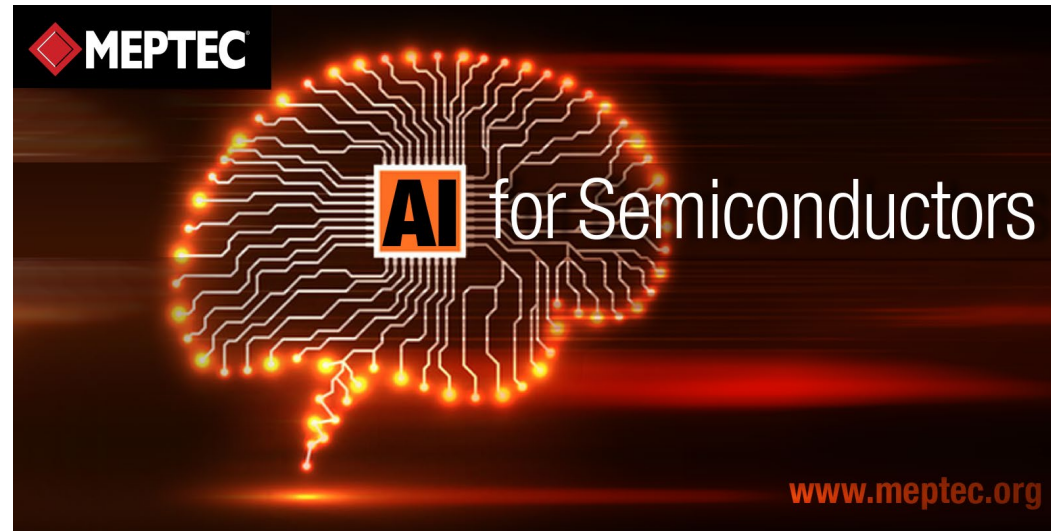




# The Role of Machine Learning (ML) in Electronic Design Automation (EDA)

Vidya A. Chhabria

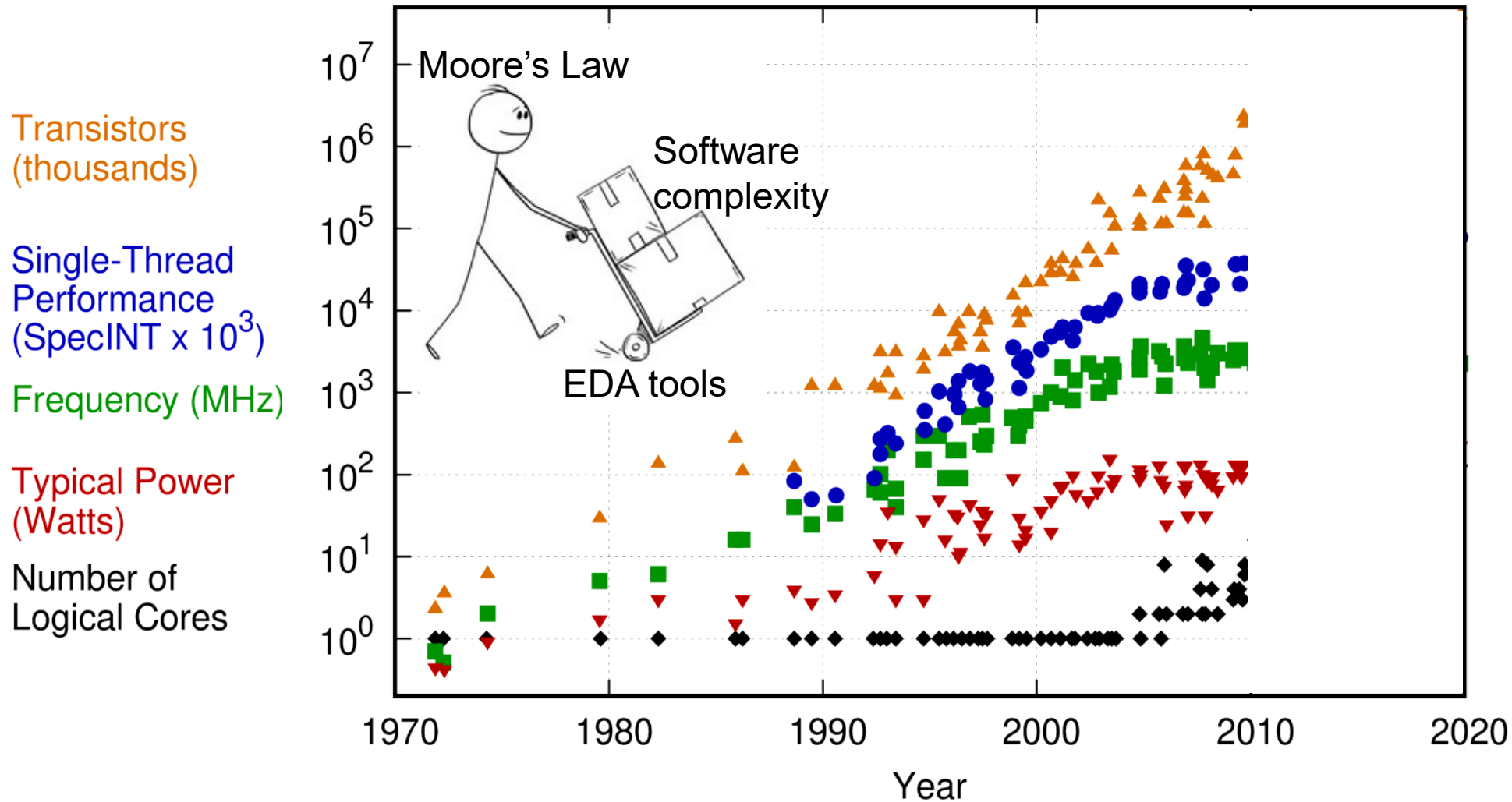
Electrical and Computer Engineering

Arizona State University (ASU)

8/9/2024

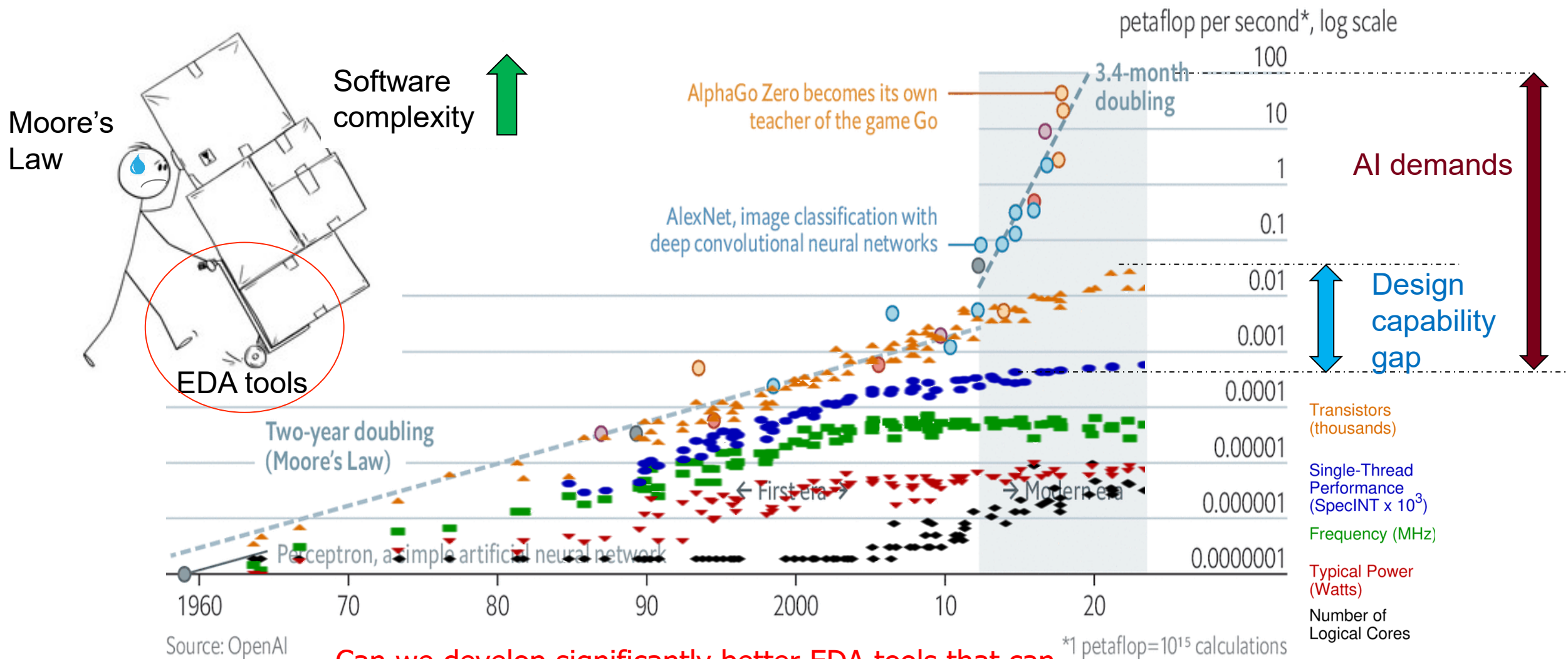


# Moore's Law and the Role of Electronic Design Automation (EDA)



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten  
New plot and data collected for 2010-2019 by K. Rupp

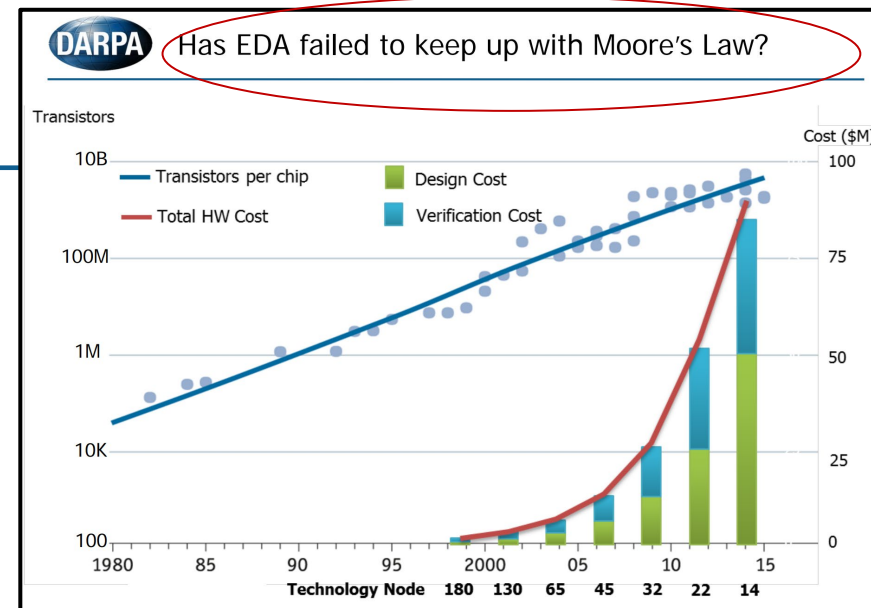
# Advent of AI and its Demands from Hardware



Can we develop significantly better EDA tools that can keep Moore's law afloat and meet software demands?

# Challenges in IC Design and EDA

- **Scale:** The problem of “N” in EDA
- **Time-to-market:** Computationally expensive EDA tools
- **Productivity:** Sub-optimal automated solutions require tremendous manual intervention for high-quality
- **Cost:** Design dollar costs, risk, and expertise are barriers

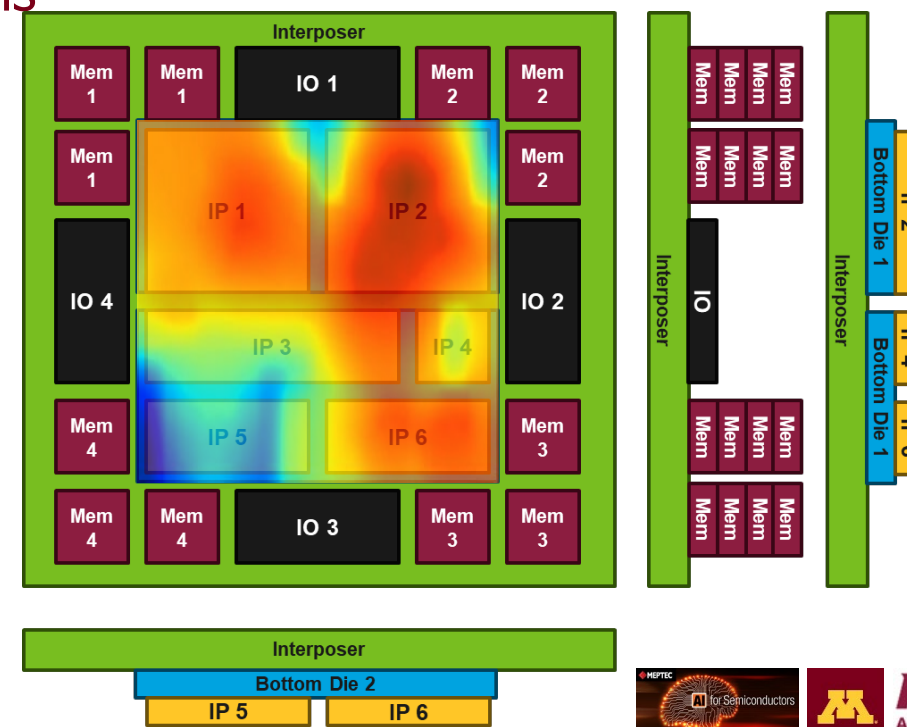


[A. Olofsson, DARPA, ISPD 2018]

## Challenges exacerbated by 2.5D and 3D chiplet-based systems

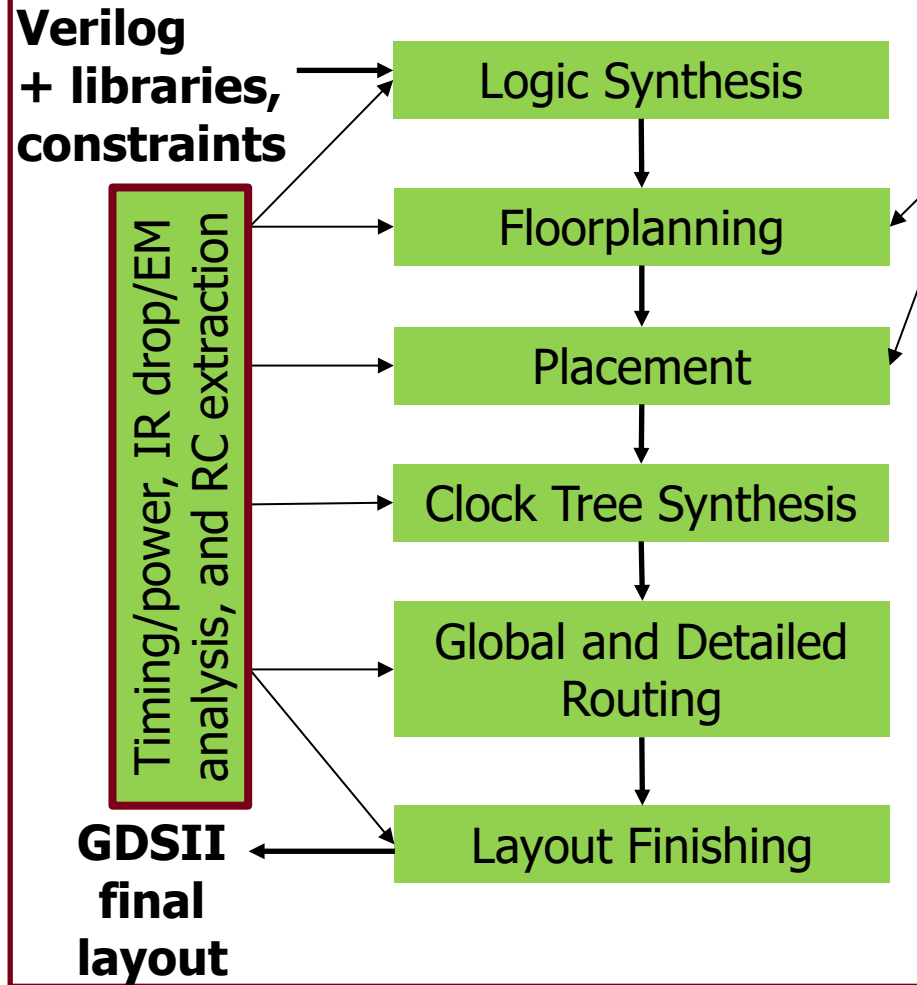
- Scaling Multiphysics solvers across chiplets is impractical
- Stacked chiplets are prone to thermal issues due to the longer path to thermal ambient
- Power delivery is critical to ensure that the chiplets do not suffer from high IR drops

Our research aims to address these challenges through **ML-based** and **open-source** EDA tools.



# Agenda

IC Design Cycle:



**24 hours, no humans – no power, performance, area loss**

**Optimize:** On-chip power delivery networks (PDNs)

**Predict:** IR drop, on-chip temperature, EM hotspots

**Generate:** Benchmark generation for PDNs

**Assist:** EDA chatbot for EDA tools

Machine Learning in tools,  
flows

Machine Learning for EDA  
infrastructure

**Software:** Python APIs for EDA tools (OpenROAD)

**Schema:** CircuitOps ML EDA data format

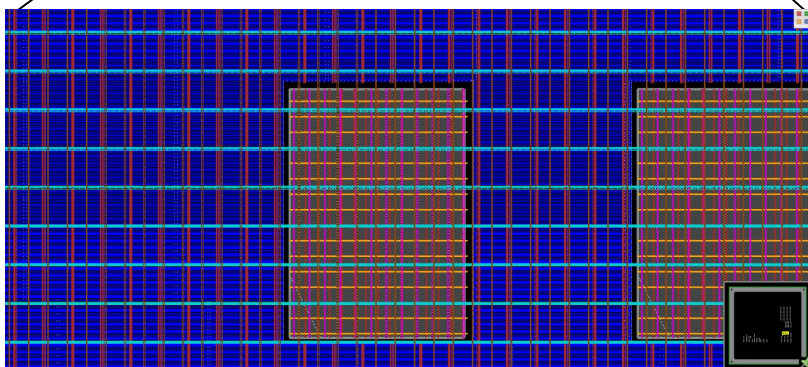
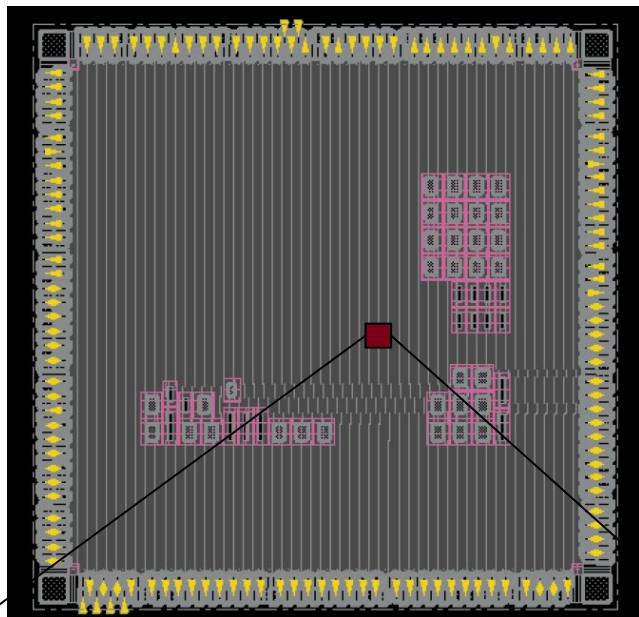
**Infrastructure:** SLICE common infra for EDA

**Datasets:** EDA Corpus: LLM dataset for EDA tool

**Design Complexity**

# "Optimize": On-chip Power Delivery Networks (PDNs)

12nm FinFET RISC V Core 53 macros



IR drop constraints

$$V_{cell} > V_{limit}$$

Typical limits:

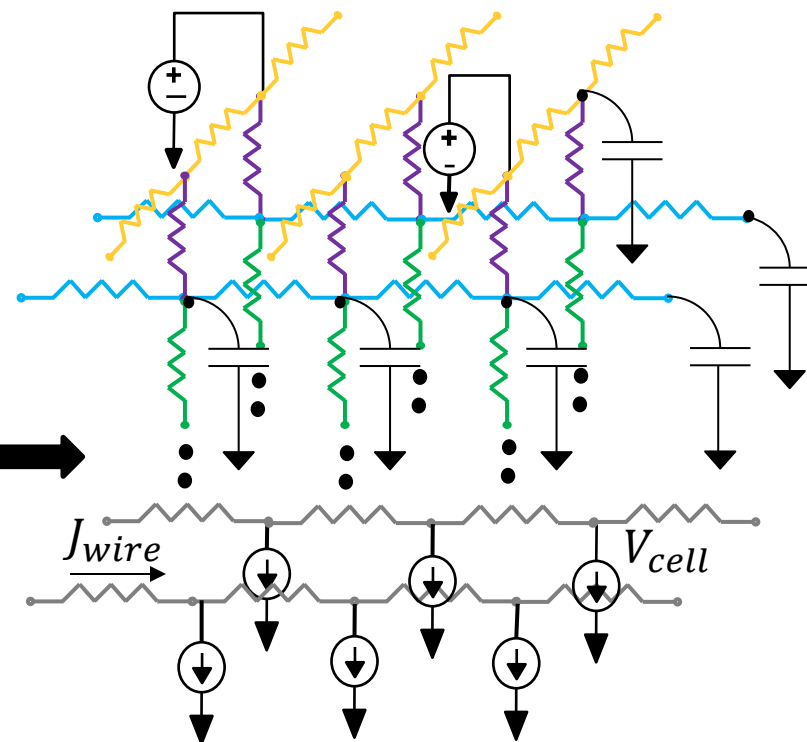
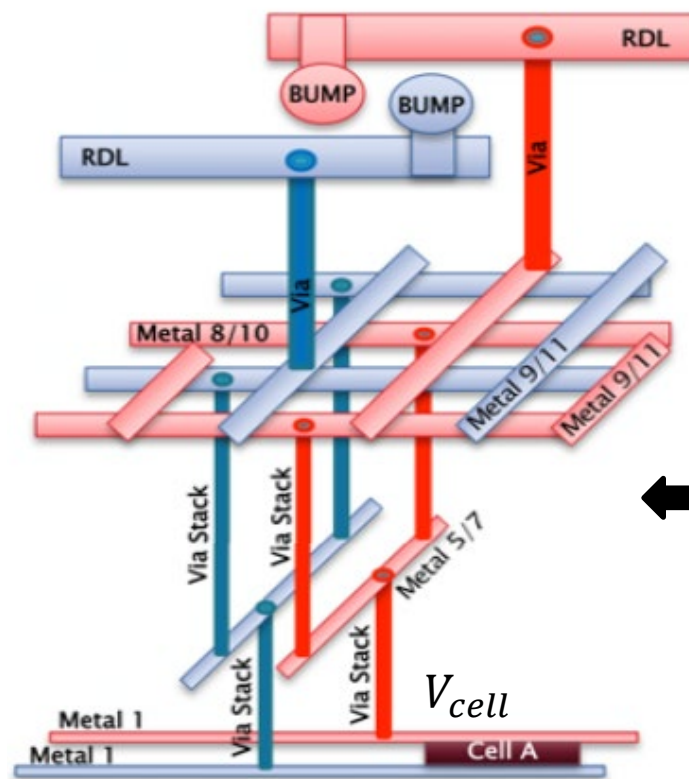
$$5\% \text{ of } V_{dd}$$

Electromigration (EM) constraints

$$J_{wire} < J_{limit}$$

Typical limits:

Layer- and tech-dependent

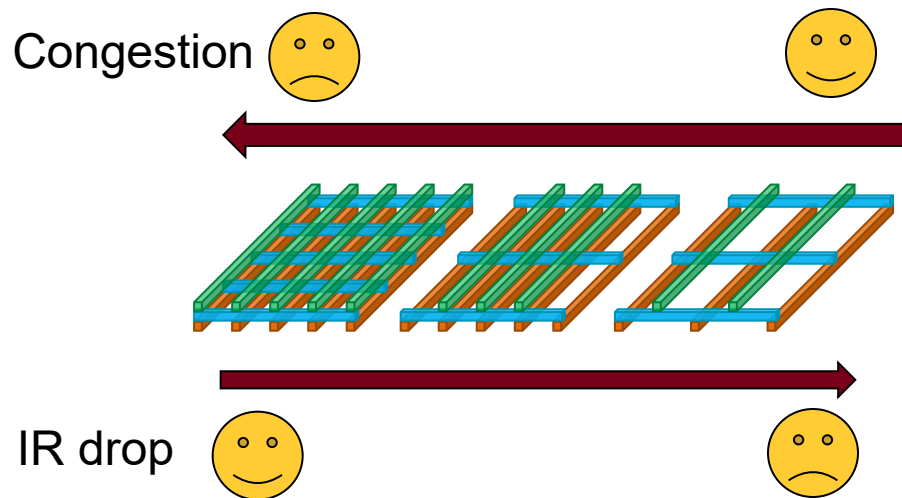


[Chhabria *et al.*, "OpenPDN: A Neural-Network-Based Framework for Power Delivery Network Synthesis," TCAD21]

# "Optimize": Conventional PDN Design and its Challenges

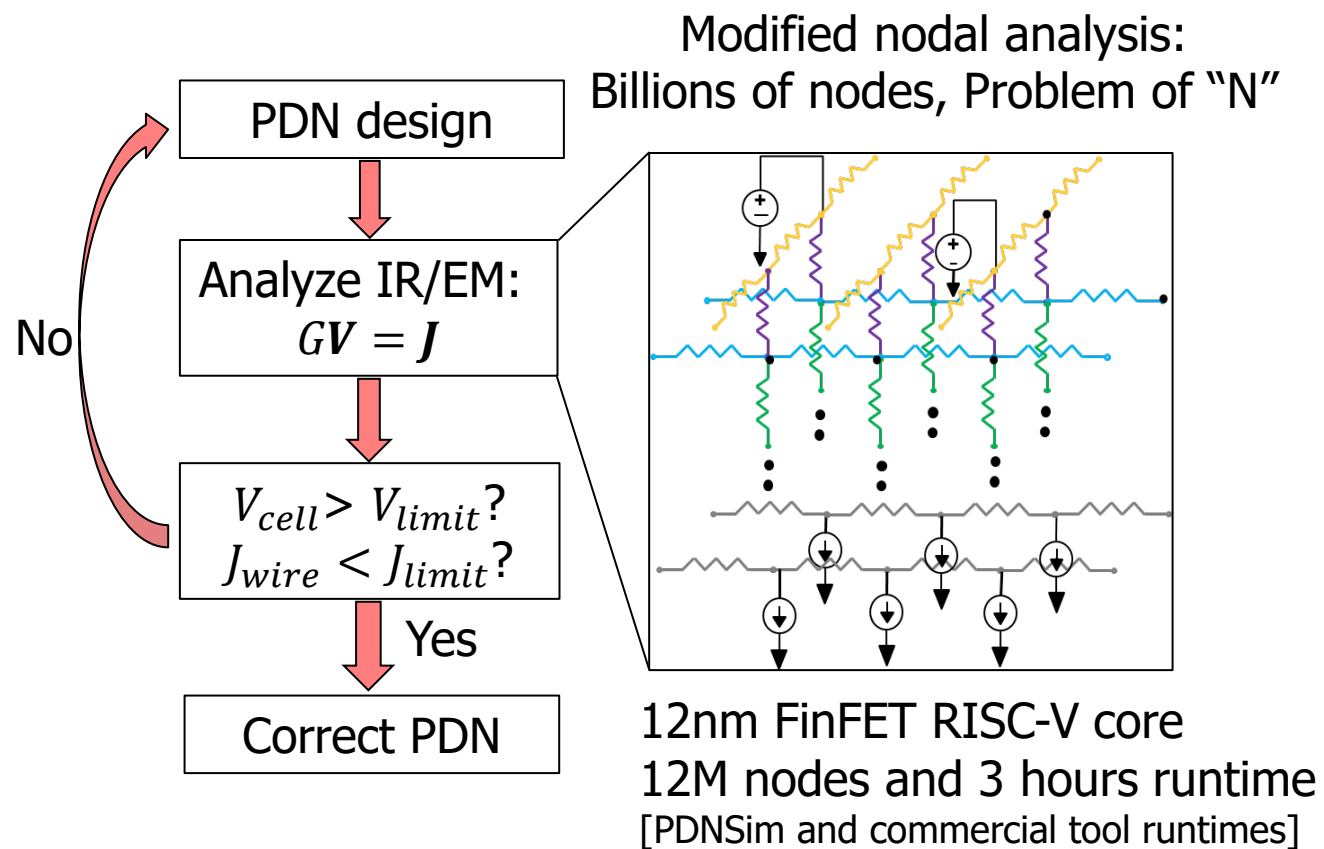
- Challenge #1:**

Trade off between IR drop and congestion



- Challenge #2:**

Optimization is iterative in nature and expensive

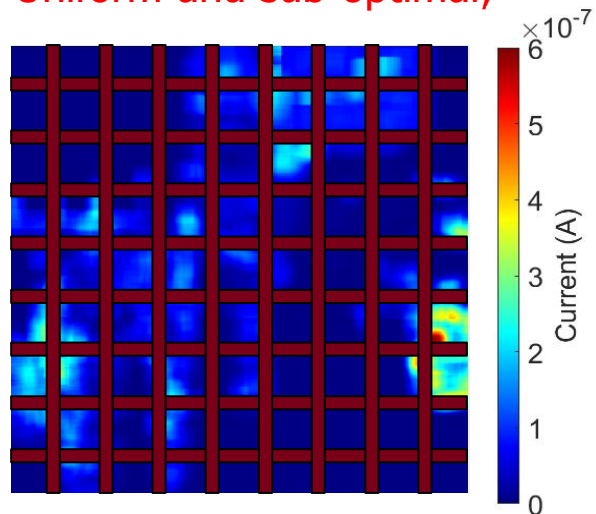


# "Optimize": Conventional PDN Design and its Challenges

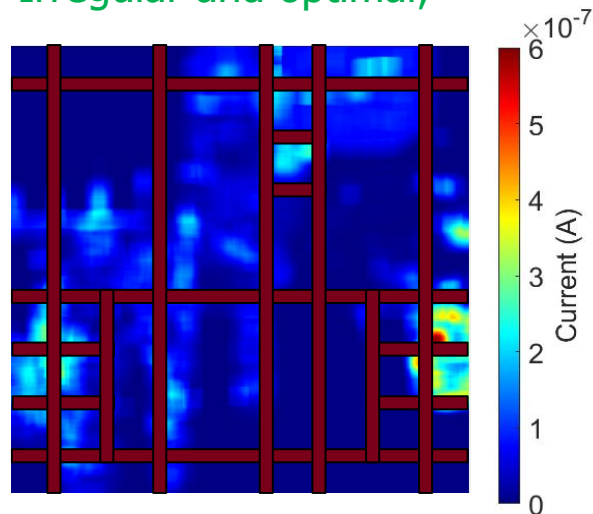
- Challenge #3:**

Uniform dense grids are sub-optimal and over designed

Meets IR/EM constraints;  
Too many routing resources;  
**Uniform and sub-optimal;**

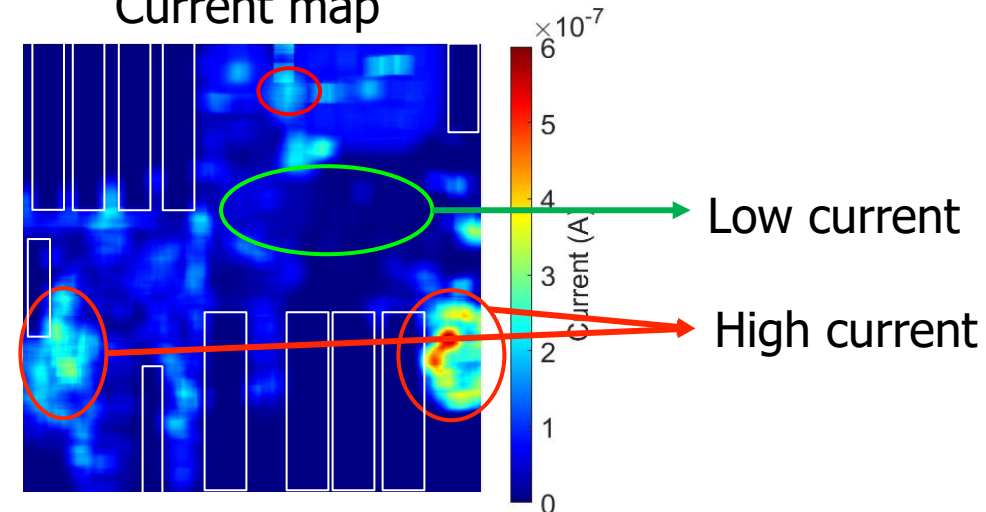


Meets IR/EM constraints;  
Few routing resources;  
**Irregular and optimal;**

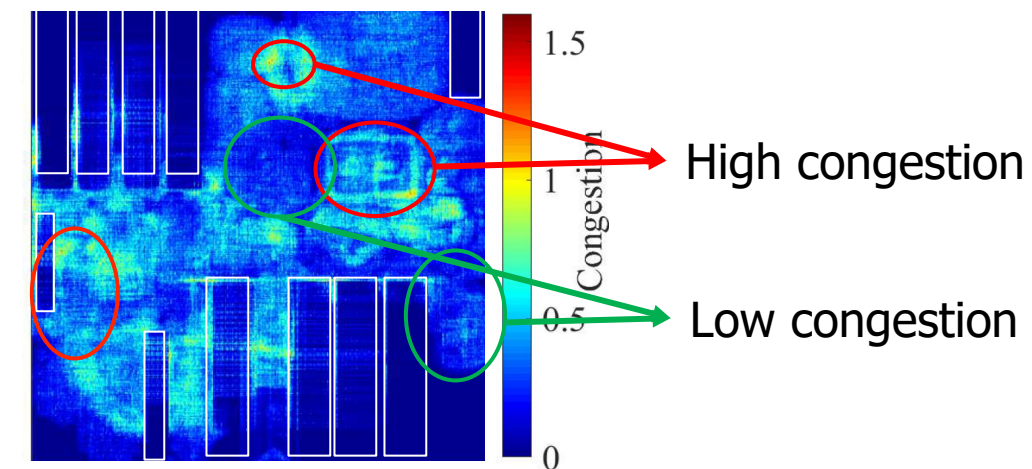


Can we automatically design an optimized power grid that meets IR/EM constraint with least routing resources?

Current map

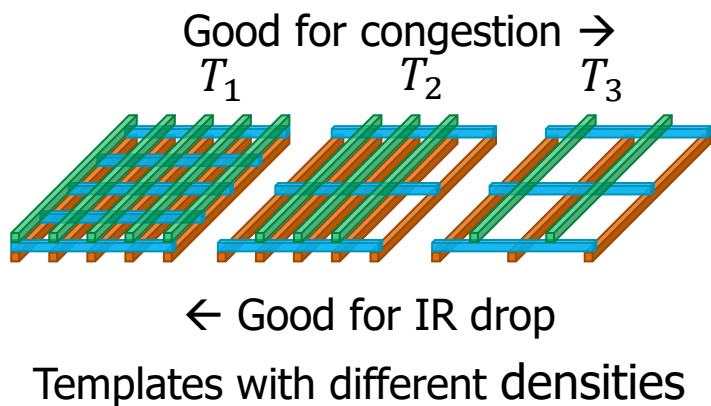


Congestion map

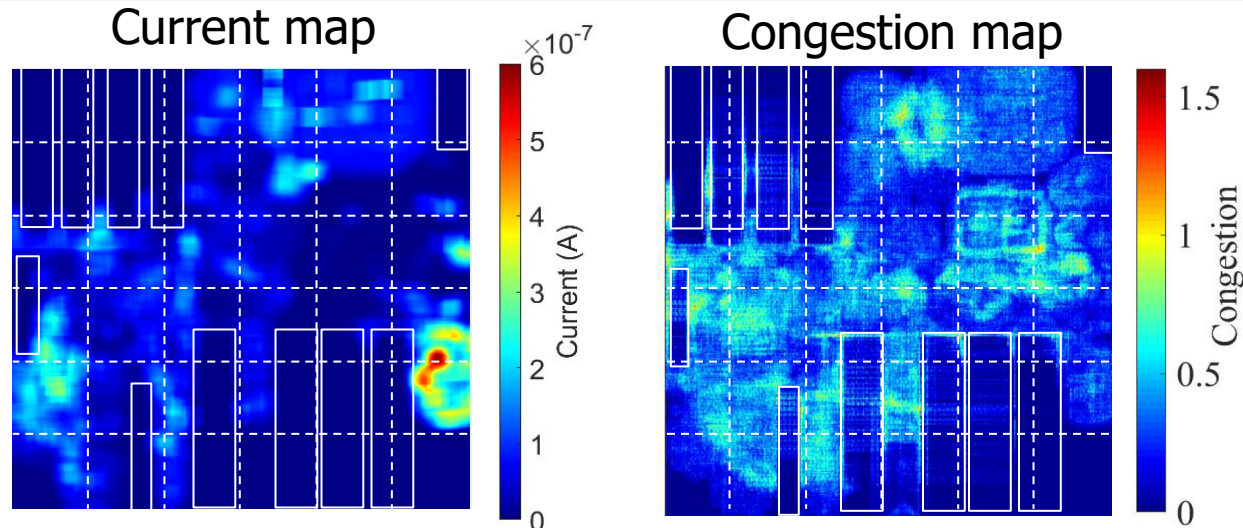


# "Optimize": OpenNPDN Template-driven PDN Synthesis

- Template-based PDN synthesis
  - PDN building blocks
  - Pre-defined, PDK-specific templates
  - Piecewise regular PDN
  - Must be stitchable



- "Which template goes where?"



3 templates per region:  $3^{36} = 1.5e17$

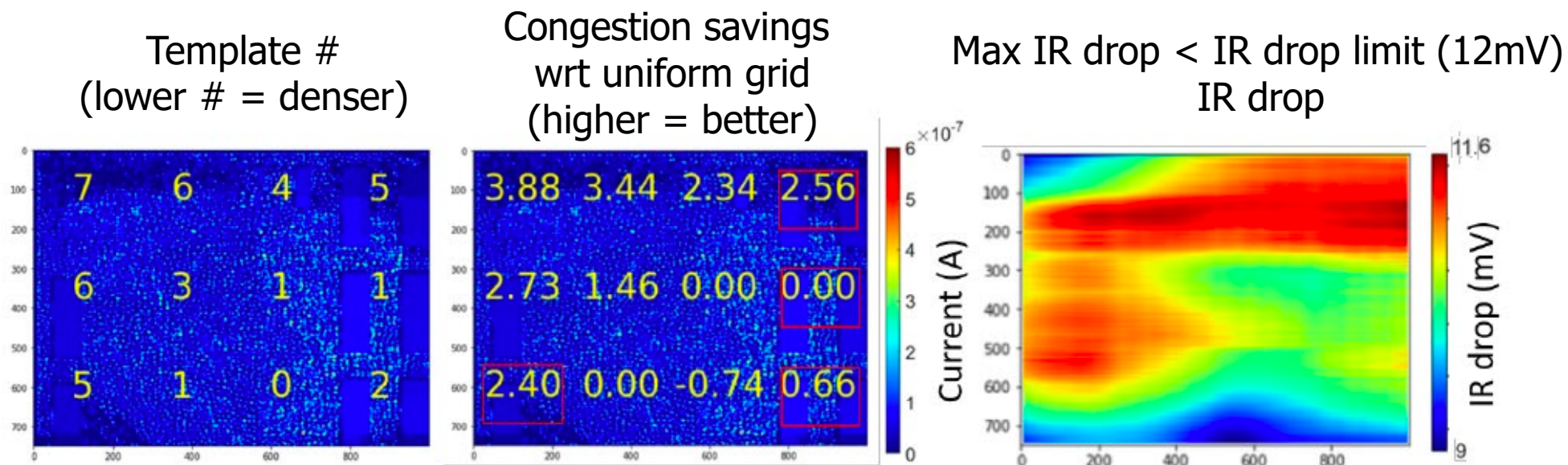
ML can help

EDA-world problem to ML-world problem mapping:

- **Features:** Images or distribution of current and congestion in the regions
- **Class:** Template ID
- **CNNs:** Image classification

# "Optimize": Template-driven PDN Synthesis Results

Post-placement

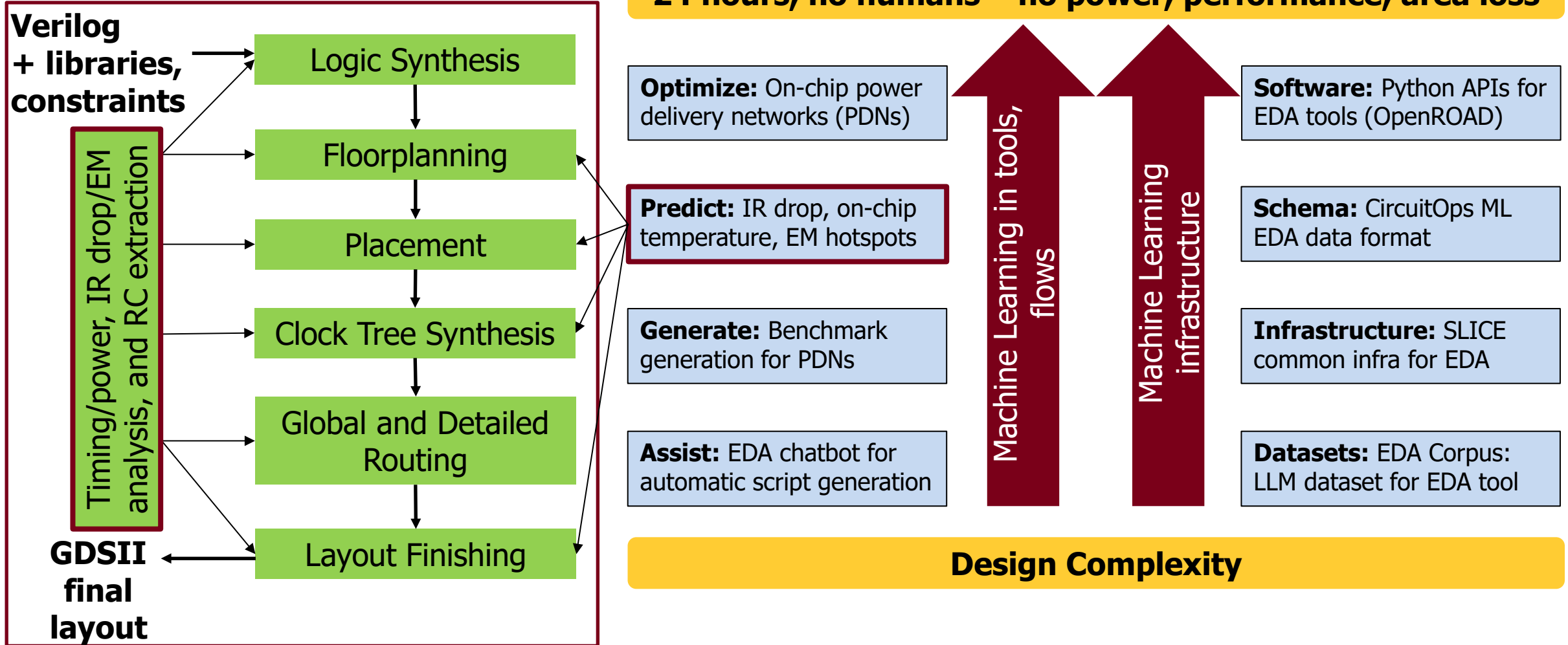


Evaluated on 12LP, 65LP, designs with ~50K – ~500K cells. Execution time: < 3m, dominated by feature extraction

<https://github.com/The-OpenROAD-Project/OpeNPDN>

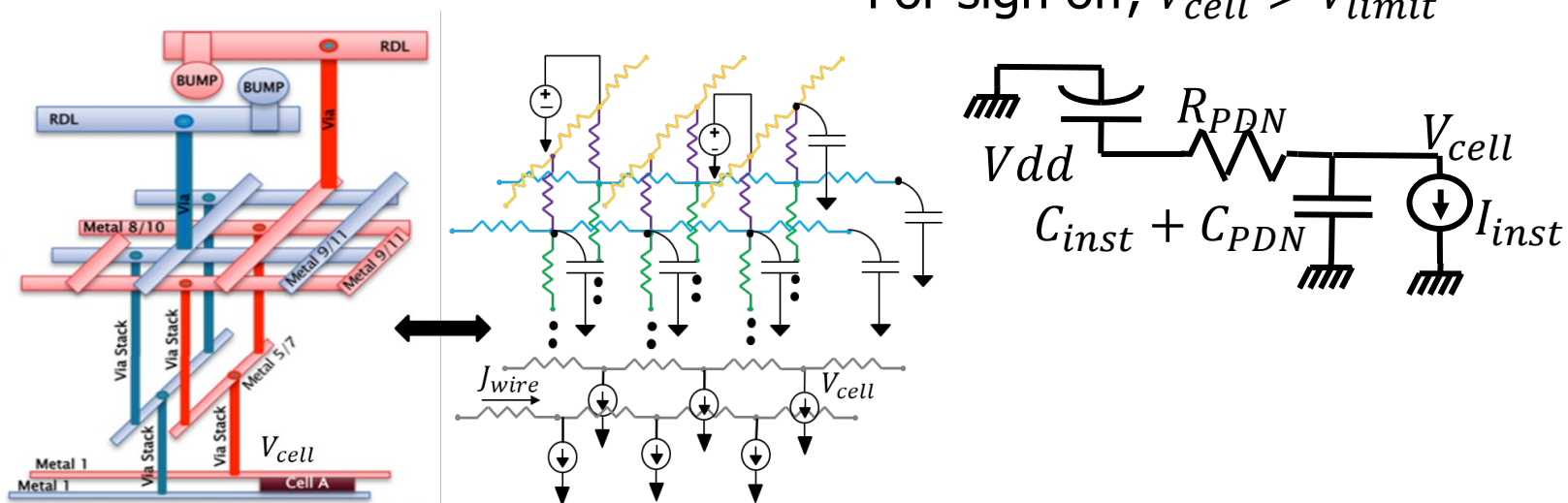
# Agenda

IC Design Cycle:



# "Predict": Thermal and Power Delivery Network (PDN) Analysis

## PDN analysis



Solution in steady state:

$$GV = J$$

System of equations with **billions** of variables!

## Thermal analysis

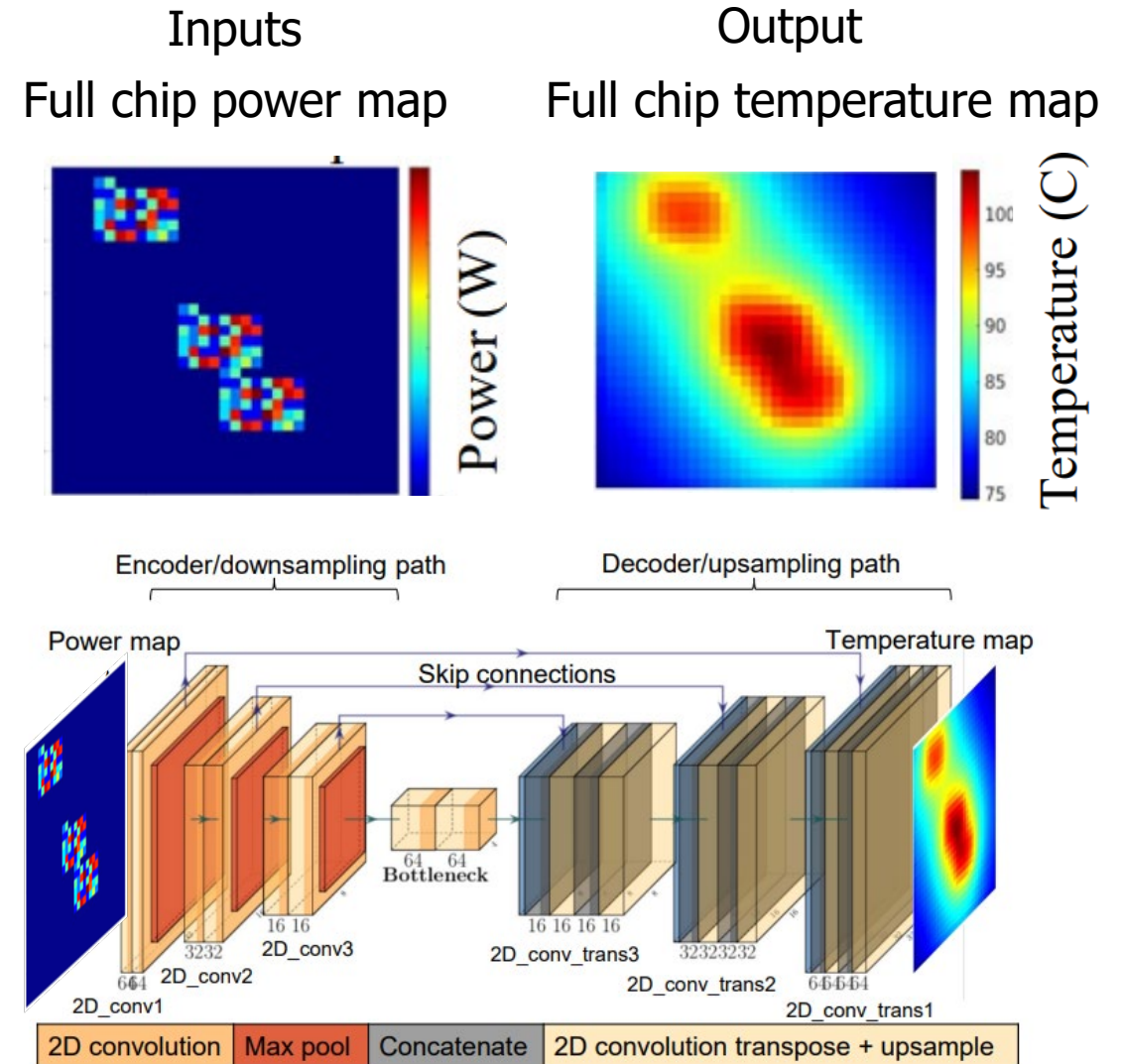
Heat flow:

$$C \frac{\partial T}{\partial t} = K(x, y, z) \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + P(x, y, z, t) \right)$$

Computationally expensive with hours of runtime on industrial size designs

# "Predict": Thermal and IR drop analysis

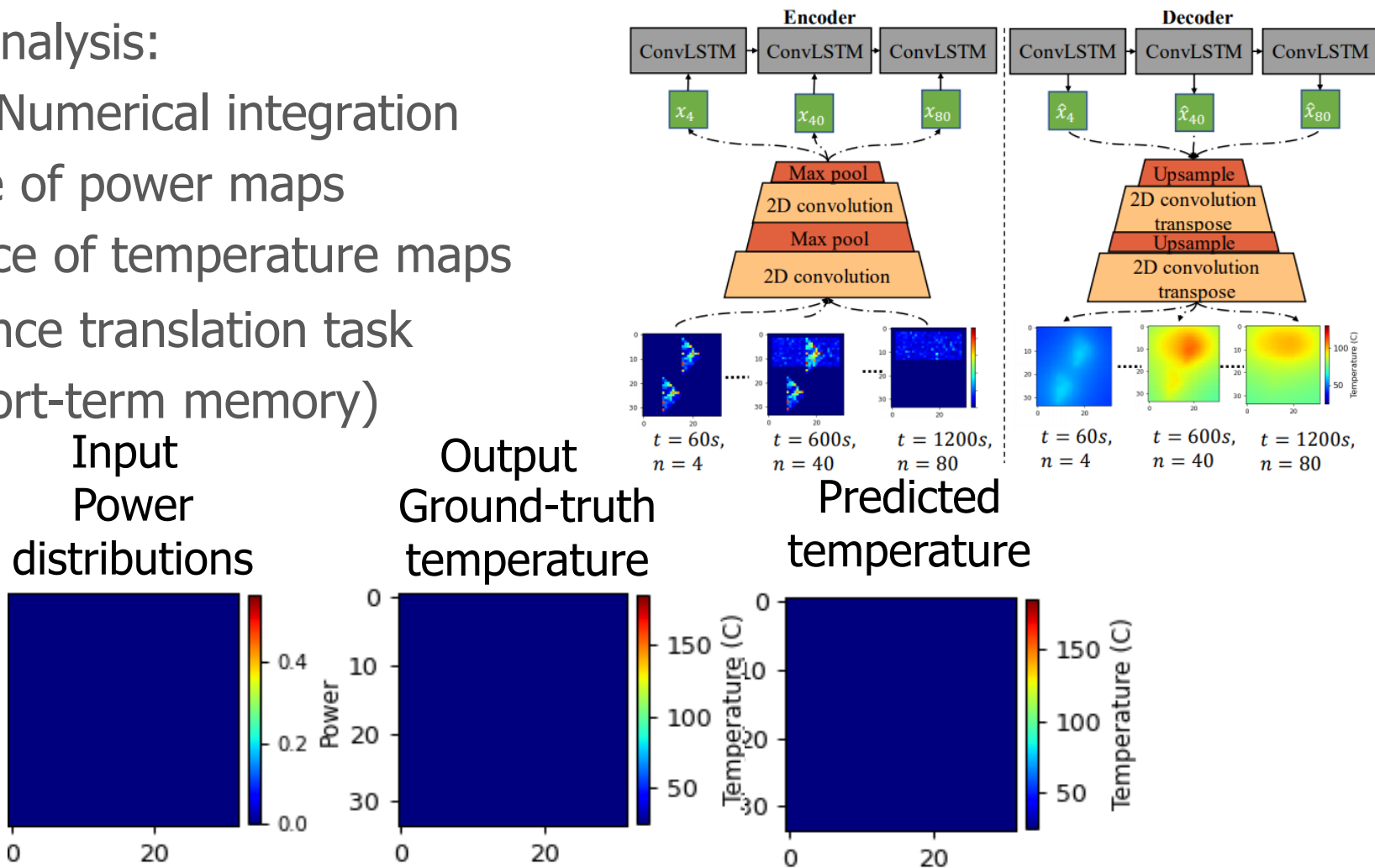
- Static thermal analysis:
  - Input: Power map
  - Output: Temperature map
- Image-to-image translation task
  - U-Nets
  - Chip-size independent
  - Transferable across different designs in the same technology



[Chhabria *et al.*, "Thermal and IR Drop Analysis Using Convolutional Encoder-Decoder Networks" ASP-DAC'21]

# “Predict”: Thermal and IR drop analysis

- Transient thermal analysis:
  - Conventionally: Numerical integration
  - Input: Sequence of power maps
  - Output: Sequence of temperature maps
- Sequence-to-sequence translation task
  - LSTMs (Long short-term memory)

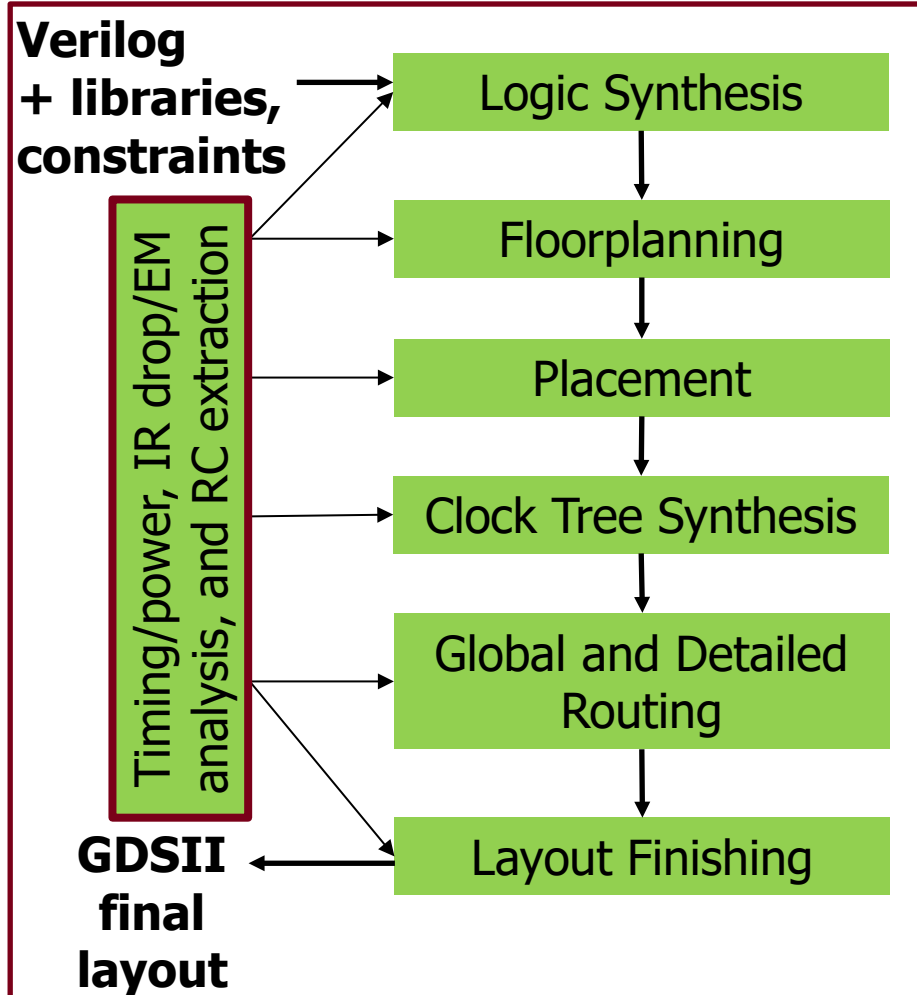


Hours of runtime down to few milliseconds!

[Chhabria *et al.*, “Thermal and IR Drop Analysis Using Convolutional Encoder-Decoder Networks” ASP-DAC’21]

# Agenda

IC Design Cycle:



**24 hours, no humans – no power, performance, area loss**

**Optimize:** On-chip power delivery networks (PDNs)

**Predict:** IR drop, on-chip temperature, EM hotspots

**Generate:** Benchmark generation for PDNs

**Assist:** EDA chatbot for automatic script generation

**Software:** Python APIs for EDA tools (OpenROAD)

**Schema:** CircuitOps ML EDA data format

**Infrastructure:** SLICE common infra for EDA

**Datasets:** EDA Corpus: LLM dataset for EDA tool

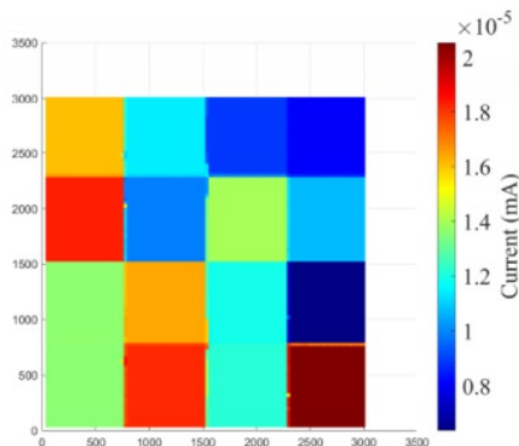
Machine Learning in tools,  
flows

Machine Learning  
infrastructure

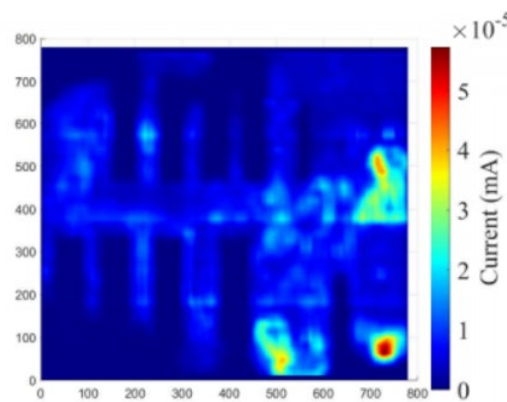
**Design Complexity**

# "Generate": Power delivery network (PDN) benchmarks

- Why are chip datasets and benchmarks so hard to find?
  - Technology and design intellectual property
  - Rapidly changing technologies with scaling
  - Requires laborious volunteer effort



IBM benchmarks



RISC-V core

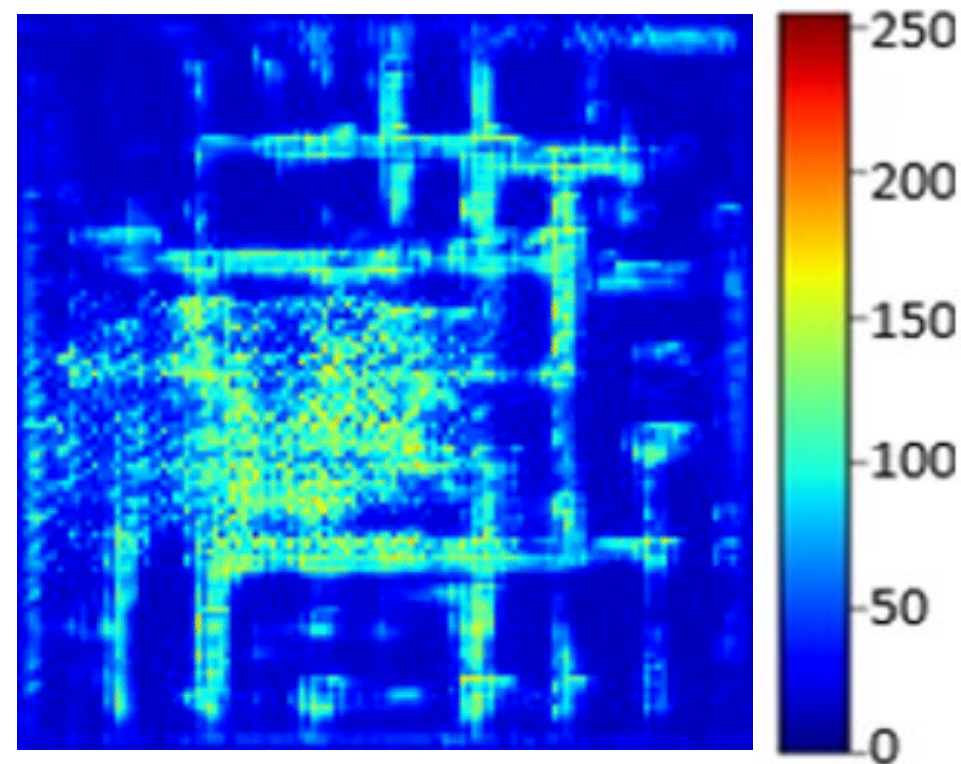
| Criteria      | Existing benchmarks | Wish list                     |
|---------------|---------------------|-------------------------------|
| Technology    | × 180 nm            | ✓ Portable                    |
| Realistic     | × Unrealistic       | ✓ Correlated to real circuits |
| IR drop       | × 10-40% of VDD     | ✓ 0.5-5% of VDD               |
| Diversity     | × Limited diversity | ✓ Large diversity             |
| #Count        | × <10               | ✓ >1000                       |
| IP protection | ✓ Yes               | ✓ Yes                         |

# "Generate": Synthetic PDN Data Generation

GANs for synthetic image data generation

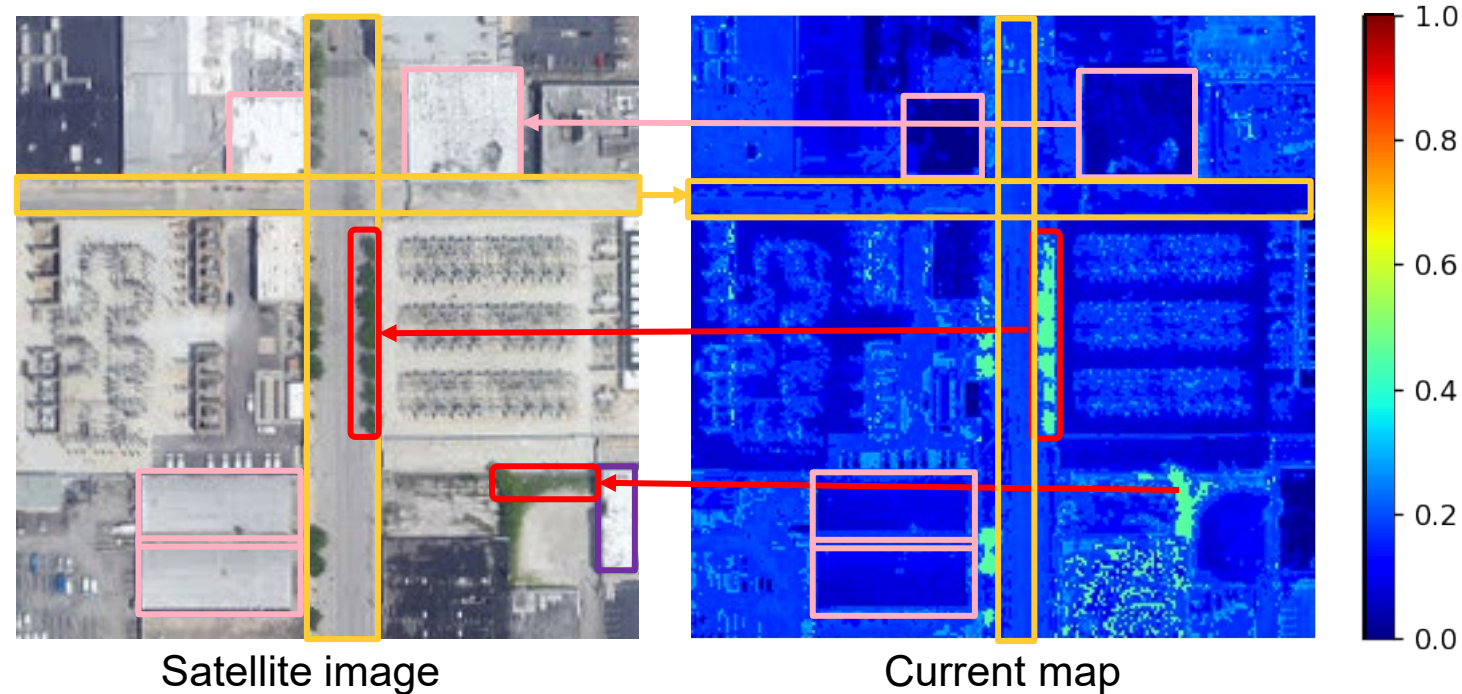
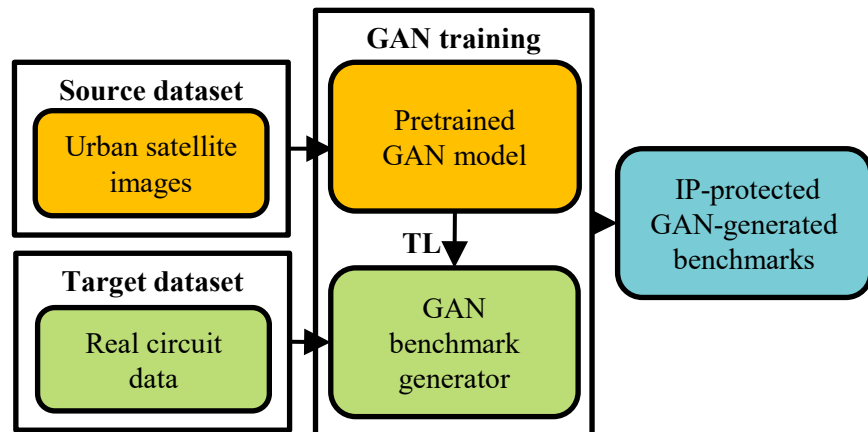


BeGAN for synthetic current map (CM) generation

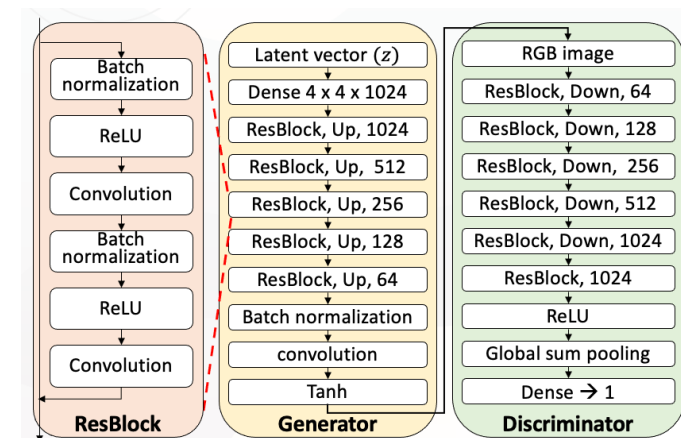


[Chhabria, Kunal, Zabihi, Sapatnekar, "BeGAN: Power Grid Benchmark Generation Using a Process-portable GAN-based Methodology," ICCAD'21]

# "Generate": Transfer Learning Framework

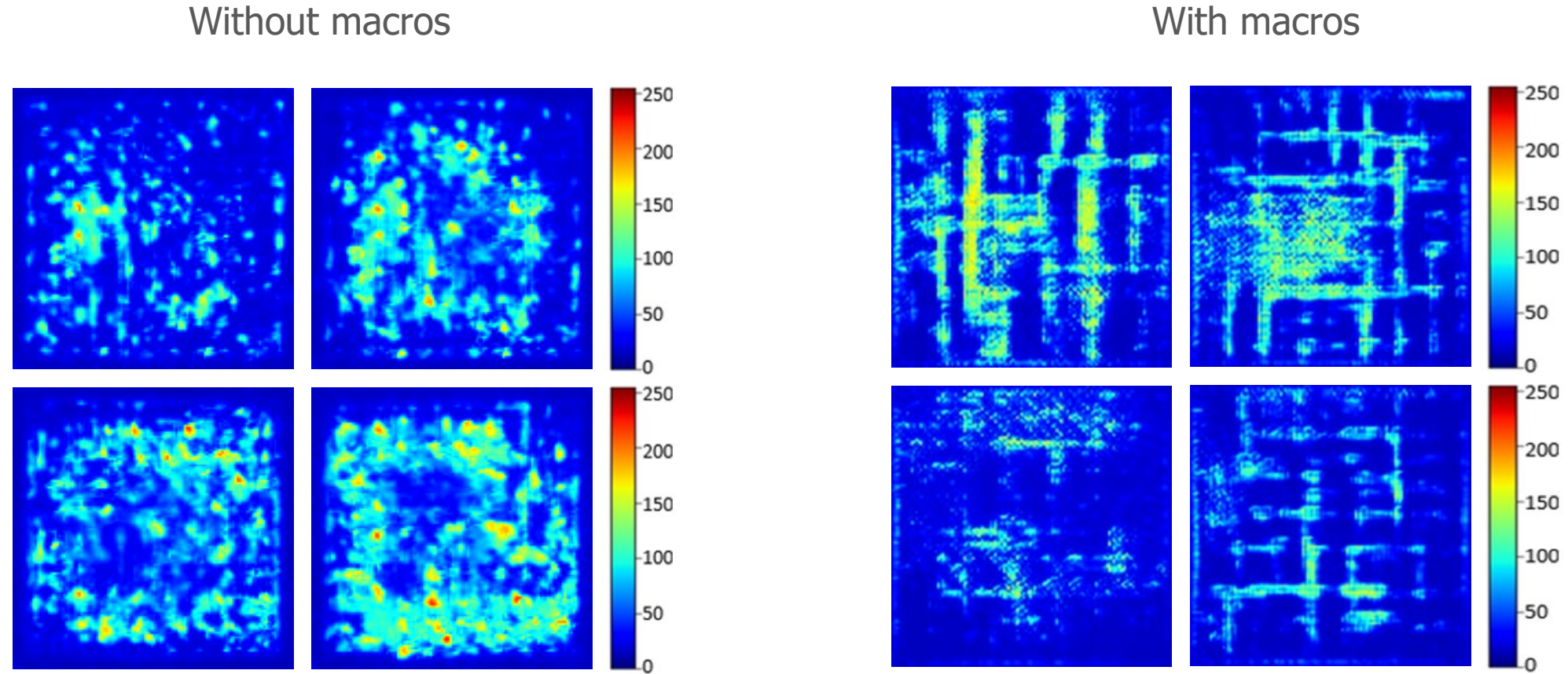


- SNGAN
  - Easy to adapt for training
  - 90M trainable parameters
  - Image size of 128x128 pixels



[Chhabria, Kunal, Zabihi, Sapatnekar, "BeGAN: Power Grid Benchmark Generation Using a Process-portable GAN-based Methodology," ICCAD'21]

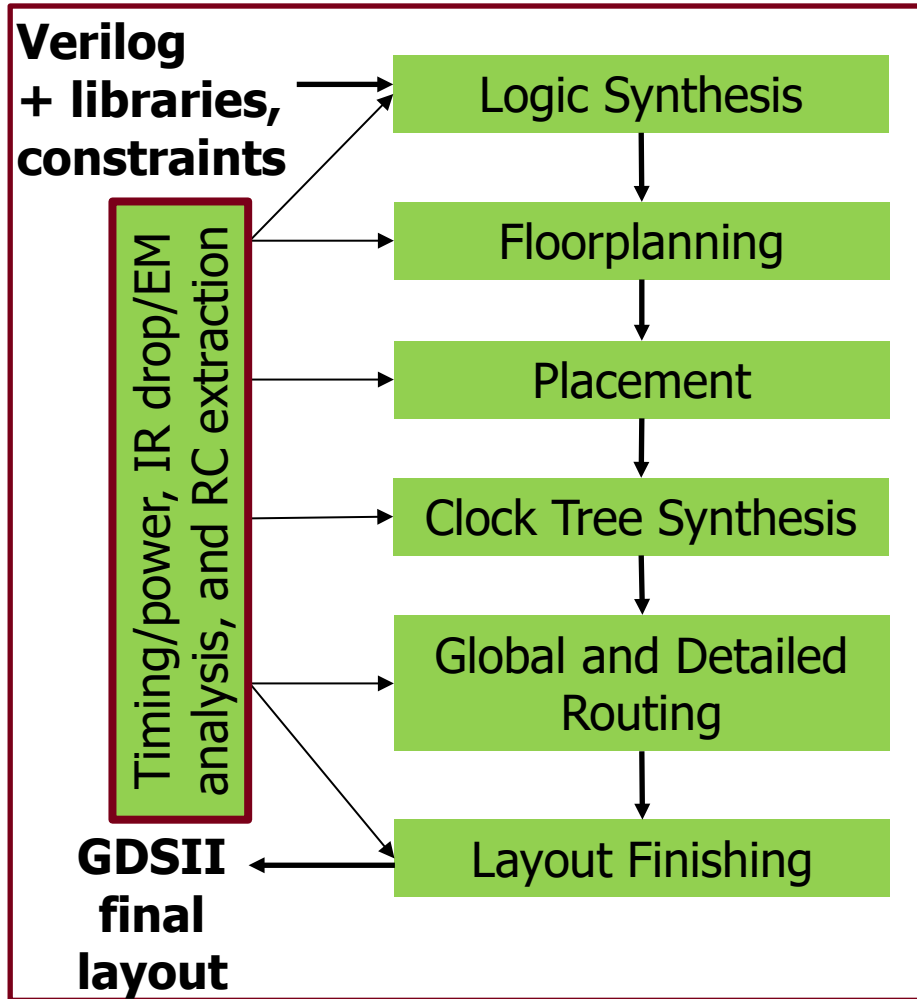
# “Generate”: Large number of generated CM images



Generated current maps have features similar to the original current maps

# Agenda

IC Design Cycle:



**24 hours, no humans – no power, performance, area loss**

**Optimize:** On-chip power delivery networks (PDNs)

**Predict:** IR drop, on-chip temperature, EM hotspots

**Generate:** Benchmark generation for PDNs

**Assist:** EDA chatbot for automatic script generation

**Software:** Python APIs for EDA tools (OpenROAD)

**Schema:** CircuitOps ML EDA data format

**Infrastructure:** SLICE common infra for EDA

**Datasets:** EDA Corpus: LLM dataset for EDA tool

Machine Learning in tools,  
flows

Machine Learning  
infrastructure

**Design Complexity**

# EDA tools and LLMs: EDA Corpus

- EDA Corpus: A Large Language Model Dataset for Enhanced Interaction with OpenROAD  
<https://github.com/OpenROAD-Assistant/EDA-Corpus>
- Over 1500 datapoints of prompt-response pairs including prompt-scripts and question-answers.
- Examples:

## Prompt:

Give me the sample code to get a list of all the output pins of instances in a design using OpenROAD Python API.

## Response:

```
# Get the current design block from the OpenROAD database
block = ord.get_db_block()
# Retrieve all instances present in the design block
insts = block.getInsts()
output_pins = []
# Iterate over each instance in the design block
for inst in insts:
    # Retrieve the pins of the instance
    pins = inst.getITerms()
    # Iterate over each pin of the instance
    for pin in pins:
        # Check if the pin is an output signal
        if pin.isOutputSignal():
            # Append the output pin to the list
            output_pins.append(pin)
return output_pins
```

B.-Y. Wu, U. Sharma et. al., EDA Corpus, LAD'24  
[Arizona State University and New York University]

## Prompt:

What is PDNGEN?

## Response:

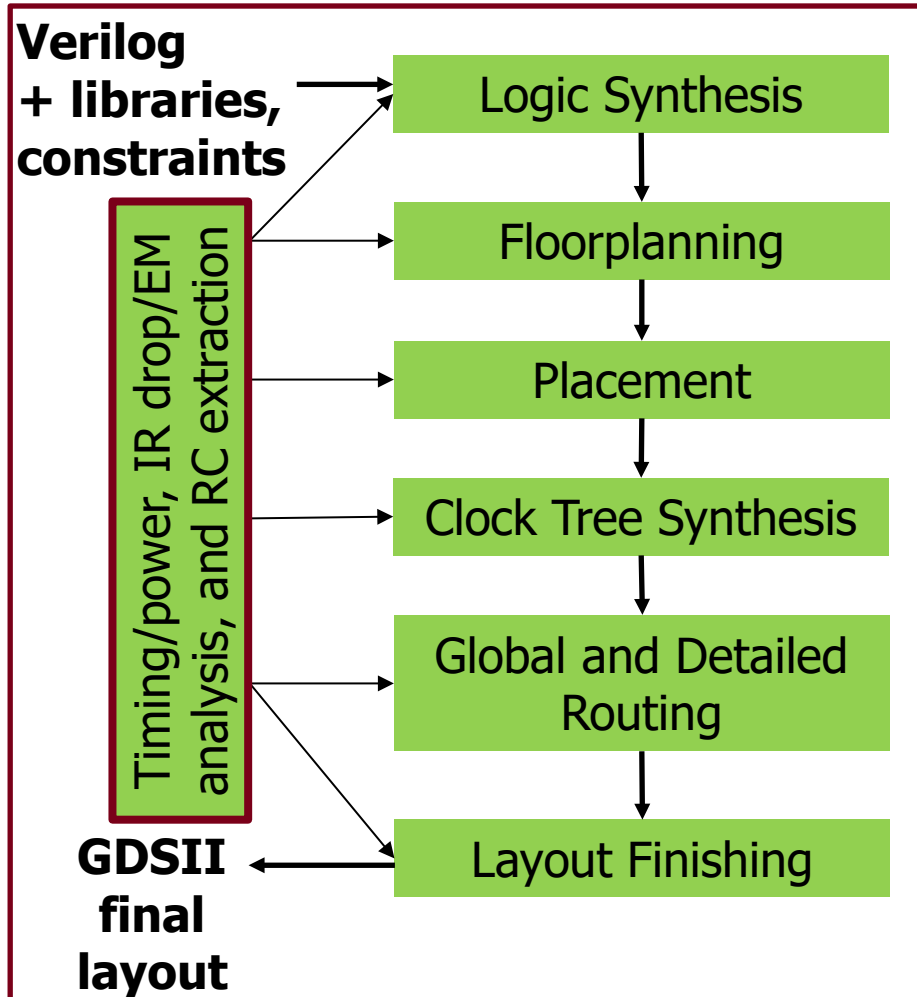
The PDNGEN module (pdn) in OpenROAD aims to simplify the process of adding a power grid into a floorplan. A user can specify a small set of power grid policies to be applied to the design, such as the layers to use, stripe width, and spacing, and then pdn will generate the actual metal straps. Power grid policies can be defined over both the standard cell area and areas occupied by macros.

| Model                                      | Prompt-script |       | Question-answer |         |       |
|--------------------------------------------|---------------|-------|-----------------|---------|-------|
|                                            | True          | False | True            | Partial | False |
| ChatGPT3.5                                 | 0             | 50    | 4               | 17      | 9     |
| ChatGPT4                                   | 3             | 47    | 6               | 17      | 7     |
| ChatGPT3.5 fine-tuned with prompt-script   | 22            | 28    | –               | –       | –     |
| ChatGPT3.5 fine-tuned with question-answer | –             | –     | 26              | 3       | 1     |

Fine-tuning ChatGPT3.5 with EDA Corpus

# Agenda

IC Design Cycle:



**24 hours, no humans – no power, performance, area loss**

**Optimize:** On-chip power delivery networks (PDNs)

**Predict:** IR drop, on-chip temperature, EM hotspots

**Generate:** Benchmark generation for PDNs

**Assist:** EDA chatbot for automatic script generation

Machine Learning in tools, flows

Machine Learning infrastructure

**Software:** Python APIs for EDA tools (OpenROAD)

**Schema:** CircuitOps ML EDA data format

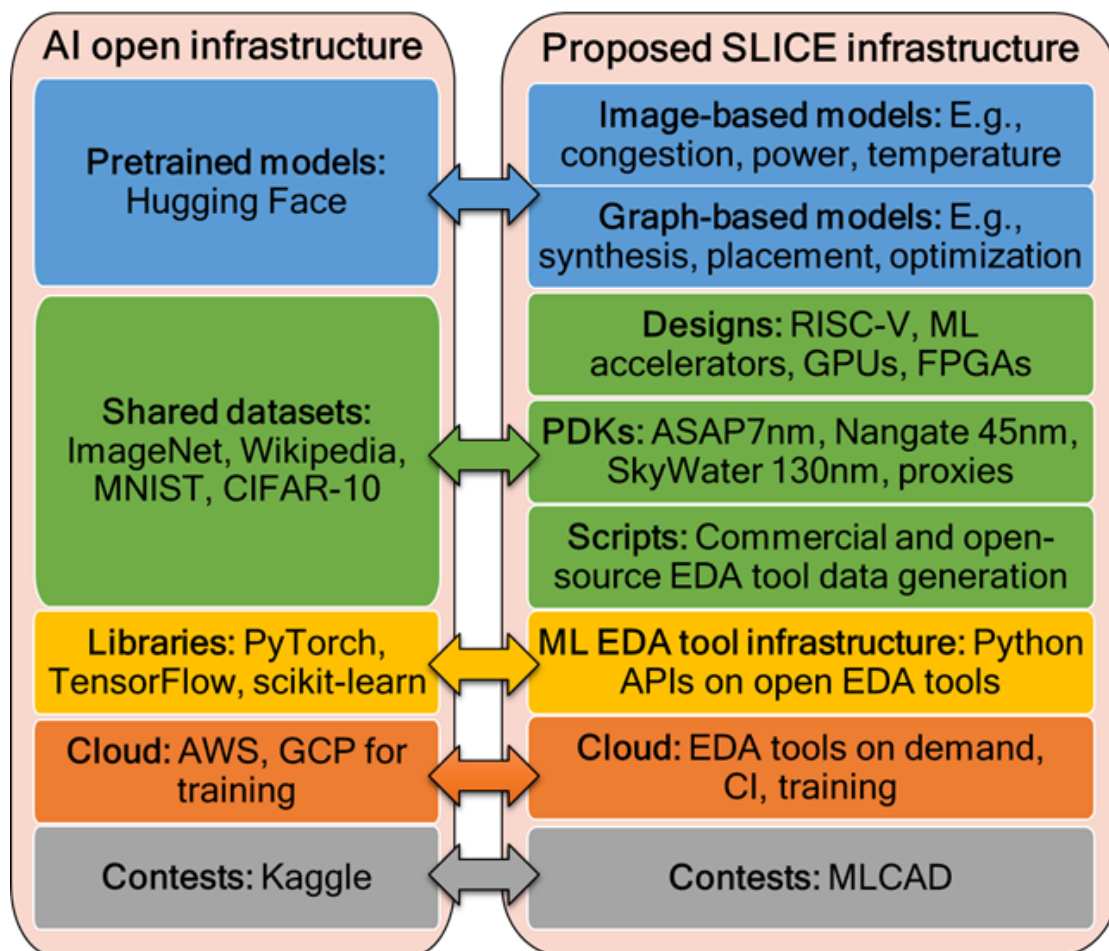
**Infrastructure:** SLICE common infra for EDA

**Datasets:** EDA Corpus: LLM dataset for EDA tool

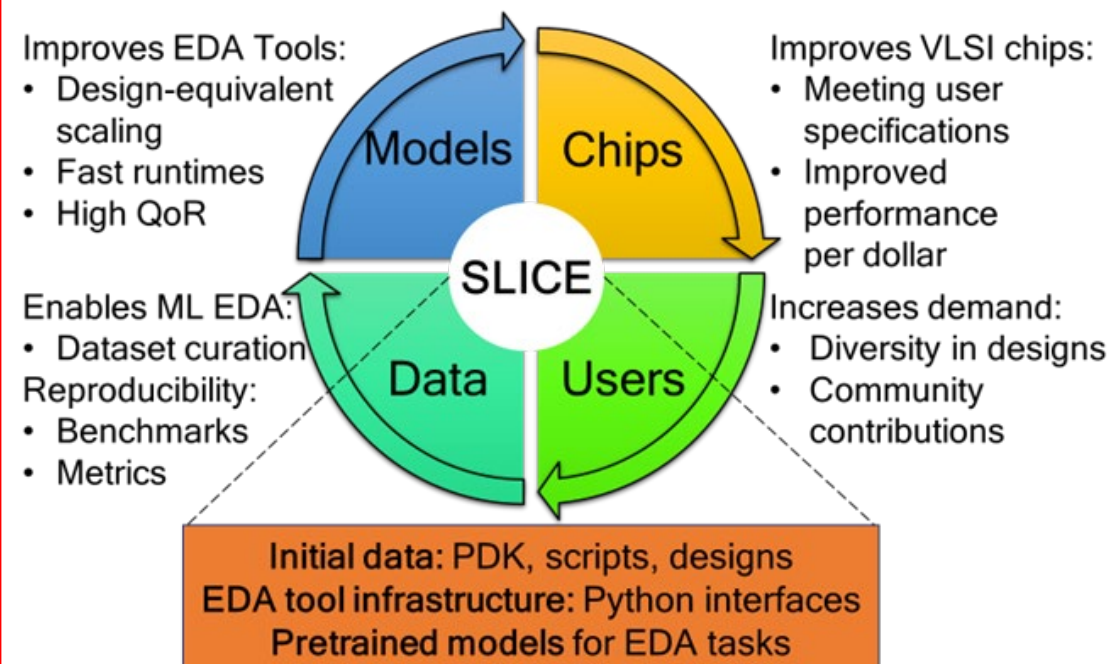
**Design Complexity**

# SLICE: Shared Infrastructure for ML EDA

## Learning from the ML Community



## ML EDA Ecosystem Flywheel



Prof. Jiang Hu, Texas A&M University. MLCAD 2023 invited talk

[SLICE website](#): Serving as a one-stop shop for ML EDA infrastructure with pointers to datasets, EDA tool flows, contests, and proxy PDKs.

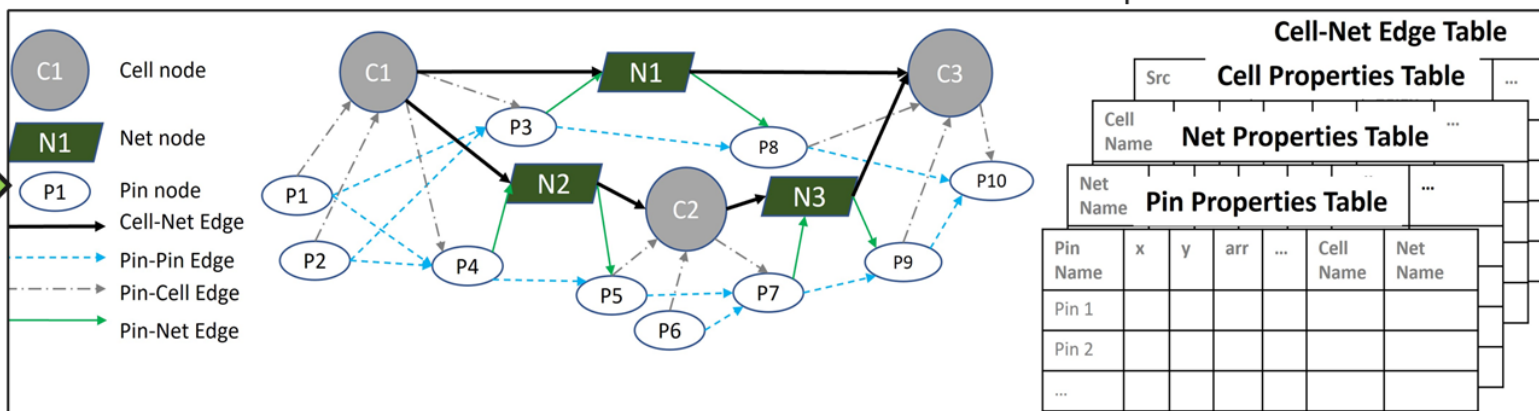
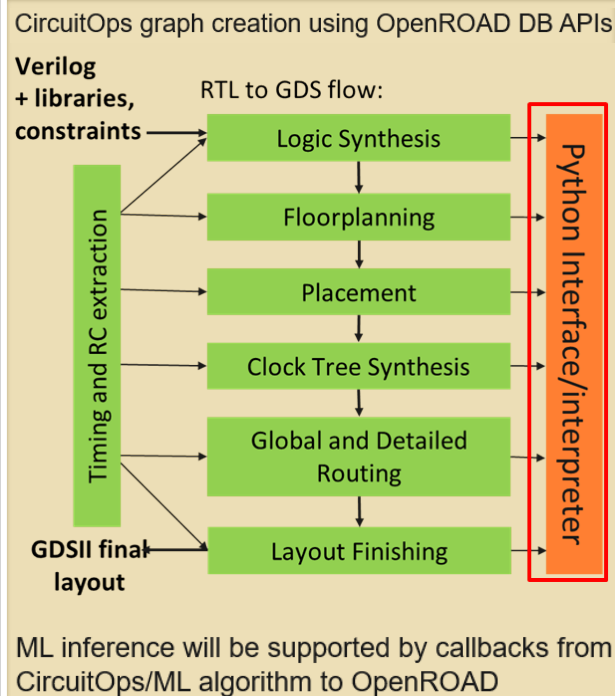
See also: March 2023 [NSF Workshop](#) on Shared Infrastructure for Machine Learning EDA

# An ML for Chip Design Playground <https://github.com/NVlabs/CircuitOps>

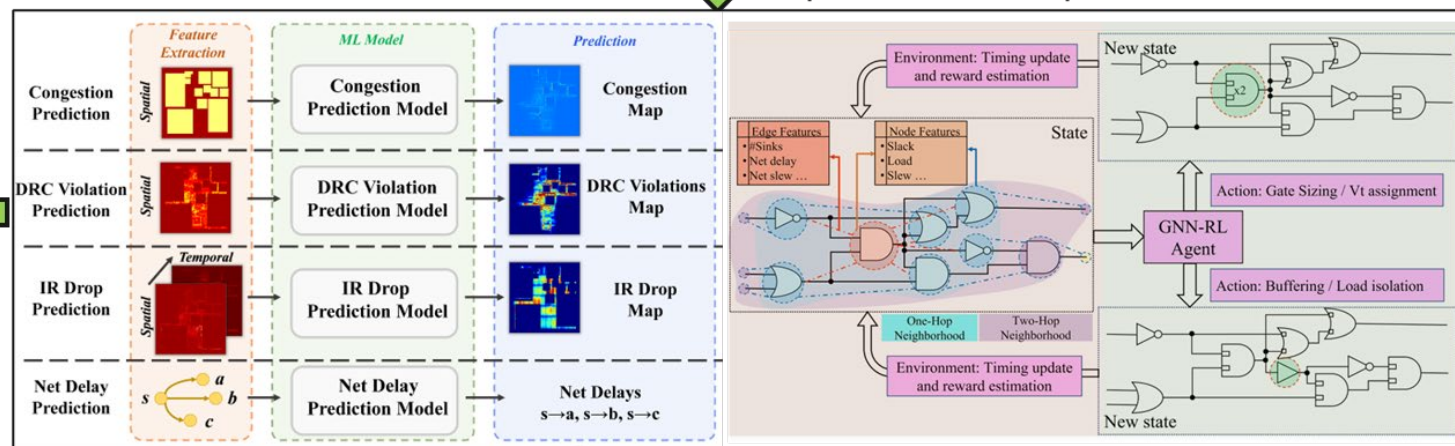


## CircuitOps: ML-friendly data representation format within OpenROAD

pandas.DataFrame features



Easy application of ML training within OpenROAD interpreter



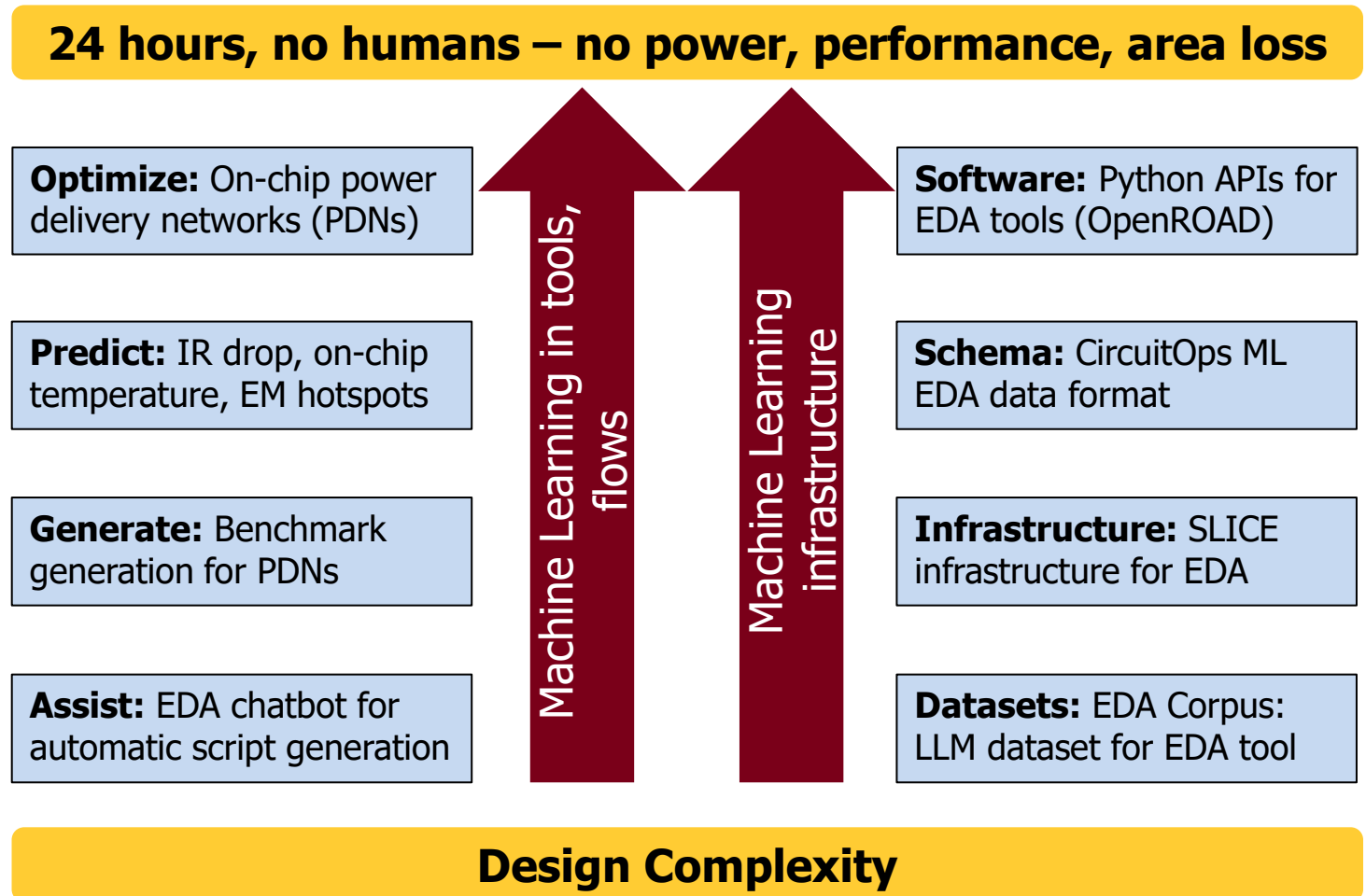
ML/RL algorithms integrated within OpenROAD

**CircuitOps and OpenROAD: Unleashing ML EDA for Research and Education**

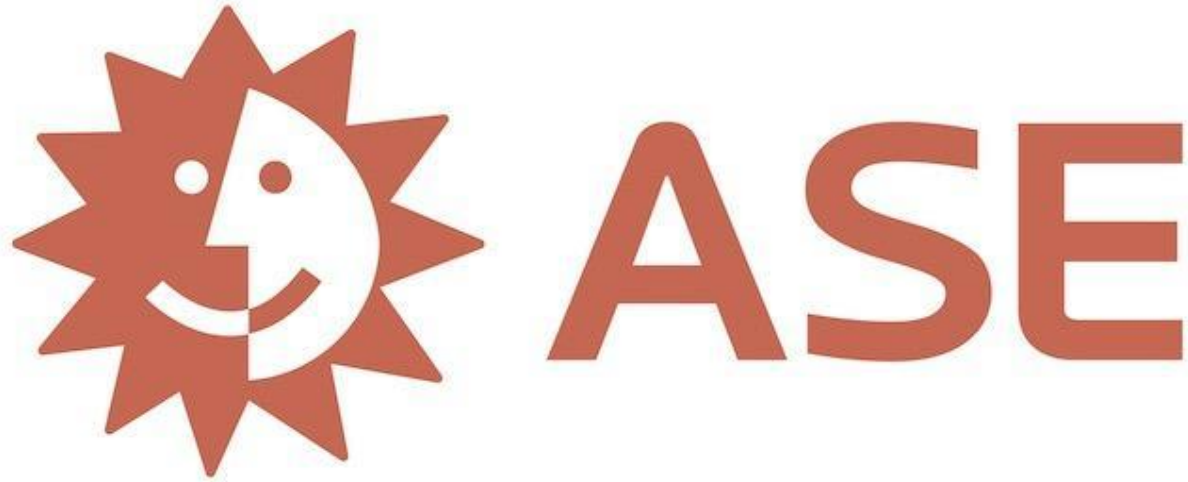
Andrew B. Kahng, UCSD  
Vidya A. Chhabria, ASU  
Bing-Yue Wu, ASU

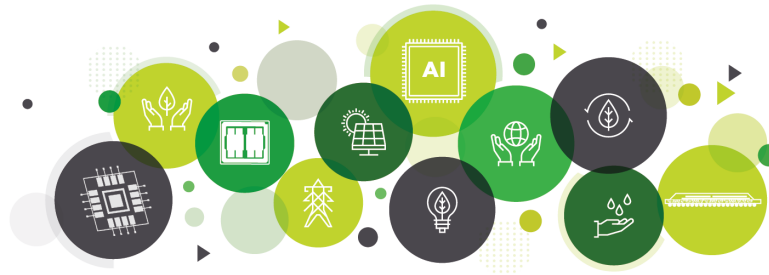
# Conclusion

- **Challenges in EDA**
  - **Scale:** The problem of “N” in EDA
  - **Time-to-market:** Computationally expensive EDA tools
  - **Productivity:** Sub-optimal automated solutions require tremendous manual intervention for high-quality
  - **Cost:** Design dollar costs, risk, and expertise are barriers to entry
- **ML enables:**
  - Fast turn-around times due to fast analysis and prediction
  - High QoR due to fast optimizations and takes out guesswork from IC design
- **Open-source:**
  - Publicly available easy to use tools and benchmarks that reduce barriers to entry



Thank you to our sponsors!





ASE is enabling the **heterogeneous integration** and **chiplets** era through **VIPack™** while delivering sustainable advanced packaging innovations for...

**AI | HPC | Data Center | Automotive | 6G | IoT | and more.**

 @aseglobal

 @aseglobal

 aseglobal.com

# COPYRIGHT NOTICE

This presentation in this publication was presented at the **AI for Semiconductors** (June 12-13, 2024). The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by MEPTEC or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

[www.meptec.org](http://www.meptec.org)