IEICE Electronics Express, Vol.22, No.12, 1–6

LETTER

Pre-silicon hardware Trojan detection in mixed-signal circuits using heterogeneous graph attention networks

Xing Hu[1, 2], **Yang Zhang**[1, 2, a)], **Jialong Song**[1, 2], **Ting Su**[1, 2], **Huan Guo**[1, 2], **Zhenyu Zhao**[1, 2], **and Keqin Li**[3]

Abstract Detecting hardware Trojans (HTs) in mixed-signal circuits is challenging due to structural complexity and cross-domain vulnerabilities between analog and digital components. Existing methods often rely on post-silicon analysis, circuit modifications, or focus solely on leakage, limiting practicality. We propose HGAT4TJ, a pre-silicon detection approach based on heterogeneous graph attention networks, which models gate- and transistor-level structures in a unified graph. This enables effective cross-domain HT detection directly from netlists without requiring golden models. Experimental results on benchmark circuits indicate that HGAT4TJ achieves 100% detection rate at the circuit level and over 97% accuracy at the node level, making it a non-invasive solution for HT detection in mixed-signal circuits.

Keywords: hardware Trojan, mixed-signal circuits, pre-silicon detection, heterogeneous graph, gate-level, transistor-level
Classification: Integrated circuits

1. Introduction

With the globalization of integrated circuit (IC) design and manufacturing, a single chip's development now involves multiple parties, including CAD tool providers, Intellectual Property (IP) vendors, and foundries. As ICs move through various stages, there are many opportunities for adversaries to introduce security risks. Hardware Trojan (HT) is one of such threats [1, 2]. A hardware Trojan refers to a malicious modification in the design or manufacturing process of an integrated circuit, often inserted to alter its functionality or introduce vulnerabilities. HTs are particularly concerning for industries, governments, and defense due to their potential to cause significant damage, including data breaches, system failures, or unauthorized control. Consequently, detecting HTs has been a major focus of research for the past two decades [3–5].

As the complexity of modern ICs increases, the prevalence of mixed-signal circuits, which combine both analog and digital components, is also increasing. This integration introduces new challenges for HT detection, particularly for HTs that exploit interactions between the digital and analog domains to build themselves [6]. These Trojans, such as the A2 Trojan [7, 8], are often more difficult to detect than digital ones, as they involve subtle modifications to the analog circuitry, triggering malicious behavior in the digital domain. Recent studies have underscored the severity of these threats [9]. Unfortunately, most existing detection methods are specifically designed for either digital circuits [10–14] or analog circuits [15–17] and are ineffective at detecting threats in mixed-signal designs. As such, there is a pressing need for effective detection solutions tailored to the unique challenges posed by mixed-signal circuits.

Several approaches have been proposed to address HT detection in mixed-signal circuits. Hou [18, 19] introduces an on-chip mechanism that detects Trojans by monitoring abnormal signal toggling. [20, 21] present an information flow tracking method that identifies cross-domain data leakage. Abedi [22] proposes a current signature-based detection technique applicable to both runtime and production phases. Pavlidis [23] proposes a built-in self-test method for detecting Trojans by monitoring internal node symmetry and using invariant signals with tolerance-based checkers. Deng [24] develops a ring oscillator-based structure along with two post-fabrication detection schemes to efficiently detect A2 Trojans. Sakamoto [25] and Su [26] utilize near-infrared imaging for non-destructive detection of logic cell modifications linked to hardware Trojans.

Despite these efforts, existing mixed-signal HT detection methods have notable limitations. Many rely on post-silicon analysis, which incurs high costs and risks, as detecting a HT after fabrication renders the design unusable, leading to significant economic losses. Other methods focus exclusively on detecting information leakage-based Trojans or require modifications to the circuit design, which will reduce their effectiveness and applicability.

To address the above challenges, we propose HGAT4TJ, a novel approach that models mixed-signal circuits as a unified heterogeneous graph, rather than treating digital and analog components separately. By explicitly capturing the interactions between digital logic gates and analog transistors within a single model, HGAT4TJ enables the detection of hardware Trojans that exploit cross-domain vulnerabilities. The key contributions are as follows:

1. To the best of our knowledge, we are the first to model the entire mixed-signal circuit as a unified heterogeneous graph, rather than treating the digital and analog components separately. Unlike traditional methods, our model is capable of handling cross-domain interactions. It differentiates between digital logic gates and analog transistors as

[1] College of computer science and Technology, National University of Defense Technology, Changsha, China
[2] Key Laboratory of Advanced Microprocessor Chips and Systems, China
[3] State University of New York, USA
[a)] zhangyang@nudt.edu.cn

DOI: 10.1587/elex.22.20250237
Received April 15, 2025
Accepted May 8, 2025
Publicized May 16, 2025
Copyedited June 25, 2025

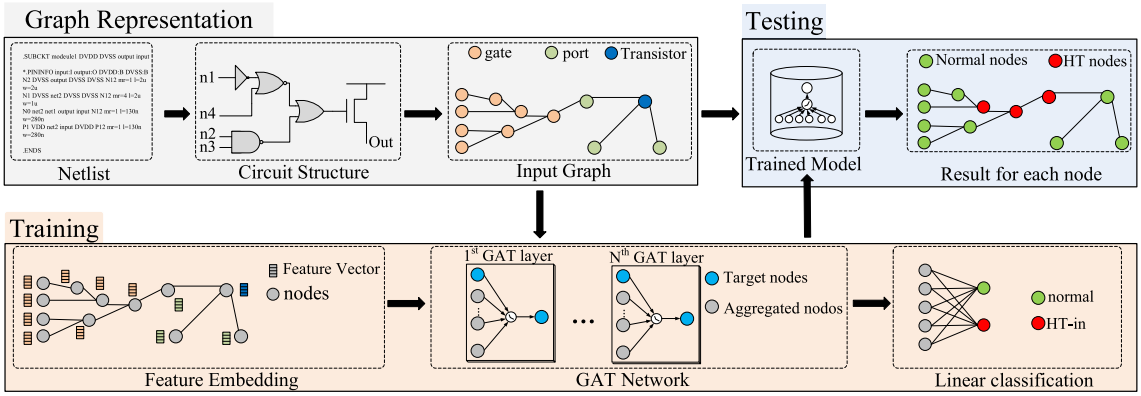

Fig. 1 Overview of HGAT4TJ (Graph Representation: the mixed-signal circuit is represented as a heterogeneous graph, where nodes correspond to circuit components and edges capture their interconnections. Training: node features are encoded and refined through GAT layers based on neighborhood information. The trained model is then evaluated for its effectiveness in detecting HTs during testing).

different node types, capturing their unique characteristics. The edges of the graph represent the signal flow, creating a structured framework for effective graph-based learning.

2. We adapt the Heterogeneous Graph Attention Network (HGAT) to exploit the heterogeneity of mixed-signal circuits, integrating structural analysis to enhance the accuracy of hardware Trojan detection. By modeling interactions between different circuit components and learning their structural dependencies, our approach achieves over 97% detection accuracy in mixed-signal circuits.

3. HGAT4TJ conducts structural analysis of the circuit's netlist during the pre-silicon phase, providing an efficient and non-invasive solution for hardware Trojan detection specifically in mixed-signal circuits. This method does not require any changes to the circuit design or additional test patterns, making it compatible with existing design flows.

2. Proposed methodology

2.1 Motivation

Detecting HTs in mixed-signal circuits is challenging due to the intricate interactions between digital and analog components. Traditional methods primarily focus on digital circuits, yet fail to address the complexities introduced by mixed-signal designs. To capture both the topological structure and the diverse interactions within mixed-signal circuits, we propose using HGAT, a heterogeneous graph-like network, to model and address these challenges.

2.1.1 IC design is graph-like

ICs inherently exhibit a networked structure, where components interact through well-defined connections. This makes a graph-based representation a natural fit for modeling circuit behavior. In this representation, nodes correspond to fundamental circuit elements, while edges represent interconnections.

A graph model not only captures the structural topology of digital circuits but also extends to mixed-signal circuits, where interactions between digital and analog components must be considered. By leveraging this structure, graph-based models facilitate systematic circuit analysis, enabling the detection of hardware anomalies, such as irregular connections or signal flows indicative of HT activity.

2.1.2 Heterogeneity in mixed-signal circuit

Unlike purely digital circuits, mixed-signal circuits integrate both binary-driven digital components and continuous-value analog components, introducing fundamental differences in their behaviors and interactions. Digital circuits operate on discrete logic states, while analog circuits process continuous signals, leading to diverse computational principles.

To address this heterogeneity, heterogeneous graph models differentiate between different types of nodes (e.g., logic gates, transistors, capacitors) and edges (e.g., signal paths, control signals). This enables a unified representation that preserves the distinctions between digital and analog domains while allowing cross-domain interactions to be effectively modeled.

By leveraging heterogeneous graph learning, the model differentiates between digital and analog components by encoding their distinct electrical behaviors and structural characteristics into separate feature spaces. As a result, the model effectively distinguishes normal circuit structures from HT-infected patterns by identifying irregular interactions or anomalies. This capability enhances pre-silicon HT detection accuracy in mixed-signal circuits, ensuring adaptability to various circuit designs and HT attack strategies.

In summary, heterogeneous graph-based models provide a powerful tool for modeling and analyzing mixed-signal circuits by incorporating their structural complexity, cross-domain interactions, and diverse component behaviors.

2.2 Overall architecture of HGAT4TJ

HGAT4TJ consists of three main phases: generation of heterogeneous graph, HGAT construction, and HGAT training and testing, which are shown in Fig. 1. In the phase of heterogeneous graph generation, the mixed-signal circuit is represented as a heterogeneous graph, where nodes represent components (ports, transistors, gates), and edges capture the relationships between them. The construction phase of HGAT involves initializing and encoding node features, as well as constructing a GAT-based network. Finally, in the training and testing phase, HGAT refines node representations to distinguish between HT-infected and normal components. The trained model is tested for its accuracy in identifying potential HTs, enabling detection across both

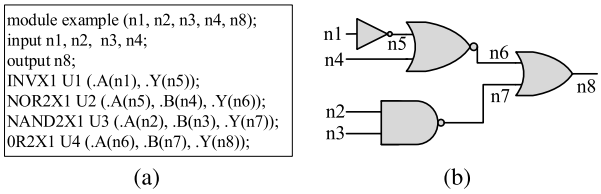

Fig. 2 The netlist and its structure in the digital domain.

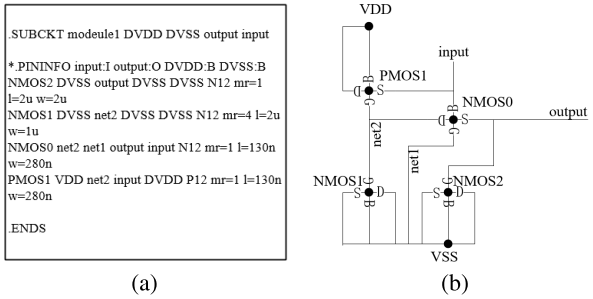

Fig. 3 The netlist and its structure in the analog domain.

digital and analog domains by analyzing intricate interactions within the circuit.

2.3 Construction of heterogeneous graph

To address the challenge of modeling interactions between digital and analog components in mixed-signal circuits, we propose a unified graph representation that captures the interconnections between both domains. In this model, nodes represent components from both the digital and analog domains, such as logic gates (AND, OR, etc.), transistors (NMOS, PMOS), and capacitors, while edges represent the connections between these components, such as signal paths. The edges are directed to reflect the flow of control and signal propagation between the components, capturing both intra-domain (digital-digital, analog-analog) and cross-domain (digital-analog) interactions.

Figure 2 and Fig. 3 illustrate the netlists and their structures in the digital and analog domains, respectively. In Fig. 4, these structures are modeled as a heterogeneous graph, where nodes represent components such as logic gates (e.g., NOT), transistors (e.g., NMOS), and ports (e.g., source). The edges, such as $n1$ and $net1$, are used to represent the connections between these components.

Specifically, for cross-domain interactions, as shown in Fig. 4, edge $n8$ in the digital domain connects to the source terminal of transistor $N0$ in the analog circuit. The digital control signal transmitted through $n8$ influences the state of the analog transistor, which is represented as an edge in the graph, effectively bridging the digital and analog domains. The model captures these critical interactions, particularly when a digital signal governs the behavior of the analog circuit or when an analog signal affects the digital domain.

2.4 Architecture of HGAT

Heterogeneous Graph Attention Network (HGAT) is a graph-based deep learning model designed to process complex relationships in structured data. In our proposed model, HGAT efficiently models the interactions between different types of components in mixed-signal circuits. We represent

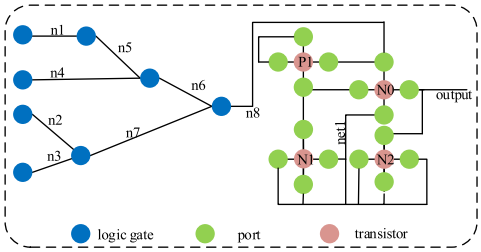

Fig. 4 Example of heterogeneous graph construction.

Table I Feature representation of different node types.

Node Type	Feature Description
port	port type: one-hot encoding of source, drain, gate, body
transistor	type: NMOS / PMOS; Width: Channel width (W); Length: Channel length (L); Aspect Ratio: Width-to-Length ratio (W/L)
capacitor	type: capacitor
resistor	type: resistor
gate	gate type: one-hot encoding of AND, OR, etc.

the circuit C as a heterogeneous graph $G = (V, E)$, where: V is the set of nodes representing components in the circuit, E is the set of edges representing the connections between components. Our goal is to predict a set of labels $y_1, \cdots, y_n$ (where $y_i \in \{0, 1\}$) for each node, where $y_i = 1$ indicates the presence of an HT in the i-th node.

2.4.1 Definition and feature construction of nodes

The nodes in the heterogeneous graph are categorized into five main types: *port*, *transistor*, *capacitor*, *resistor*, and *gate*, each representing a different functional unit in the circuit (shown in Table I).

Port nodes represent the terminals of a transistor, including the source, drain, gate, and body. Each port node is characterized by its type, which serves as its primary feature. This feature is encoded as a one-hot vector to distinguish between different port types. Each transistor node is characterized by specific parameters such as the transistor type (e.g., NMOS, PMOS), along with physical properties such as width, length, and width-to-length ratio. These characteristics form the feature vector of the transistor node, enabling the model to distinguish between different transistor types and configurations. Capacitor and resistor nodes are characterized by their type, providing distinct features that capture their electrical behavior. Gate nodes represent logic gates (AND, OR, etc.), which are fundamental in digital circuits. Each gate node is encoded with a one-hot vector based on its type, allowing the model to capture the logical operations performed by each gate.

2.4.2 Network structure and components of HGAT

HGAT utilizes an attention mechanism to account for the heterogeneous nature of the graph, where each node can have different types of neighbors and edges. For each node, the attention score is computed based on the neighboring nodes and the connection (edge) between them.

HGAT comprises three main parts: an embedding layer, GAT layers, and a linear layer. The embedding layer is responsible for encoding the nodes within the designs, which

involves generating the nodes, defining their connections, and initializing their features. The GAT layers use attention mechanisms to iteratively update node representations by learning the importance of neighboring nodes and their connections. The linear layer performs the classification task, mapping the learned node representations to the final output labels. HGAT and the linear layers are implemented according to the design outlined in [27, 28].

Unlike traditional graph convolutional neural networks that treat all nodes and edges uniformly, HGAT assigns different importance weights to each connection, allowing it to capture the distinct behaviors of different components in mixed-signal circuits.

2.5 Training and testing with HGAT

The HGAT model is trained using labeled datasets derived from mixed-signal circuits, which contain both Trojan-free and Trojan-infected samples. During training, the model learns to distinguish between HT-free and HT-infected nodes through supervised learning. To achieve this, it optimizes a binary classification loss function that guides the model in accurately identifying HT nodes within the circuit graphs.

In training, each node is assigned a label that identifies whether it is part of an HT. The loss function, typically a cross-entropy loss, is minimized to improve classification accuracy. The model iteratively refines its attention weights to learn important structural features that distinguish HT-infected nodes from normal nodes.

Once trained, HGAT is applied to unseen mixed-signal circuits for HT detection. The previously trained model analyzes the graph structure by leveraging the attention mechanism, allowing it to identify anomalous nodes that indicate potential HTs. Upon detecting an HT node, further analysis of its connections and neighboring nodes is performed to assess the extent of the attack and its potential impact.

3. Evaluation

3.1 Experimental setup

The experiments are conducted using the PyTorch framework on a server with an Intel i9 processor, clocked at 3.0 GHz, and equipped with 16 GB of memory. Both training and hardware Trojan detection tasks are performed using an NVIDIA GeForce RTX 3090 graphics card.

3.2 Dataset construction

The benchmark dataset aggregates functionally heterogeneous circuit architectures from TrustHub [29] and open-source IP cores [30]. These configurations collectively span signal conversion, power management, secure data transmission, and real-time control functionalities, thereby enabling holistic evaluation of hardware trustworthiness across multi-domain operational scenarios.

To further clarify the variability of the constructed dataset, we explicitly detail the injected HTs: digital and mixed-signal Trojan variants sourced from established benchmark libraries TrustHub [29] (e.g., T1000–T1600, covering combinational, sequential, and rare-event triggers) and MS-HT [7] (analog/mixed-signal triggers). These HTs are randomly embedded at various circuit nodes, ensuring diverse activation scenarios with varied signal surroundings and functional interactions. Each injected HT triggers under predefined conditions to modify circuit behavior at critical locations, termed *attack points* (see Fig. 5), thereby closely simulating realistic malicious scenarios. Table II summarizes the 31 benchmarks, clearly identifying the benchmark circuits, the integrated HTs, and the functionalities.

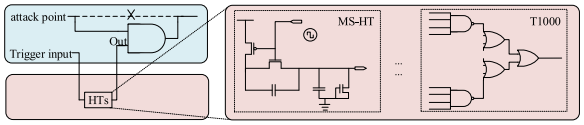

Fig. 5 Example of dataset with HTs injected.

3.3 Dataset splitting and evaluation metrics

To ensure effective HGAT model training and evaluation, the 31 circuits were partitioned into 80% (24 circuits) for training and 20% (7 circuits) for testing, maintaining strict circuit-level segregation to prevent data leakage between the training and testing sets. The training set captures discriminative features between normal and HT-infected nodes, while the test set contains exclusively unseen circuits for rigorous generalization assessment.

We classify nodes as HT-free or HT-infected using these metrics: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Evaluation metrics include the true positive rate (TPR = TP / (TP + FN)), the false positive rate (FPR = FP / (FP + TN)), and the precision (Acc = (TN + TP)/(TN + TP + FN + FP)).

3.4 Results and discussion

As demonstrated in Table III, the proposed HGAT4TJ framework achieves perfect circuit-level HT detection (100% HT detection) across all evaluated mixed-signal circuits, with an overall detection accuracy of 97.66% and a minimal false positive rate of 1.88%. Note that the HT detection rate indicates whether a circuit-level HT presence is detected, while the detection accuracy measures node-level classification correctness across all circuit nodes. This performance is consistently effective for both digital and analog Trojan variants without relying on circuit redesign or golden reference models. Although the 72.14% average recall reflects partial node-level detection, the framework strategically identifies critical subsets of HT nodes (33%–95% per instance) — sufficient to disrupt malicious functionality through key component localization, as modern HTs require coordinated activation of multiple nodes. Specifically, in cases with lower recall scores (e.g., RS232-NSupport-T1300), certain Trojan nodes closely mimic normal node characteristics, making them more difficult to distinguish from legitimate nodes. Nonetheless, the proposed method reliably detects crucial activation nodes within these subsets, effectively neutralizing the potential threat despite partial node-level identification.

While false-positive nodes are observed, their number is extremely limited and they are spatially isolated without structural correlation. These nodes are insufficient to form a functional HT and can be effectively removed through a sim-

Table II Benchmark circuits for hardware Trojan detection.

Benchmark circuit	Function	Trojan Type	Circuit Instances (No.)
ca_prng	Pseudo-random Number Generator	MS-HT	ca_prng_MS(1)
cf_cordic	Trigonometric Calculation (CORDIC)	MS-HT	cf_cordic_MS(1)
fast_antilog	Antilogarithmic Calculation	MS-HT	fast_antilog_MS(1)
fast_log_highacc	High Accuracy Logarithmic Calculation	MS-HT	fast_log_highacc_MS(1)
fast_log_pipelined	Pipelined Logarithmic Calculation	MS-HT	fast_log_pipelined_MS(1)
i2c	Serial Communication (I2C)	MS-HT	i2c_MS(1)
s15850, s35932, s38417	ISCAS89 Benchmark Circuit	MS-HT	s15850_MS, s35932_MS, s38417_MS(3)
RS232	Serial Communication (RS232 Interface)	MS-HT	RS232_MS(1)
RS232_OPAMP	RS232 Interface with operational amplifier	T1000–T1600	RS232_OPAMP-T1000-T1600(7)
RS232-PSupport	RS232 Interface with positive auxiliary circuit	T1000–T1600	RS232-PSupport-T1000-T1600(7)
RS232-NSupport	RS232 Interface with negative auxiliary circuit	T1000–T1600	RS232-NSupport-T1000-T1600(7)
Note: Trojan types: T1000–T1600 [29], MS-HT [7].			

Table III Experimental results of HGAT4TJ framework.

Benchmark	Detection of Circuit-level	Detection of Node-level		
		Recall	FPR	Acc
ca_corrdic_MS	100.00%	95.00%	4.38%	95.61%
ca_prng_MS	100.00%	95.00%	0.40%	99.50%
i2c_MS	100.00%	95.00%	0.51%	99.38%
S15850_MS	100.00%	95.00%	2.23%	97.76%
RS232-NSupport-T1300	100.00%	50.00%	1.80%	97.57%
RS232-OPAMP-T1400	100.00%	41.67%	2.11%	96.82%
RS232-PSupport-T1500	100.00%	33.33%	1.78%	96.99%
Average	100.00%	72.14%	1.88%	97.66%

Table IV Results on HT-free benchmarks.

Benchmark	TP	TN	FP	FN
i2c_PSupport-free	0	832	3	0
ca_prng_NSupport-free	0	1034	3	0
RS232-NSupport-free	0	598	11	0
RS232-OPAMP-free	0	599	14	0
RS232-PSupport-free	0	598	11	0

ple filtering strategy applied after the HGAT4TJ inference. Specifically, we adopt a manual neighborhood consistency rule: nodes predicted as HT but unsupported by any of their immediate neighbors in the heterogeneous graph are excluded from the final alert set. This is based on the observation that actual HT nodes typically appear in connected clusters due to shared activation or payload logic.

This filtering strategy complements the GAT's message-passing mechanism and is applied post-inference, following the results reported in Table III. After applying the filter, most false positives — especially those that are spatially isolated — are effectively removed, while the recall remains unchanged.

To further verify that these isolated false positives do not result in circuit-level misclassification, we evaluate multiple HT-free benchmark circuits. As shown in Table IV, each design yields only a small number of isolated false positives (ranging from 3 to 14), which are insufficient to constitute functional Trojans. This confirms that the framework does not trigger circuit-level false alarms and maintains high detection reliability under benign conditions.

Table V presents a comparison between the proposed HGAT4TJ method and other recent techniques for HT detection. Unlike traditional GNN (Graph Neural Network)

Table V Comparison with other techniques.

Techniques	Method	Pre-silicon Detection	Without Modification	Limited HT
HGAT4TJ	GNN	✓	✓	-
[10]- [14]	GNN	✓	✓	digital
[18] [19]	runtime detection	×	✓	-
[20] [21]	IFT	✓	✓	leakage
[22]	current sensing	×	×	-
[23]	built-in-self-test	×	✓	-
[24]	acceleration	✓	×	-
[25] [26]	imaging	×	✓	-

methods such as [10–14], which are specifically designed to detect digital-only HTs, HGAT4TJ is capable of detecting both digital and analog HTs. In contrast to IFT (Information Flow Tracing) methods like [20, 21], which can only detect information leakage-based HTs in mixed-signal circuits, HGAT4TJ is not restricted to specific HT types. Moreover, while [18, 19, 22, 23], and [25, 26] all rely on post-silicon detection, which can result in higher costs and less efficient Trojan identification during the design phase, the detection of an HT at this stage often leads to significant financial losses. If an HT is found, the entire fabrication process may need to be halted, or the chip may need to be modified, causing delays in product release and necessitating costly re-fabrication. In contrast, HGAT4TJ leverages heterogeneous graph neural networks for pre-silicon detection, providing a more efficient and cost-effective solution for early Trojan identification. Additionally, unlike [24], which requires modifications to the circuit design, HGAT4TJ does not necessitate any such changes.

4. Conclusion

This paper presents a novel pre-silicon hardware Trojan detection approach for mixed-signal circuits using heterogeneous graph neural networks. Unlike traditional methods that primarily focus on digital circuits, HGAT4TJ explicitly accounts for the structural differences between digital and analog components by defining distinct node types and modeling their interactions within a heterogeneous graph. This representation enables HGAT to effectively capture the complex cross-domain dependencies inherent in mixed-signal circuits. By leveraging graph learning techniques, our method provides a non-intrusive, pre-silicon detection

framework that identifies HTs without requiring circuit modifications. Experimental results demonstrate that HGAT4TJ achieves high detection accuracy, making it a promising solution for securing IC designs against emerging threats.

Future work will scale the approach to larger and more complex circuits. We will also investigate neural-level mechanisms — such as uncertainty-aware attention and structural priors — to reduce node-level false positives and improve detection precision.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61832018.

References

[1] M. Tehranipoor and F. Koushanfar: "A survey of hardware Trojan taxonomy and detection," IEEE Des. Test Comput. **27** (2010) 10 (DOI: 10.1109/MDT.2010.7).

[2] U. Guin, *et al.*: "Counterfeit integrated circuits: a rising threat in the global semiconductor supply chain," Proc. IEEE **102** (2014) 1207 (DOI: 10.1109/JPROC.2014.2332291).

[3] M. Elshamy, *et al.*: "Digital-to-analog hardware Trojan attacks," IEEE Trans. Circuits Syst. I, Reg. Papers **69** (2022) 573 (DOI: 10.1109/TCSI.2021.3116806).

[4] W. Hu, *et al.*: "An overview of hardware security and trust: threats, countermeasures, and design tools," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. **40** (2021) 1010 (DOI: 10.1109/TCAD.2020.3047976).

[5] A. Jain, *et al.*: "Survey of recent developments for hardware Trojan detection," 2021 IEEE International Symposium on Circuits and Systems (ISCAS) (2021) 1 (DOI: 10.1109/ISCAS51556.2021.9401143).

[6] Q. Wang, *et al.*: "Hardware Trojans embedded in the dynamic operation of analog and mixed-signal circuits," 2015 National Aerospace and Electronics Conference (NAECON) (2015) 155 (DOI: 10.1109/NAECON.2015.7443059).

[7] K. Yang, *et al.*: "A2: analog malicious hardware," 2016 IEEE Symposium on Security and Privacy (SP) (2016) 18 (DOI: 10.1109/SP.2016.10).

[8] M.M. Bidmeshki, *et al.*: "Revisiting capacitor-based Trojan design," 2019 IEEE 37th International Conference on Computer Design (ICCD) (2019) 309 (DOI: 10.1109/ICCD46524.2019.00047).

[9] A. Kwong, *et al.*: "RAMBleed: reading bits in memory without accessing them," 2020 IEEE Symposium on Security and Privacy (SP) (2020) 695 (DOI: 10.1109/SP40000.2020.00020).

[10] R. Yasaei, *et al.*: "Golden reference-free hardware Trojan localization using graph convolutional network," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **30** (2022) 1401 (DOI: 10.1109/TVLSI.2022.3191683).

[11] R. Yasaei, *et al.*: "GNN4TJ: graph neural networks for hardware Trojan detection at register transfer level," 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE) (2021) 1504 (DOI: 10.23919/DATE51398.2021.9474174).

[12] K. Hasegawa, *et al.*: "R-HTDetector: robust hardware-Trojan detection based on adversarial training," IEEE Trans. Comput. **2** (2023) 333 (DOI: 10.1109/TC.2022.3222090).

[13] L. Wu, *et al.*: "Automated hardware Trojan detection at LUT using explainable graph neural networks," IEEE/ACM International Conference on Computer Aided Design (ICCAD) (2023) 1 (DOI: 10.1109/ICCAD57390.2023.10323915).

[14] K. Hasegawa, *et al.*: "Node-wise hardware Trojan detection based on graph learning," IEEE Trans. Comput. **3** (2025) 749 (DOI: 10.1109/TC.2023.3280134).

[15] K. Kunal, *et al.*: "GANA: graph convolutional network based automated netlist annotation for analog circuits," 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE) (2020) 55 (DOI: 10.23919/DATE48585.2020.9116329).

[16] N. Gupta, *et al.*: "DELTA: designing a stealthy trigger mechanism for analog hardware Trojans and its detection analysis," Proc. 59th ACM/IEEE Design Automation Conference (2022) 787 (DOI: 10.1145/3489517.3530666).

[17] J. Talukdar, *et al.*: "Automatic structural test generation for analog circuits using neural twins," IEEE International Test Conference (ITC) (2022) 145 (DOI: 10.1109/ITC50671.2022.00022).

[18] Y. Hou, *et al.*: "On-chip analog Trojan detection framework for microprocessor trustworthiness," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. **38** (2019) 1820 (DOI: 10.1109/TCAD.2018.2864246).

[19] Y. Hou, *et al.*: "R2D2: runtime reassurance and detection of A2 Trojan," 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST) (2018) 195 (DOI: 10.1109/HST.2018.8383914).

[20] M.M. Bidmeshki, *et al.*: "Information flow tracking in analog/mixed-signal designs through proof-carrying hardware IP," Proc. Conference on Design, Automation & Test in Europe (2017) 1707 (DOI: 10.23919/date.2017.7927268).

[21] X. Guo, *et al.*: "When capacitors attack: formal method driven design and detection of charge-domain Trojans," 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE) (2019) 1727 (DOI: 10.23919/DATE.2019.8714906).

[22] M. Abedi, *et al.*: "High-precision nano-amp current sensor and obfuscation based analog Trojan detection circuit," IEEE International Symposium on Circuits and Systems (ISCAS) (2022) 3324 (DOI: 10.1109/ISCAS48785.2022.9937796).

[23] A. Pavlidis, *et al.*: "SymBIST: symmetry-based analog and mixed-signal built-in self-test for functional safety," IEEE Trans. Circuits Syst. I, Reg. Papers **68** (2021) 2580 (DOI: 10.1109/TCSI.2021.3067180).

[24] D. Deng, *et al.*: "Novel design strategy toward A2 Trojan detection based on built-in acceleration structure," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. **39** (2020) 4496 (DOI: 10.1109/TCAD.2020.2977069).

[25] J. Sakamoto, *et al.*: "Non-destructive hardware Trojan circuit screening by backside near infrared imaging," 2023 IEEE Physical Assurance and Inspection of Electronics (PAINE) (2020) 1 (DOI: 10.1109/PAINE58317.2023.10317961).

[26] T. Su, *et al.*: "Improving the ability of thermal radiation based hardware Trojan detection," 33rd USENIX security symposium (USENIX security 24) (2024).

[27] P. Velickovic, *et al.*: "Graph attention networks," International Conference on Learning Representations (2018).

[28] Z. Hu, *et al.*: "Heterogeneous graph transformer," Proc. Web Conference (2020) 2704 (DOI: 10.1145/3366423.3380027).

[29] B. Shakya, *et al.*: "Benchmarking of hardware Trojans and maliciously affected circuits," Journal of Hardware and Systems Security **1** (2017) 85 (DOI: 10.1007/s41635-017-0001-6).

[30] N. Muralidhar, *et al.*: "Contrastive graph convolutional networks for hardware Trojan detection in third party IP cores," 2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST) (2021) 181 (DOI: 10.1109/HOST49136.2021.9702276).