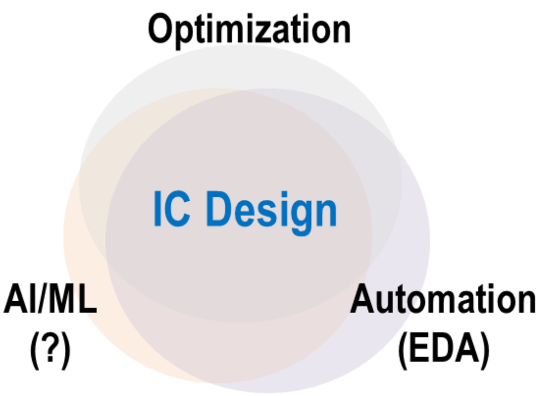
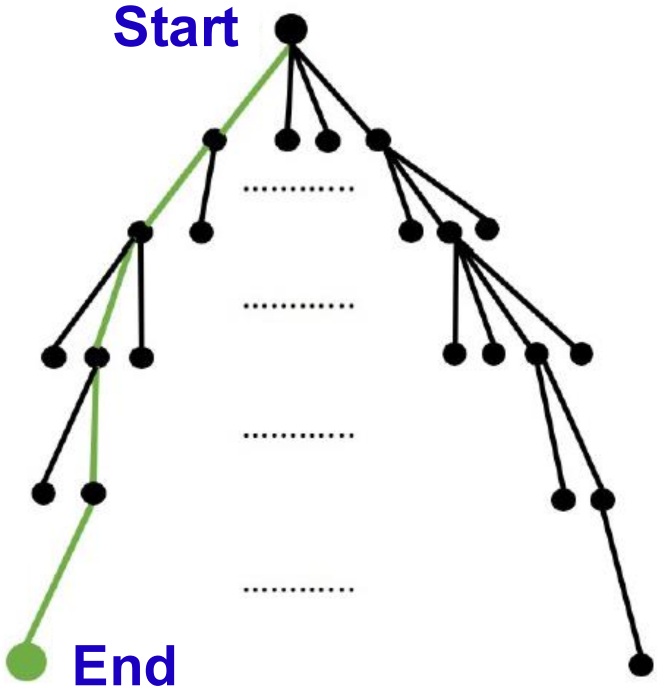
AI and ML with OpenROAD: Some Possibilities!

ECE 260C, Spring 2025
Andrew B. Kahng



Design Optimization Lives in a Box



- **Start** to **End**: expensive!
  - O(year) for product
  - O(weeks) for SP&R and Opt
- Goal: best possible **End**
- Constraint: stay in "Box"
  {compute}
  X {licenses}
  X {people}
  X {weeks}

Designers always need more leverage!

“Machine Learning in EDA”: Why

A. You need models to have predictions
B. You need predictions to leverage in exploration
C. What you can’t predict, you guardband
D. What you don’t explore, you leave on the table
E. C and D are bad for product quality and schedule

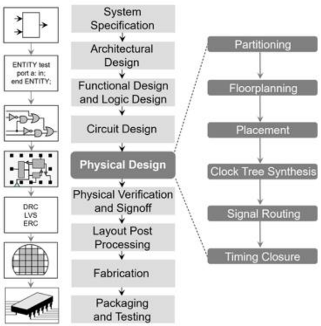
“Moore’s Law slowdown” → in an Era of Optimization

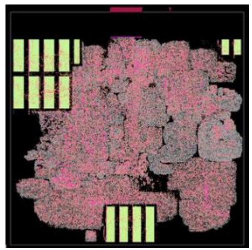
"Machine Learning in EDA": What

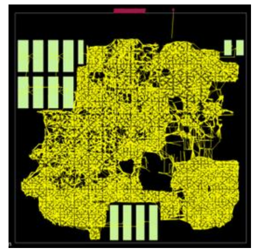
- **Predict**
  - Will RouteOpt finish with clean signoff, <1000 DRVs by tomorrow night?
- **Classify**
  - Out of these 50 floorplans + budgets, which 3 should go into trial SP&R?
- **Estimate**
  - How many hold buffers will tool eventually add into this post-CTS layout?
- **Guide / advise**
  - What P&R tool setup/script will obtain the best QOR within next 36 hours?

What’s Different (and Difficult) for AI/ML?

1. Changing **abstractions**, **formats**, and **the design itself**



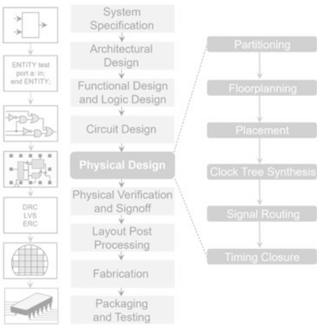


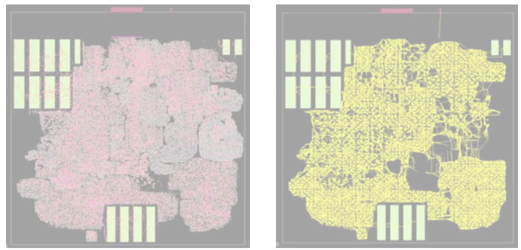


LEF, DEF, .LIB, SVRF, UPF, CPM, APL, SPEF, SDC, GDS, .V, VHDL, .SP, ITF, TLU+, NXTGRD, PFM, QRCtech, SDF, CCS, LVF…

What’s Different (and Difficult) for AI/ML?

1. Changing **abstractions**, **formats**, and **the design itself**



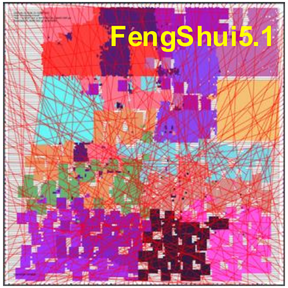


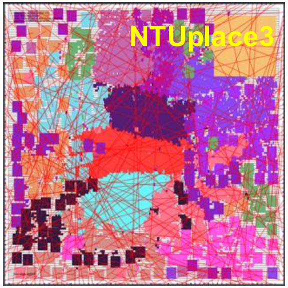
LEF, DEF, .LIB, SVRF, UPF, CPM, APL, SPEF, SDC, GDS, .V, VHDL, .SP, ITF, TLU+, NXTGRD, PFM, QRCtech, SDF, CCS, LVF…

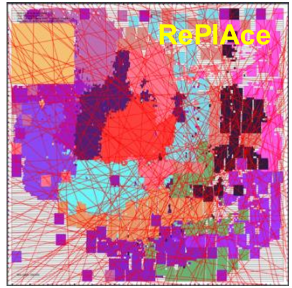
2. **Long chains** of distinct, intractable **discrete optimizations**

- “Practical optimization” = metaheuristics on top of metaheuristics
  - *Scale, multimodality, dynamism, diversity*
    → 1000s of hidden commands and options in a commercial placer !
- Objectives are **ad hoc**
- Trajectories are **chaotic**
- Outcomes have **distributions**

Outcomes <u>Heavily</u> Mediated by Heuristics!

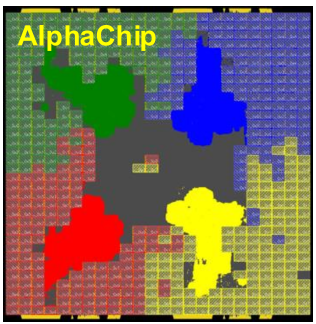


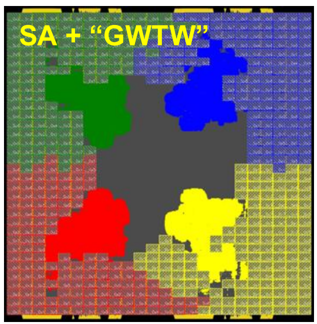


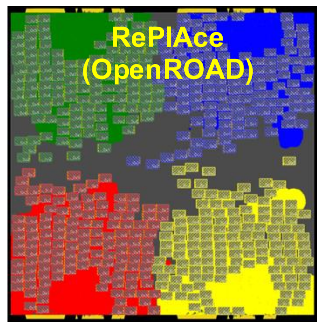


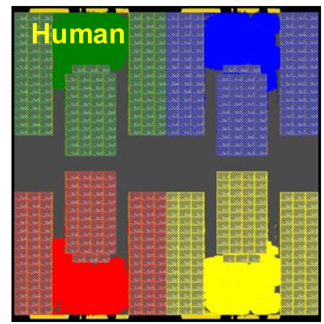
- “ibm01” from ISPD-2002, “Bookshelf” format
- 12.7K instances
- FengShui ~2005; NTUplace3 ~2008; RePlAce ~2018

- “CT-Ariane133 X4”, protobuf format [link]
- 532 macros, 332K cells in TSMC 7nm
- AlphaChip 2024, SA/GWTW 1983/1994; RePlAce 2020-









“ibm01” thanks: Prof. Patrick Madden, Binghamton Univ.

What's Different (and Difficult) for AI/ML?

1. Changing **abstractions**, **formats**, and **the design itself**

LEF, DEF, .LIB, SVRF, UPF, CPM, APL, SPEF, SDC, GDS, .V, VHDL, .SP, ITF, TLU+, NXTGRD, PFM, QRCtech, SDF, CCS, LVF…

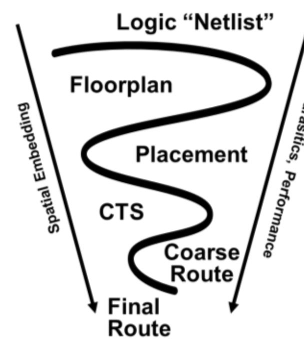
2. **Long chains** of distinct, intractable **discrete optimizations**

- "Practical optimization" = metaheuristics on top of metaheuristics
  - *Scale, multimodality, dynamism, diversity*
    → 1000s of hidden commands and options in a commercial placer !
- Objectives are **ad hoc**
- Trajectories are **chaotic**
- Outcomes have **distributions**

3. **Loops are expensive**

(often, fatally so)

- Design process must "converge" both spatial embedding **and** performance

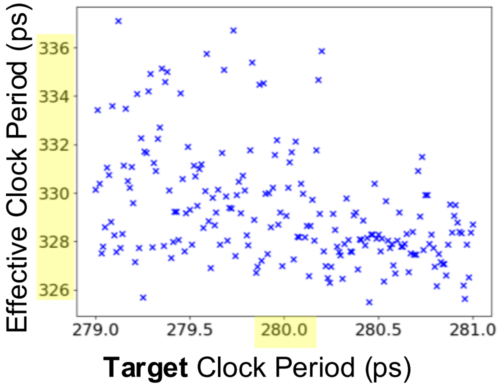


How?

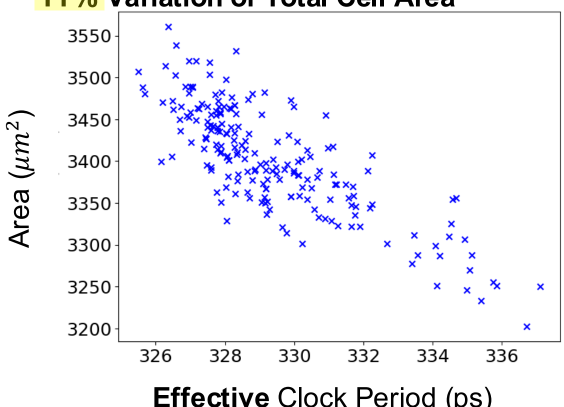
With Chained Chaotic Optimizations ?!?

Logic Synth

3.4% **Variation of Effective Clock Period**

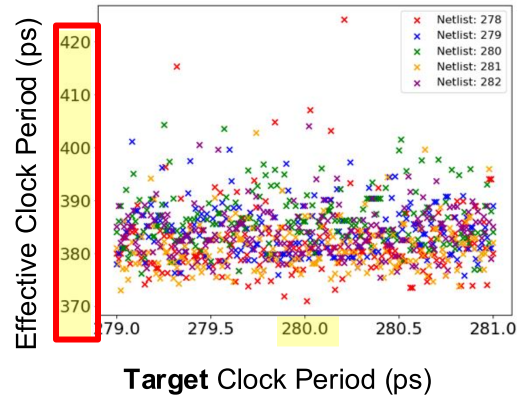


11% **Variation of Total Cell Area**

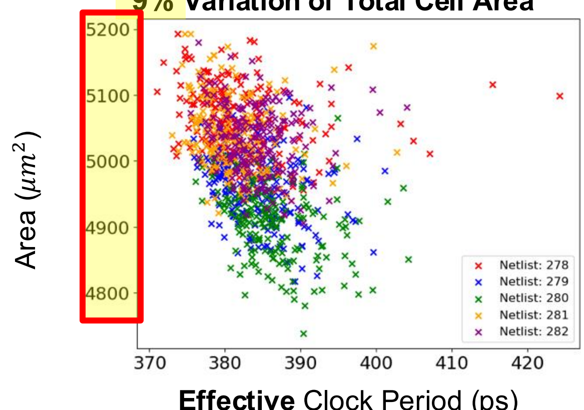


Place & Route

15% **Variation of Effective Clock Period**



9% **Variation of Total Cell Area**



Variation of metrics: $100 \times (\frac{max}{min} - 1)$

Implications for Optimization in IC Design

Catechism:

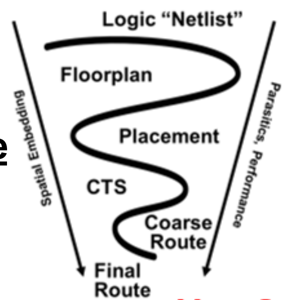
A. You need **models** to have predictions

B. You need predictions to **leverage** in exploration

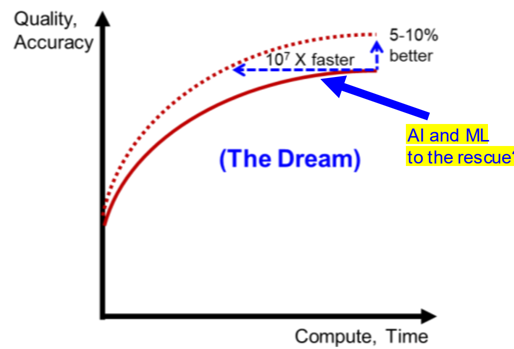
C. What you can't predict, you guardband

D. What you don't explore, you leave on the table

- Predictions today are **Constructive**
  - Quick-and-dirty, under the hood
- Optimizations today are **Iterative**
  - "Construct by Correction"

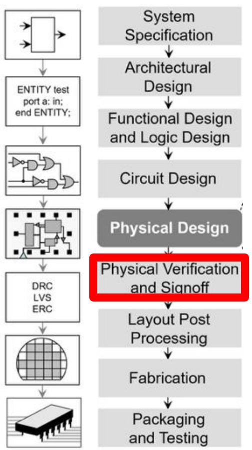


How?

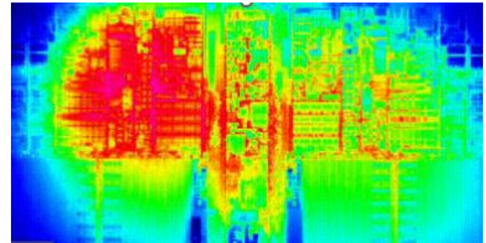


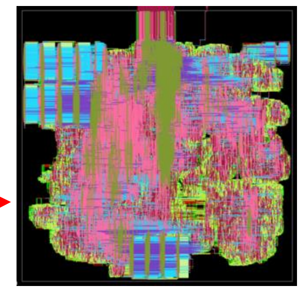
- **Need Fast Simulations** to guide Optimization
  - "Construct by Corrections …
    … that are Correct by Construction"

Signoff in Design: Optimize Max, not Sum!



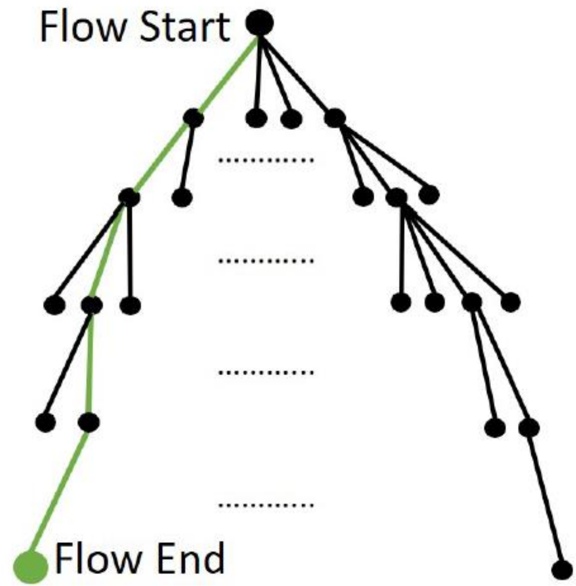
- **Signoff** is a defined business interface
  - Establishes “Who pays for the scrap?”
- Golden, foundry-qualified tools perform signoff analyses and simulations
  - Typically with very long batch runtimes
- Many design steps must optimize “max”
  - Max timing path delay determines max frequency
  - Max wiring congestion determines routing feasibility
- Inverse problems galore
  - Worst-case stimuli → e.g., “rogue wave in power grid”
  - “Whack-a-mole”, “ping-pong” are in the lexicon of design → can new AI foundation models help?





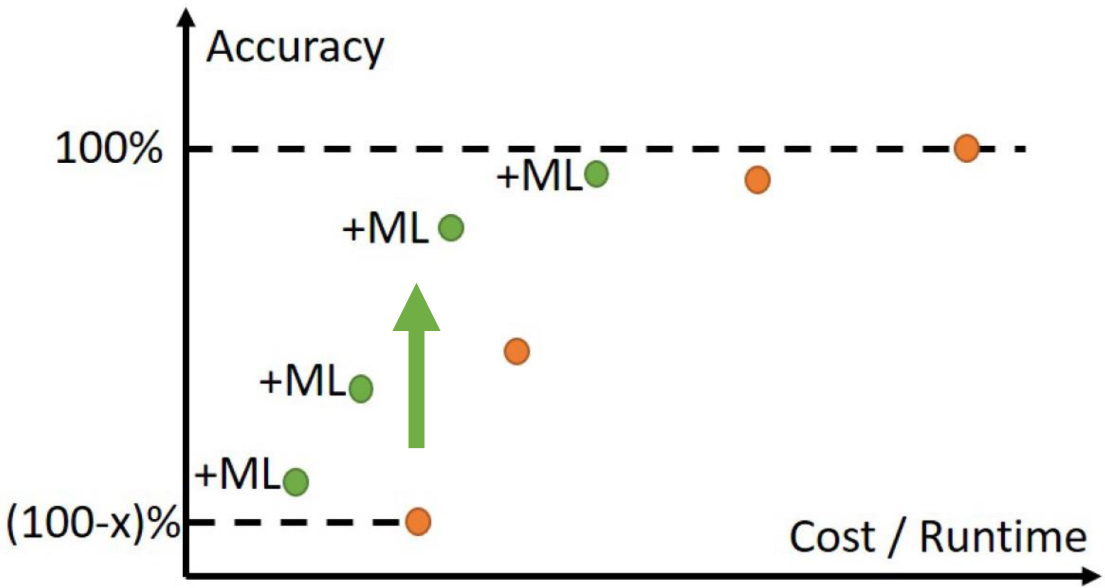
4-Stage "Roadmap" of ML in EDA

1. Mechanization and Automation
2. Orchestration of Search and Optimization
3. Pruning via Predictors and Models
4. From Reinforcement Learning through Intelligence



Huge space of tool, command, option trajectories through design flow

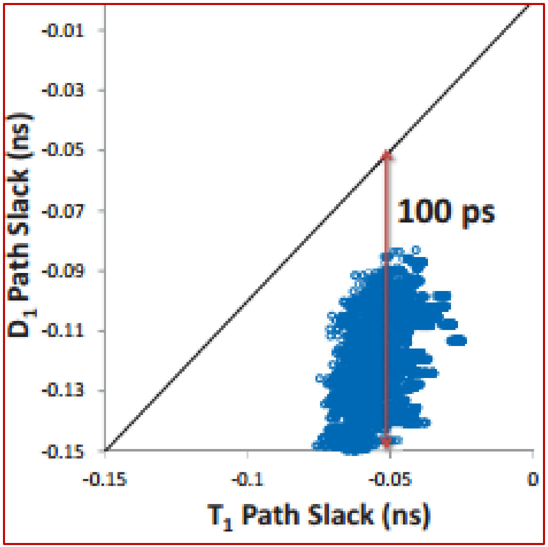
No-Brainer: Shift Accuracy-Cost Tradeoffs

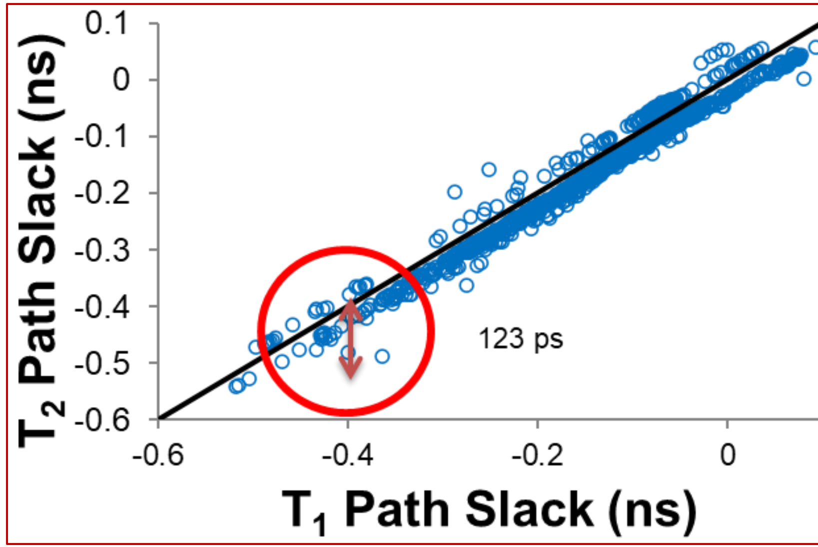


Tagline: "It's Just Physics!"

ML to Fix Timing Miscorrelation

UCSD, [DATE-2014]



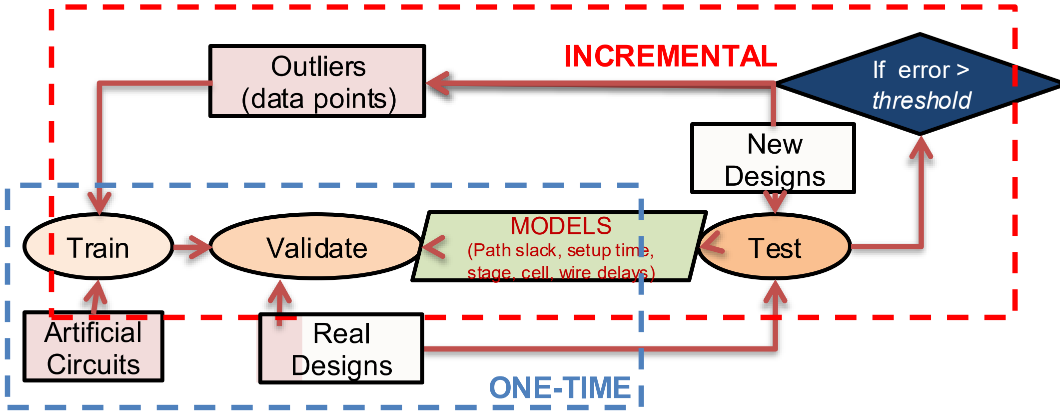


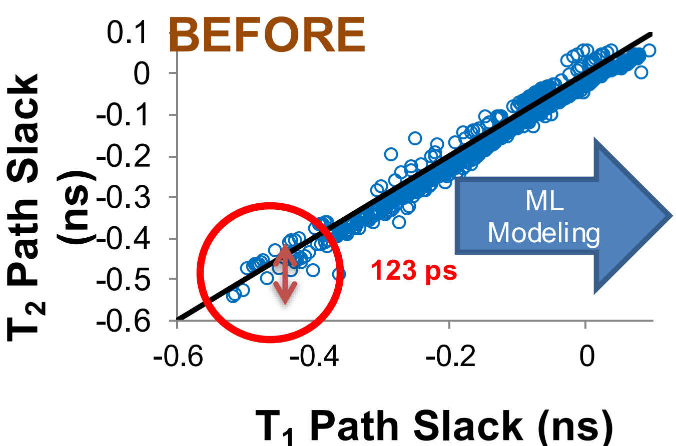
D = non-signoff timer (e.g., in P&R tool)
T = "golden" timer (e.g., signoff-qualified)

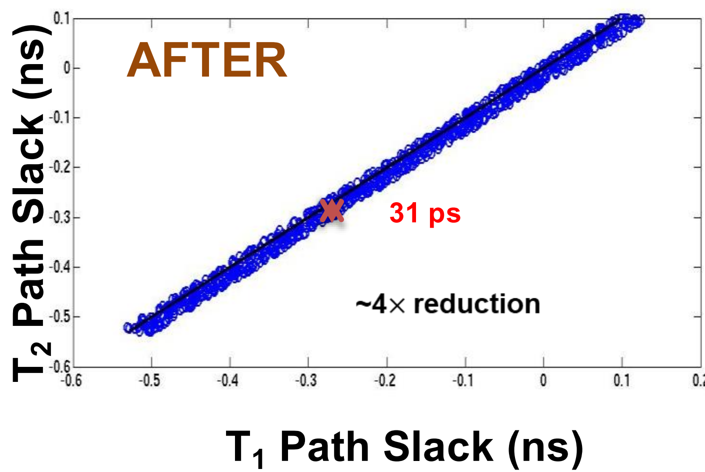
- **Can you explain the slack miscorrelation?**
- **What is the impact of the miscorrelation?**

“gt1-gt2”: ML to Erase Miscorrelation

Can also erase miscorrelation between D, T



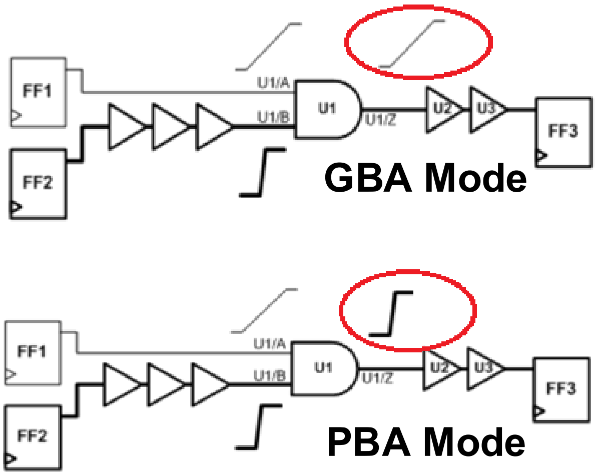


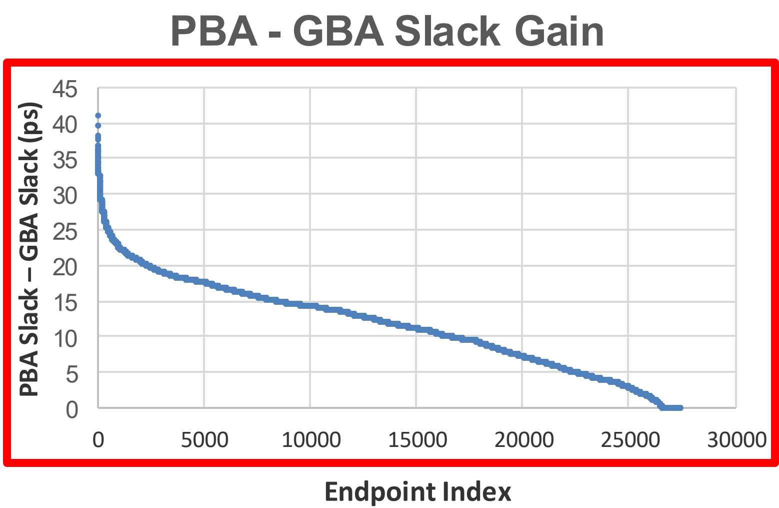


Predicting PBA from GBA

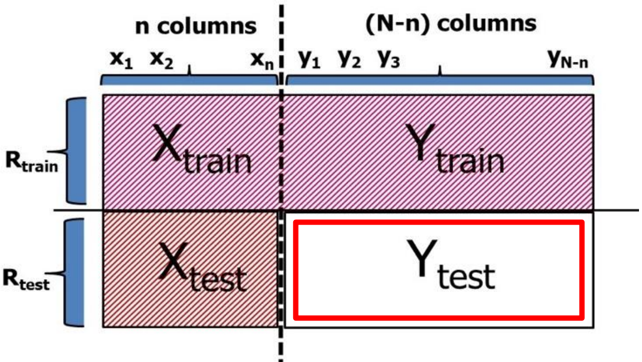
UCSD, ICCD18

- PBA (Path-Based Analysis) is less pessimistic but more expensive than GBA (Graph-Based Analysis)
- **ML to predict PBA timing from GBA timing**
  → Better and faster outcomes from P&R, Opt

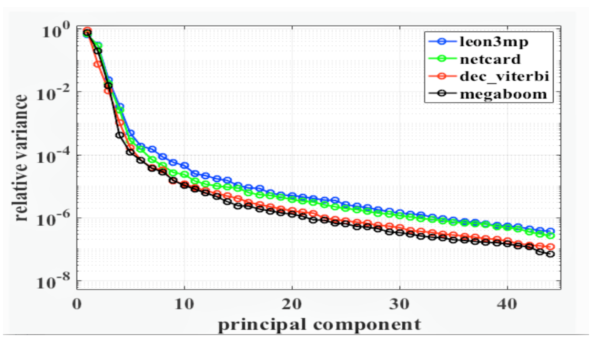




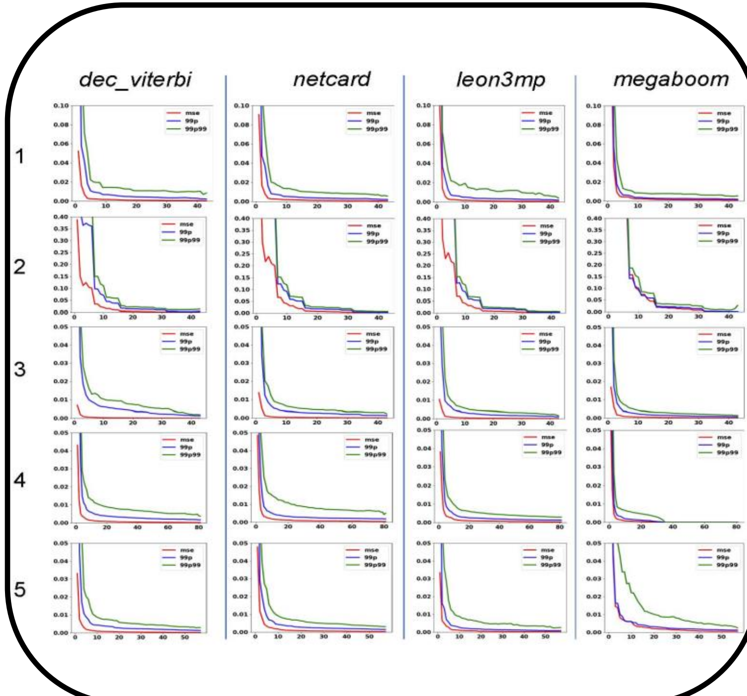
Timing at "Unobserved Corners" UCSD, DATE19



Predicting missing delay values
= *matrix completion* problem



PCA: low-dimensional modeling task

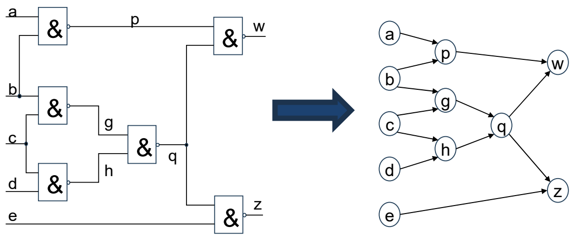


STA at few *known corners* → predict timing at all *unknown corners*

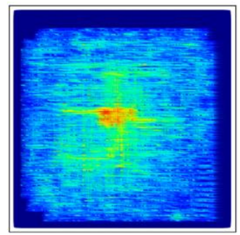
"It's Just Physics!"

A Call-Out: Semiconductor Design Data

- **Many types**
  - Formal specs
  - HDLs
  - Graphs
  - Hierarchies
  - Tabular data
  - Images




```
// Memory Write Block
// Write Operation : When we_0 = 1, cs_0 = 1
always @ (address_0 or cs_0 or we_0 or data_0
or address_1 or cs_1 or we_1 or data_1)
begin : MEM_WRITE
  if ( cs_0 && we_0 ) begin
     mem[address_0] <= data_0;
  end else if  (cs_1 && we_1) begin
     mem[address_1] <= data_1;
  end
end
```



- **IC data characteristics**
  - Constantly changing
    - technologies, designs, tools, …
  - Non-standard forms (even, “Tower of Babel”)
  - No massive redundancy
  - Different shapes and scales
  - No “Zipf’s Law”

- **Is unsupervised learning even possible?**
- **Who owns, and who can use, what data?**

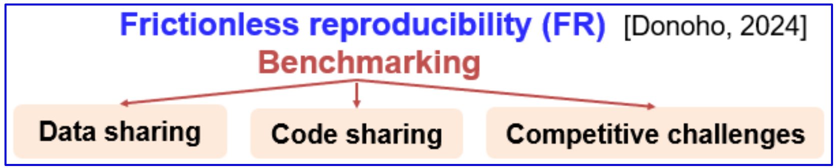
Thanks: Igor Markov, Synopsys

A Call-Out: Semiconductor Design **Data**

- **Proprietary**
  - Designs
  - Technologies
  - Design methods
  - EDA tools
- **Expensive**
  - E.g., design flows take weeks to run
- **Closer to physics → harder to access**
  - Materials
  - Equipment
  - Process, Devices
  - …

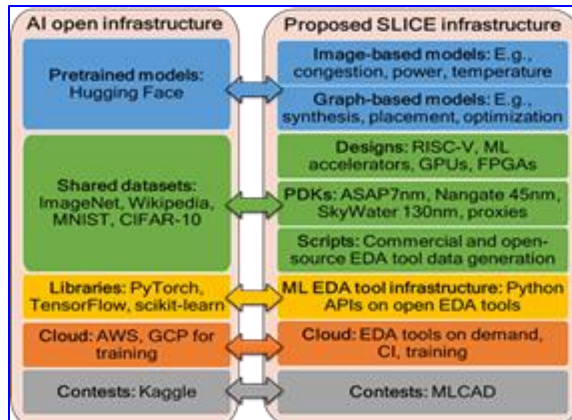
Semiconductor Design Data: Well-Lamented Gaps

- **Proprietary**
  - Designs
  - Technologies
  - Design methods
  - EDA tools



- **Expensive**
  - E.g., design flows take weeks to run

- **Closer to physics → harder to access**
  - Materials
  - Equipment
  - Process, Devices
  - …

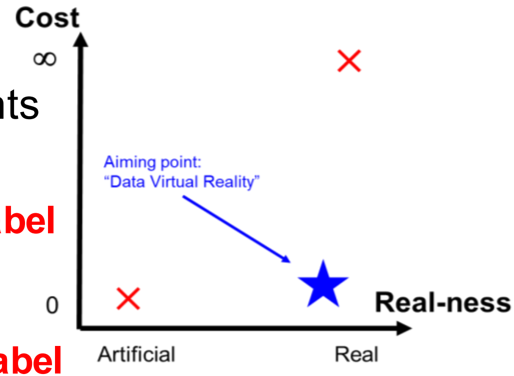


https://slice-ml-eda.github.io/

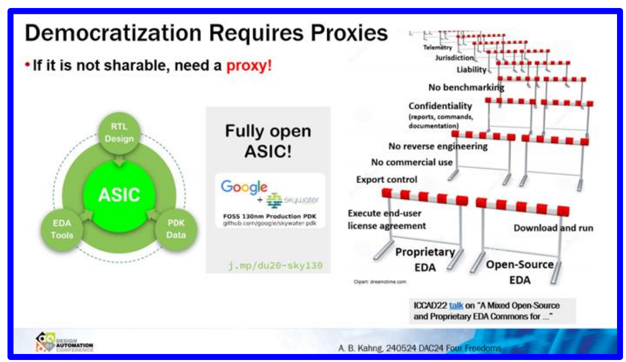
A Call to Action: Must Develop “Proxies”

“if it can’t be shared, need a proxy !”

- Proprietary PDK (Process Design Kit) data
- Commercial EDA End-User License Agreements
  - **No benchmarking (!)**
  - **Copyrighted command language → Tower of Babel**
    - report_timing, report_checks, report_timing_analysis, check_timing, …
  - **Copyrighted report formats → <u>more</u> Tower of Babel**
    - ^ and v vs. r and f , *** vs. === , …
  - Can’t share/upload ML data or models!



- **Foundation Models** will require **Data**, which will require **Proxies** !
  - Tech files, device models, “safe names”, design enablements / tool setups, sharable results and metrics, …
    (“journey of a thousand miles …”)

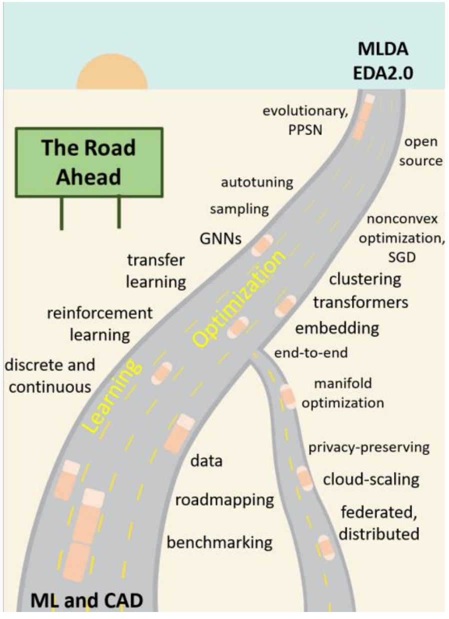


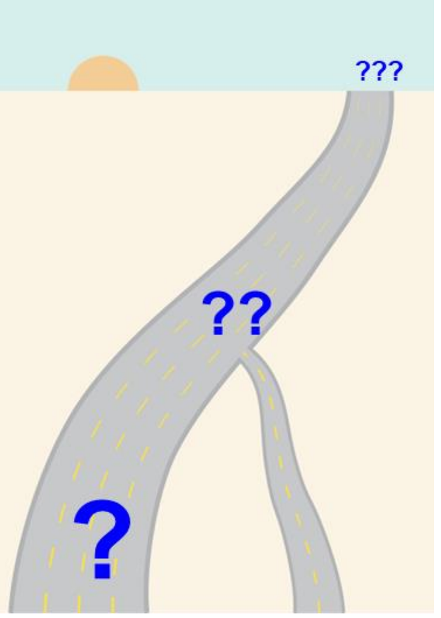
Er, EDA Foundation Models ?

- Physics
- Logic
- Circuits
- Multimodal

Towards Foundation Models for Scientific Machine Learning: Characterizing Scaling and Transfer Behavior

Shashank Subramanian, shashanksubramanian@lbl.gov, Lawrence Berkeley National Lab
Peter Harrington, pharrington@lbl.gov, Lawrence Berkeley National Lab
Kurt Keutzer, keutzer@eecs.berkeley.edu, UC Berkeley
Wahid Bhimji, wbhimji@lbl.gov, Lawrence Berkeley National Lab
Dmitriy Morozov, dmorozov@lbl.gov, Lawrence Berkeley National Lab
Michael W. Mahoney, mmahoney@stat.berkeley.edu, LBNL, ICSI, and UC Berkeley
Amir Gholami, amirgh@berkeley.edu, ICSI, UC Berkeley





"ML for CAD/EDA", 2020 talk, paper

"Foundation Models for CAD/EDA", 202?

Possibilities With OpenROAD

"Machine Learning in EDA": Why

A. You need models to have predictions
B. You need predictions to leverage in exploration
C. What you can't predict, you guardband
D. What you don't explore, you leave on the table
E. C and D are bad for product quality and schedule

"Moore's Law slowdown" → in an Era of Optimization

Kahng ECE 260C SP25 3

"Machine Learning in EDA": What

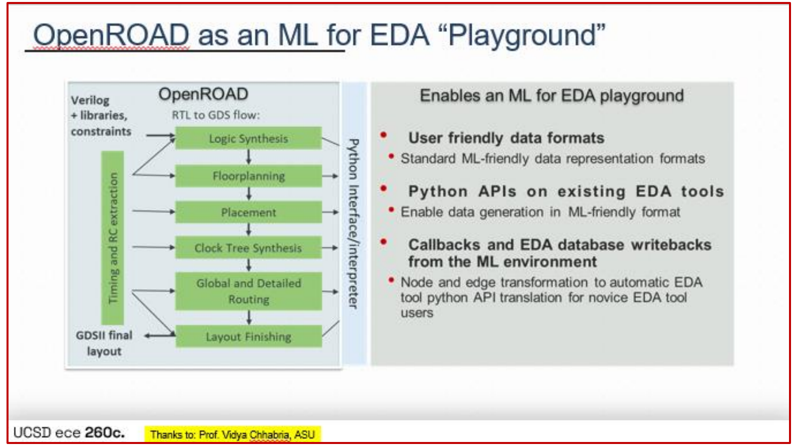
- **Predict**
  - Will RouteOpt finish with clean signoff, <1000 DRVs by tomorrow night?
- **Classify**
  - Out of these 50 floorplans + budgets, which 3 should go into trial SP&R?
- **Estimate**
  - How many hold buffers will tool eventually add into this post-CTS layout?
- **Guide / advise**
  - What P&R tool setup/script will obtain the best QOR within next 36 hours?

Kahng ECE 260C SP25 4

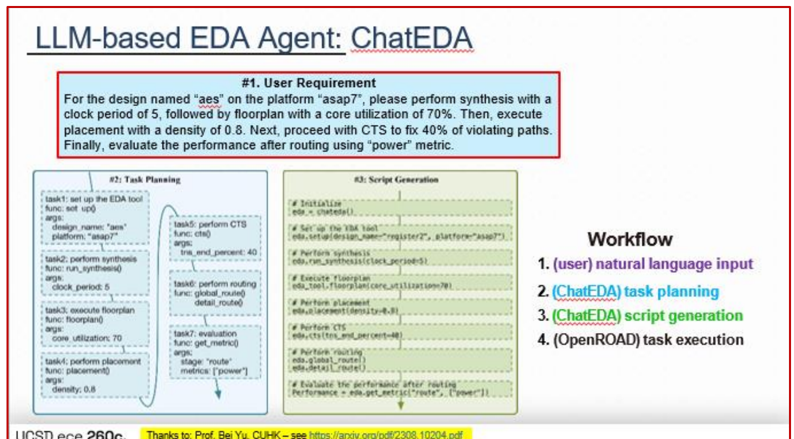
Have seen examples in lectures …

Examples: Lecture 1

- Lecture 1: CircuitOps (ASU, Nvidia)

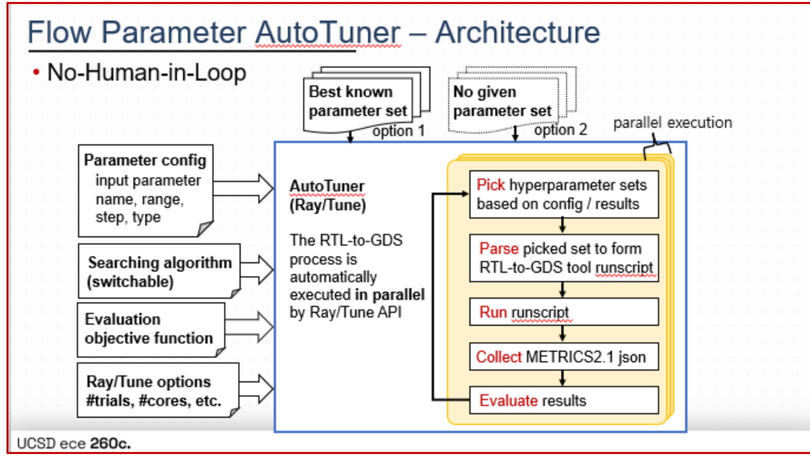


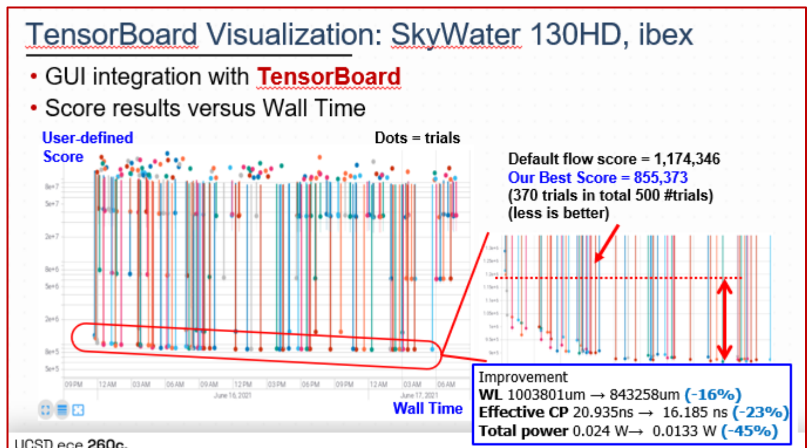
- Lecture 1: ChatEDA (CUHK)



Examples: Lecture 1

- Lecture 1: AutoTuner (UCSD)

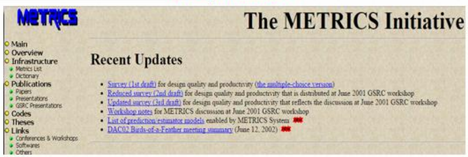




Examples: Lecture 1

- Lecture 1: METRICS (UCLA, UCSD)

"METRICS" (DAC00, ISQED01)



- **METRICS 1.0** (1999; DAC00, ISQED01)
  - **"Measure to Improve"** http://vlsicad.ucsd.edu/GSRC/metrics
- **METRICS 2.0** (WOSET-2018) was proposed as an update of **METRICS 1.0**
- **METRICS2.1** is proposed as a standard, with a concrete realization in OpenROAD

UCSD ece **260c.**

METRICS2.1: **Standard** Naming ! https://github.com/ieee-ceda-datc/datc-rdf-Metrics4ML

- **Problem**: "Tower of Babel" (names, formats that are all different and proprietary)
- **Solution**: "METRICS"
  - General and extensible
  - Syntax and semantics to support future addition of new metrics
- **No ambiguity!!!**
  - Any desired measurement must map to a unique METRICS2.1 metric
  - Every METRICS2.1 metric must map to a unique interpretation as a measurement
  - Two-way mapping is crucial to avoid future confusion
- Can also capture the same metric at different stages of the design flow
- **Free, open and frictionless – agnostic to EDA provider**

UCSD ece **260c.**

METRICS2.1 Examples https://github.com/ieee-ceda-datc/datc-rdf-Metrics4ML

Sample metrics

Metric	**Description**
timing__setup__wns	Setup worst negative slack in the design
timing__setup__wns__clock:clk_a	Setup worst negative slack for clock "clk_a" in the design
timing__setup__wns__analysis_view:slow	Setup worst negative slack for analysis view "slow"
power_total	Total power consumption
power__leakage	Total leakage power
power__leakage__clock	Total leakage power in the clock network

Many applications: data for machine learning, CI/CD infrastructure for software quality, ...

UCSD ece **260c.**

Examples: Lecture 2

- Lecture 2: The PDN Chicken-Egg (Arm, UMN, UCSD)

Template-based PDN Synthesis in Floorplan and Placement Using Classifier and CNN Techniques

Vidya A. Chhabria[1], Andrew B. Kahng[2], Minsoo Kim[2], Uday Mallappa[2], Sachin S. Sapatnekar[1], and Bangqi Xu[2]
[1]University of Minnesota; [2]University of California, San Diego

- Critical cells want to be near each other. Supplying extra power to this hotspot will force the cells apart.

pdngen Improvements / Problem Statements

- Automatic connectivity rules (add_pdn_connect)
  - Create rules automatically for defining grid connectivity based on layers used, instances present, etc.
- Edge connectivity
  - Enable the power grid to connect to edge ports for macros, right now this is only possible on pad cells and standard cells, for macros we only connect from the top which requires macros use lower metals for power and routing. This is not a huge problem in nodes with a large number of metal layers, but for sky130, ihp130, etc, this imposes a large penalty.
- Power grid reinforcement (eco power grid)
  - Later in most flows, after detailed placement, it may be possible to determine if additional wires are needed to ensure IR drop stays within limits based on the power requirements of those areas. On the opposite side, it maybe possible to prune the grid if IR drop is not an issue and free up routing resources. [cf. https://arxiv.org/abs/2110.14184]
- Automatic power grid definition (hard)
  - Given an IR drop goal, create a power grid that meets this requirement based on the parasitic resistances and estimated power (either from placement or good guesses)

Kahng ECE 260C SP25 49

- **Can ML predict the "convergence point"?**

(What can/should the flow do with such a prediction?)

Other examples: useful skew vs. synthesis; multi-bit FF clustering vs. place/opt; …

Examples: Lecture 13

- Lecture 13: BlobPlace (UCSD, POSTECH)

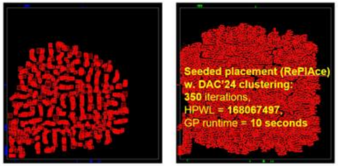
“Oracle” clusters – do they exist?

- “**Oracle**” clustering: hypothetical “**optimal**” clustering → generates “best” BlobPlace sol.
- *“Is there a clustering of a netlist that can lead the “best” final placement ?”* – Bodhi
- Spatial proximity ?
- Timing paths ?
- Avoid hot spots ?
- Or …?
- **Amenable to ML if these clusters do exist!**

Kahng ECE 260C SP25 41

Effect of seeded placement

- Design: Ariane (NG45)
- **Flat placement (RePlAce):** 440 iterations, HPWL = 171438951, GP runtime = 21 seconds



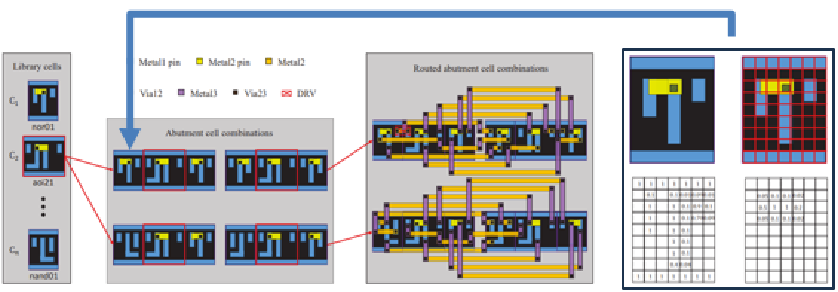
Kahng ECE 260C SP25 38

Examples: Lectures 7, 15

- Lectures 7, 15: Pin Access (UCSD)

ML for Pin Access Analysis ?

- **Pin accessibility prediction (PAP) : pattern recognition**
  - Predict DRV occurrences due to bad pin accessibility [YuFCHT20ISPD]
  - **"Close the loop"** : use PAP for placement refinement
- **Pin access generation : pattern generation**
  - Can we learn the pin access pattern from well-established tools ?
  - Try Generative Adversarial Network (GAN) or diffusion models ?
  - **"Close the loop":** replace the default pin access engine in OpenROAD



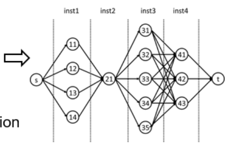
Pin accessibility evaluation for C_2 considering abutting cells

Feature extraction (pixel representation)

15

Cluster-Based Access Pattern Selection

- Instance ordering
  - Sort instances in the cluster according to x coordinate of the lower-left corner
- Graph construction
  - Vertex = access pattern
  - Shortest path from *s* to *t* is the best pattern combination



- DP formulation
  - Similar as previous formulation
  - No iteration

Kahng ECE 260C SP25 Thanks: Dr. Bangqi Xu 56

Examples: Lecture 12

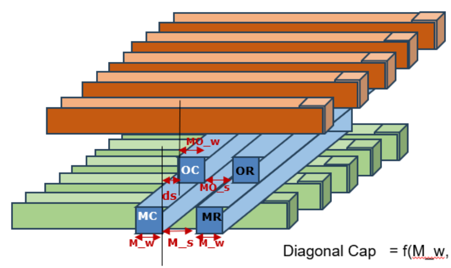
- Lecture 12: RCX Model Creation (Athena/Nefelus)

(2.5D) Extraction Model Creation

- Determine the characteristics of the process from the foundry technology information (e.g. itf or ict file)
  - Layer specs, wire shape, dielectrics stack, WEE tables, Thickness tables, Rho tables, …
- Create a large number of patterns of interconnect and transistor-level test structures
  - Interconnect-level test structures differ from transistor-level structures
- Run each pattern through a golden reference extractor
  - Field solver requires proper shape, enlargements, thickness and dielectrics
- Based on golden reference results, build extraction models
- Verify model results versus test results and design layouts

Kahng ECE 260C SP25 Thanks: Dimitris Fotakis and David Overhauser 27

OpenRCX Pattern E



Measure
CC_MC_OC
CC_MC_gnd

Diagonal Cap = f(M_w, M_s, ds, M, Mover)

Kahng ECE 260C SP25 Thanks: Dimitris Fotakis and David Overhauser 51

ML Target #1: "Virtual Buffering"

- Motivation: Netlist changes during ERC fix, timing fix (+ MBFF clustering, CTS, scan stitch, hold fix, antenna fix, etc.) how many netlists are there during P&R ?!?
  - Topology changes require accesses to GPU memory → slowdown of GPU-accelerated placers
- Goal: Predict future netlist changes, account for them during placement to avoid "buffer surprise"
  - E.g., pre-allocate space that will be used by buffers, upsizes

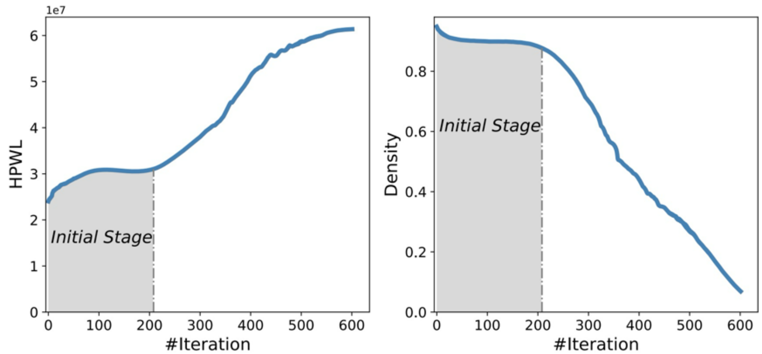
Theme: Prediction for Prevention ("doomed runs" etc.)

Comment 1: What is the ACCURACY requirement? E.g., costs of FP, FN

Comment 2: "Acting on a prediction changes what is being predicted, so BE CAREFUL WHAT YOU ASK FOR (from ML)" *("BCWYAF")*

ML Target #2: Placement Initialization

- Analytical placement = nonlinear optimization, sensitive to initial solutions
- Goal: reduce #iterations, improve placement QoR



Theme: ML for Warm Start e.g., as in BlobPlace's seeded blob placement

Comment 1: What are challenges?
representation, embedding, model architecture, training, scalability, generalization (transfer, fine-tuning, …) across designs and technologies, data

ML Target #3: Handoffs at Interstices

- **Co-evolutions, Co-optimizations are often at arm's length**
- **Interstices = opportunities for "*Conditioning Magic*" via ML**

Co-optimizations	**"Magic"**
• Netlist – Backend	• Netlist
• Hierarchy – Floorplan	• Netlist Partitioning
• Floorplan – SP&R	• Block shaping *+ boundaries*
• Synthesis – P&R	• Placement screens
• Place – Route	• Route screens
• GRoute – DRoute	• Route guides
	• Corners *+ endpoint SDCs*
	• Constraints
	• Tool/engine recipes
	• ...

Netlist Tomograph

Placement Tomography

Themes: Initialization, TAT, Sampling
from Chaos to Autotuning
+ Modern Compute (cloud, multicore)

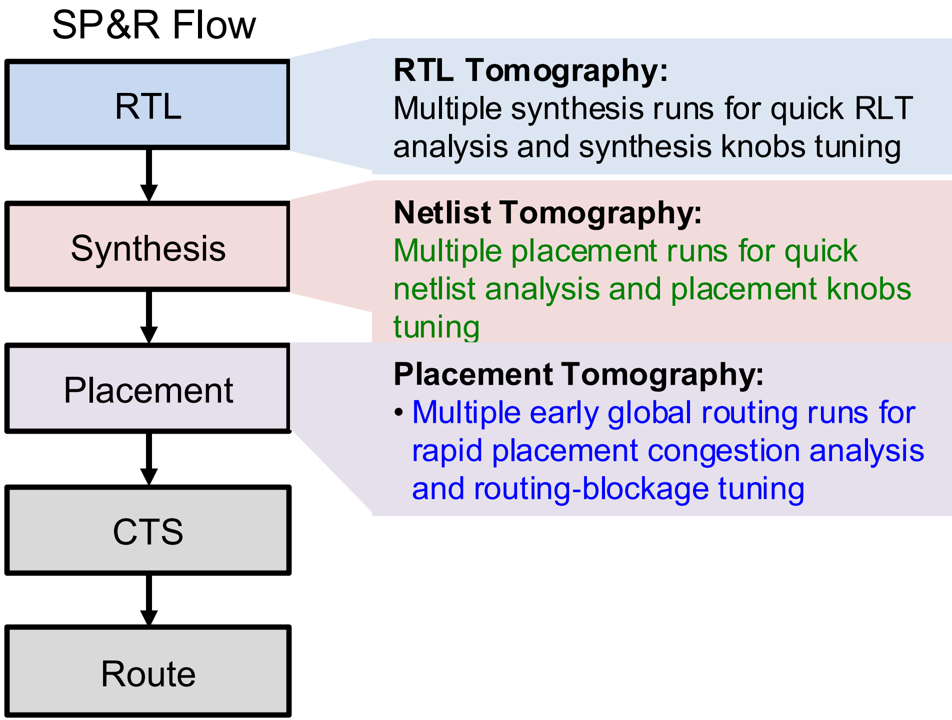
Comment 1: Key concept = "Tomography"

ISPD-2024 paper, slides

Concept of "Tomography"

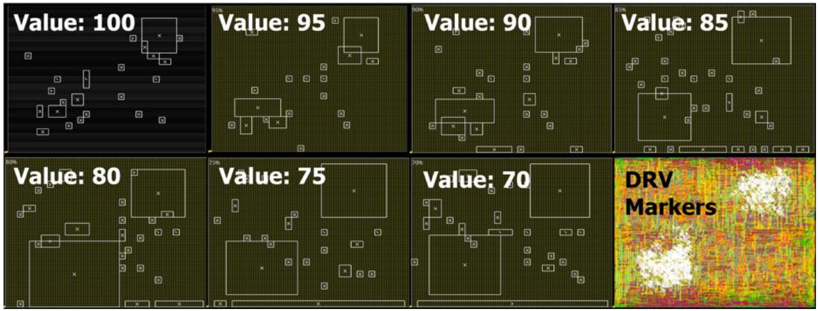
Tomography = multiple cross-sectional **views** of a solid object

- **IDEA:** Use many quick tool (GPU, proxy) runs and parallel execution to extract multiple views/reports **in one unit of time**



Placement Tomography (ICCAD24) [Link]

- **Placement Tomography:** Use routing blockage with varying routing-resource values → generate multiple congestion views



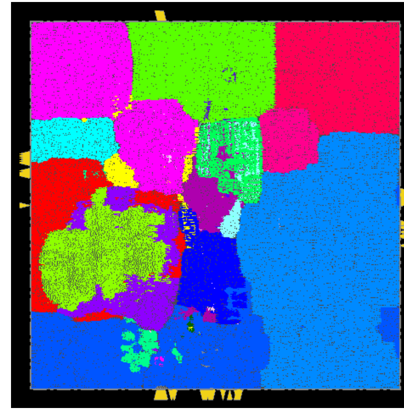
- ICCAD24 method has three main elements
  - **DRVNet** model: predicts layer-wise DRV hotspots
  - **BlkgComp** model: compares two routing blockage configurations
  - **RL Agent**: uses DRVNet, BlkgComp to sample routing blockages for #DRVs mitigation

Tech	Design	Method	#DRV	WL (mm)	WNS (ns)	TNS (ns)	Power (mW)
GF12	NOVA	No Blkg	1,242	1.000	-0.562	-782	1.000
		Human	403	0.994	-0.655	-707	0.994
		Ours	442	0.990	-0.475	-186	0.994
	LDPC	No Blkg	1,739	1.000	-0.583	-632	1.000
		Human	345	0.990	-0.472	-604	0.980
		Ours	346	0.990	-0.438	-717	0.980
	CA53	No Blkg	3,316	1.000	-0.369	-1,495	1.000
		Human	942	0.992	-0.615	-783	0.986
		Ours	902	0.991	-0.454	-696	0.986

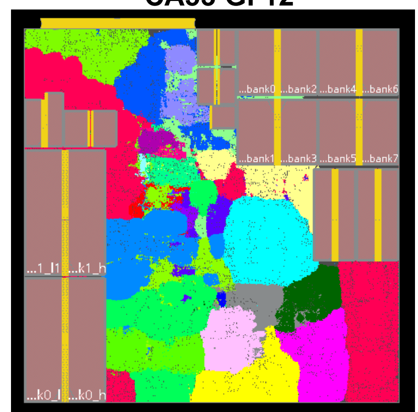
Can Also Try Netlist or RTL Tomography!

- Multiple views of a netlist
  - Slightly perturb the floorplan or SDC
  - Run each through place or placeopt
- Analyze placement runs
  - Instances in congested regions
  - Failing endpoints
- Set available placement knobs
  - **Placement blockage**
  - Cell padding
  - **Soft guides**
  - …
- *How about RTL tomography?*

NOVA GF12

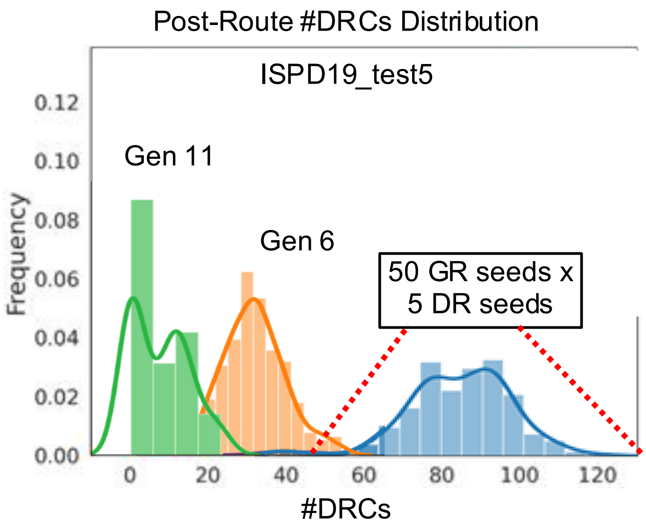


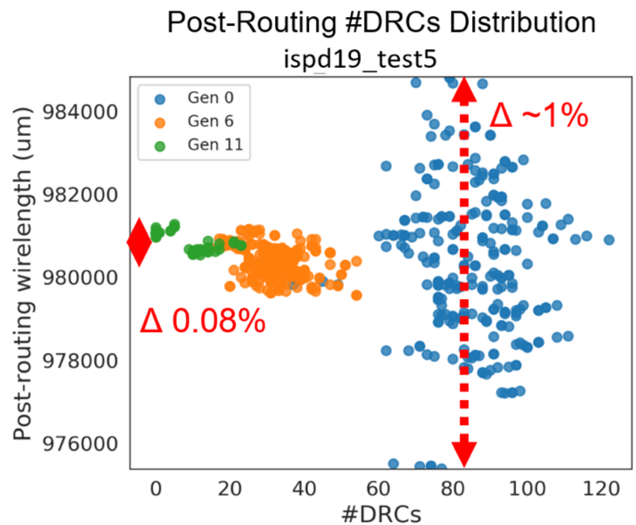
CA53 GF12



ML Target #4: Evolution in Optimizers

- Example/Observation: global routing solution (route guides) sets the detailed router's initial solution space
- Can we create "good" GR solutions from "not-good" solutions?
- **UCSD "genetic / adaptive multi-start metaheuristic"** using patching-based hybridization





AMS, CAMS

Theme: "Parallel Problem Solving from Nature"
Learn-to-Optimize, Adaptive Learning, Multi-fidelity BO, Evolutionary Optimization, Reinforcement Learning, … (+ metaheuristics)

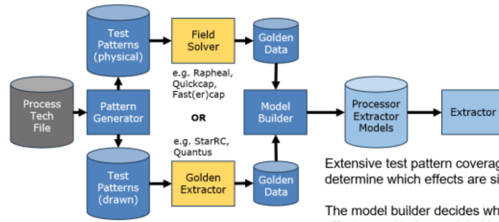
ML Target #5: "The Analog Hole"

(2.5D) Extraction Model Creation

- Determine the characteristics of the process from the foundry technology information (e.g. itf or ict file)
  - Layer specs, wire shape, dielectrics stack, WEE tables, Thickness tables, Rho tables, …
- Create a large number of patterns of interconnect and transistor-level test structures
  - Interconnect-level test structures differ from transistor-level structures
- Run each pattern through a golden reference extractor
  - Field solver requires proper shape, enlargements, thickness and dielectrics
- Based on golden reference results, build extraction models
- Verify model results versus test results and design layouts

Kahng ECE 260C SP25 Thanks: Dimitris Fotakis and David Overhauser 27

Golden Reference Tool to Build Models



Extensive test pattern coverage helps determine which effects are significant

The model builder decides which effects need to be modeled in the extractor

Note: OpenRCX does NOT include explicit Metal Fill in the modeling patterns

Kahng ECE 260C SP25 Thanks: Dimitris Fotakis and David Overhauser 28

- **Field solvers, golden tools, patterns, … Can we skip this?**
- **Idea: build OpenRCX table from massive post-P&R def, spef**

Theme: "It's Just Physics" could mean "No More Secrets" !!!

Comment 1: There is massive data in standard tool reports, outputs
Comment 2: ML can help "bypass" PDK data for material, device, BEOL …

AI, ML (+ OpenROAD): Some Takeaways

- What's Different (and Difficult)
  - **Layers** of hierarchy, abstractions and boundaries in any given instance of "design problem"
  - "Construct by correction" in chained chaotic, discrete (+ high-stakes) **optimization** *(how to use predictions?)*
  - **Extreme** multiscale in space, time
  - **Infinite variety** of possible layouts, shapes, constraints, designs
  - **Data is unavailable** → **proxies** needed *(+ privacy-preservation, obfuscation, trust/verification etc.)*
- Many Opportunities
  - **Full-stack proxy data** generation will unblock **many** next steps ! *("#1 with a bullet")*
  - "Stubs and supports" for physical modeling and design of digital twins, 3D heterogeneous integration
  - Fast and accurate optimization of multiphysics behavior without detailed PDE simulations
  - Leverage datasets, insights, models developed for other engineering and design domains?
  - **Pathfinding** (from materials to systems) into the "beyond-everything" future
  - **[Opportunities in RTL-to-GDSII: see Lecture Slides, Example ML Targets, and Themes!]**
- Trajectories of Connecting Silos
  - Materials + Mechanical sciences + **EE/CS + Optimization + AI**
  - **Electrical-Thermal → Mechanical →** Aging/Reliability/Noise → …
  - **Digital (HW + SW) → Analog →** Mechanical → (hydraulic, thermal, magnetic, fluidic) → …