DATE 25
DESIGN, AUTOMATION AND TEST IN EUROPE
THE EUROPEAN EVENT FOR ELECTRONIC SYSTEM DESIGN & TEST
31 MARCH – 2 APRIL 2025
LYON, FRANCE
CENTRE DE CONGRÈS DE LYON

WideGate: Beyond Directed Acyclic Graph Learning in Subcircuit Boundary Prediction

Jiawei Liu[1,2], Zhiyan Liu[1], Xun He[1], Jianwang Zhai[1], Zhengyuan Shi[2,3], Qiang Xu[2,3], Bei Yu[3], Chuan Shi[1]

[1] Beijing University of Posts and Telecommunications
[2] National Center of Technology Innovation for EDA
[3] The Chinese University of Hong Kong

Apr. 2, 2025

Outline

Introduction

Background: Subcircuit Boundary Prediction

- Predict subcircuit boundaries in and-inverter graphs (AIGs):
  - Subcircuit input boundary prediction (IBP).
  - Subcircuit output boundary prediction (OBP).
- The importance of AIGs:
  - Provide a fundamental representation of logic functions.
  - Essential for the synthesis, simulation, verification, and testing stages.
- The applications of subcircuit boundary prediction[1]:
  - Functional verification
  - Logic optimization
  - Malicious logic detection

[1] Z. He, et al. (2021). "Graph learning-based arithmetic block identification," in *Proc. ICCAD*, pp. 1–8.

Existing Solutions: DAG-based Graph Neural Networks

- Boolean circuits can be viewed as directed acyclic graphs (DAGs).
- Existing methods typically adopt DAG-based graph neural networks (GNNs):
  1. ABGNN[2]
  2. FGNN2[3]
- **Are DAG-based GNNs really suitable for the subcircuit boundary prediction task?**
  - **Reachability assumption**: valuable information is often located within the fanin/fanout cone.
  - **Homophily assumption**: neighboring nodes often belong to the same class.

[2] Z. He, et al. (2021). "Graph learning-based arithmetic block identification," in *Proc. ICCAD*, pp. 1–8.

[3] Z. Wang, et al. (2024). "Fgnn2: A powerful pre-training framework for learning the logic functionality of circuits," *TCAD*.

Key Point: Boundary Intercorrelation

- The output boundaries of the same subcircuit often exhibit strong positional correlations:
  - Capture these correlations is crucial for accurately identifying subcircuit boundaries.
  - Non-compliance with the reachability assumption.

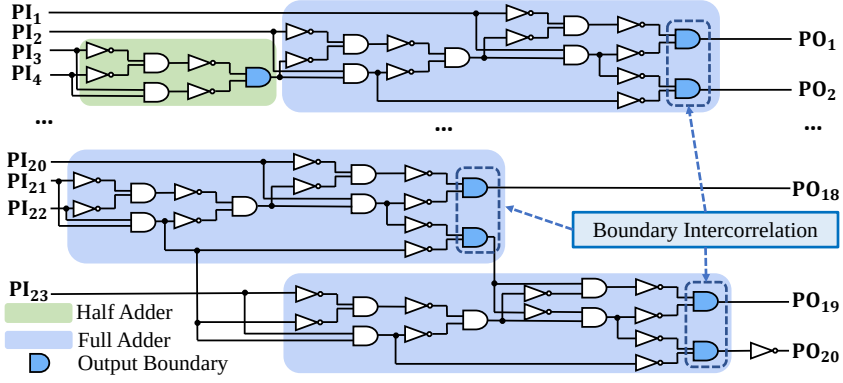
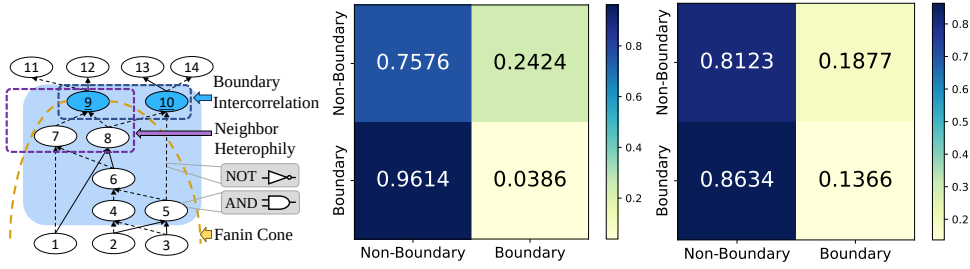


Illustration of subcircuits and output boundaries in an AIG.

Key Point: Neighbor Heterophily

- Most boundary nodes have non-boundary nodes as their direct neighbors.
  - E.g., among the neighbors of input boundary nodes, 96.14% are non-boundary nodes.
  - Non-compliance with the homophily assumption.



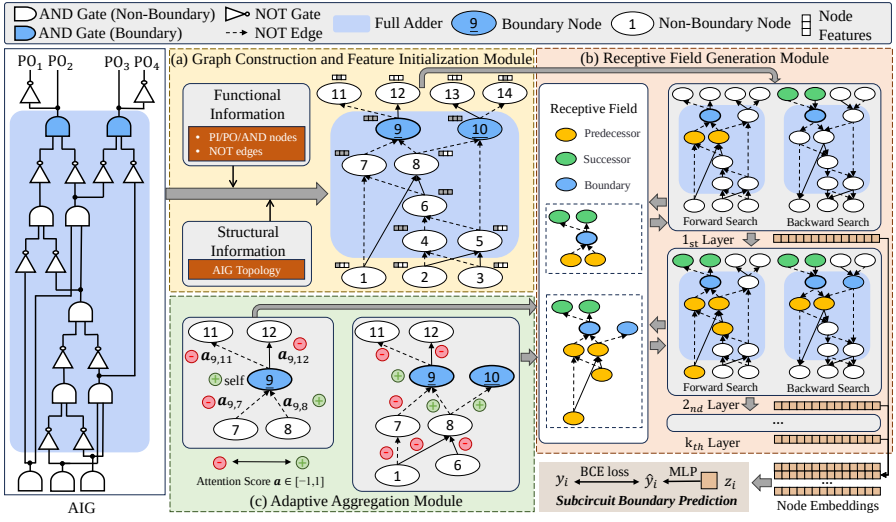
(a) Neighbor Heterophily (b) Input Boundary CM (c) Output Boundary CM

The neighbor heterophily and compatibility matrices (CM) in subcircuit boundary prediction task.

Methodologies

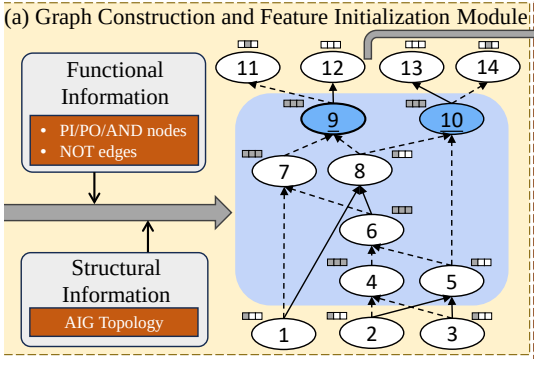
Overall Framework: WideGate

- WideGate consists of several components:



Part One: Graph Construction and Feature Initialization Module

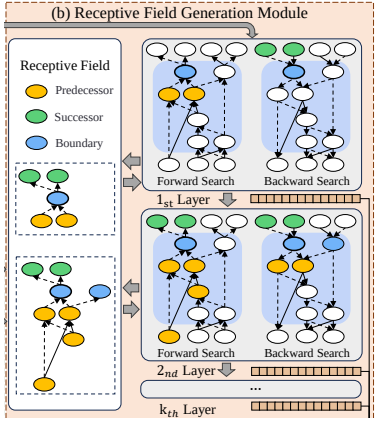
- Graph construction:
  - Nodes: PI, AND, PO.
  - Directed edges: pointing from source nodes to target nodes.
- Feature initialization:
  - The first dimension: whether the node is an AND node,
  - The second and third dimensions: whether the two input edges have NOT edges.



Part Two: Receptive Field Generation Module

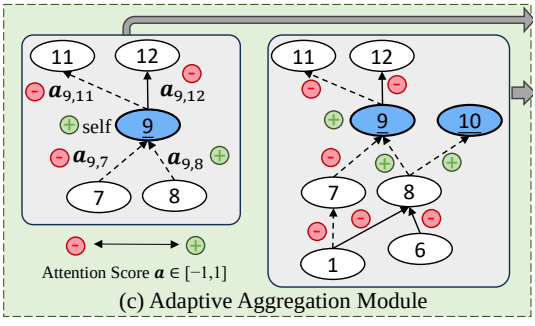
- Bidirectional search:
  - Both forward and backward searches in each step.
  - Efficiently integrated into GNN learning.

$$\begin{aligned} \boldsymbol{h}_i^{(2)} &= \boldsymbol{A}_{\rightarrow}\boldsymbol{A}_{\rightarrow}\boldsymbol{h}_i^{(0)}\boldsymbol{W}_{\rightarrow}^{(1)}\boldsymbol{W}_{\rightarrow}^{(2)} + \boldsymbol{A}_{\leftarrow}\boldsymbol{A}_{\leftarrow}\boldsymbol{h}_i^{(0)}\boldsymbol{W}_{\leftarrow}^{(1)}\boldsymbol{W}_{\leftarrow}^{(2)} \\ &+ \boldsymbol{A}_{\leftarrow}\boldsymbol{A}_{\rightarrow}\boldsymbol{h}_i^{(0)}\boldsymbol{W}_{\rightarrow}^{(1)}\boldsymbol{W}_{\leftarrow}^{(2)} + \boldsymbol{A}_{\rightarrow}\boldsymbol{A}_{\leftarrow}\boldsymbol{h}_i^{(0)}\boldsymbol{W}_{\leftarrow}^{(1)}\boldsymbol{W}_{\rightarrow}^{(2)}. \end{aligned} \tag{1}$$



Part Three: Adaptive Aggregation Module

- Self-information aggregation.
  - Motivation: boundary nodes are certainly of the same category as itself.
- Negative attention mechanism.
  - Motivation: the categories of neighboring nodes are unknown.



(c) Adaptive Aggregation Module

- The adaptive aggregation of WideGate is as follows:

$$
\begin{aligned}
\boldsymbol{m}_i^{(l)} &= \boldsymbol{h}_i^{(l-1)}, \\
\boldsymbol{m}_{\overrightarrow{\mathcal{N}_i}}^{(l)} &= \sum_{j \in \overrightarrow{\mathcal{N}_i}} \frac{\tanh\left(\boldsymbol{a}^\top \left[\boldsymbol{h}_i^{(l-1)}, \boldsymbol{h}_j^{(l-1)}\right]\right)}{\sqrt{d_i d_j}} \boldsymbol{h}_j^{(l-1)}, \\
\boldsymbol{m}_{\overleftarrow{\mathcal{N}_i}}^{(l)} &= \sum_{j \in \overleftarrow{\mathcal{N}_i}} \frac{\tanh\left(\boldsymbol{a}^\top \left[\boldsymbol{h}_i^{(l-1)}, \boldsymbol{h}_j^{(l-1)}\right]\right)}{\sqrt{d_i d_j}} \boldsymbol{h}_j^{(l-1)}.
\end{aligned}
\tag{2}
$$

- The complete message-passing function of WideGate is as follows:

$$
\boldsymbol{h}_i^{(l)} = \sigma\left(\boldsymbol{m}_i^{(l)} \boldsymbol{W}_{self}^{(l)} + \beta \boldsymbol{m}_{\overrightarrow{\mathcal{N}_i}}^{(l)} \boldsymbol{W}_{\rightarrow}^{(l)} + (1-\beta) \boldsymbol{m}_{\overleftarrow{\mathcal{N}_i}}^{(l)} \boldsymbol{W}_{\leftarrow}^{(l)}\right). \tag{3}
$$

- After L layers of aggregation, the embeddings of all layers are used to predict $\hat{y}_i$:

$$
\hat{y}_i = \sigma\left(\left[\boldsymbol{h}_i^{(0)}, \ldots, \boldsymbol{h}_i^{(L)}\right] \boldsymbol{W}_{pred}\right). \tag{4}
$$

Evaluation

Baselines

- Baselines:
  1. GraphSAGE[4]
  2. HOGA[5]
  3. RelGCN[6]
  4. FGNN2 and FGNN2 + Pretrain[7]

[4] W. Hamilton, et al. (2017). "Inductive representation learning on large graphs," in *Proc. NeurIPS*.

[5] C. Deng, et al. (2024). "Less is more: Hop-wise graph attention for scalable and generalizable learning on circuits," in *Proc. DAC*.

[6] M. Schlichtkrull, et al. (2018). "Modeling relational data with graph convolutional networks," in *Proc. ESWC*.

[7] Z. Wang, et al. (2024). "Fgnn2: A powerful pre-training framework for learning the logic functionality of circuits," *TCAD*.

Datasets

- We use an extended version of the FGNN2 dataset:
  - Including adders, subtractors, multipliers, and dividers.
  - Synthesized into gate-level netlists using Synopsys Design Compiler.
  - Converted into AIG format using ABC.
  - The word lengths of circuits range from 8 bits to 32 bits
  - The number of nodes varies from a few hundred to several thousand.

TABLE I The statistics of AIG datasets.

	Architectures		#
Adder	Brent-Kung Cond-Sum Hybrid Koggle-Stone Ling Sklansky	Sklansky Block Carry Look-head Carry Look-head Carry Select Carry-skip Ripple-Carry	1173
Subtractor	Hybrid Koggle-Stone Ling	Brent-Kung Cond-Sum Sklansky	645
Multiplier	Array Booth-Encoding Wallace Dadda	Overturned-stairs (4,2) compressor (7,3) counter Redundant binary addition	3788
Divider	Array		1180
Total	/		6786

Metrics

- Metrics:
  1. **Recall** and **F1-score**: focus only on boundary nodes.
  2. **True Negative Rate (TNR)**: focus only on non-boundary nodes.
  3. **Accuracy**: focus on both boundary and non-boundary nodes.
  4. **Avg. Runtime**: evaluates the model's training speed.

Experiment 1: Main Results

- Compared to the strongest baseline (FGNN2+Pretrain), WideGate improves:
  - Recall: +43.01%, F1-score : +29.87%, Avg. Runtime: -94.74%.
- WideGate performs best in predicting both non-boundary and boundary nodes.
  - HOGA performs well on non-boundaries but poorly on boundaries (F1-score=0.4925).
  - The pretraining step in FGNN2 improves the F1-score but at the cost of a decrease in TNR.

TABLE II Results on the subcircuit output boundary prediction (OBP) task. (Label ratio=5%)

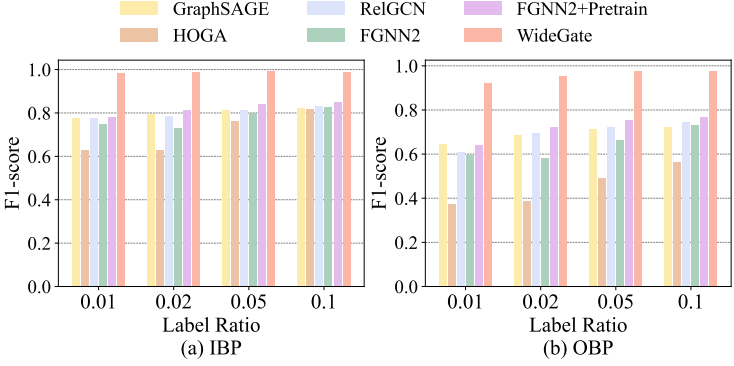
Metrics	GraphSAGE	HOGA	RelGCN	FGNN2	FGNN2+Pretrain	WideGate	Improvement (%)
Recall ↑	0.6298	0.3900	0.6448	0.5544	0.6806	**0.9733**	43.01%
F1 score ↑	0.7128	0.4925	0.7223	0.6624	0.7522	**0.9769**	29.87%
TNR ↑	0.9716	0.9599	0.9710	0.9753	0.9733	**0.9960**	2.12%
Accuracy ↑	0.9131	0.8623	0.9151	0.9032	0.9232	**0.9921**	7.46%
Avg. Runtime (s) ↓	1.2705	2.3797	1.8795	65.3075	63.9194	**3.1669**	94.74%

TABLE III Results on the subcircuit input boundary prediction (IBP) task. (Label ratio=5%)

Metrics	GraphSAGE	HOGA	RelGCN	FGNN2	FGNN2+Pretrain	WideGate	Improvement (%)
Recall ↑	0.8271	0.7040	0.8297	0.8047	0.8781	**0.9935**	13.14%
F1 score ↑	0.8115	0.7623	0.8137	0.8003	0.8412	**0.9938**	18.14%
TNR ↑	0.9502	0.9662	0.9506	0.9514	0.9506	**0.9986**	3.35%
Accuracy ↑	0.9267	0.9162	0.9276	0.9234	0.9367	**0.9976**	6.50%
Avg. Runtime (s) ↓	1.3063	2.4101	1.8993	68.5559	67.8344	**3.1541**	94.85%

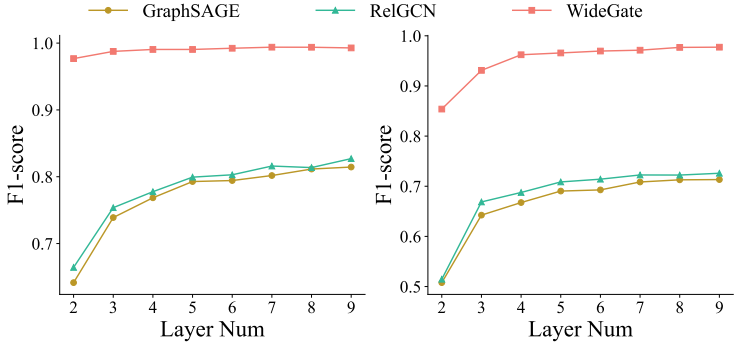
Experiment 2: Impact of Label Ratio

- WideGate exhibits stable and optimal performance for each label ratio:
  - F1-score significantly higher than the baseline models.
- WideGate's advantage is more pronounced at lower label ratios:
  - For a label ratio of 0.01, WideGate: +25.78% on the IBP task, +42.60% on the OBP task.
- Strong generalization ability:
  - WideGate (1% of the training data) surpasses baselines (10% of the training data).



Experiment 3: Impact of Aggregation Layers

- WideGate achieves significantly higher F1-scores than GraphSAGE and RelGCN.
- The 2-layer WideGate performs much better than the 9-layer GraphSAGE.



Experiment 4: Ablation Study

- The complete model achieves the best performance.
- Forward search is essential for both tasks.
- Backward search plays a significantly larger role in the OBP task than the IBP task.
- Only forward search + adaptive aggregation module still surpasses the baseline models.

TABLE IV Ablation results on OBP and IBP tasks.

	OBP	IBP
WideGate (w/o forward search)	0.6960	0.8042
WideGate (w/o backward search)	0.7898	0.9842
WideGate (w/o self-aggregation)	0.9665	0.9910
WideGate	**0.9769**	**0.9938**

Conclusion

Conclusion

- Investigate the limitations of DAG-based GNNs in the subcircuit boundary prediction:
  - Reachability assumption and homophily assumption.
- Identify two key challenges in GNN-based subcircuit boundary prediction:
  - Boundary intercorrelation and neighbor heterophily.
- The proposed WideGate:
  - Receptive field generation module: capture the boundary intercorrelation.
  - Adaptive aggregation module: account for neighbor heterophily.
- Conduct extensive experiments on subcircuit boundary prediction tasks:
  - Significantly improves prediction accuracy and training efficiency.
  - Strong generalization ability.

THANK YOU!