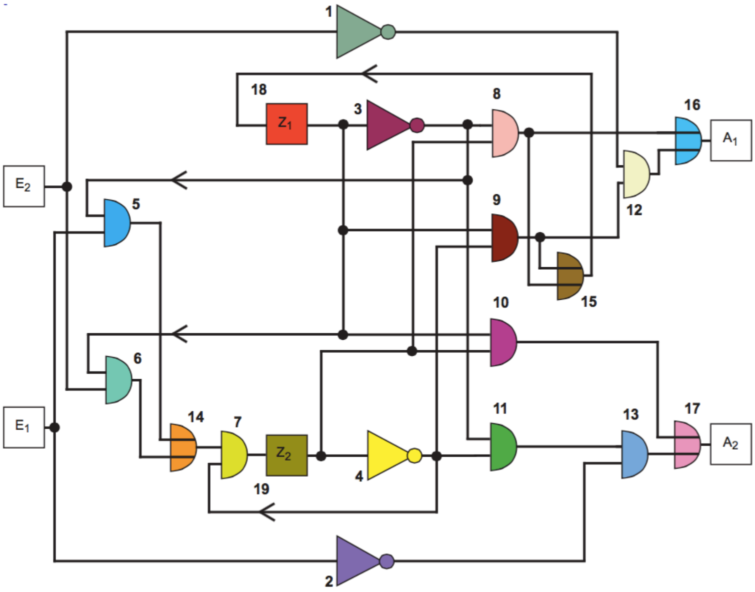
香港中文大學
The Chinese University of Hong Kong

Machine Learning in EDA

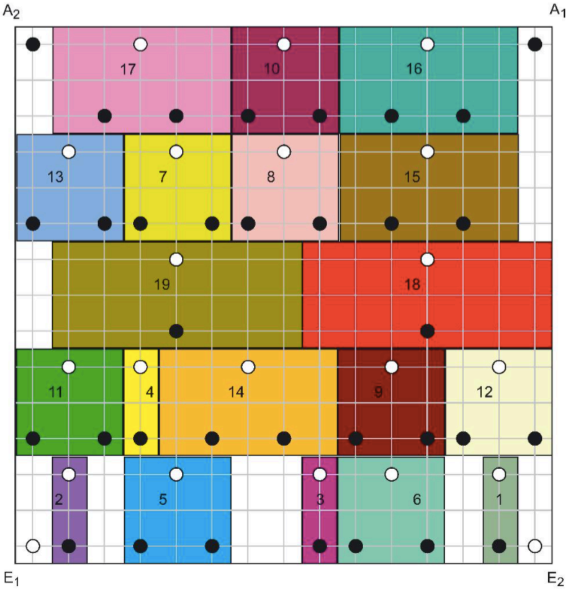
Bei Yu
Department of Computer Science & Engineering
Chinese University of Hong Kong
byu@cse.cuhk.edu.hk

September 30, 2021

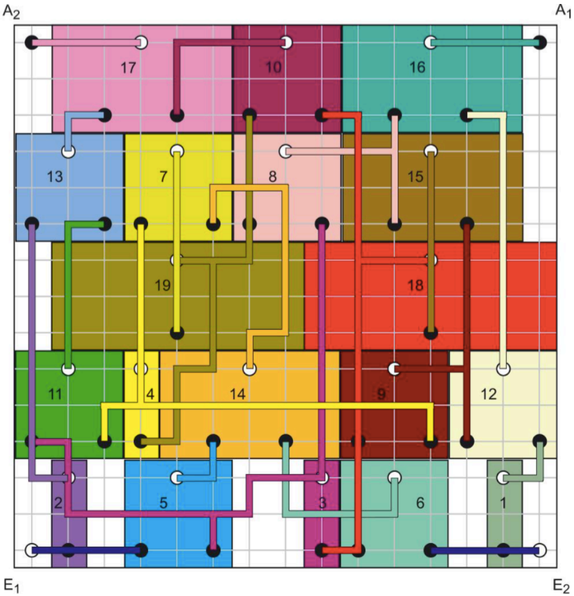
EDA Toy Example [Jens Vygen,2006]



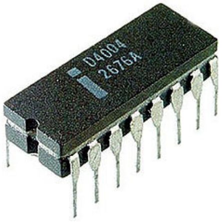
EDA Toy Example [Jens Vygen,2006]



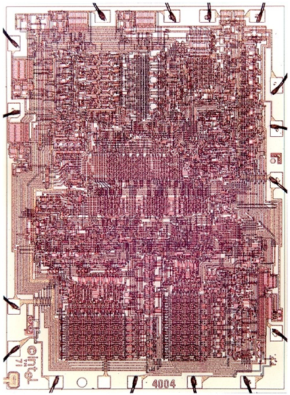
EDA Toy Example [Jens Vygen,2006]



When was the first Microprocessor?

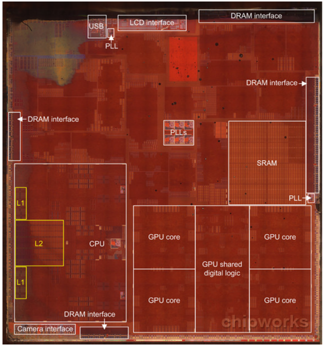


(a)

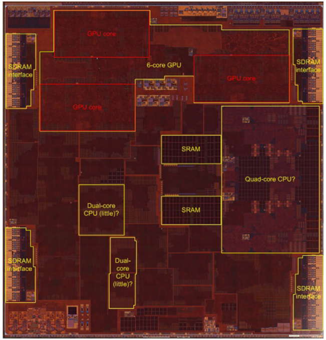


(b)

1971, Intel 4004.



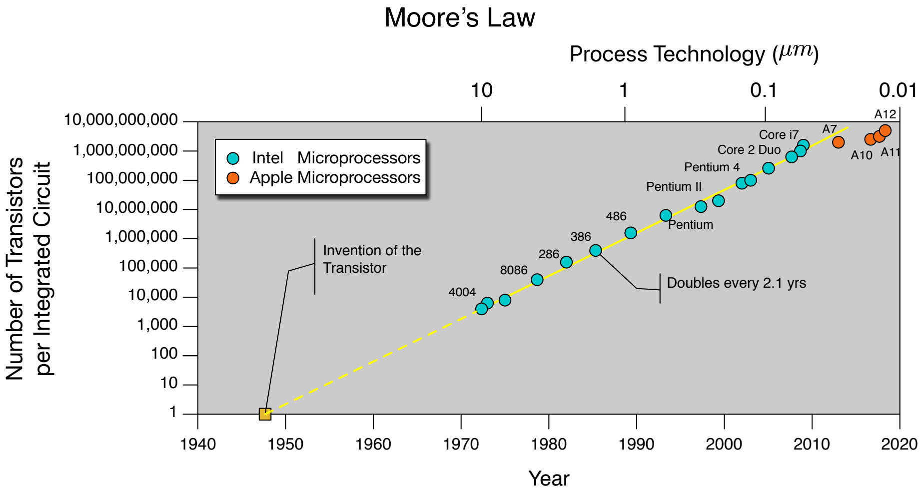
Apple A7 (2013)

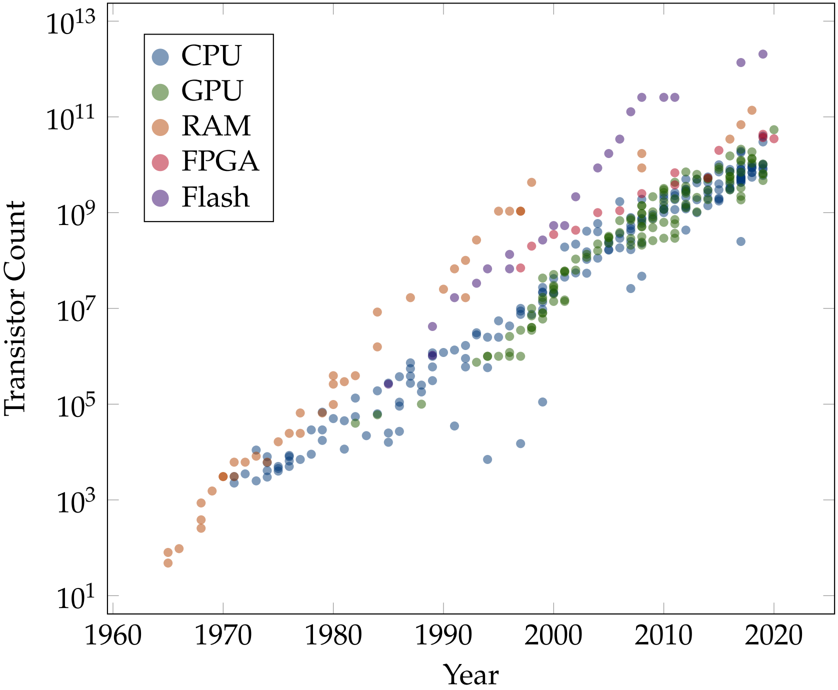


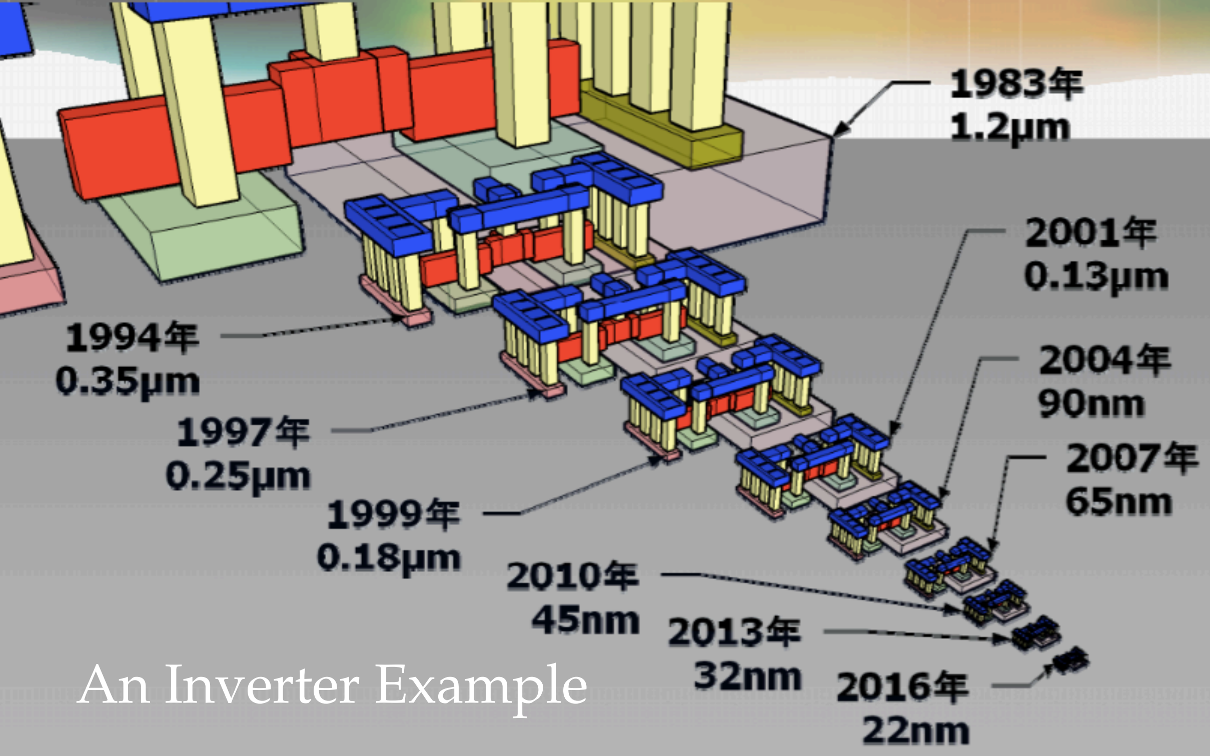
Apple A10 (2016)

- 1,000,000,000 Transistors
- $102mm^2$ die size
- 1.3GHz

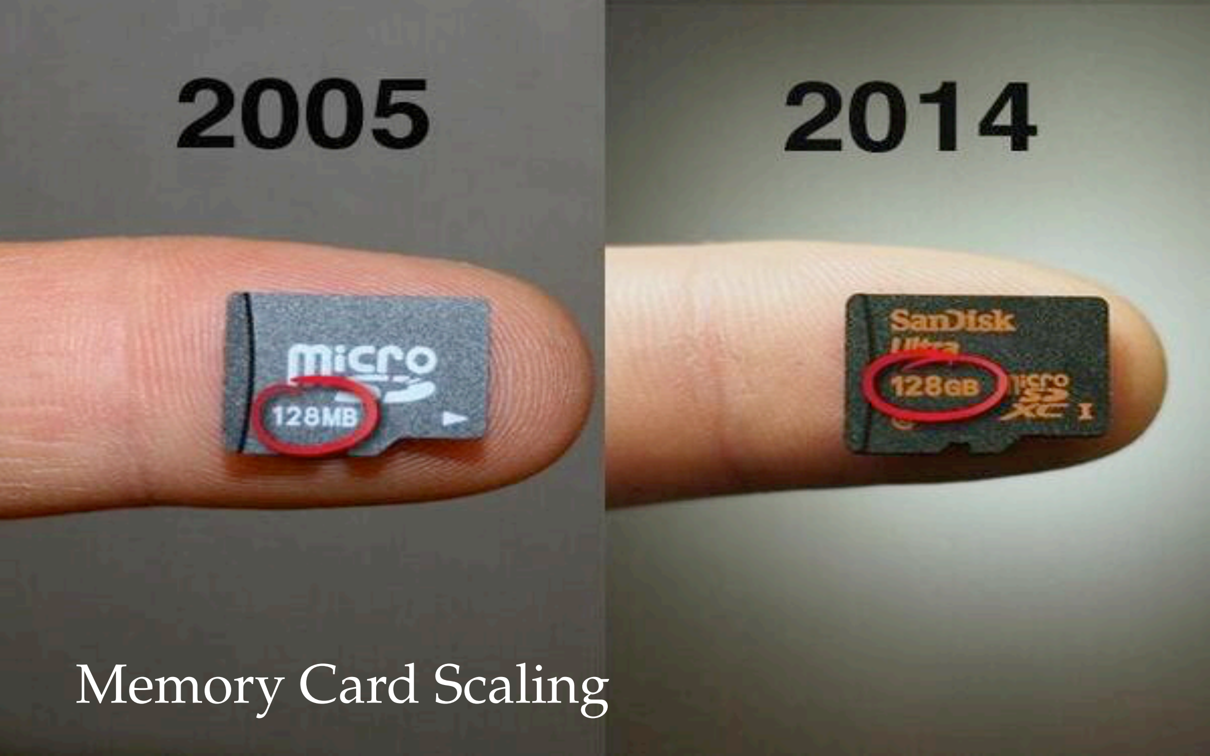
- 3,300,000,000 Transistors
- $125mm^2$ die size
- 2.34GHz


Moore’s Law
Process Technology (μm)
10
1
0.1
0.01
10,000,000,000
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
1,000
100
10
1
Number of Transistors per Integrated Circuit
Intel Microprocessors
Apple Microprocessors
Invention of the Transistor
4004
8086
286
386
486
Pentium
Pentium II
Pentium 4
Core 2 Duo
Core i7
A7
A10
A11
A12
Doubles every 2.1 yrs
1940
1950
1960
1970
1980
1990
2000
2010
2020
Year


CPU
GPU
RAM
FPGA
Flash
Transistor Count
10^{13}
10^{11}
10^{9}
10^{7}
10^{5}
10^{3}
10^{1}
1960
1970
1980
1990
2000
2010
2020
Year

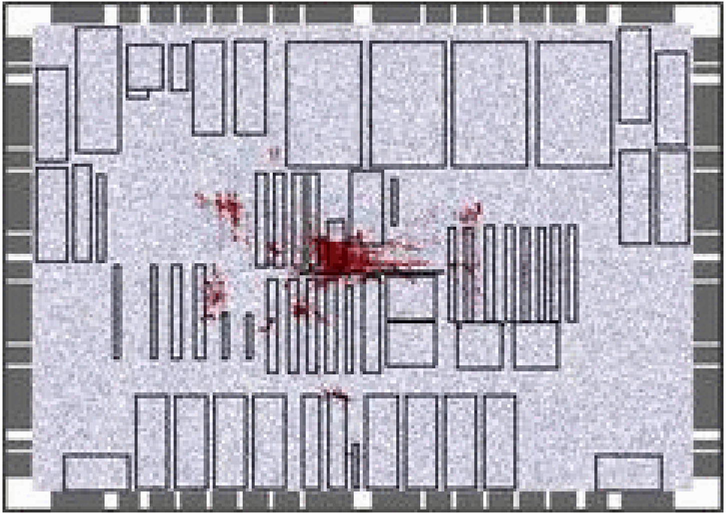


An Inverter Example



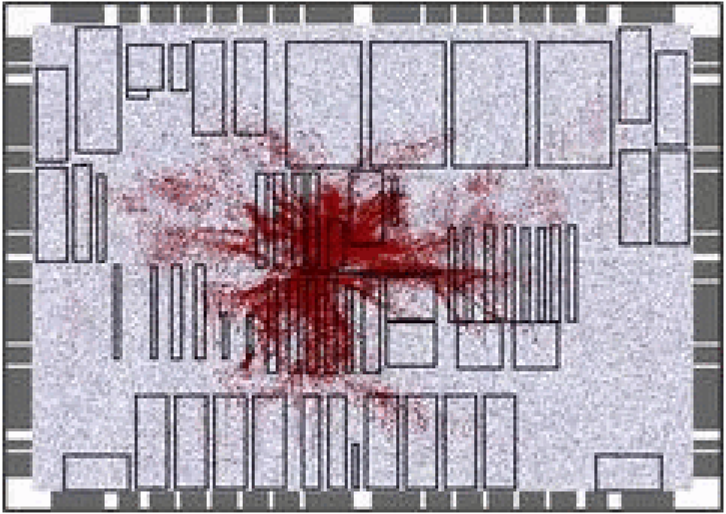
Memory Card Scaling

Challenge: Larger and Larger Scale



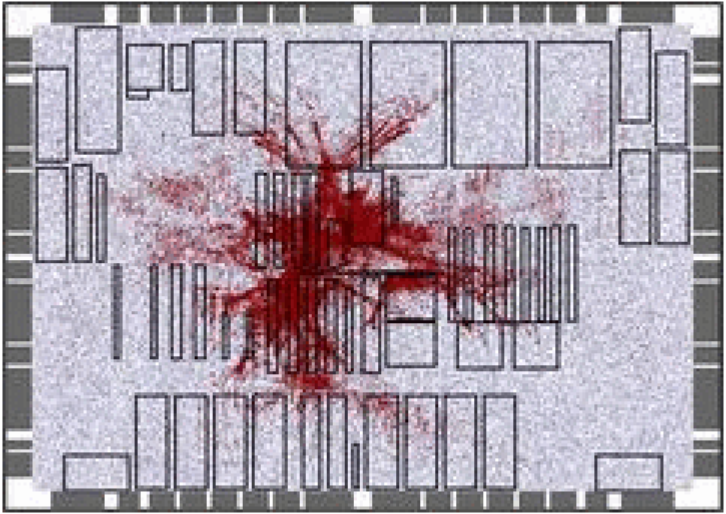
Placement [Lu+,DAC'14]: 221K nets, 63 fixed macros and 210K movable cells.

Challenge: Larger and Larger Scale



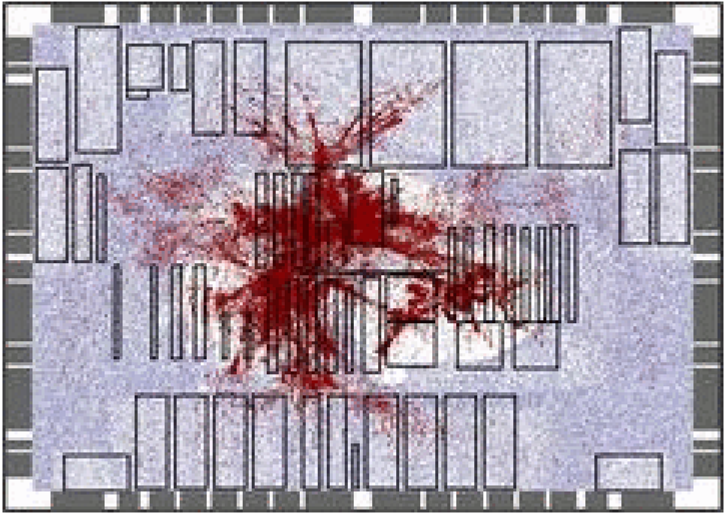
Placement [Lu+,DAC'14]: 221K nets, 63 fixed macros and 210K movable cells.

Challenge: Larger and Larger Scale

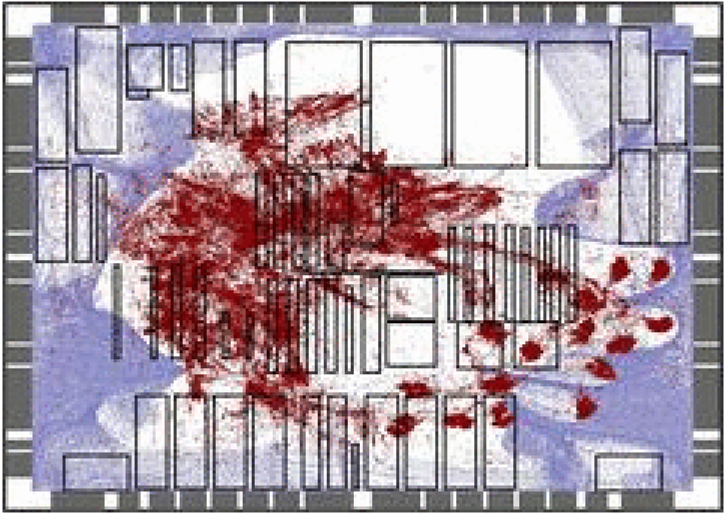


Placement [Lu+,DAC'14]: 221K nets, 63 fixed macros and 210K movable cells.

Challenge: Larger and Larger Scale

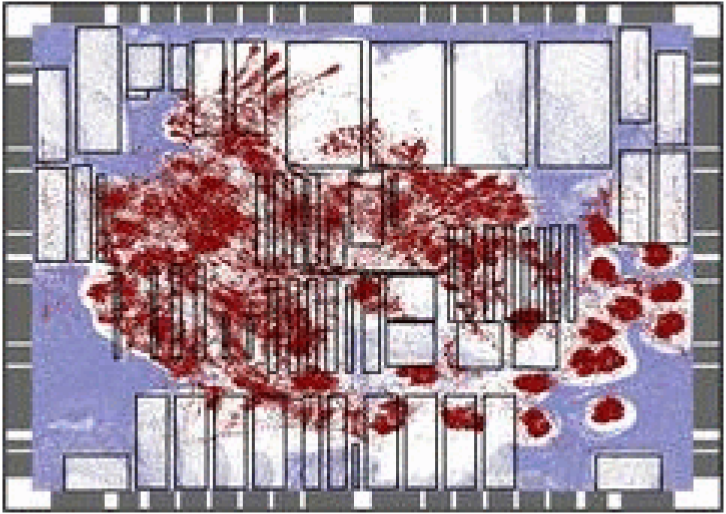


Placement [Lu+,DAC'14]: 221K nets, 63 fixed macros and 210K movable cells.

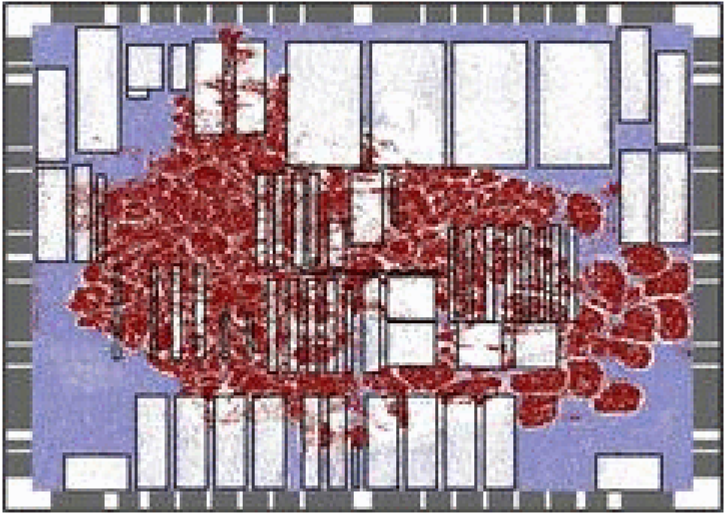


Placement [Lu+,DAC'14]: 221K nets, 63 fixed macros and 210K movable cells.

Challenge: Larger and Larger Scale

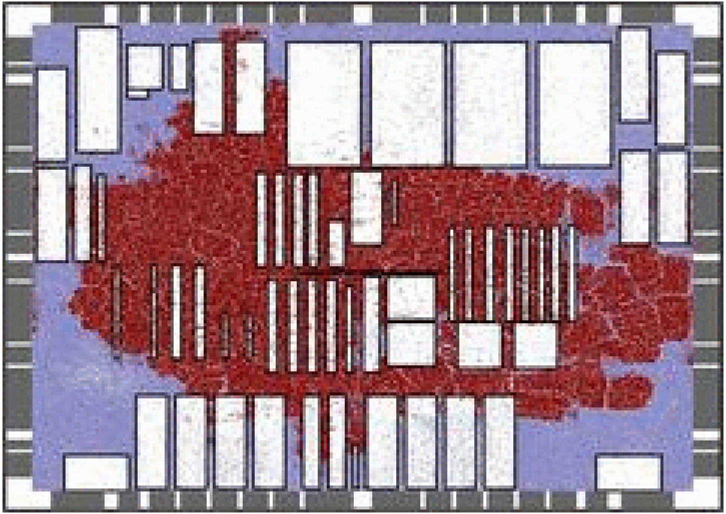


Placement [Lu+,DAC'14]: 221K nets, 63 fixed macros and 210K movable cells.



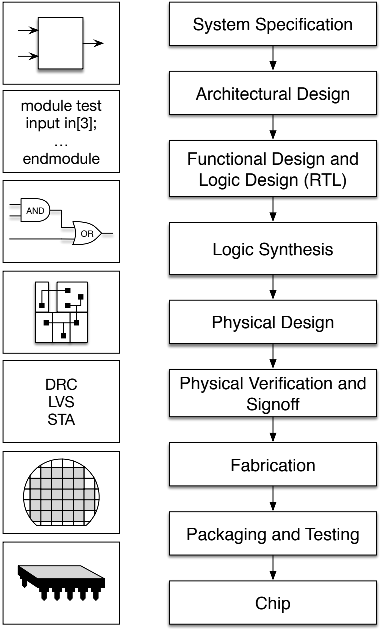
Placement [Lu+,DAC'14]: 221K nets, 63 fixed macros and 210K movable cells.

Challenge: Larger and Larger Scale

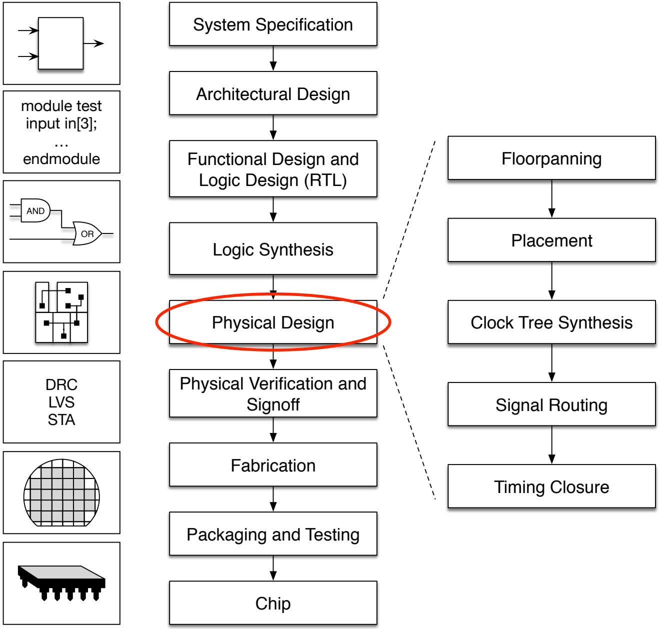


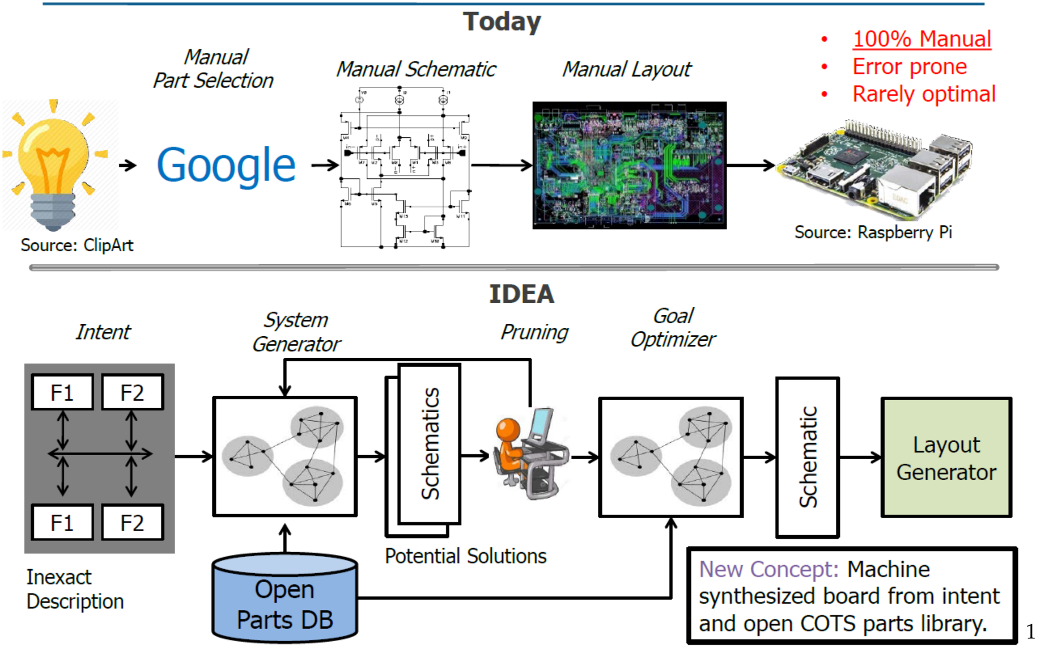
Placement [Lu+,DAC'14]: 221K nets, 63 fixed macros and 210K movable cells.

Challenge: Complicated Design Flow



Challenge: Complicated Design Flow





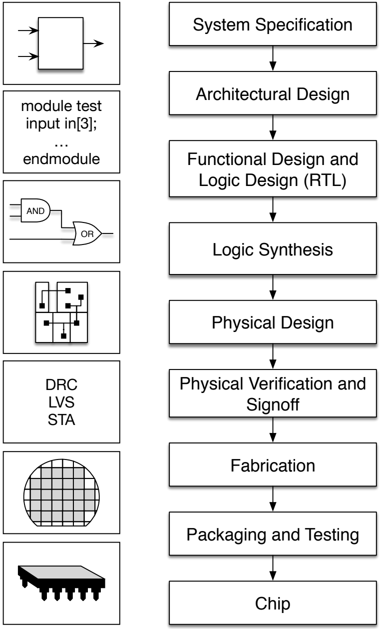
[1] https://www.darpa.mil/program/intelligent-design-of-electronic-assets

Outline

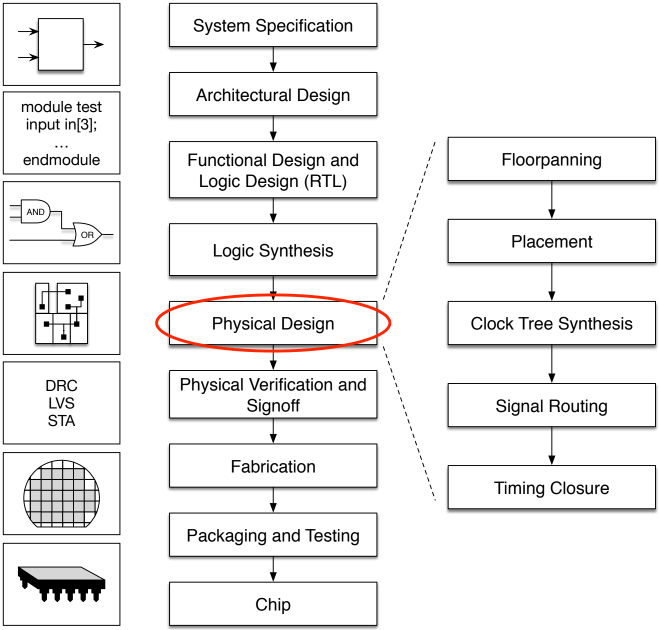
2. Integrating Active Learning
3. Integrating Deep Learning
4. Integrating Deep Learning Engine
5. On Irregular Structure Learning: Graph Learning

Active Learning

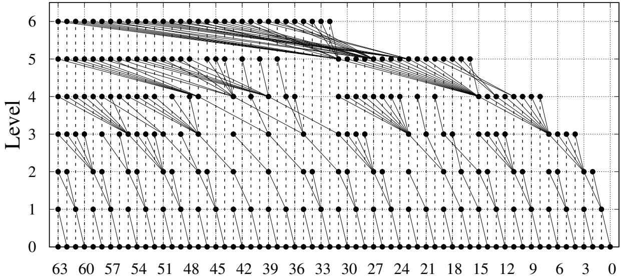
Challenge: Complicated Design Flow

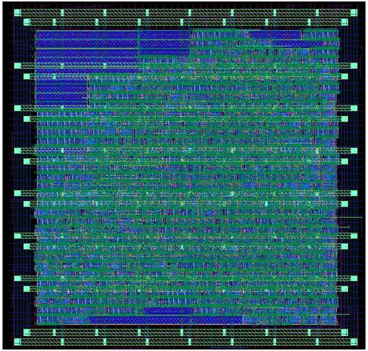


Challenge: Complicated Design Flow



Gaps Between Design Stages



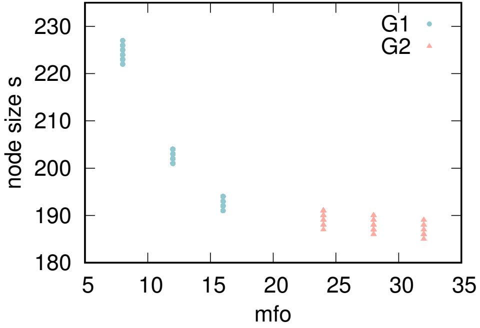


Case Study: Adder Design

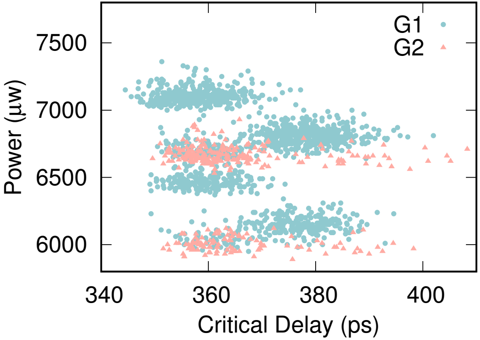
- Logic synthesis v.s. physical synthesis
- Constraints mapping between two synthesis stages is difficult.

Why Smart Sampling ?

Front-End Team Perspective:



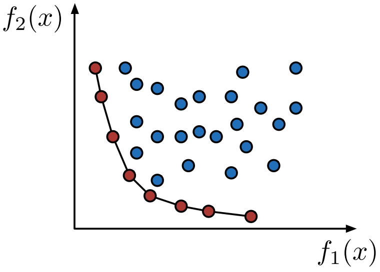
Back-End Team Perspective:

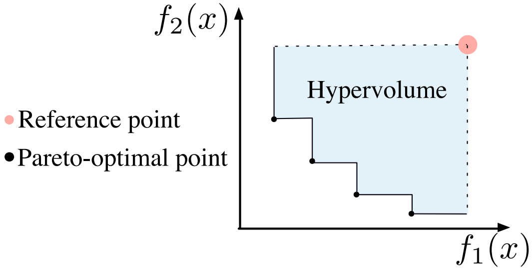


- Run design tools with all solutions is time-consuming.
- For 3K solutions, running time is $3000 \times 5 = 15$K mins.
- What we care: Pareto Frontier Curve

Pareto Frontier

- All the points are not dominated by any other point.
- Evaluation: Hyper-volume.
  - Size of the region bounded by the Pareto frontier and reference point.
  - Each dimension of reference point is the maximum value on that dimension.

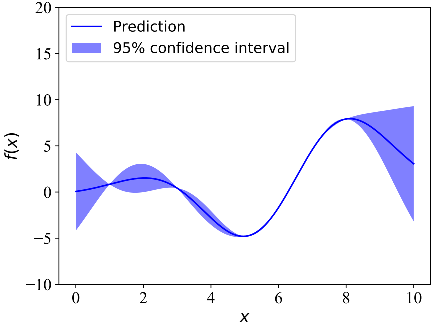




Active Learning Flow [TCAD'19][1]

Regression

- Gaussian process model;
- A prediction consists of a mean and a variance;
- Off-the-shelf library for implementation.



[1] Y. Ma, S. Roy, J. Miao, J. Chen and B. Yu, "Cross-Layer Optimization for High Speed Adders: A Pareto Driven Machine Learning Approach", TCAD'19.

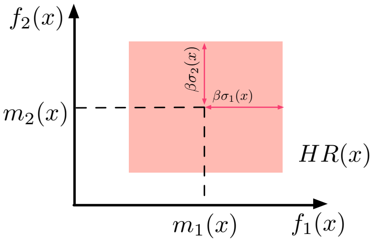
Uncertainty

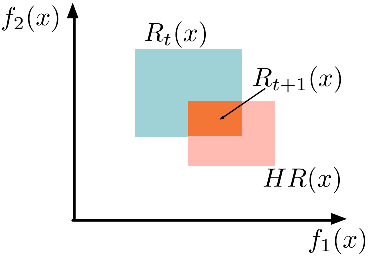
- Given the prediction (m, σ), a hyper-rectangle is defined as

$$HR(x) = \{\mathbf{y} : m_i(x) - \beta\sigma_i(x) \le y_i \le m_i(x) + \beta\sigma_i(x)\}$$

- The uncertainty region is defined as:

$$R_{t+1}(x) = R_t(x) \cap HR(x)$$

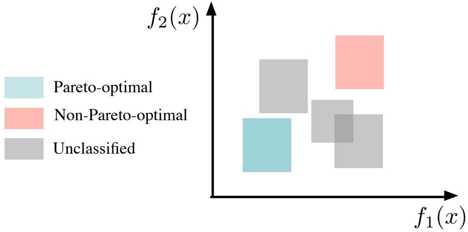


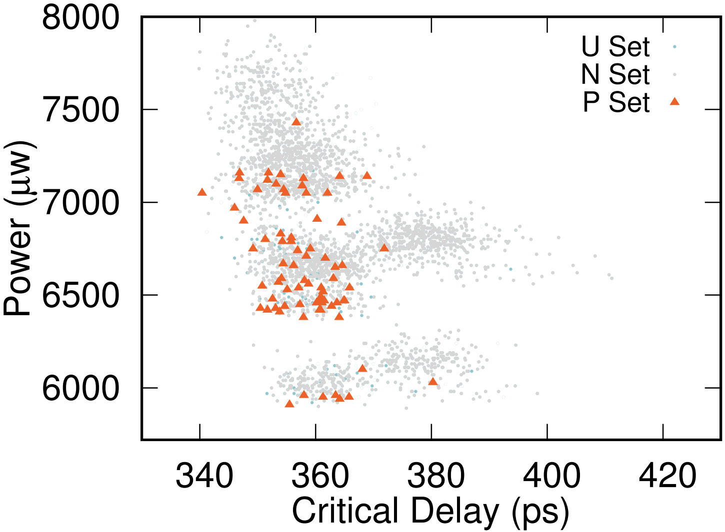


Active Learning Flow [TCAD'19]

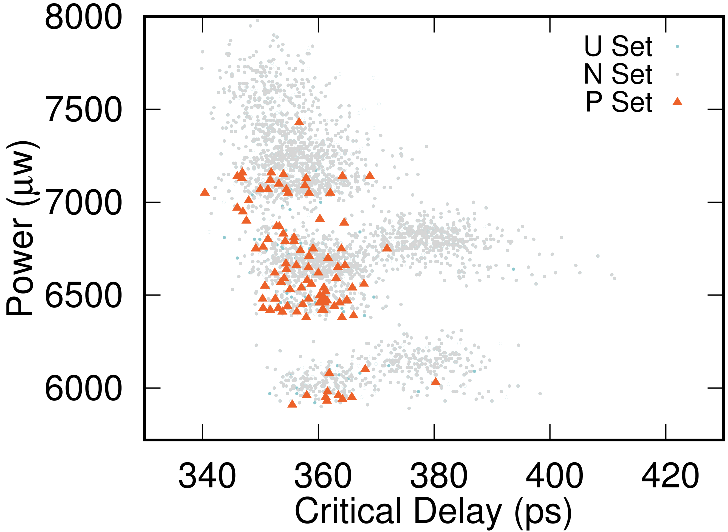
Classification

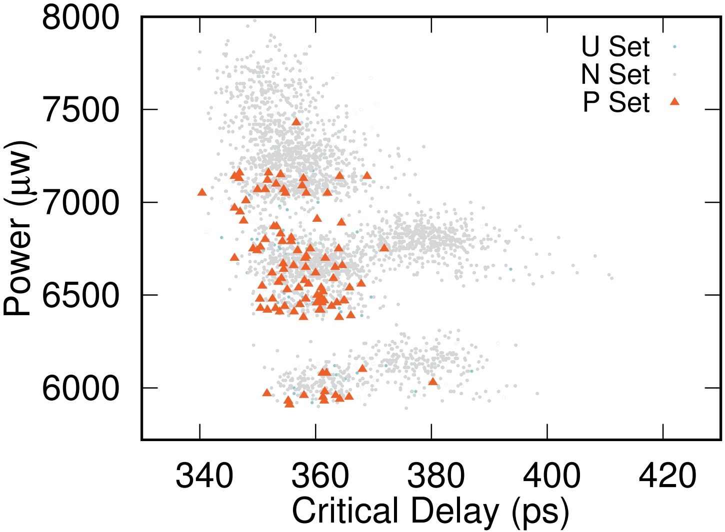
$$x \in \begin{cases} P, & \text{if} \quad \max(R_t(x)) \le \min(R_t(x')) + \delta, \\ N, & \text{if} \quad \max(R_t(x')) \le \min(R_t(x)) + \delta, \\ U, & \text{otherwise.} \end{cases}$$



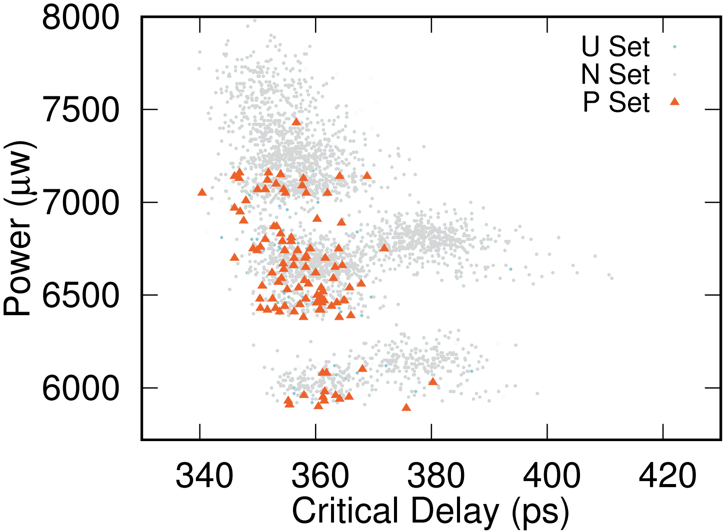


8000
7500
7000
6500
6000
Power (µw)
340
360
380
400
420
Critical Delay (ps)
U Set
N Set
P Set

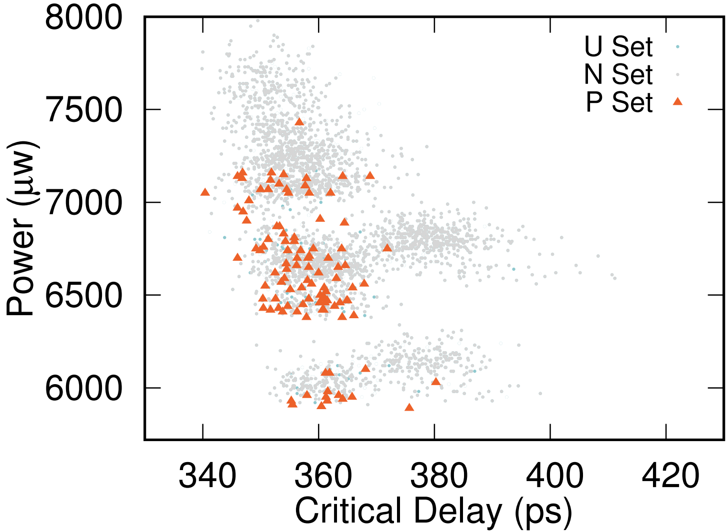

8000
7500
7000
6500
6000
Power (µw)
340
360
380
400
420
Critical Delay (ps)
U Set
N Set
P Set

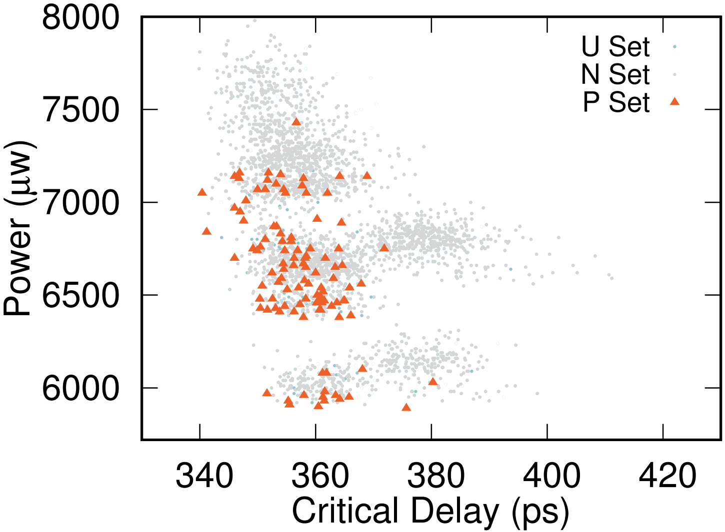


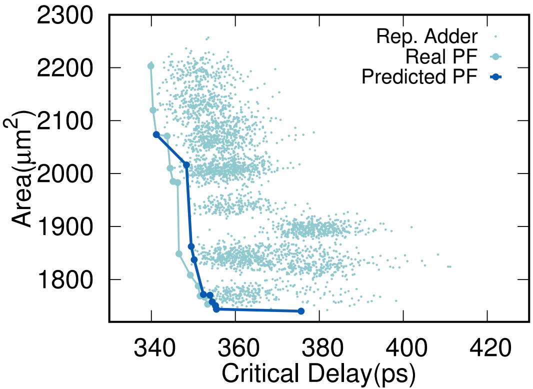
U Set
N Set
P Set
8000
7500
7000
6500
6000
Power (µw)
340
360
380
400
420
Critical Delay (ps)

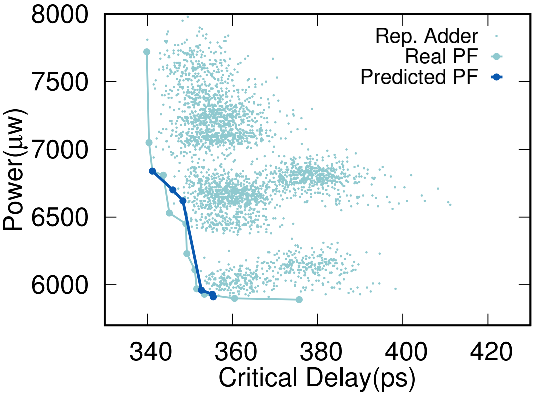


8000
7500
7000
6500
6000
Power (µw)
340
360
380
400
420
Critical Delay (ps)
U Set
N Set
P Set

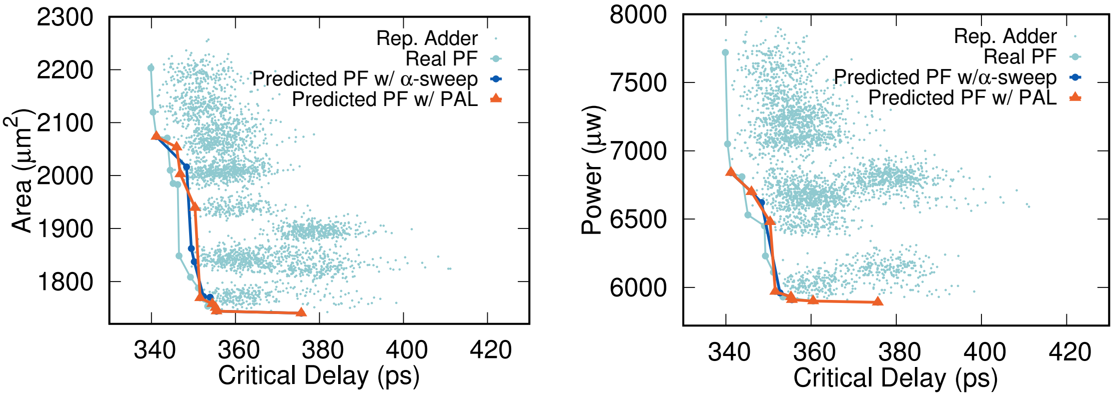

8000
7500
7000
6500
6000
Power (µw)
340
360
380
400
420
Critical Delay (ps)
U Set
N Set
P Set


U Set
N Set
P Set
Power (µw)
8000
7500
7000
6500
6000
340
360
380
400
420
Critical Delay (ps)

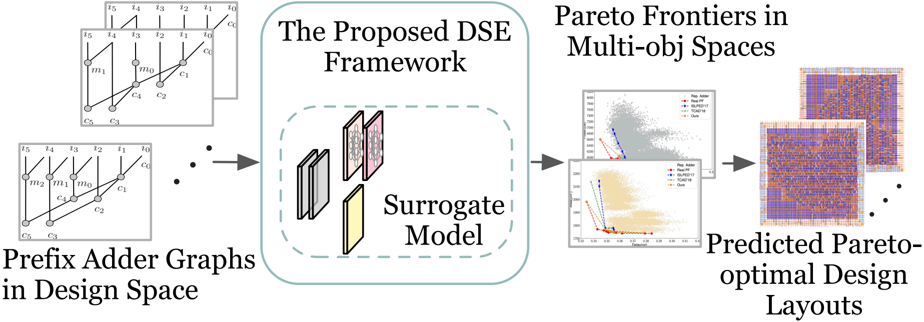

Rep. Adder
Real PF
Predicted PF
Area(µm^2)
2300
2200
2100
2000
1900
1800
340
360
380
400
420
Critical Delay(ps)


Rep. Adder
Real PF
Predicted PF
Power(µw)
8000
7500
7000
6500
6000
340
360
380
400
420
Critical Delay(ps)

Pareto-Frontier Results



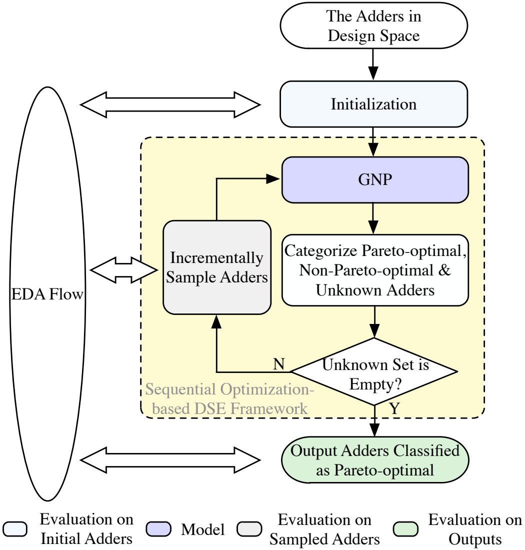
Learning Better Features for Prefix Adder Structure [TCAD'21][2]



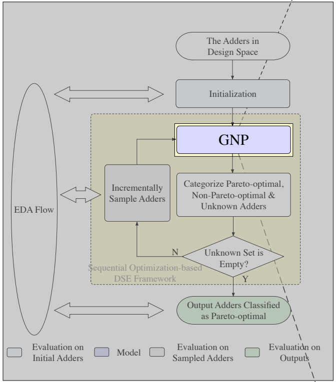
The adder design space exploration.

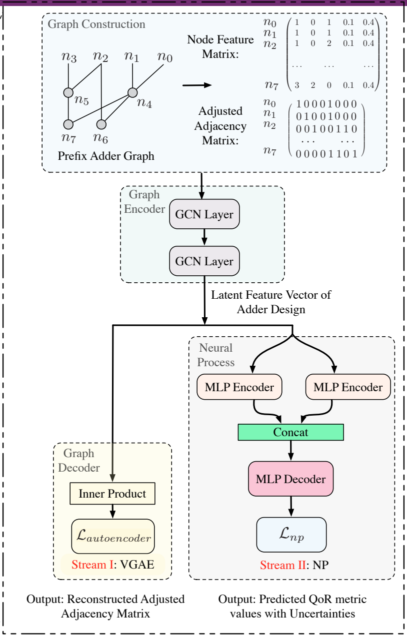
[2] H. Geng, Y. Ma, Q. Xu, J. Miao, S. Roy and B. Yu, "High-Speed Adder Design Space Exploration via Graph Neural Processes," TCAD'21.

Our solution: Overview (I) [TCAD'21]



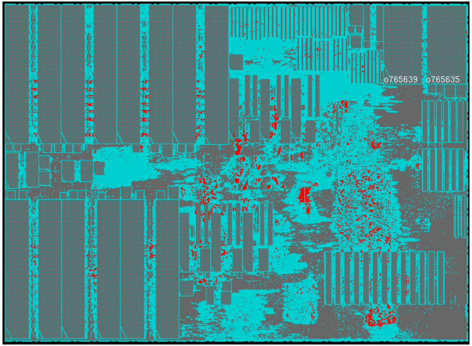
Our solution: Overview (II) [TCAD'21]





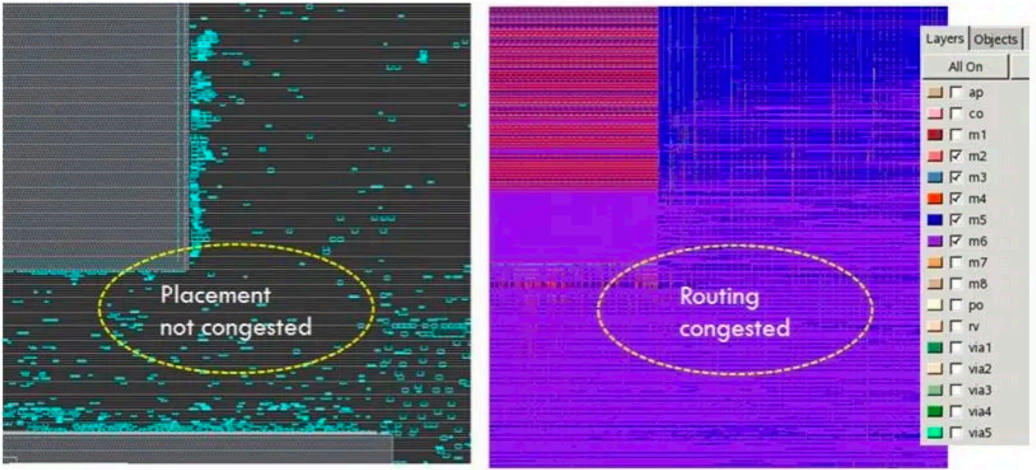
Integrating Deep Learning

Challenge: Larger and Larger Scale

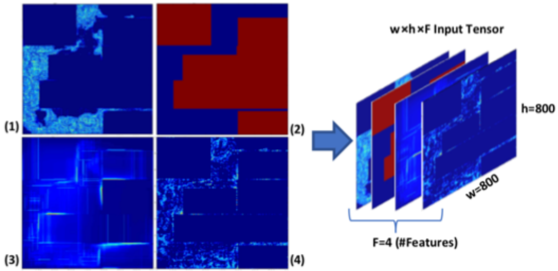


- Back-end is time consuming
- Accurate connectivity should be predictable
- Better estimation means efficient design-to-market budget

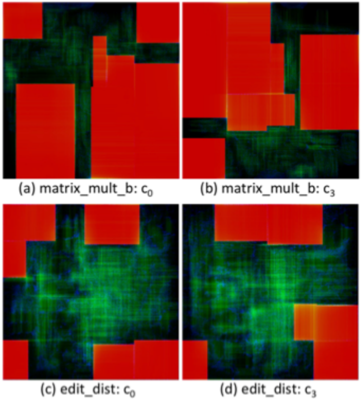
Complicated Relationship in Placement & Route



Features Extraction



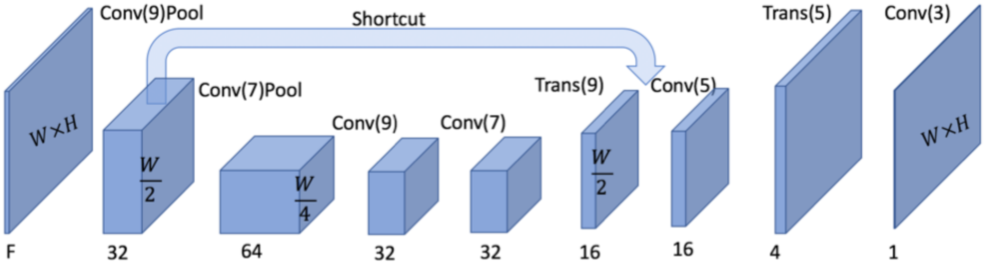
Input tensor constructed by stacking 2D features:
(1) Pin density, (2) macro (3) long-range RUDY, (4) RUDY pins



Input features for #DRV prediction.
Red: macro region
Green: global long-range RUDY
Blue: global RUDY pins

[3] Xie+, "RouteNet: Routability prediction for Mixed-Size Designs Using Convolutional Neural Network", ICCAD'18.

Proposed Model - Hotspot Detection



Filter size indicated in ()

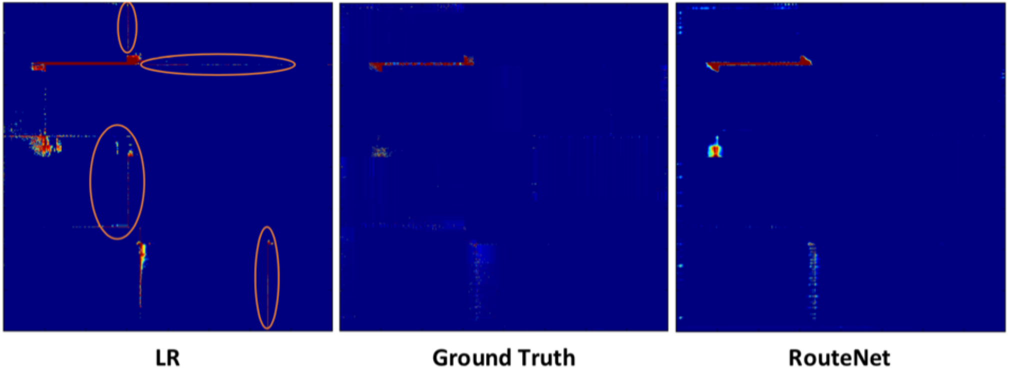
$$Y^{clip}_{i_{mn}} = min(Y_{i_{mn}}, c)$$

$$Loss = \sum_{i=1}^{N} \sum_{m=1}^{w} \sum_{n=1}^{h} ||f_{hotspot}(X_{i_{mn}}) - Y^{clip}_{i_{mn}}||_2 + \lambda ||W||_2$$

Pixel-wise loss function

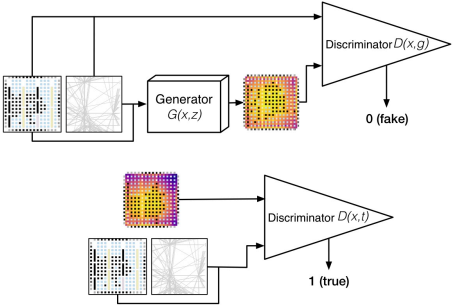
[3] Xie+, "RouteNet: Routability prediction for Mixed-Size Designs Using Convolutional Neural Network", ICCAD'18.

DRC Hotspot Detection Evaluation



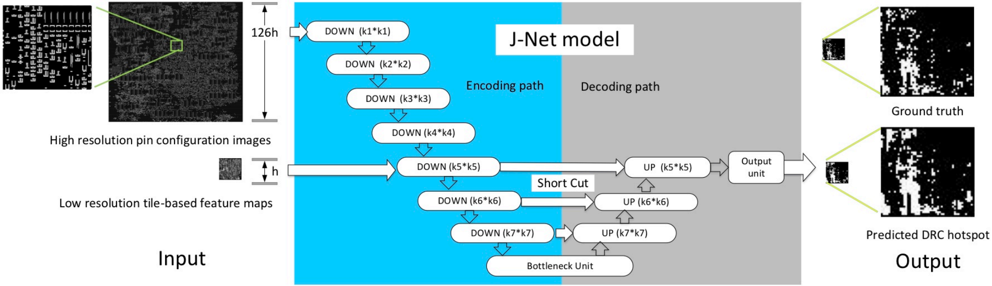
[3] Xie+, "RouteNet: Routability prediction for Mixed-Size Designs Using Convolutional Neural Network", ICCAD'18.

Conditional GAN [4]



- The inputs of this network include $image_{place}$ and $image_{connect}$;
- The target image is the routing heat map image $image_{route}$;
- It only uses the post-placement information without routing information.

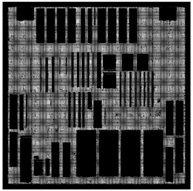
[4] Yu+, "Painting on Placement: Forecasting Routing Congestion using Conditional Generative Adversarial Nets", DAC'19.

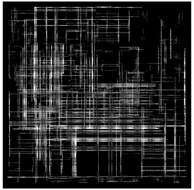


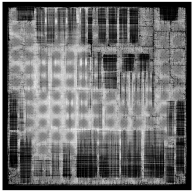
[5] Hung+, "Transforming global routing report into drc violation map with convolutional neural network", ISPD'20.

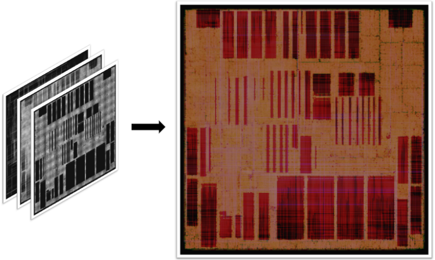
Our Study: Feature Extraction


Capacity


Pin number


Net density

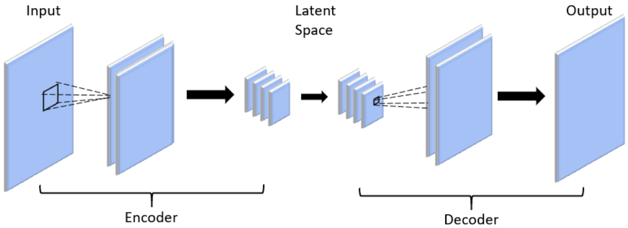

Congestion


Feed features into different RGB channels

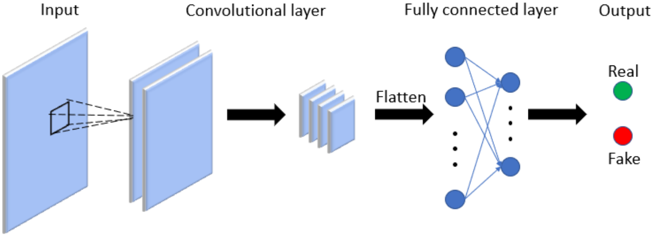
Our Study: Network Structure

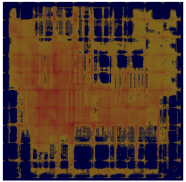
Generator:

Predictor

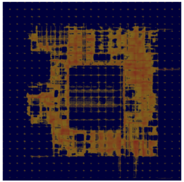


Discriminator:

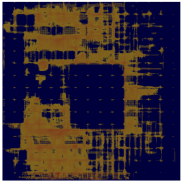




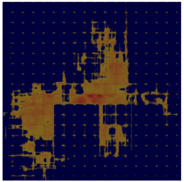
(a) adaptec1



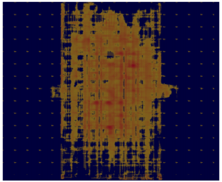
(b) adaptec3



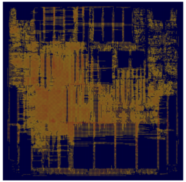
(c) adaptec5



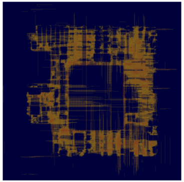
(d) bigblue3



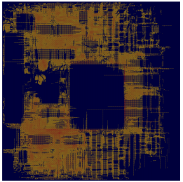
(e) newblue2



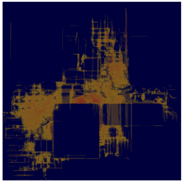
(f) adaptec1



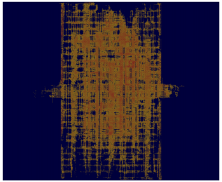
(g) adaptec3



(h) adaptec5



(i) bigblue3



(j) newblue2

Top row: predicted congestion map; Bottom row: actual congestion map.

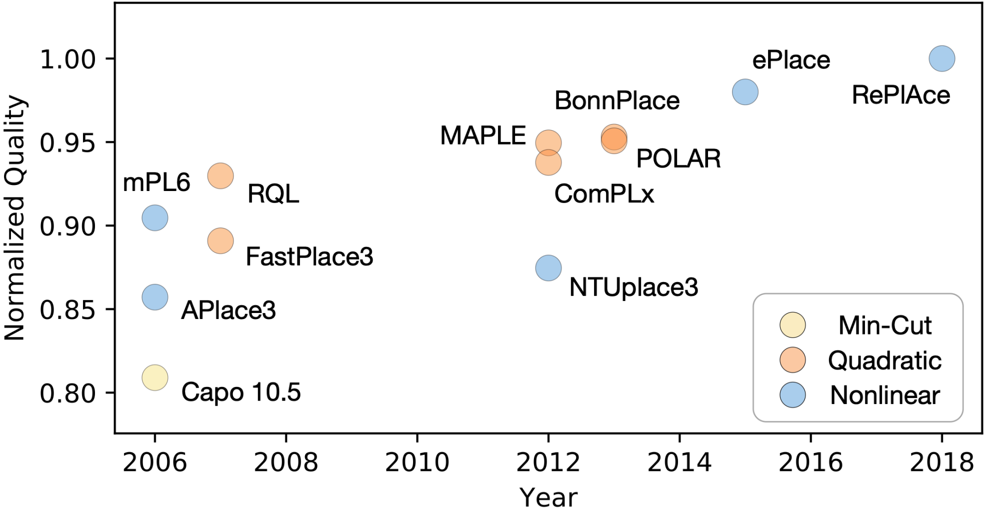
Plugged In Global Router [MLCAD'19][6]

	NTUgr2			NCTU-gr			NTHU-Route 2.0			Ours		
	TOF	WL	T(s)	TOF	WL	T(s)	TOF	WL	T(s)	TOF	WL	T(s)
a1	0	5.60	177.2	0	5.44	102.53	0	5.36	207.11	0	5.38	207.37
a3	0	13.41	157.7	0	13.11	154	0	13.15	225.84	0	13.10	263.89
a4	0	12.29	59.2	0	12.19	63.6	0	12.17	56.11	0	12.23	77.29
a5	0	16.03	520.4	0	15.95	381.71	0	15.53	549.98	0	15.64	611.39
b1	0	5.85	428.4	0	5.97	204.32	0	5.57	406.78	0	5.60	283.41
n2	0	7.66	27.6	0	7.59	35.73	0	7.59	30.82	0	7.59	30.65
n6	0	18.55	487.3	0	18.27	238.72	0	17.69	968.39	0	17.67	439.83
a2	0	5.36	39.8	0	5.27	36.02	2	5.23	93.4	0	5.24	51.23
n5	0	23.90	1220.1	0	23.46	281.47	18	23.14	721.52	0	23.21	399.4
b3	0	13.47	206	0	13.17	99.4	32	13.07	307.45	0	13.10	126.38
b2	2	9.42	6616.8	4	9.10	171.35	84	9.00	400.29	8	9.01	189.17
n1	38	4.87	14339.2	108	4.70	120.39	144	4.60	483.1	18	4.63	140.05
n4	148	13.55	16327.4	172	13.00	158.86	242	12.88	1032.89	172	12.90	449.65
b4	212	23.96	4478.6	512	23.17	277.64	266	22.78	2145.56	160	22.74	930.7
n3	31136	17.96	36325.5	37182	10.80	21053	n/a	n/a	>24 hrs	31050	10.70	2603.55
Total	31536	191.90	81411.2	37978	181.19	23378.74	>788	>167.78	>86400.00	31606	178.74	6803.96
Ratio	1.01	1.07	11.97	1.21	1.02	3.44	n/a	n/a	n/a	**1.00**	**1.00**	**1.00**

[6] Z. Zhou, Z. Zhu, J. Chen, Y. Ma, B. Yu, T. Ho, G. Lemieux and A. Ivanov, "Congestion-aware Global Routing using Deep Convolutional Generative Adversarial Networks", MLCAD'19.

Integrating Deep Learning Engine

Recent Development of Placement



Typical Nonlinear Placement Algorithm

$$\min_{\mathbf{x},\mathbf{y}} \quad \mathrm{WL}(\mathbf{x},\mathbf{y}),$$
$$\text{s.t.} \quad D(\mathbf{x},\mathbf{y}) \le t_d$$

Objective of nonlinear placement

$$\min \quad \underbrace{\left(\textstyle\sum_{e\in E} \mathrm{WL}(e;\mathbf{x},\mathbf{y})\right)}_{\text{Wirelength}} + \lambda \underbrace{D(\mathbf{x},\mathbf{y})}_{\text{Density}}$$

Challenges of Nonlinear Placement

Low efficiency

- >3h for 10M-cell design

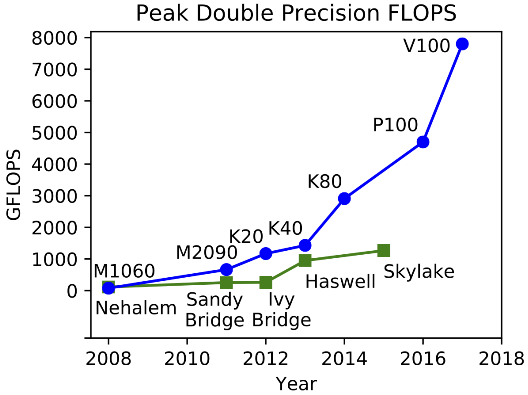
Limited acceleration

- Limited speedup, e.g. mPL, due to clustering

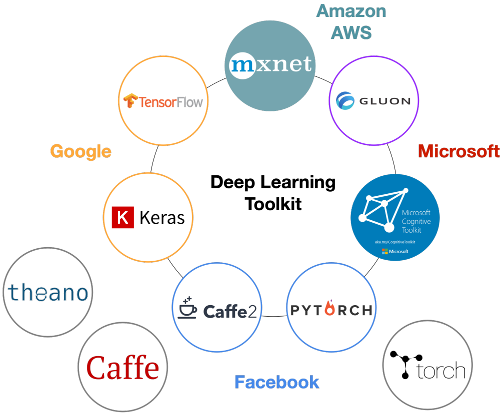
Huge development effort

- >1year for ePlace/RePlAce

Advances in Deep Learning Hardware/Software



Over **60x** speedup in neural network training since 2013



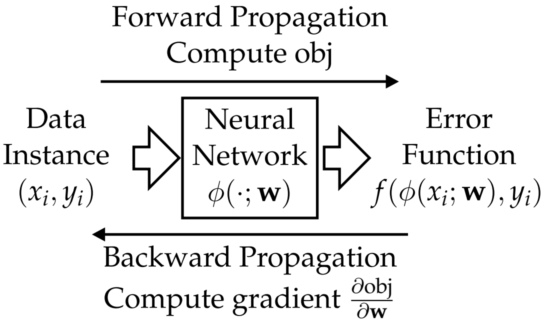
Deep learning toolkits

DREAMPlace Strategies

- Cast the non-linear placement problem into a neural network training problem.
- Leverage deep learning hardware (GPU) and software toolkit (e.g. `Pytorch`)
- Enable ultra-high parallelism and acceleration while getting the state-of-the-art results.

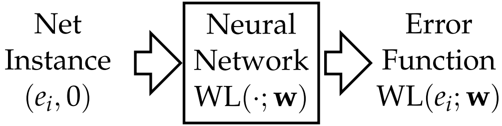
[7] Lin+, "DREAMPlace: Deep Learning Toolkit-Enabled GPU Acceleration for Modern VLSI Placement", DAC'19.

$$\min_{\mathbf{w}} \sum_{i}^{n} f(\phi(x_i; \mathbf{w}), y_i) + \lambda R(\mathbf{w})$$



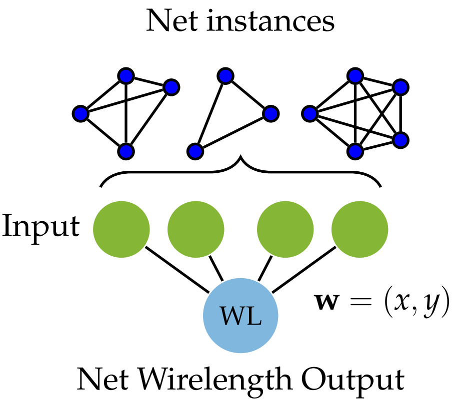
Train a neural network

$$\min_{\mathbf{w}} \sum_{i}^{n} \mathrm{WL}(\phi(x_i; \mathbf{w}), y_i) + \lambda D(\mathbf{w})$$



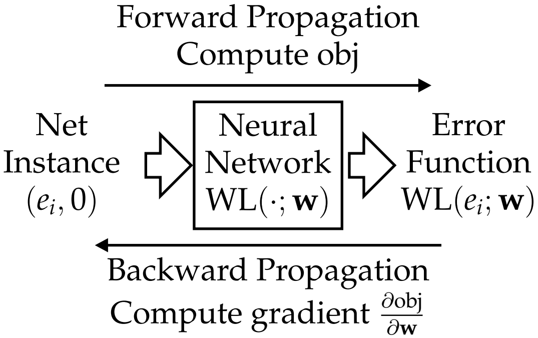
Solve a placement

Casting the placement problem into neural network training



Train a neural network

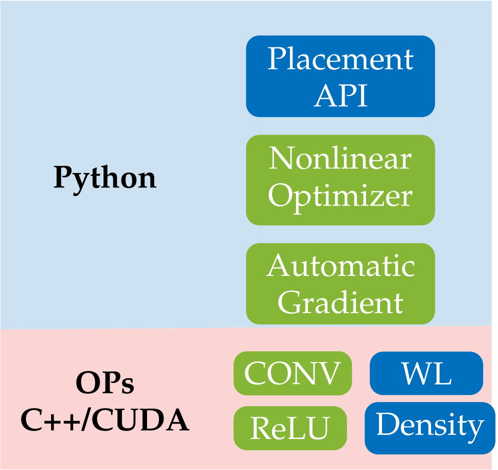
$$\min_{\mathbf{w}} \sum_{i}^{n} \mathrm{WL}(e_i; \mathbf{w}) + \lambda D(\mathbf{w})$$



Solve a placement

DREAMPlace Architecture [8]

Leverage highly optimized deep learning toolkit PyTorch



Match RePlAce

Nesterov's Method

[TCAD'18,Cheng+]

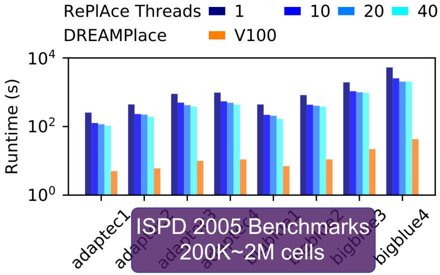
[8] Cheng+, "RePlAce: Advancing solution quality and routability validation in global placement", TCAD'18.

Experimental Results

DREAMPlace
- CPU: Intel E5-2698 v4 @2.20GHz
- GPU: 1 NVIDIA Tesla V100
- Single CPU thread was used

RePlAce [TCAD'18,Cheng+]
- CPU: 24-core 3.0 GHz Intel Xeon
- 64GB memory allocated

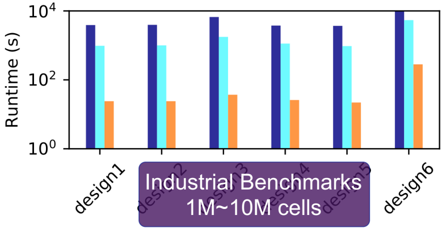
34x speedup



Same quality of results!

10M-cell design
finishes within **5min c.f. 3h**

43x speedup



Future Directions

New Solvers

SGD, ADAM, etc.

New Objectives

Routability, timing, etc.

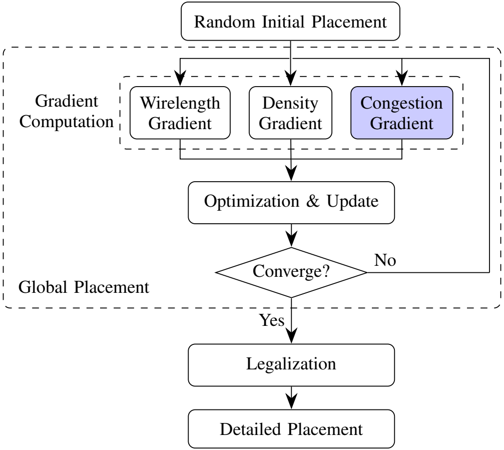
DREAM
BIGGER

Gate sizing,
floorplanning,
...

Applicable to Other CAD Problems

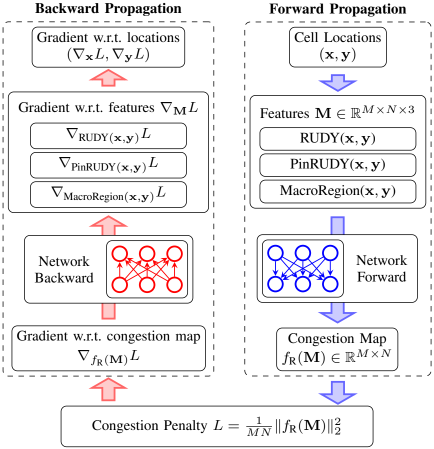
Multi-GPU,
distributed computing,
mixed precision,
...

New Accelerations



Overall Flow

[9] S. Liu, Q. Sun, P. Liao, Y. Lin and B. Yu, "Global placement with deep learning-enabled explicit routability optimization", DATE'21.



- Features Extraction:

$$RUDY(\mathbf{x}, \mathbf{y}); PinRUDY(\mathbf{x}, \mathbf{y}); MacroRegion(\mathbf{x}, \mathbf{y}).$$

$$\mathbf{M} : \mathbb{R}^{|\mathbf{x}|} \times \mathbb{R}^{|\mathbf{y}|} \longrightarrow \mathbb{R}^{M\times N\times 3}.$$

- Prediction Network:

$$f_{\mathrm{R}} : \mathcal{X} \subset \mathbb{R}^{M\times N\times 3} \longrightarrow \mathcal{Y} \subset \mathbb{R}^{M\times N}.$$

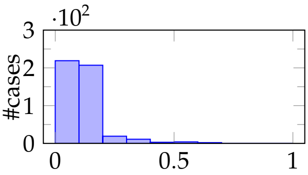
- Congestion Penalty Computation:

$$L(\mathbf{X}) = \frac{1}{MN}\|\mathbf{X}\|_2^2.$$

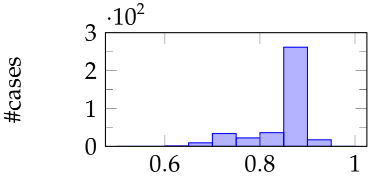
- Mathematical Expression:

$$L(f_R(\mathbf{M}(\mathbf{x}, \mathbf{y})))$$

Prediction Accuracy

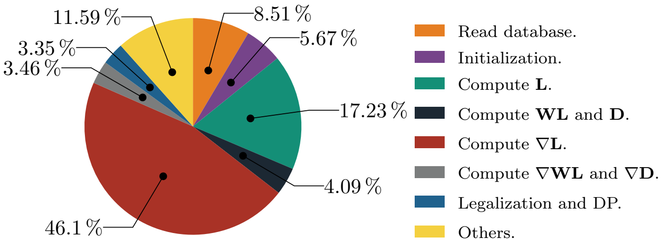


(a) NRMS statistics.



(b) SSIM statistics.

Prediction model evaluation.



Placement Improvement

Table: Experiment results on ISPD 2015 benchmarks.

Benchmark	NTUplace4dr [10]				DREAMPlace [7]				Ours			
	H-CR	V-CR	WL (e+06 um)	RT (s)	H-CR	V-CR	WL (e+06 um)	RT (s)	H-CR	V-CR	WL (e+06 um)	RT (s)
des_perf_1	0.101	0.038	1.32	331	0.143	0.129	1.23	10.868	0.153	0.126	1.23	44.07
des_perf_a	0.022	0.038	2.25	345	0.015	0.021	2.05	12.834	0.020	0.028	1.91	44.51
des_perf_b	0.001	0.002	1.75	349	0.005	0.010	1.71	11.829	0.004	0.010	1.71	45.99
fft_1	0.125	0.093	0.52	79	0.106	0.063	0.45	7.656	0.101	0.061	0.45	43.66
fft_2	0.821	0.002	0.53	113	0.664	0.006	0.44	7.636	0.665	0.006	0.44	48.05
fft_a	0.116	0.015	0.82	111	0.248	0.016	1.08	7.317	0.191	0.015	0.97	42.78
fft_b	0.211	0.067	1.05	101	0.177	0.026	1.21	7.942	0.142	0.047	1.12	46.81
matrix_mult_1	0.156	0.057	2.57	297	0.165	0.340	2.19	13.69	0.168	0.334	2.19	52.9
matrix_mult_2	0.210	0.073	2.41	344	0.253	0.238	2.28	13.69	0.251	0.242	2.28	52.25
matrix_mult_a	0.017	0.028	3.65	374	0.145	0.113	5.45	15.30	0.020	0.024	3.49	47.91
matrix_mult_b	0.032	0.035	3.67	307	0.044	0.025	4.51	14.91	0.020	0.028	3.47	50.5
matrix_mult_c	48.956	29.719	126.71	2674	0.089	0.017	4.87	14.38	0.029	0.016	3.42	48.41
pci_bridge32_a	0.110	0.056	0.54	121	0.192	0.098	0.52	7.33	0.076	0.036	0.43	44.64
pci_bridge32_b	0.001	0.004	0.77	95	0.001	0.005	0.83	8.08	0.002	0.008	0.65	43.72
superblue12	0.034	0.495	46.70	10813	0.125	0.374	38.1	96.18	0.131	0.379	36.46	547.8
superblue14	0.064	0.056	29.50	7010	0.041	0.051	26.1	54.26	0.055	0.081	25.28	168.35
superblue16_a	0.186	0.031	33.40	7068	0.090	0.013	28.2	54.92	0.164	0.028	28.70	170.21
superblue19	0.022	0.089	20.50	7890	0.033	0.093	17.0	42.23	0.039	0.091	16.70	97.84
Average	0.131	**0.069**	8.94	2102.82	0.141	0.091	7.68	**22.28**	**0.124**	0.087	**7.27**	91.13

Achieve up to 9.05% reduction in the congestion rate and 5.30% reduction in routed wirelength compared with DREAMPlace and NTUplace4dr.

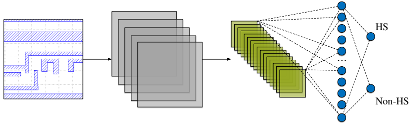
[10] Huang+, "NTUplace4dr: a detailed-routing-driven placer for mixed-size circuit designs with technology and region constraints", TCAD'17.

[7] Lin+, "DREAMPlace: Deep Learning Toolkit-Enabled GPU Acceleration for Modern VLSI Placement", DAC'19.

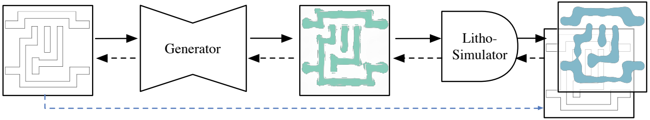
Graph Learning

Challenge: Irregular Structure Learning

- Verification [Yang et.al TCAD'2018]

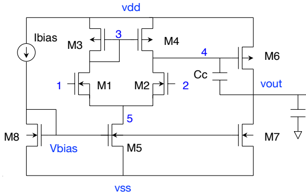


- Mask optimization [Yang et.al DAC'2018]

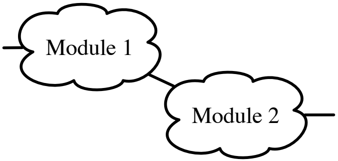


More Considerations

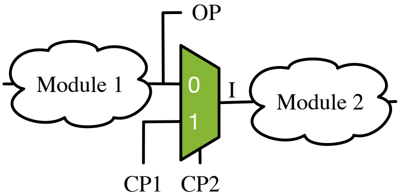
- Existing attempts still rely on regular format of data, like images;
- Netlists and layouts are naturally represented as graphs;
- Few DL solutions for graph-based problems in EDA.



- Fig. (a): Original circuit with bad testability. Module 1 is unobservable. Module 2 is uncontrollable;
- Fig. (b): Insert test points to the circuit;
- (CP1, CP2) = (0, 1) $\rightarrow$ line I = 0; (CP1, CP2) = (1, 1) $\rightarrow$ line I = 1;
- CP2 = 0 $\rightarrow$ normal operation mode.



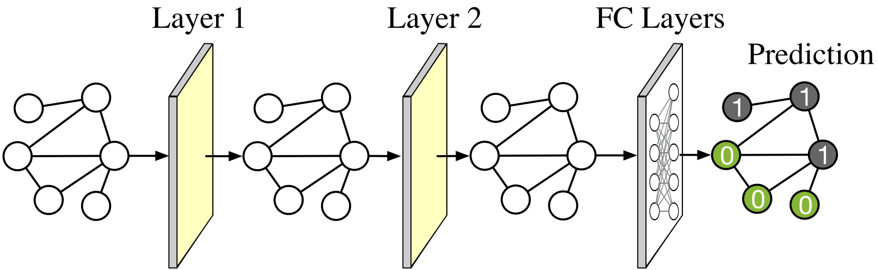
(a)



(b)

Node Classification [DAC'19][11]

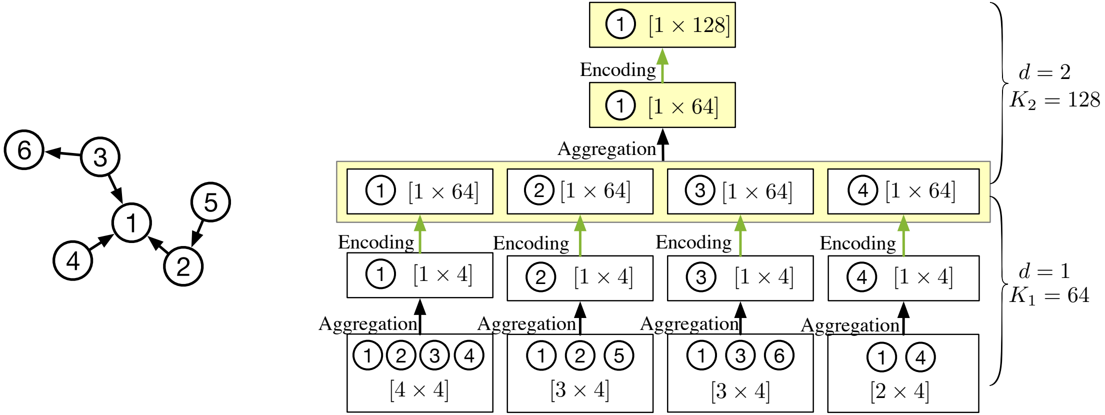
- Represent a netlist as a directed graph. Each node represents a gate.
- Initial node attributes: SCOAP values [12].
- Graph convolutional networks: compute node embeddings first, then perform classification.



[12] Goldstein+, "SCOAP: Sandia Controllability/Observability Analysis Program", DAC'80.

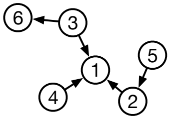
[11] Y. Ma, H. Ren, and B. Khailany, H. Sikka, L. Luo, K. Natarajan, and B. Yu, "High Performance Graph Convolutional Networks with Applications in Testability Analysis", DAC'19.

Node Embedding



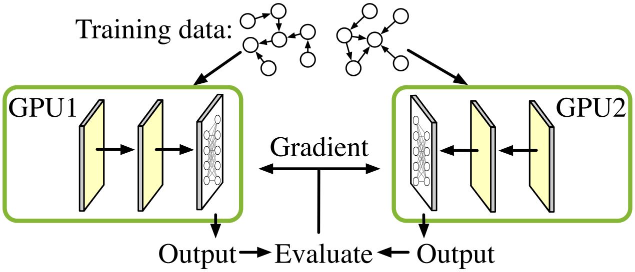
Efficient Inference

- Neighborhood overlap leads to duplicated computation $\rightarrow$ poor scalability.
- Transform weighted summation to matrix multiplication.
- Potential issue: adjacency matrix is too large.
- Fact: adjacency matrix is highly sparse! It can be stored using compressed format.



$$
\boldsymbol{G}_d = \boldsymbol{A} \cdot \boldsymbol{E}_{d-1} =
\begin{array}{c}
\\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\begin{bmatrix}
1 & w_1 & w_1 & w_1 & 0 & 0 \\
w_2 & 1 & 0 & 0 & w_1 & 0 \\
w_2 & 0 & 1 & 0 & 0 & w_2 \\
w_2 & 0 & 0 & 1 & 0 & 0 \\
0 & w_2 & 0 & 0 & 1 & 0 \\
0 & 0 & w_1 & 0 & 0 & 1
\end{bmatrix}
\end{array}
\times
\begin{bmatrix}
\boldsymbol{e}_{d-1}^{(1)} \\
\boldsymbol{e}_{d-1}^{(2)} \\
\boldsymbol{e}_{d-1}^{(3)} \\
\boldsymbol{e}_{d-1}^{(4)} \\
\boldsymbol{e}_{d-1}^{(5)} \\
\boldsymbol{e}_{d-1}^{(6)}
\end{bmatrix}
$$

- Adjacency matrix cannot be split as conventional way.
- A variant of conventional data-parallel scheme.
  - Each GPU process one graph instead of one "chunk";
  - Gather all to calculate the gradient.



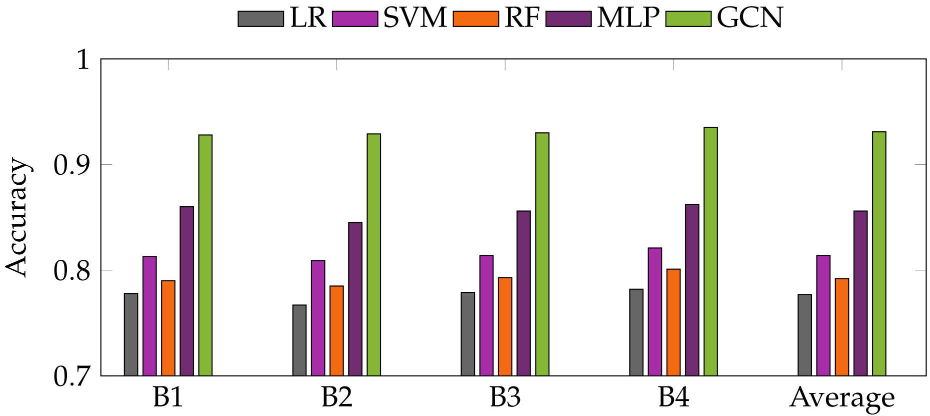
Benchmarks

- Industrial designs under 12nm technology node.
- Each graph contains > 1M nodes and > 2M edges.

Design	#Nodes	#Edges	#POS	#NEG
B1	1384264	2102622	8894	1375370
B2	1456453	2182639	9755	1446698
B3	1416382	2137364	9043	1407338
B4	1397586	2124516	8978	1388608

Classification Results Comparison

- Baselines: classical learning models with feature engineering in industry;
- GCN outperforms other classical learning algorithms.

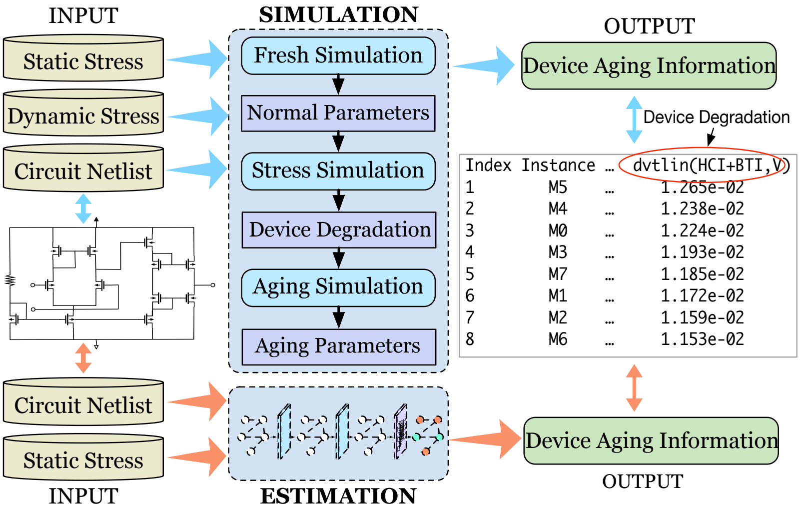


Testability Results Comparison

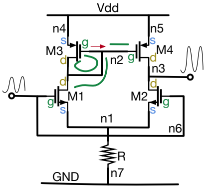
- Without loss on fault coverage, 11% reduction on test points inserted and 6% reduction on test pattern count are achieved.

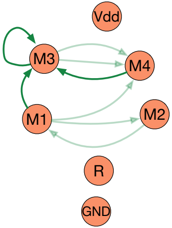
Design	Industrial Tool			GCN-Flow		
	#OPs	#PAs	Coverage	#OPs	#PAs	Coverage
B1	6063	1991	99.31%	5801	1687	99.31%
B2	6513	2009	99.39%	5736	2215	99.38%
B3	6063	2026	99.29%	4585	1845	99.29%
B4	6063	2083	99.30%	5896	1854	99.31%
Average	6176	2027	**99.32%**	**5505**	**1900**	**99.32%**
Ratio	1.00	1.00	**1.00**	**0.89**	**0.94**	**1.00**

Example 2: Aging Degradation Estimation

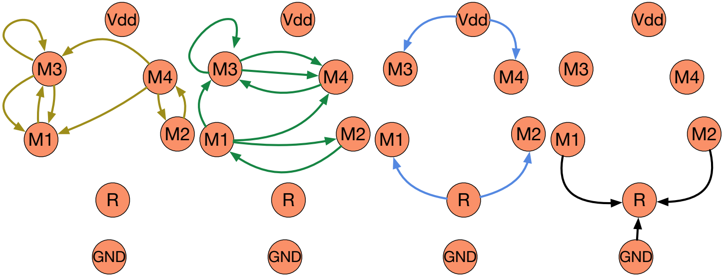


Heterogeneous Graph Representation[ASPDAC'21][13]





$$
\begin{array}{c}
 \\ M1 \\ M2 \\ M3 \\ M4 \\ R \\ Vdd \\ GND
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} M1 & M2 & M3 & M4 & R & Vdd & GND \end{array} \\
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{array}
$$



[13] T. Chen, Q. Sun, C. Zhan, C. Liu, H. Yu and B. Yu, "Analog IC Aging-induced Degradation Estimation via Heterogeneous Graph Convolutional Networks", ASPDAC'21.

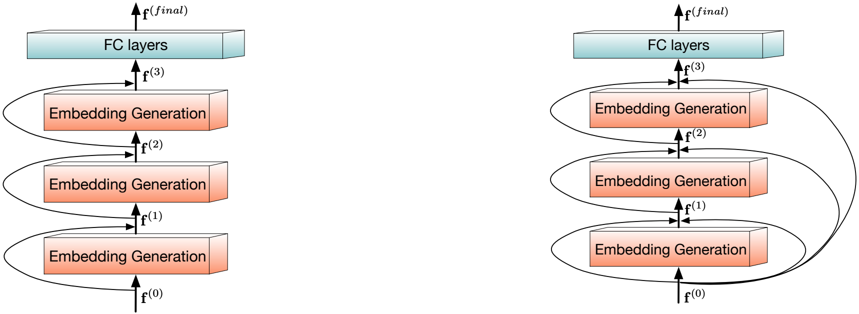
Unified Latent Space Mapping

$$\boldsymbol{F}^{(0)} = \sum_{t \in \mathcal{O}_{\mathcal{V}_{hmg}}} \boldsymbol{X}_t \cdot \boldsymbol{U}_t \in \mathbb{R}^{|\mathcal{V}_{hmg}| \times \tau}.$$

Heterogeneous embedding generation

$$\boldsymbol{F}^{(l)} = \sigma\left(\mathrm{CONCAT}\left(\tilde{\boldsymbol{A}} \cdot \boldsymbol{F}^{(l-1)}, \boldsymbol{F}^{(l-1)}\right) \cdot \boldsymbol{W}^{(l)}\right), \tilde{\boldsymbol{A}} \triangleq \sum_{r \in \mathcal{R}_{\mathcal{E}_{hmg}}} w_r \tilde{\boldsymbol{A}}_r.$$

Deep H-GCN [TCAD'21][14]



Deep heterogeneous embedding generation

$$\boldsymbol{F}^{(l)} = \sigma\left(\mathrm{CONCAT}\left(\tilde{\boldsymbol{A}} \cdot \boldsymbol{F}^{(l-1)}, \boldsymbol{F}^{(l-1)}, \boldsymbol{F}^{(0)}\right) \cdot \boldsymbol{W}^{(l)}\right), \boldsymbol{W}^{(l)} \leftarrow (1-\beta_l)\boldsymbol{I} + \beta_l \boldsymbol{W}^{(l)}.$$

[14] T. Chen, Q. Sun, C. Zhan, C. Liu, H. Yu and B. Yu, "Deep H-GCN: Fast Analog IC Aging-induced Degradation Estimation", TCAD'21.

Experimental results: benchmarks

Table: Statistics of Designs (industrial 5nm PLL design)

Design	#trans.	#device	#net
1	4,348	99,009	18,155
2	4,382	99,696	18,299
3	3,999	179,758	31,303
4	3,998	185,480	33,819
5	4,980	692,480	111,308
6	523	31,279	6,002
7	6,398	452,109	76,807
8	1,998	96,749	16,006

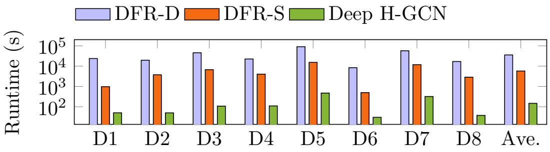
Table: Device Type

Type	#Param.
MOS	51
MOS spice	75
DIO/ESD	8
Cap	12
R	6
VSource	1

Results

Table: Accuracy

Design	DFR tool (static)		GCN [15]		GCNII [16]		H-GCN [13]		Deep H-GCN [14]	
	MAE	r^2 Score	MAE	r^2 Score	MAE	r^2 Score	MAE	r^2 Score	MAE	r^2 Score
1	4.009	0.181	1.316	0.703	1.332	0.691	0.914	0.821	**0.824**	**0.843**
2	4.072	0.194	1.389	0.596	1.339	0.619	0.893	0.814	**0.856**	**0.839**
3	4.543	0.327	4.070	0.588	4.166	0.599	2.302	0.817	**2.012**	**0.840**
4	4.515	0.332	4.111	0.588	4.177	0.551	2.575	0.746	**2.350**	**0.815**
5	4.160	0.277	3.750	0.521	4.021	0.354	2.525	0.787	**2.454**	**0.816**
6	3.962	0.395	2.077	0.802	2.092	0.802	1.661	0.834	**1.541**	**0.865**
7	4.319	0.266	3.166	0.685	3.168	0.689	2.889	0.826	**2.704**	**0.874**
8	4.594	0.224	3.491	0.637	3.748	0.610	2.670	0.786	**2.503**	**0.840**



[15] Hamilton+,"Inductive representation learning on large graphs", NIPS'17.

[16] Chen+, "Simple and deep graph convolutional networks", ICML'20.

[13] T. Chen, Q. Sun, C. Zhan, C. Liu, H. Yu and B. Yu, "Analog IC Aging-induced Degradation Estimation via Heterogeneous Graph Convolutional Networks", ASPDAC'21.

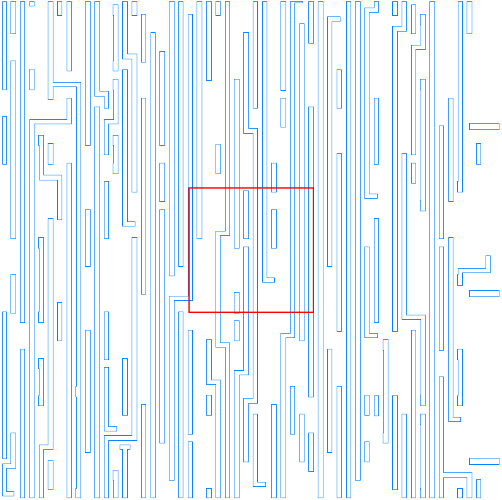
[14] T. Chen, Q. Sun, C. Zhan, C. Liu, H. Yu and B. Yu, "Deep H-GCN: Fast Analog IC Aging-induced Degradation Estimation", TCAD'21.

Conclusion

- Active learning leverages the gap between two different design spaces
- Extention: parameter tuning?

- Enable Deep Learning in back-end phase
- Extension: Integrated in Global/Detailed Placement

- A GCN-based methodology is proposed for netlist representation;
- GCN shows superior performance to classical learning models in test points insertion problem and circuit reliability analysis.



Acknowledgment

CUHK colleagues:

- Prof. Evengeline FY. Young; Prof. TY. HO, Prof. Martin DF. Wong

Industrial Liaison:

- Piyush Pathak, Frank Gennari, Ya-Chieh Lai, Jin Miao, Joydeep Mitra, Jhih-Rong Gao, Jian Kuang, Wen-Hao Liu, Zhuo Li, Charles J. Alpert (Cadence).
- Yi Zou, Jing Su (ASML); Minsik Cho (IBM); Subhendu Roy (Intel);
- Mark Ren, Brucek Khailany (nVIDIA)

Academic Liaison:

- David Z. Pan (UT Austin), Xuan Zeng, Fan Yang, Changhao Yan, Jianli Chen (Fudan);
- Shiyan Hu (University of Southampton); Wenjian Yu (Tsinghua University);
- Xin Li (Duke University), Song Chen (USTC); Wenxing Zhu (Fuzhou Univ);
- Bing Li, Ulf Schlichtmann (TUM); Yibo Lin (Peking University)

THANK YOU!