AI-Assisted Chip Design Tutorial

HotChips

Stelios Diamantidis
Synopsys Distinguished Architect

25 August 2025

AI-driven Optimization for Chip Design

1. Motivation – Why AI for Optimization

2. The Reinforcement-Learning Optimization Paradigm
   – Search spaces, acquisition functions, metrics/KPIs, pareto fronts, learning

3. Applications of RL-driven Optimization
   – Physical design, micro-architecture, search-based verification, test, analog, 3D exploration

4. Augmenting RL with GenAI – A World of Opportunity
   – Optionality vs. optimality, evolution of human-compute i/f, data abstractions

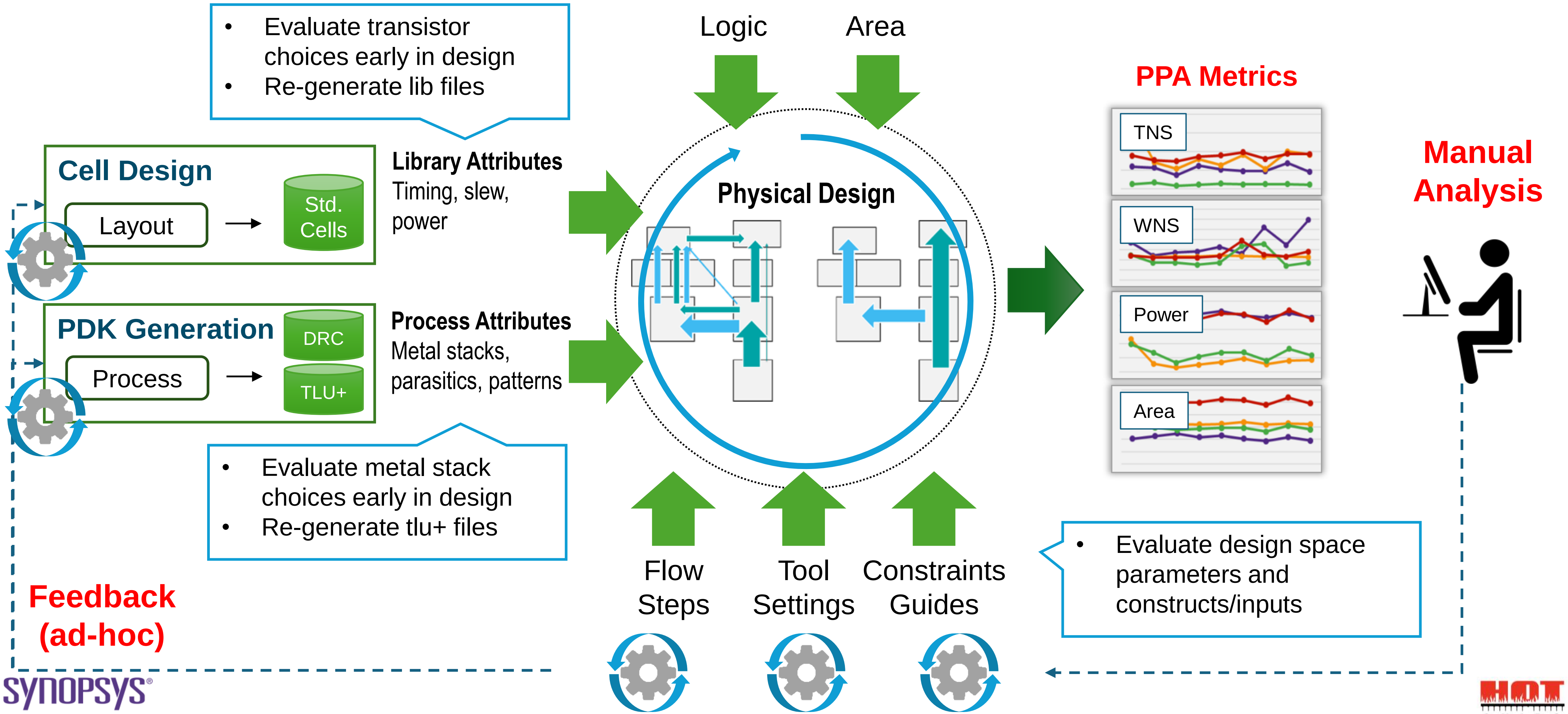
Disclaimer

- This is a technology tutorial
- Several examples have been drawn from Synopsys research in AI
- The capabilities presented may not be indicative of Synopsys products
- For product-related information, please contact Synopsys sales

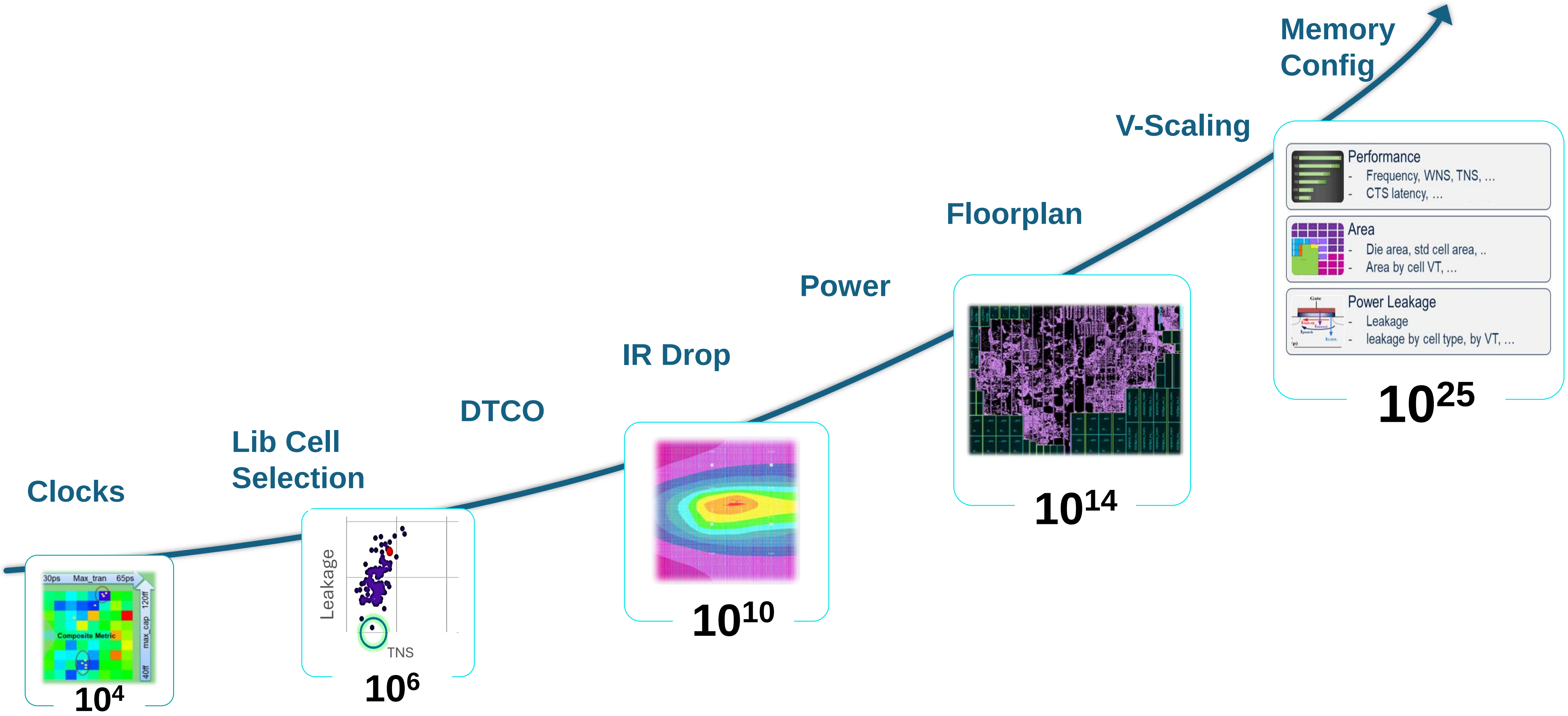
Motivation – Why AI for Optimization

AI-Assisted Hardware Design Tutorial – HotChips 2024

Chip Design: A Near-Infinite Problem Space

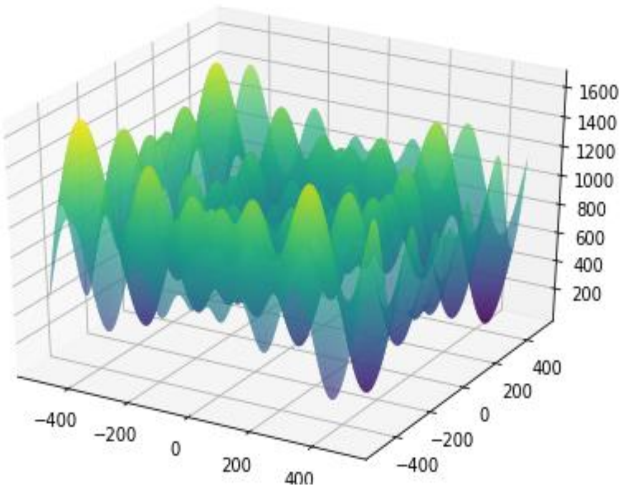


Design Complexity Grows Exponentially



Implication of Design Complexity

Quality of Result

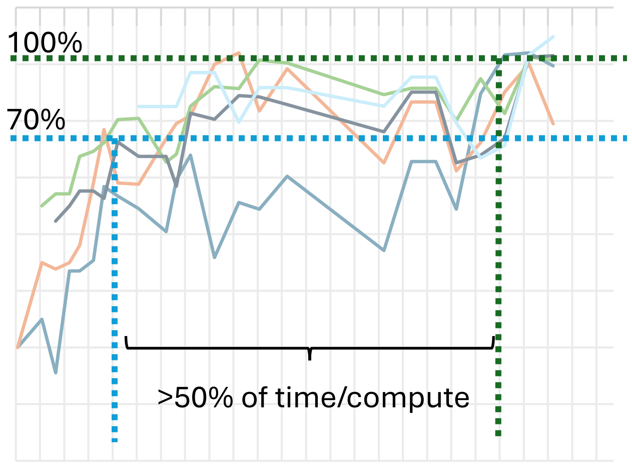


Discontinuous solution space

Noisy, non-convex, non-differentiable

Hard to break out of local minima

Throughput

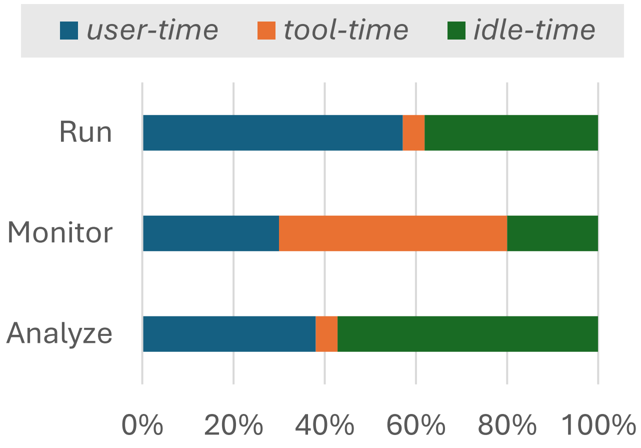


Hard to identify causal relationships

Only evaluate few variables at a time

Long latency (up to several days)

Cost

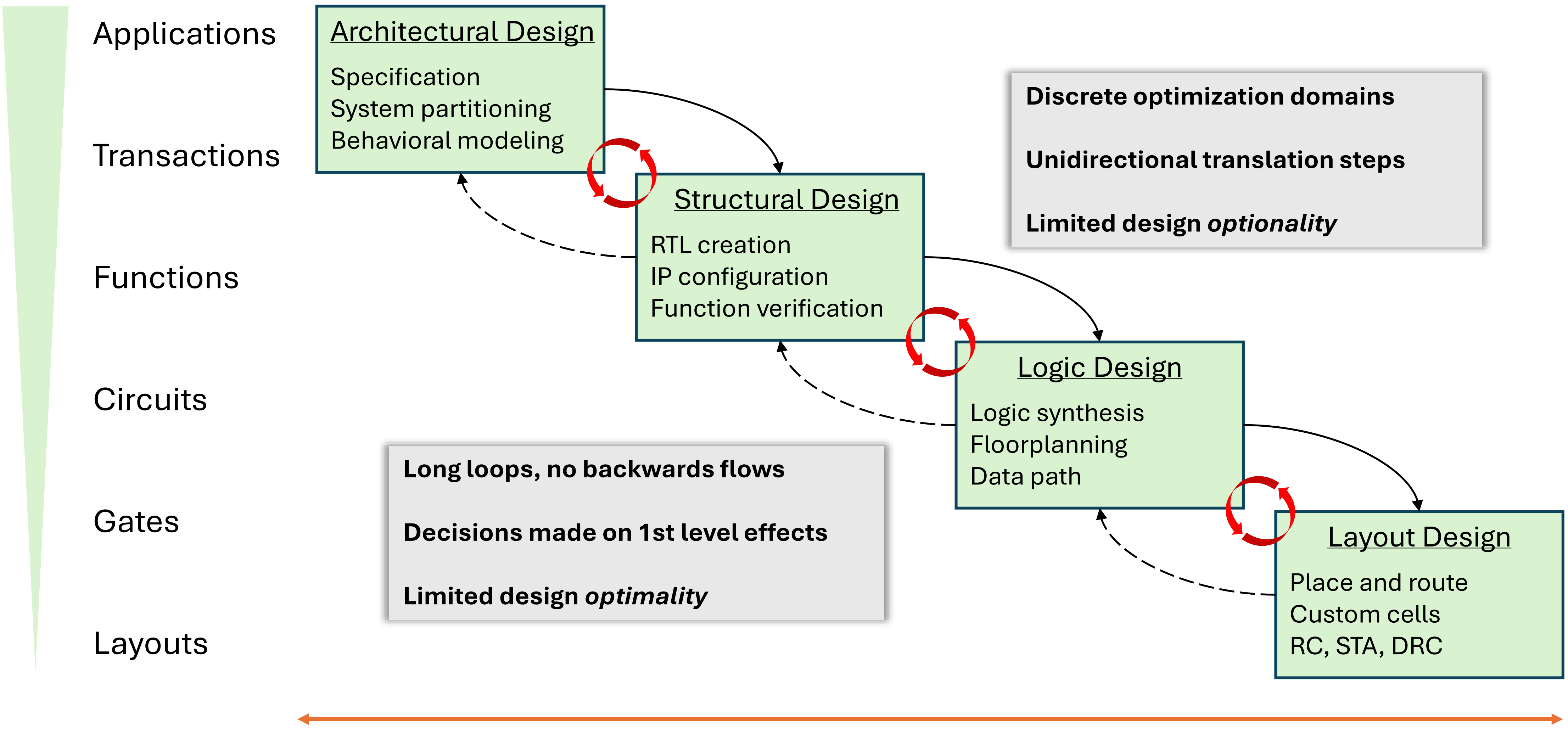


Underutilized compute resources

Limited reuse across projects

Decision fatigue!

A Cascade of Intractable Problems

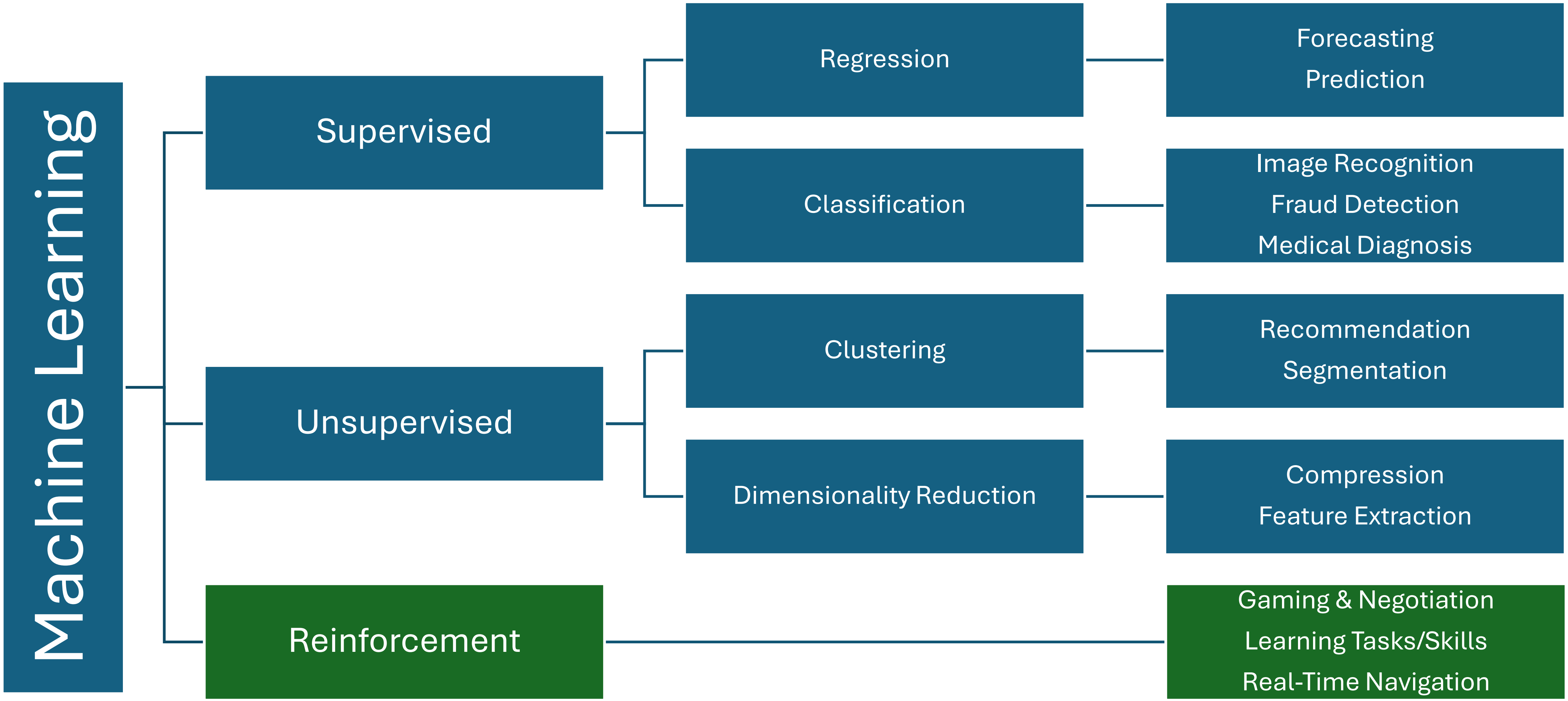


10s-100s of engineers, 18-24 months of development

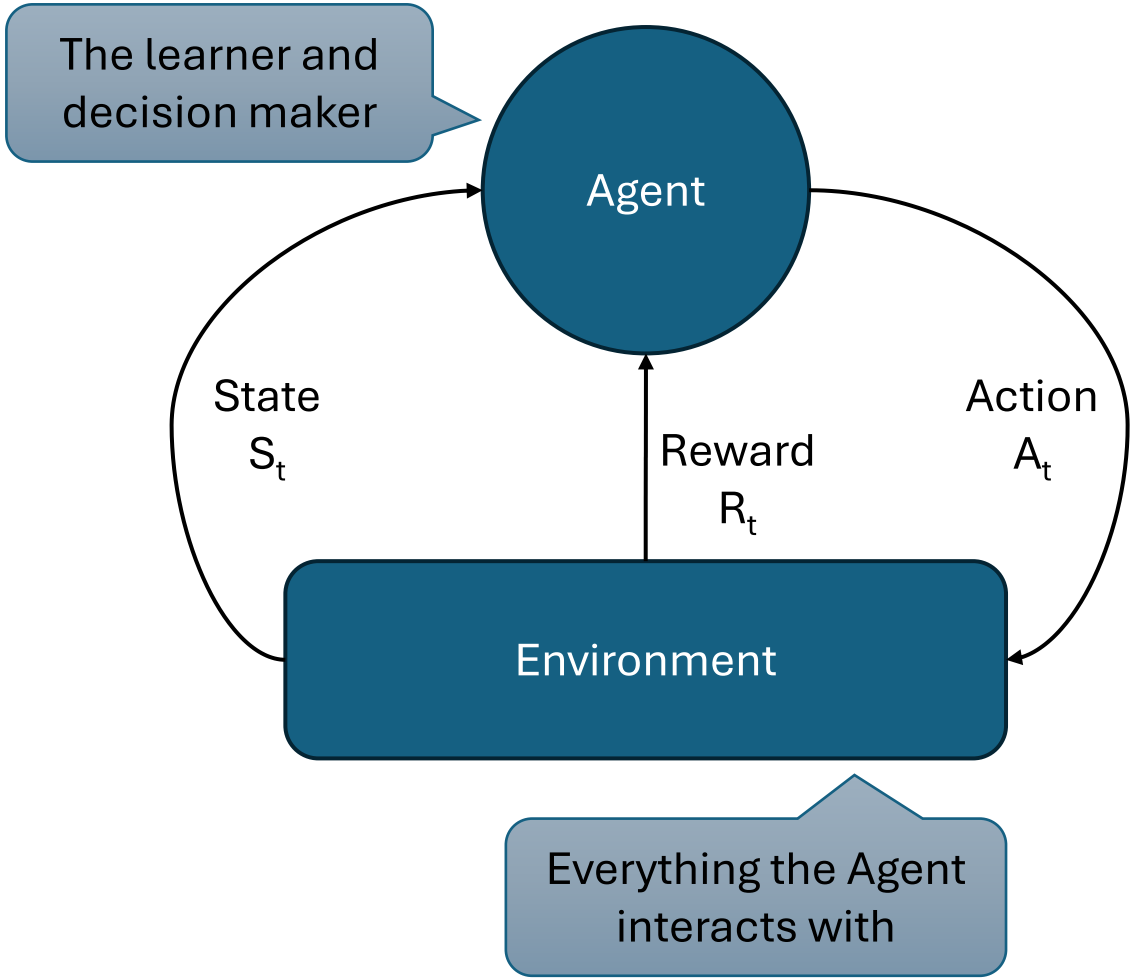
The Reinforcement-Learning Optimization Paradigm

AI-Assisted Hardware Design Tutorial – HotChips 2024

What is Reinforcement Learning (RL)?



How Does Reinforcement Learning Work?



https://www.synopsys.com/glossary/what-is-reinforcement-learning.html

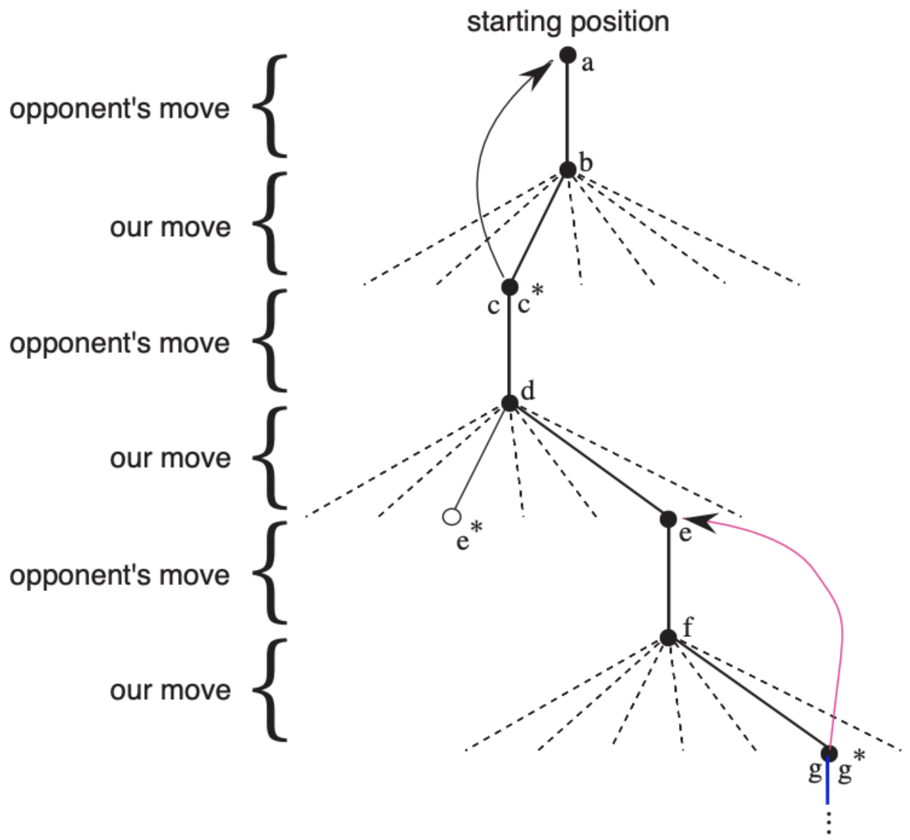
- The Agent interacts with the environment to sample trajectories of states & rewards:
    1) The Agent observes the environment's state S_t
    2) The Agent selects an action A_t and applies it to the environment
    3) The Agent receives a reward R_t
        - Goal is to maximize this reward <u>over time</u>
    4) A new state S_{t+1} is entered

- Generally, the Agent implements a mapping from states to probabilities of possible actions
- RL algorithms can be model-based or model-free (twin)
    - Value-based: Estimate value function given enough trajectories (SARSA, Q-learning)
    - Policy-based: Directly estimate optimal policy (Monte-Carlo, deterministic policy gradient)

SYNOPSYS®

HOT CHIPS

Benefits of Reinforcement Learning



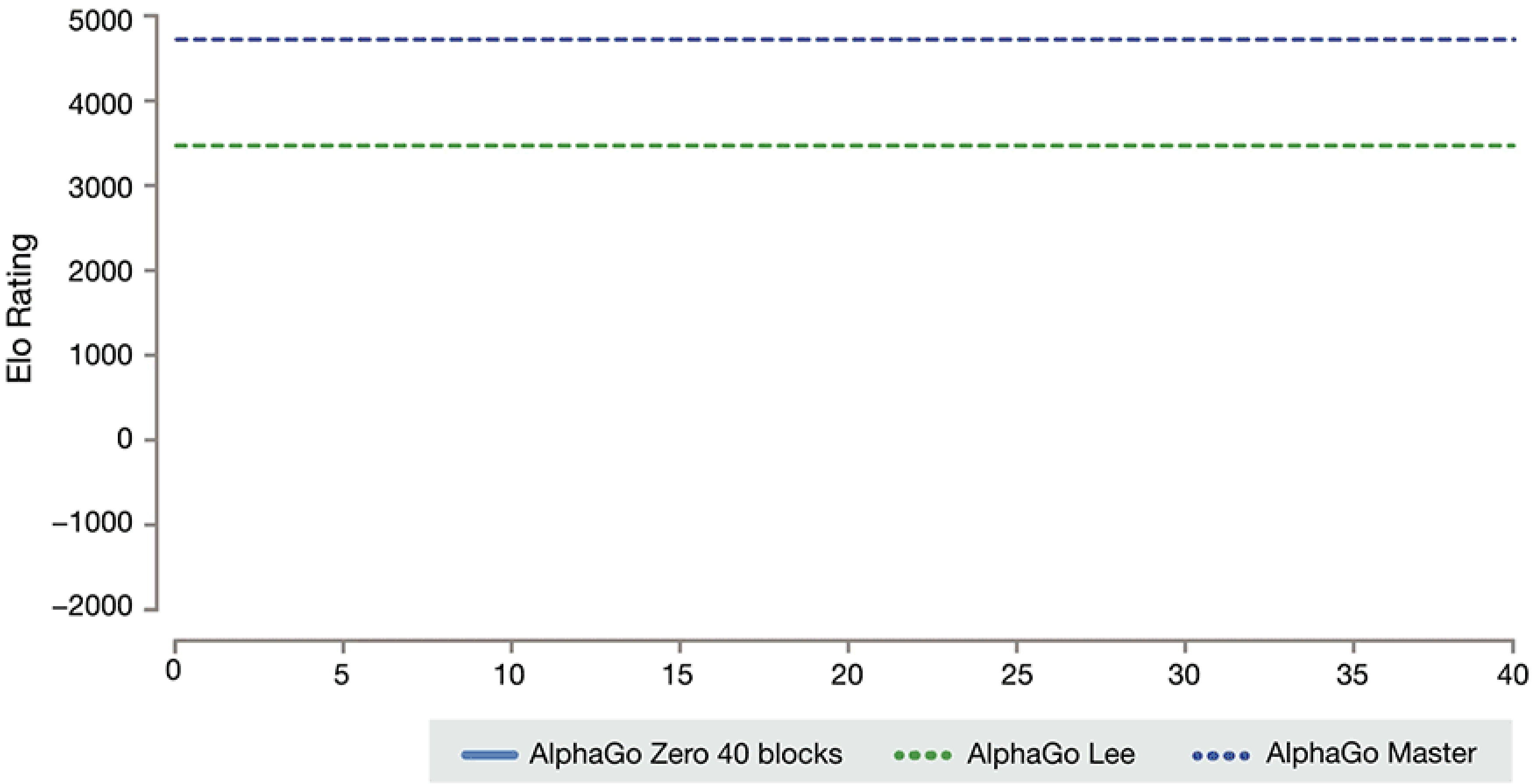
Citation. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). The MIT Press.

- Focuses on the problem as a whole
  - RL understands the goal, and can trade off short-term rewards for long-term benefits
- Does not need a separate data collection step
  - Training data is the Agent's experience, not a separate set established a-priori
- Works in dynamic, uncertain envs
  - RL is inherently adaptive and built to respond to changes in the environment

RL can seek a long-term goal while exploring various possibilities autonomously

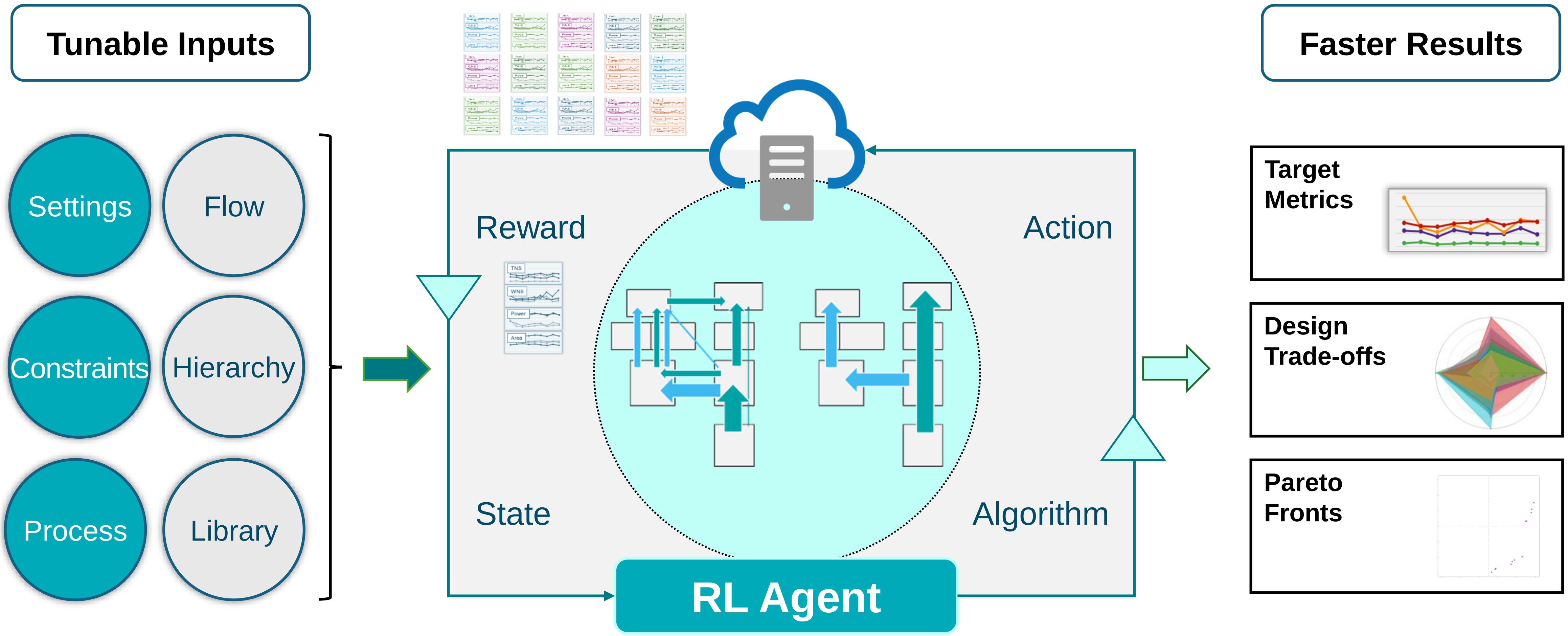
RL Example: Learning to Play GO

DeepMind AlphaGo goes from zero to world champion in 40 days

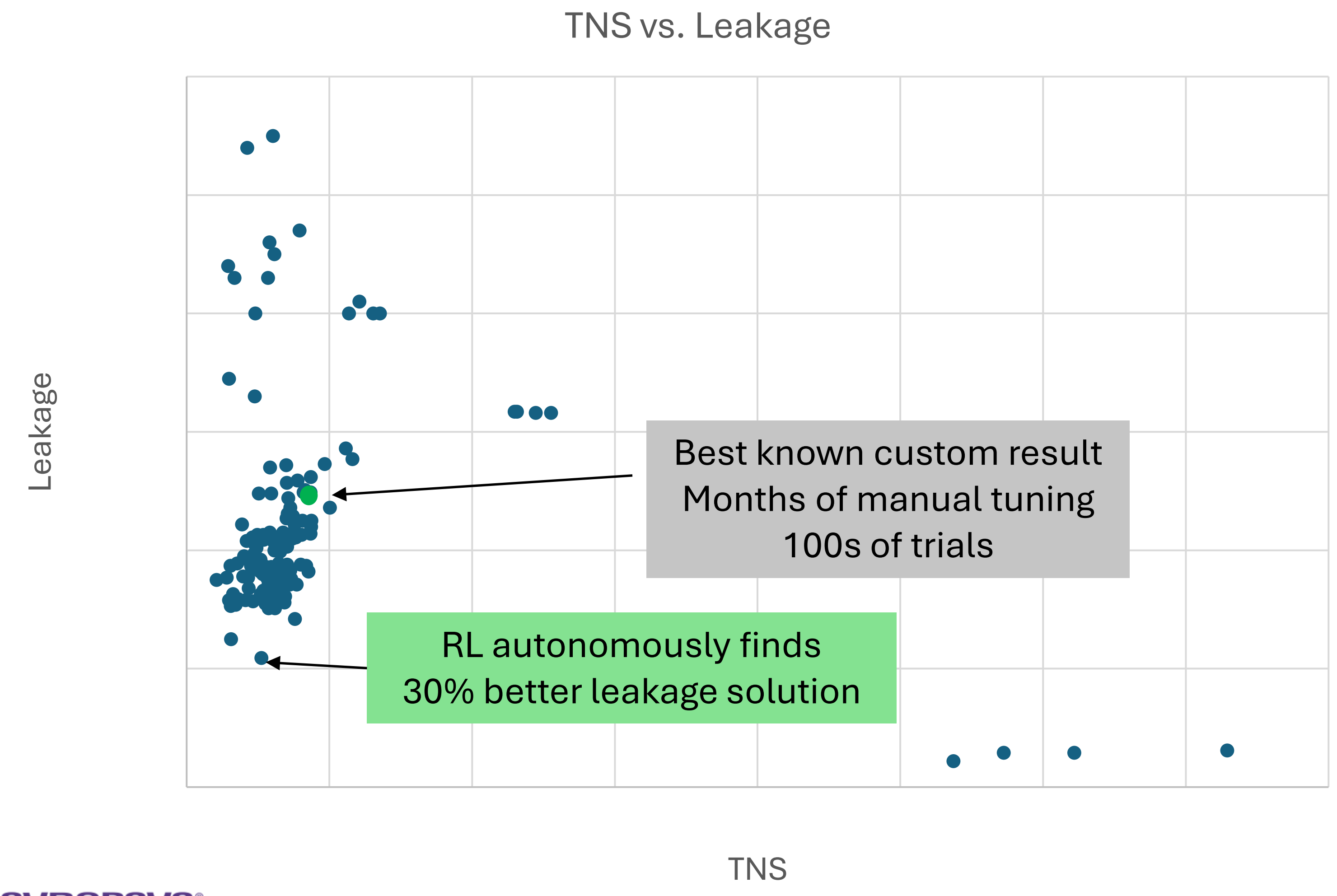


Example: https://deepmind.com/blog/alphago-zero-learning-scratch/

Applying RL to Chip Design Problems



AI-Assisted Design Search



Problem Statement:

Achieve lower leakage while maintaining timing

Search Space

- Design, tool, flow parameters
- Library cell parameters

Objectives (prioritized)

- Leakage
- TNS
- Secondary (e.g. DRC etc.)

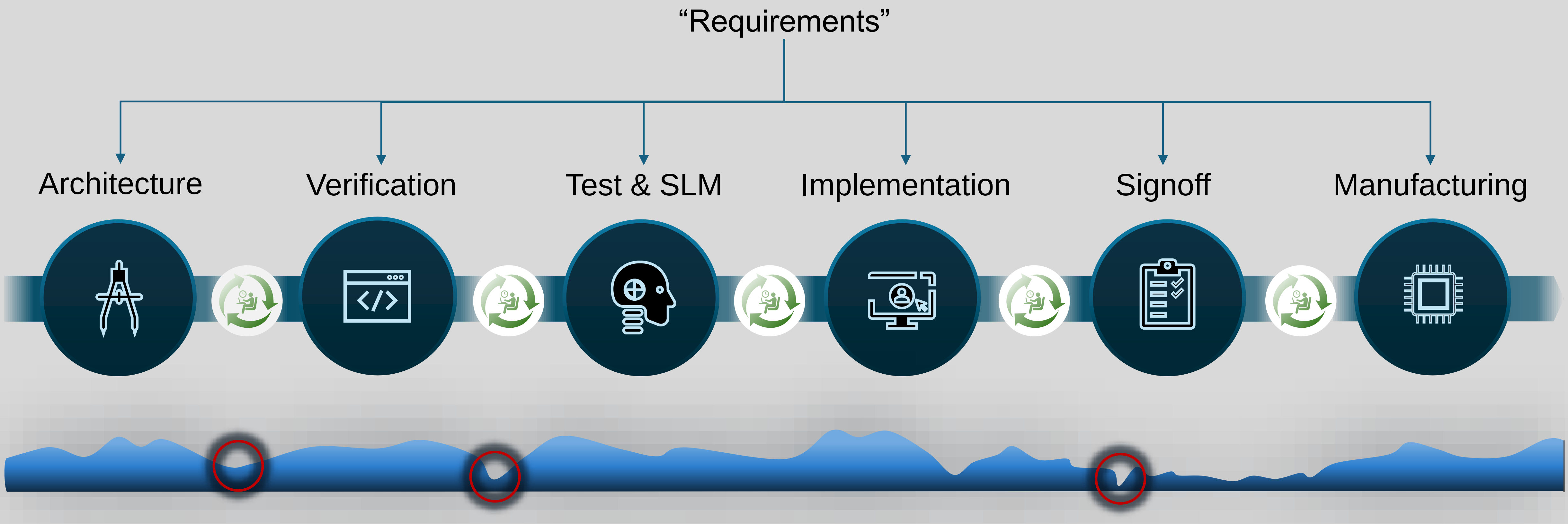
SYNOPSYS®

HOT CHIPS

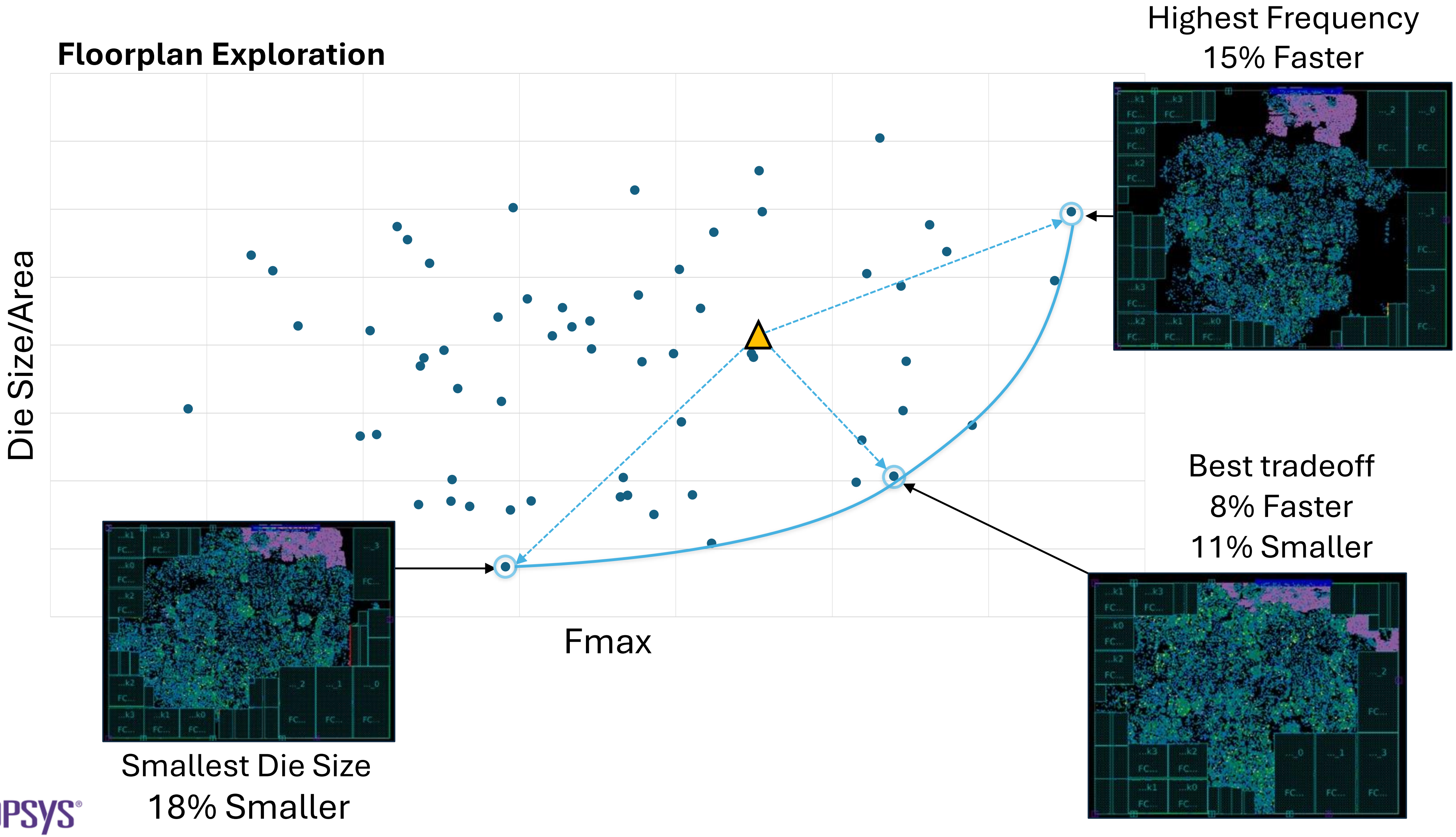
Applications of RL-driven Optimization

AI-Assisted Hardware Design Tutorial – HotChips 2024

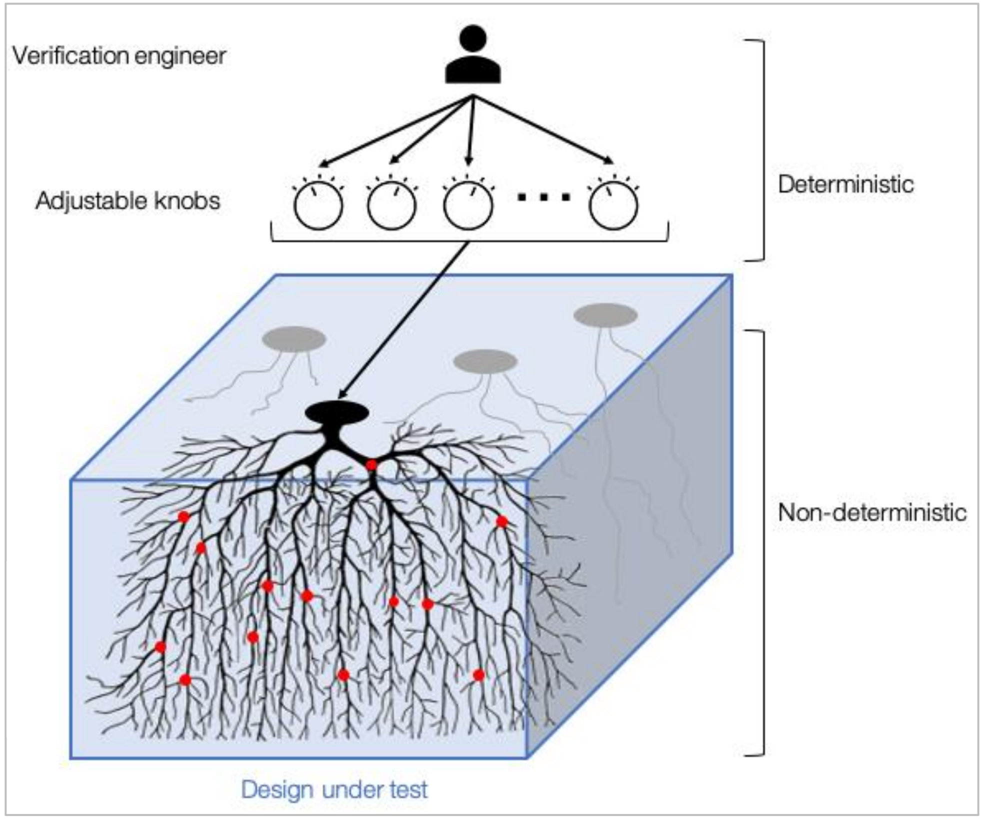
Opportunities to Apply RL Opt. Throughout the Flow



AI-Assisted Digital Implementation



AI-Assisted Verification



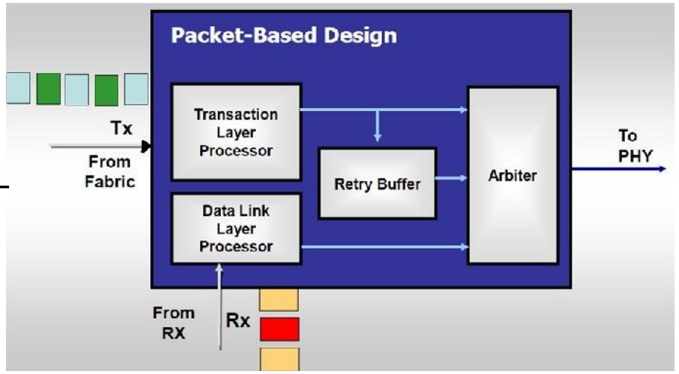
https://community.arm.com/arm-research/b/articles/posts/efficient-bug-discovery-with-machine-learning-for-hardware-verification

Test Strategy (AI Agent)

Goal: Learn sequences of actions that lead to bug detection

Next state
Reward/penalty for the op choice

Next op
Maximize likelihood of transition to buggy state



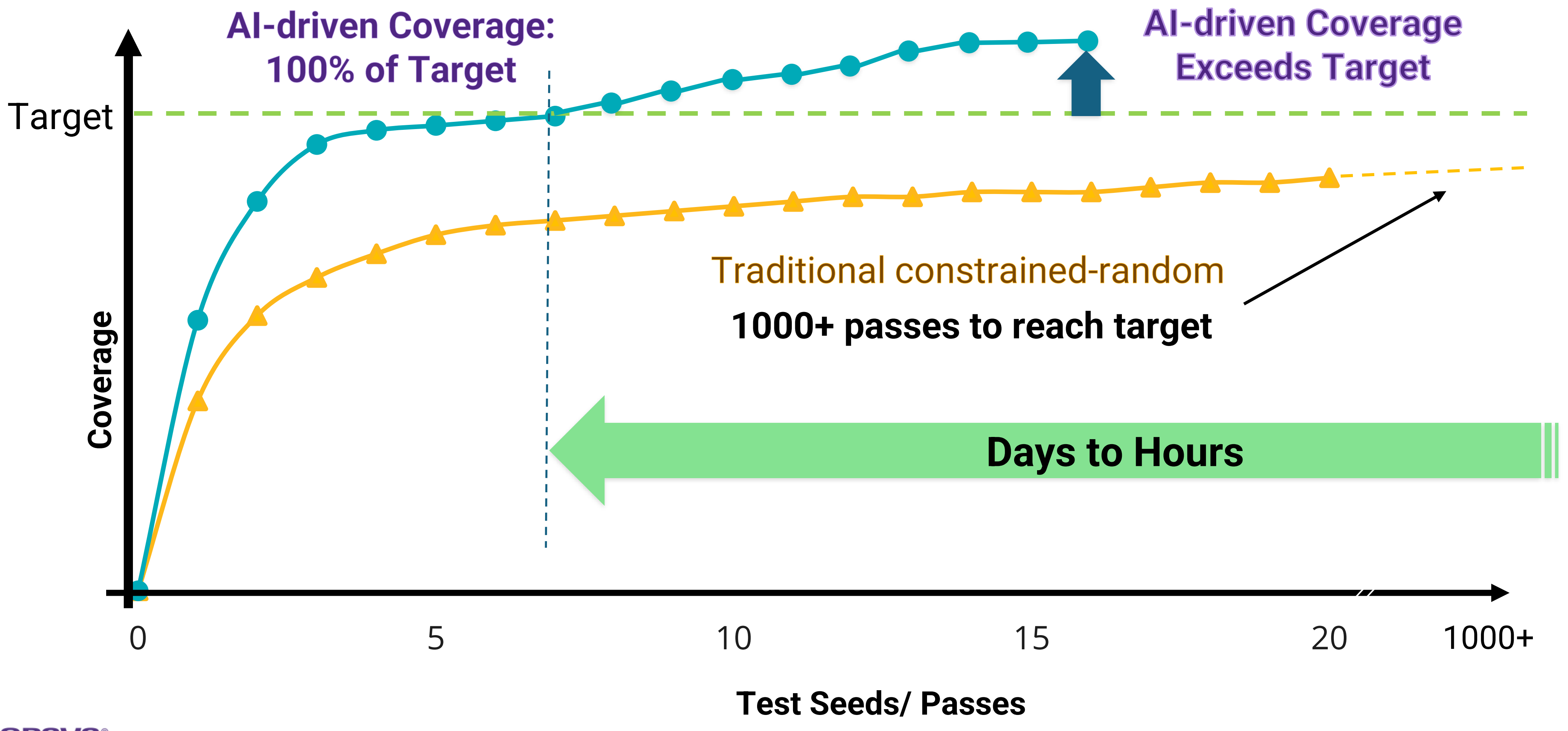
Using RL to optimize the verification process

SYNOPSYS®

HOT CHIPS

Example: Scheduling Highest ROI Tests

Enabling Faster Time To Closure



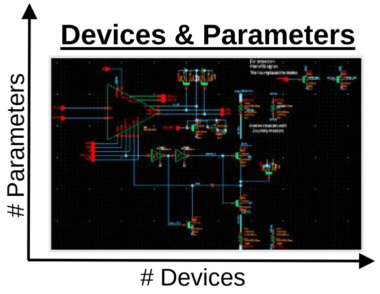
SYNOPSYS®

HOT CHIPS

And Many More Applications..

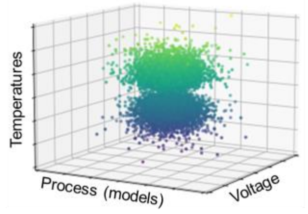
Circuit Optimization

Devices & Parameters



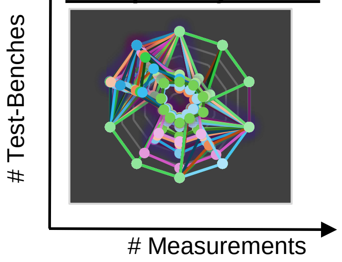
X

1000’s PVT Corners

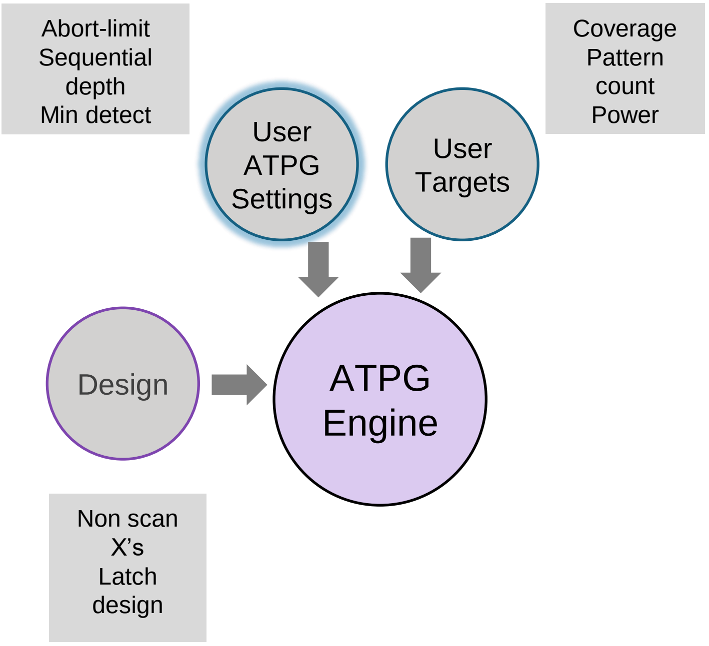


X

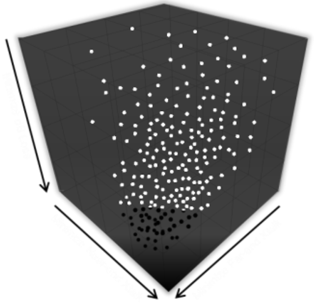
Multiple Objectives



Test/ATPG



3D Integration

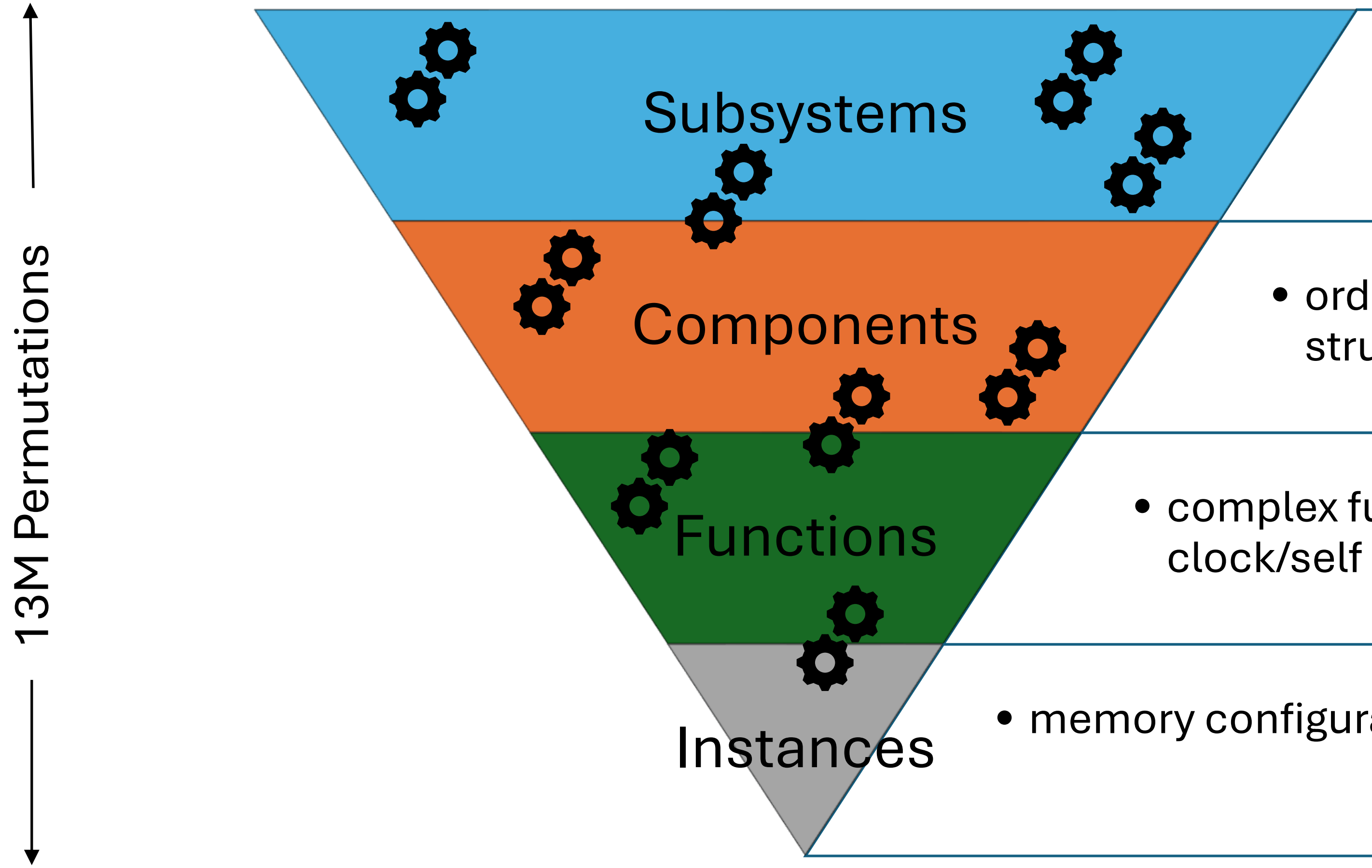


Autonomous multi die exploration

SYNOPSYS®

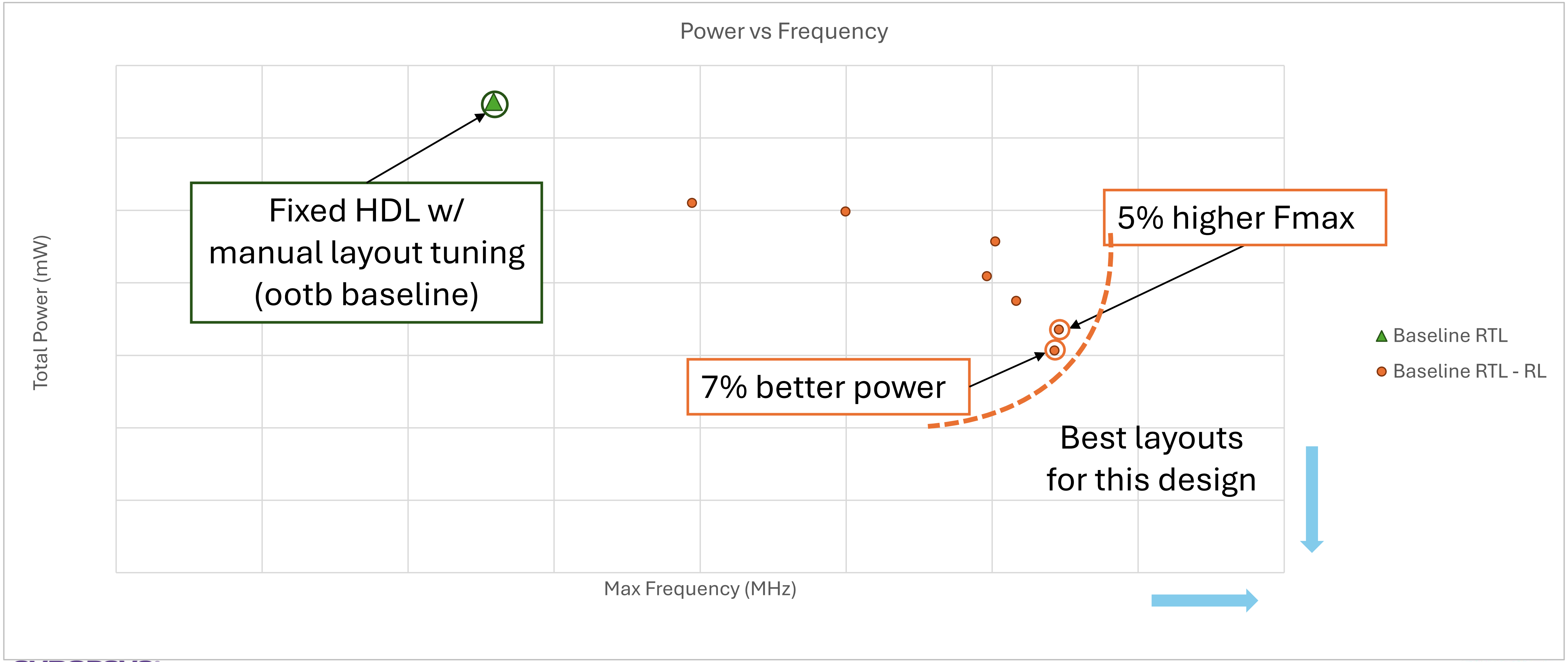
HOT CHIPS

How About Optimizing Across Design Abstractions?

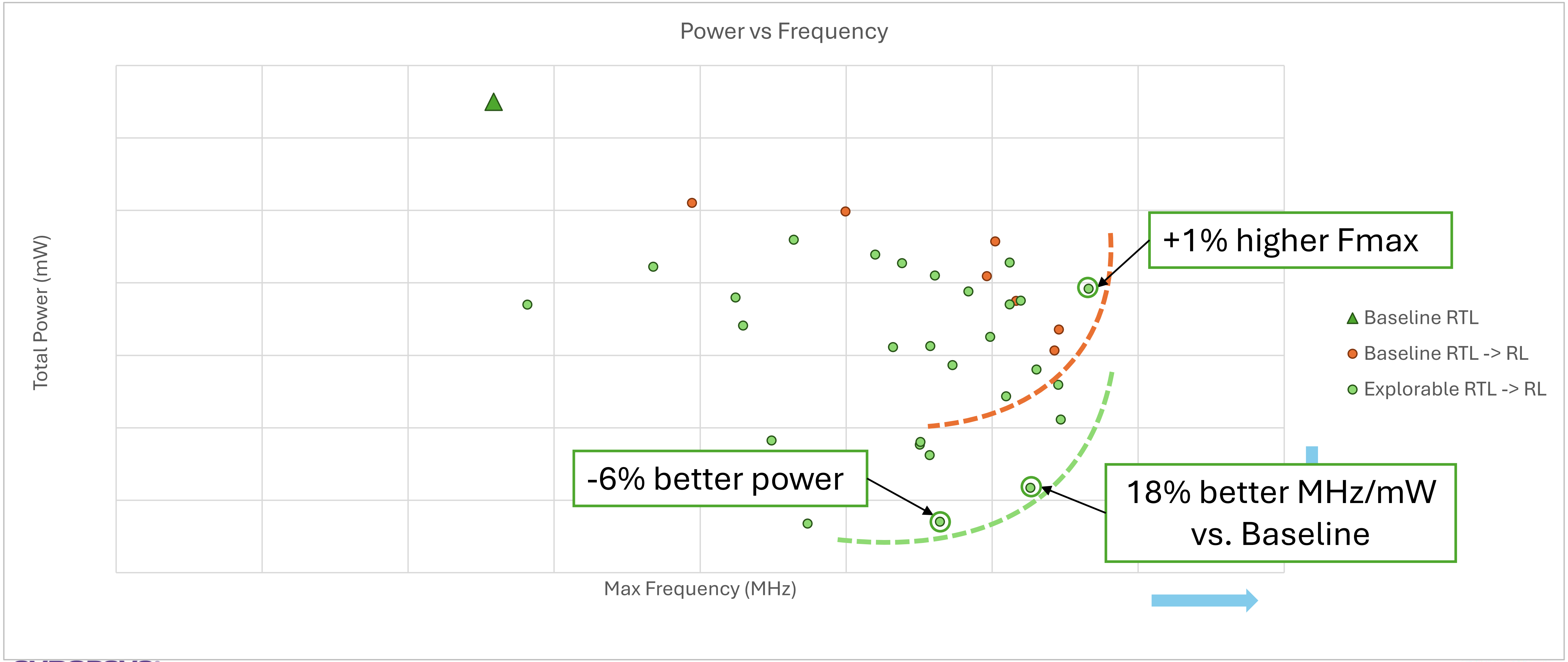


- ISA, #ALU, FPU, issue ports, caches
- order of operations, loop structure/content
- complex function mapping, clock/self gating
- memory configuration, cell mapping

Single-Abstraction: RL-based Layout Opt.



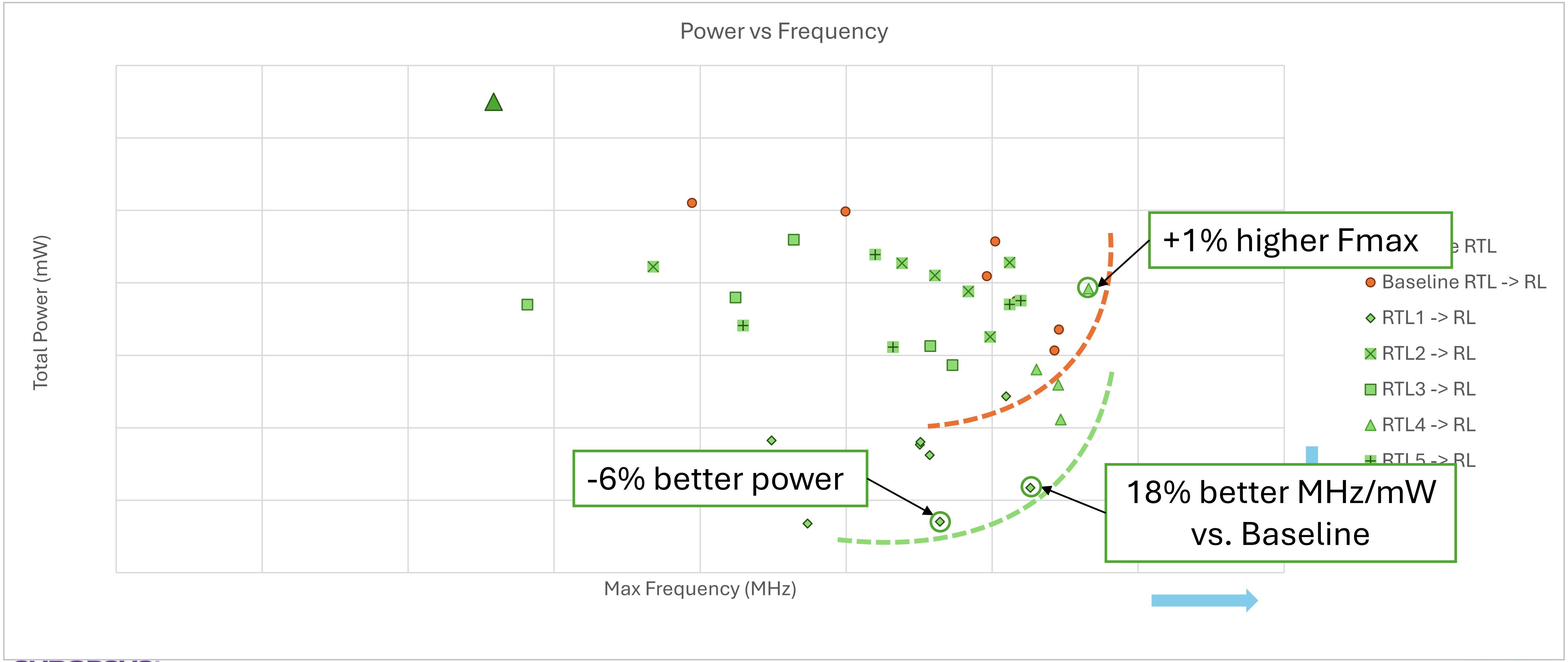
Multi-Abstraction: Functions-to-Layout Opt.



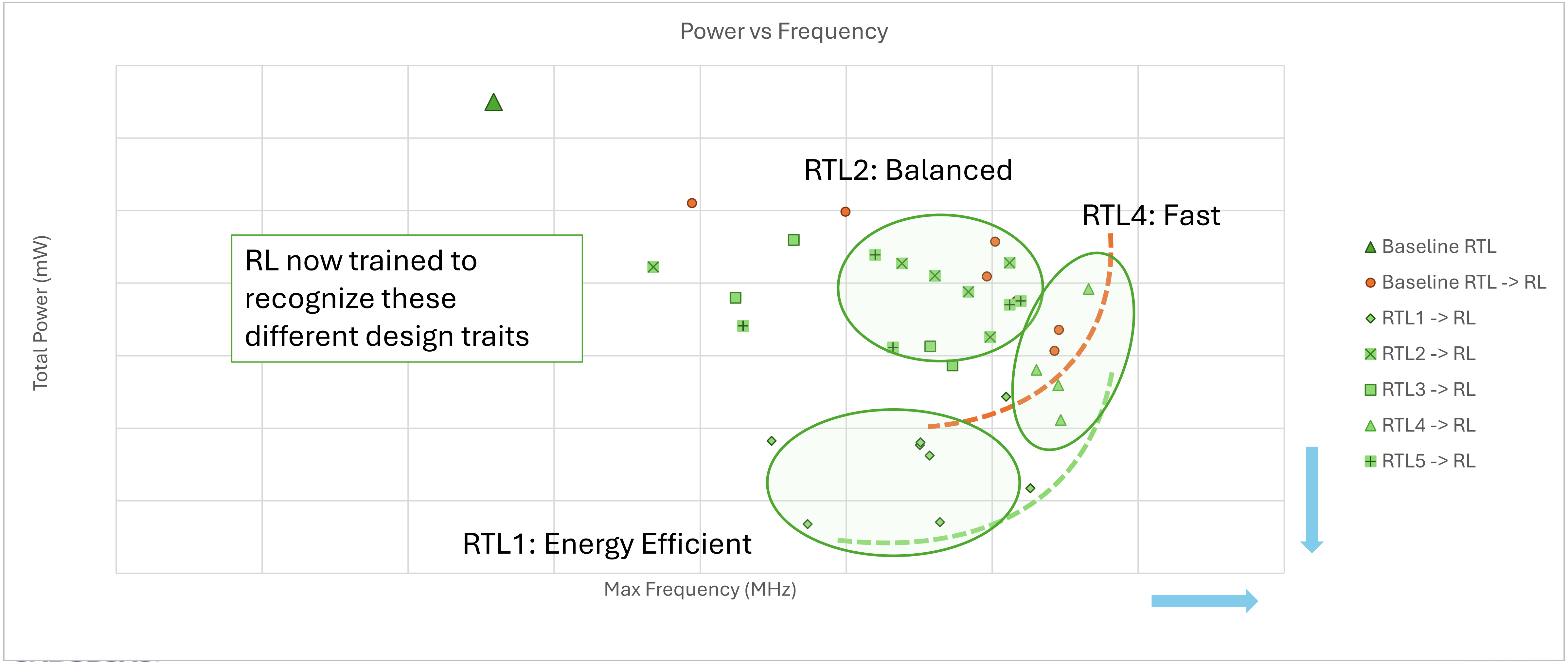
SYNOPSYS®

HOT CHIPS

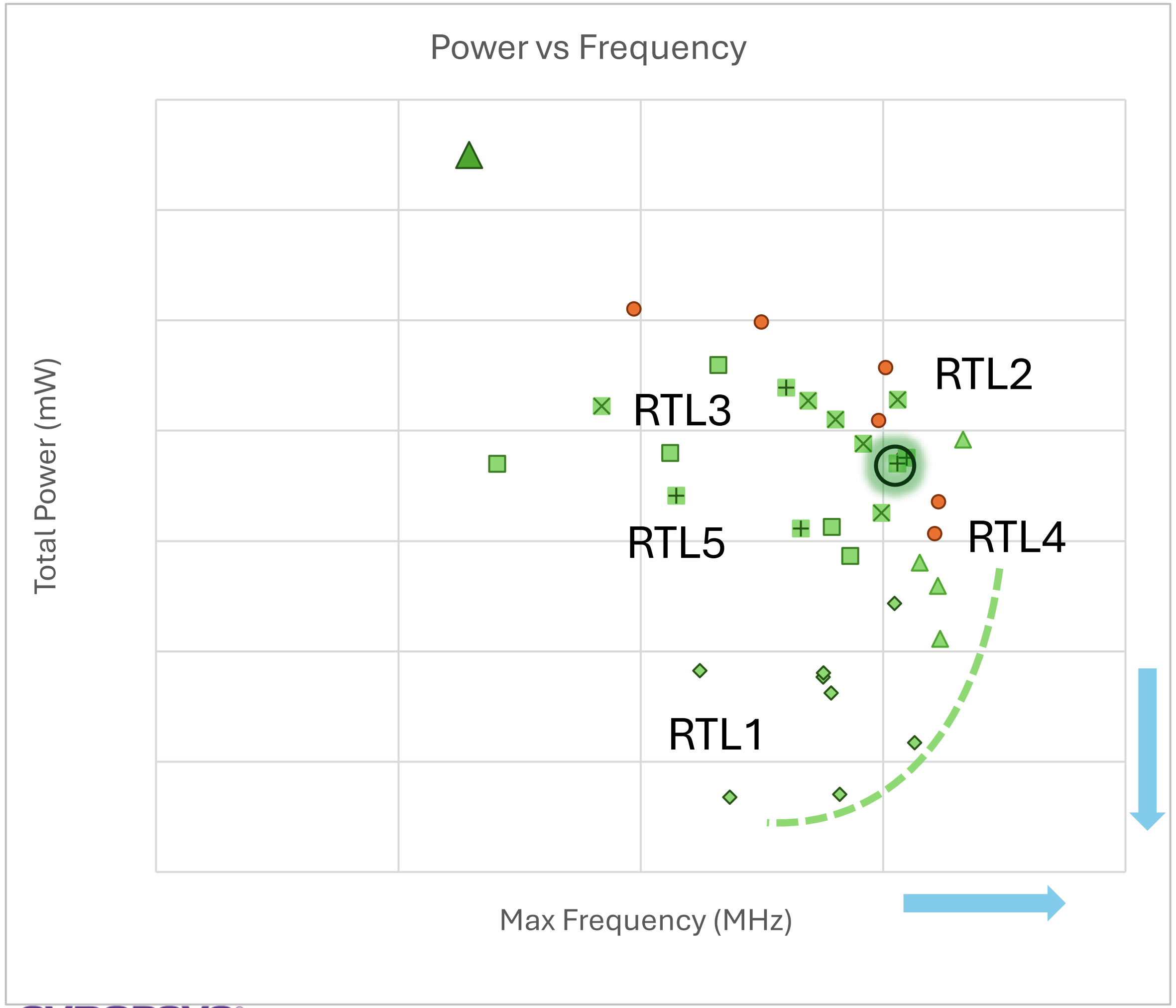
Top-20 Results: 5 Different Design Configs

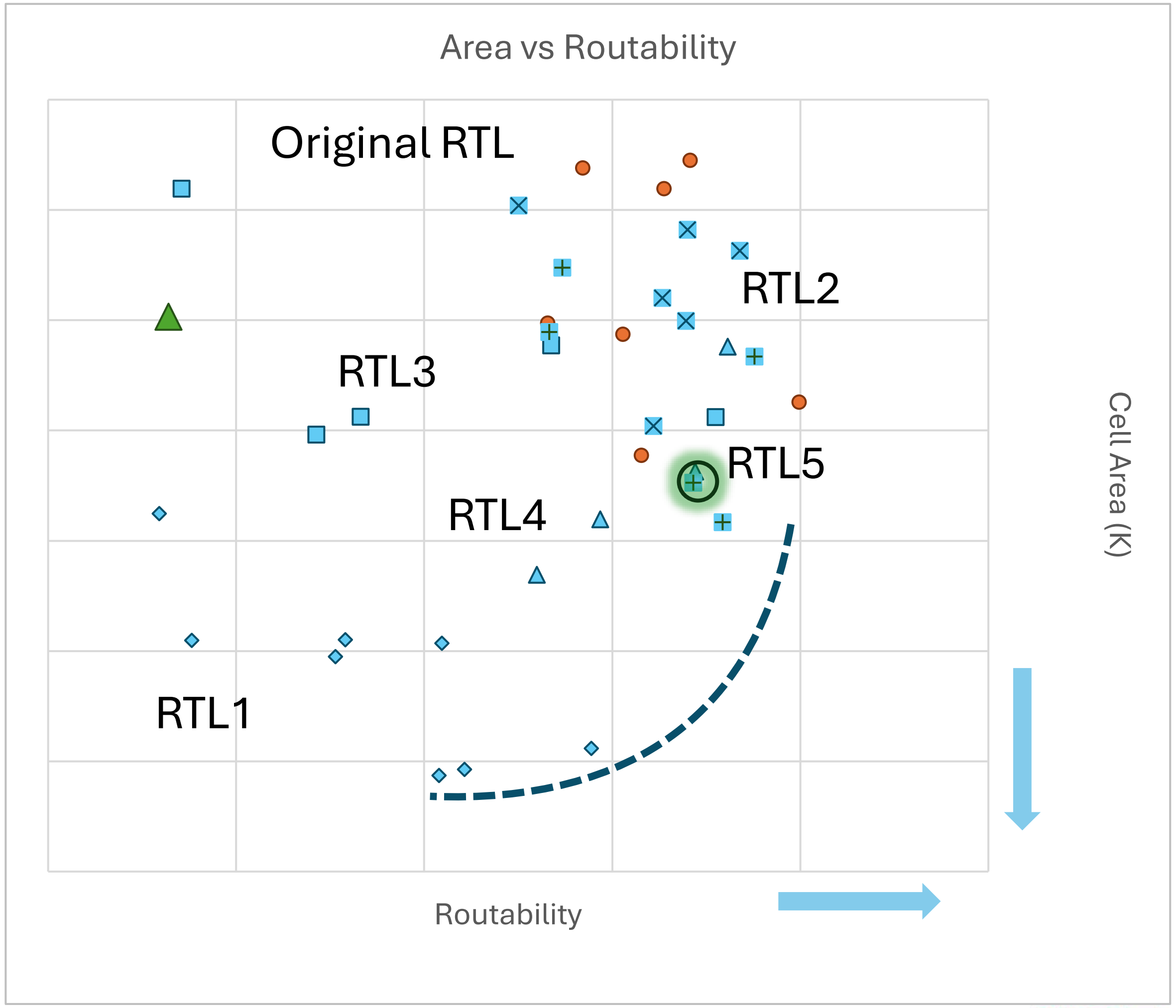


Functions with Different Layout Characteristics



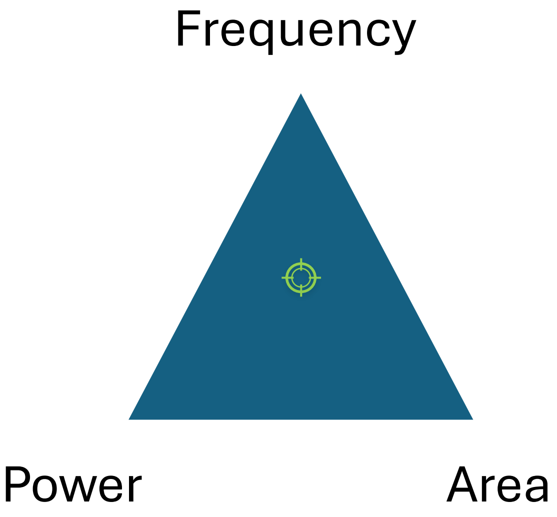
So, Which Design Variant is the Best One?



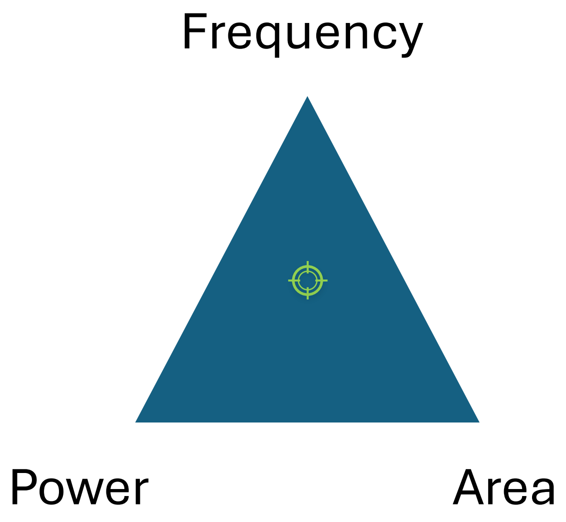


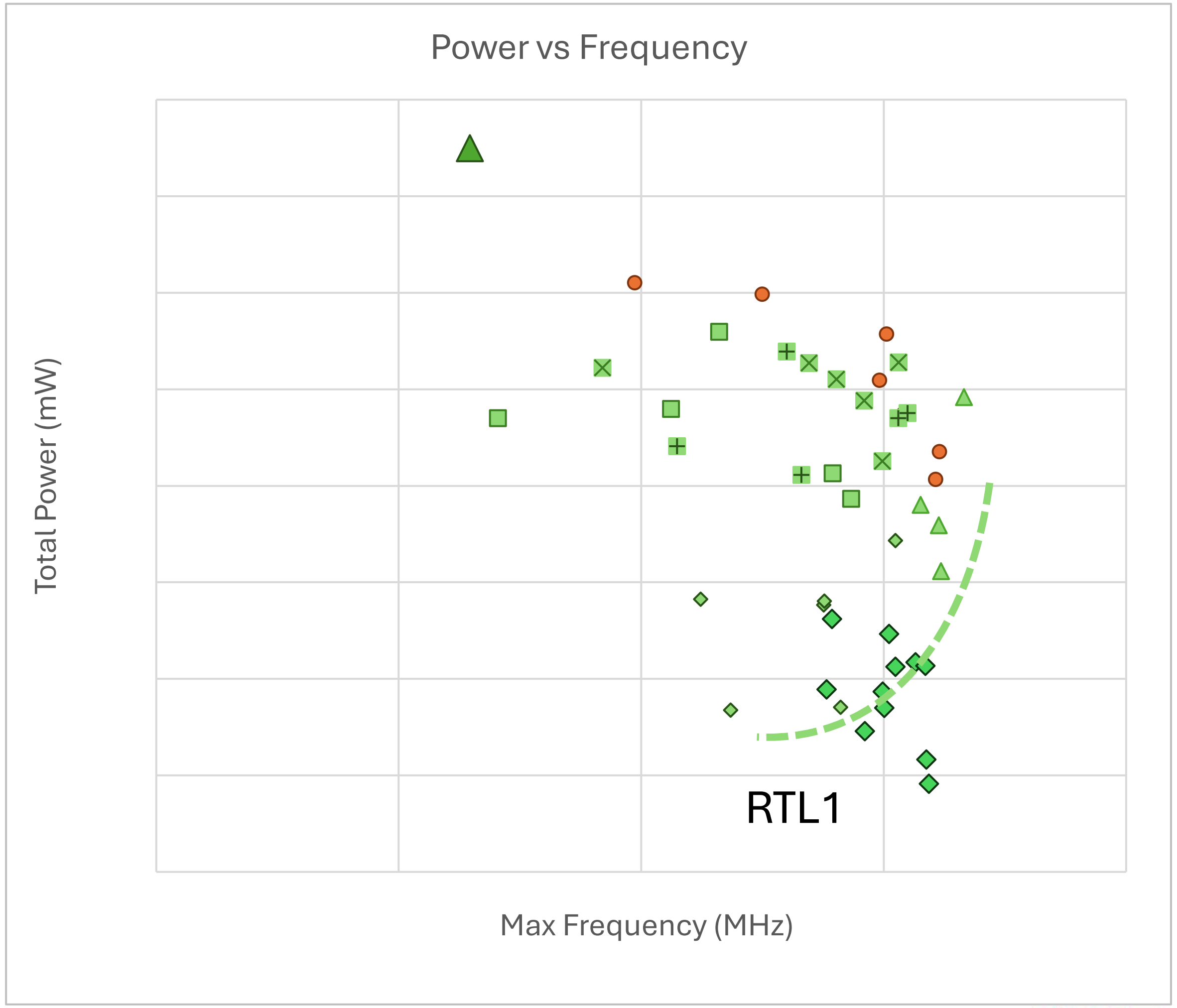
Recentering Design Functions-to-Layout

Using AI to quickly traverse problem space towards ‘learned’ solutions

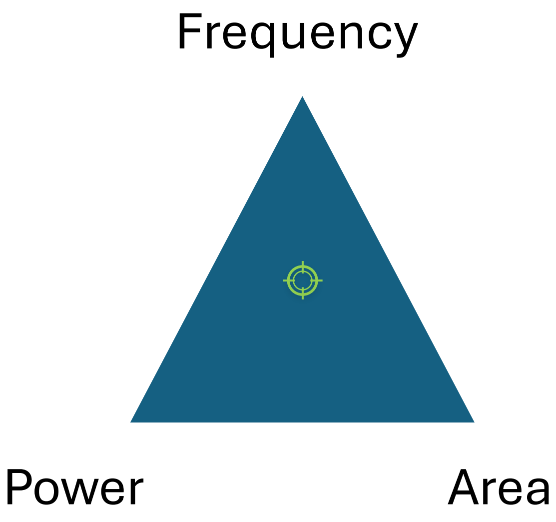


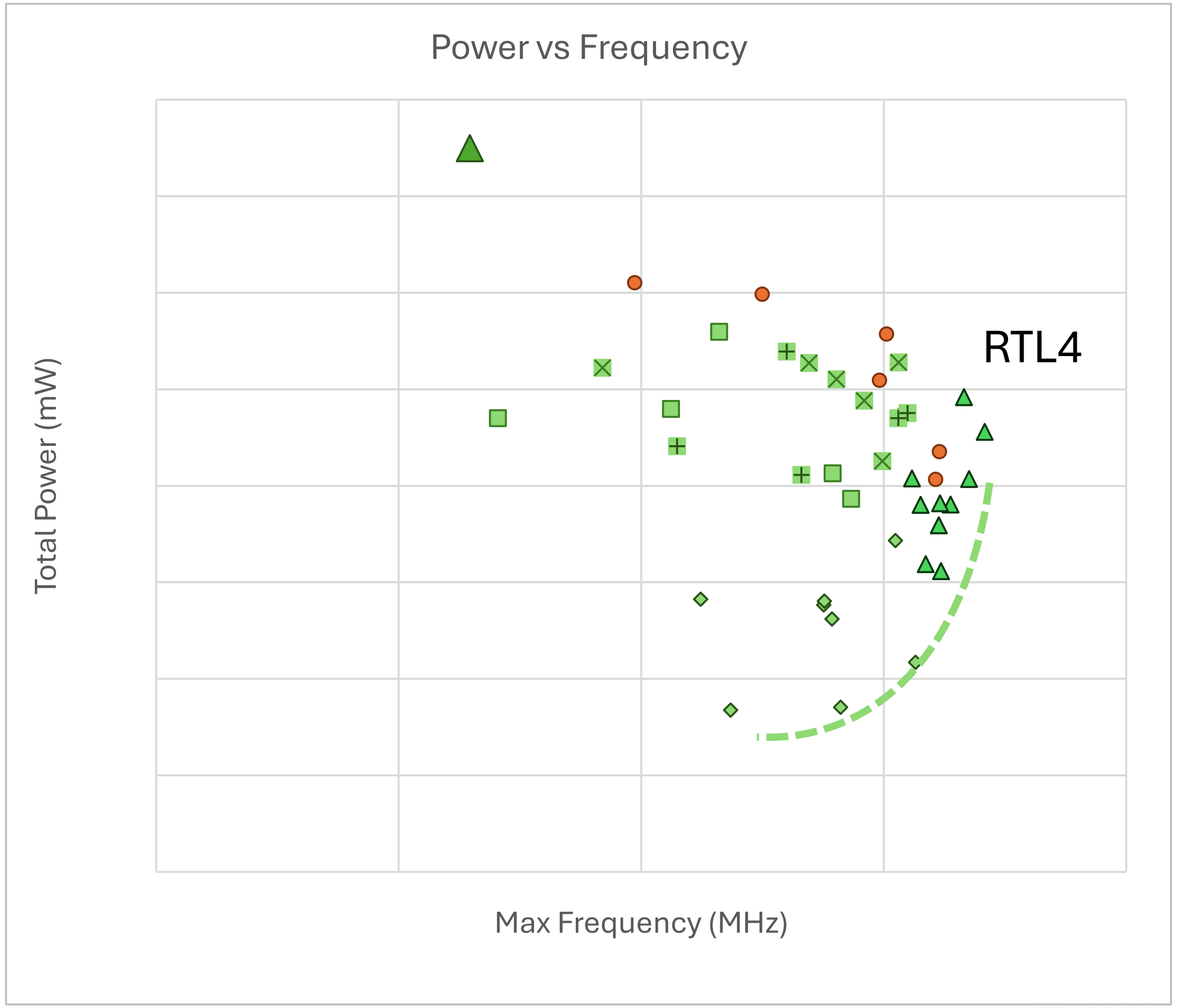
1) Objective: Energy Efficiency



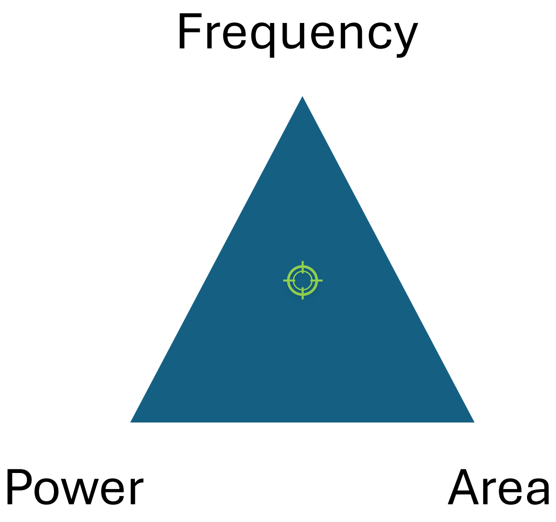


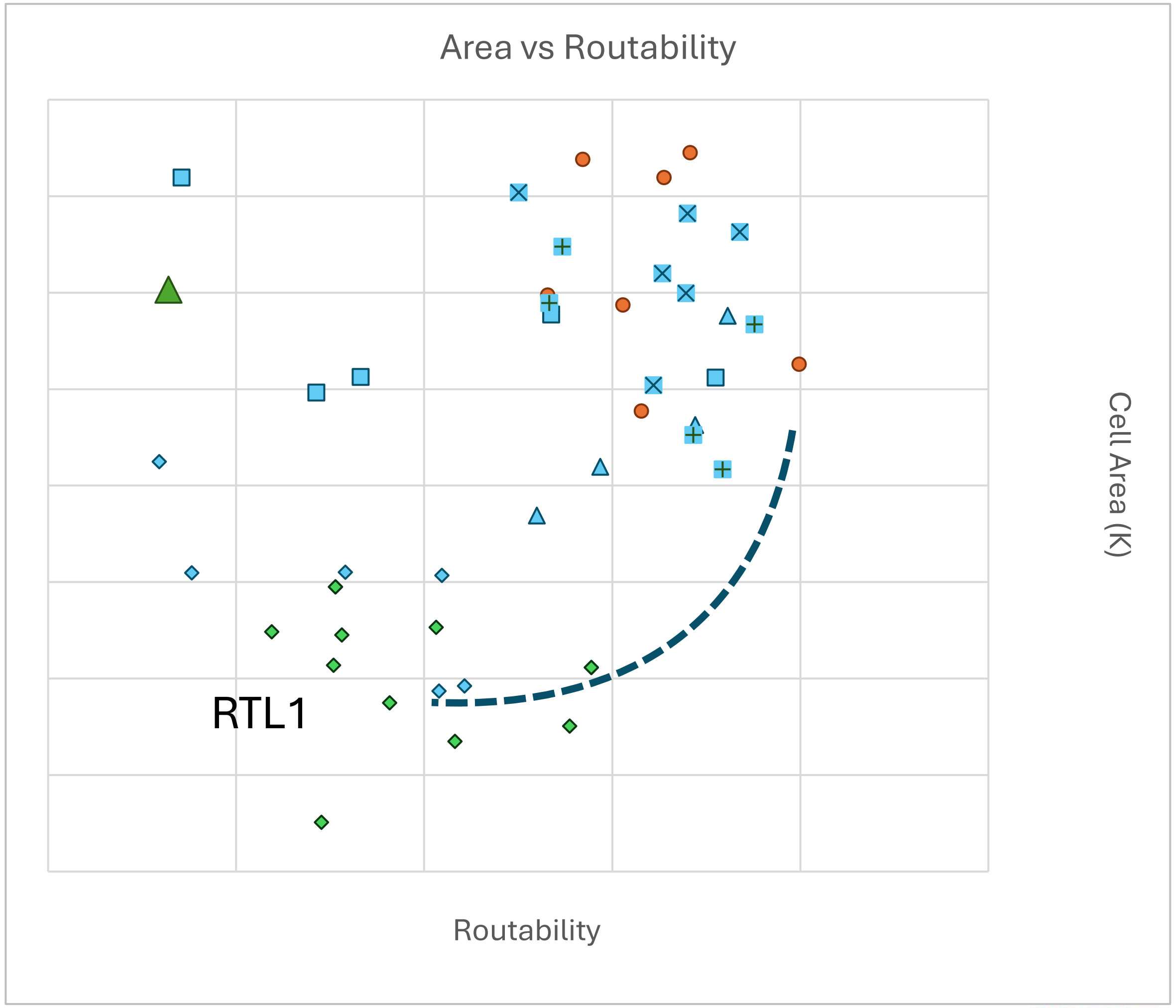
2) Objective: Performance





3) Objective: Area





Limitations of the RL-based Opt. Paradigm

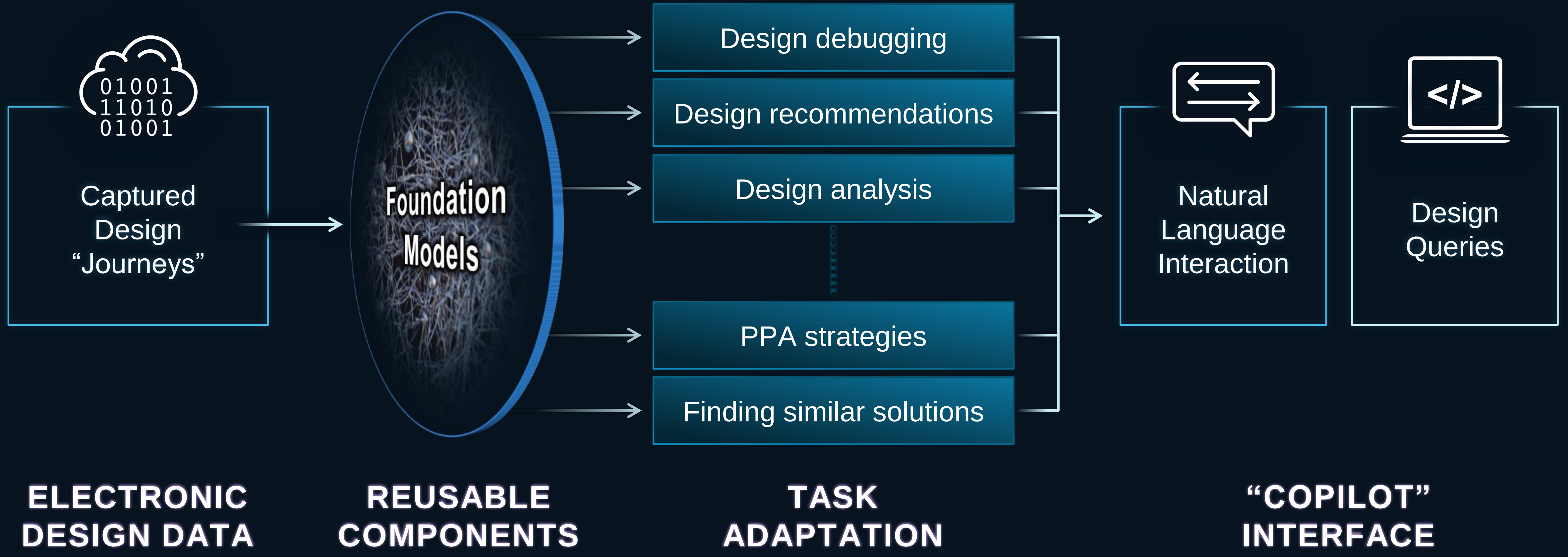
1. Creating design variants is a high-effort task
    - Verifying even a single version of a design is difficult, how to scale?

2. Evaluating design variants can be slow
    - Typically involves synthesis, P&R, timing/power/IR/etc. analysis

Augmenting RL with GenAI – A World of Opportunity

AI-Assisted Hardware Design Tutorial – HotChips 2024

2nd Wave of AI : Generative Models Coming into Play



Ref: Adapted from the Stanford Institute for Human-Centered Artificial Intelligence's (HAI) Center for Research on Foundation Models (CRFM)

Augmenting RL Opt. with Generative AI

- RL (Optimization)
  - Uses tool engines to create and evaluate design data on the fly
  - Signoff-accurate: Results are outcomes of existing tooling
  - Overall slower, relies on process-level distribution
  - Semi-automatic: Requires significant effort in describing design spaces, outcome metrics

Good at identifying **optimality**

- GenAI (Generation)
  - Captures data history from prior design journeys
  - Speculative: Results are outcomes of trained neural networks
  - Overall faster, relies on data-level parallelization
  - Highly autonomous: Capable of traversing the data abstraction stack quickly, and with limited guidance

Good at generating **optionality**

Remember: Limitations of RL-based Opt.

1. Creating design variants is a high-effort task
    - Verifying even a single version of a design is difficult, how to scale?

Optionality

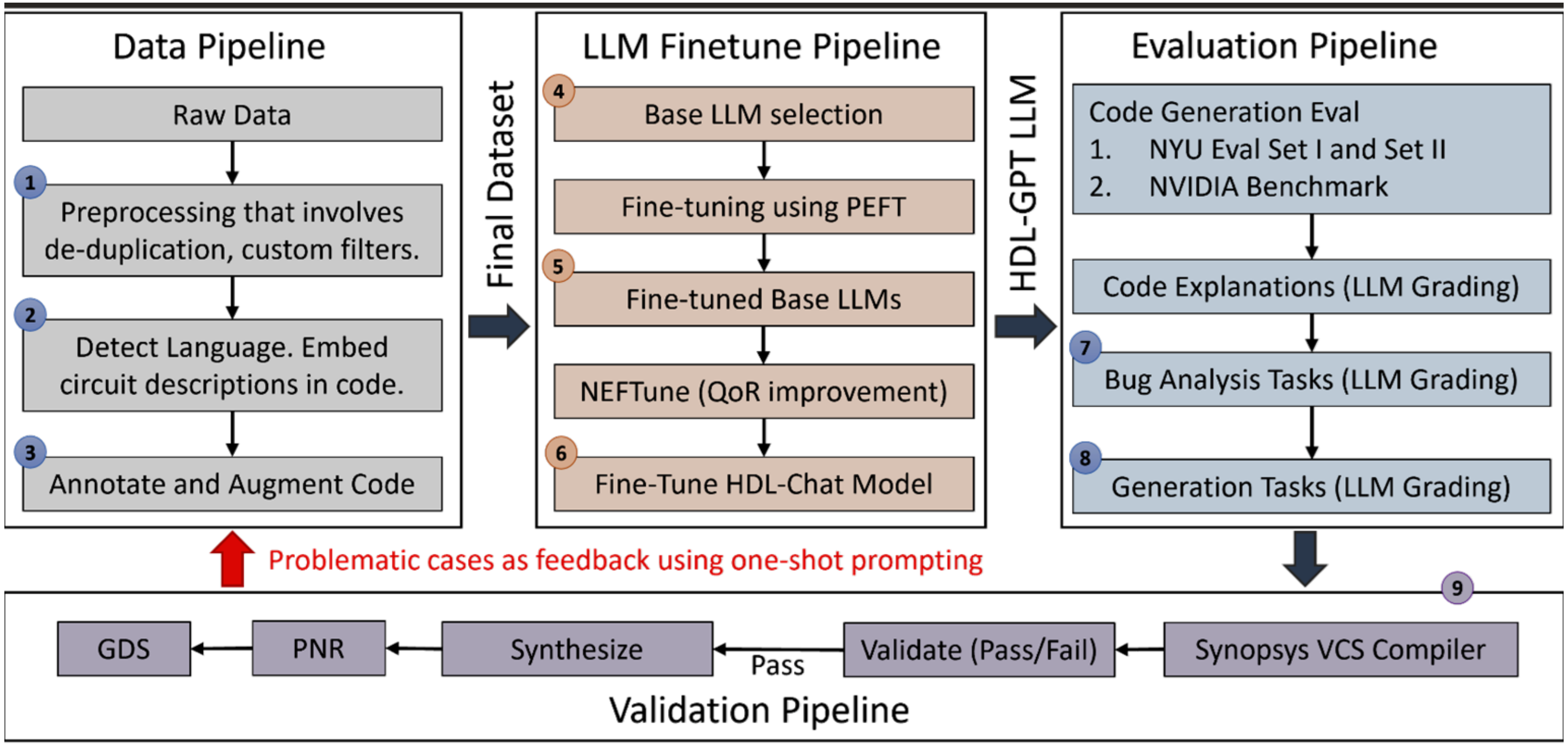
2. Evaluating design variants can be slow
    - Typically involves synthesis, P&R, timing/power/IR/etc. analysis

Optimality

Optionality

1. Research in HDL Generation

Workflow for Data, Fine-tuning, Evaluation, and Verification/Feedback Pipeline for HDL-GPT



Comparison vs NYU Eval Set II

Challenge	HDL-GPT
Getting Started	1
Basics	1
Vectors	0.89
Module Hierarchy	0.33
Procedures	0.88
More Features	0.63
Basic Gates	0.88
Multiplexers	1
Arithmetic Circuits	0.57
K-Map to Circuits	0.75
Latches & Flip-flops	0.94
Counters	0.5
Shift Registers	0.44
Cellular Automata	0.67
FSM	0.61
Larger Circuits	0.71
Find bugs	0.6
Average	**0.73**

Bhuvnesh Kumar, Saurav Nanda, Ganapathy Parthasarathy, Pawan Patil, Austin Tsai and Parivesh Choudhary, "HDL-GPT: High-Quality HDL is All You Need," 2024 Design Automation Conference (DAC)

arXiv:2407.18423v1 [cs.LG] 25 Jul 2024

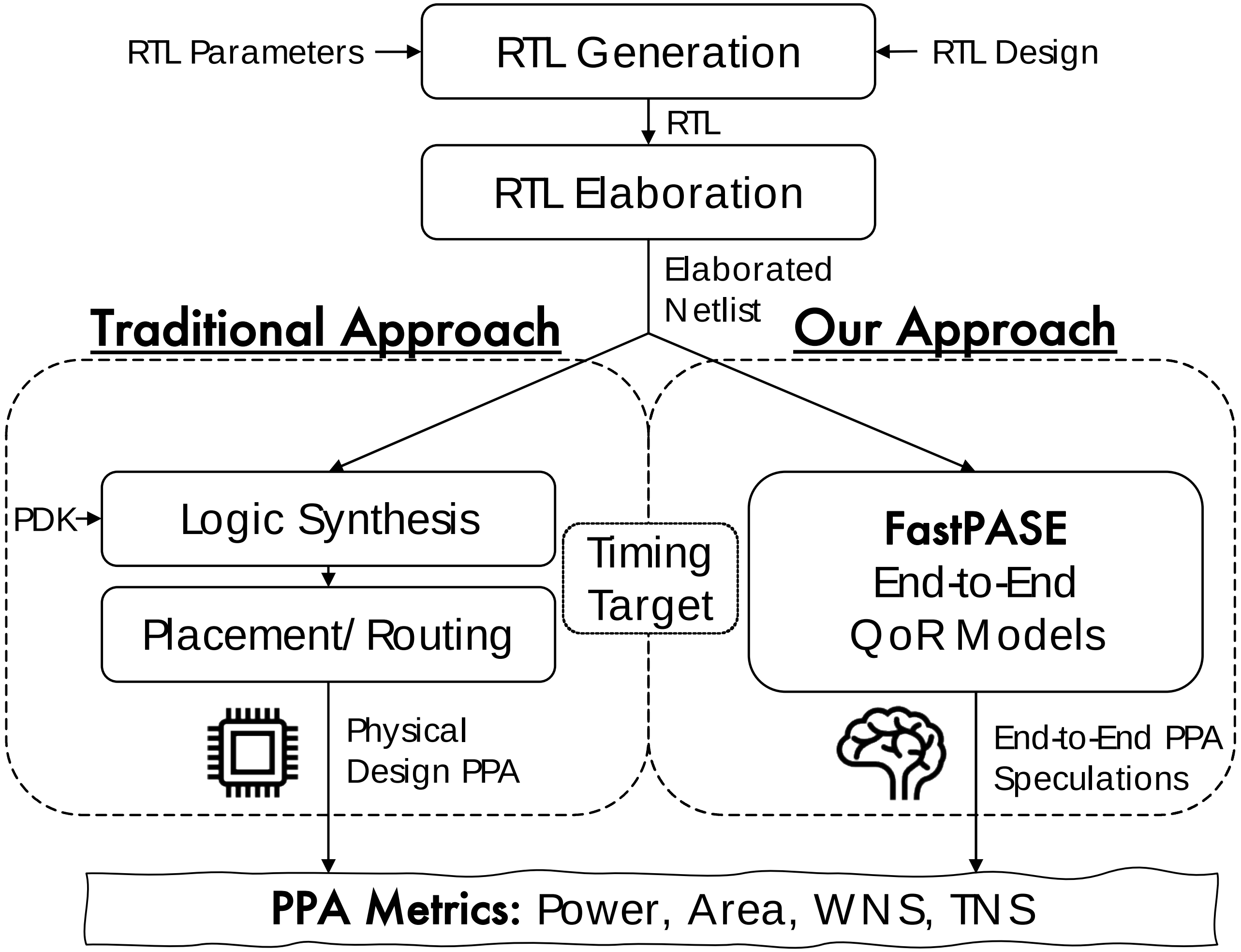
SYNOPSYS

HOT CHIPS

Optimality

2. Research in PPA Speculation

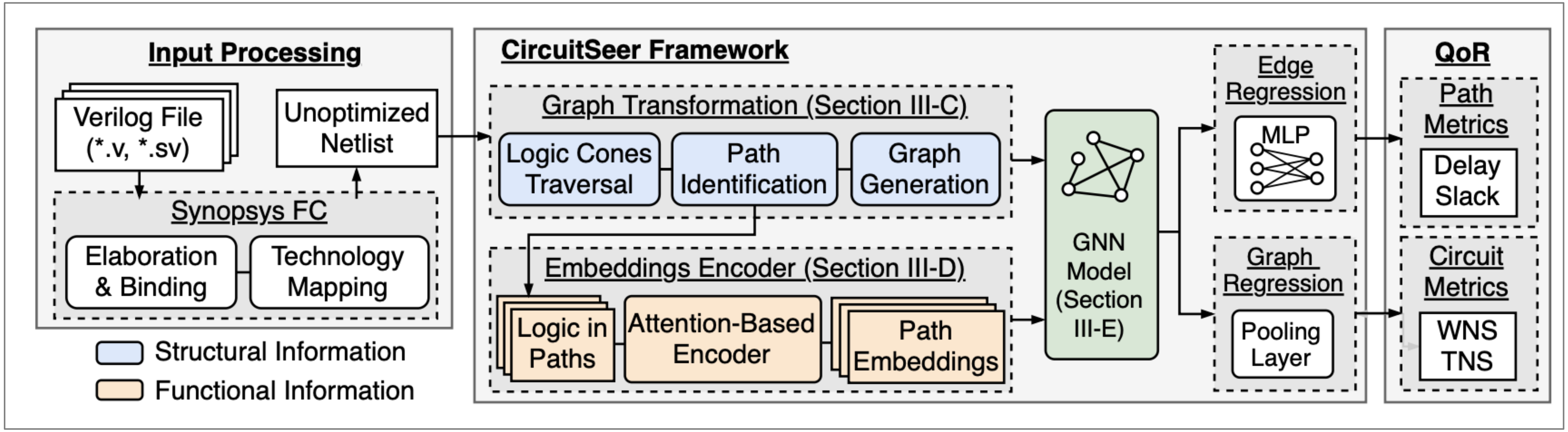
- Challenge: Evaluation of design options slow, compute intensive
- Approach: Use GCNs for end-to-end PPA speculation
- 10X faster evaluation, broader and deeper search

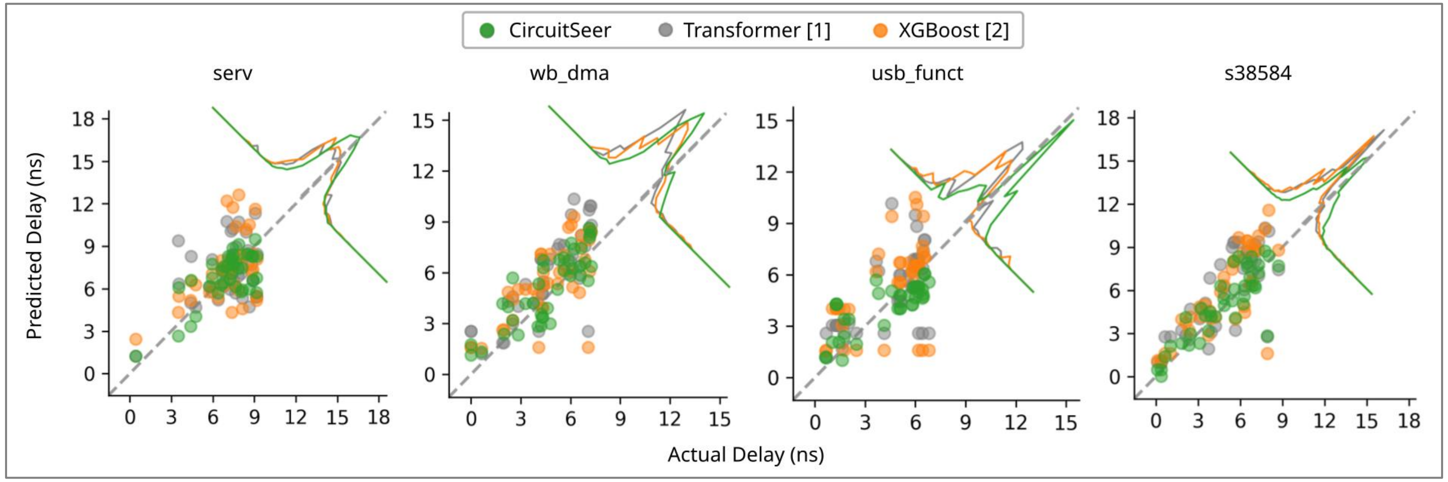


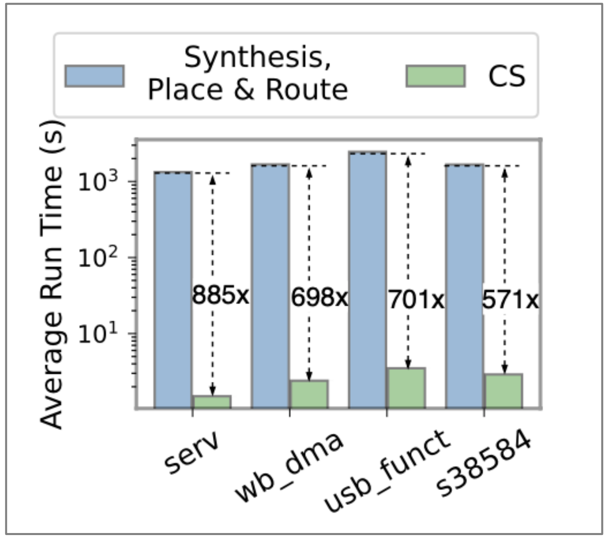
A. Levy, J. Walston, S. Samanta, P. Raina and S. Diamantidis, "FastPASE: An AI-Driven Fast PPA Speculation Engine for RTL Design Space Optimization," 2024 25th International Symposium on Quality Electronic Design (ISQED)

Using GCNs to Accelerate Design Evaluation

Optimality







>500X faster exploration TAT

S. Gandham, J. Walston, S. Samanta, and S. Diamantidis, "CircuitSeer: RTL Post-PnR Delay Prediction via Coupling Functional and Structural Representation" 2024 ICCAD (accepted)

Summary – AI-Assisted Design

- RL has enabled the 1st wave of AI in chip design (optimization)
  - Applications up and down the data abstraction stack
- GenAI is opening up opportunities to tackle the design process holistically
  - Traverse data abstractions more efficiency
- Next level challenges emerge
  - High-level planning driven by reuse, past experiences
  - Fast assessment of design quality for functional correctness and performance
- Technology advancing very rapidly, accelerating pace of AI-assisted design

Thank you!