


Layout Symmetry Annotation for Analog Circuits with Graph Neural Networks

Xiaohan Gao[1], Chenhui Deng[2], Mingjie Liu[3], Zhiru Zhang[2], David Z. Pan[3], Yibo Lin[1]

[1]Department of Computer Science, Peking University

[2]ECE Department, Cornell University

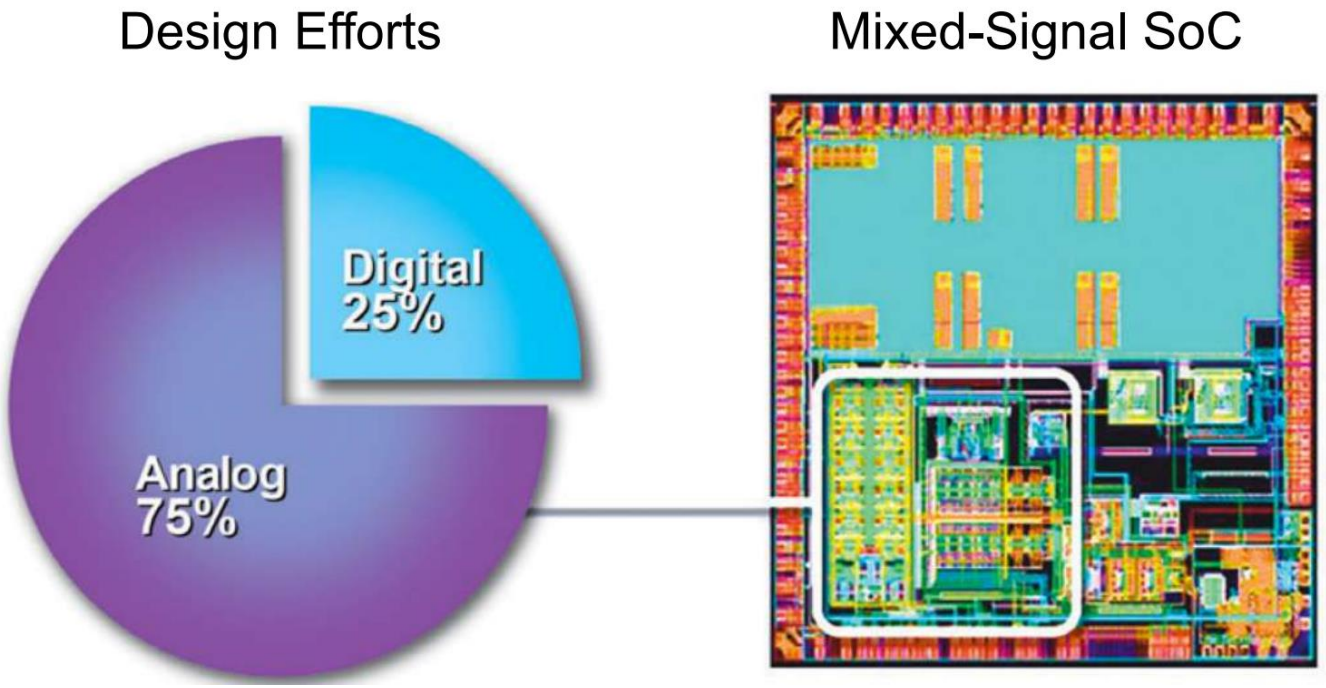
[3]ECE Department, The University of Texas at Austin

Outline

- Introduction and Problem Formulation
- SymDetect: Our Proposed Framework
- Experimental Results
- Conclusions and Future Work

Analog Layout Automation

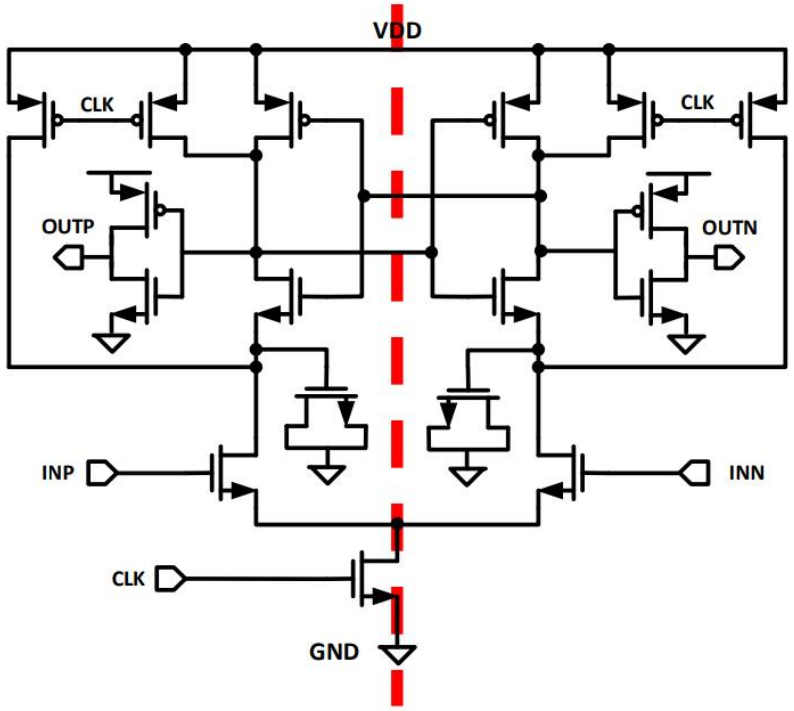
- Analog/mixed signal IC design still relies heavily on manual design



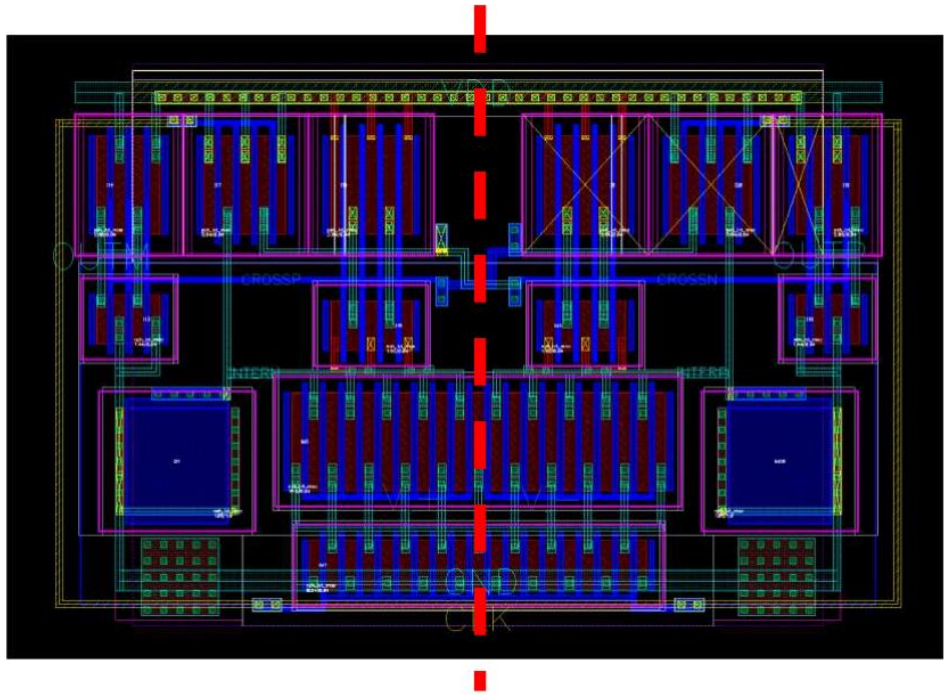
- Automatic constraint annotation is a critical step to existing analog layout automation tools, such as ALIGN [DAC'19, Kunal+], MAGICAL [ICCAD'19, Xu+]

Symmetry Constraint Annotation

- Symmetry Constraint
  - Symmetry constraint is one of the substantial and representative constraints for layout design



Comparator Schematic



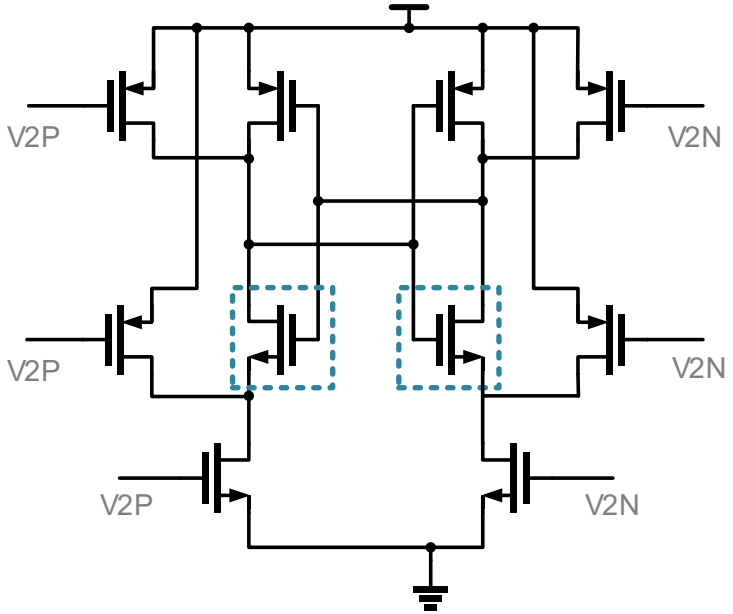
Comparator Layout

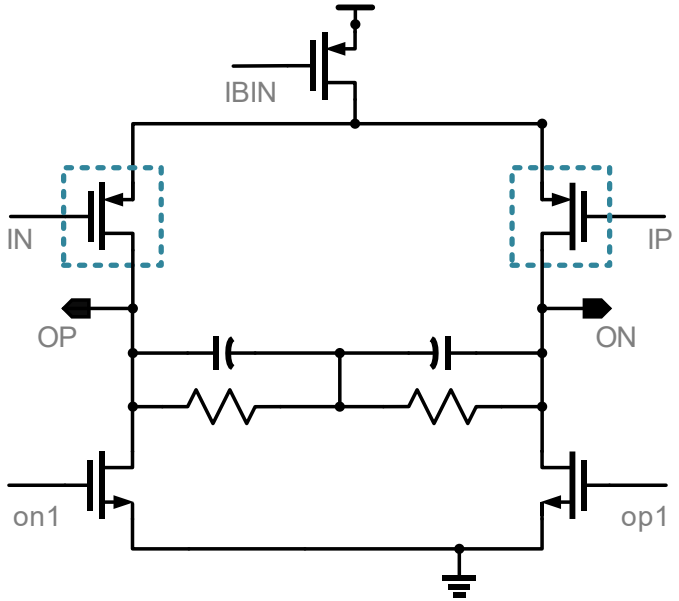
Prior Works and Challenges

- Categorized into two major types
  - Circuit analysis
  - Graph matching
- Circuit analysis
  - Traditional circuit analysis [ICCAD'93, Charbon+]
- Graph matching
  - Analog constraints extraction based on the signal flow analysis [ASICON'05, Zhou+]
  - S3DET: detecting symmetry constraint by [ASPDAC'11, Liu+]

Prior Works and Challenges

- Challenges
  - The performance highly-correlated to empirical parameters
  - The surrounding structures of symmetry constraints varies considerably
  - No incorporation of designer expertise





Graph Neural Networks (GNNs)

- Graph Neural Networks
  - Graph neural networks are powerful in modeling implicit features of graph structure data
  - Analog circuits → hypergraphs
- Transductive and inductive settings
  - Transductive

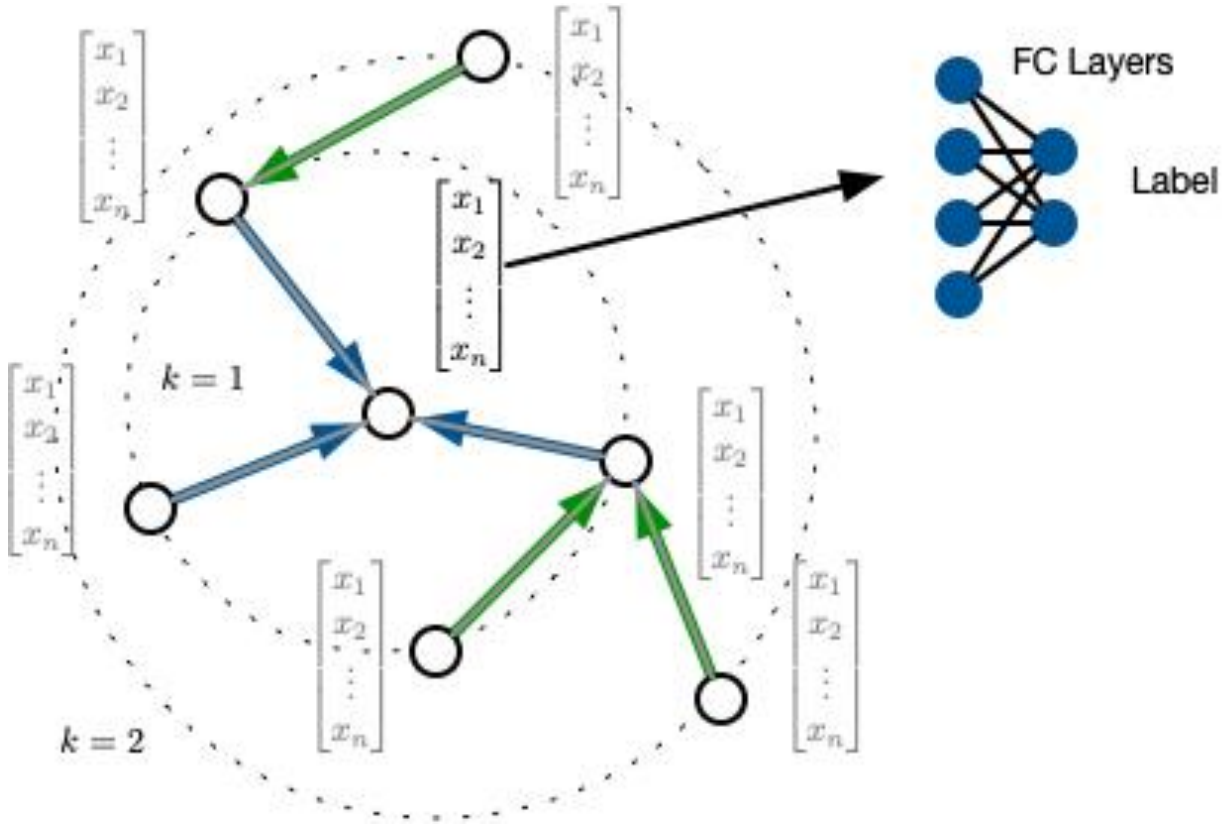
    GCN [ICLR'17, Kipf+]
  - Inductive

    GraphSage [NeurIPS'17, Hamilton+]
  - Transductive and inductive

    GAT [ICLR'18, Velickovic+]

GNNs: Node Embedding

- Node Embedding
  - Map each node in a graph into a low-dimensional space
  - Capture the insight from local graph structure and node features
- Sample and aggregation scheme
  - Compute node embeddings by sampling neighbor nodes and aggregating their embeddings

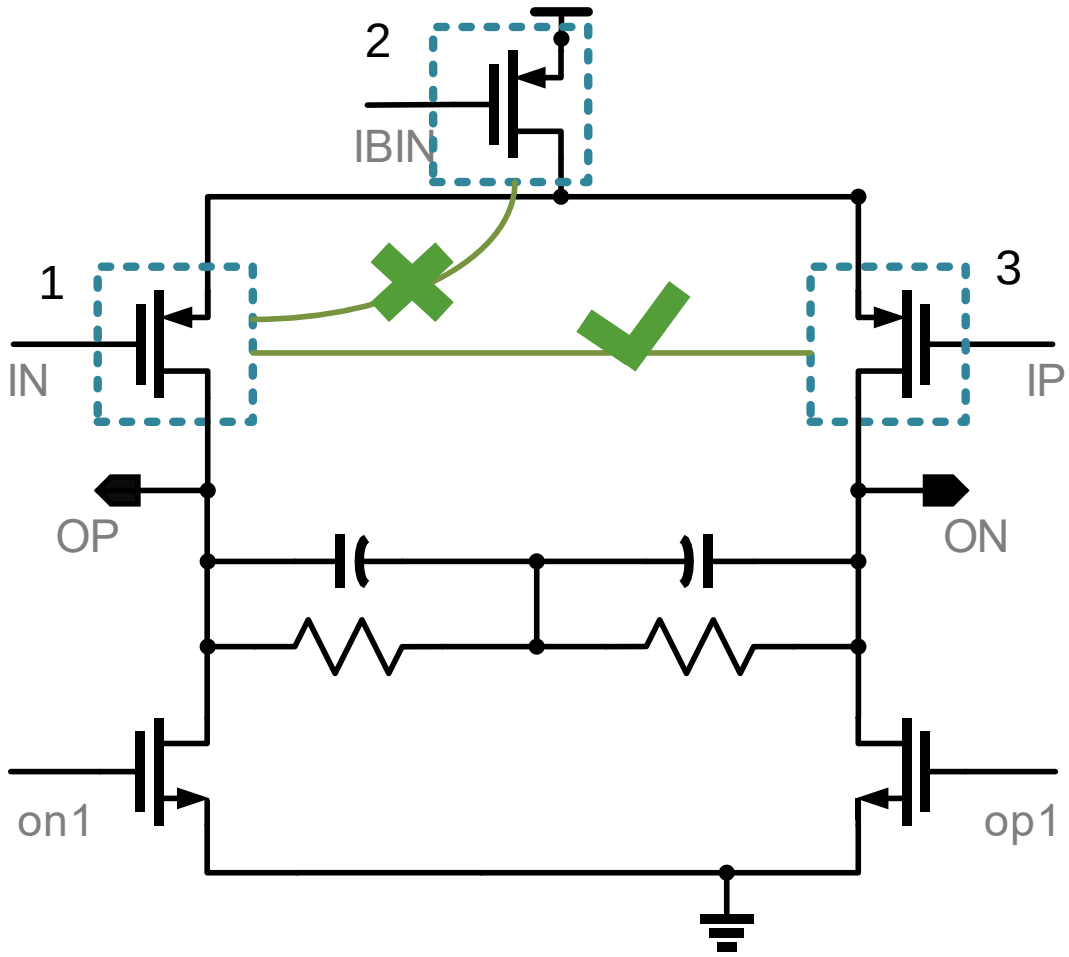


Problem Formulation

- Input
  - netlist
  - labeled symmetry pairs
- Output
  - For a new circuit

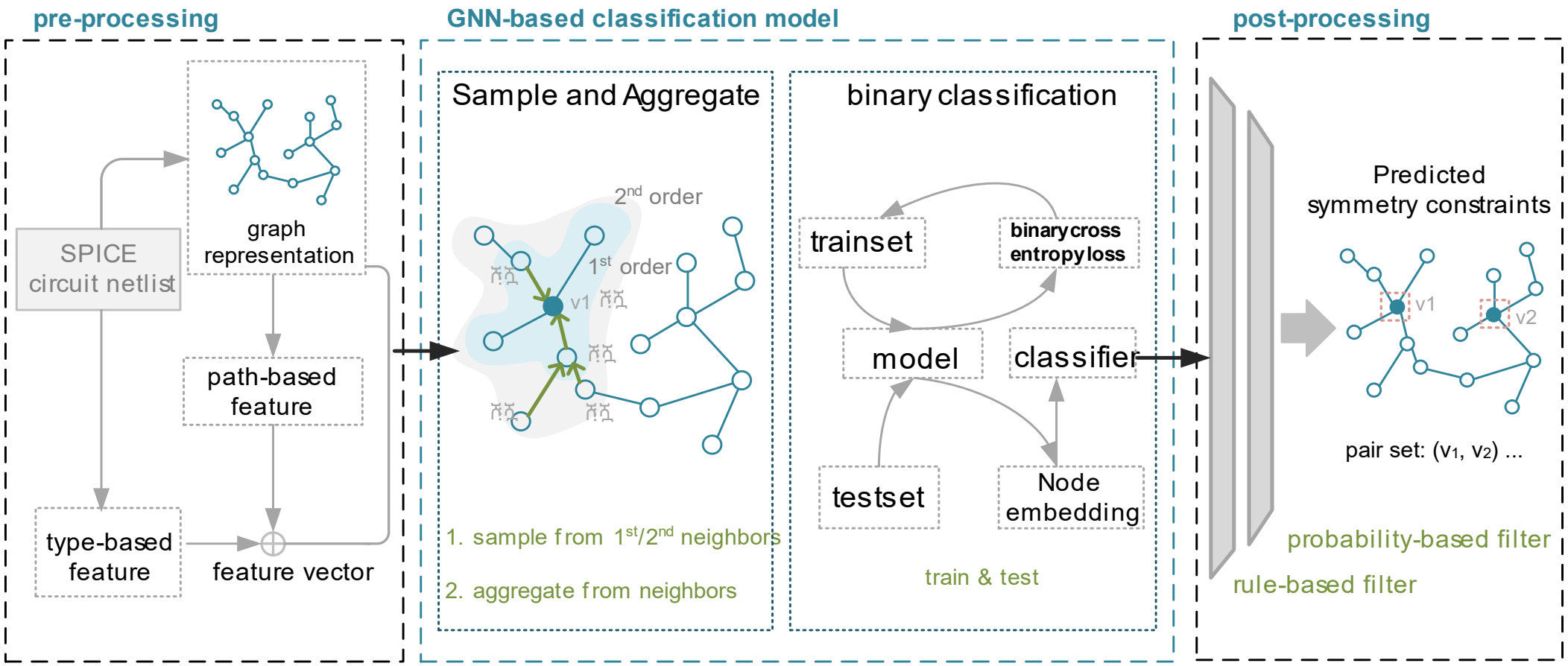
    to predict symmetry pairs
- Objective
  - to build a model with:

    maximum annotation accuracy



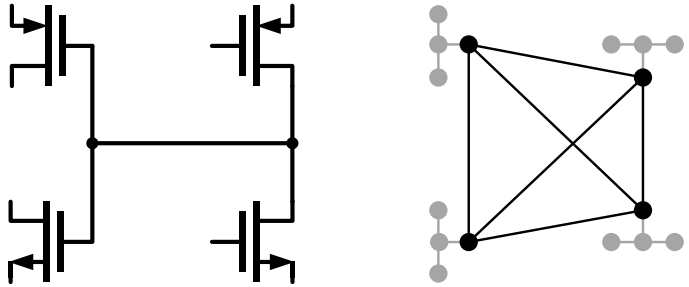
SymDetect: Overall Flow

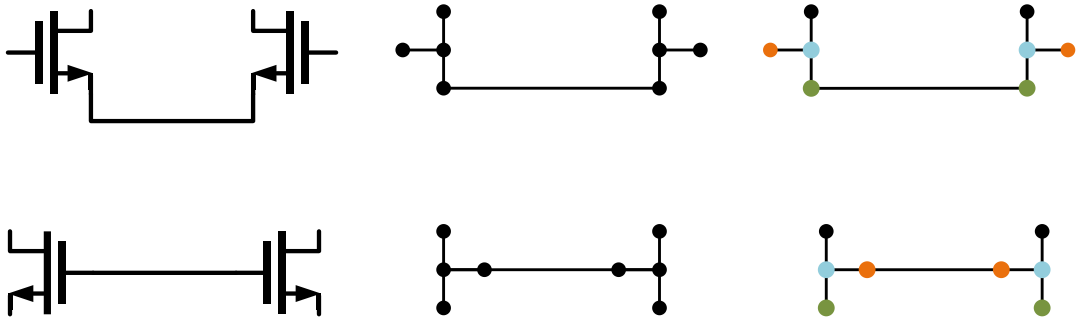
- Overall Flow of SymDetect



SymDetect: Graph Representation

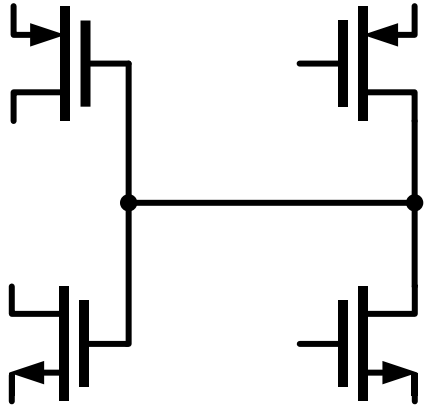
- Graph Representation
  - Both **device instances** and **device pins** are recognized as **graph nodes**
  - Edges connecting pin nodes with their corresponding device nodes
  - The pin nodes form a clique if sharing the same net

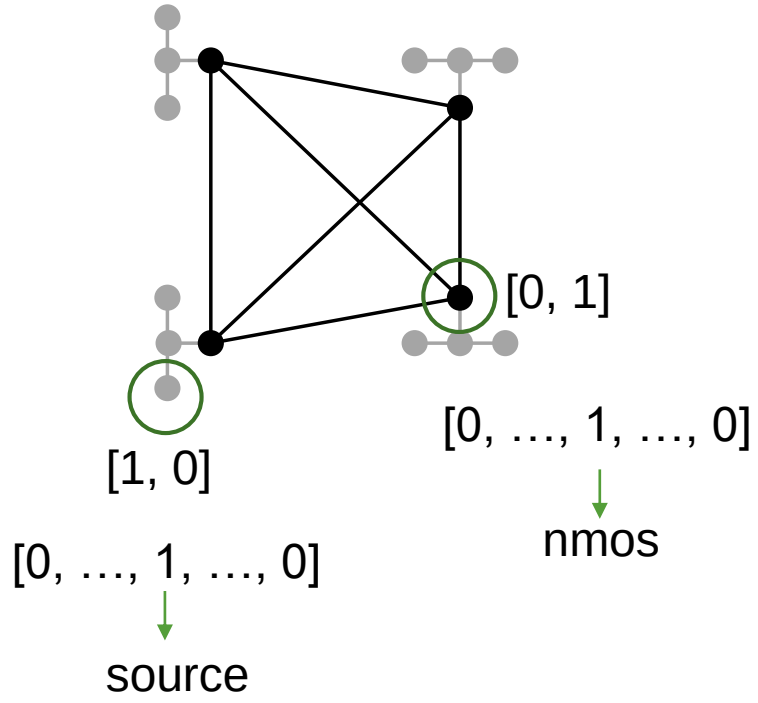




SymDetect: Feature Extraction

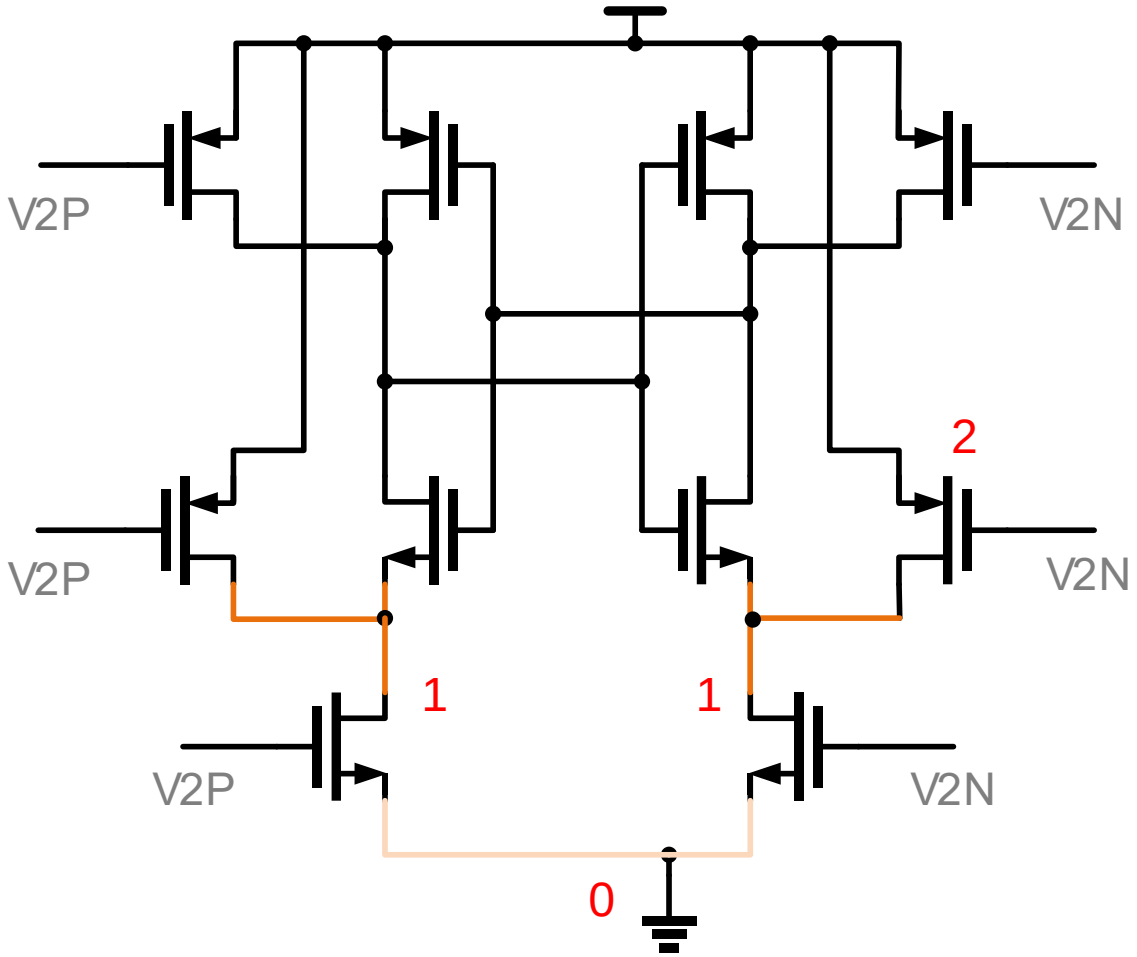
- Feature extraction
  - Type-based feature
  - Path-based feature
- Type-based feature
  - Type information as part of node feature
  - A 2-dimension vector to indicate whether a node is a device or a pin
    [0, 1] for a device, [1, 0] for a pin
  - A 13-dimension **one-hot vector**
    device types: capacitor, resistor, diode, NMOS, PMOS, IO
    pin types: source, drain, gate, substrate, passive, cathode of a diode, anode of a diode





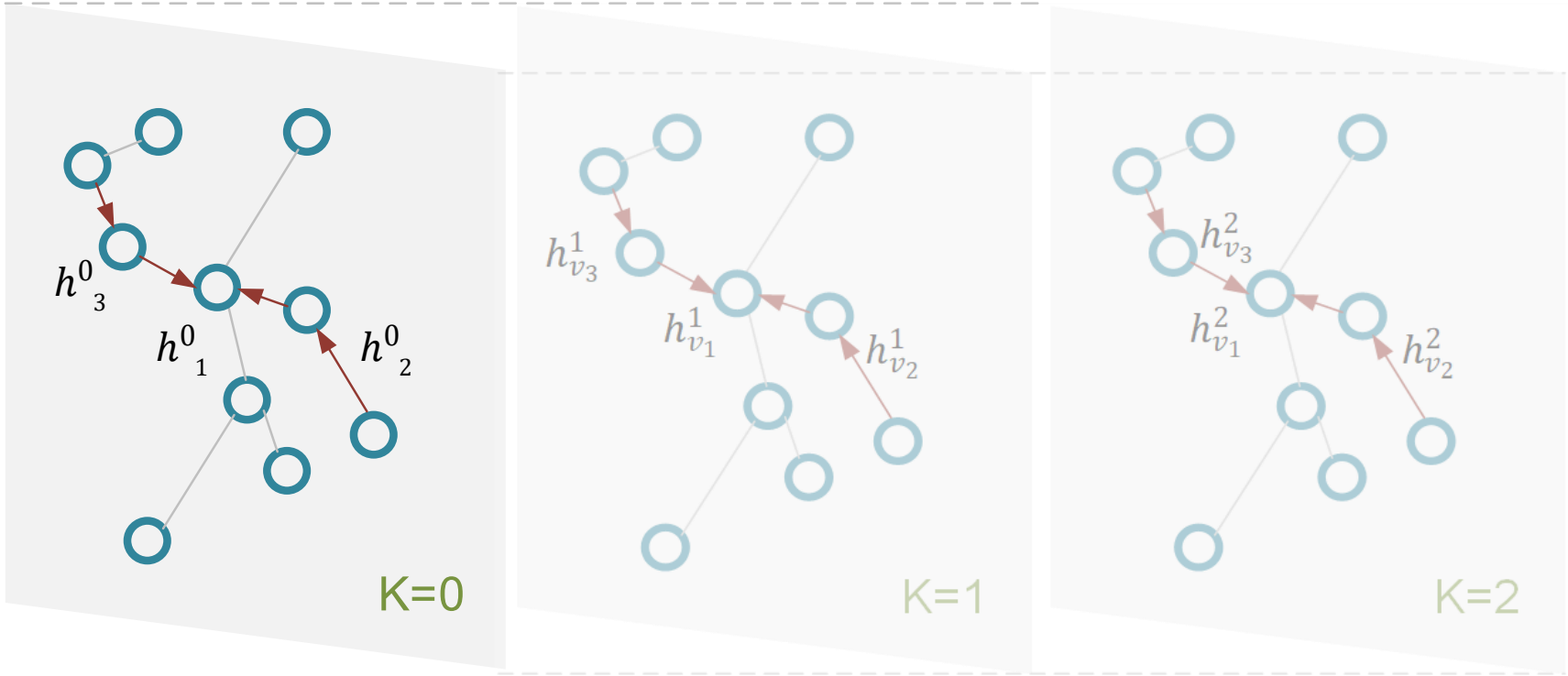
SymDetect: Feature Extraction

- Feature Extraction
  - Type-based feature
  - Path-based feature
- Path-based feature
  - Characterize the global position of each node by computing the length of VSS/GND-sourced paths



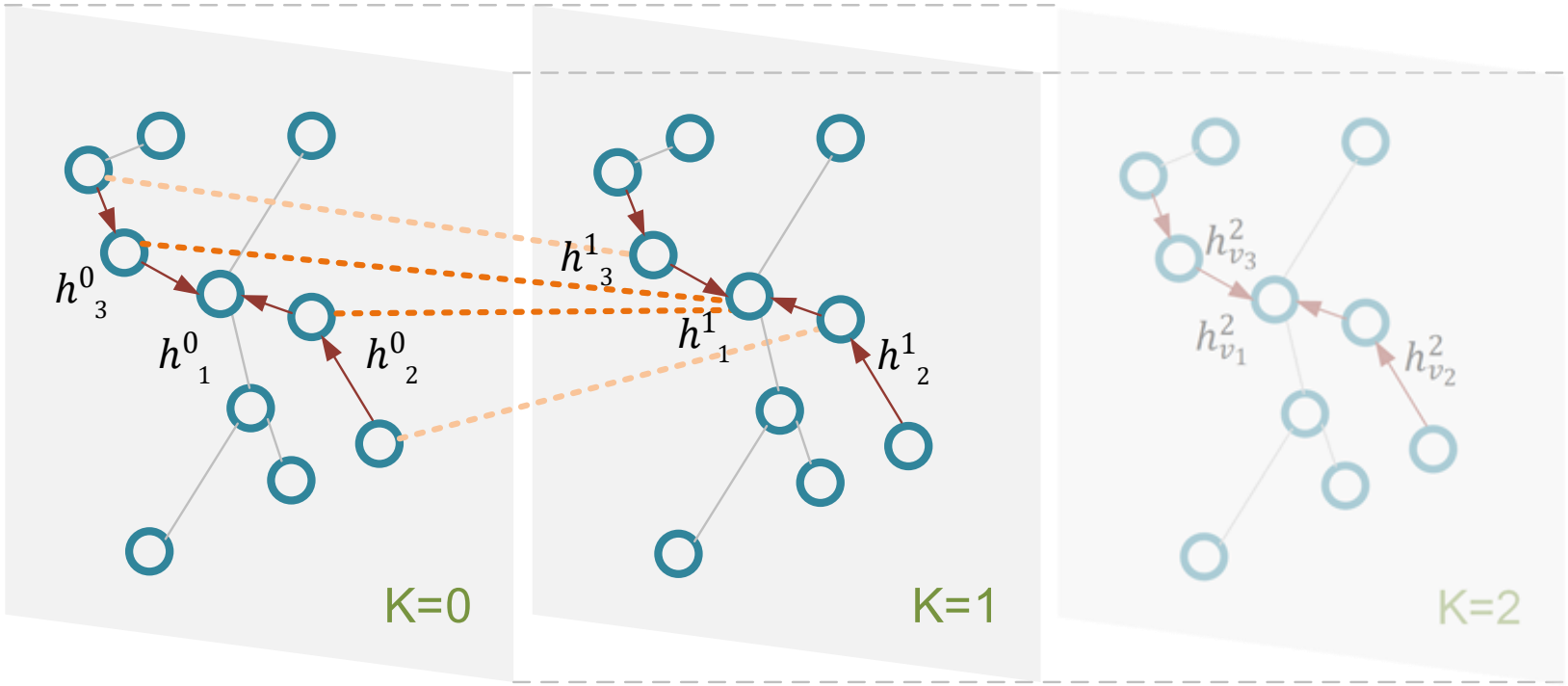
SymDetect: Sample and Aggregate Model

- Sage
  - Sample from neighborhood



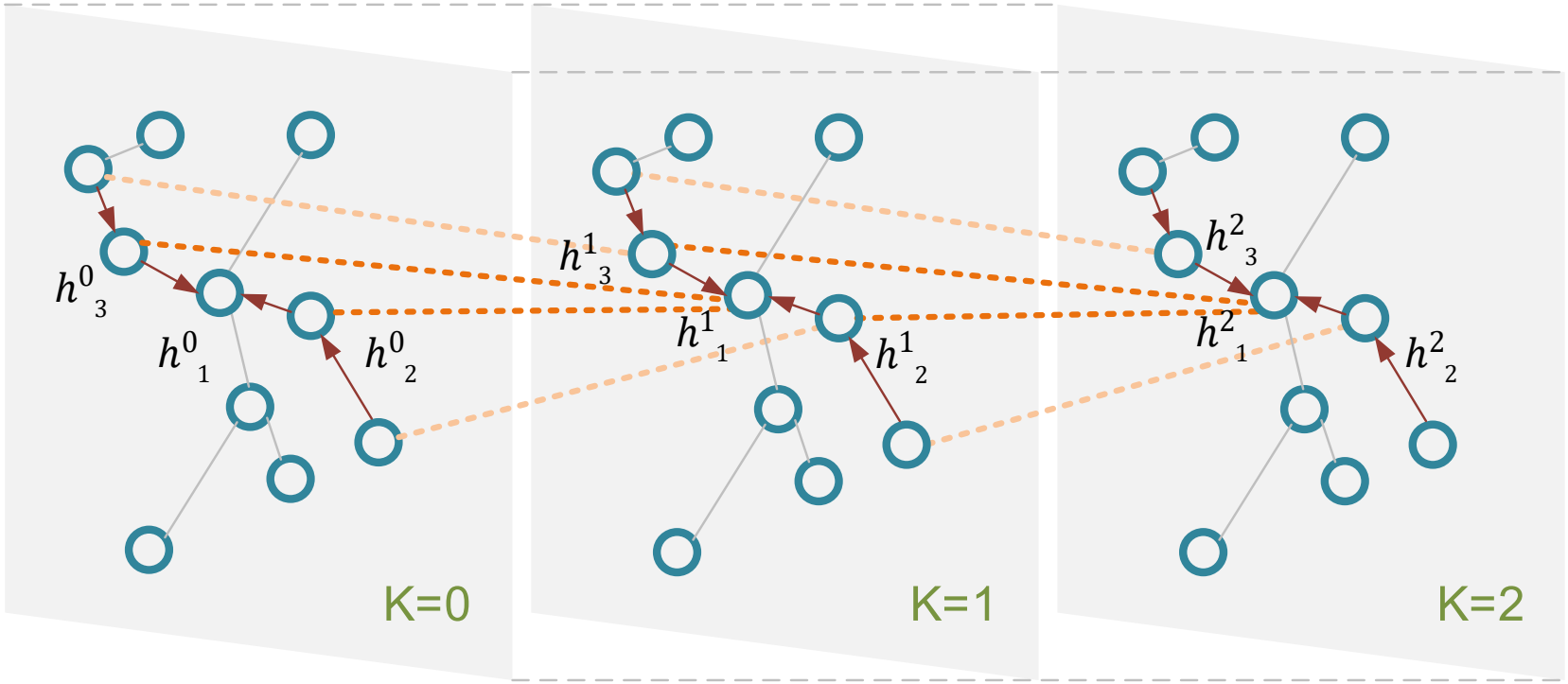
SymDetect: Sample and Aggregate Model

- Sage
  - Aggregate by aggregator function
  - $h_v^k = \sigma(W \cdot \{h_v^{\{k-1\}} \oplus MEAN(h_u^{\{k-1\}}, \forall u \in \boldsymbol{N}(v))\})$



SymDetect: Sample and Aggregate Model

- Sage
  - Take aggregation depth K=2



SymDetect: Binary Classification

- Binary Classification
  - We designate a **label** y for each node pair (v_1, v_2)

    1 for symmetric, 0 for non-symmetric
  - We calculate the probability of label $y = 1$ by **bilinear scoring function**:

    $$prob = \sigma(z_1^T W z_2)$$

    where $\sigma(\cdot)$ is an activation function, z_1 and z_2 are node embeddings
  - We apply binary **cross entropy loss** to train our GNN model:

    $$loss = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(prob_i) + (1 - y_i) \cdot \log(1 - prob_i)$$

SymDetect: Post Processing

- Post Processing
  - To reduce false alarms
- Rule-based filter
  - The **sizes and types** of two symmetric nodes are supposed to be identical
- Probability-based filter
  - A device node appears in no more than one symmetry pair
  - The filter sorts all candidate pairs by predicted probabilities
  - Pick the pair with **higher probability**

Experimental Setup

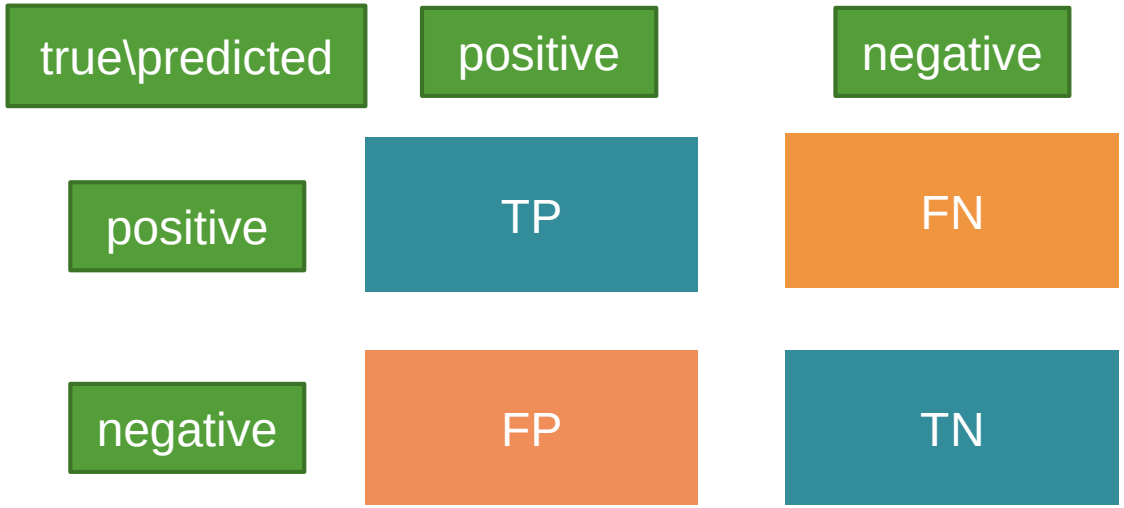
- Baseline methods
  - S^{3}DET [ASPDAC'20, Liu+]
  - signal flow analysis (SFA) [ICCAD'19, Xu+]
- Dataset statistics

Datasets	Circuits	Nodes	Edges	Valid pairs	Pos/Neg
S^3-leaf	10	1378	9149	1522	89/1433
ALIGN-leaf	5	580	2134	576	48/528
OTA	5	684	3422	750	45/705

Experimental Setup

- Evaluation metric
  - We adopt: *true positive rate*(*TPR*), *false positive rate*(*FPR*), and *F1-score* and *ROC*

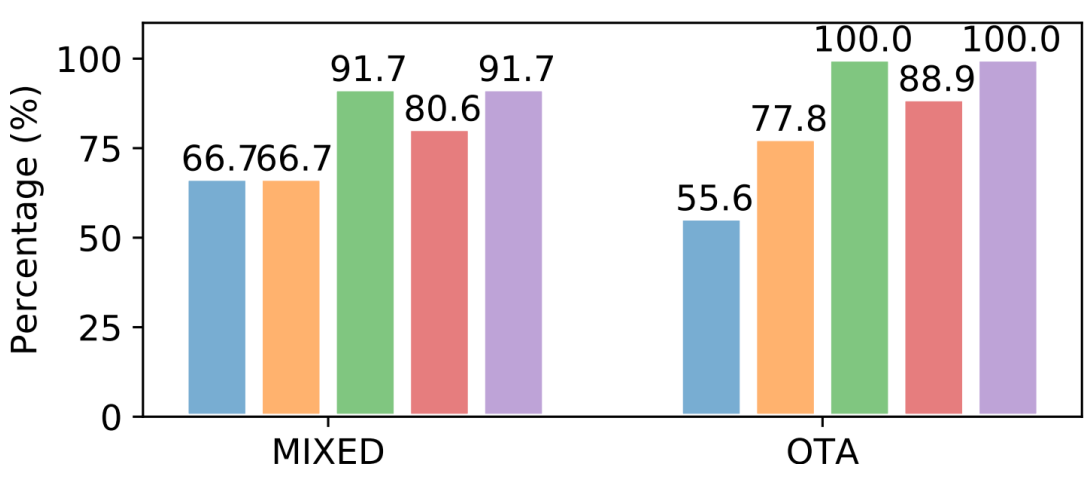
  - $TPR = \frac{TP}{P}$
  - $FPR = \frac{FP}{N}$
  - $F1 - score = \frac{2TP}{2TP+FN+FP}$
  - *ROC* plots the *TPR* against the *FPR*

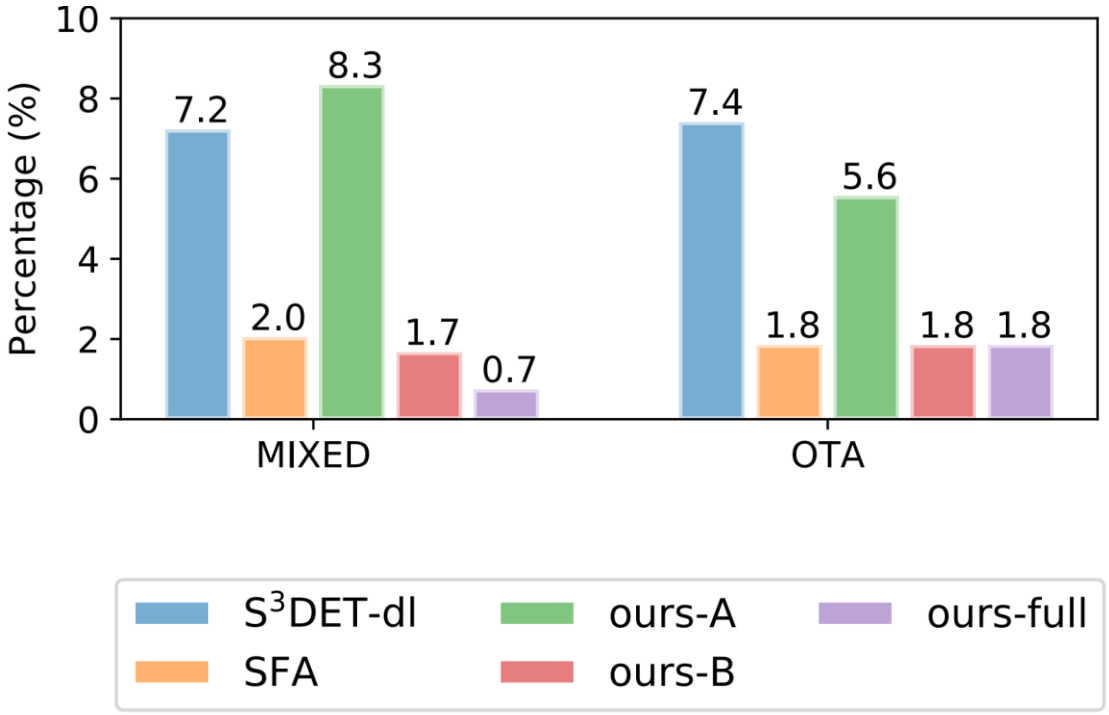


confusion matrix

Experimental Results

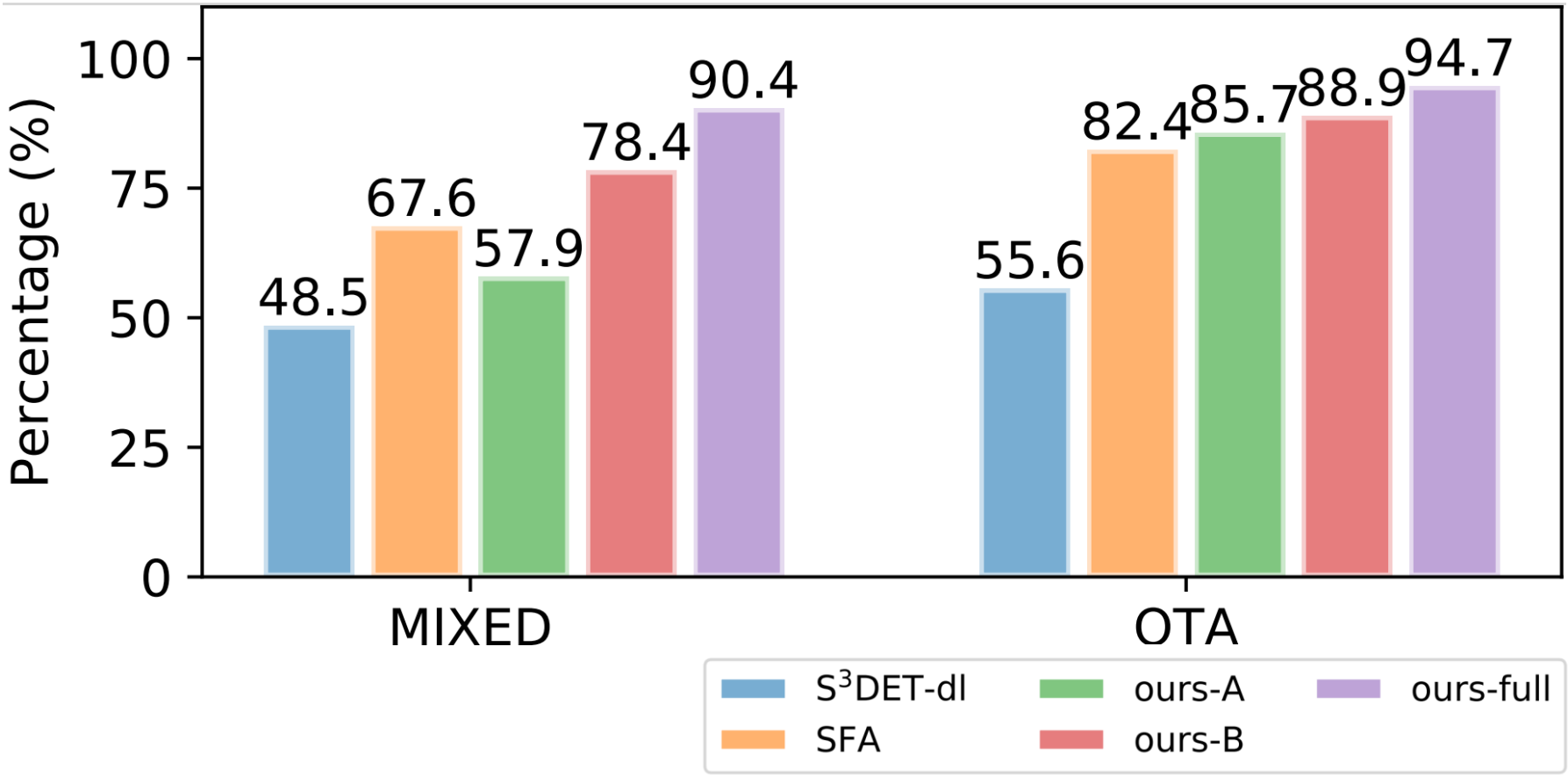
- Experimental Results
  - TPR and FPR:





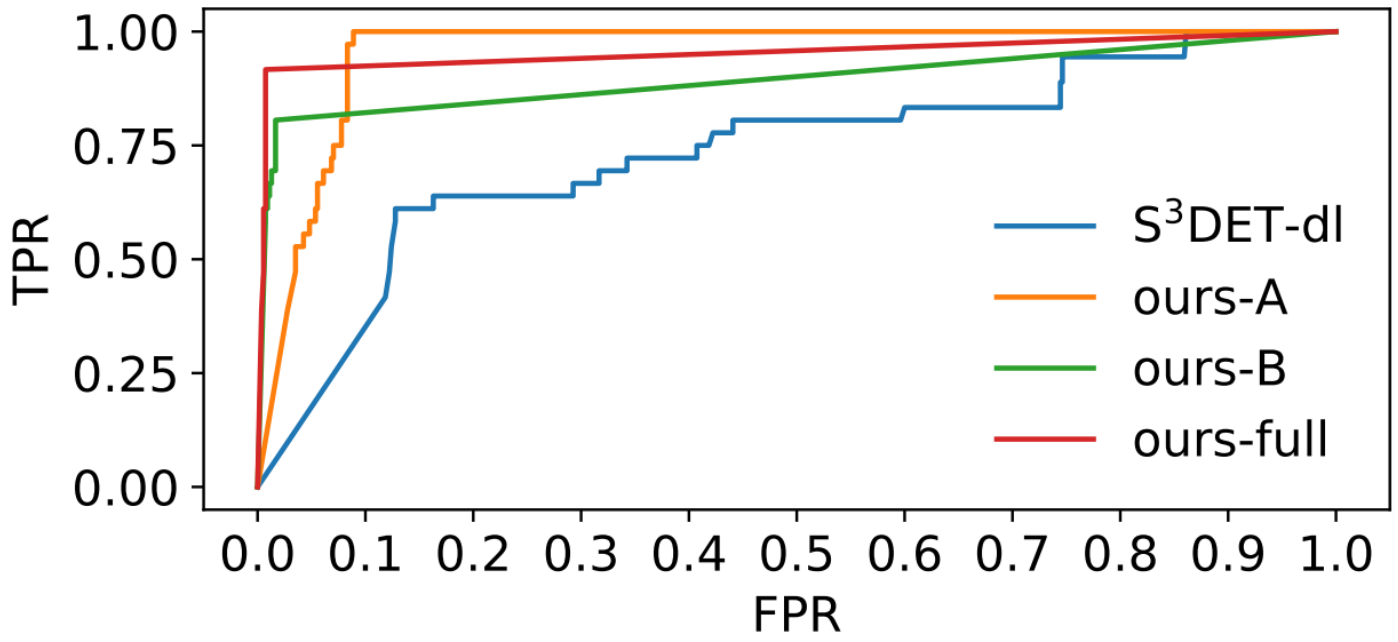
Experimental Results

- Experimental Results
  - F1-score:



Experimental Results

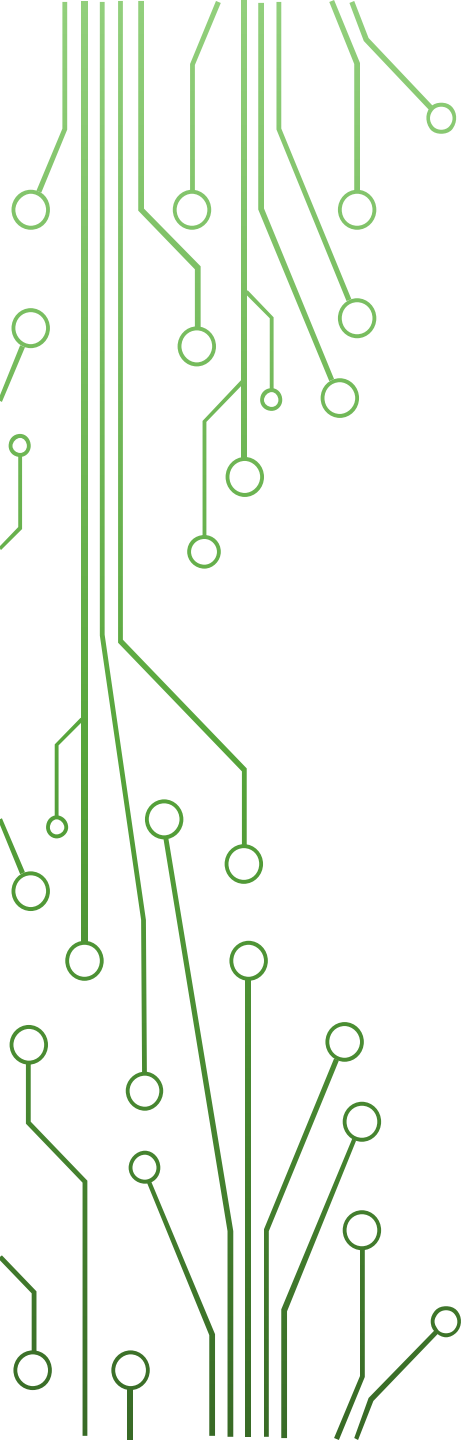
- Experimental Results
  - ROC:



  - Runtime:
  - The one-shot training time costs less than 2 minutes
  - The inference time per circuit (about 0.1s) is comparable to the other two approaches

Conclusions and Future Work

- Conclusions
  - A **graph learning based** framework for layout symmetry annotation in analog circuits
  - Node feature extraction from **both local information** (device/pin type) and **global graph structure** (path length from VSS/GND)
  - A training technique to learn the node similarity from the **imbalanced data**
  - The proposed approach **outperforms** previous symmetry annotation
- Future work
  - Extend our GNN-based approach to **system symmetry constraint**
  - We can also include **other constraints** such as matching constraint or shielding constraint



北京大学高能效计算与应用中心
Center for Energy-efficient Computing and Applications

Thanks!
Questions are welcome

Email: xiaohangao@pku.edu.cn