Use Cases and Deployment of ML in IC Physical Design

Amur Ghose, aghose@ucsd.edu

Andrew B. Kahng, abk@ucsd.edu

Sayak Kundu, sakundu@ucsd.edu

Yiting Liu, yil375@ucsd.edu

Bodhisatta Pramanik, bopramanik@ucsd.edu

Zhiang Wang, zhw033@ucsd.edu

Dooseok Yoon, d3yoon@ucsd.edu

Motivation

- AI/ML techniques have been applied to many IC physical design challenges, e.g.:
  - Hyperparameter autotuning for better PPA tool settings
  - ML Predictions of routing hotspots, doomed runs, and PPA
  - Routing blockage creation to improve routability and PPA
- But: practical challenges are seen in ML **deployment**

  ➔ ***Why have so many efforts fallen short?***
- This talk:
  - Issues surrounding data for ML
  - High-level principles for deployment
  - Basic "checklists" for data, models, and use cases
  - Context for MLOps and LLM-based application development

Agenda

- Data
  - Data Outside vs. Inside IC Design
  - Challenges and Ongoing Efforts (Academia and Industry)
- ML Deployment
  - Key Performance Indicators (KPIs) and Checklists
  - Machine Learning Operations (MLOps) and Commoditization
- Challenges for LLM Deployment
  - LLM X EDA: Software Engineering Issues
  - Challenges from EDA Flows

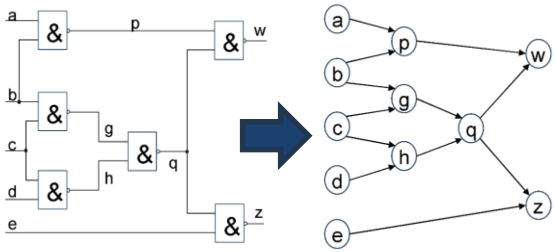
Data in PD: Scope, Modalities, Challenges

- Data is a core concern in ML for IC design

Example Challenges

Generalization across modalities

- Diverse IC data types
  - Formal specs
  - HDL
  - Graphs
  - Hierarchies
  - Tabular data
  - Images
- Hard for GenAI to interpret



Scarce and proprietary data

- IC design data is **costly** to produce
  - Huge scale, as well !
- High-quality public data is **scarce**
- Unshareable due to **proprietary** rights
  - PDK data
  - Commercial libraries
  - Soft IP data
  - EDA vendor data

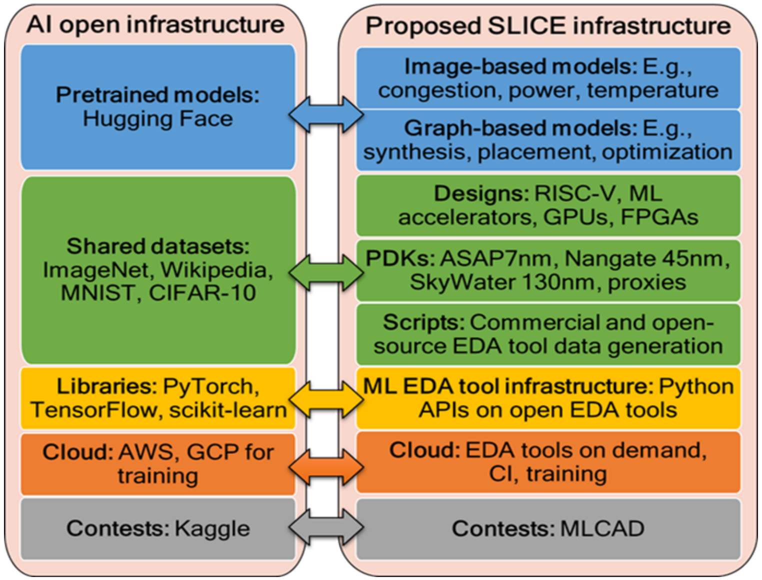
Data quality

- Larger datasets do not guarantee better ML models
- Common problems:
  - Outdated, stale data
  - Incomplete coverage
  - Risks such as data poisoning

Academic Efforts

Many initiatives, contributions to mitigate data scarcity

- Artificial netlist generators (ANG+), proxy PDKs (ASAP7+)
- Open-source toolchains (OpenROAD, iEDA, Yosys etc.)
  - *Continuous updates to public, reproducible baseline results, benchmarks*
- IEEE CEDA DATC
  - *ML EDA formats, datasets + proxy design enablements*
- The SLICE project
  - *Enabling a sharable ML infrastructure*
- MLCAD24: Reproducibility initiatives
- NSF "ImageNets for EDA" Workshop
- …



Industry Efforts

Contributions to ML infrastructure and shared datasets

- GT-NVIDIA contest
  - Enable LLM-assisted design automation
- Si2's AI/ML Schema Open Standards Working Group
  - Developing standardized **schema** specification to support AI/ML methods
  - Goal: enable academia-industry collaboration
- Google open-source ("N7") Ariane RISC-V core
  - → New PD benchmark reflecting sub-10nm process technology
  - + Scaled 2x, 4x variants

AI Flywheel and Frictionless Reproducibility

- **AI flywheel**

More users → More data → Better AI → Better product → More users

- **Frictionless reproducibility (FR)** [Donoho, 2024]

Benchmarking

Data sharing | **Code sharing** | **Competitive challenges**

Benchmark requirements

- Periodically update and diversify
- Prioritize representative datasets
- Use consensus evaluation metrics
- Apply standardized testing protocols

Impact goals

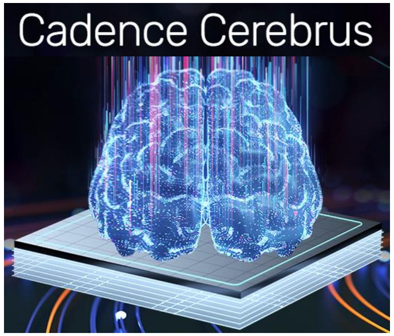
- Reflect real-world scenarios
- Ensure model accuracy and robustness
- Drive progress with meaningful comparisons

Agenda

- Data
  - Data Outside vs. Inside IC Design
  - Challenges and Ongoing Efforts (Academia and Industry)
- ML Deployment
  - Key Performance Indicators (KPIs) and Checklists
  - Machine Learning Operations (MLOps) and Commoditization
- Challenges for LLM Deployment
  - LLM X EDA: Software Engineering Issues
  - Challenges from EDA Flows

ML Deployment: 3 Basic Strategy Elements

- **Focus on optimizing existing design processes**
  - Make measurable improvements while building **trust** in ML integration
  - Examples:
    - Cadence Cerebrus
    - Synopsys DSO.ai



- **Aim for incremental improvements**
  - Less risk (minimum project cost)
  - Simpler rollbacks if necessary
- **Treat data as a first-class, up-front concern**
  - Robust data is the foundation for ML success
  - Data quality determines model performance
    → effective data management is essential

UCSD

Key Performance Indicators (KPIs)

- **Progress Tracking fuels success ← KPIs !**
  - Assessment of progress toward expected outcomes
  - Feedback for timeline adjustment
- **Common KPIs for ML projects**
  - **Operational efficiency** KPIs: how ML improves business processes
  - **Customer satisfaction** KPIs: how ML enhances user engagement and satisfaction
  - **Revenue growth** KPIs: how ML improves sales and marketing
- **KPIs for ML deployment in IC physical design**
  - Improvements in **license utilization** or efficiency of license usage
  - Number of RTL or P&R **iterations per week**
  - **Ratio of (automated vs. human)** explorations of floorplan or timing closure recipes
  - …

Checklists: Data and ML Methods

- **Does the output actually follow from all the input data?**
  - Without implicit (silent, unspoken) assumptions and human intuition
  - Without magically solving NP-hard problems, PDEs, etc.
- **Is there enough training data to fully capture the functionality?**
  - Isolated corner cases can be a problem
- **Does a given ML method work well with a given data type?**
  - LLMs X numerical data, graphs, etc.
  - Deep Learning X big structured data, multiscale data

Data

- **Is there enough training data to train a given ML model?**
  - Data-hungry: GenAI and Deep Reinforcement Learning
  - Less data-hungry: Gradient-Boosted Decision Trees (XGBoost)
  - Data-frugal: Bayesian methods
- **Is there too much data for a given ML model?**
  - May need to use RAG and/or a Mixture of Experts
- **Are the model training speed and cost consistent with updates to data?**
  - Can model freshness (relevance) be maintained?
  - New tool versions, design ECOs, library models, …

ML Methods

Acknowledgments and thanks: Dr. Igor Markov, Synopsys

More Checklists: ML Model Outputs

- **What are the comparisons to existing baselines?**
  - Pitfall: lack of strong baselines, benchmarks
- **How do output errors scale with size?**
- **Must output errors be found?**
- **How are output errors tolerated?**
  - Verify and fail?
  - Verify and retry?
  - Hot-patching (e.g., hallucinations, statistics)?

Outputs consumed by People:	**Outputs consumed by Tools:**
• Is output size limited?	• Are there too many errors (at scale)?
• Is correctness obvious?	• Is there a fast verifier and corrector?
• Are fixes obvious?	• Do ML outputs improve final QoR?
• Do ML outputs save or waste time?	

UCSD

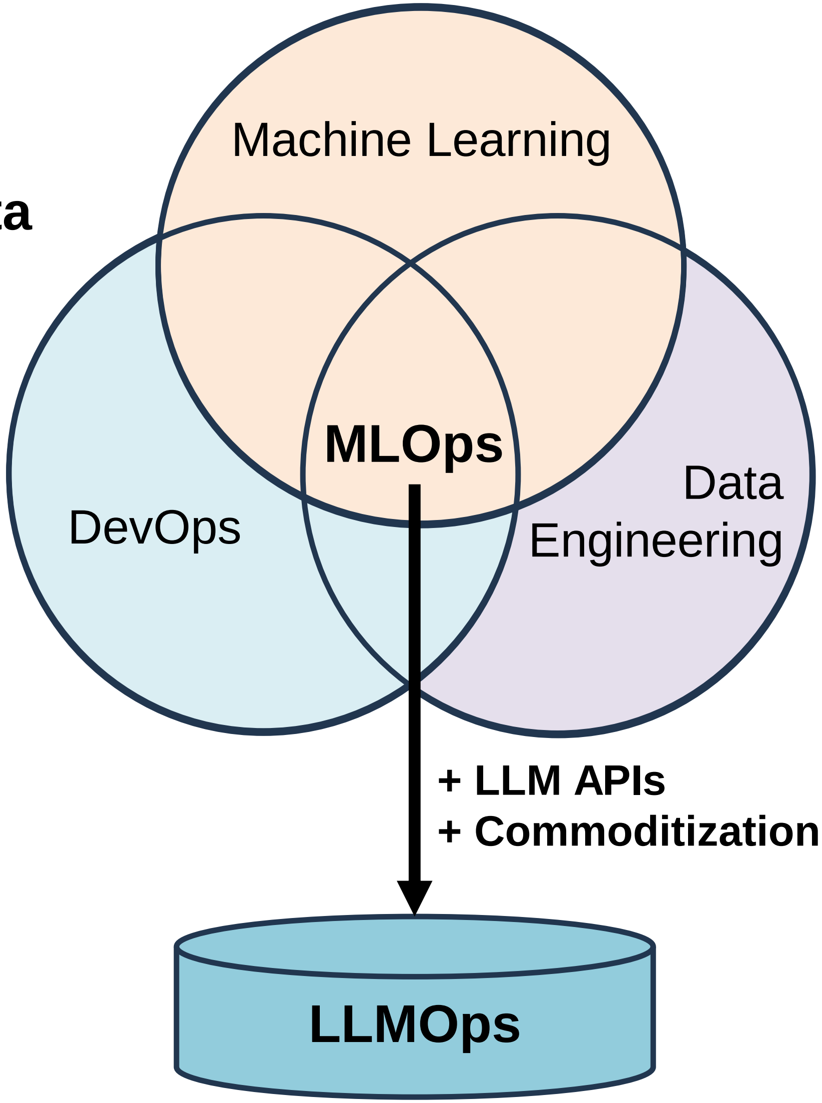
Acknowledgments and thanks: Dr. Igor Markov, Synopsys

What About “MLOps”?

- **Wikipedia definition:** “[A]n engineering practice that leverages three contributing disciplines: **machine learning**, **software engineering** (especially DevOps), and **data engineering**”

Checklists for MLOps

- **How will data be archived from runs?**
  - E.g., streaming vs. batch
- **What elements of run data?**
  - E.g., logs, scripts, reports, collaterals, …
- **How to manage the lifecycle of design data, or the model store?**
- **Who creates VectorDBs (for RAG) and fine-tuned models – and when should these be updated?**
- **Where is the compute?**
  - E.g., volume, cost of data movement



Engage with MLOps NOW !

LLMOps: The New Paradigm

- LLM-only applications often call for a **simpler MLOps subset** !
- Work with API for a model, not the model itself

- **Much lower costs** due to no training/serving, only API calls
- Monitoring/observability is in effect most of it

LLMOps: Commoditization "Via Osmosis"

- VC funded startups can enter
- As can traditional monitoring companies e.g., Datadog

LangSmith

W&B

DATADOG

LlamaIndex

- Easier LLM monetization lowers cost of MLOps
- "Via Osmosis": lower subset's cost → then entire cost

No MLOps →($) **LLMOps** →($) **Full** MLOps

No MLOps → Full MLOps: $$$$

Agenda

- Data
  - Data Outside vs. Inside IC Design
  - Challenges and Ongoing Efforts (Academia and Industry)
- ML Deployment
  - Key Performance Indicators (KPIs) and Checklists
  - Machine Learning Operations (MLOps) and Commoditization
- Challenges for LLM Deployment
  - LLM x EDA: Software Engineering Issues
  - Challenges from EDA Flows

LLMs for Software Engineering

- Very trendy, already 3 young unicorns
  - *Cognition, Magic, Cursor/Anysphere*
- No sign of slowing down
- Revenue: $100M+ for Cursor alone
- Newcomers keep coming: Aider, Zed, ...
- Next: an EDA copilot? **No.**
  - *"When you put the resulting netlist into P&R, make sure to fence this region with top-side ports folded onto two layers, and turn on higher-effort congestion mitigation with the following set of path groups …"*

LLMs Excel at Particular Codebases

Many criteria "must" be met !

For a self-driving car:
"Clear Weather"

- Length of execution chain before a tangible result
- Open-source **monorepo** in a popular language
- Low, ideally **zero** important third-party **dependencies**
- In-line, sufficient **documentation** for codebase
- Low bar for **domain knowledge**
- Little need for integration of non-text based information
- Small **compute** requirement to test a small change
- **Integration** with tools, e.g., browser-based UI render

Many standard codebases already satisfy these !

On the Other Hand … EDA ?

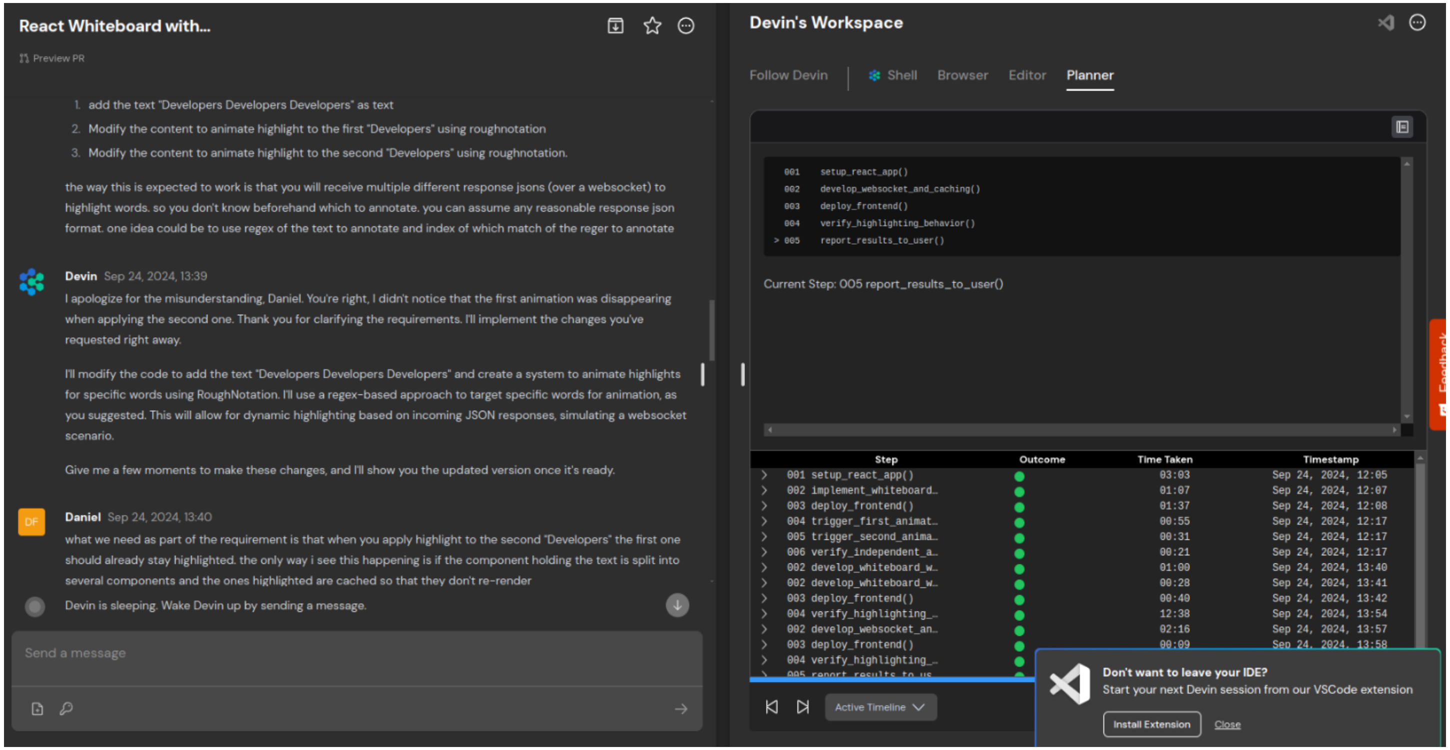
EDA lacks a lot of this…

For a self-driving car:
“Snowstorm”

- SOTA EDA and open-source EDA are ~disjoint
- Extensive **domain knowledge** is a must
- **Feedback** on changes can take **days** or even weeks
- Mixture of Verilog, C++, Python for ORFS
- Endless **sea of self-contained tools**: Yosys, ABC, …
- Lack of in-line comments and lots of documentation
- Reliance on platform kits that are **third-party imports**
- Copyright/IP rights + LLM leaks = **nightmare** scenario
- **… (… --- …)**

Open Source (alone) Isn't the Answer

- React with Devin *(compare this with OpenROAD-flow-scripts …)*

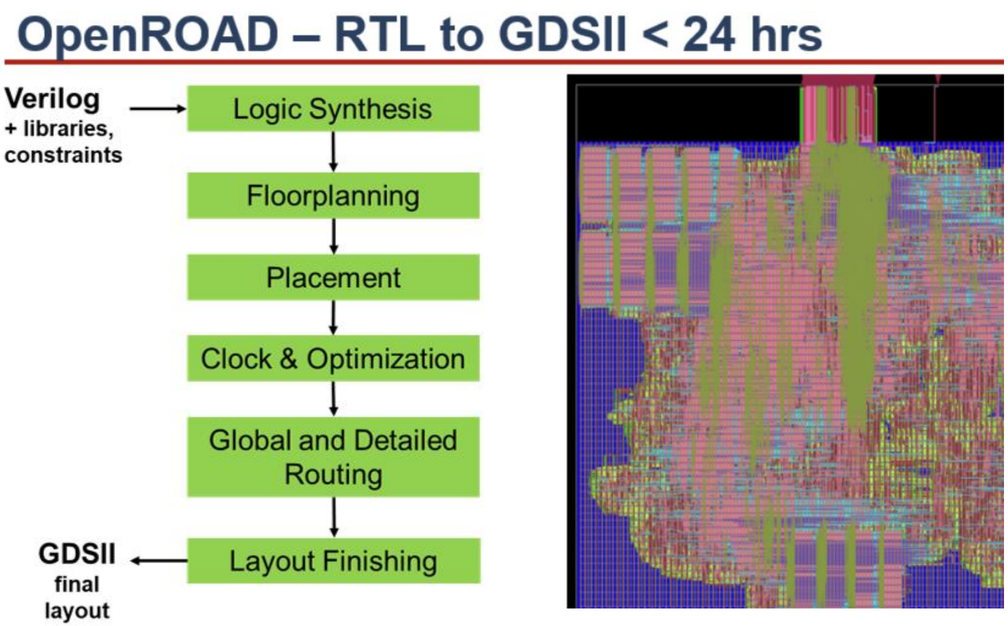


- Easily broken-down subtasks
- Predictable timeline of human feedback
- Browser window/terminal responds to changes
- Self-contained, modular changes

UCSD

Roadblocks and Culture Clashes ☹

- Trinity of roadblocks: Data, Integration, Modularity
  - Large nested EDA structures (netlists, 5-box data model, etc.)
  - Third-party black boxes: Yosys, ABC, etc.
  - EDA “chains” tools together to make a flow
    - Versus self-sufficient APIs that respond quickly



24 hrs = **Fantastic** for EDA !!!

24 hrs = **Awful** for an iterating LLM-based agent !!!

And: **Frictionless Reproducibility** vs. **Proprietary Outputs / No Benchmarking**

And Yet …

- Roadblocks and culture clashes notwithstanding:
  LLMs + EDA is receiving very active VC funding and attention !

SILOGY

Test/Debug

SILIMATE

Verilog Copilot

LLMs for chip design

LLMs for Chip Design

By Garry Tan

Y Combinator CEO soliciting LLMs for chip design companies

ChipAgents AI

General agents for chip design

Acknowledgments and Thanks

- **For “Deployment” content in a recent IEEE webinar**
  - Igor Markov and Thomas Anderson (Synopsys)
  - Scot Weber (AMD)
  - Rod Thorne and Steve Brown (Cadence)
  - Siddhartha Nath (Intel)
  - Tuck-Boon Chan (Qualcomm)
- **For organizing this special session**
  - Professor Youngsoo Shin (KAIST)
  - ASP-DAC 2025 organizing committee

Summary

- **Data for AI/ML is still fraught, still top of mind**
  - Make **Frictionless Reproducibility** and the AI flywheel real !
  - “Competitive Challenges” = baselines, benchmarks, culture change !
- **Deployment of AI/ML in physical design**
  - **High-level precepts**
  - **KPIs**
  - **Checklists** for data, models, outputs, …
- **LLM + EDA**
  - **Trinity of roadblocks: Data, Integration, Modularity**
  - Deep culture clashes
- **And Yet …**
  - Many exciting possibilities and challenges (opportunities)
  - **Industry and investors are eager to see successes!**

Thank You

UCSD