DESIGN, AUTOMATION & TEST IN EUROPE
09 – 13 March 2020 · ALPEXPO · Grenoble · France
The European Event for Electronic System Design & Test

Towards Decrypting the Art of Analog Layout: Placement Quality Prediction via Transfer Learning

Mingjie Liu*, Keren Zhu*, Jiaqi Gu, Linxiao Shen, Xiyuan Tang, Nan Sun, and David Z. Pan
Dept. of Electrical and Computer Engineering
The University of Texas at Austin

* Indicates equal contributions.

Outline

- Introduction and Motivation
- UT-AnLay Dataset with MAGICAL
- Placement Quality Prediction
- Improved Data Efficiency with Transfer Learning
- Conclusions

Outline

- Introduction and Motivation
- UT-AnLay Dataset with MAGICAL
- Placement Quality Prediction
- Improved Data Efficiency with Transfer Learning
- Conclusions

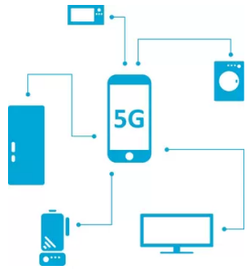
Analog/Mixed-Signal IC Demand

- High demand of analog/mixed-signal (AMS) IC in emerging applications

Advanced computing

Healthcare

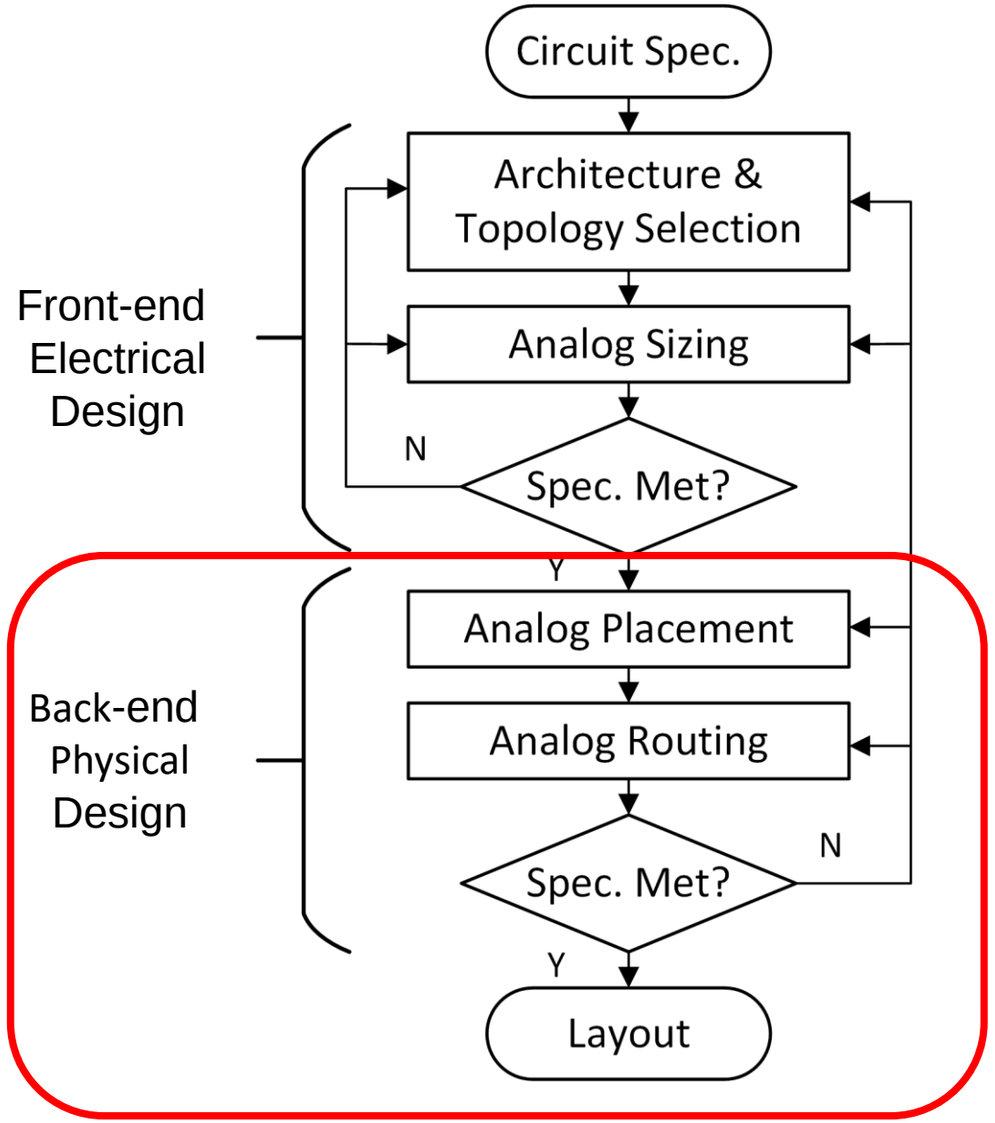
Automotive



Communication

Image Sources: IBM, Ansys, public technology

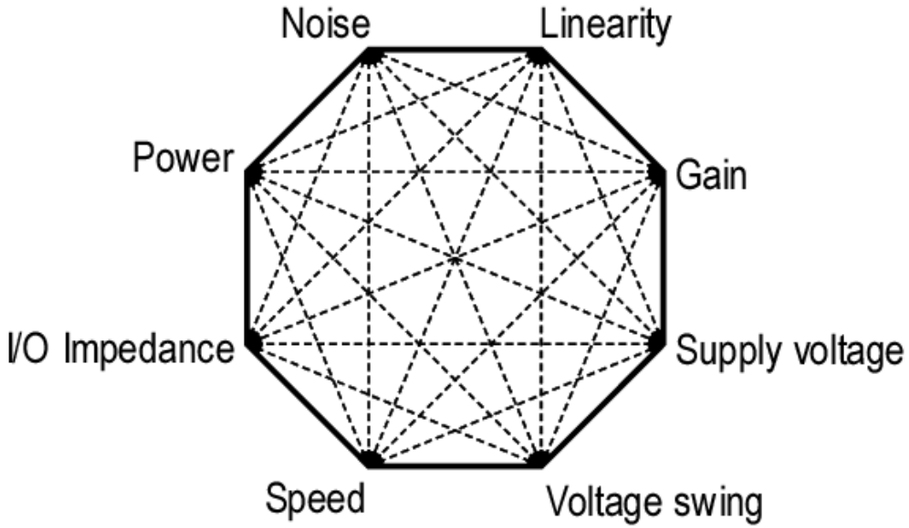
Analog/Mixed-Signal IC Design Challenges



- Repetitive iterations and feedback during manual design flows
- Close interactions with circuit designers and layout engineers
- Our focus is on back-end physical design (layout) stage
- Provide design closure and guarantee to ***meet specification***, manufacturability, reliability, etc…

Challenges in Analog Layout Design

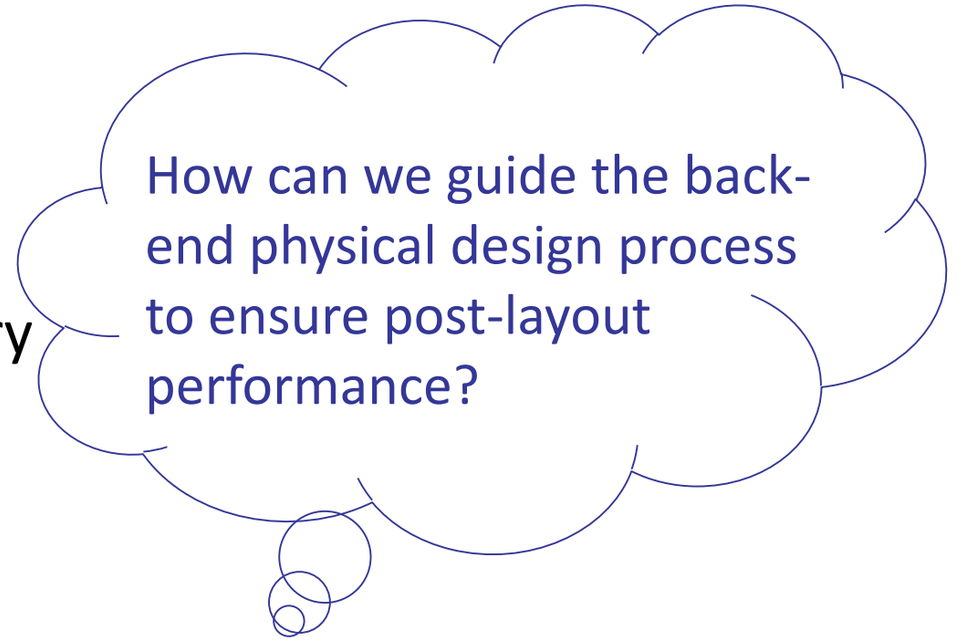
- Multiple performance trade-offs
  - No uniform representation for performance
  - Each design is “unique”
- Complex layout dependent effects
  - BSIM4 model >250 parameters
  - Increased parasitic
  - Well proximity effect (WPE), substrate noise coupling, etc…
- Complex layout design rules



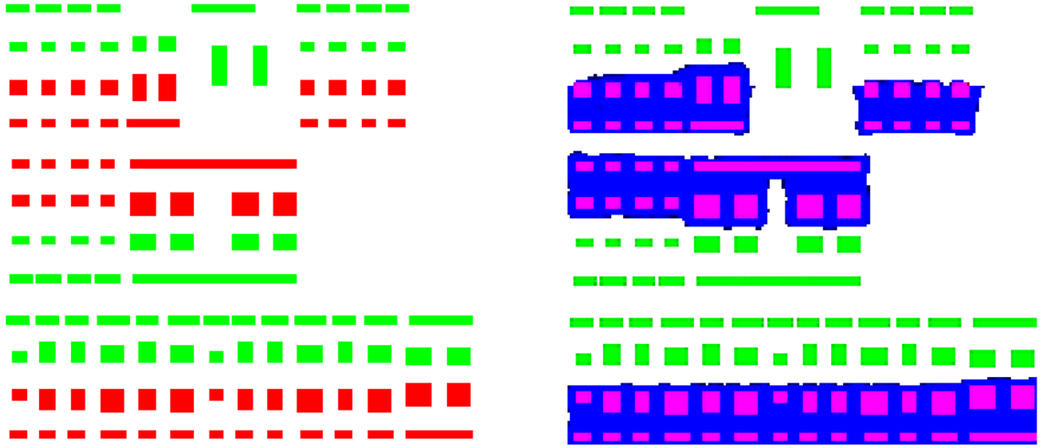
Behzad Razavi, 2000

Prior Work on Analog Layout

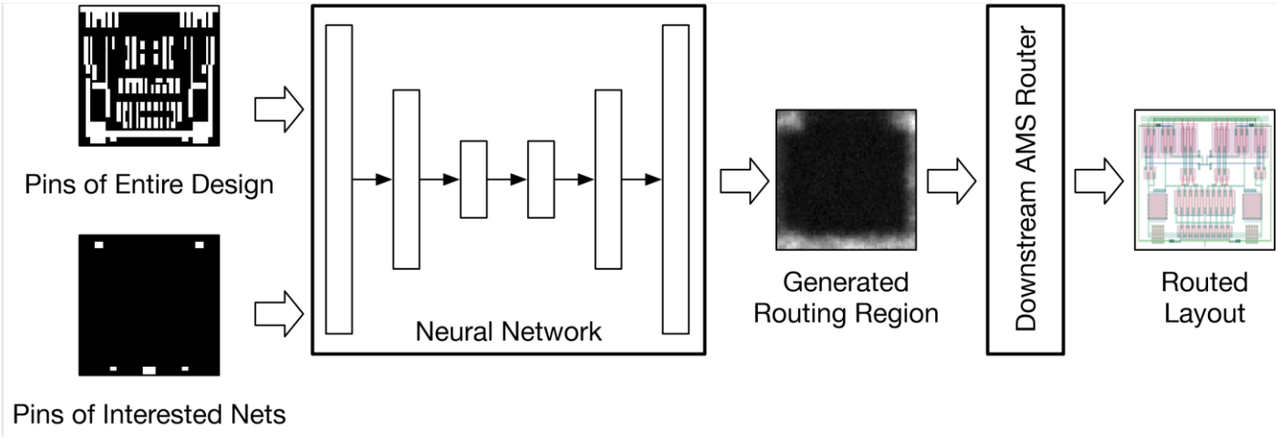
- Sensitivity analysis based optimization:
  - ✓Model how parasitic effects performance
  - ✓Some guarantee on performance
  - ×Simulations are too expensive for systems
- Heuristic constraint based:
  - ✓Encode in layout algorithms
  - ✓Enforced satisfiability for crucial effects: symmetry
  - ×Difficult to enumerate
  - ×No room for trade-offs: contradictory constraints
  - ×Limited guarantee towards performance



Prior Work on Analog Layout



WellGAN [Xu et al., DAC, 2019]



GeniusRoute [Zhu et al., ICCAD, 2019]

- Leveraging generative neural network
  - × Data hungry algorithms
  - × Good human layout examples
  - × Technology dependent: difficult to share data
  - × No explicit optimization on performance

Decrypting the Art of Analog Layout

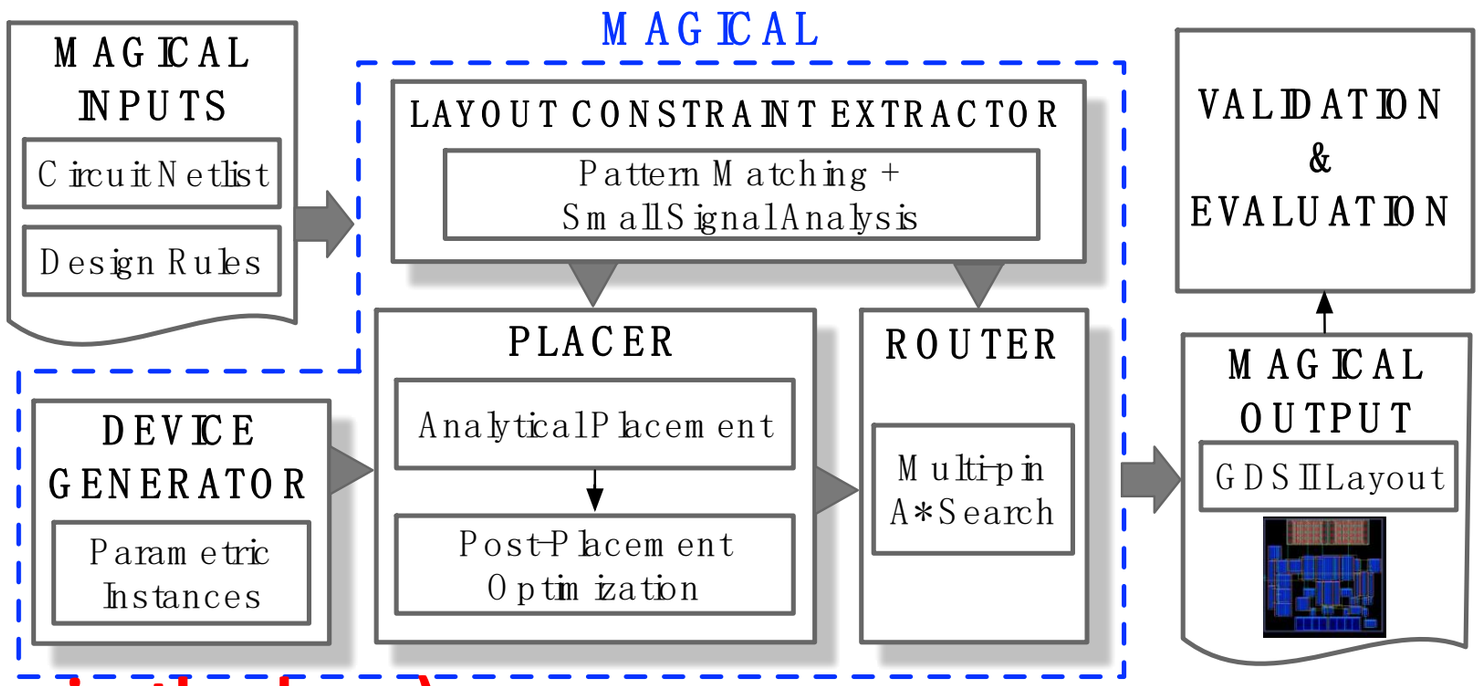
- "The process of constructing layouts for analog and mixed signal circuits have stubbornly defied all attempts at automation." [Hastings, The Art of Analog Layout, 2001]
- Our contributions:
  - A model for placement quality prediction for fast design space explorations
  - Automatically generated simulated layout training data with MAGICAL
  - 3D convolutional neural network with coordinate channel embeddings
  - Leveraging transfer learning for improved data efficiency
  - Open-sourced on GitHub: https://github.com/magical-eda/UT-AnLay

Outline

- Introduction and Motivation
- UT-AnLay Dataset with MAGICAL
- Placement Quality Prediction
- Improved Data Efficiency with Transfer Learning
- Conclusions

MAGICAL Layout System

- Input: unannotated netlist
- Output: GDSII Layout
- Key Components:
  - Device Generation
  - Constraint Extraction
  - Analog Placement
  - Analog Routing



- **Fully-automated (no-human-in-the-loop)**
- Guided by analytical, heuristic, and machine learning algorithms
- **Open-sourced** on GitHub: https://github.com/magical-eda/MAGICAL

Analytical Global Placement

- Objective:

$$Objective = f_{WL} + a \cdot f_{OL} + b \cdot f_{BND} + c \cdot (f_{SYM}^{x} + f_{SYM}^{y})$$

- Performance:
  - Wirelength term (half-perimeter wirelength):

$$f_{WL} = \Sigma_{n_k}(\max_{i \in n_k} x_i - \min_{i \in n_k} x_i + \max_{i \in n_k} y_i - \min_{i \in n_k} y_i)$$

- Relaxed Constraints:

f_{OL} : *Device overlap cost*

f_{BND}: *Layout boundary cost*

f_{SYM}^{x} , f_{SYM}^{y}: *Device symmetry constraint*

- Wirelength term (half-perimeter wirelength):

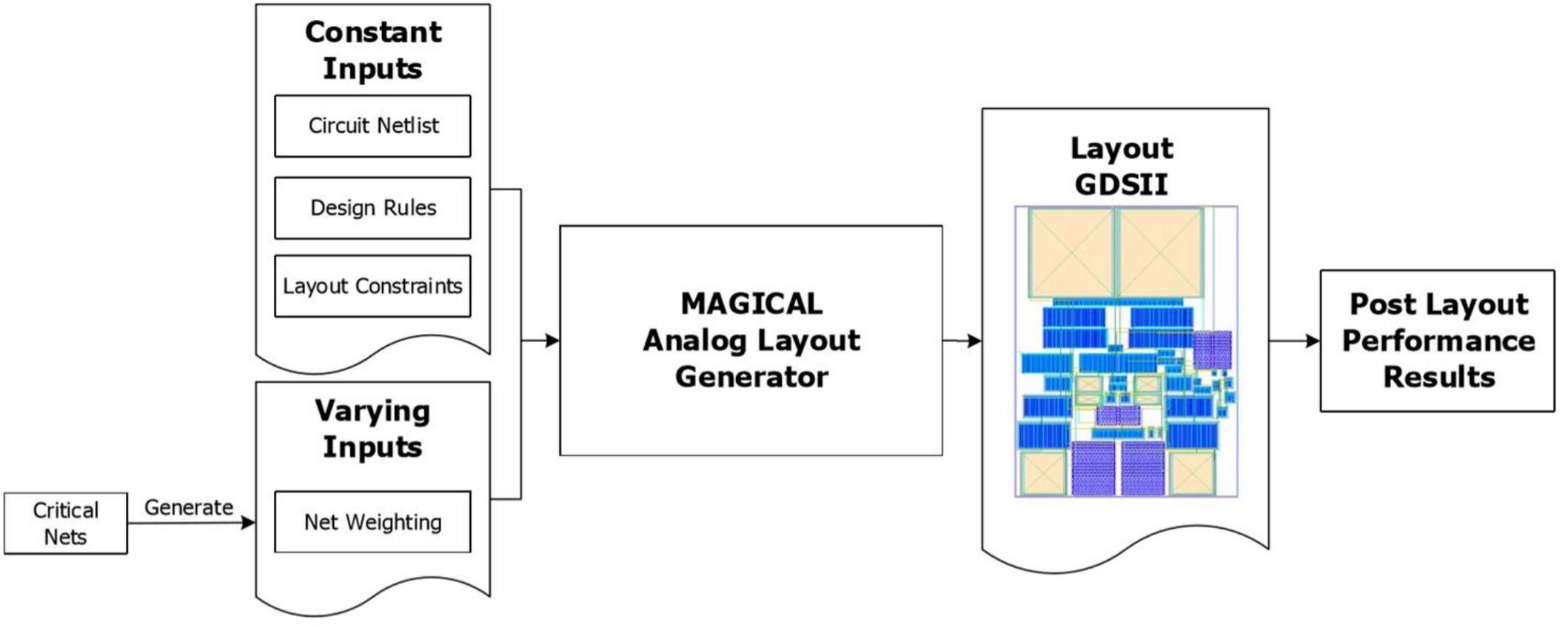
$$f_{WL} = \Sigma_{n_k}(\max_{i \in n_k} x_i - \min_{i \in n_k} x_i + \max_{i \in n_k} y_i - \min_{i \in n_k} y_i)$$

× Not a good indication of performance

× Different nets should have different importance

$$f_{WL} = \Sigma_{n_k} \alpha_{n_k}(\max_{i \in n_k} x_i - \min_{i \in n_k} x_i + \max_{i \in n_k} y_i - \min_{i \in n_k} y_i)$$

- Different penalty term indicating net criticality
- Allow multiple layout solutions for same schematic


Constant Inputs
Circuit Netlist
Design Rules
Layout Constraints
Varying Inputs
Net Weighting
Critical Nets
Generate
MAGICAL Analog Layout Generator
Layout GDSII
Post Layout Performance Results

- UT-AnLay Dataset:
  - Industrial level parasitic extraction and simulation tool
  - Custom designed testing benchmark suite
  - Over 16,000 different layout for each design

Design	Stage	Compensation	Layouts
OTA1	3	Nested Miller	16376
OTA2	3	Nested Miller	16381
OTA3	2	Miller	16384
OTA4	2	None	16363

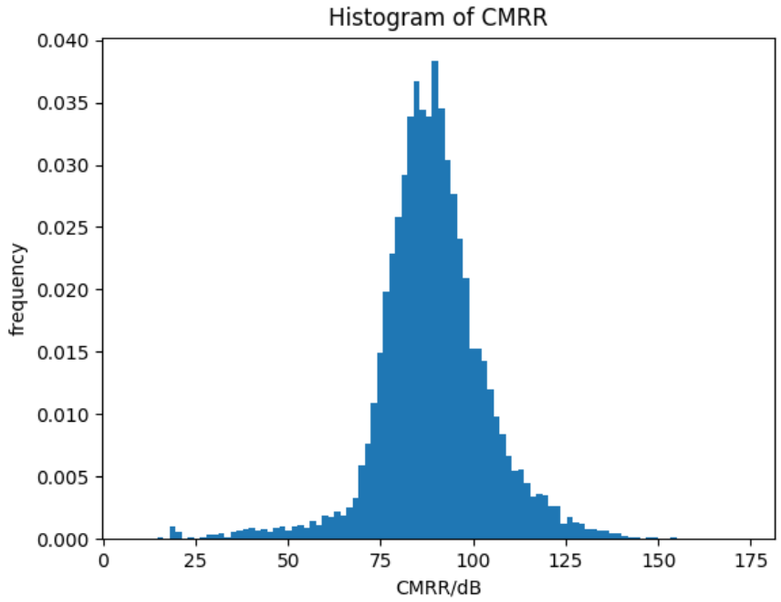
- UT-AnLay Dataset:
  - Circuit netlist
  - Device boundary box
  - Device placement coordinates (with pin coordinates)
  - Post layout simulation results

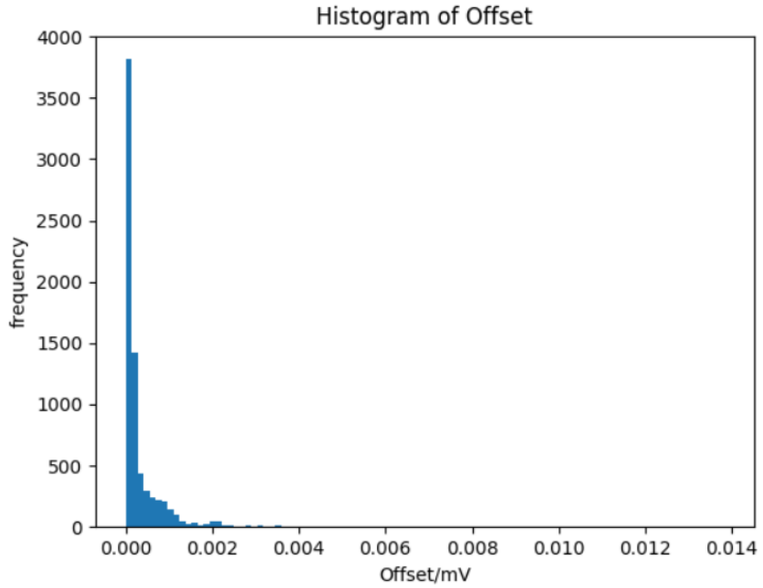
  X Routing information

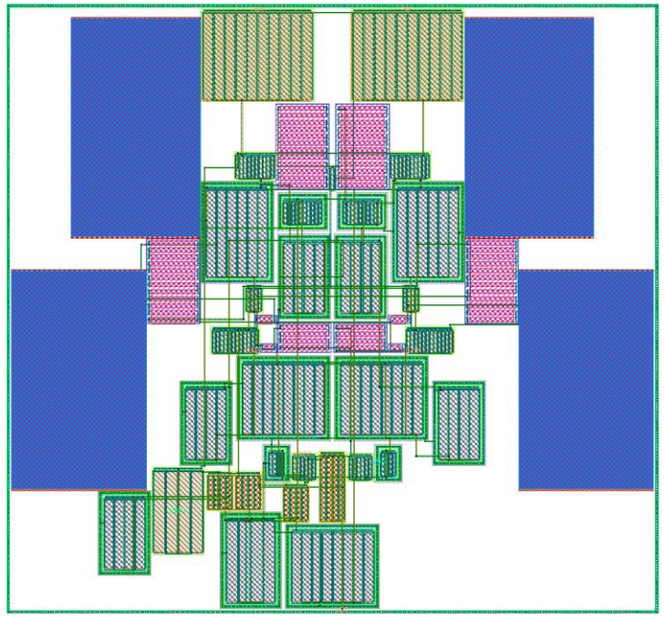
  X Currently only OTA circuits

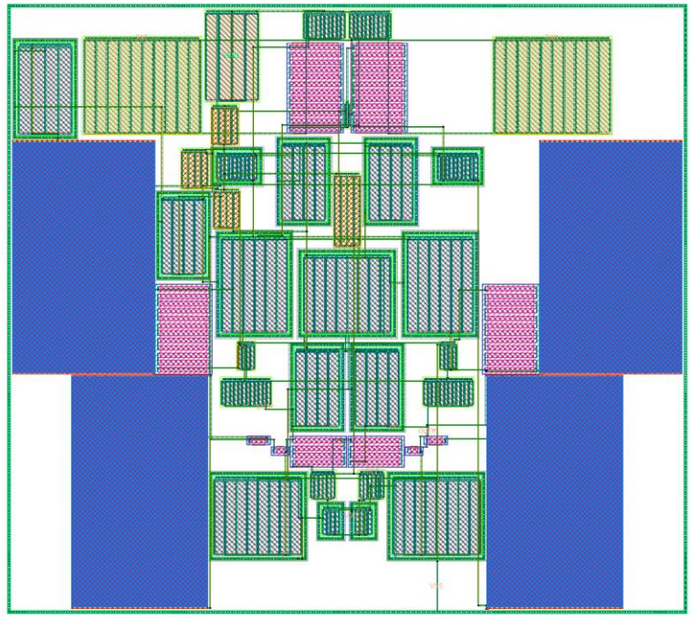
  X Currently only in TSMC 40nm technology

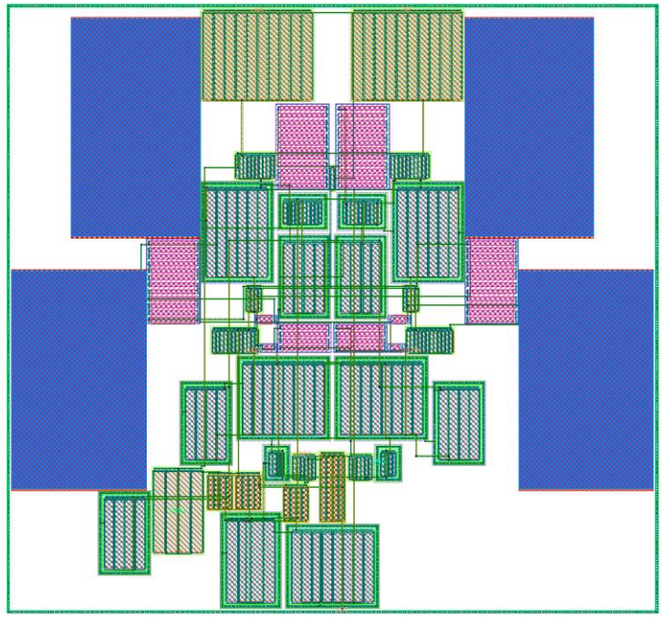
- UT-AnLay Dataset:
  - Noticeable difference in layout implementations
  - High variations in some performance
    - CMRR and offset -> Large variation
    - Gain, power, phase margin etc. -> Small variation

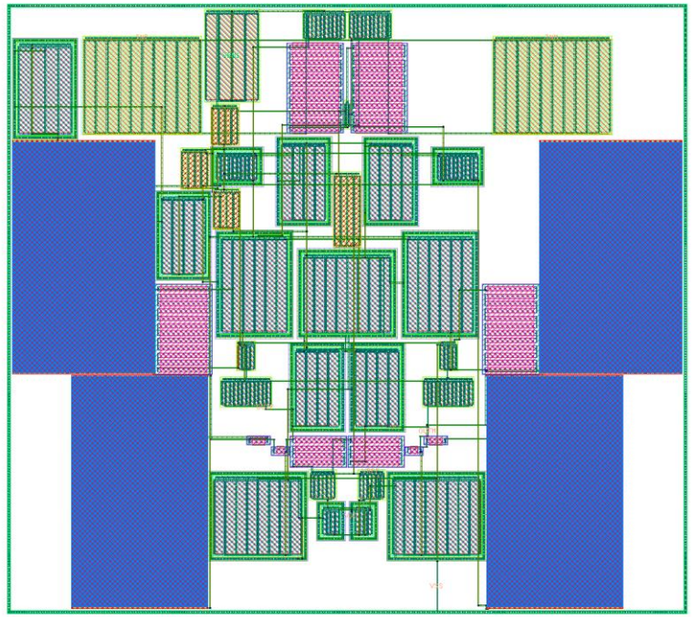












Offset (mV)	~ 0
CMRR (dB)	110

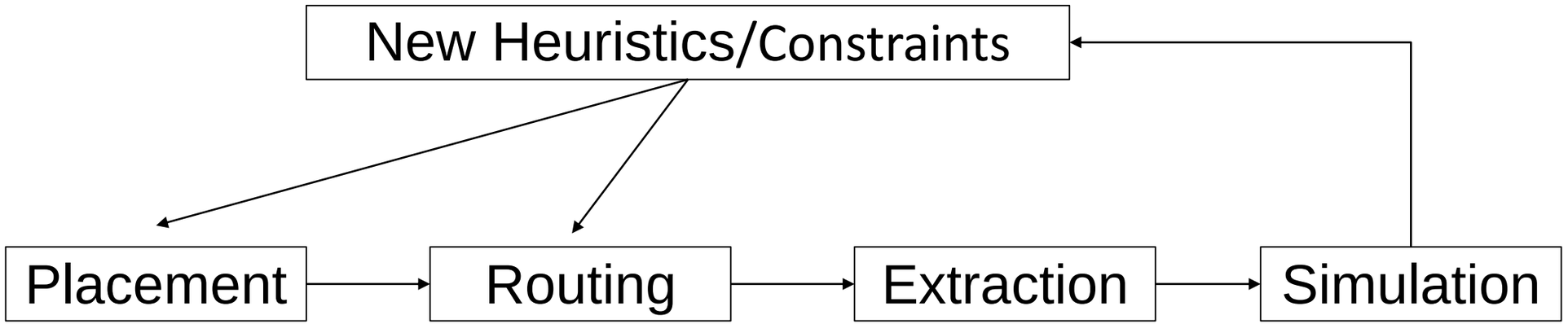
???

Offset （mV)	5.0
CMRR (dB)	76.3

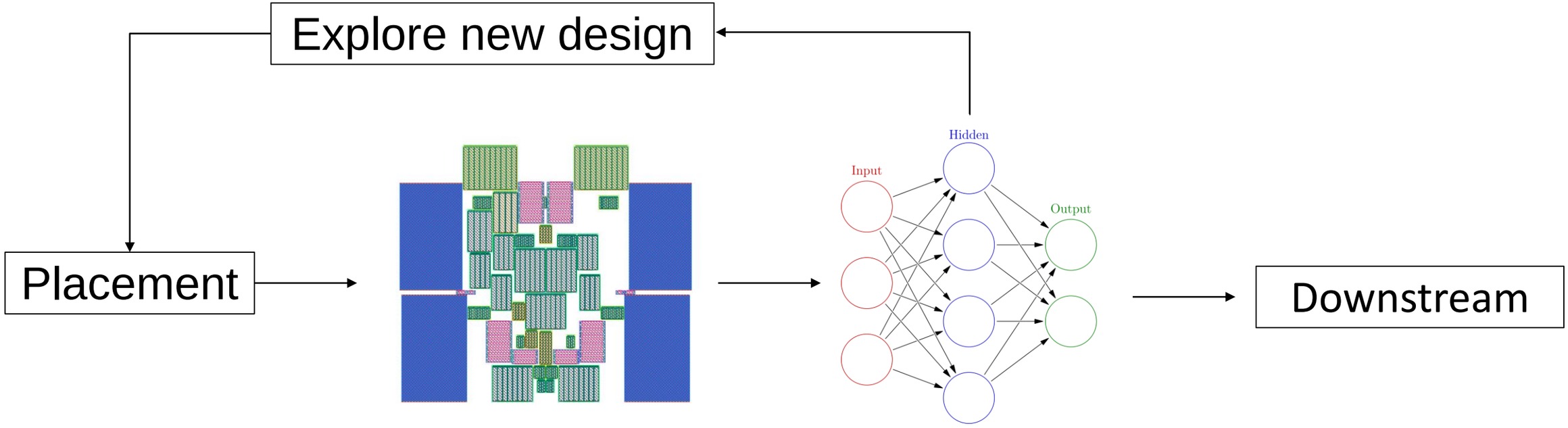
Outline

- Introduction and Motivation
- UT-AnLay Dataset with MAGICAL
- Placement Quality Prediction
- Improved Data Efficiency with Transfer Learning
- Conclusions

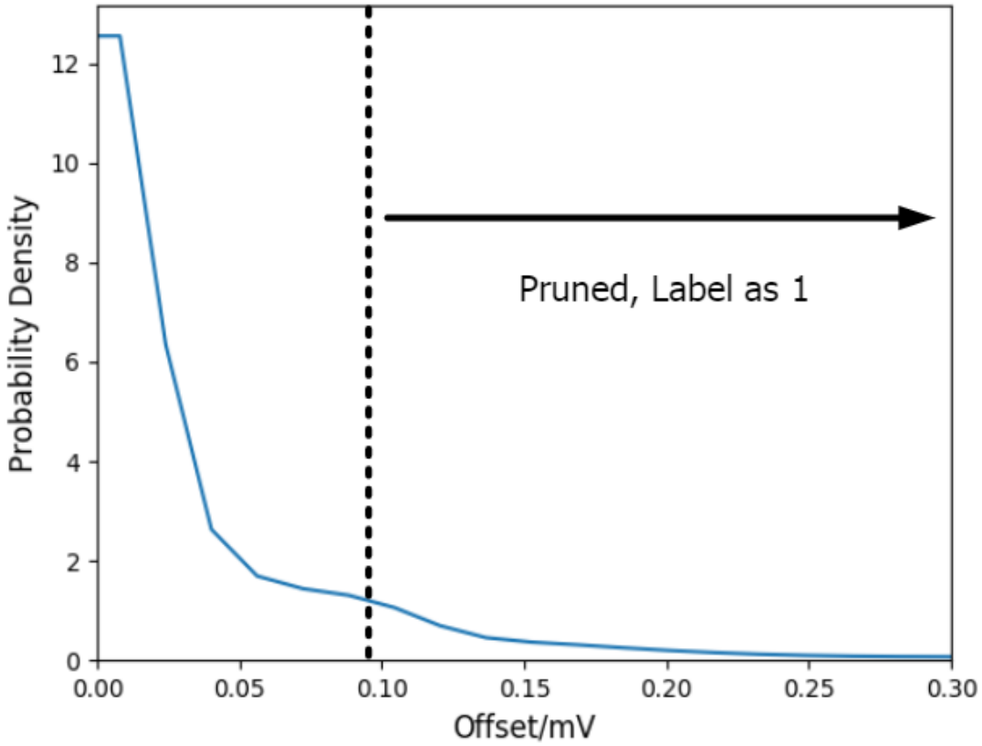
- Traditional automated analog layout generators
  - Human in the loop
  - Infer new heuristics and constraints
  - Poor generalizability and little flexibility



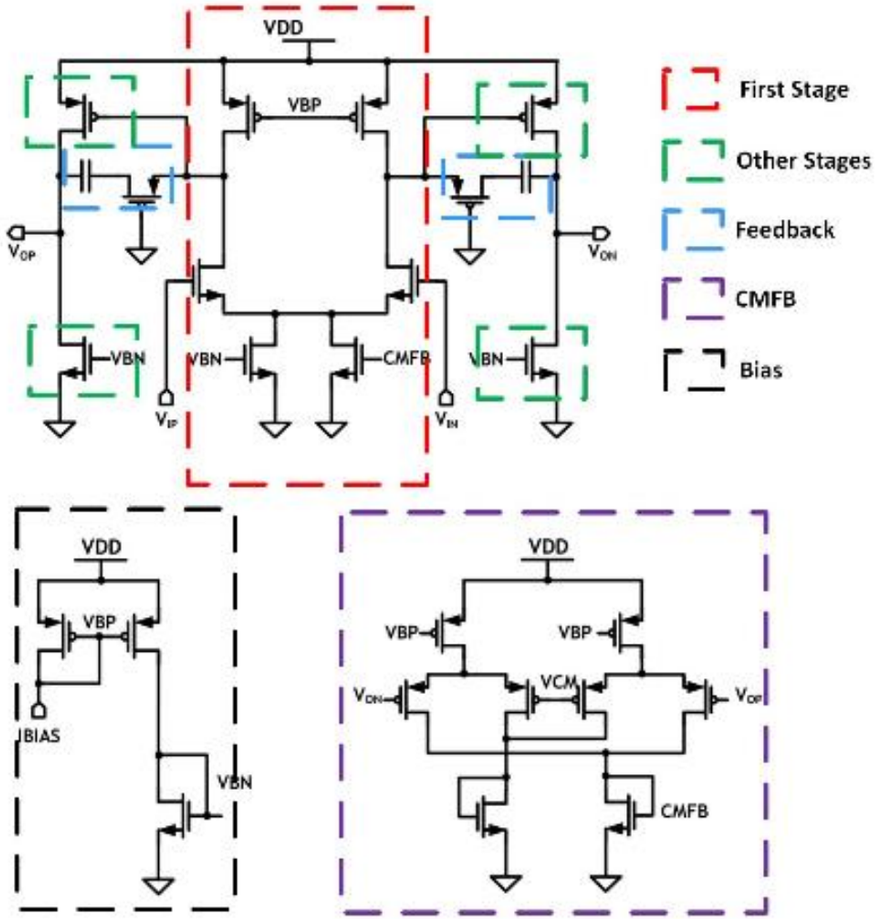
- Design exploration and early design pruning
  - Generating layout is “cheap”
  - Verifying functionality is expensive
  - Predict performance in early design stage

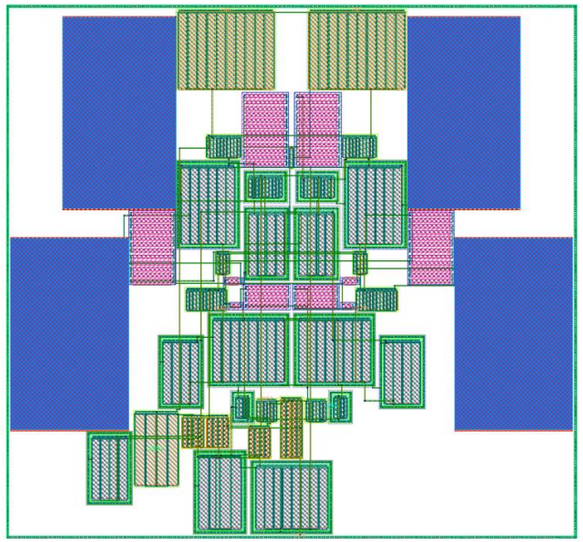


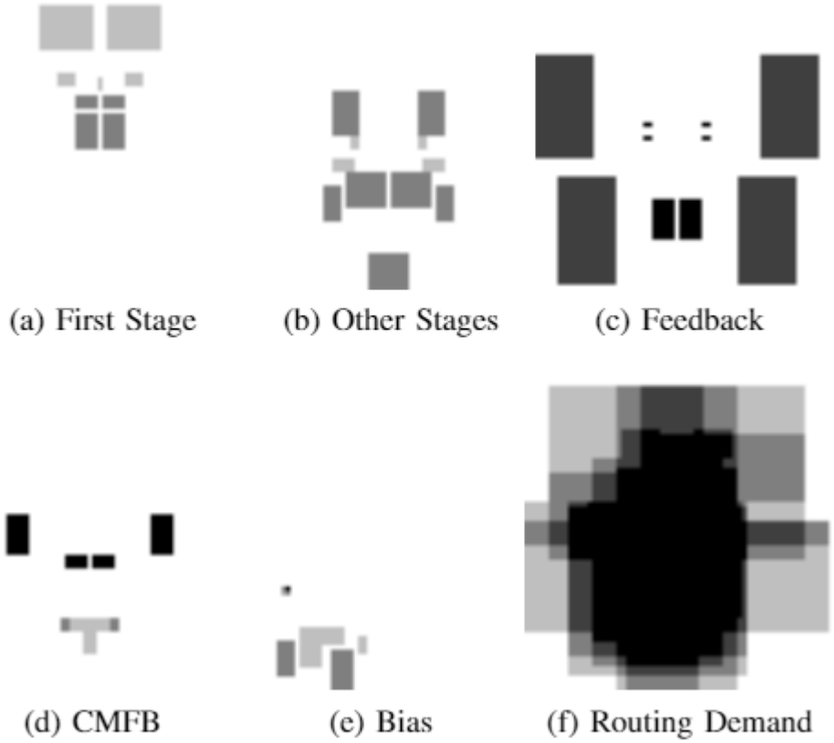
- Define layout quality with post layout simulation
  - Layout sensitive performance: CMRR and offset
- Formulate problem into binary classification
  - Balanced: Worst 25% vs Best 25%
  - Imbalanced: Worst 25% vs Rest 75%



- Placement feature extraction:
  - Device location and size: images
  - Circuit topology: separating devices to different channels
  - Device types: image intensity
  - Pin location: routing congestion map

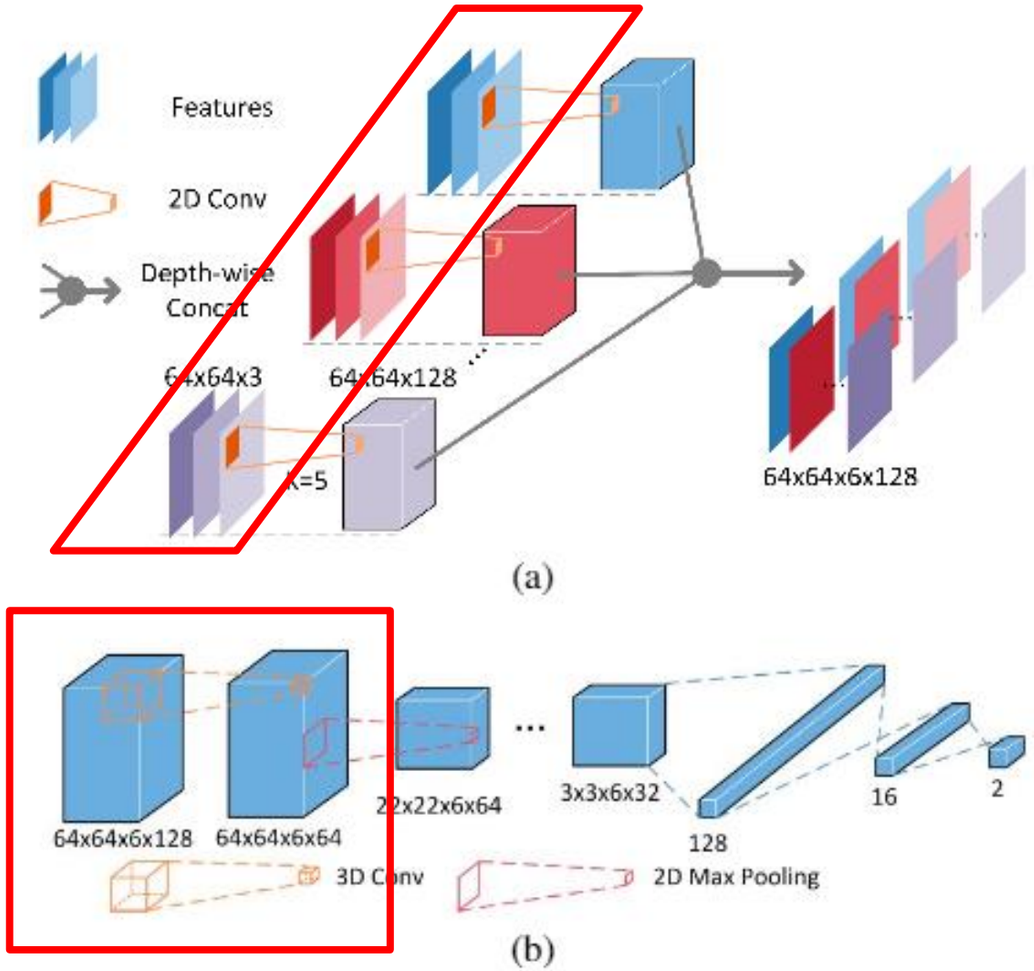






(a) First Stage (b) Other Stages (c) Feedback

(d) CMFB (e) Bias (f) Routing Demand



- Coordinate channel embedding
  - Additional channel for numerical coordinates

[R. Liu et al., NIPS, 2018]

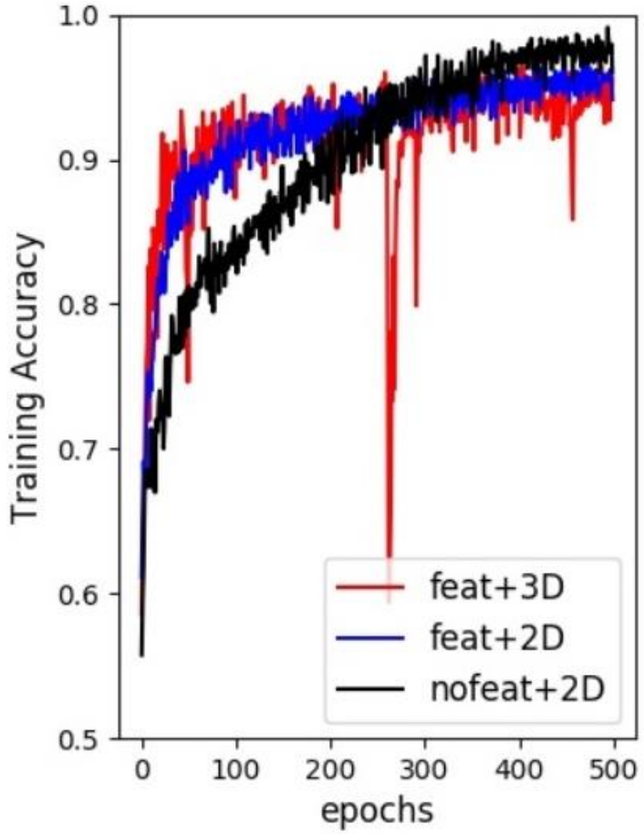
- 3D CNN
  - Depth-wise convolution
  - Interactions between different channels

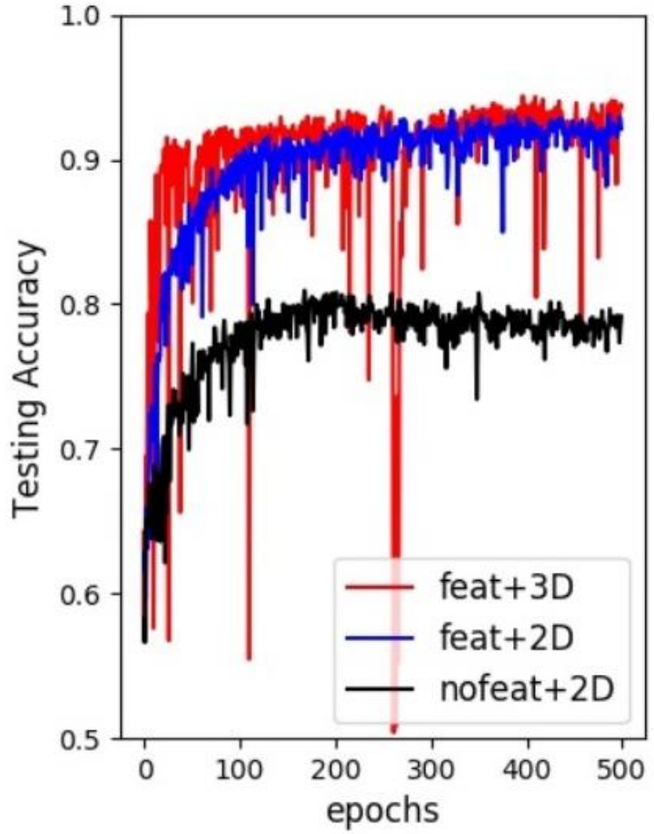
[S. Ji et al., IEEE T. Pattern Anal, 2013

- Dataset: OTA1 with balanced labeling
- Feature is of utmost importance
- 3D CNN helps a little with generalization

TABLE III: Baseline Model Comparisons

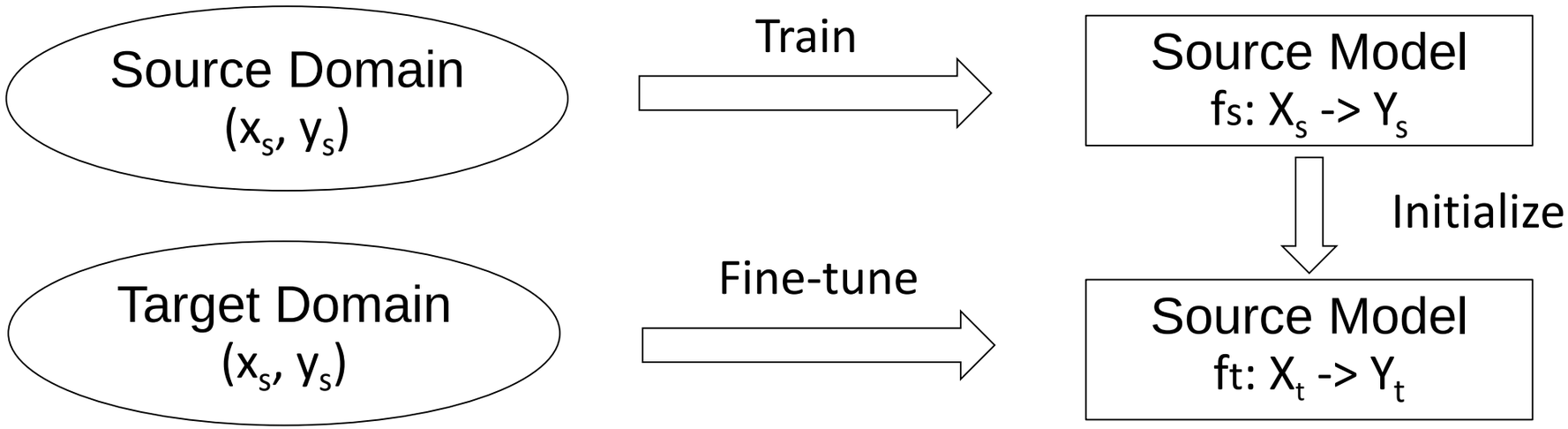
Model	Training Accuracy	Testing Accuracy
nofeat+2D	97.95%	78.44%
nofeat+3D	79.23%	78.32%
feat+2D	96.19%	91.94%
feat+3D	95.51%	**93.83%**





Outline

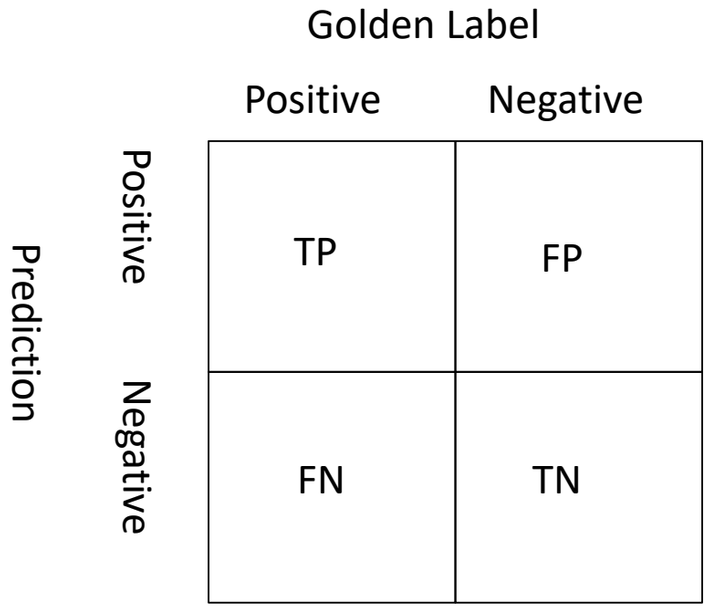
- Introduction and Motivation
- UT-AnLay Dataset with MAGICAL
- Placement Quality Prediction
- Improved Data Efficiency with Transfer Learning
- Conclusions



- Transfer learning:
  - Train model on source domain with abundant data
  - Fine-tune model on target domain with limited data

- Source Domain: OTA1
- Target Domains:
  - OTA2: Same schematic and performance metric different sizing
  - OTA3: Different schematic same performance metric
  - OTA4: Different schematic and performance metric
- Data utilization α:
  - Percentage of available training data in target domain
  - 20% reserved for testing $\alpha_{max} = 0.8$

- Accuracy
  - Acc = $\frac{TP+TN}{TP+FP+FN+TN}$
  - Measures overall performance
- False omission rate (FOR)
  - FOR = $\frac{FN}{TN+FN} = \frac{Predicted\ good\ with\ bad\ performance}{Total\ predicted\ good\ design}$
  - Percentage of leaked bad designs
  - Random selection: 25%



Detailed results on precision, recall and F1 score omitted for simplicity

- Transfer learning improves results compared with random initialization

Design	α	Accuracy (%)		FOR (%)	
		w	w/o	w	w/o
OTA1	0.80	90.29	–	8.32	–
OTA2	0.80	90.96	92.61	7.12	**4.99**
	0.10	90.10	80.40	**7.57**	22.00
	0.01	88.28	74.05	**9.60**	22.38
	0.00	70.10	–	21.28	–
OTA3	0.80	90.23	91.33	8.38	**6.26**
	0.10	87.29	79.98	**9.75**	23.84
	0.01	81.21	77.32	**13.40**	26.76
	0.00	74.73	–	18.72	–
OTA4	0.80	89.81	92.05	6.06	**4.09**
	0.10	88.70	74.33	**9.54**	22.90
	0.01	81.05	76.68	**16.95**	21.92
	0.00	49.72	–	22.77	–

w are transfer learning results, w/o are retraining with random initialization

- Transfer learning improves results compared with random initialization
- Results improves with more training data in target domain

Design	α	Accuracy (%)		FOR (%)		
		w	w/o	w		w/o
OTA1	0.80	90.29	–	8.32		–
OTA2	0.80	90.96	92.61	7.12	↑	**4.99**
	0.10	90.10	80.40	**7.57**		22.00
	0.01	88.28	74.05	**9.60**		22.38
	0.00	70.10	–	21.28		–
OTA3	0.80	90.23	91.33	8.38	↑	**6.26**
	0.10	87.29	79.98	**9.75**		23.84
	0.01	81.21	77.32	**13.40**		26.76
	0.00	74.73	–	18.72		–
OTA4	0.80	89.81	92.05	6.06	↑	**4.09**
	0.10	88.70	74.33	**9.54**		22.90
	0.01	81.05	76.68	**16.95**		21.92
	0.00	49.72	–	22.77		–

w are transfer learning results, w/o are retraining with random initialization

- Transfer learning improves results compared with random initialization
- Results improves with more training data in target domain
- Satisfactory results in transfer learning with only 1% data (160 layouts)

Design	α	Accuracy (%)		FOR (%)	
		w	w/o	w	w/o
OTA1	0.80	90.29	–	8.32	–
OTA2	0.80	90.96	92.61	7.12	**4.99**
	0.10	90.10	80.40	**7.57**	22.00
	0.01	88.28	74.05	**9.60**	22.38
	0.00	70.10	–	21.28	–
OTA3	0.80	90.23	91.33	8.38	**6.26**
	0.10	87.29	79.98	**9.75**	23.84
	0.01	81.21	77.32	**13.40**	26.76
	0.00	74.73	–	18.72	–
OTA4	0.80	89.81	92.05	6.06	**4.09**
	0.10	88.70	74.33	**9.54**	22.90
	0.01	81.05	76.68	**16.95**	21.92
	0.00	49.72	–	22.77	–

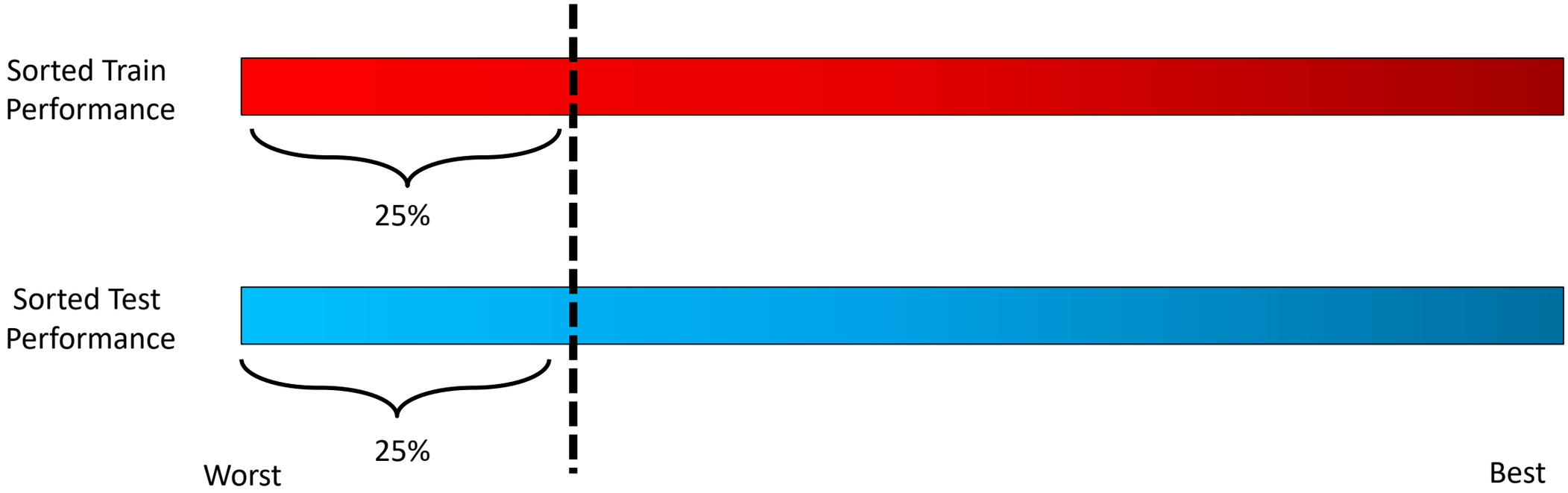
w are transfer learning results, w/o are retraining with random initialization

- Transfer learning improves results compared with random initialization
- Results improves with more training data in target domain
- Satisfactory results in transfer learning with only 1% data (160 layouts)
- Applying baseline model without transfer learning produce bad results

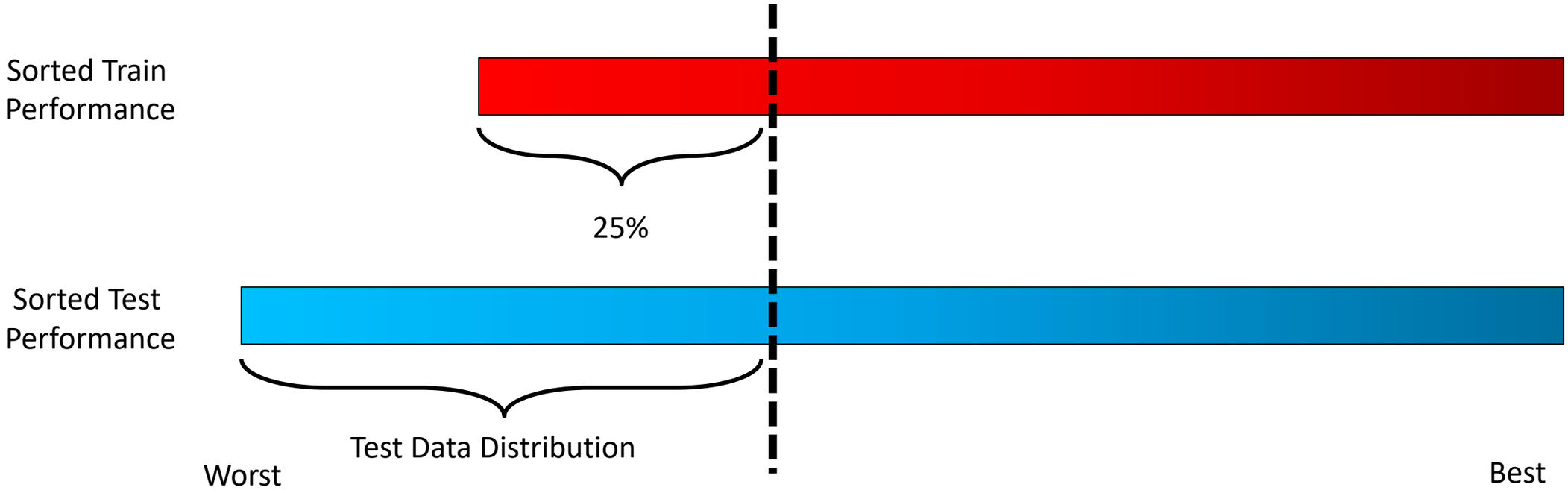
Design	α	Accuracy (%)		FOR (%)	
		w	w/o	w	w/o
OTA1	0.80	90.29	–	8.32	–
OTA2	0.80	90.96	92.61	7.12	**4.99**
	0.10	90.10	80.40	**7.57**	22.00
	0.01	88.28	74.05	**9.60**	22.38
	0.00	70.10	–	21.28	–
OTA3	0.80	90.23	91.33	8.38	**6.26**
	0.10	87.29	79.98	**9.75**	23.84
	0.01	81.21	77.32	**13.40**	26.76
	0.00	74.73	–	18.72	–
OTA4	0.80	89.81	92.05	6.06	**4.09**
	0.10	88.70	74.33	**9.54**	22.90
	0.01	81.05	76.68	**16.95**	21.92
	0.00	49.72	–	22.77	–

w are transfer learning results, w/o are retraining with random initialization

- What is the limit? Can 0.1% data work (16 layouts)?
- Testing data 20%, 200 times of training
- Issues: Data distributions in train/test varies significantly

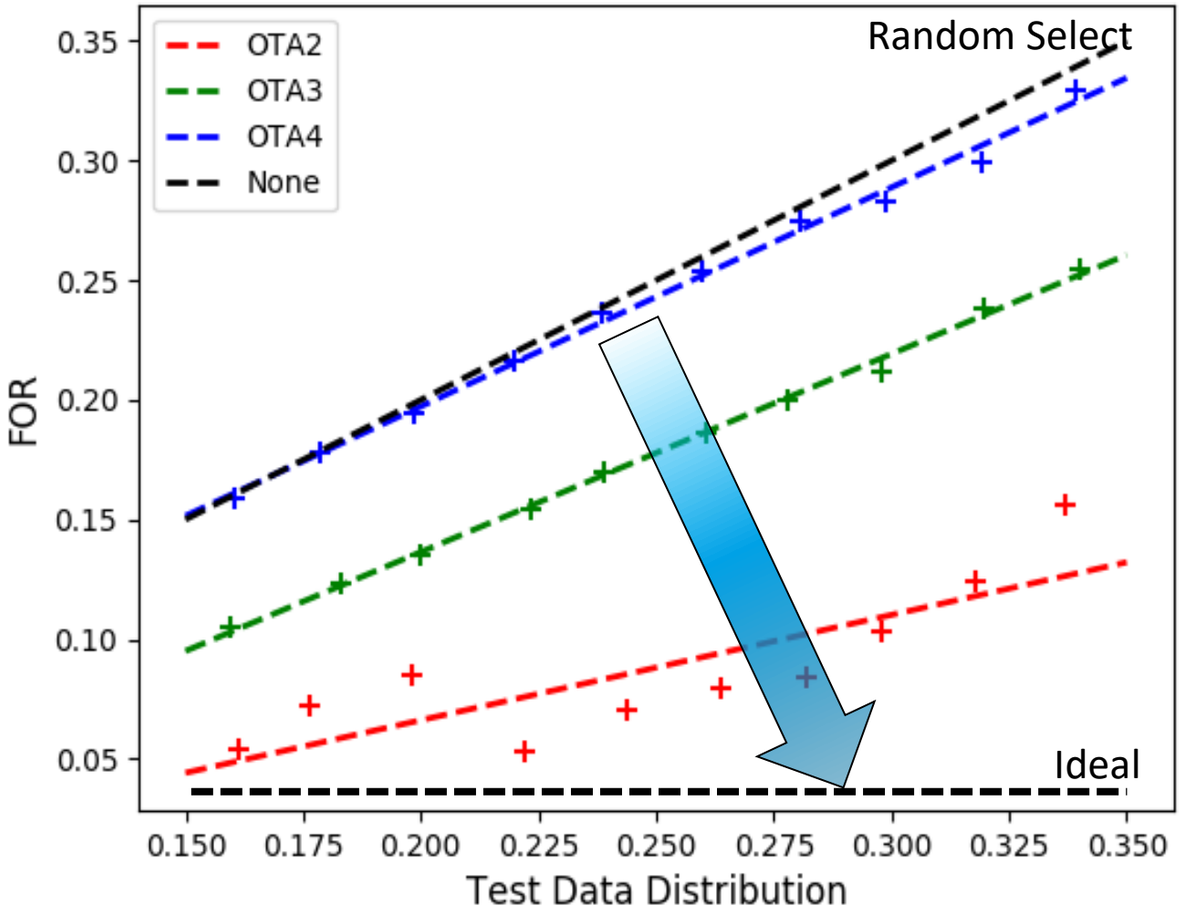


- What is the limit? Can 0.1% data work (16 layouts)?
- Testing data 20%, 200 times of training
- Issues: Data distributions in train/test varies significantly



- Random select 16 layouts (0.1%) as transfer learning data
- Label training data according to relative rank in training set
- Relabel testing set according to the training set
- Repeat experiment for 100 times for each transfer target

- Improvement gained from transfer learning is related with task difficulty
- OTA2: Same design and performance metric different sizing
- OTA3: Different design and same performance metric
- OTA4: Different design and different performance metric



Outline

- Introduction and Motivation
- UT-AnLay Dataset with MAGICAL
- Placement Quality Prediction
- Improved Data Efficiency with Transfer Learning
- Conclusions

Conclusions

- UT-AnLay:
  - A dataset for post layout performance modeling
  - Include placement solution and post layout simulation results
- Our preliminary work:
  - Placement quality prediction
  - Improved data efficiently with transfer learning
- Open-sourced data and model:
  - https://github.com/magical-eda/UT-AnLay
- Open-sourced MAGICAL layout generator:
  - https://github.com/magical-eda/MAGICAL

Thank You