



北京大学  
PEKING UNIVERSITY

# 基于机器学习的集成电 路后端设计及加速

林亦波

高能效计算与应用中心

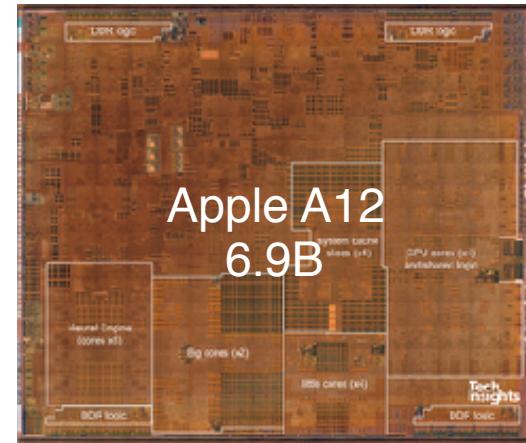
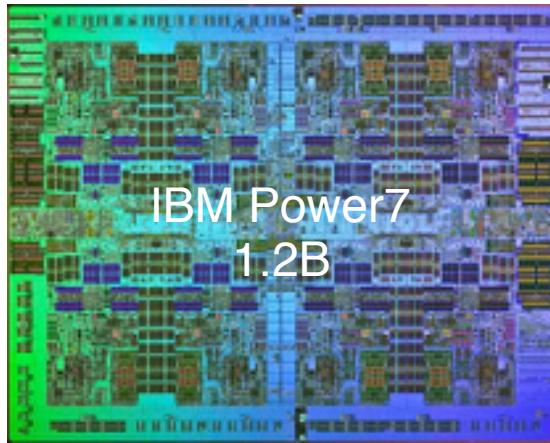
<http://yibolin.com/>

# 报告纲要

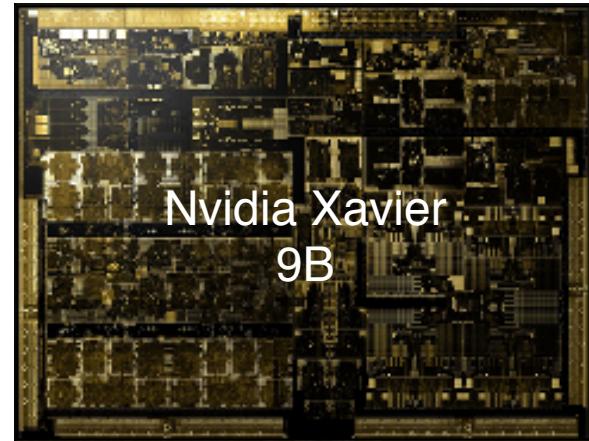
- 超大规模集成电路（VLSI）设计的挑战
- 机器学习带来的机遇
- 机器学习应用实例
  - 物理设计 [DAC'19最佳论文][ICCAD'19]
  - 物理验证 [ISPD'18, TCAD'18][DAC'19最佳论文提名]
- 总结



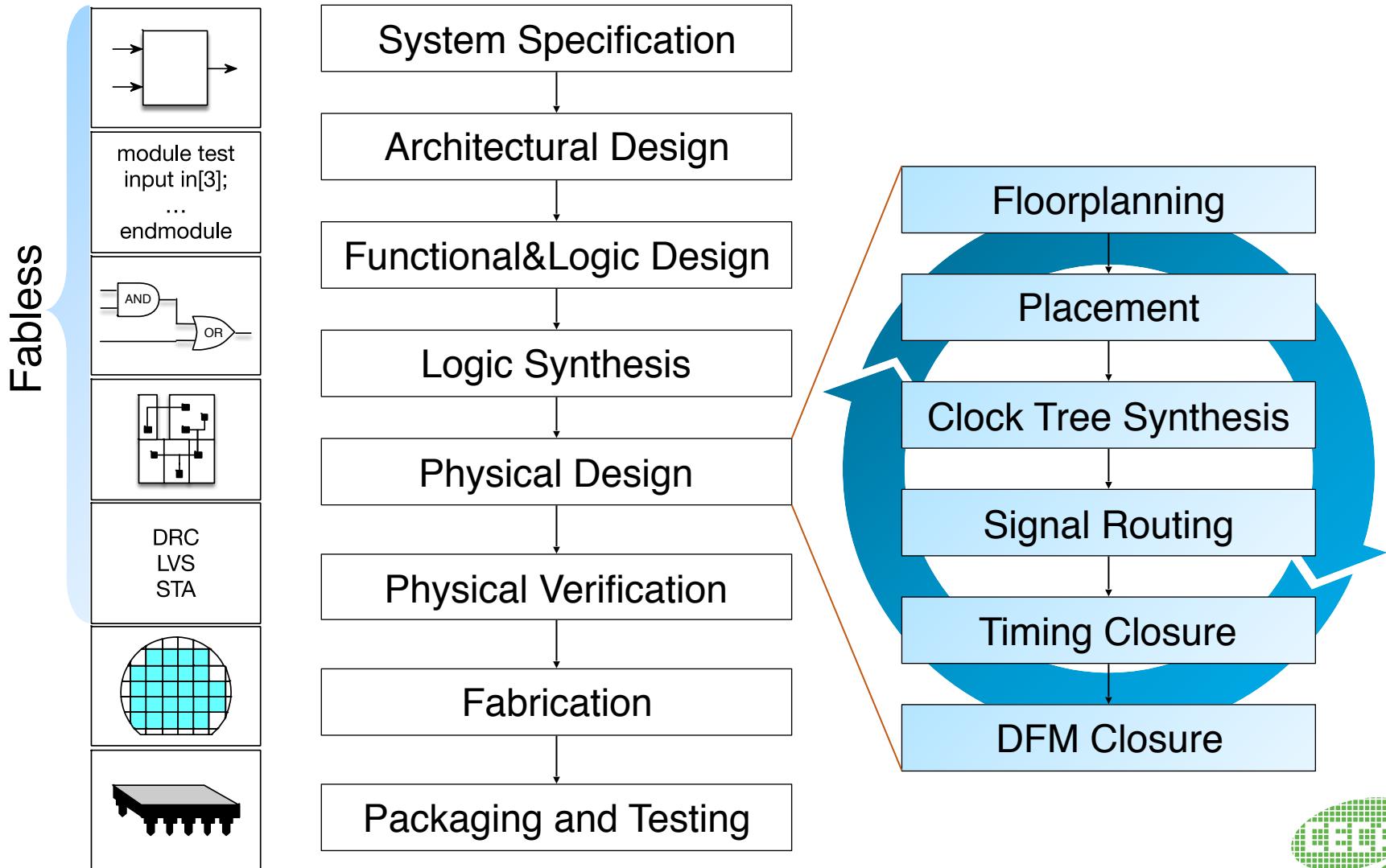
# Modern VLSI Layouts



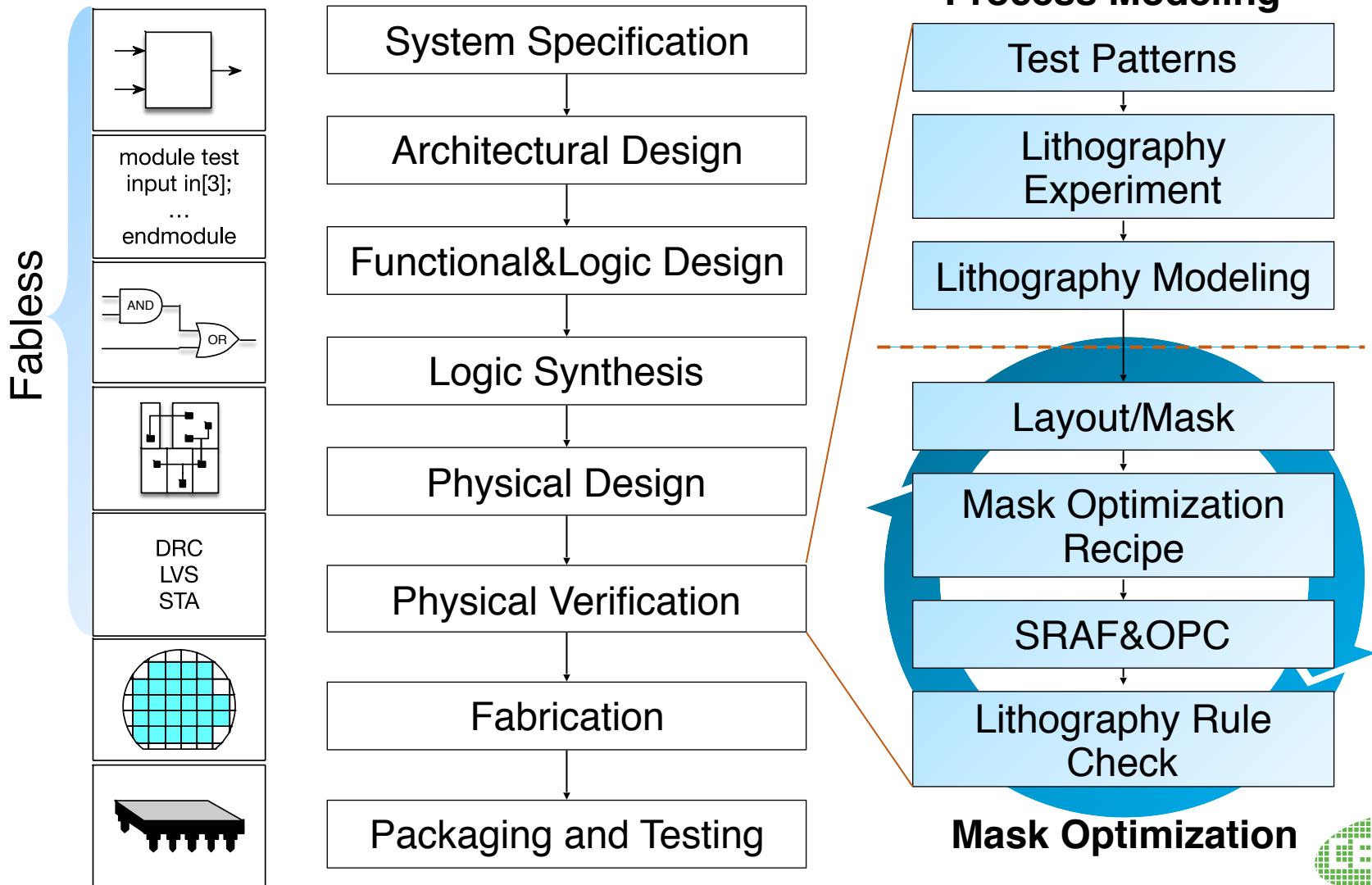
- Large scale: billions of transistors
- Complicated design flow
- Long design cycles



# IC Design Flow – Silicon Compiler



# IC Design Flow – Silicon Compiler



# Challenges for VLSI Design

- Long and complicated design flow
- Nested chicken-egg loops
  - P&R, OPC&SRAF...
- Nearly all problems are NP-hard
- Stacking of metaheuristics
- High expectation to optimality
  - Shoot for even 1% improvement
- Single iteration is expensive
  - One iteration of backend flow may take days
- Require many iterations for convergence



# Enormous Opportunities for Machine Learning



Automotive



Appliances



Healthcare



Entertainment



# Enormous Opportunities for Machine Learning



你是什么垃圾？

AI智能识别，不惧灵魂拷问

我分享传播也是公益  
成为第 39930 个“3小时公益”参与者

玻璃杯 属于  
可回收物  
“回收后加工可再利用”

» 打开手机淘宝 扫一扫查询垃圾分类 «

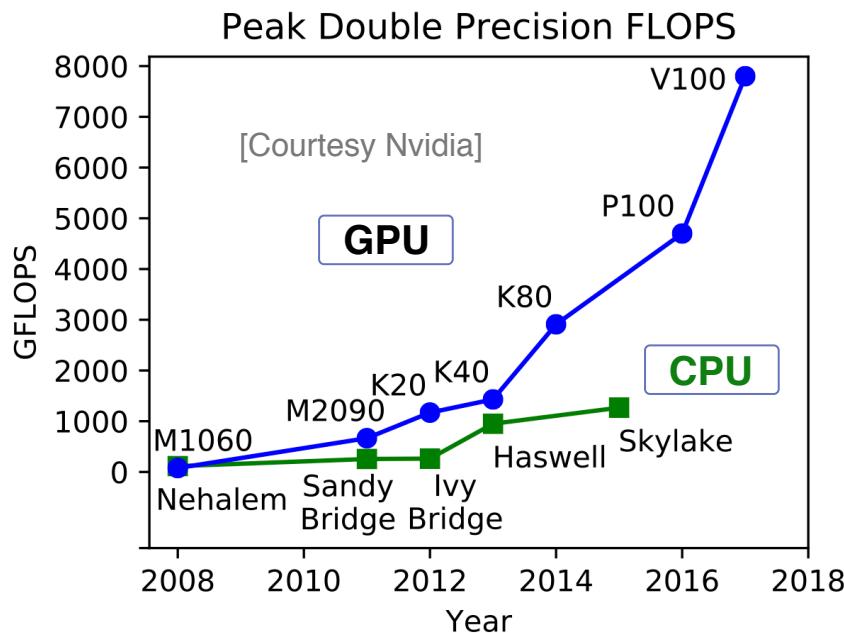
搜索 你是什么垃圾

垃圾分类

Trash Classification



# Advances in Deep Learning Hardware/Software



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



# 报告纲要

- 超大规模集成电路（VLSI）设计的挑战
- 机器学习带来的机遇
- 机器学习应用实例
  - 物理设计 [DAC'19最佳论文][ICCAD'19]
  - 物理验证 [ISPD'18, TCAD'18][DAC'19最佳论文提名]
- 总结



# VLSI Placement

**Placement is critical to VLSI design quality and design closure**

## Input

Gate-level netlist

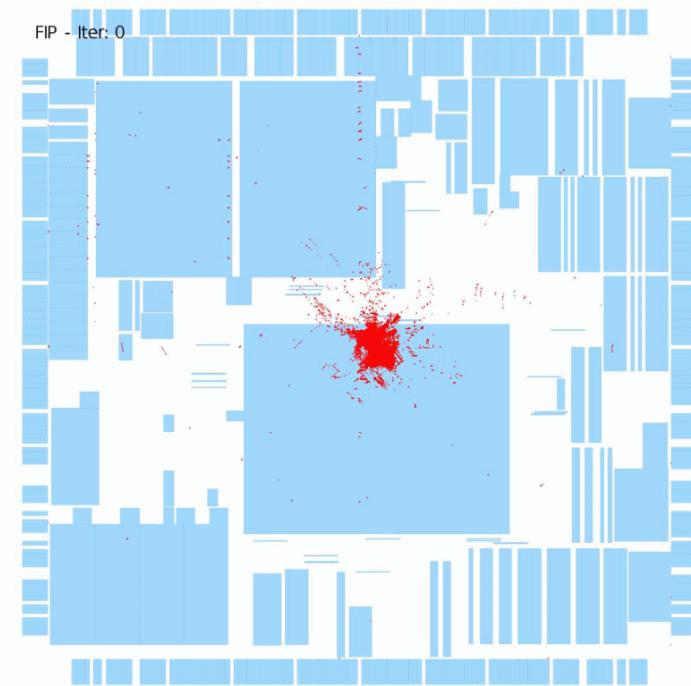
Standard cell library

## Output

Legal placement solution

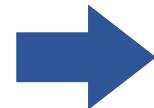
## Objective

Optimize wirelength,  
routability, etc.



# Nonlinear Placement Algorithm

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}} \quad & WL(\mathbf{x}, \mathbf{y}), \\ \text{s.t.} \quad & D(\mathbf{x}, \mathbf{y}) \leq t_d \end{aligned}$$



**Objective of nonlinear placement**

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

## Challenges of Nonlinear Placement

Low efficiency

- > 3h for 10M-cell design

Limited acceleration

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

Huge development effort

- > 1 year for ePlace/RePIAce

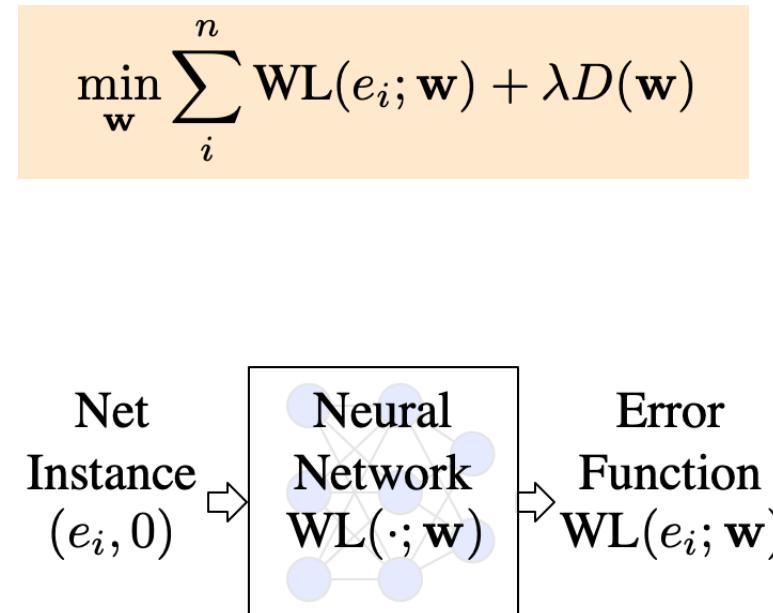
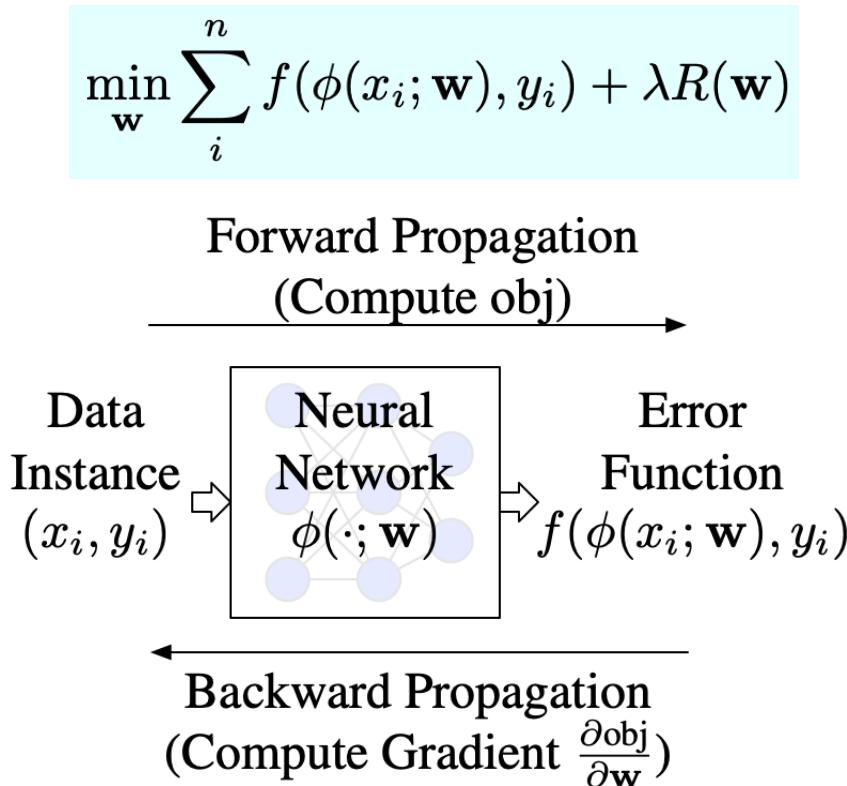


# DREAMPlace Strategies

- We propose a novel **analogy** by casting the nonlinear placement optimization into a neural network training problem
- Greatly 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



# Analogy between NN Training and Placement



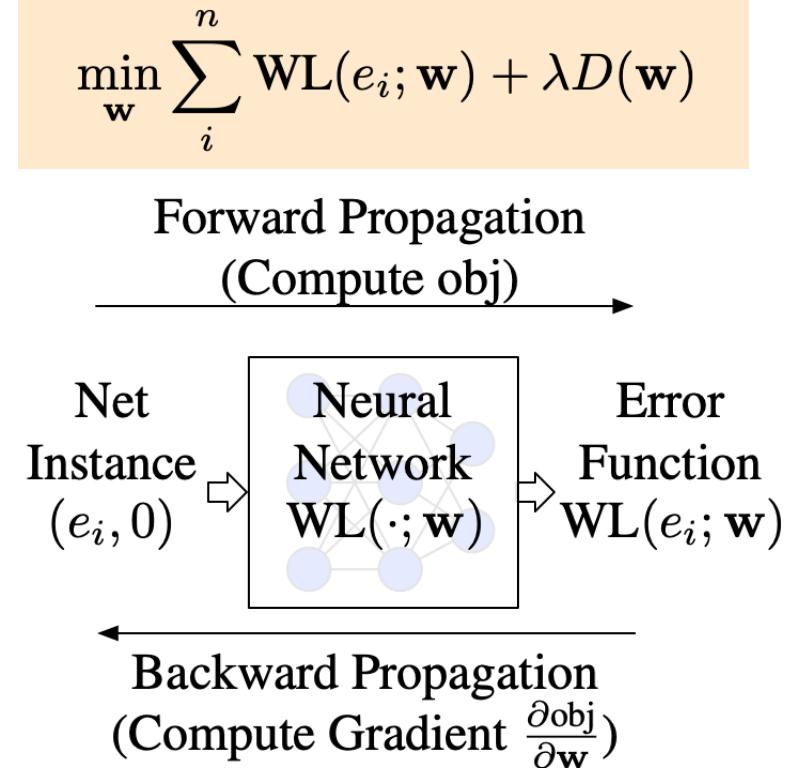
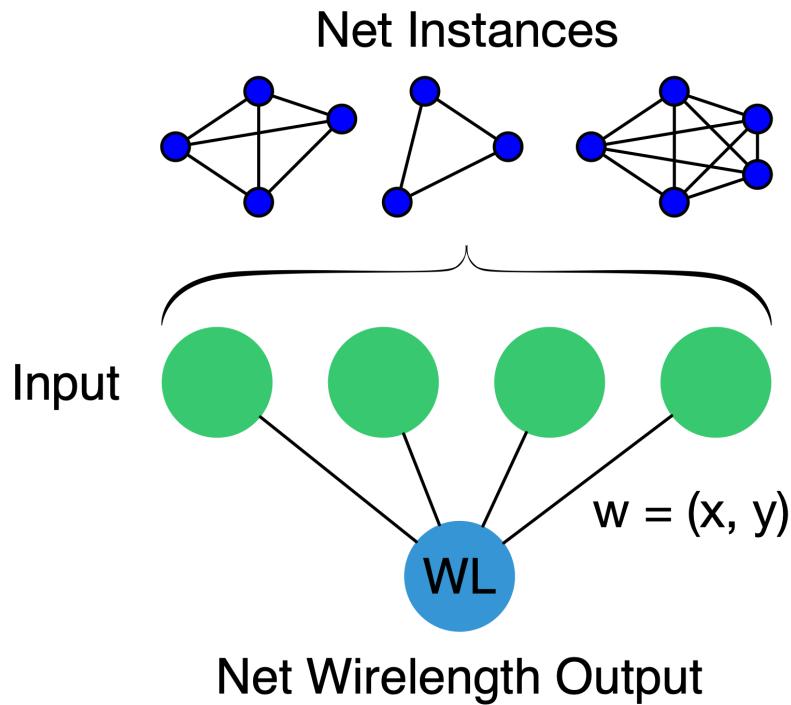
Train a neural network

Solve a placement

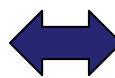


# Analogy between NN Training and Placement

Casting the placement problem into neural network training



Train a neural network

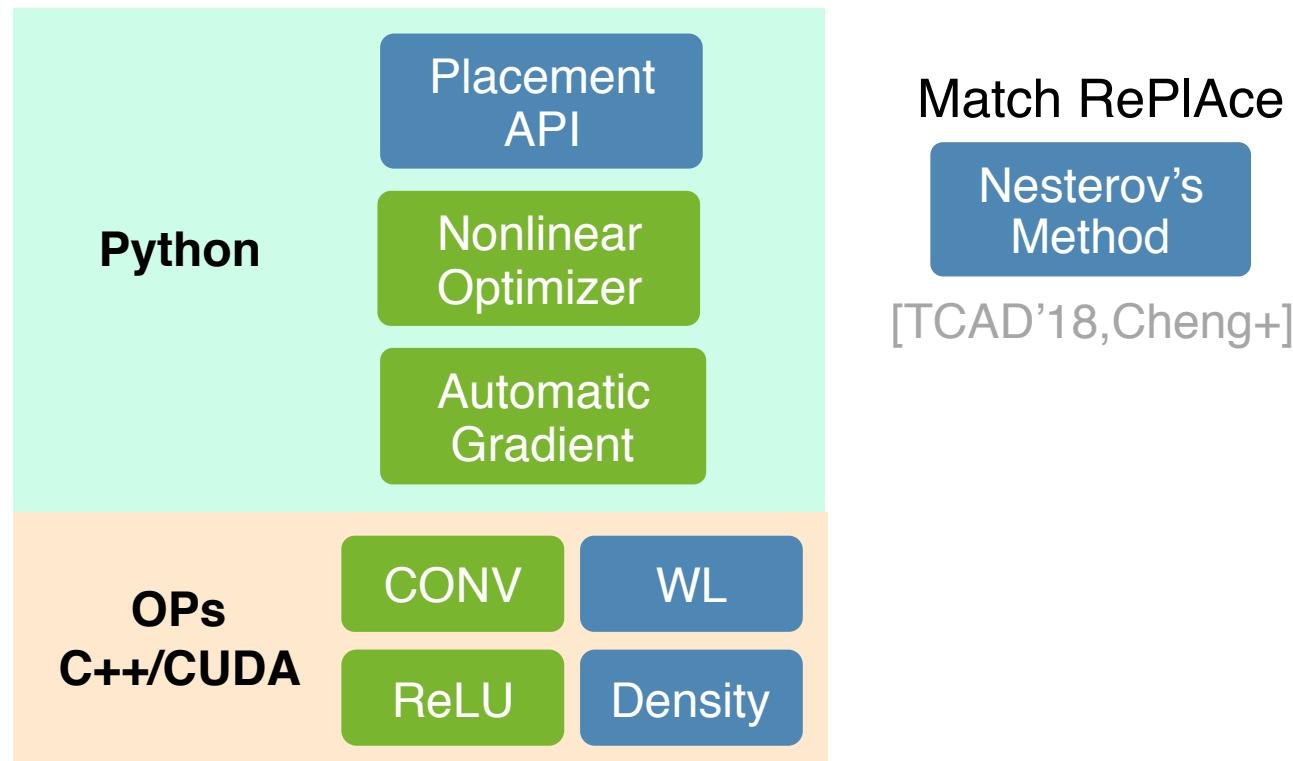


Solve a placement



# DREAMPlace Architecture

Leverage highly optimized deep learning toolkit PyTorch



# Experimental Results

## DREAMPlace

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

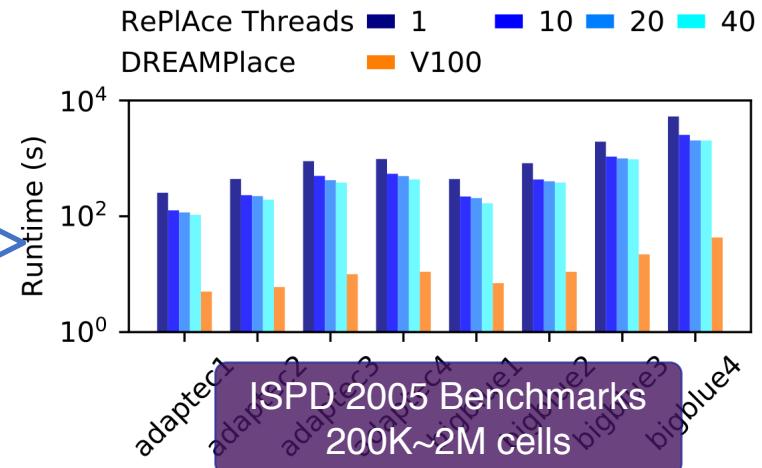
## RePIAce [TCAD'18, Cheng+]

- CPU: 24-core 3.0 GHz Intel Xeon
- 64GB memory allocated

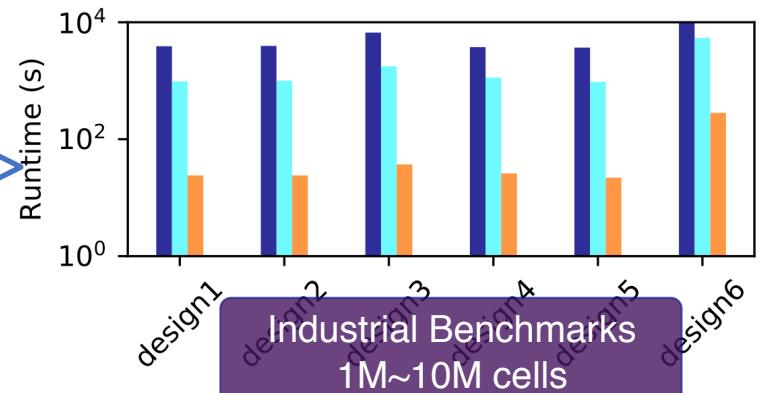
Same quality of results!

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

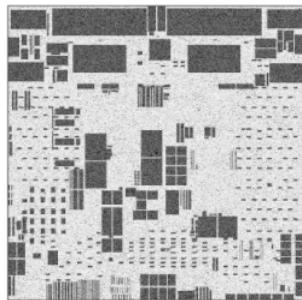
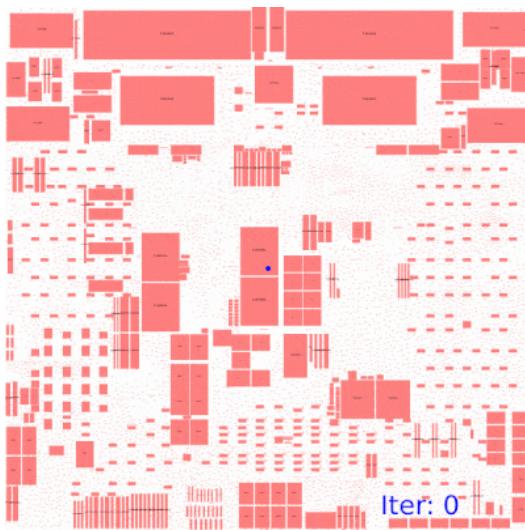
34x  
speedup



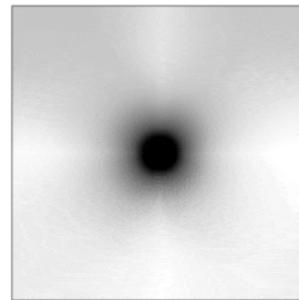
43x  
speedup



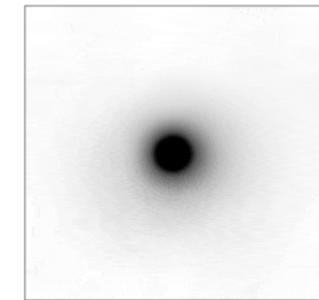
# Bigblue4 (2M-Cell Design)



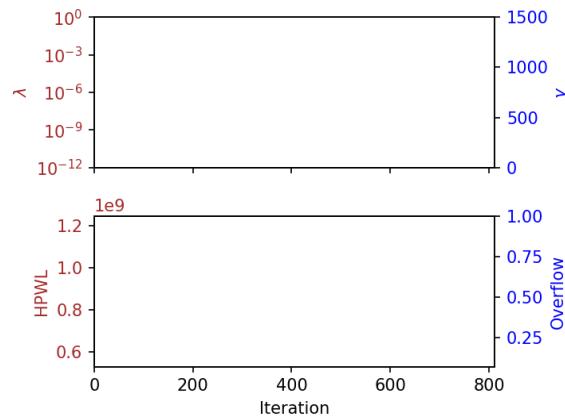
Density Map



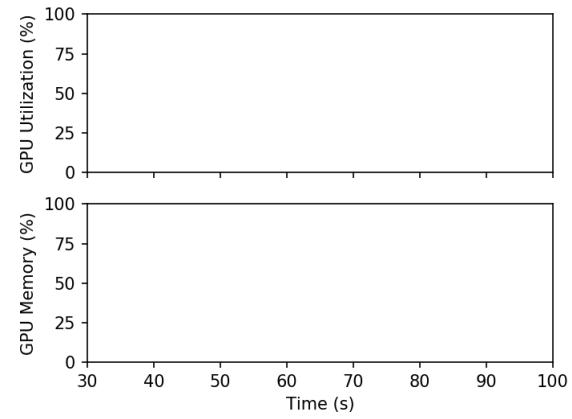
Potential Map



Field Map



Placement Metrics



GPU Usage on Titan Xp

Code release: <https://github.com/limbo018/DREAMPlace>



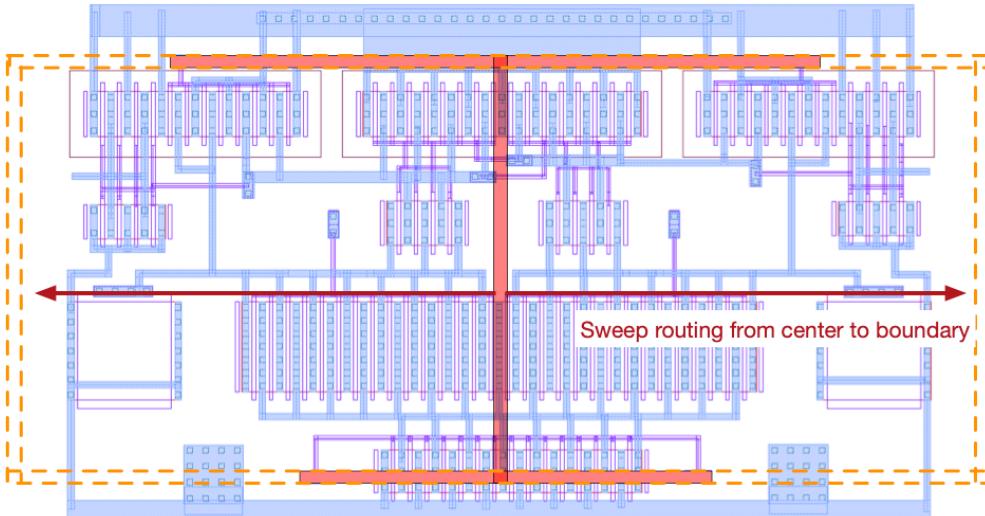
# Routing Guidance: GeniusRoute

Routing for analog circuits, e.g., comparator

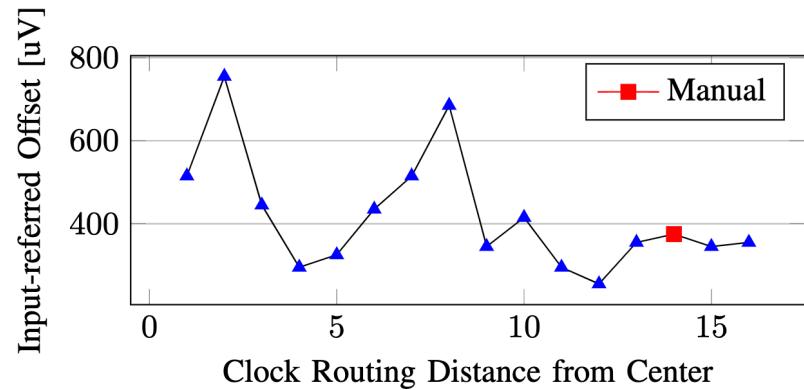
- Sensitive performance to clock routing

Existing manual layouts

- Hard to encode designer expertise into rules



Sweep the routing of the clock net



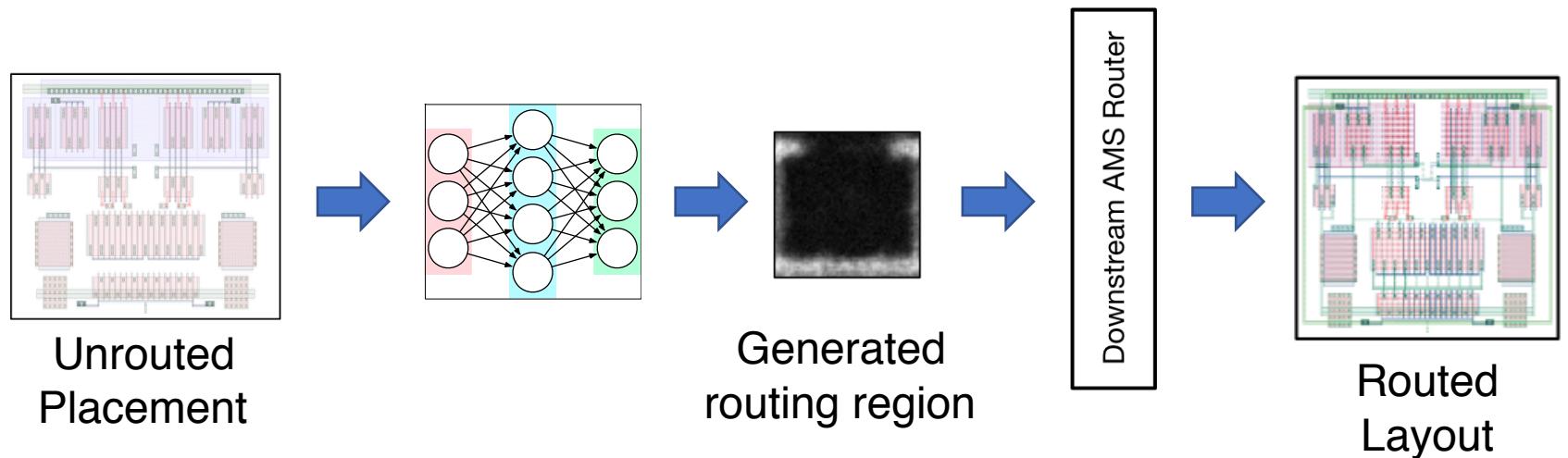
# Learn from Sea of Layouts

Learn from manual layouts

- Encode designer expertise into neural networks!

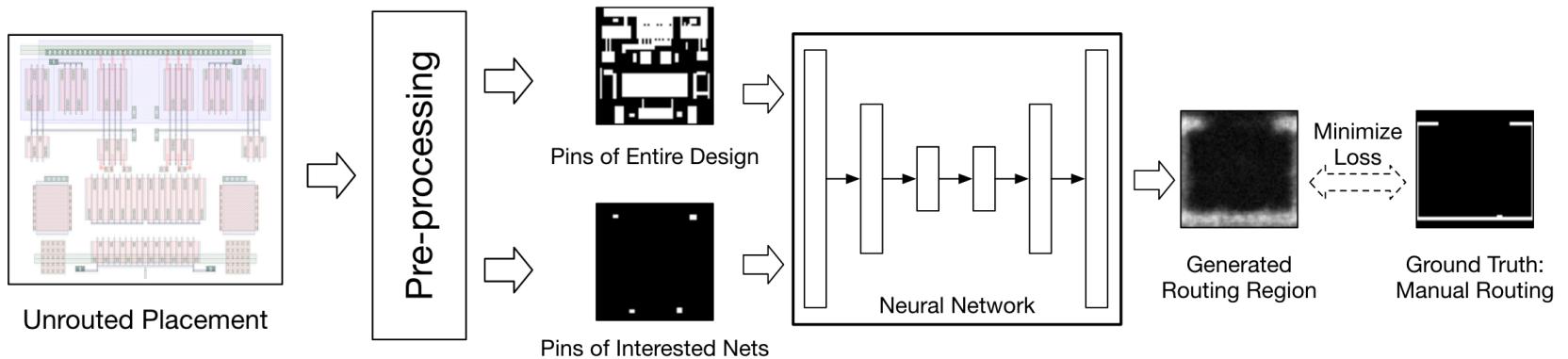
Generate routing guidance for critical nets

- Clock, power/ground, critical signal nets
- Probability map of routing



# Generate Routing Guidance

- Adopt autoencoder for routing region generation
- Compare with routing from manual layouts



# GeniusRoute Results

- Test on comparators and OTAs
- Evaluate with post layout simulation
- Compare with manual layout and previous methods

	Schematic	Manual	ICCAD'10	W/o guide	GeniusRoute
Offset ( $\mu\text{V}$ )	/	480	1230	2530	<b>830</b>
Delay (ps)	102	170	180	164	<b>163</b>
Noise ( $\mu\text{V}_{\text{rms}}$ )	439.8	406.6	437.7	439.7	<b>420.7</b>
Power ( $\mu\text{W}$ )	13.45	16.98	17.19	16.82	<b>16.80</b>

Closest results to the manual layout

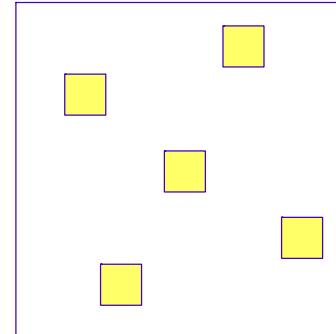
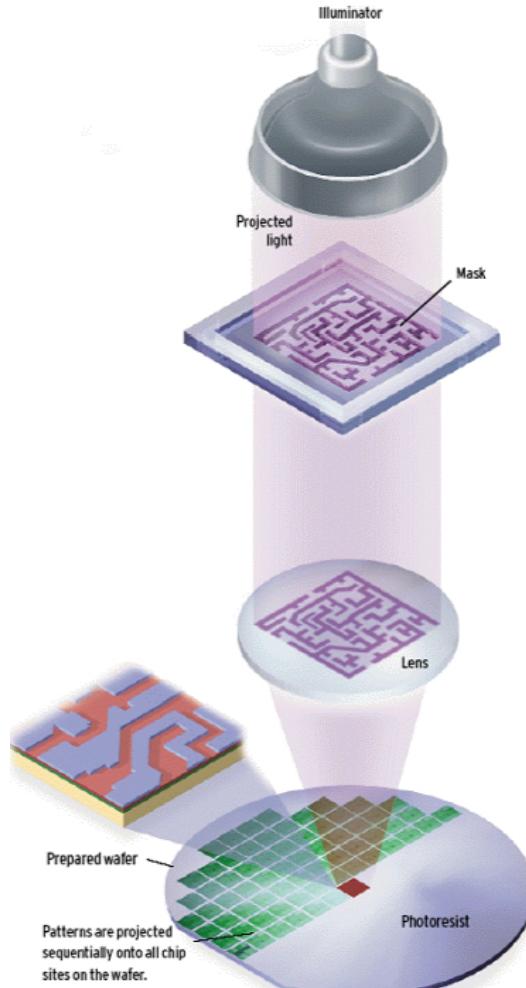


# 报告纲要

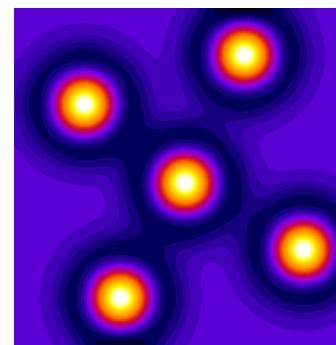
- 超大规模集成电路（VLSI）设计的挑战
- 机器学习带来的机遇
- 机器学习应用实例
  - 物理设计 [DAC'19最佳论文][ICCAD'19]
  - 物理验证 [ISPD'18, TCAD'18][DAC'19最佳论文提名]
- 总结



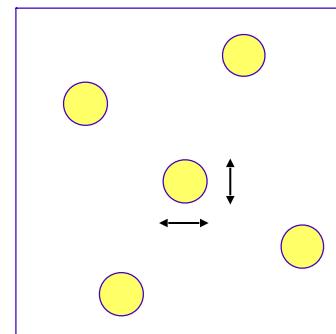
# Lithography Simulation



Contact Mask



Aerial Image  
(Light intensity map)

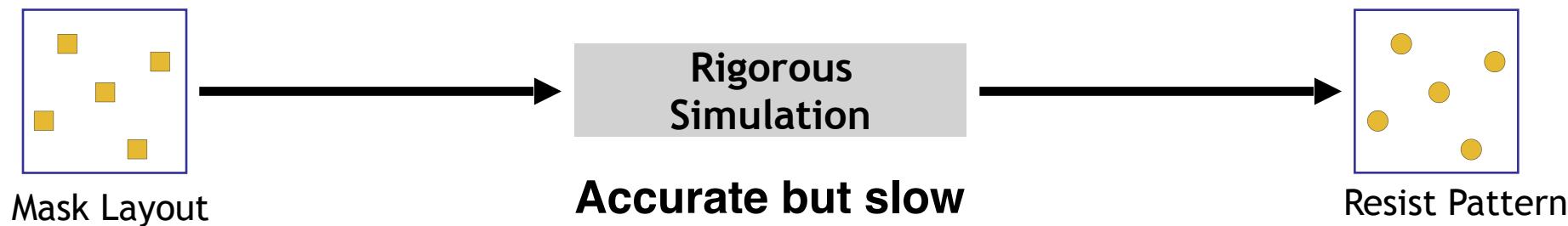


Resist Pattern



# Challenges in Lithography Modeling

Rigorous simulation: physics-based simulation, e.g., Synopsys S-Litho

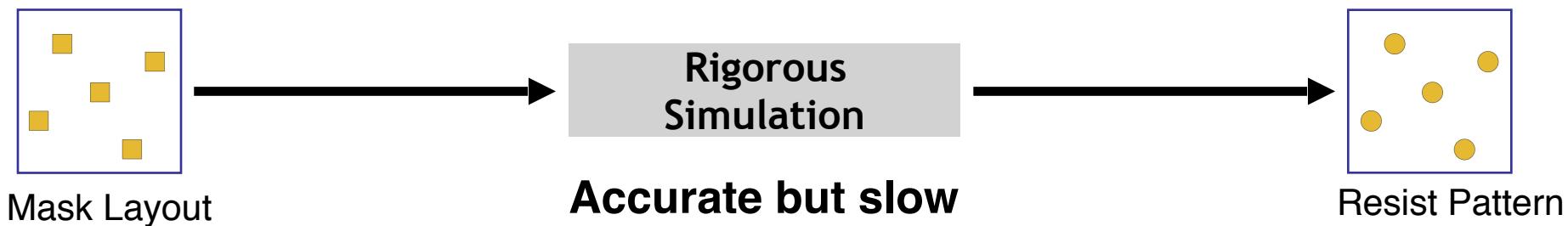


- Simulating  $2 \mu\text{m} \times 2 \mu\text{m}$  using Synopsys S-Litho  $\Rightarrow \sim 1$  minute
- A  $2 \text{ mm} \times 2 \text{ mm}$  chip contains 1M such clips  $\Rightarrow 1.9$  years!
- Intel Ivy Bridge 4C:  $160 \text{ mm}^2$

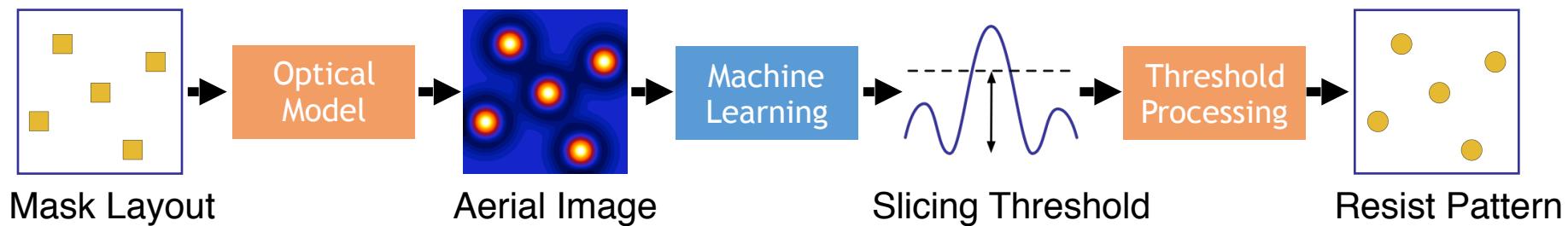


# ML for Lithography Modeling

Rigorous simulation: physics-based simulation, e.g., Synopsys S-Litho



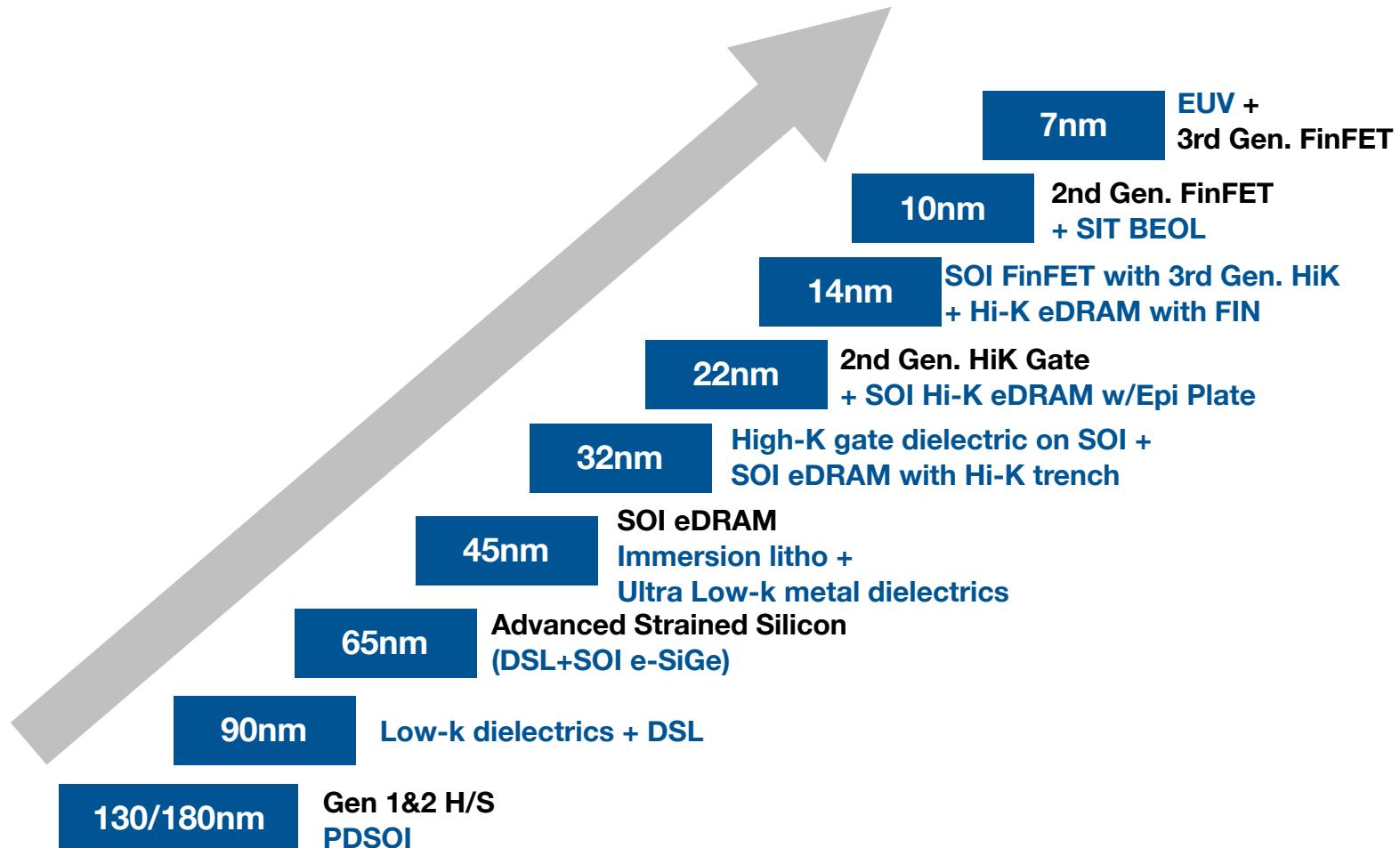
Machine learning for resist modeling [Watanabe+, SPIE'17] [Shim+, SPIE'17]



Speeds up the resist modeling stage



# VLSI Technology Nodes

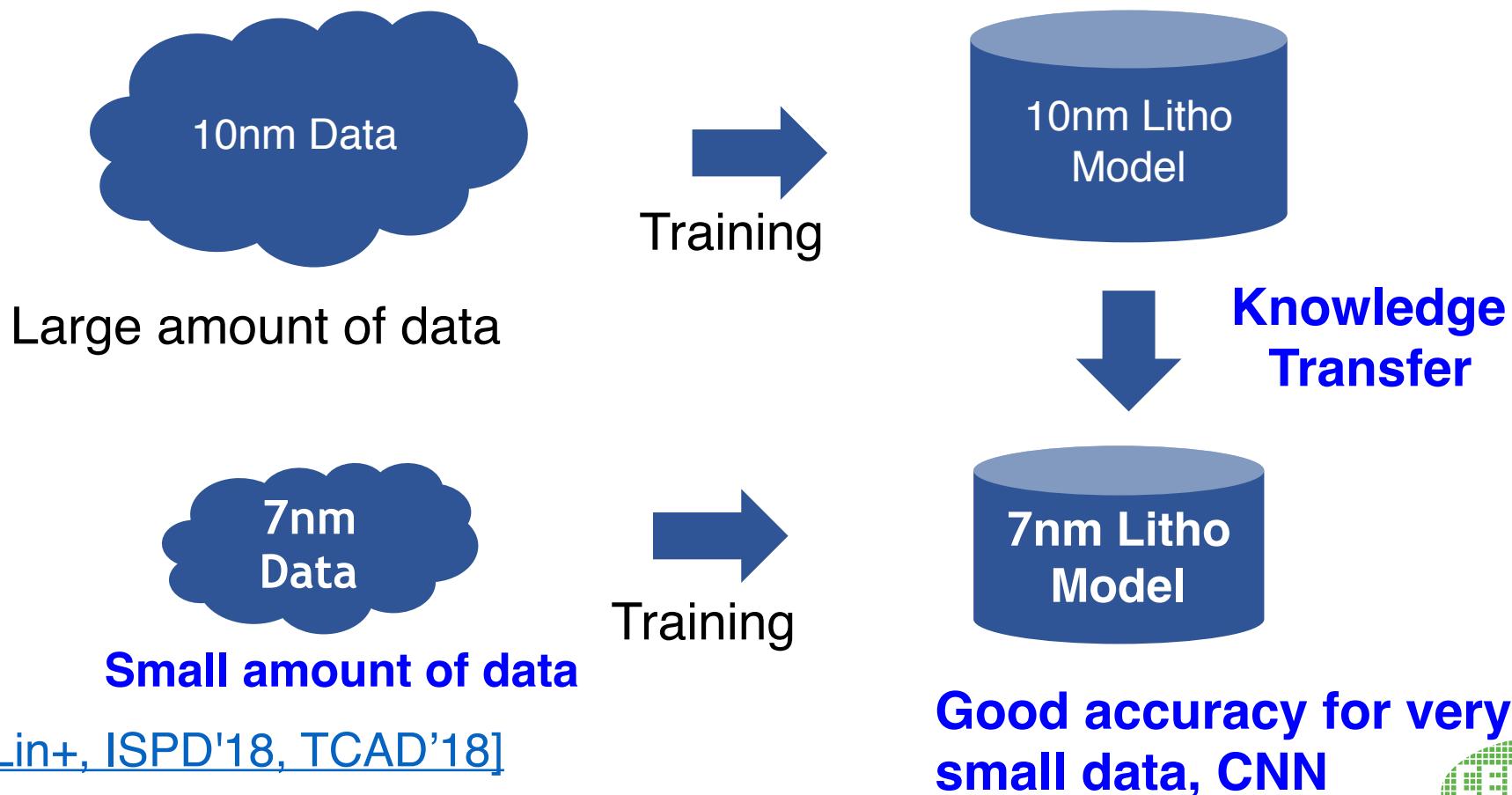


[electronicsforu, 2017]



# Transfer Learning for Lithography Modeling

Training with limited new tech. data + older tech.



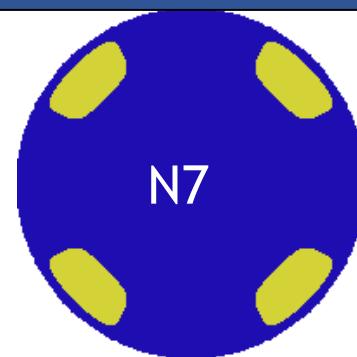
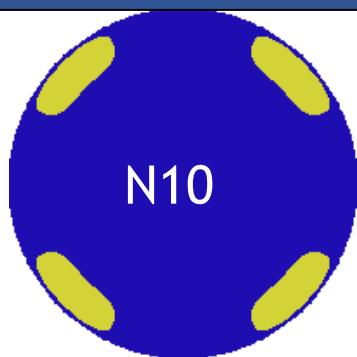
[Lin+, ISPD'18, TCAD'18]



# Technology Transition from N10 to N7

Contact Layer Design Rules [Liebmann, SPIE'15]		
	N10	N7
Patterning	LELE	LELELE

	N10	N7 <sub>a</sub>	N7 <sub>b</sub>
Design Rule	A	B	B
Optical Source	A	B	B
Resist Material	A	A	B



Resist A

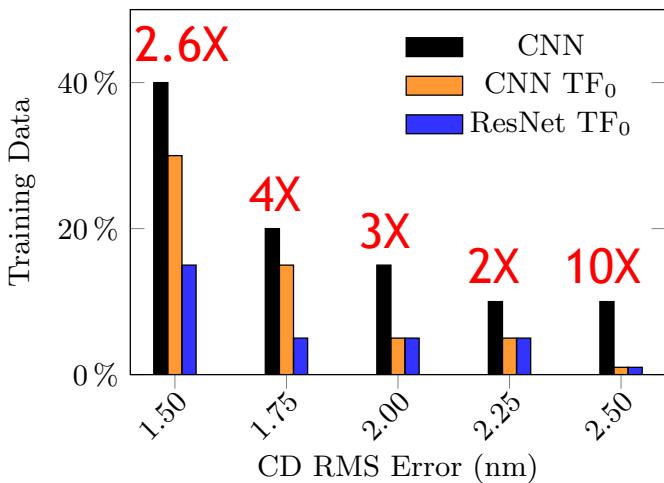
Resist B

Different dissolution slopes



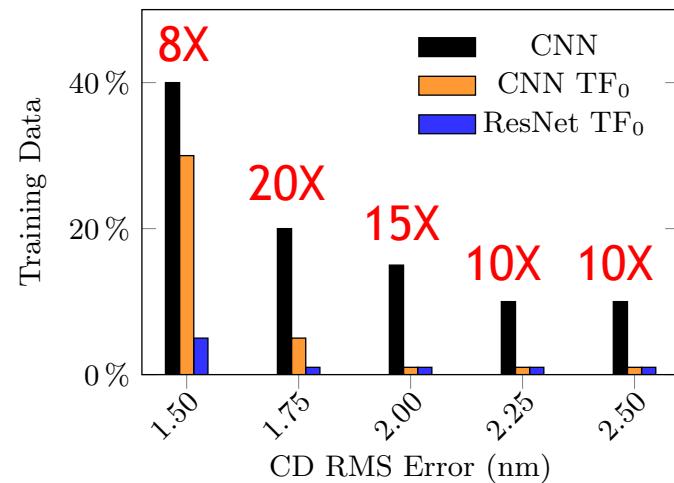
# Data Reduction from Knowledge Transfer

From N10 to N7<sub>b</sub>



2~10X reduction of  
training data

From N7<sub>a</sub> to N7<sub>b</sub>

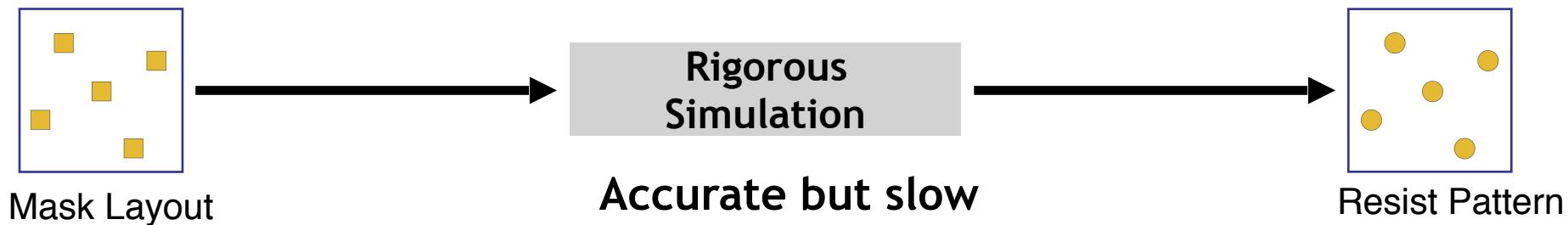


8~20X reduction of  
training data

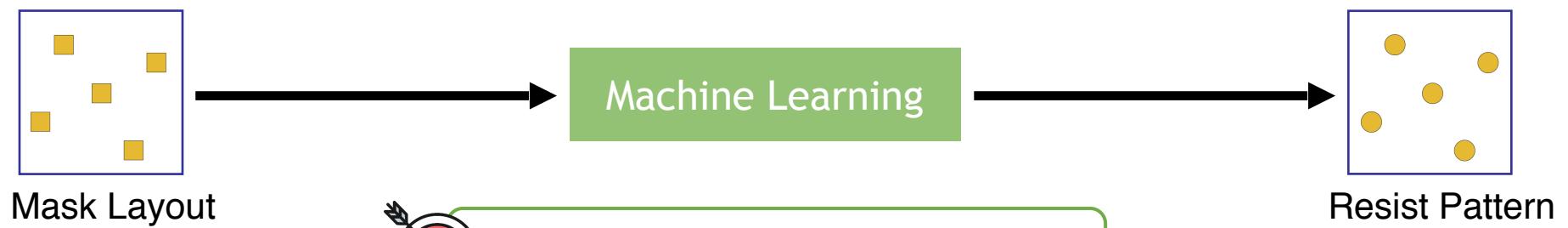


# End-to-End Lithography Modeling

Rigorous simulation: physics-based simulation, e.g., Synopsys S-Litho



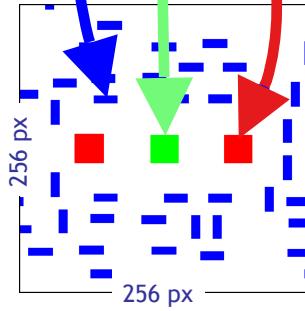
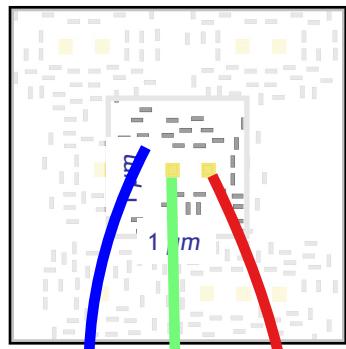
Machine learning for end-to-end lithography modeling



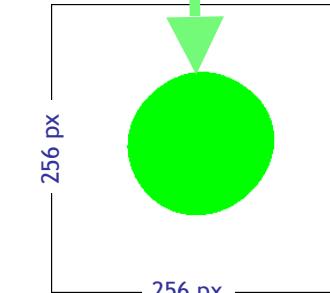
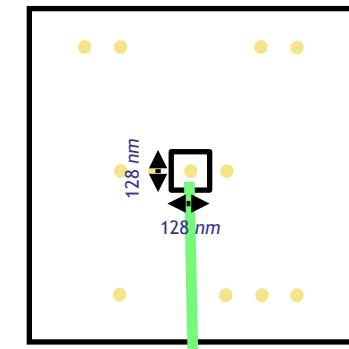
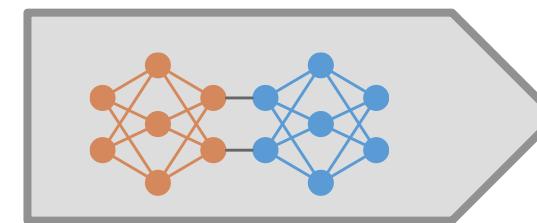
[[Ye+, DAC'19](#)] Best paper nomination



# Image Translation for Lithography Modeling



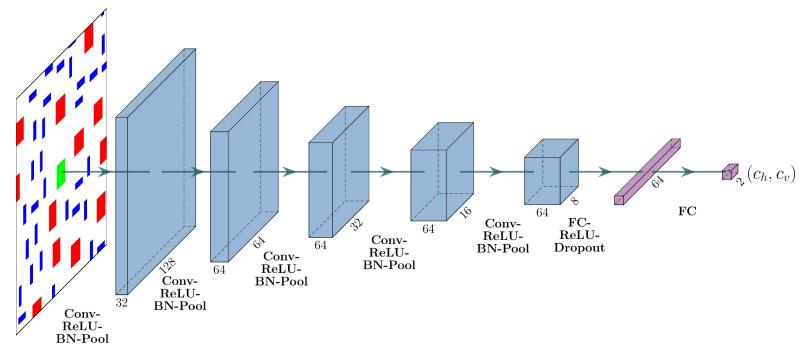
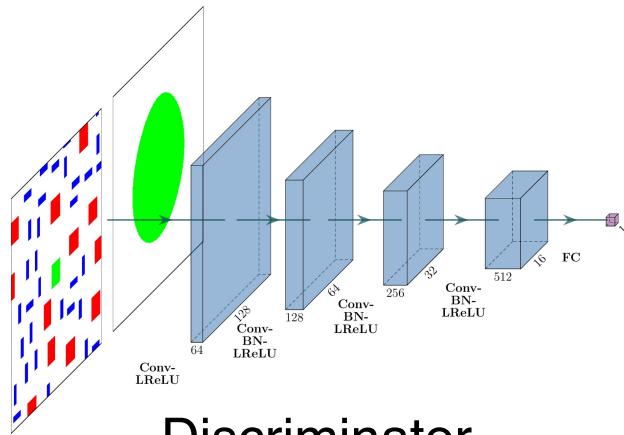
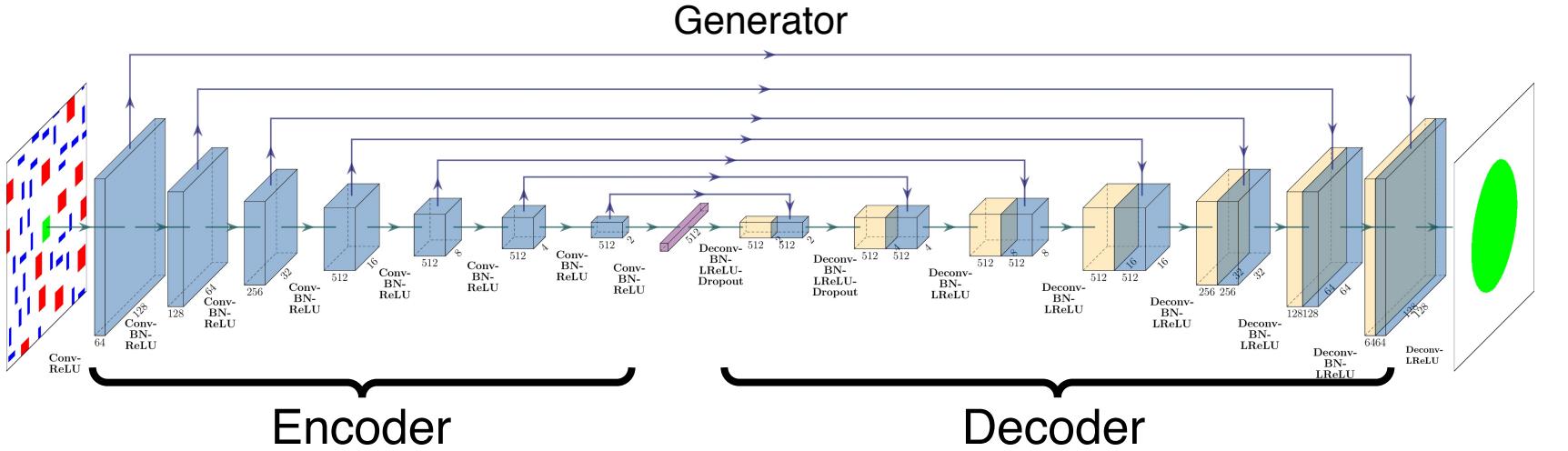
Different elements encoded on different image channels



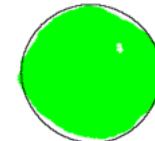
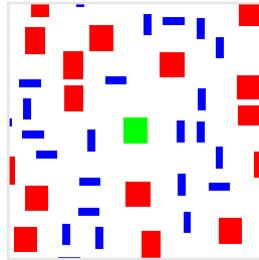
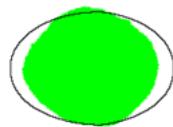
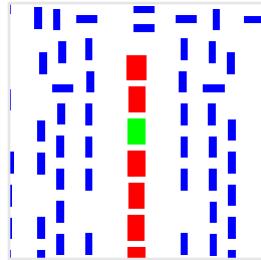
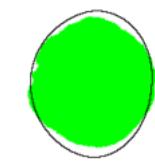
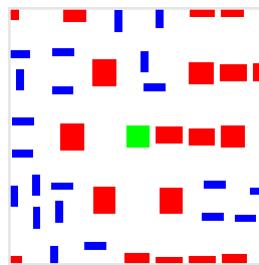
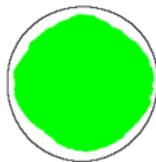
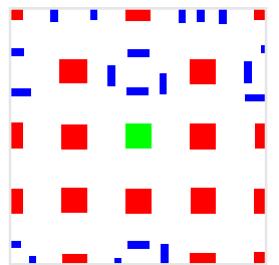
Resist pattern zoomed in for high-resolution/accuracy



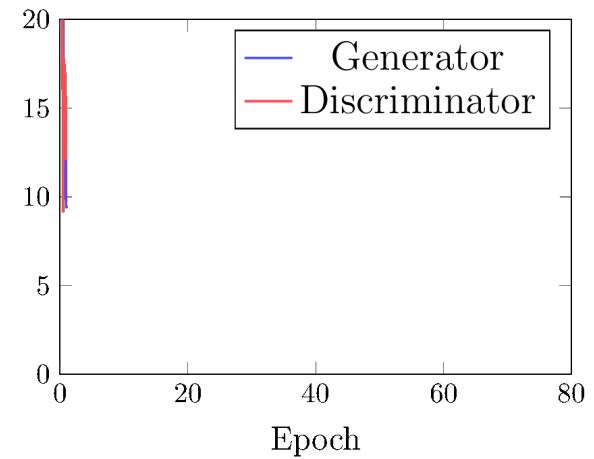
# LithoGAN Architecture



# LithoGAN Visualization



Loss



Model advancement progress



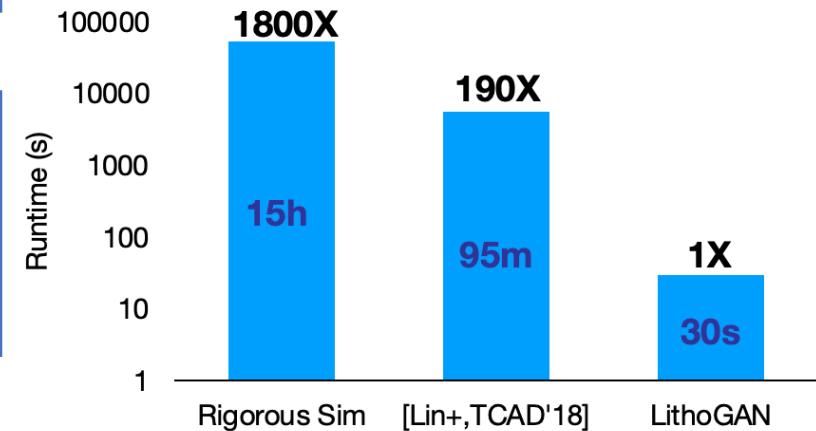
# Experimental Results

## Setup

- Python w/ TensorFlow
- 3.3GHz Intel i9 CPU & Nvidia TITAN Xp GPU

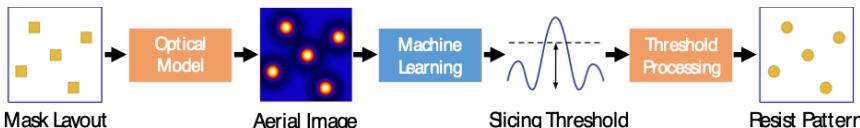
## Datasets

- Different types of contact arrays [\[Lin+, TCAD'18\]](#)
  - 982 mask clips at 10nm node (N10)
  - 979 mask clips at 7nm node (N7)
- 75-25 rule for train/test split



## Methods

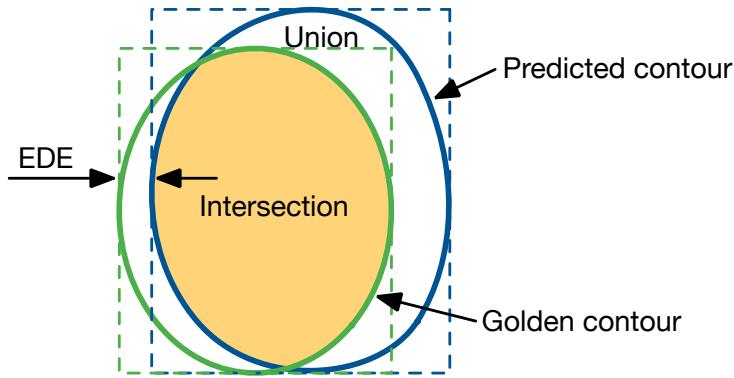
- Rigorous sim using S-Litho: golden patterns
- [\[Lin+, TCAD'18\]](#): Optical sim using Calibre + threshold prediction using CNN+post processing



**Compelling runtime speedup for early technology exploration**



# Experimental Results



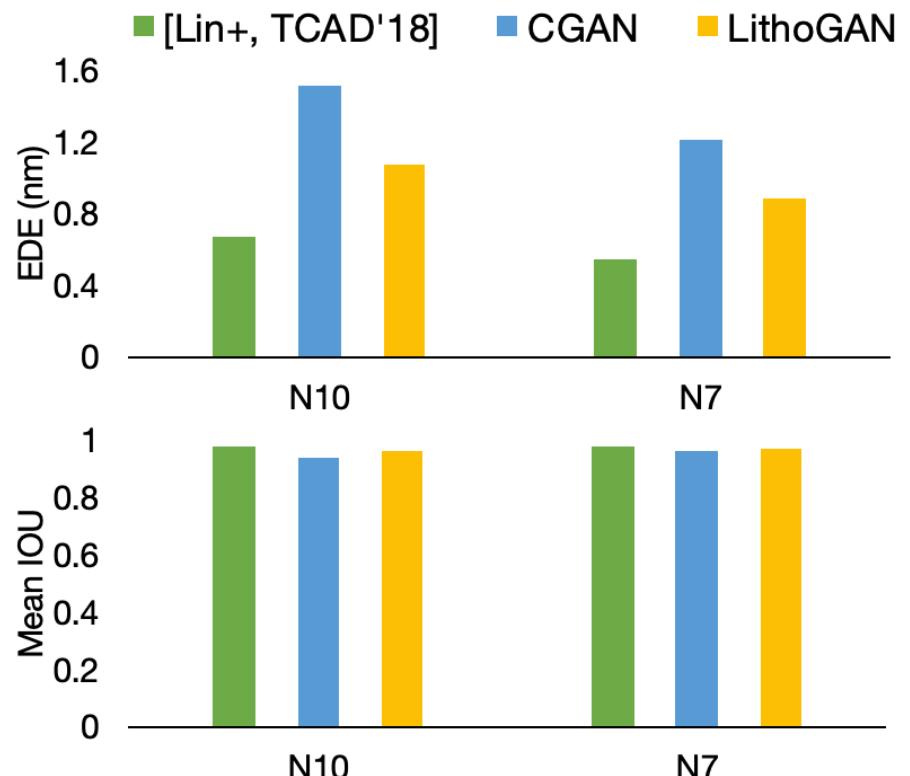
## Accuracy measures

### Edge Displacement Error (EDE)

- Distance between the golden edge and the predicted one of the bounding boxes
- The **smaller**, the **better**
- Captures bounding box mismatch

### IOU = Intersection/Union

- The **larger**, the **better**
- Captures contour mismatch



Competent accuracy for lithography usage (in consultation with industry)



# 总结

- Machine learning brings
  - New modeling opportunities
  - New optimization techniques
  - New hardware acceleration, e.g., GPU acceleration
  - New software platforms, e.g., Tensorflow, PyTorch
- Hammers and bridges for conventional EDA flow
  - Reformulate the problems, e.g., neural network training, image-to-image translation
  - Accurate and efficient information feedback from late stages to early stages



# Open Problems

- Connectivity feature representation
  - How to encode hypergraph as input to models
  - Or develop models that are friendly to ultra-large graphs
- Optimization-friendly ML models and ML-friendly optimization techniques
  - Target at integrating ML models into optimization
  - Need to consider fidelity, smoothness, accuracy, convergence rate
- Generalization guarantee
  - How far can ML models generalize
  - How to know whether a model is applicable to new data





# OpenBELT

## 开源 EDA 端到端框架的设想

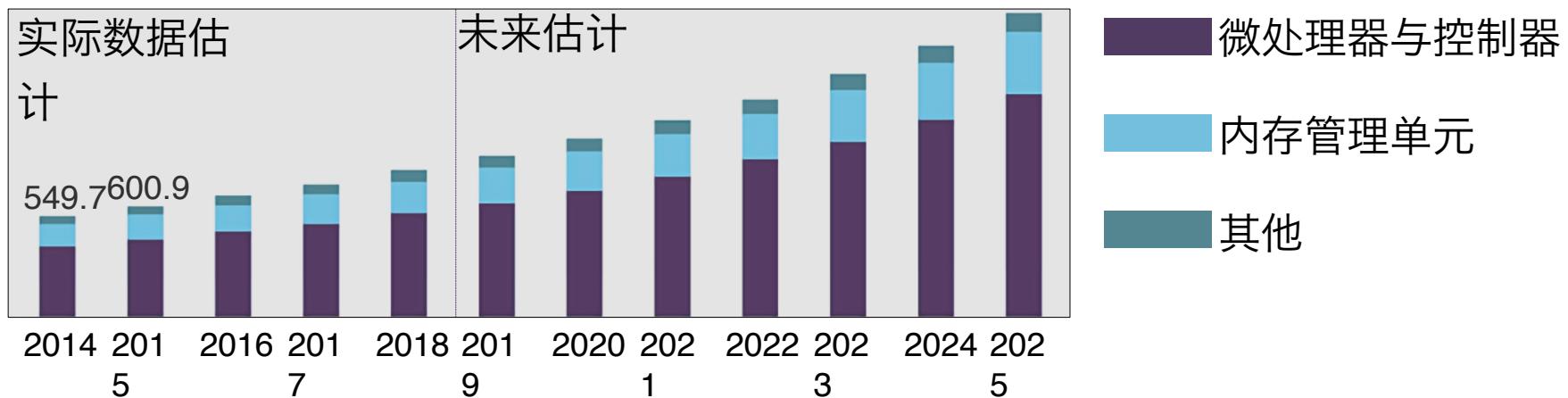
Credit to 罗国杰

北京大学高能效计算与应用中心

# 机遇与挑战：EDA的庞大国内市场

- 我国作为Synopsys和Cadence最大客户之一，学界与工业界广泛采用其EDA工具
- 我国每年需要花费数亿美元来购买EDA工具的使用许可

中国EDA工具市场份额 (million USD)



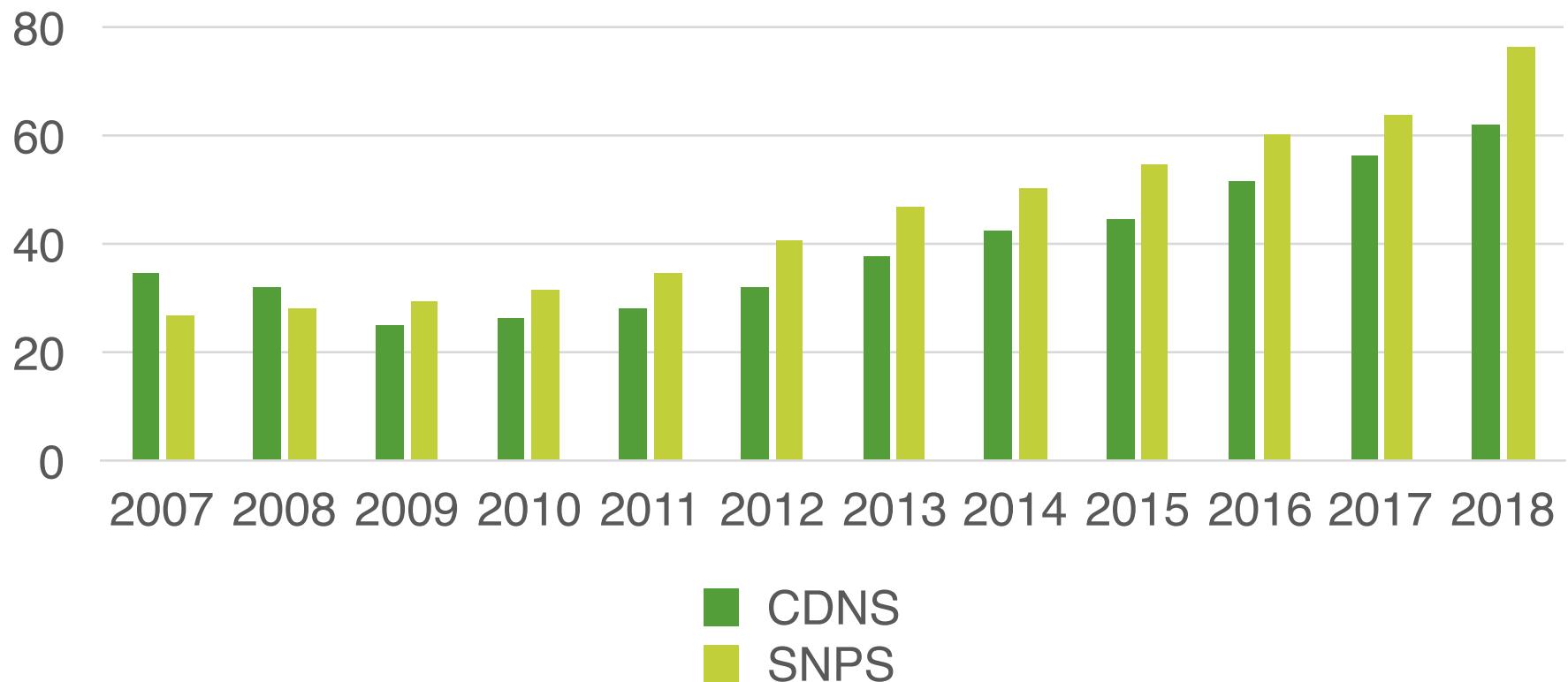
# 挑战：中国EDA资源差距

- 研发投入的巨大差距
  - 中国近十年的研发总投入  
≈ Synopsys三个月的研发投入
- 研发人数的巨大差距
  - 全球研发：~20000人
  - 中国研发：~2500人
    - 国际EDA公司：~1900人
    - 中国EDA公司：~600人
- 技术完整程度的差距
  - 中国EDA公司仅能供应占设计流程1/3的工具，且欠缺关键工具
- 芯片设计公司的支出差距2018年
  - 中国设计公司营业额占全球36.3%
  - EDA工具支出占比远低于其他地区：1.6% vs 7.4%



# EDA年度研发费用参考数据

年度研发费用（亿元人民币）



# 人力需求估算

## R&D工程师需求

- $9 \text{ 亿} / (15 \text{ 万} * 2) = 3000$
- 研发费用9亿USD每年
- R&D工程师15万USD年薪
  - glassdoor.com数据
- 在校研究生人数
  - $30 / (3/5) = 50$  博士生
  - IEEE TCAD论文容量
    - 每年  $180 = 15 \text{ 篇}/期 * 12 \text{ 期}$
    - 中国占1/6估算
  - 博士生5年发表3篇

如何高效地培养EDA工程师？



# Synopsys EDA 课程体系 (本科)

- 算法与结构化编程 Algorithms and Structural Programming
- 模拟集成电路 Analog Integrated Circuits
- 应用概率论 Applied Probability
- 数据结构 Data Structures
- EDA导论 EDA Introduction
- 硬件描述语言 Hardware Description Languages
- 集成电路设计导论 IC Design Introduction
- 算法导论 Introduction to Algorithms
- 线性代数 Linear Algebra
- 内存电路图设计基础 Memory Schematic Design Basics
- 数值方法 Numerical Methods
- 操作系统与系统编程 Operating Systems and System Programming
- 概率论与数理统计 Probability Theory and Mathematical Statistics
- 编程语言与编译器 Programming Languages and Compilers
- 技术写作 Technical Writing
- 算法理论 Theory of Algorithms
- Unix系统管理 Unix System Administration



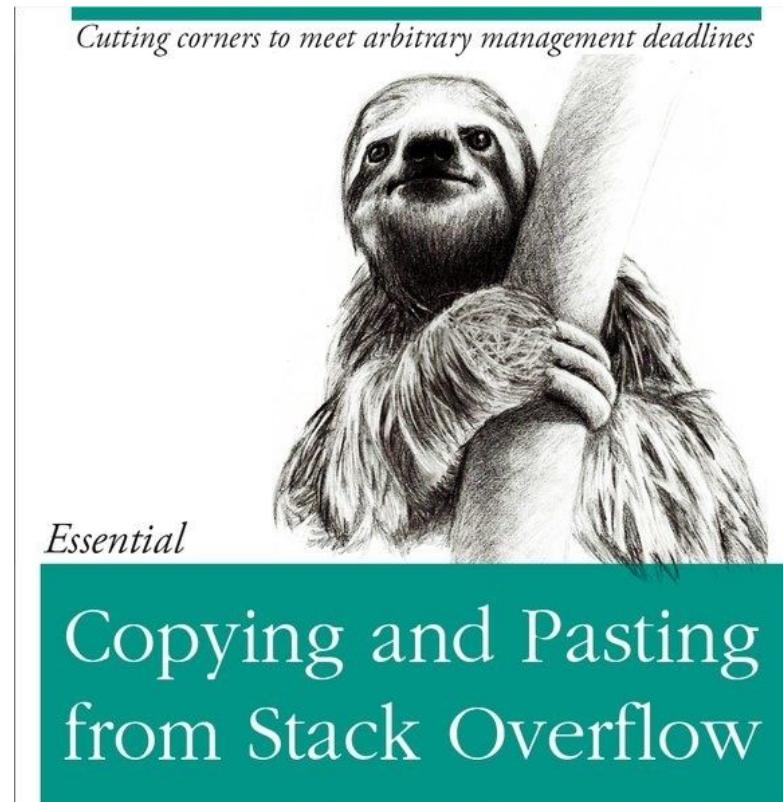
# Synopsys EDA 课程体系 (硕士)

- 编译器设计 Compilers Design
- 复变函数 Complex Functions
- 计算几何 Computational Geometry
- 计算机语言工程 Computer Language Engineering
- 当代软件开发套件 Contemporary Software Development Kits
- 数据库管理系统 Database Management System
- 数据库 Databases
- 程序语言设计 Design of Programming Languages
- 离散数学与概率学 Discrete Mathematics and Probability
- EDA数学方法 EDA Mathematical Methods
- 傅立叶变换 Fourier Transformations
- 模糊逻辑 Fuzzy Logic
- 集成电路设计算法 IC Design Algorithms
- 集成电路图设计算法 IC Schematic Design Algorithms
- 集成电路验证算法 IC Verification Algorithms
- 集成电路互连的建模与优化 Modeling and Optimization of IC Interconnects
- 面向对象编程 Object-Oriented Programming
- 运筹研究2 Operational Research 2
- C++编程 Programming C++
- 半导体器件与技术 Semiconductor Devices and Technology
- 软件开发技术 Software Development Technology



# 开源平台赋能开发者社区

- 通用开发者社区
  - stackoverflow.com
  - github.com
  - leetcode.com
- 非正式的技术讨论
  - arxiv.org
  - twitter.com
  - quora.com
  - reddit.com



O'REILLY®

The Practical Developer  
@ThePracticalDev



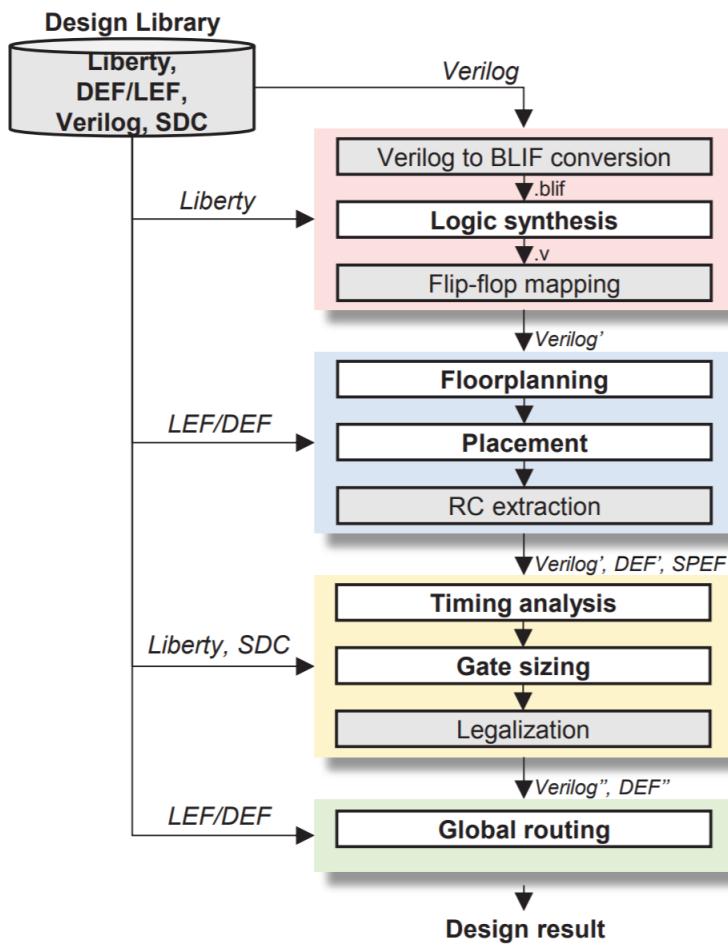
# 开源EDA端到端工具链

- 意义
  - EDA 开发者的教育和培训的规模化
  - 开放的 EDA 社区消除专业技术交流壁垒
  - 端到端的开源工具链吸纳社会资源
    - 科技经费：通过定制芯片开发，吸引自然科学和工程探索领域的部分经费
    - 校企合作：技术需求的表达，技术成果的评测和交付
    - VC 投资：降低小团队独立创业和生存的难度
- 难点
  - EDA 技能跨越算法层面和物理层面，涉及众多复杂专业知识
  - 开发完整的 EDA 工具链需要众多社区开发者的支持与维护
  - 运行大规模设计的 EDA 工具的需要大量的高性能服务器



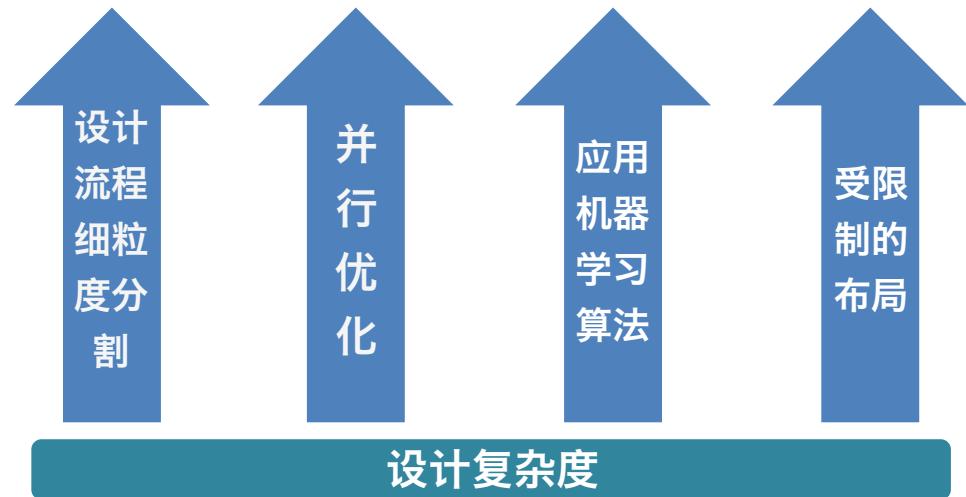
# 国际开源EDA项目

DATC项目：集成竞赛工具



OpenROAD项目

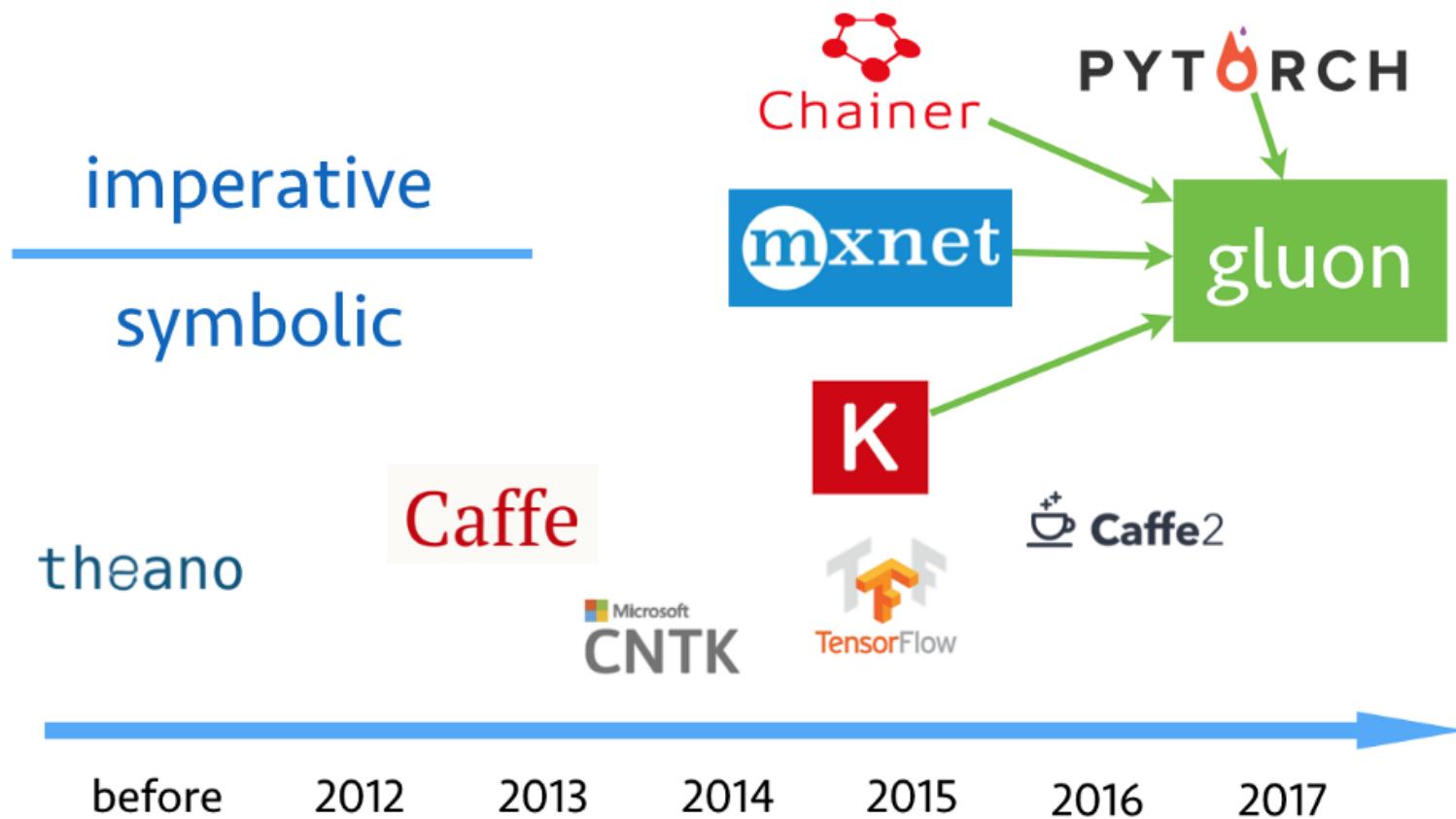
目标：24小时内完成，无需人工参与



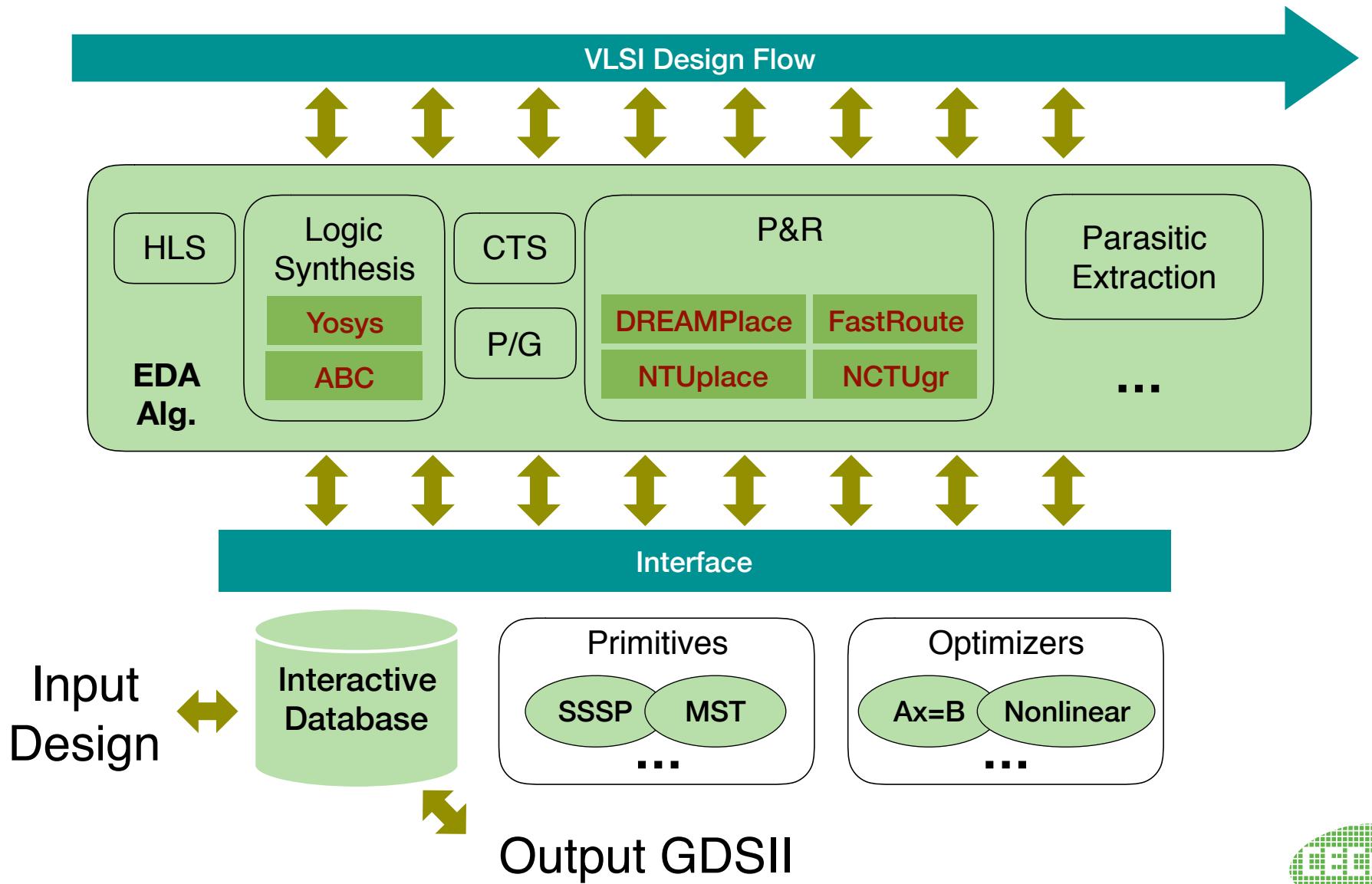
由UCSD大学的Andrew B. Kahng 领头，通过开源工具研究与开发，以实现美国国防高级研究计划局(DARPA)提出的电子设备智能设计(IDEA)计划的目标。



# 借鉴深度学习框架？

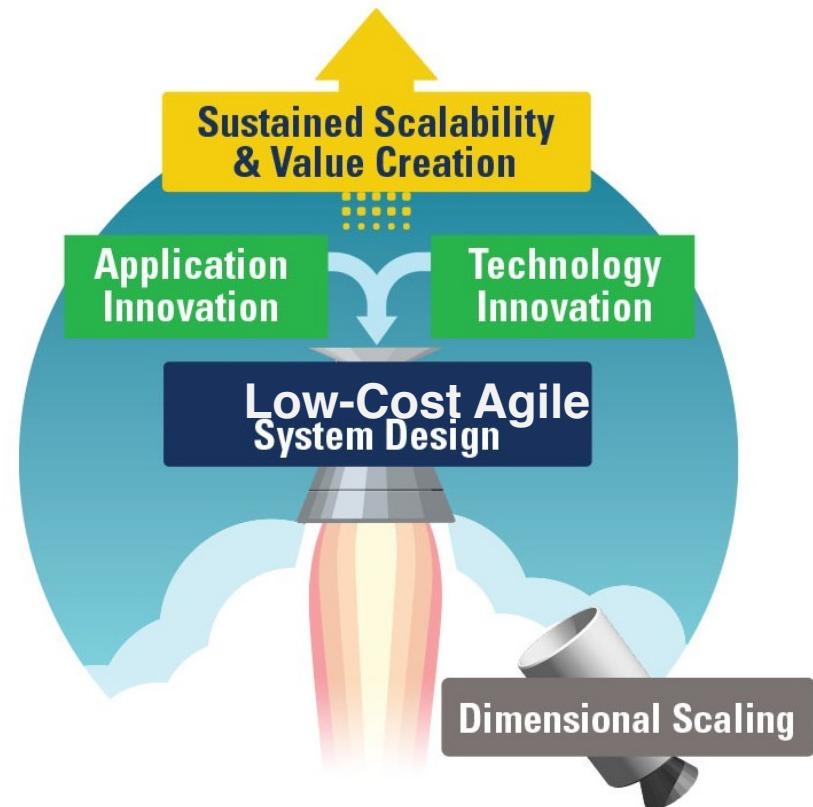


# OpenBELT Framework



# 后摩尔定律时代的半导体创新

- 价值创造依赖可持续创新
  - 以往是摩尔定律的尺寸缩放
- 应用创新
  - 为快速变化的应用带来价值
- 工艺创新
  - 为结构/材料/器件的进步带来价值
- 二者均需低成本的敏捷设计
  - 低成本的应用系统实现
  - 有效地连接应用和新工艺



Source: Todd Austin 2017



# 致谢

UT Austin  
David Pan教授团队

北京大学  
高能效计算与应用中心

香港中文大学  
余备教授团队

中科院微电子所  
韦亚一教授团队

Nvidia Research Lab  
Mark Ren  
Brucek Khailany

日本东芝存储  
MLG团队

Cadence  
Charles J Alpert团队

IMEC  
Peter Debacker团队





北京大学  
PEKING UNIVERSITY

谢谢！

# 自由与开源 (Free and Open-Source)

- 自由与开源软件 (FOSS)
  - Users have the freedom to
    - run,
    - copy,
    - distribute,
    - **study**,
    - change and
    - Improve
  - the software
  - <https://www.gnu.org/>
- 开源硬件
  - 开源硬件是可以通过公开渠道获得的硬件设计，任何人可以对已有的设计进行学习，修改，发布，制作和销售。
  - 硬件设计的源代码的特定的格式可以为其他人获得，以方便对其进行修改。
  - 理想情况下，开源硬件使用随处可得的电子元件和材料，标准的过程，开放的基础架构，无限制的内容和[开源的设计工具](#)，以最大化个人利用硬件的便利性。
  - 开源硬件提供人们在控制他们的技术自由的同时[共享知识](#)并鼓励硬件设计开放交流贸易。
  - <https://www.oshwa.org>

