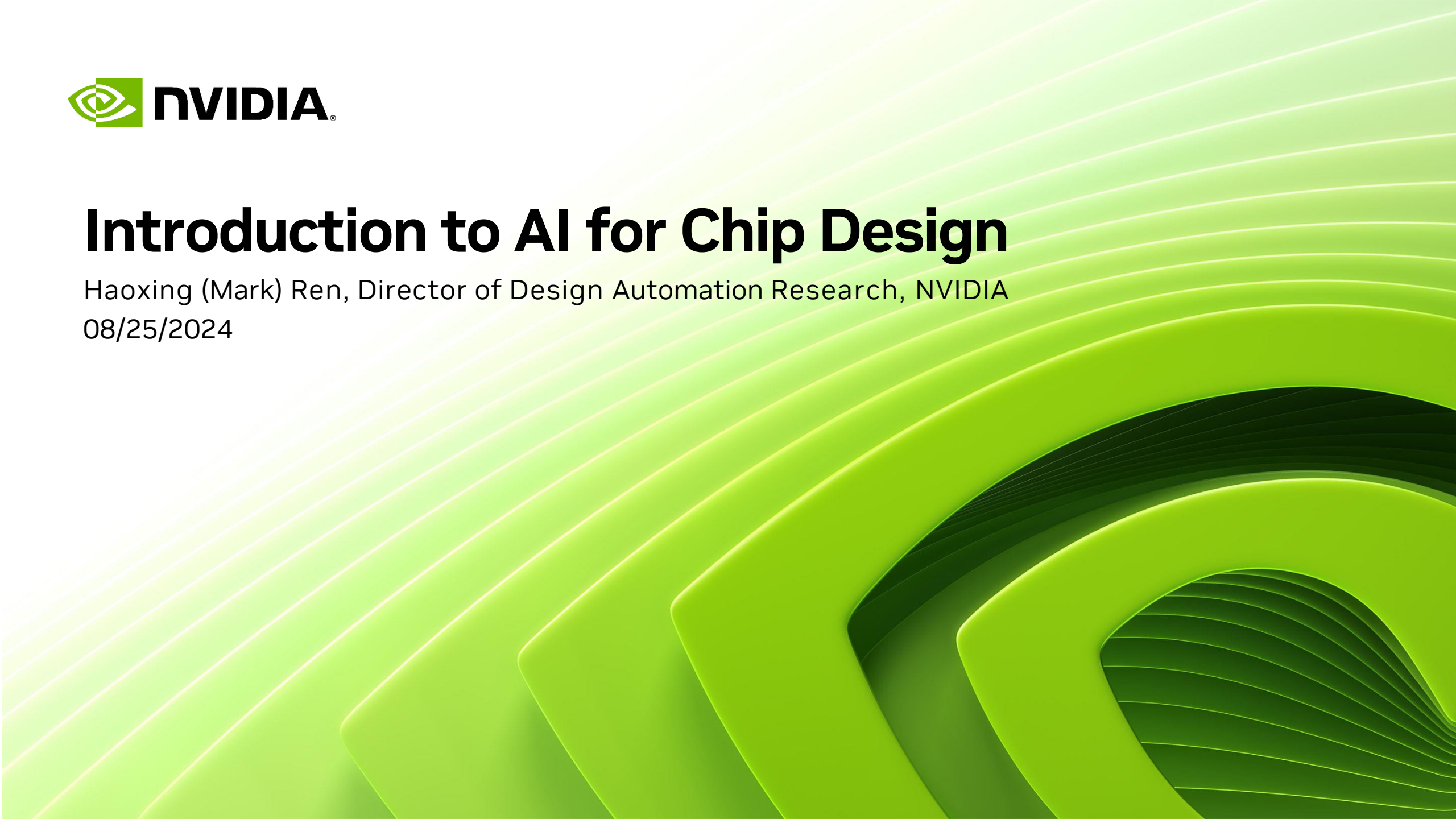


Introduction to AI for Chip Design

Haoxing (Mark) Ren, Director of Design Automation Research, NVIDIA

08/25/2024

AI for Design Performance and Productivity

Analysis

Faster
Predictive
Cross-Stage

Optimization

Faster
More scalable
Better results

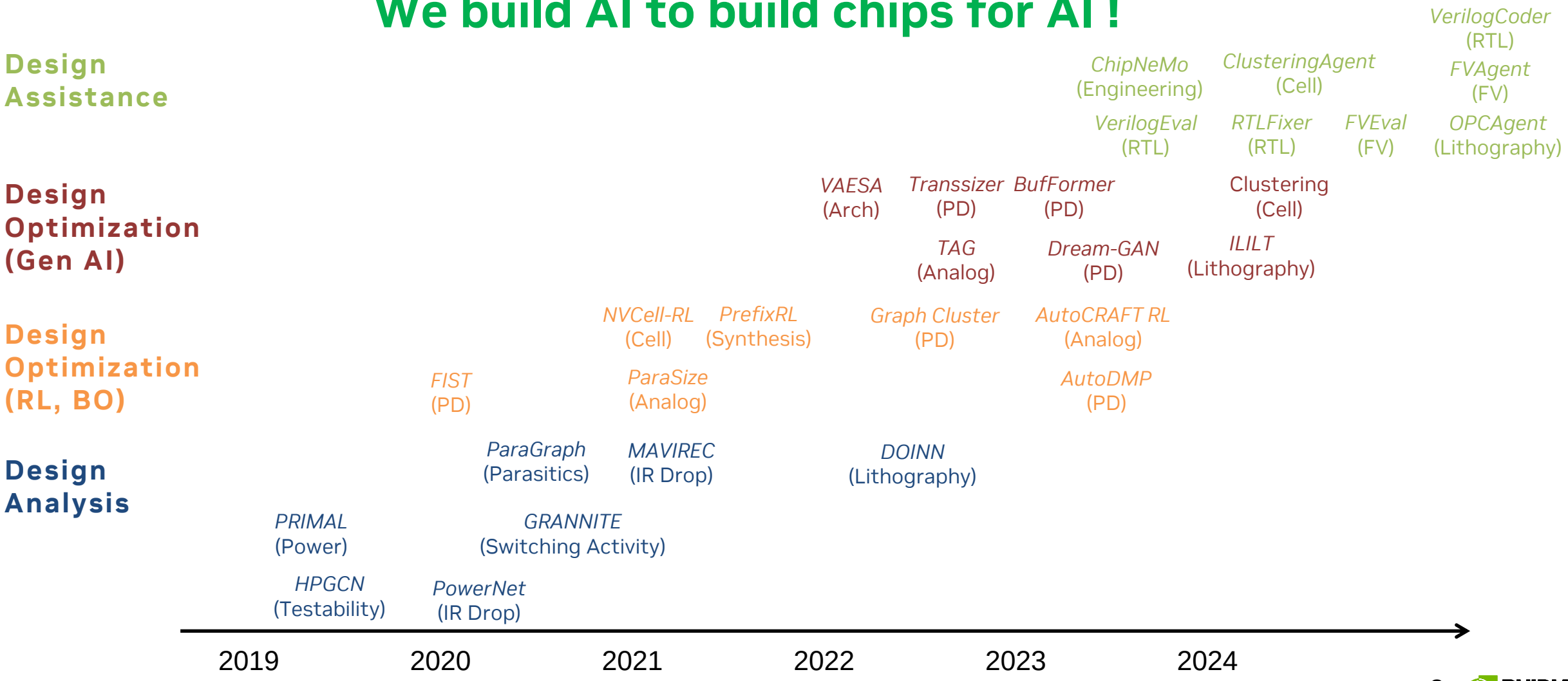
Assistance

Know-how
Coding
Task automation

NVIDIA

AI for Chip Design Research @ NVIDIA

We build AI to build chips for AI !



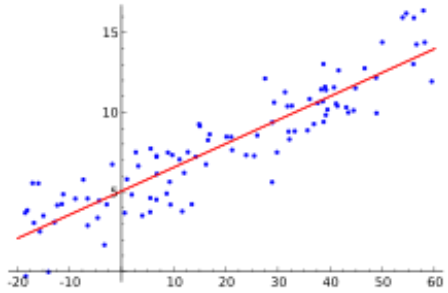
AI Techniques

- **Analysis**
  - Classical ML
  - Deep learning
- **Optimization**
  - Bayesian optimization
  - Reinforcement learning
- **Optimization**
  - Generative AI
- **Assistance**
  - LLM

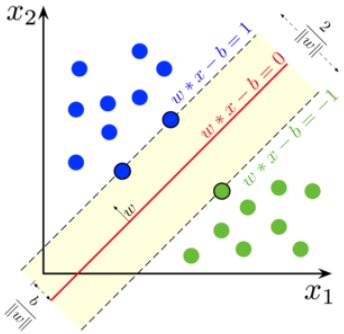
AI Techniques

- **Analysis**
  - Classical ML
  - Deep learning
- **Optimization**
  - Bayesian optimization
  - Reinforcement learning
- **Optimization**
  - Generative AI
- **Assistance**
  - LLM

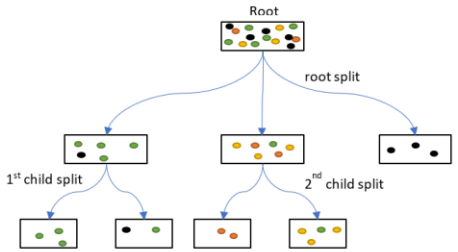
Linear Regression



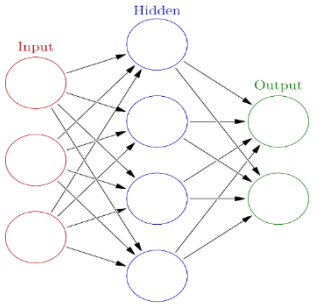
Support Vector Machine



Decision Tree



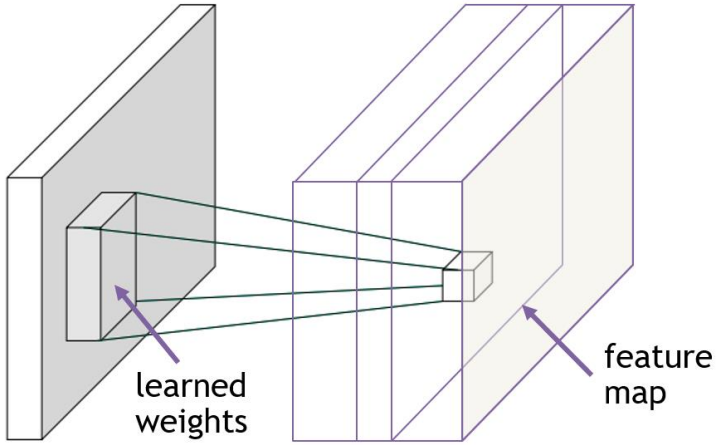
Neural Network



Suitable for small structured data

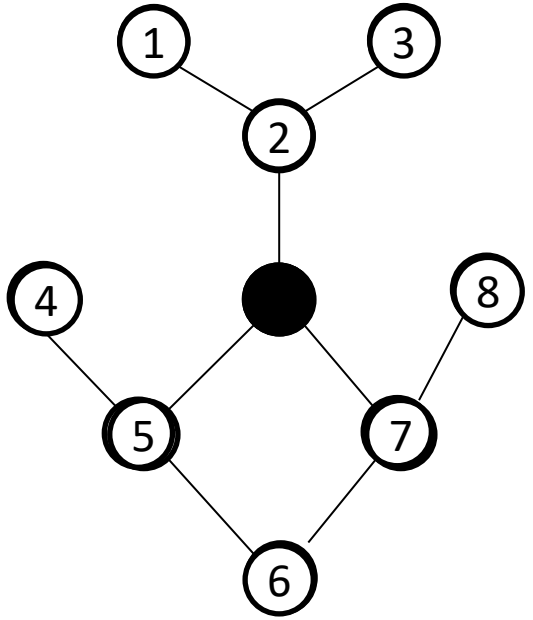
AI Techniques

- **Analysis**
  - Classical ML
  - Deep learning
- **Optimization**
  - Blackbox optimization
  - Reinforcement learning
- **Optimization**
  - Generative AI
- **Assistance**
  - LLM



CNN

Suitable for physical design data



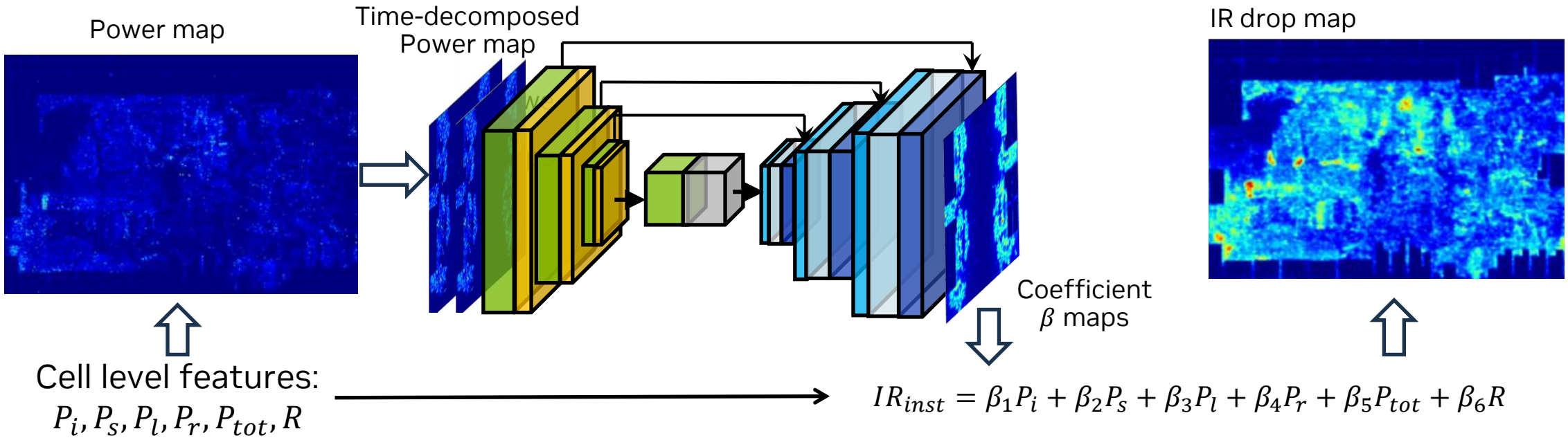
GNN

Suitable for circuit netlist data

Faster Analysis – IR Drop Estimation

IR drop estimation is important for physical design, but it takes hours

Use AI to predict IR drop from cell level features



94% accuracy in 3 second vs 3 hr in commercial tools

V.A. Chhabria et al, MAVIREC: ML-Aided Vectored IR-Drop Estimation and Classification

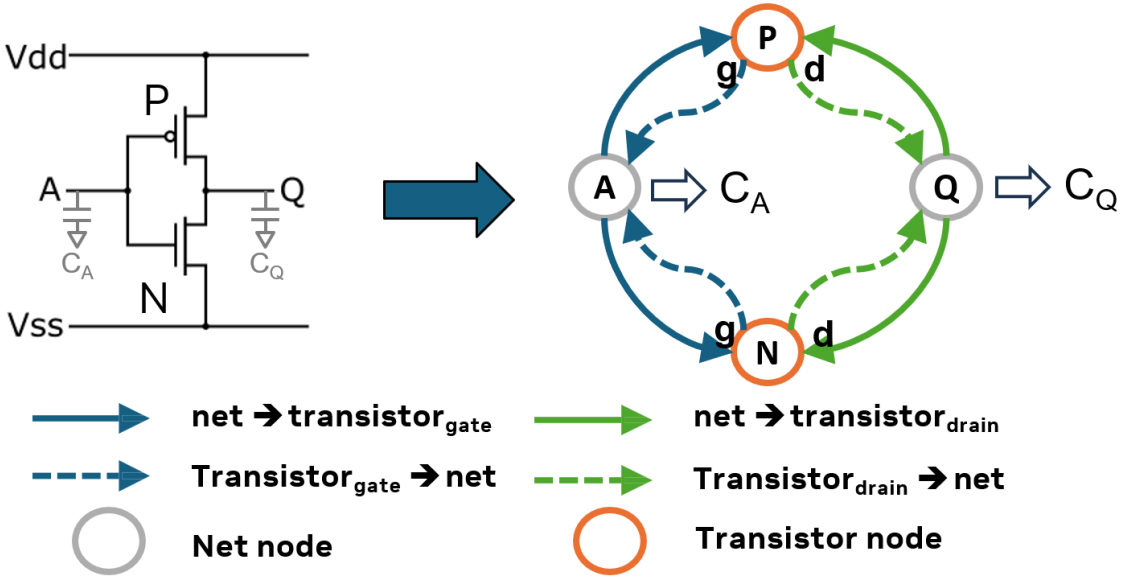
NVIDIA

Cross-Stage Analysis – Parasitics Prediction

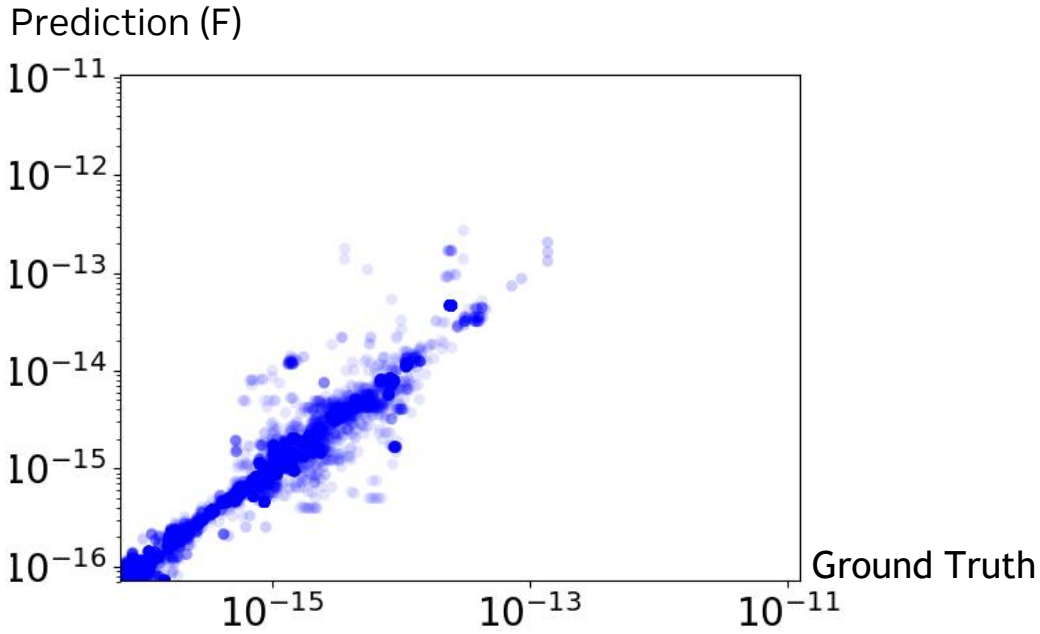
Impact of layout parasitics on schematic design

Use AI to predict layout parasitics from schematic

Convert schematic to graph and learn with GNN



Circuit Schematics to Heterogenous Graph Conversion



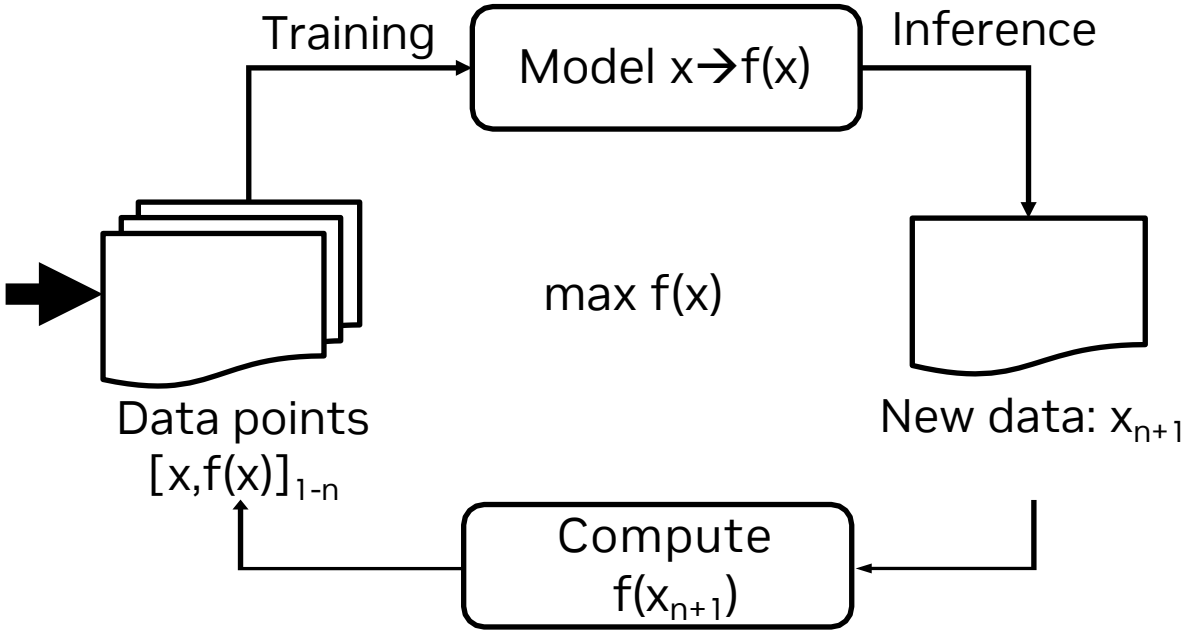
MAE=0.852fF MAPE=15%
Simulation error reduced to <10%

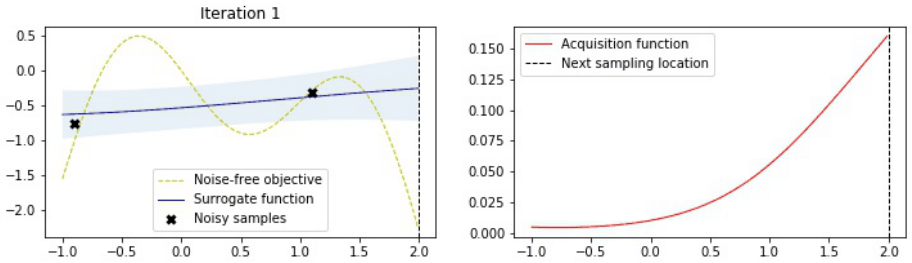
H. Ren et al, ParaGraph: Layout Parasitics and Device Parameter Prediction using Graph Neural Networks

NVIDIA

AI Techniques

- **Analysis**
  - Classical ML
  - Deep learning
- **Optimization**
  - Bayesian optimization
  - Reinforcement learning
- **Optimization**
  - Generative AI
- **Assistance**
  - LLM





Build a probability model of the objective function from data space and use it to select the most promising data to sample next

Parameter Optimization – Macro Placement

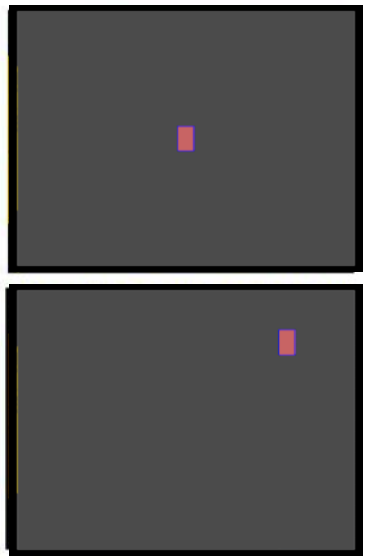
Macro placement quality is very important for physical design

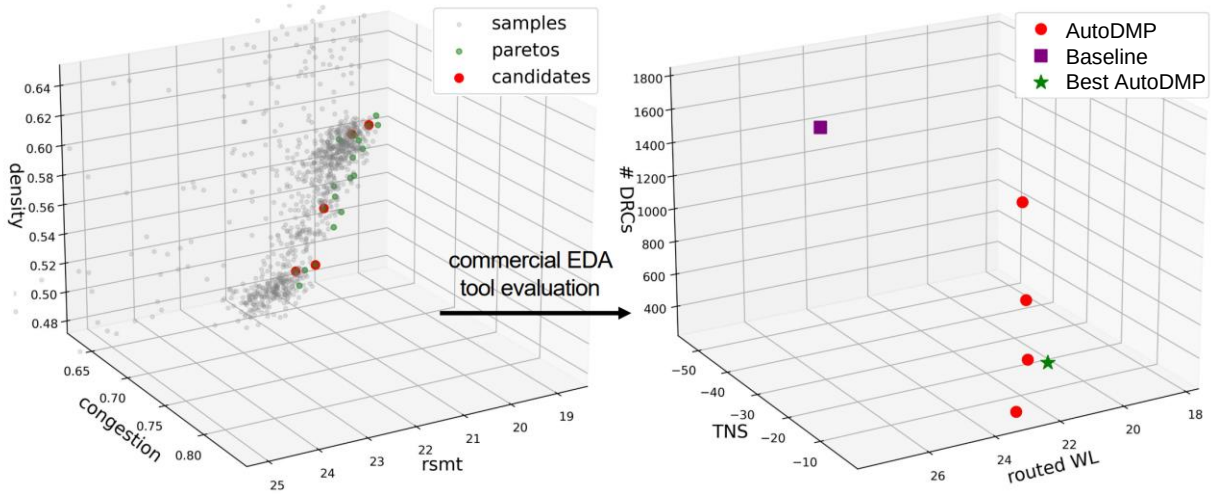
Placement parameters have a huge impact on macro placement

Multi-objective Bayesian optimization: wirelength, congestion, density

Find better macro placement with open-source GPU accelerated placement tools

Parameter	Search Range
*horiz. initial position	[0.2, 0.8] (%)
*vert. initial position	[0.2, 0.8] (%)
*horiz. macro halo	technology dep.
*vert. macro halo	technology dep.
target density d_{target}	$[a_{\text{util}} - 0.2, a_{\text{util}}]$ (%)
density weight	$[1e^{-6}, 1.0]$
smooth HPWL model	{LSE, WA}
smooth HPWL initial γ_0	[0.10, 0.50]
GD initial LR lr_0	$[1e^{-4}, 1e^{-2}]$
GD LR decay	[0.99, 1.0]
GD optimizer	[Adam, Nesterov]
# horiz. global bins	{256, 512, 1024, 2048}
# vert. global bins	{256, 512, 1024, 2048}
λ update lower coeff. L	[0.90, 0.99]
λ update upper coeff. U	[1.01, 1.15]
λ update Δ HPWL_{REF}	$[1.5e^5, 5.5e^5]$

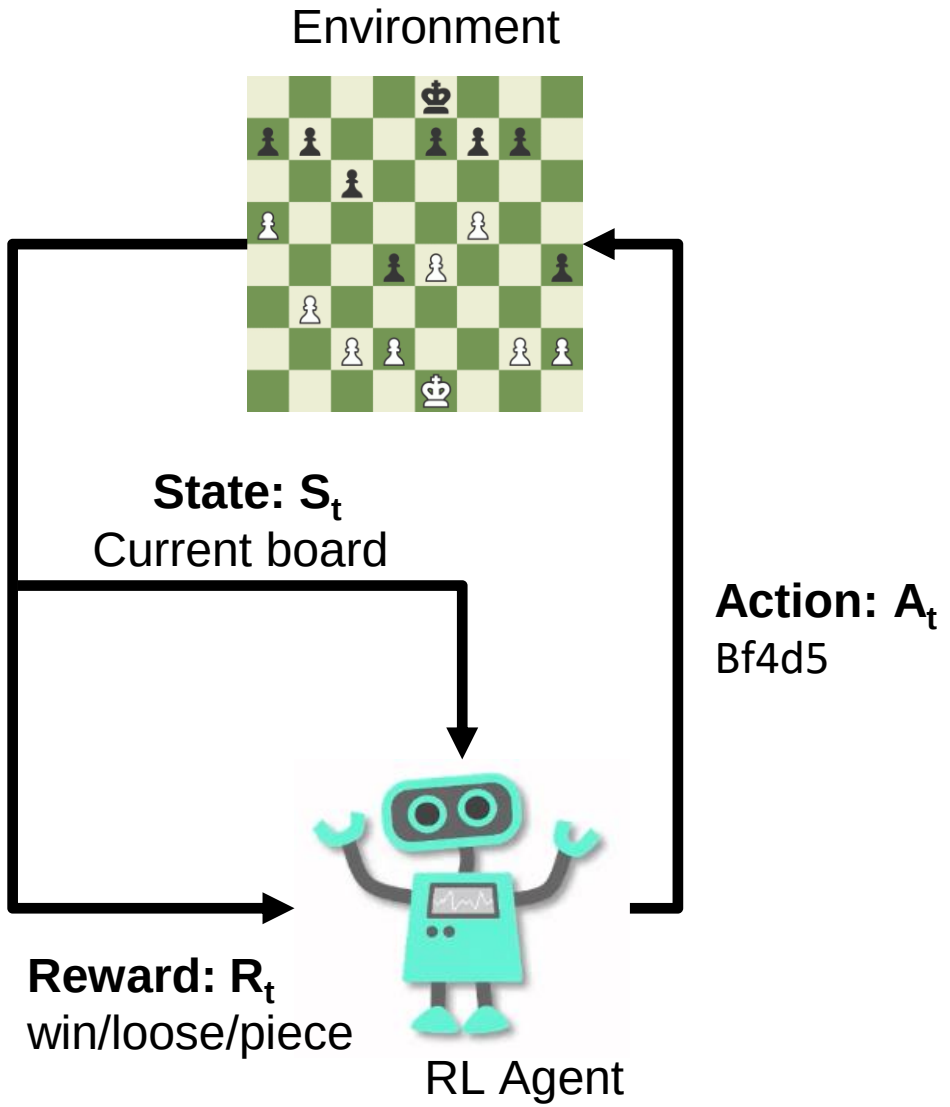




A. Agnesina et al, AutoDMP: Automated DREAMPlace-based Macro Placement

AI Techniques

- **Analysis**
  - Classical ML
  - Deep learning
- **Optimization**
  - Bayesian optimization
  - Reinforcement learning
- **Optimization**
  - Generative AI
- **Assistance**
  - LLM



Objective → Reward
Variables → Action

Fix Design Rule Check(DRC)

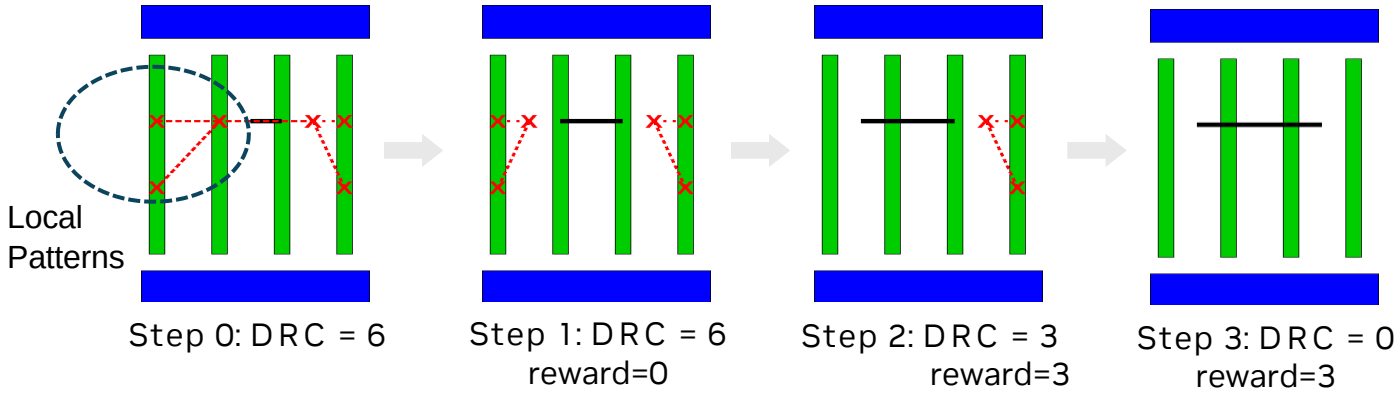
Too many DRC rules to consider for cell layout
RL agent learns to fix DRC automatically

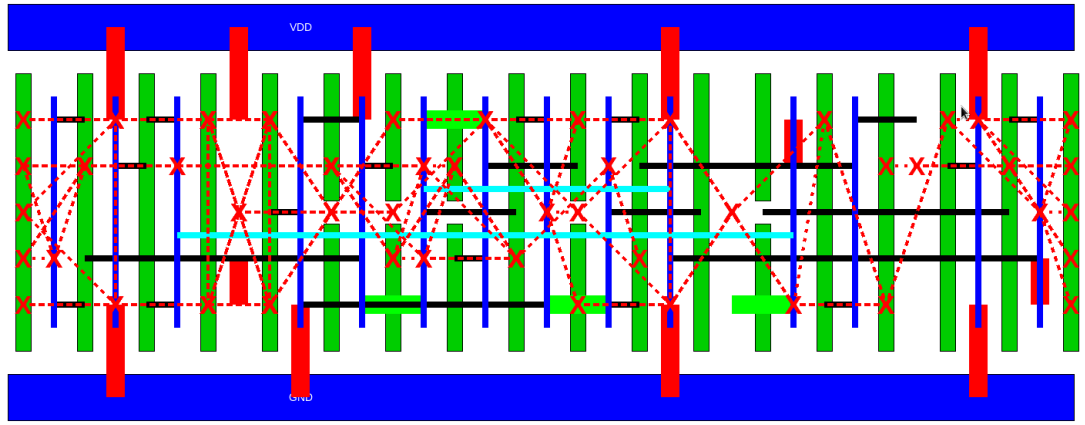
Action: adding additional M0 grid to reduce DRCs

State: current layout images

Reward: DRC reduction

Used in NVCell layout generator in production



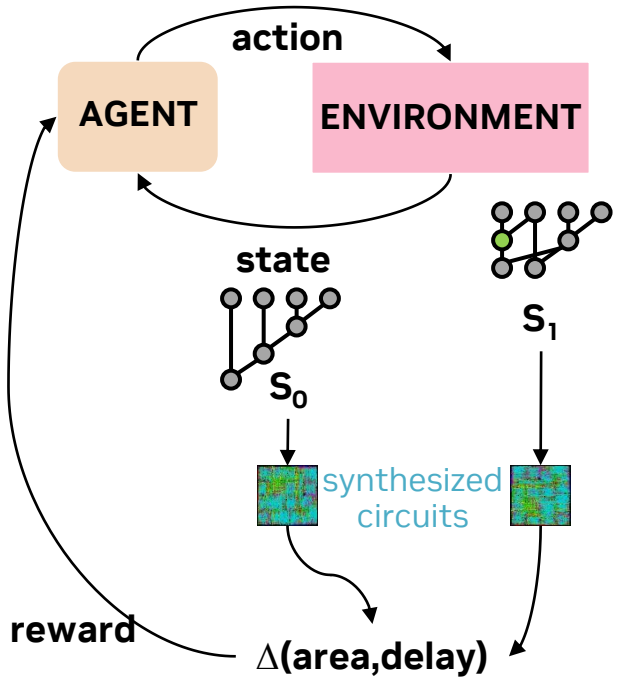


H. Ren et al, NVCell: Standard Cell Layout in Advanced Technology Nodes with Reinforcement Learning

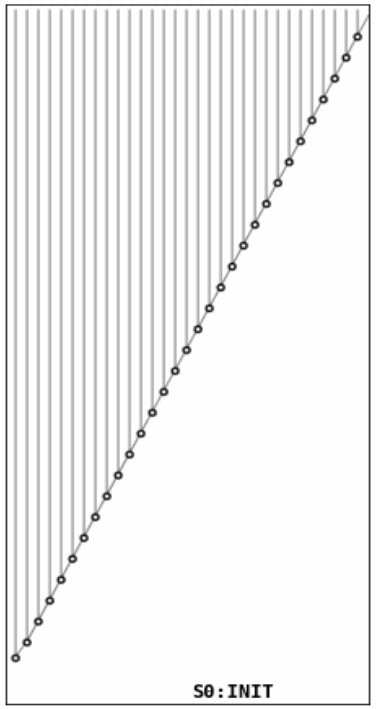
Design Better Datapath

Datapath synthesis important for GPU

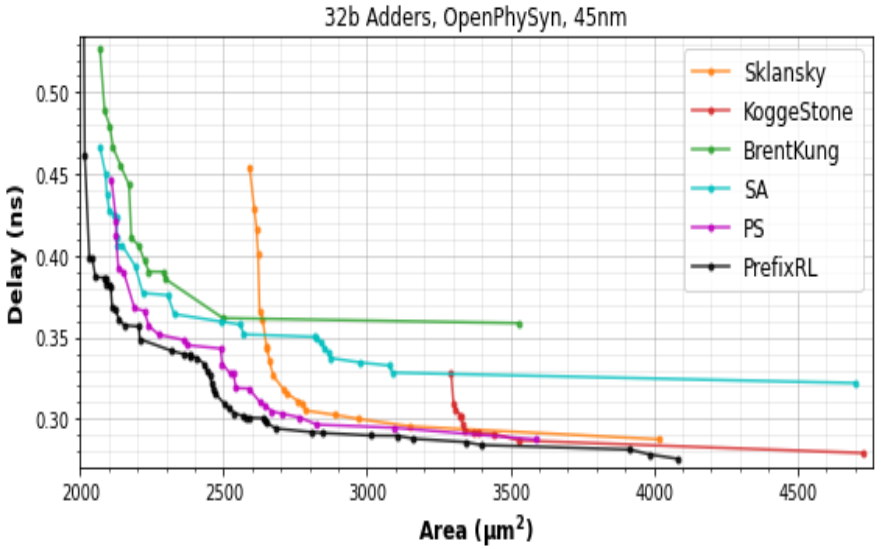
Optimize prefix adder structure with RL



Deep Q learning
Circuit synthesis in-the-loop



Action Space
Add/Delete prefix graph nodes



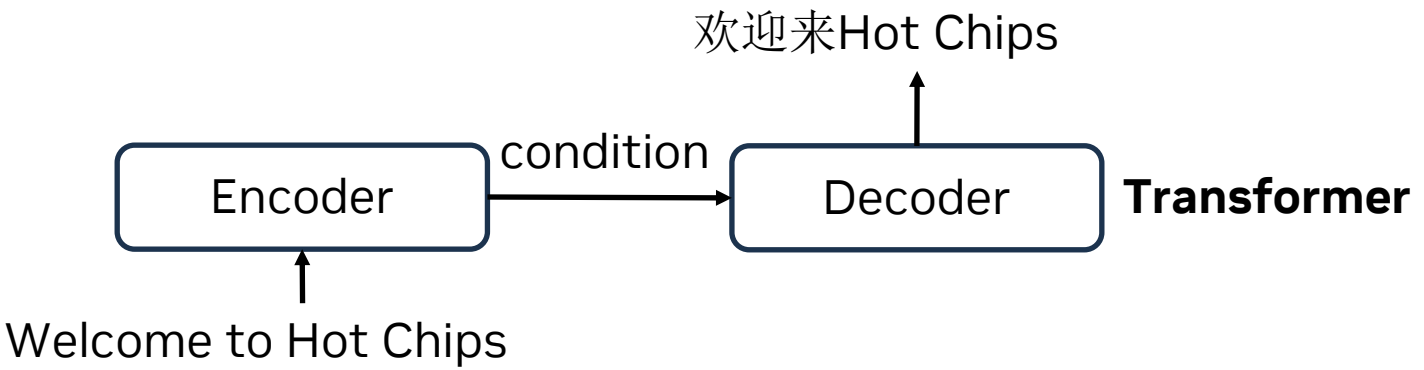
PrefixRL achieves better results than well known adder architectures

Roy et al, PrefixRL: Optimization of parallel prefix circuits using deep reinforcement learning

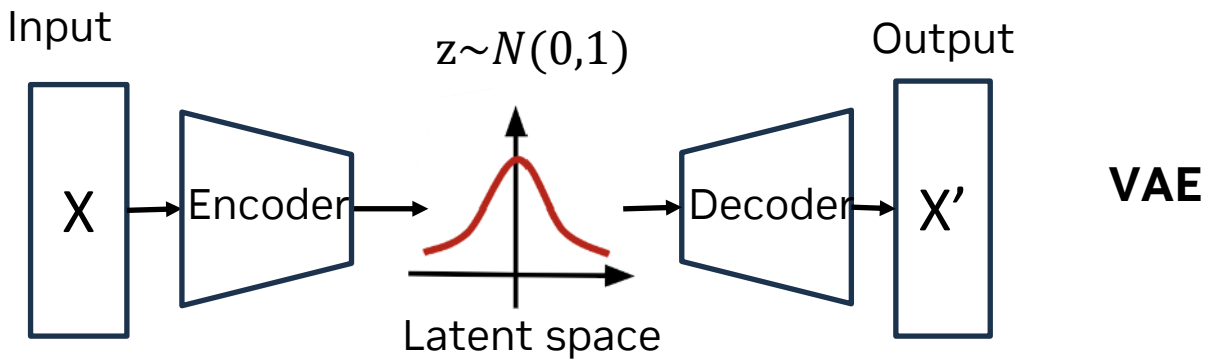
AI Techniques

- **Analysis**
  - Classical ML
  - Deep learning
- **Optimization**
  - Bayesian optimization
  - Reinforcement learning
- **Optimization**
  - Generative AI
- **Assistance**
  - LLM

Generate optimal design points



Representation Learning for Optimization

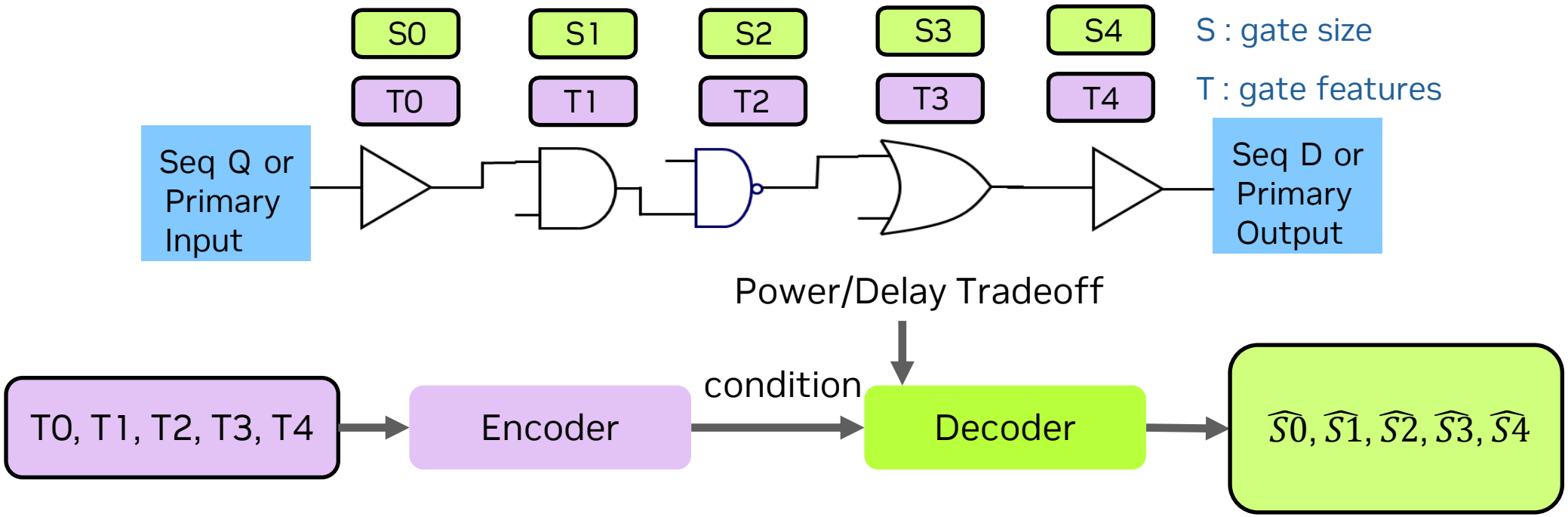


NVIDIA

Generate Optimal Gate Size

Timing/power optimization such as gate sizing affects scalability of PD tools

Model a path of gates as a sequence, generate optimized gate sizes using Transformer



100X – 1000X speedup compared to traditional optimization with similar PPA

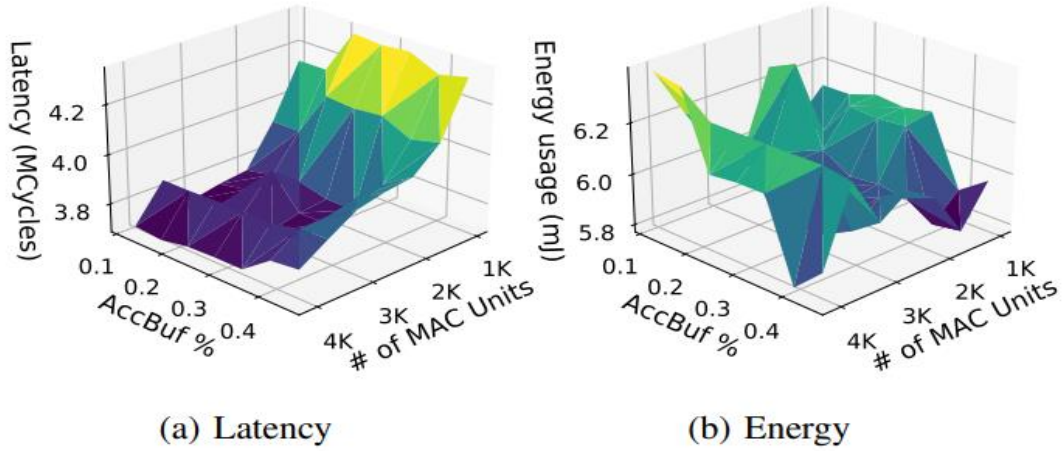
S. Nath et al, Transizer: A Novel Transformer-Based Fast Gate Size

Optimization – Accelerator Design

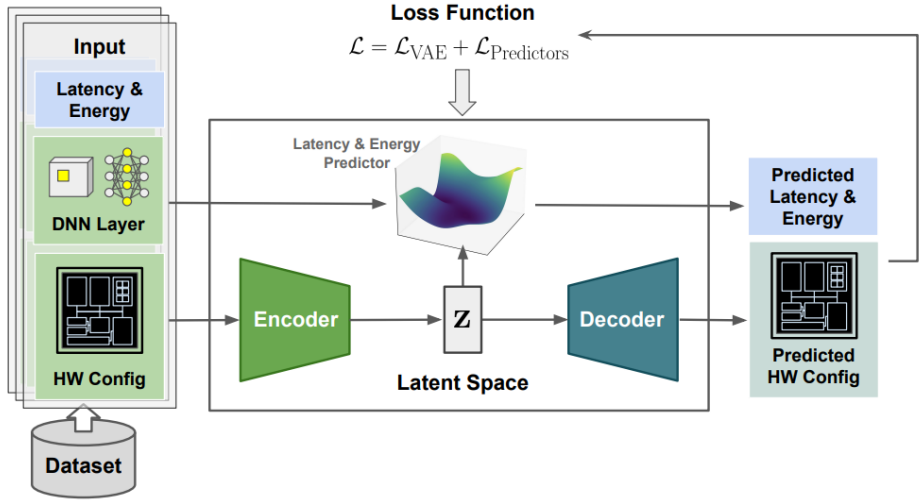
Irregular landscape of neural network accelerator design space

Optimize on the latent space (reduced dim, smooth) learned using VAE

6.8X sample efficiency and 5% better performance



(a) Latency (b) Energy



Q. Huang et al, Learning A Continuous and Reconstructible Latent Space for Hardware Accelerator Design

NVIDIA

AI Techniques

- **Analysis**
  - Classical ML
  - Deep learning
- **Optimization**
  - Bayesian optimization
  - Reinforcement learning
- **Optimization**
  - Generative AI
- **Assistance**
  - LLM

LLM is good at

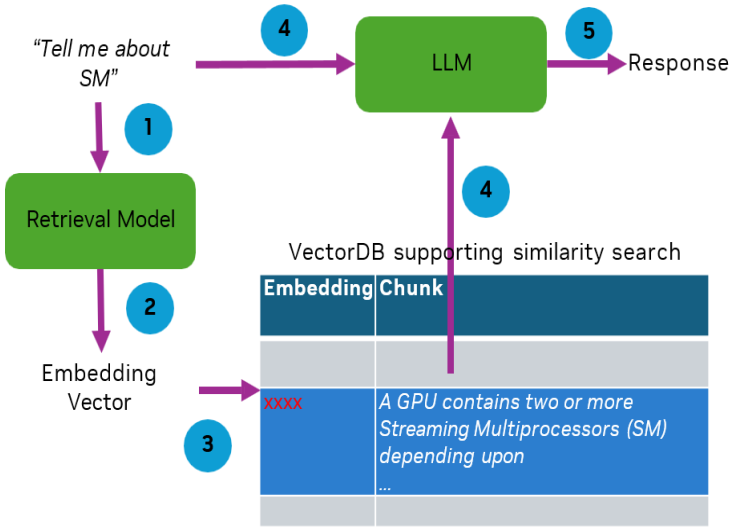
Open question answering
Closed question answering
Coding
Extraction
Rewriting
Classification
Summarization
Reasoning
...

LLM is a generalist
Leverage pre-trained models

NVIDIA

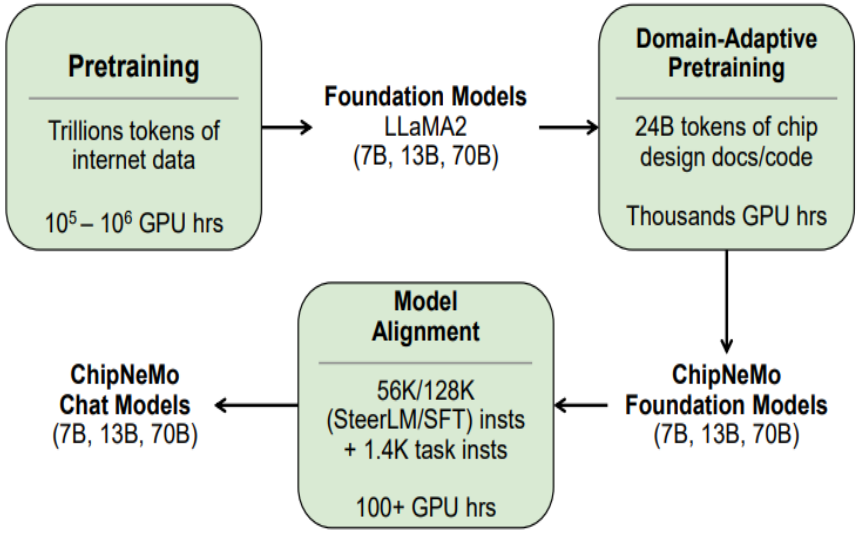
Make LLM Learn to Do Chip Design

In-Context Learning



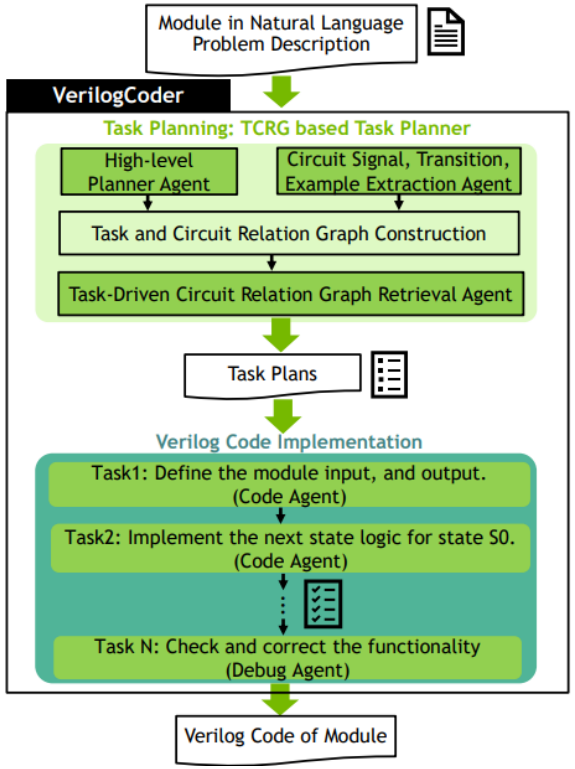
Retrieval Augmented Generation (RAG)

Parameter Training



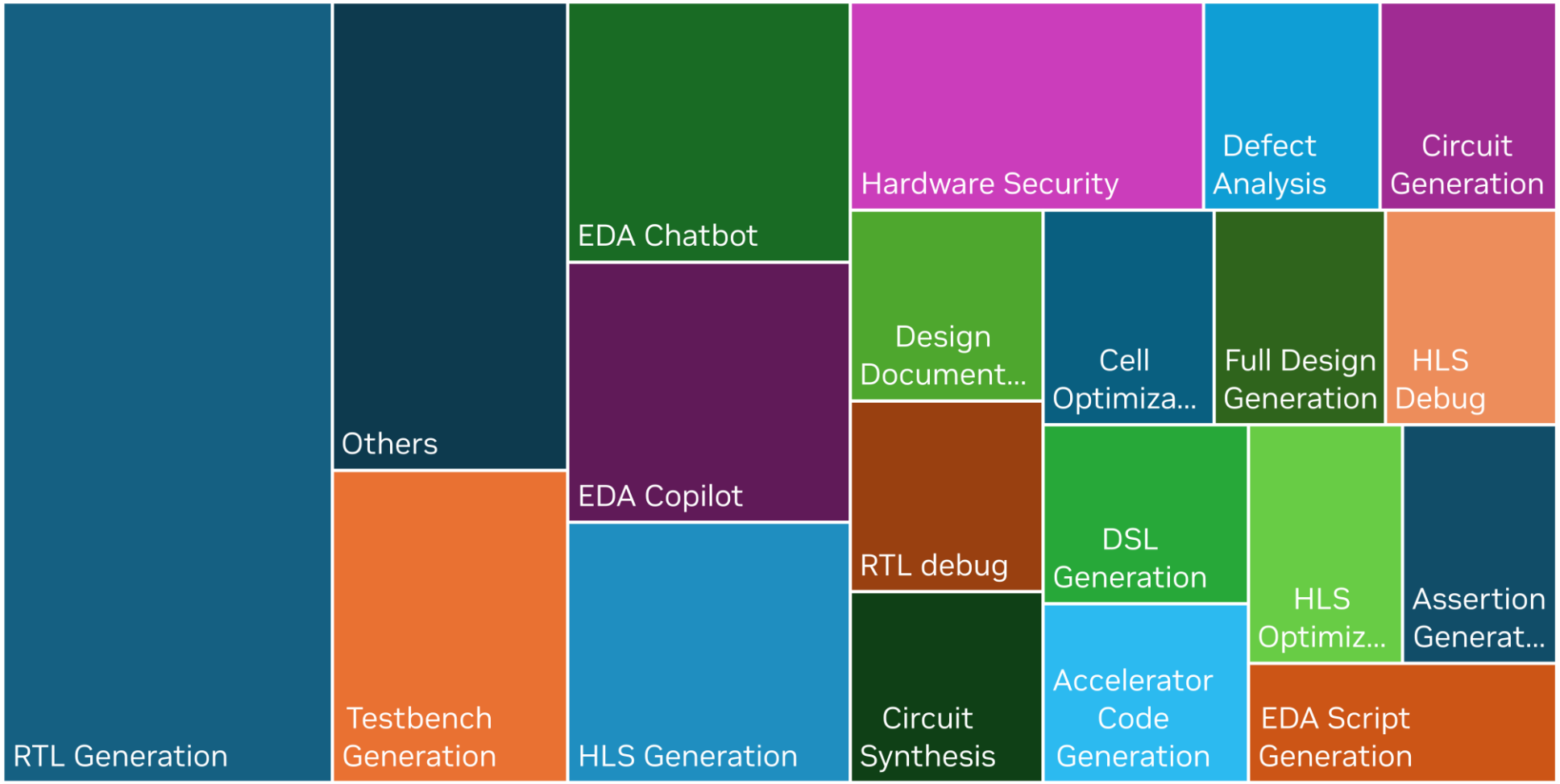
M. Liu et al, ChipNeMo

Agent



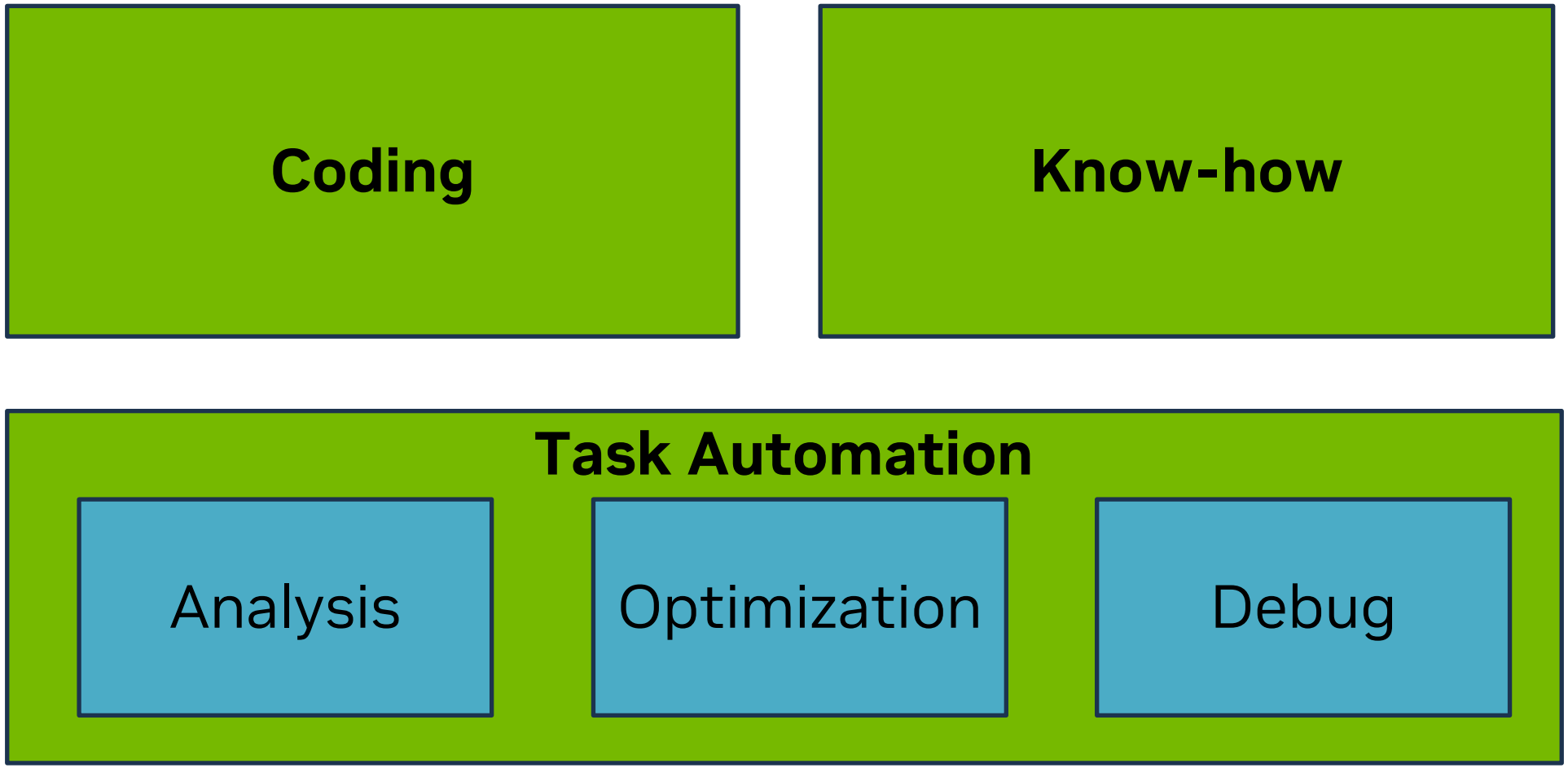
C.-T. Ho et al, VerilogCoder

LLM for Chip Design – Cambrian Explosion



Papers in ISLAD 2024 (LLM-Aided Design) : islad.org

Design Assistance – LLM



NVIDIA

Coding Assistance – EDA Script Generation

Generate scripts for specific tasks (VLSI)

Question:
write a code in *TOOL1* to print the number of flop cells in the give rectangle (0,0,100,100)

Answer:

```
# Sure, here is a TOOL1 code to print the number of
↪  flop cells in the given rectangle
↪  (0,0,100,100):

# This code gets the top_ref of the design
top_ref = env().get_top_reference()
# This code gets the cells of the rectangle in the
↪  top_ref
cells = env().layout().cells_in_rectangle( top_ref,
↪  rectangle(0, 0, 100, 100), hierarchical)
# This code gets the cells that are flop cells
flop_cells = cells.type("is_flop_cell")
# This code prints the number of flop cells
print(flop_cells.size())
```

Question:
Write a *TOOL2* tcl proc to find intersecting logic from pins '$pinA' to pin '$pinB'.

Answer:

```
#Sure, here is the code:
proc myfunc {pinA pinB} {
    # Get all the pins in the fanout of the given pin
    set fanout_pins [fanout -from $pinA]
    # Get all the pins in the fanout of the given pin
    set fanin_pins [fanin -to $pinB]
    # Get the intersection of the two collections
    set intersection [common_logic $fanout_pins
    ↪  $fanin_pins]
    # Return the intersection
    return $intersection
}
```

M. Liu et al, ChipNeMo: Domain-Adapted LLMs for Chip Design

NVIDIA

Know-how Assistance – Engineering Chat Bot

Answer questions about designs, infrastructures, tools, flows, HW domains, etc.

Question:
why do we add retiming stages?

Answer:
Retiming stages are added to improve the timing of the design and to account for the physical distance between different parts of the chip. By adding retiming stages, we can reduce the amount of time it takes for signals to travel between different parts of the chip, which can help to improve the overall performance of the design. Additionally, retiming stages can also help to reduce the amount of noise and interference that can occur on the signals, which can further improve the performance of the design.

Question:
How do I run multiple seeds of random tests using testgen framework?

Answer:
You can run multiple seeds of random tests using the Testgen framework by using the *rand_tgen* target. This target allows you to specify a variety of options for the stopping criteria of random runs using the same Testgen testlist(s).

```
nvmk -c . rand_tgen TESTLIST=my_testlist MON=1
↪  TGEN.LOOP=50
```

This command will run 50 iterations of the *my_testlist* testlist, each with a different seed. You can specify the number of iterations and the seed mode using the TGEN.LOOP and SEED options, respectively.

M. Liu et al, ChipNeMo: Domain-Adapted LLMs for Chip Design

NVIDIA

Analysis Assistance – Bug Report Analysis

Summarize bug report, predict task assignment

Prompt:
BugId : 002
Synopsis : Request power/current measurement on fabrics using fabric test module
Module : Silicon - DFT
Description :
Hi all,
We want to evaluate silicon current (power) consumed by fabric macros in a chip. This will help us correlate this power to pre-Si power estimates.
Here are the conditions we want to look at for this data:
Fabric test module activity: 0/25/50/75/100%
Corners:
typical_low_voltage
typical_high_voltage
We want to isolate fabric current (power) as much as possible, so we want to enable only the logic needed for fabric test modules and the fabrics themselves.
If this information can be broken down per module (there are 6 in the chip), that will help.
Severity : 5-Performance
Priority : 5-Normal
Comments :
#1 : Author : User A
Is this a requirement only for current or for future chips as well ?
#2 : Author : User B
There will likely be interest in future chips to perform this exercise as we push the boundaries of fabric power density for better area.
#3 : Author : User A
What is needed in this bug ?
@User B It is not clear.
Can you list the specific needs with timelines and priority

Technical Summary:
In NVIDIA's internal bug management system, a bug has been filed regarding the power and current measurements of the fabrics in the latest chip. The request is to evaluate the silicon current (power) consumed by the fabric macros in the latest chip and to correlate this power to pre-Si power estimates. The conditions for the measurements are the fabric test module activity at 0/25/50/75/100% and the corners are typical_low_voltage and typical_high_voltage. The data needs to be broken down per module (there are 6 in the latest chip). Only the logic needed for fabric test modules and the fabrics themselves need to be enabled.

Managerial Summary:
User A asks User B if the need for evaluating the silicon current of fabric macros in a chip is only for that specific chip or for future ones as well. User B replies that there will likely be interest in future chips to perform this exercise as they push the boundaries of fabric power density for better area. User A then asks what is needed in this bug and asks for specific needs with timelines and priority.

Task Assignment:
User B

M. Liu et al, ChipNeMo: Domain-Adapted LLMs for Chip Design

Closing Thoughts

- BO and RL continue to drive better PPA for chip design
- Generative AI trained on optimized data to speed up traditional optimizations by orders of magnitude
- LLM models and agents to significantly improve chip design productivity by providing design assistance as chatbots and copilots and automating more manual design tasks.
  - The importance of reliable and efficient inference infrastructure.
- Call for action: need more datasets and benchmarks: VerilogEval, FVEval, LLM4HWDesign, …

NVIDIA

