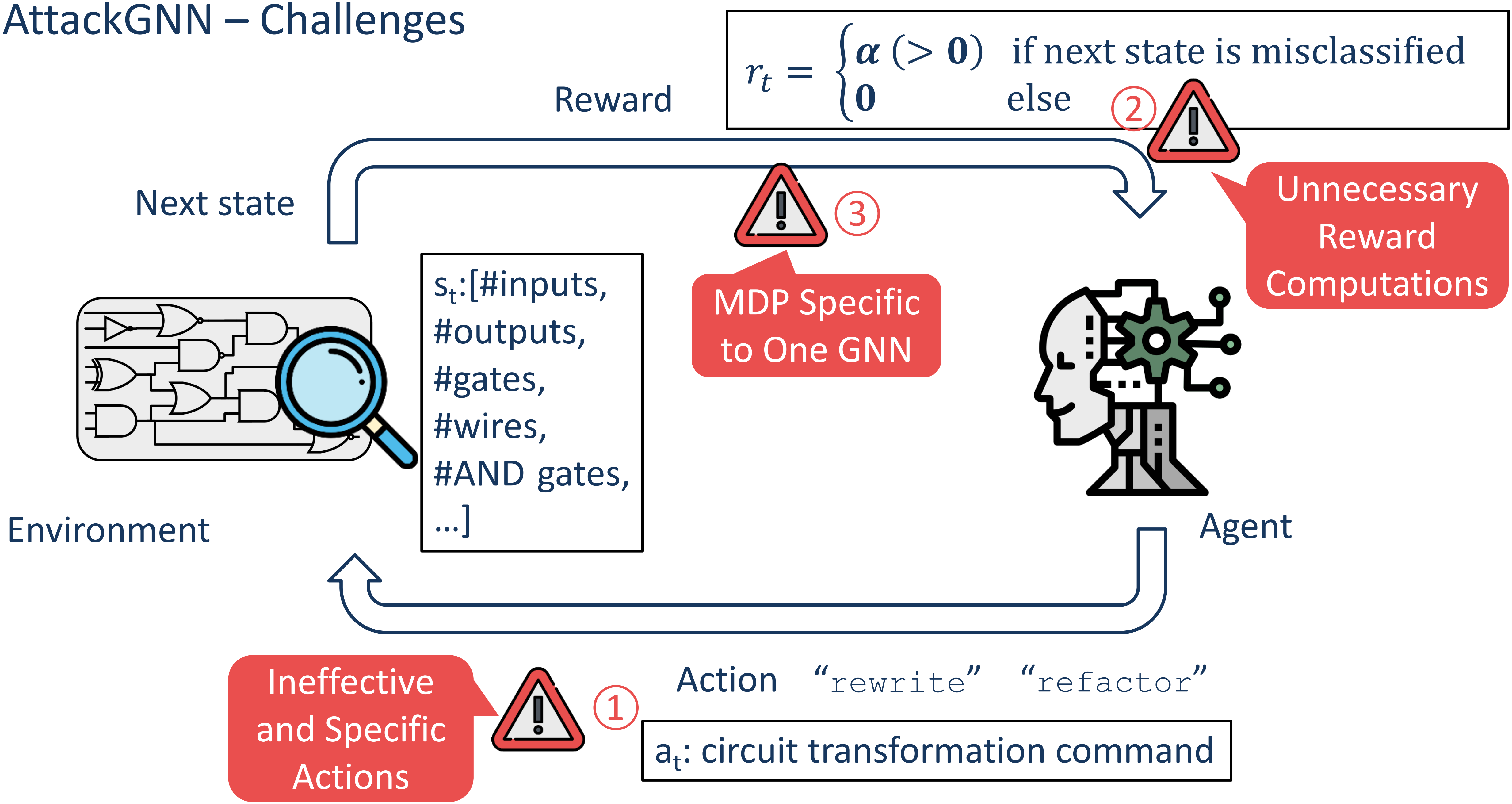
Action “`rewrite`” “`refactor`”

a_t: circuit transformation command

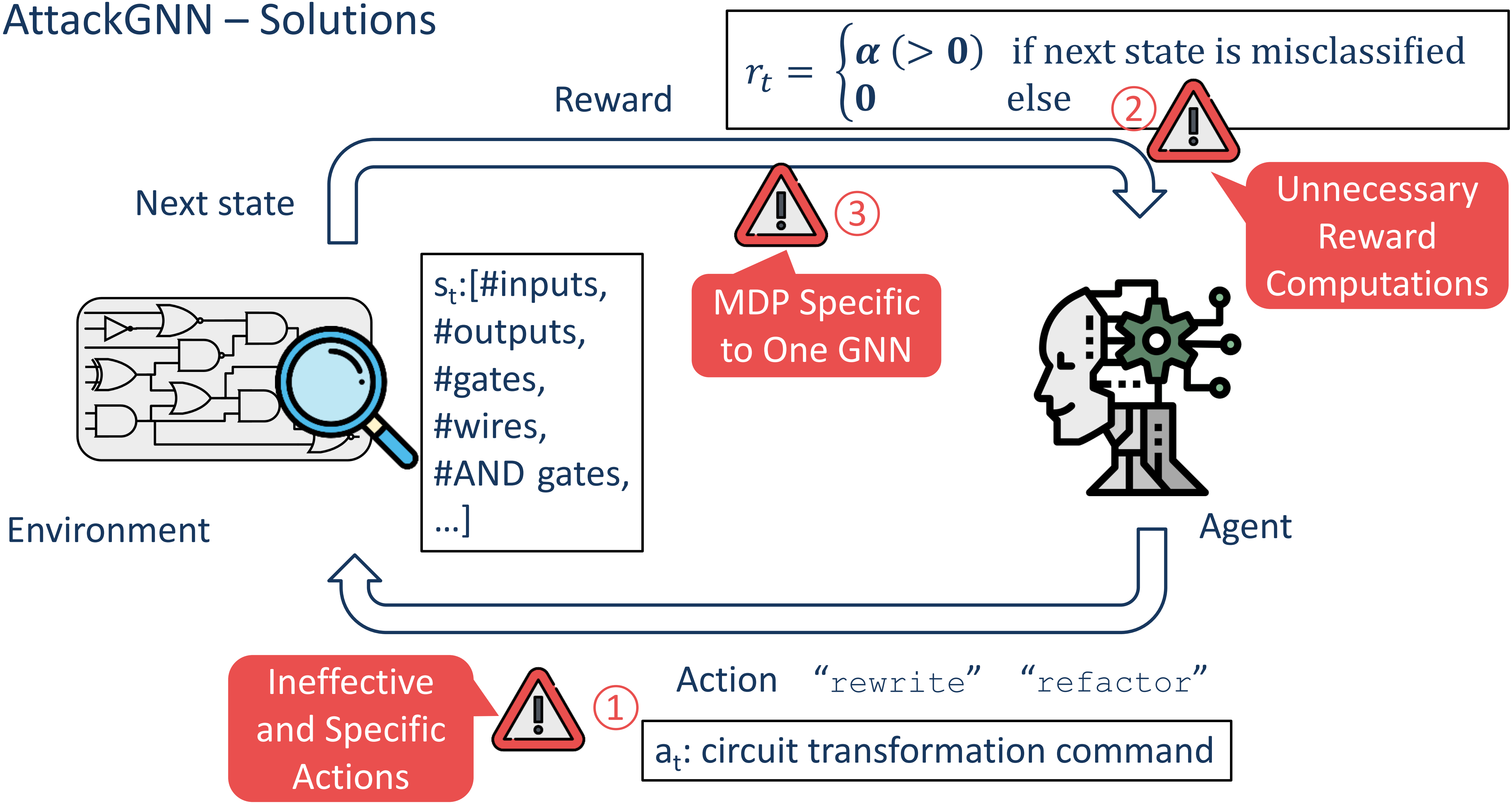
AttackGNN – Challenges



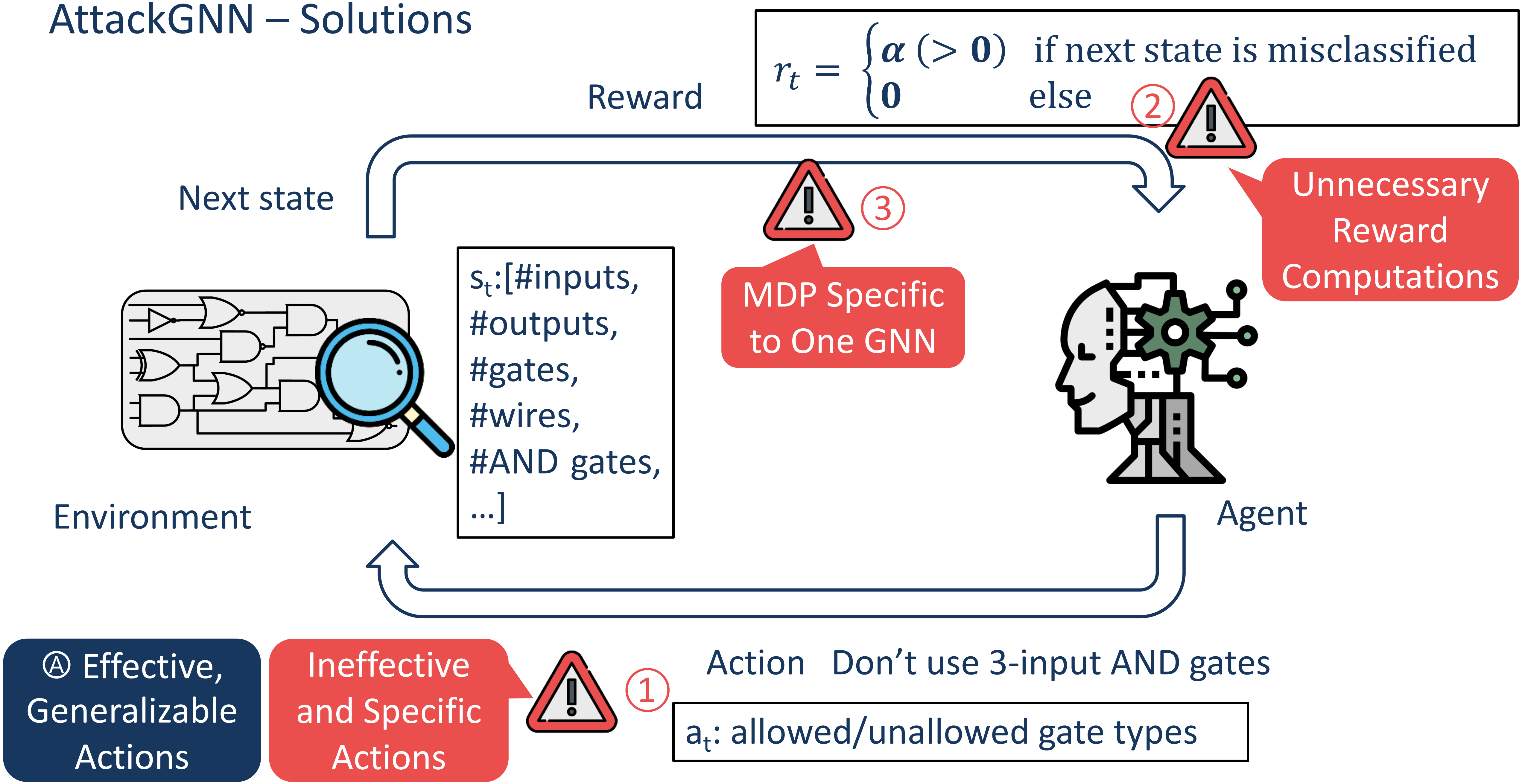
AttackGNN – Challenges



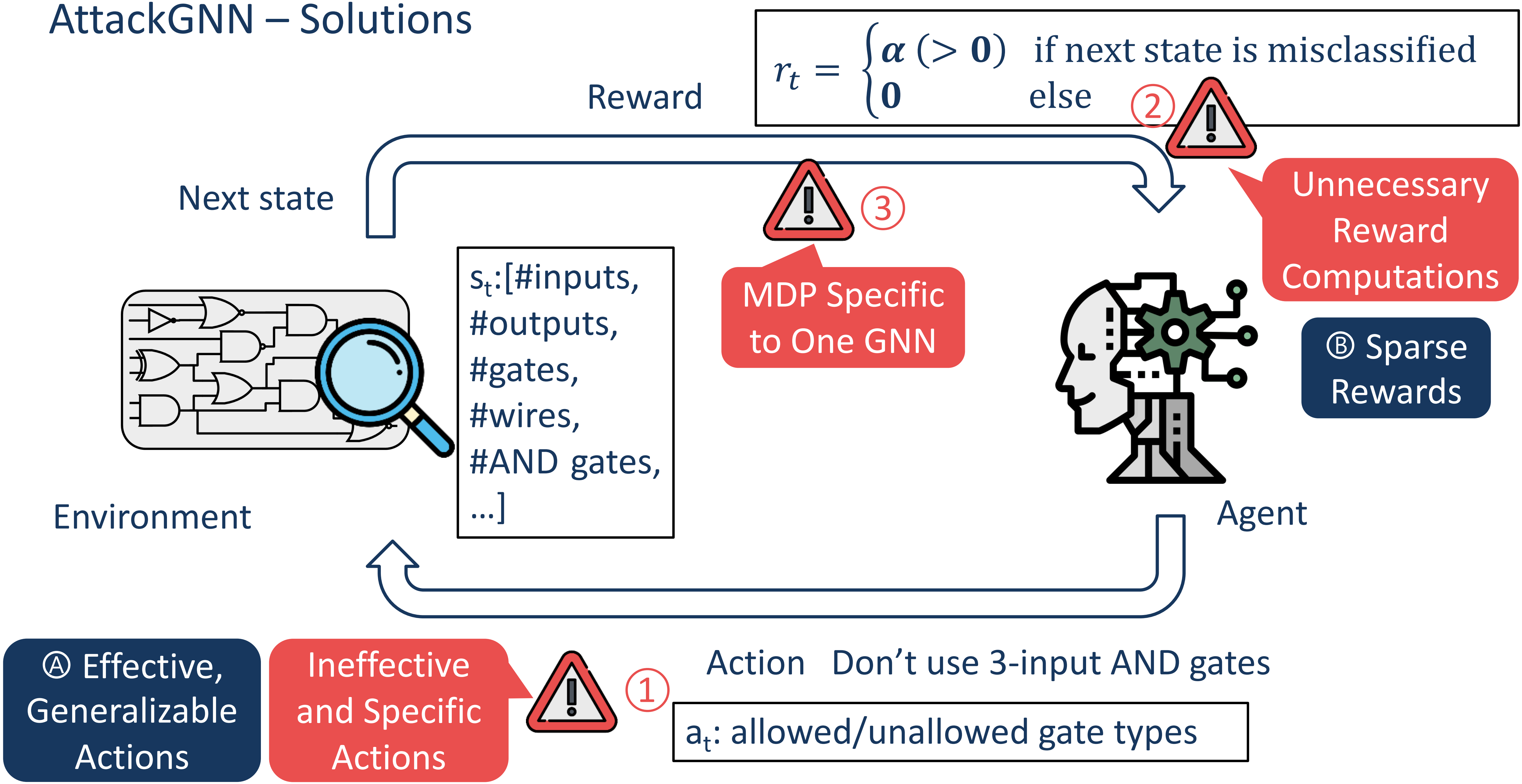
AttackGNN – Solutions



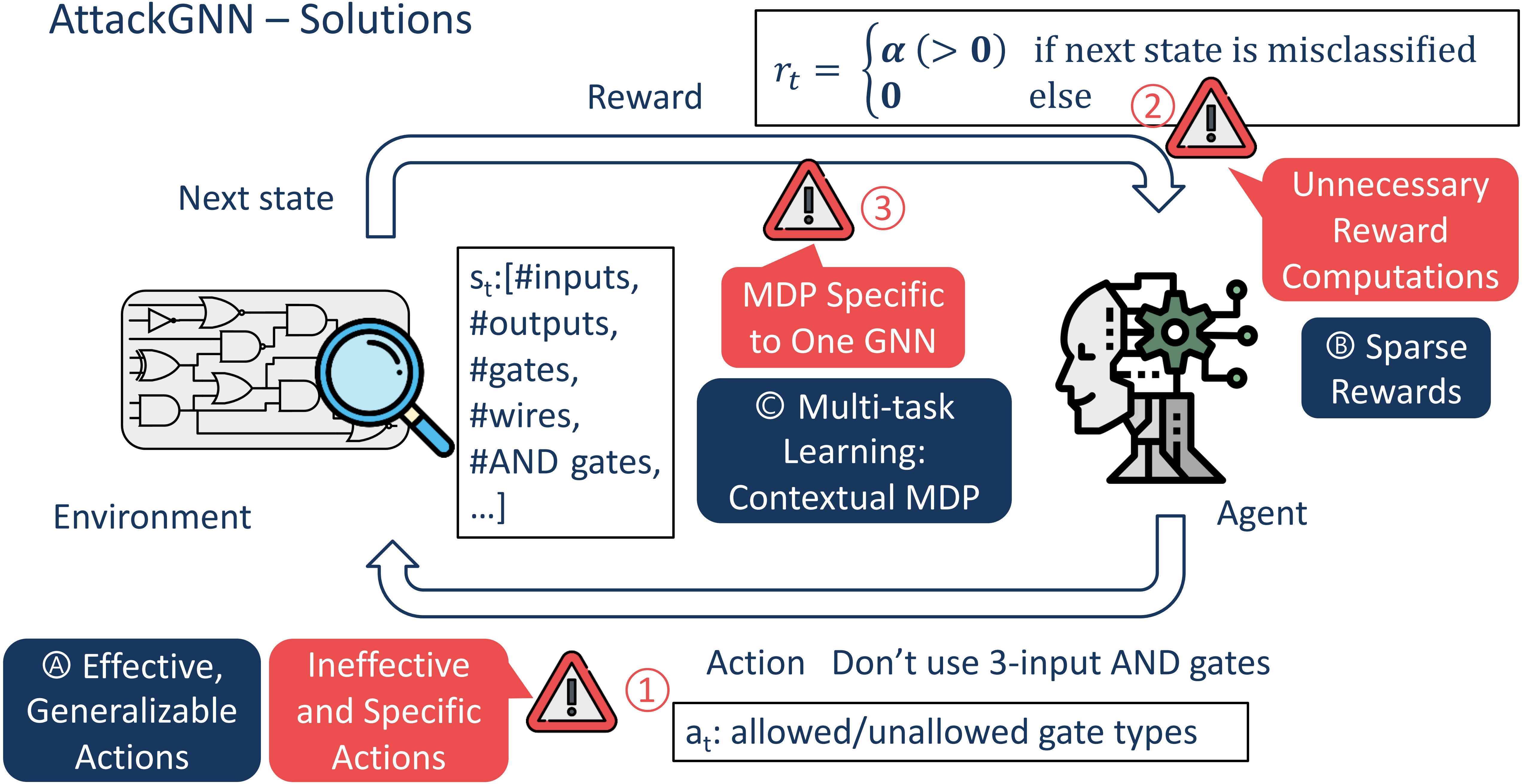
AttackGNN – Solutions



AttackGNN – Solutions

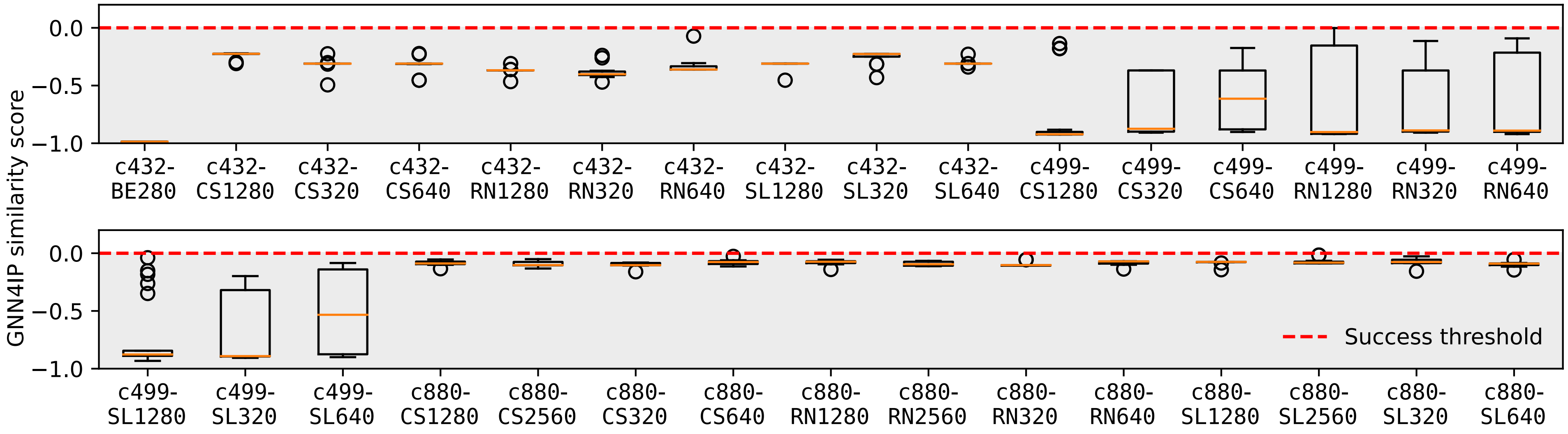


AttackGNN – Solutions



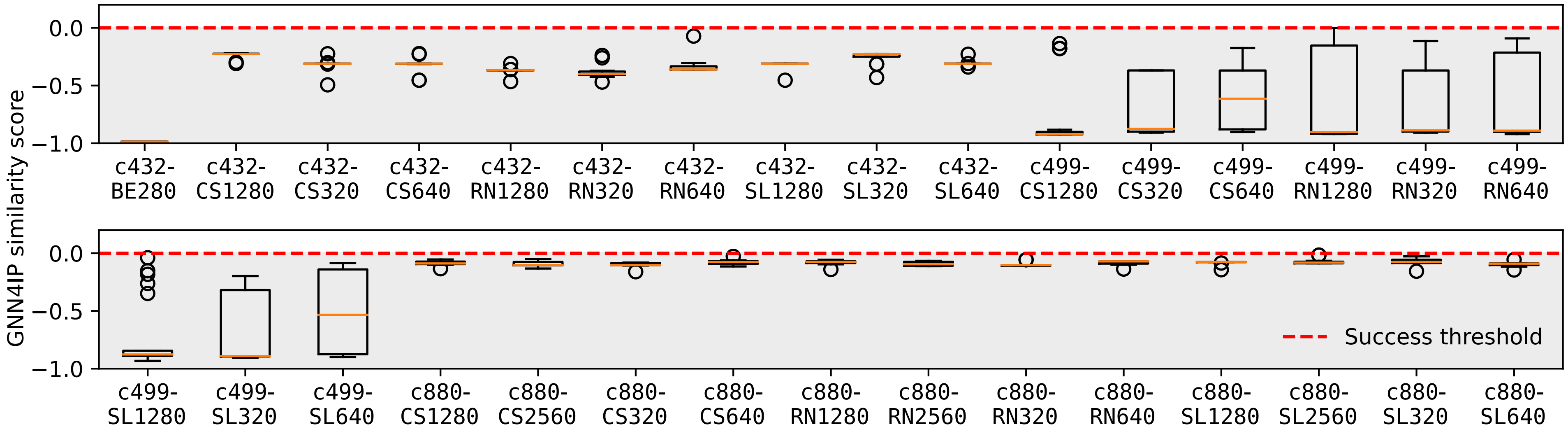
AttackGNN – Results

Against GNN4IP (IP Piracy Detection GNN)

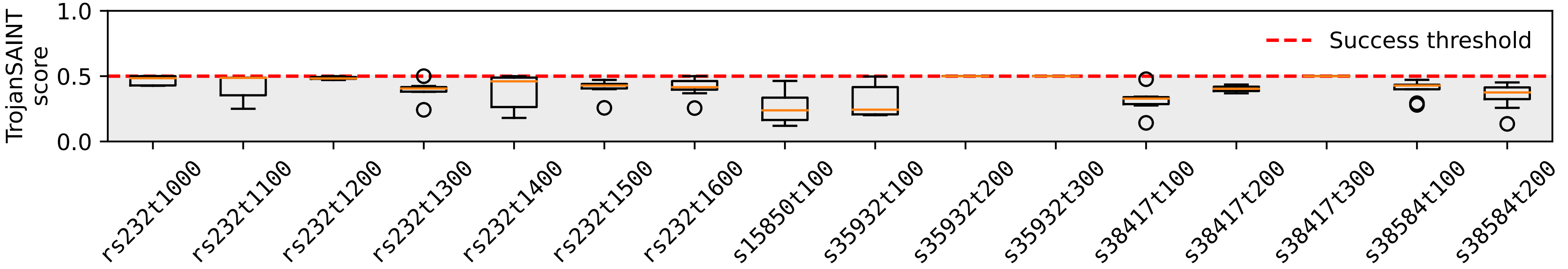


AttackGNN – Results

Against GNN4IP (IP Piracy Detection GNN)

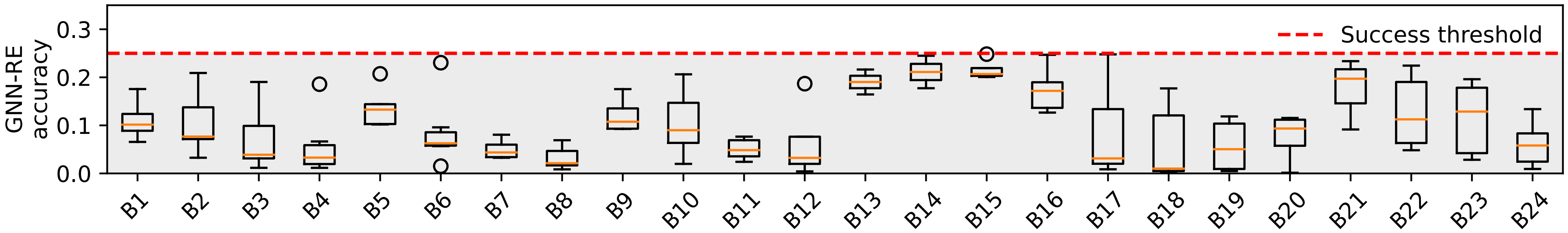


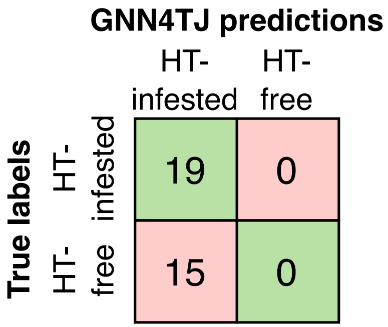
Against TrojanSAINT (Trojan Locator GNN)



AttackGNN – Results

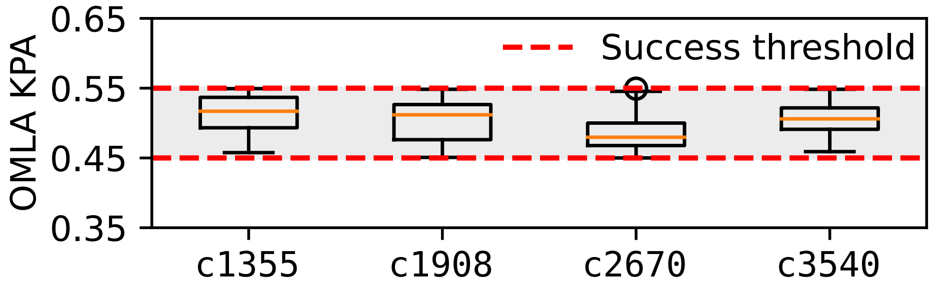
Against GNN-RE
(Reverse Eng. GNN)





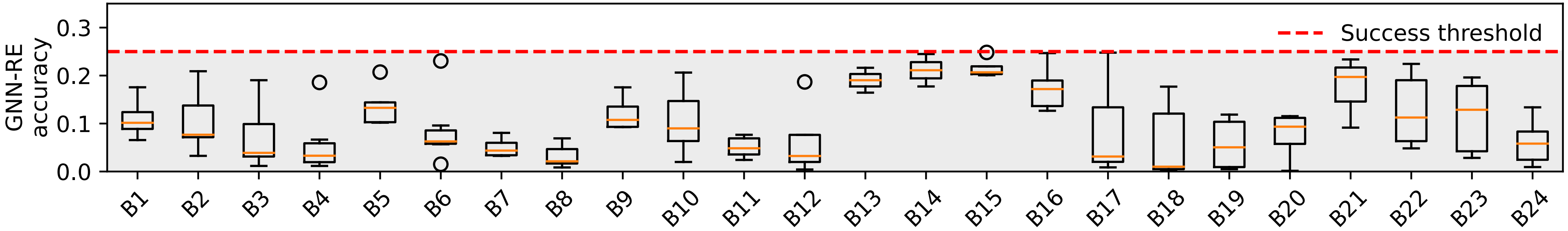
Against GNN4TJ
(Trojan Detector GNN)

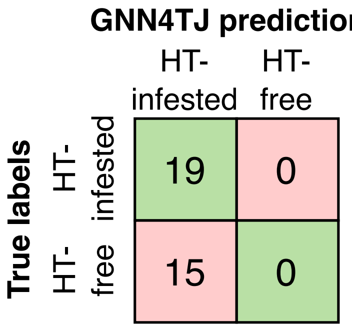
Against OMLA
(De-obfuscation GNN)



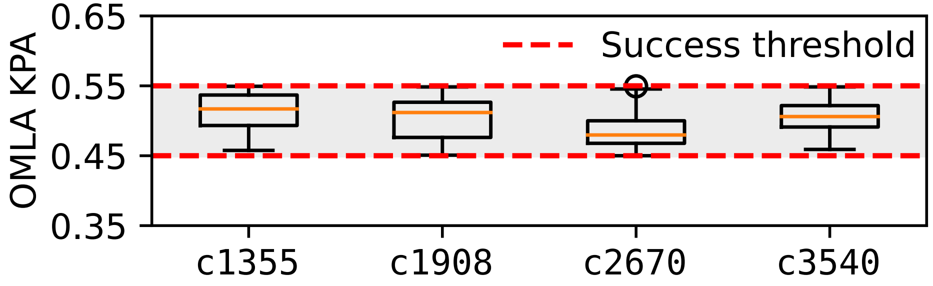
AttackGNN – Results

Against GNN-RE
(Reverse Eng. GNN)





Against OMLA
(De-obfuscation GNN)



Against GNN4TJ
(Trojan Detector GNN)

Success rate of all GNNs against AttackGNN-generated adversarial circuits: **0%**

GNNs used in hardware security are **not robust**!

Thank You

Vasudev Gohil
vasudevgohil.com

Secure and Trustworthy Hardware (SETH) Lab
https://seth.engr.tamu.edu
Texas A&M University

References

[1] Yasaei, Rozhin, Shih-Yuan Yu, and Mohammad Abdullah Al Faruque. "Gnn4tj: Graph neural networks for hardware trojan detection at register transfer level." In Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1504-1509, IEEE, 2021.

[2] Lashen, Hazem, Lilas Alrahis, Johann Knechtel, and Ozgur Sinanoglu. "TrojanSAINT: Gate-level netlist sampling-based inductive learning for hardware Trojan detection."arXiv preprint arXiv:2301.11804, 2023.

[3] Yasaei, Rozhin, Shih-Yuan Yu, Emad Kasaeyan Naeini, and Mohammad Abdullah Al Faruque. "GNN4IP: Graph neural network for hardware intellectual property piracy detection." In 58th ACM/IEEE Design Automation Conference (DAC), pp. 217-222, IEEE, 2021.

[4] Alrahis, Lilas, Abhrajit Sengupta, Johann Knechtel, Satwik Patnaik, Hani Saleh, Baker Mohammad, Mahmoud Al-Qutayri, and Ozgur Sinanoglu. "GNN-RE: Graph neural networks for reverse engineering of gate-level netlists." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 41, no. 8: 2435-2448, 2021.

[5] Alrahis, Lilas, Satwik Patnaik, Muhammad Shafique, and Ozgur Sinanoglu. "OMLA: An oracle-less machine learning-based attack on logic locking." IEEE Transactions on Circuits and Systems II: Express Briefs 69, no. 3: 1602-1606, 2021.