29th Asia and South Pacific Design Automation Conference
Jan. 22-25, 2024
Incheon Songdo Convensia, South Korea

LIPSTICK: Corruptibility-Aware and Explainable Graph Neural Network-based Oracle-Less Attack on Logic Locking

Yeganeh Aghamohammadi & Amin Rezaei

CONTENTS

01 Introduction

02 Preliminary Study

03 LIPSTICK Attack

04 Experimental Results

05 Conclusion

01
Introduction

1.1. Background
1.2. Research Gaps
1.3. Research Questions
1.4. Contributions

1.1. Background

Outsource manufacturing → Hardware IP theft and overproduction

Logic locking → Adding extra key-controlled gates to the circuits

Types of attacks → Oracle Guided (OG) and Oracle Less (OL)

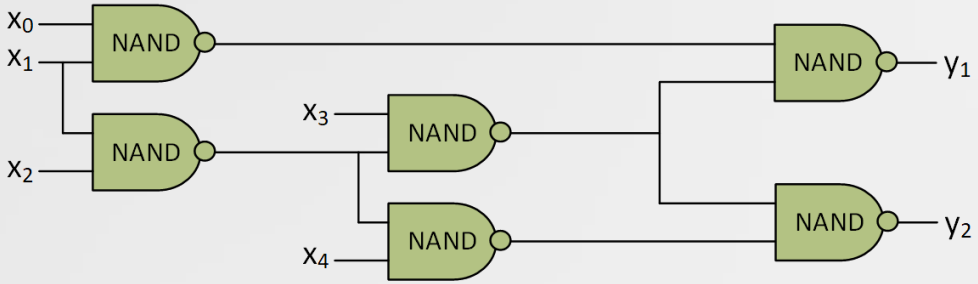
Machine learning → Facilitates attacks (especially OL attacks)

GNN model → Handles non-Euclidean data

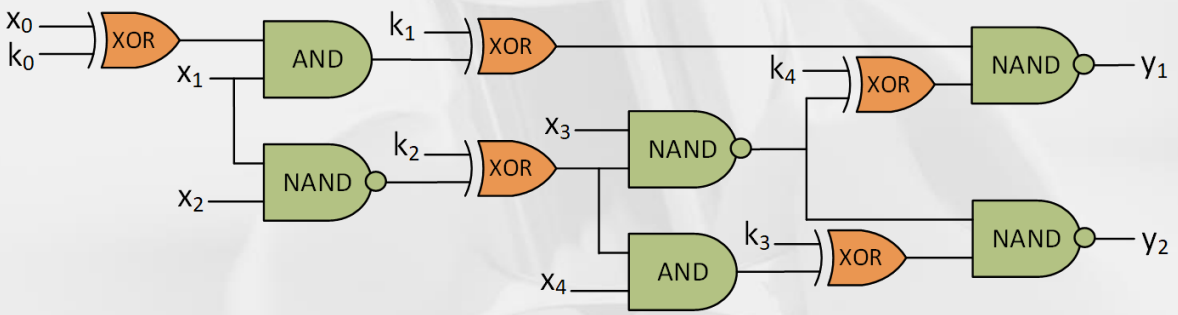
1.1. Background - Example

Traditional logic locking*:

$$\mathbf{K}^* = k_4^* \, k_3^* \, k_2^* \, k_1^* \, k_0^* = 01010$$



Original circuit f(X)



Locked circuit g(X,K)

$$\exists K^* : g(X, K^*) \equiv f(X)$$

* J. A. Roy et al., "EPIC: Ending piracy of integrated circuits," In DATE, 2008.

1.2. Research Gaps

Observation 1: ML-based OL attacks are inherently **approximate** attacks. → SOTA OL attacks try to find a "good enough" key. → The more "similar" to the correct key, the better.

Is it?

Observation 2: SOTA OL attacks do not consider the behavior of the circuit under the reported key compared with the intended functionality. → The "good enough" definition of the key does not consider output corruptibility.

Observation 3: A holistic security assessment of logic locking techniques is overlooked. → SOTA methods assume that the higher the prediction accuracy, the better the attack (and the worse the logic locking scheme).

Is it enough?

1.3. Research Questions

Question 1: Why does the **accuracy** of SOTA GNN attacks differ drastically from the reported key's **precision**?

Sub question: Can integrating circuit functionality metrics into GNN models result in the discovery of a more relevant key?

Question 2: What features of the logic-locked circuits let the model infer the reported key?

Sub question: What is the degree of each feature's influence?

1.4. Contributions

Contribution 1: Proposing an effective GNN-based OL attack on logic locking that takes the circuit's functionality into account in addition to its structure.

Contribution 2: Providing explainability of the inferred key by the proposed attack that functions as a rule-of-thumb for designers on how to safeguard their precious hardware designs.

Contribution 3: Showcasing the model's prediction accuracy and key precision on seen and unseen logic-locked benchmarks.

02
Preliminary Study

2.1. Definitions
2.2. SOTA Accuracy Metric
2.3. SOTA GNN Attacks

2.1. Definitions

Prediction accuracy: It resembles how well a given prediction matches its actual value.

Key precision: It shows how closely a locked circuit under a given key operates to the original circuit.

Logic locking problem statement:

Original circuit: $F = \{0,1\}^n \rightarrow \{0,1\}^m$

Locked circuit: $G = \{0,1\}^p \times \{0,1\}^n \rightarrow \{0,1\}^m$

$$\exists\, K^* = \left(k^*_{p-1}, k^*_{p-2}, \dots, k^*_1, k^*_0\right): \{0,1\}^p \mid G(X, K^*) \equiv F(X)$$

Key error rate: $ER(K^a)$ is the number of input patterns in which $G(X, K^a) \neq F(X)$ divided by all the patterns.

Note 1: $ER(K^*) = 0$

Note 2: The key error rate is fundamentally dependent on the circuit's **functionality** rather than its structure.

Key hamming distance: $HD(K^a, K^*) = \sum_{i=0}^{p-1}(k^a_i \oplus k^*_i): \{0,1, \dots p\}$

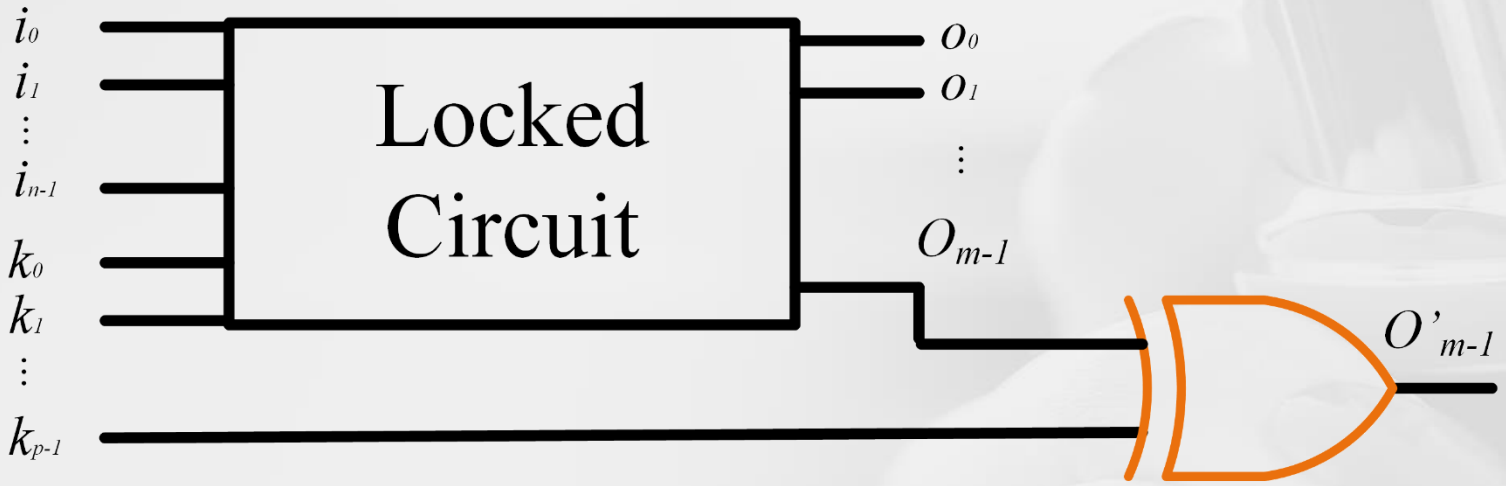
The SOTA assumption of a “good enough” key is having low HD.

2.2. (Critique of the) SOTA Accuracy Metric

Proposition: The smaller the $\mathrm{HD}(\mathrm{K}^{\mathrm{a}}, \mathrm{K}^{*})$ the higher the key precision of the locked circuit G under K^{a}.

Counterexample:

- Consider a locked circuit with a key size of $p - 1$.
- Increase the key size to p by XORing one of the outputs with an additional key-bit.
- $\exists\ \mathrm{K}^{\mathrm{a}}$ in which just the new key-bit is incorrect: $\mathrm{HD}(\mathrm{K}^{\mathrm{a}}, \mathrm{K}^{*}) = 1$ (very low), but $\mathrm{ER}(\mathrm{K}^{\mathrm{a}}) = 1$ (very high).



Takeaway: The above proposition is <u>not</u> accurate.

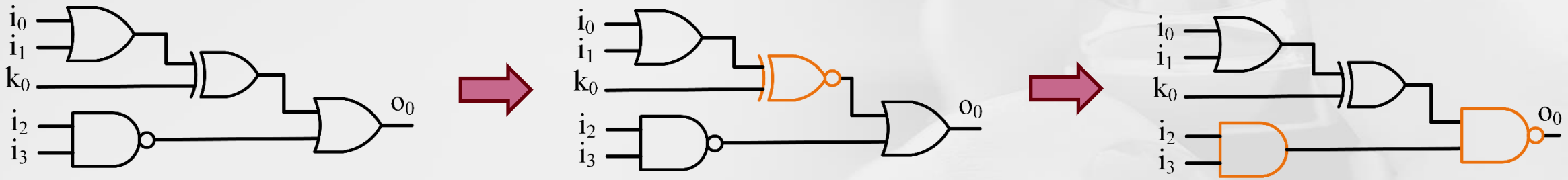
2.3. (Critique of the) SOTA GNN Attacks

Proposition: State-of-the-art GNN-based attacks can report an approximate key K^{a} of the locked circuit G in which $\mathrm{HD}(\mathrm{K}^{\mathrm{a}}, \mathrm{K}^{*})$ is very small.

Counterexample:

- Consider OMLA* with prediction accuracy of ~80%, i.e., $\mathrm{HD}(\mathrm{K}^{\mathrm{a}}, \mathrm{K}^{*}) \cong 0.2p$.
- Replace all the XOR gates with XNOR and push the inverters to the fanouts with bubble pushing.
- The new correct key is the complement of the previous one.
- The attack prediction accuracy drops significantly to ~56%, i.e., $\mathrm{HD}(\mathrm{K}^{\mathrm{a}}, \mathrm{K}^{*}) \cong 0.44p$.

This is not much better than a random guess.



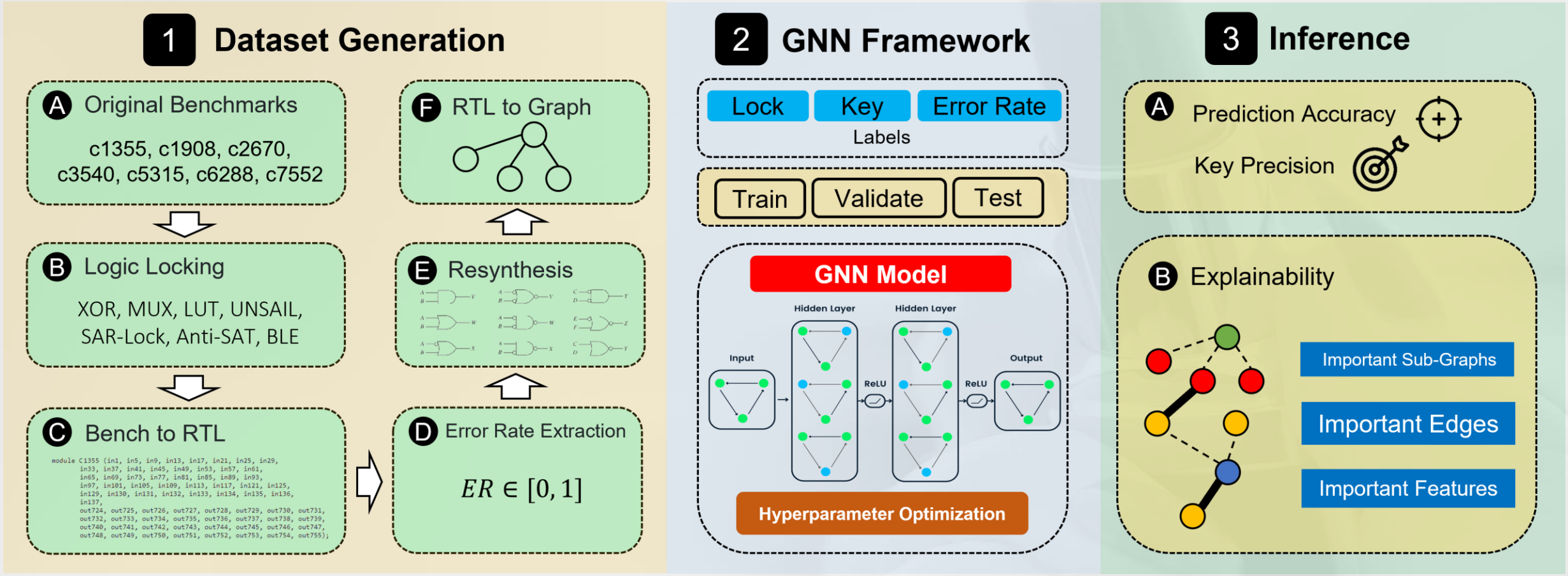
Takeaway: The above proposition is not accurate.

* L. Alrahis et al., "OMLA: An oracle-less machine learning-based attack on logic locking," In IEEE Transactions on Circuits and Systems II, 2022.

03
LIPSTICK Attack

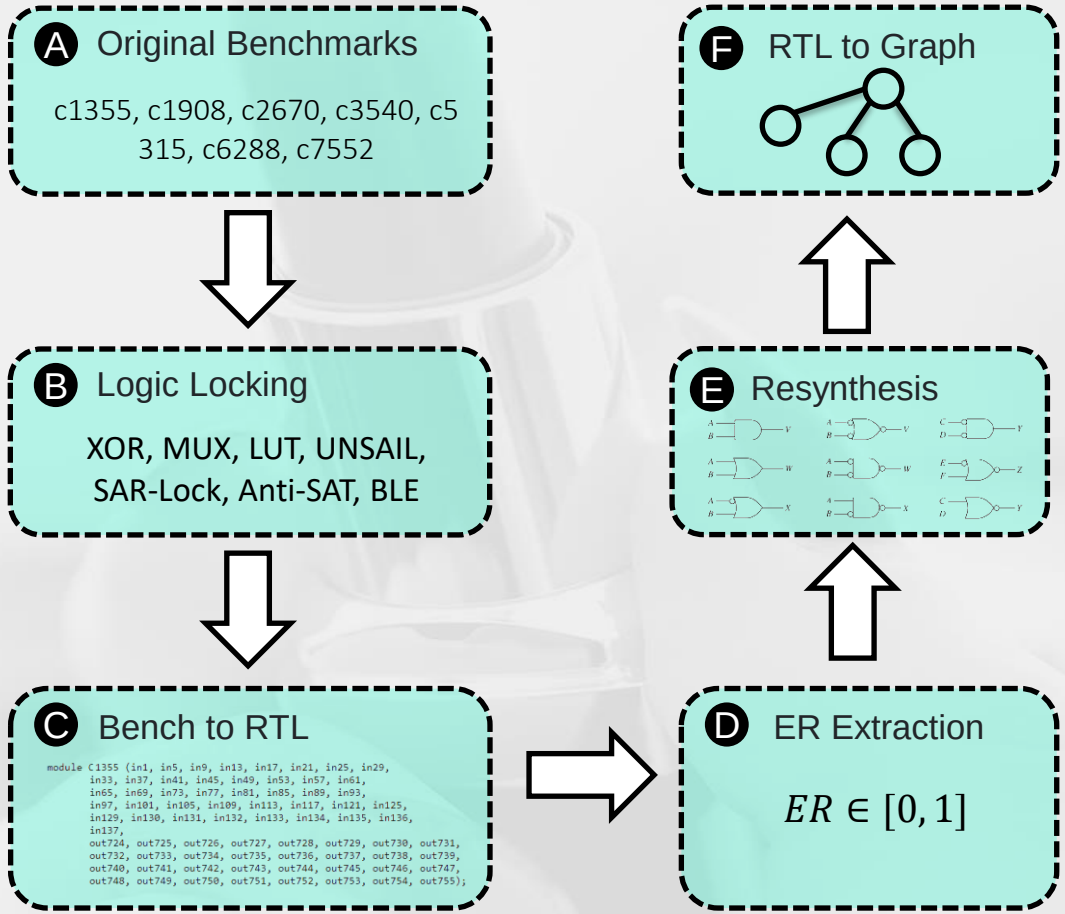
3.1. Attack Framework
3.2. Dataset Generation
3.3. GNN Framework
3.4. Inference

3.1. Attack Framework



3.2. Dataset Generation

- Seven of the ISCAS'85 (.Bench files)
- Seven logic locking methods
- Convert .Bench to RTL (.V) using ABC tool.
- Extract ER of 10 random wrong keys + the correct key using ModelSim.
- Apply bubble-pushing to create 10 resynthesized versions of each benchmark.
- Convert RTL benchmarks to Graphs using netlist-to-subgraph tool in OMLA.
- Overall, 5390 data elements



3.3. GNN Framework

GNN as undirected graph: $G = (V, E, X, A)$

Adjacency matrix

Vertex set

Edge set

Node feature matrix

Architecture: Graph Isomorphism Network (GIN)*

Training phase: Increase the model's prediction accuracy + key precision.

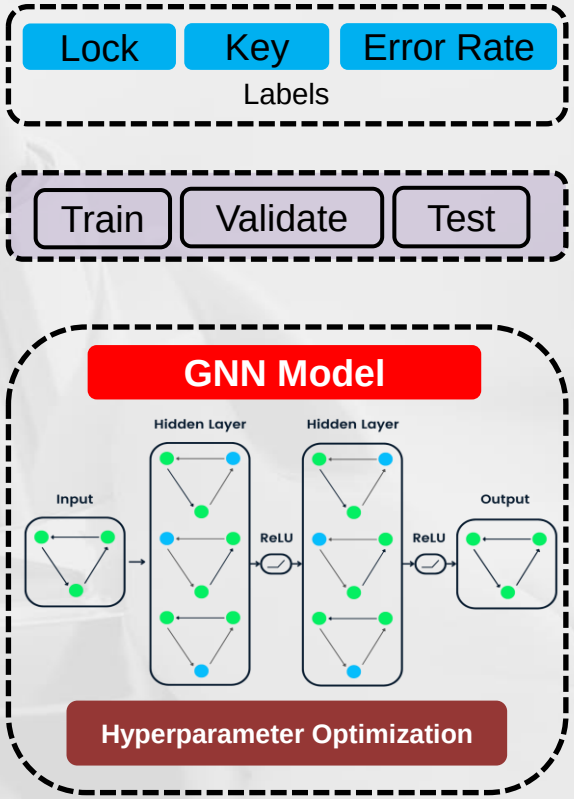
Goal: Predict a more relevant key with low output corruptibility.

Hyperparameter tuning:

Learning Rate (LR): After 100 epochs, LR gets its 0.01 value for the next 100 epochs.

Activation function: Leaky ReLU to keep the value of x using the maximum function f(x) = max(0.01x, x).

Early stopping strategy: if, after 5 consecutive iterations, the model does not achieve greater accuracy or if the loss value increases to 1.

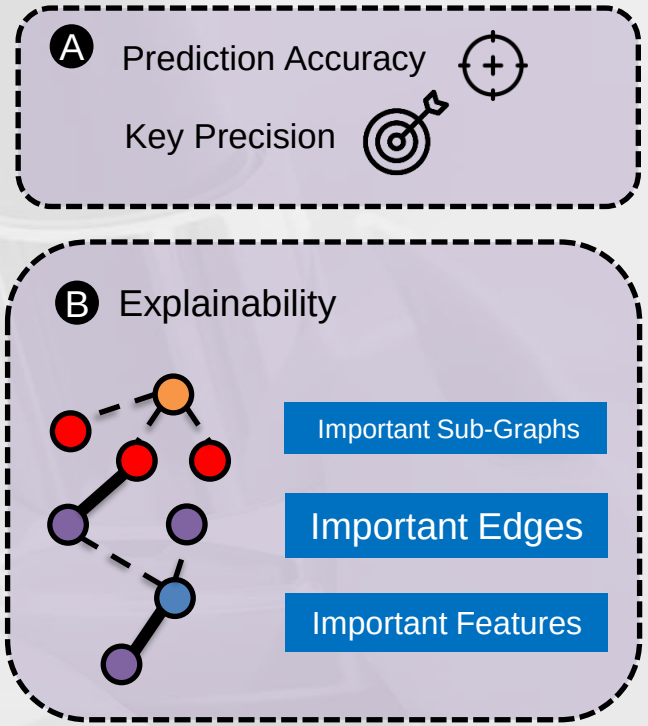


* K. Xu et al., "How powerful are graph neural networks?," In ICLR, 2019.

3.4. Inference

Testing phase: Test the model's prediction accuracy and reported key precision

Explainability: Feed the trained model to PGExplainer* → Use a parametric explanation network built on a graph-generative model to provide topological explanations.



D. Luo et al., "Parameterized explainer for graph neural network," In NIPS, 2020.

04

Experimental Results

4.1. Attack Results

4.2. Explainability Results

4.1. Attack Results - OMLA

In OMLA*, the model's prediction accuracy does not correlate with the reported key precision. → A model's accuracy of 80% does not assure high key precision.

This is the same in other SOTA GNN-based attacks. → They solely focus on the structures of the circuits, not their functionality.

OMLA's model prediction accuracy stays the same when using random feature map assignment. → It does not distinguish the gates in its inference.

OMLA's prediction accuracy and reported key precision under different feature maps

Prediction Accuracy	Key Precision	Epoch	Feature Map Description
80.78%	59.75%	350	Default
80.63%	61.33%	350	Random Assignment
77.63%	62.29%	350	Highest Assignment to Lowest #Gates

* L. Alrahis et al., "OMLA: An oracle-less machine learning-based attack on logic locking," In IEEE Transactions on Circuits and Systems II, 2022.

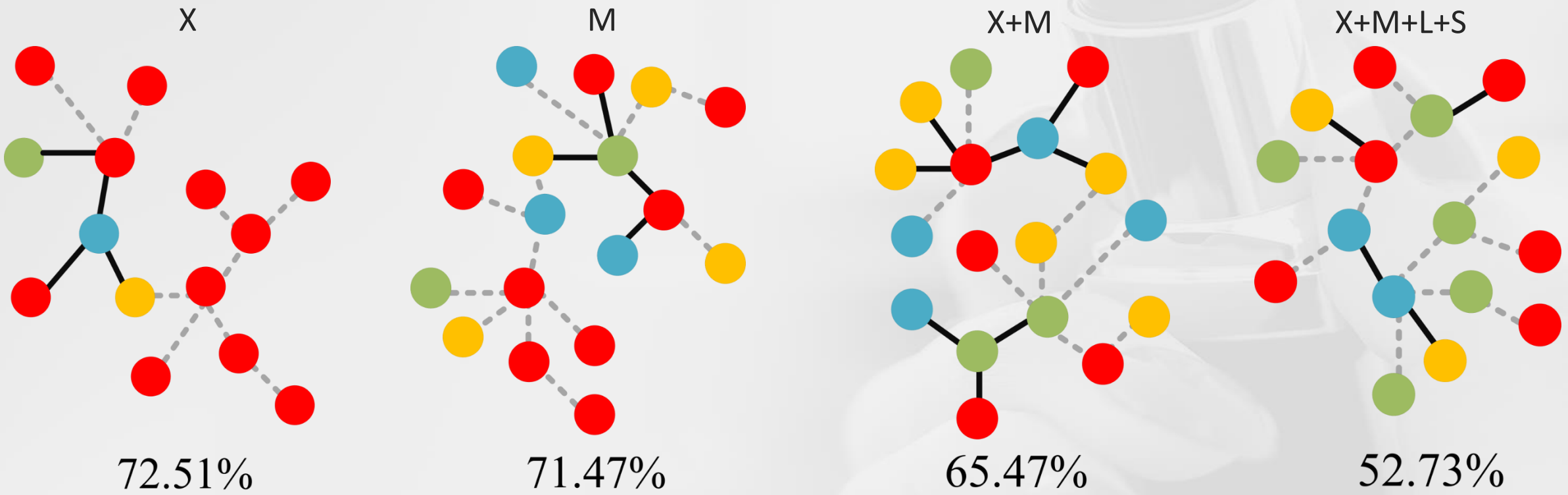
4.1. Attack Results - LIPSTICK

LIPSTICK's prediction accuracy and reported key precision under random seen and unseen benchmarks

Locking Scheme	Prediction Accuracy	5 Random Key Precision	10 Random Key Precision	50 Random Key Precision
X	92.64%	79.84%	75.57%	74.97%
M	93.11%	79.41%	75.44%	75.66%
L	92.75%	78.57%	75.68%	75.54%
S	93.43%	79.19%	76.21%	75.94%
X, M, L	85.50%	74.86%	70.63%	70.75%
X, L, S	84.16%	74.33%	70.58%	70.06%
X, M, S	82.22%	75.78%	69.16%	68.65%
M, L, S	84.87%	75.44%	70.33%	69.28%
X, M, L, S	76.95%	69.19%	65.39%	67.03%
X, M, L, S, B	51.23%	50.63%	49.97%	50.27%

4.2. Explainability Results

- Colored nodes represent different features.
- Black edges illustrate the patterns that PGExplainer was able to find.



05
Conclusion

5.1. Summary
5.2. Acknowledgment

5.1. Summary

- LIPSTICK: A corruptibility-aware and explainable GNN-based OL attack on different logic locking methods
- Achieve higher **prediction accuracy** and higher **key precision** compared with SOTA works.
- Incorporate circuit functionality + structural parameters → Guide the model into a more relevant key.
- Include resynthesized versions of the same circuit. → Learn features from different structural views.
- Involve logic-locked circuits with both correct and wrong key labels. → Learn from wrong keys too.
- Receive info on the importance of each feature on model decision. → Safeguard against attacks.

5.2. Acknowledgment

This work is supported by the **National Science Foundation** under Award No. 2245247.

THANK YOU

Yeganeh Aghamohammadi

Amin Rezaei