
Summary Report

# NSF Workshop on Shared Infrastructure for Machine Learning EDA

| | |
|---|---|
| Workshop Date: | March 10, 2023 |
| Workshop location: | Kenneth H. Keller Hall<br>200 Union Street SE<br>Minneapolis, MN 55455 |
| Sponsor: | National Science Foundation |

## Acknowledgement

This workshop was sponsored by the National Science Foundation (NSF) CISE/CCF Division under grant (CCF-2310319). We thank the NSF Program Director, Dr. Sankar Basu, for the support of this workshop. We are grateful to all the workshop speakers, attendee participants, and roundtable panelists for their insightful and stimulating presentations and discussions. Many of the participants have directly contributed to the writing of this report. The workshop program and a complete list of the speakers, attendees, and panelists are provided in the appendix.

## Workshop Organizing Committee

- Yiran Chen, Duke University
- Vidya A. Chhabria, Arizona State University
- Cong "Callie" Hao, Georgia Tech
- Ramesh Harjani, University of Minnesota
- Jiang Hu[1], Texas A&M University
- Andrew B. Kahng, UC San Diego
- Mike Quinn, Texas A&M University
- Sachin S. Sapatnekar, University of Minnesota
- Aakash Tyagi, Texas A&M University

[1]For questions and comments, please contact Jiang Hu (jianghu@tamu.edu).

## Executive Summary

Machine Learning (ML) in Electronic Design Automation (EDA) is an emerging field that is being explored for its potential to revolutionize chip design and verification. ML EDA uses machine learning algorithms to extract knowledge from data in chip design and verification, which can lead to unprecedented efficiency compared to conventional approaches. However, ML EDA faces a key bottleneck in data preparation, which can be very costly. Currently, each development team individually repeats the same effort on collecting data to varying degrees of success. A shared infrastructure would facilitate a "flywheel" effect: more shared data would lead to better models, which could lead to improved design tools and flows with larger gains, inviting more users who draw from the infrastructure, who in turn, could feed more data into the ecosystem. The wider participation from this virtuous cycle perpetuates the flywheel, resulting in progressively greater benefit to both the research and practitioner communities at large.

To identify key challenges and opportunities in the creation of a shared infrastructure for ML EDA, the NSF sponsored a one-day workshop on the University of Minnesota campus on March 10, 2023. The workshop assembled over 70 academic researchers (from 25 universities), industry experts (from 11 companies), government officials, and other stakeholders who came together with the objective to seek answers to the following questions:

- Is shared infrastructure (data, interface, testcases, scripts, etc.) for ML EDA needed? What are the examples of applications?
- What is an achievable scope for the shared infrastructure? Data based on only academic tools or additional commercial tools? Data based on only public domain PDKs or beyond?
- How will the shared infrastructure interoperate with existing EDA infrastructure?
- What are the major challenges and hurdles to the shared infrastructure? Any solutions to the challenges?
- How will academia and industry collaborate on this effort?
- How can the shared infrastructure be made sustainable and extensible in the long run?

The workshop was organized in three sessions as follows: Session 1 covered the theme of challenges facing the proliferation of ML EDA in academic research and industry practice. Session 2 shared recent progress made in the creation of datasets and proxies to enable and support the ecosystem surrounding the shared infrastructure. Session 2 also hosted breakout sessions to seek expert views on the goals and driving questions of the workshop. Session 3 brought invited speakers from the VC community and concluded with a panel discussion on the theme of "Pervasive AI in EDA through a shared ML infrastructure." This report summarizes the views expressed by the speakers, attendees, and panelists in written and spoken forms during the workshop.

A clear message from the workshop was that the IC design community in general, and the ML EDA community in particular, can greatly benefit from a high-quality, robust, and usable infrastructure that enables innovation with algorithms and models. Building on experience from the artificial intelligence (AI) community, the flywheel model is most effective when this shared infrastructure is open. Desiderata for shared infrastructure for ML EDA, which includes code, data, models, and support, should address the following elements.

- It should cover digital, analog, and mixed-signal design for multiple design paradigms, including but not limited to ASIC, SoC, FPGA, and heterogeneous integration.
- It should not only include open-sourced libraries and algorithms, but also distributed training framework recipes; modern, large-scale designs with high-quality labels; and datasets that conform to a standard form of shareable database.

- Where sharing is not possible (in the design/EDA industry, PDK models, IPs, and commercial EDA tools fall into this class), high-quality proxies (e.g., open PDKs, open IPs, open-source EDA tools) must be part of this shared infrastructure.
- It must address common and repetitive issues for the design community, such as the time-consuming and error-prone process of transferring data from design/reports to ML frameworks.

Ultimately, the shared infrastructure should enable proof of concept and allow for the effective use of new ML techniques while maintaining a high bar for quality, so that it can keep the ML EDA flywheel spinning.

The report concludes with the following additional overarching recommendations.

- To jumpstart the ML EDA flywheel, the government should support funded projects, where key components of the shared infrastructure for ML EDA are constructed. The projects are expected to build the initial momentum for spinning the flywheel of ML EDA ecosystem.
- ML EDA has many requirements and constraints that are specific to the semiconductor industry, but should also learn from experiences on shared infrastructures from AI and replicate successful strategies where possible, e.g., from the success of ImageNet, Kaggle, Hugging Face.
- The shared infrastructure should leverage existing open-source EDA resources, e.g., OpenROAD, OpenDB, open-source PDK, ALIGN, and MAGICAL, wherever possible.
- To mitigate the lack of data and barriers to access proprietary data, stakeholders in ML EDA should pursue methods for addressing the gap through approaches such as (1) benchmark release by industry for EDA contests; or (2) the use of generative ML models for synthetic yet realistic data.
- Security and privacy must be considered in constructing and maintaining the shared infrastructure for ML EDA.
- To make the shared infrastructure effort sustainable in the long term, it must involve collaboration with industry and other organizations, e.g., IEEE and Linux Foundation/CHIPS Alliance.

# Contents

# 1 Shared Infrastructure for ML EDA: What and Why?

*If you want to go fast, go alone. If you want to go far, go together.*

*– African Proverb*

## 1.1 The Need for Shared Infrastructure

Over the past few years, the adoption of machine learning (ML) applications in electronic design automation (EDA) has grown by leaps and bounds. An indicator of this trend can be observed in the increasing number of related publications (Fig. 1) and the growing number of adoptions in industrial EDA tools and design flows. Machine learning has been used in almost every part of the design process, including high-level synthesis, logic synthesis, verification, test, and physical design, and also in optimizing parameters of the design flow to obtain higher quality designs than ever before. The reason for this trend is that ML techniques can extract knowledge from data and achieve knowledge reuse with unprecedented efficiency compared to conventional approaches and human experts.

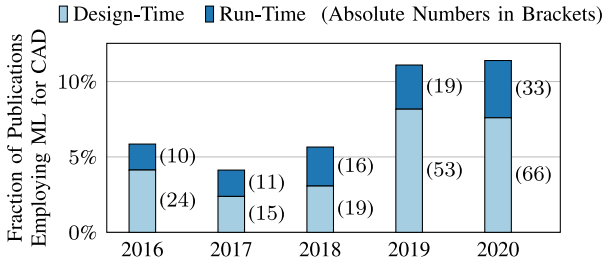



Figure 1: ML EDA publications in TCAD, DAC, ICCAD, ASP-DAC and ESWeek [1].

Further rapid progress of ML EDA is hindered by the lack of infrastructure support. Here, infrastructure refers to a collection of essential software resources used repeatedly in the development of ML EDA, such as training data, pre-trained models, generative models, scripts for data generation, software interface between EDA tools and ML platforms, testcases and testbenches, etc. The preparation of training data for ML EDA can be very costly and time-consuming. For instance, obtaining a labeled data sample through a design flow easily takes 3 hours or more for modern chip designs. Thus, 1000 data samples would require more than 4 months to generate. Furthermore, it is estimated that about 70% of ML EDA development time is spent on data preparation. Currently, each development team, whether in academia or industry, independently repeats the same effort in collecting data. Although industrial companies seem to own vast amounts of design data, the data are largely uncurated, fragmented across different teams, and far from ready to use. This repeated data preparation effort is unnecessary and leads to a significant waste of computing and human resources, ultimately slowing down the development turnaround time. Additionally, the lack of transparency and sharing can make it difficult to test and validate new techniques and approaches, which can slow progress in the field and limit its potential impact.

Furthermore, the infrastructure plays a crucial role in the overall ML EDA ecosystem, which can be considered as an instance of the flywheel of machine learning systems shown in Fig. 2. In the flywheel, (1) design data are collected and curated; (2) the data are utilized to construct and refine machine learning models; (3) the models are integrated with EDA software tools; (4) the tools are applied on chip designs and generate more data. The ever-broadening participation from this virtuous self-perpetuating cycle keeps the flywheel turning, resulting in progressively greater

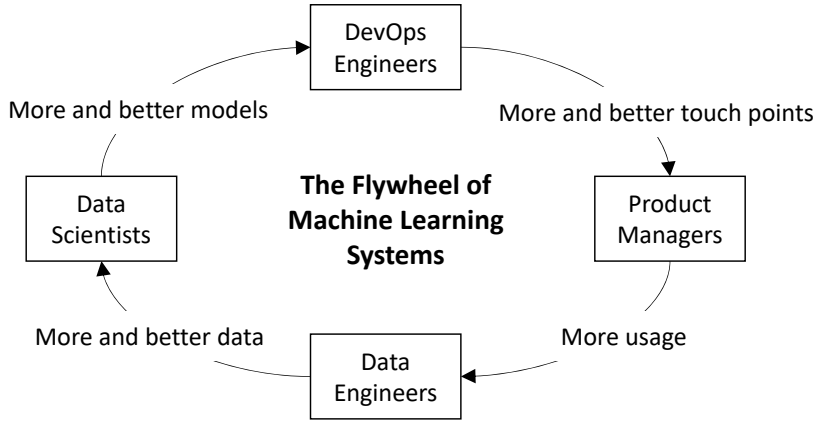



Figure 2: The flywheel of machine learning systems [Courtesy of Ruchir Puri].

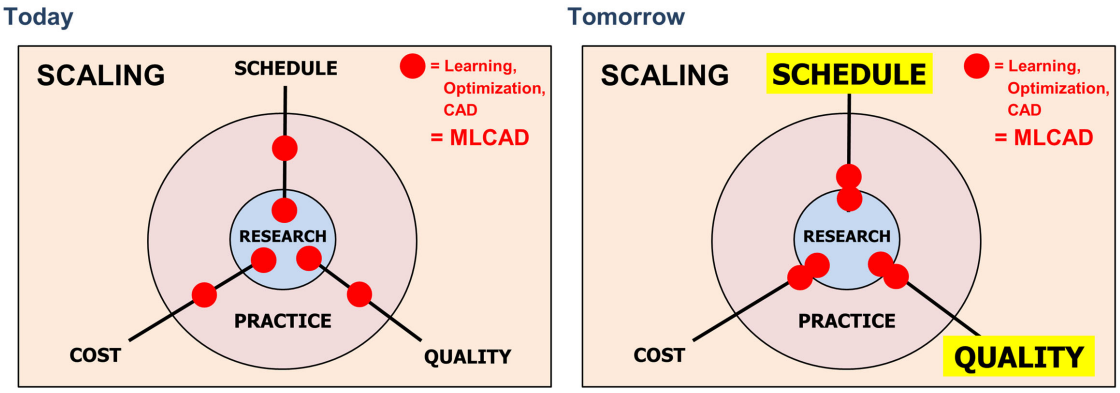



Figure 3: "Last scaling levers": Shared Infrastructure brings the red dots (ML, Opt, CAD) of Research and Practice closer together [3].

benefit to the research and practitioner communities at large. Today, the ML EDA flywheel is barely moving, if not at a standstill. Unfortunately, a direct and palpable consequence of this state is that innovation is stymied in many areas of chip design. For instance, the resource-intensive (human and compute) field of functional verification, that would otherwise act as the perfect playground for ML given its problem formulation, baseline solutions, and abundance of data, is starved for innovation as evident from scant application of advanced ML technology, zero shared source code, and zero sharing of trained models/weights.

Sharing among the EDA community can effectively address the infrastructure bottleneck. Collaborative sharing would eliminate the need for duplicated efforts and maximize the reuse of existing infrastructure. Additionally, sharing would facilitate the development of standardized datasets and protocols for data collection and analysis, making it simpler to reproduce and validate new techniques and approaches. By working together and pooling their resources, researchers and practitioners in the EDA community can accelerate progress in the field and unlock its full potential.

Today's design infrastructure is well into the last scaling levers, trying to claw back inefficiencies and whatever we left on the table while "riding the Moore's Law wave" [2]. In this scenario, the existential need is to derive future gains from efficiency improvements. As shown in Fig. 3, the difference between today on the left, and tomorrow on the right, is efficiency: bringing research and practice together and finding synergies between learning, optimization, and CAD.

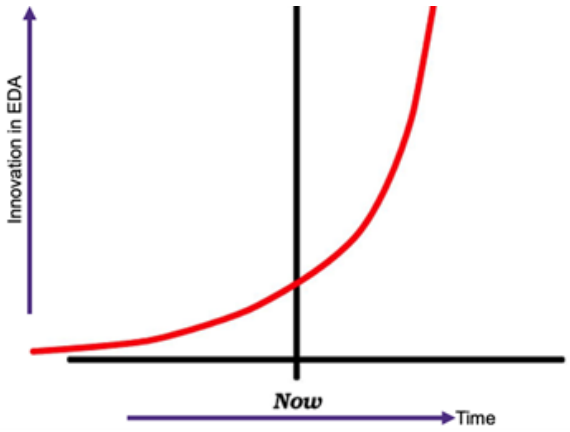



Figure 4: The promise of the ML EDA Flywheel with a shared infrastructure.

The gains extend beyond efficiency to (a) reliability with consistency in data, interface, test scripts and faster fault detection, and (b) scalability with the ease of scaling for larger team collaboration. As suggested in Fig. 4, we may well be at the cusp of setting forth on the long promised road rich with innovation in EDA, but the promise may only be realized with a humming ML EDA flywheel and the road traveled together with a shared infrastructure.

### 1.2 What Can We Learn from the Machine Learning Community?

The ML community has long recognized the value of sharing standard datasets, models, and open-sourcing code to promote the growth of the field. This approach has enabled researchers and practitioners to build on each other's work, test and compare different approaches, and develop new techniques and applications. By its very nature and definition, ML relies on massive data quantity and quality. It is therefore not surprising that areas with open-source data have benefited the most from the "AI revolution". For example, GPT-3 claims nearly 500B tokens, DALL-E claims 650M images, and ImageNet contains 14M images in over 20K categories [4]. Fei-Fei Li, who established ImageNet, has famously said "The paradigm shift of the ImageNet thinking is that while a lot of people are paying attention to models, let's pay attention to data. Data will redefine how we think about ML models."

The open-source ecosystem of data, models, and (largely) the "ML infrastructure" in these fields has enabled what Ruchir Puri of IBM describes as the "Flywheel of ML Systems" shown in Fig. 2, where more shared data can lead to better models, which can result in improved design tools and flows with larger gains, inviting more users who draw from the infrastructure, who in turn, could feed more data into the system. ML and its flywheel have uniquely benefited most in ecosystems where the feedback has been open-source, high volume, and high quality. For instance, on the Hugging Face platform [5] dubbed as the "Home of Machine Learning," the number of models has increased from 69,878 in September 2022 to 150,062 in March 2023, nearly doubling in a span of six months! The content and use of AI frameworks, platforms, and libraries such as TensorFlow, PyTorch, etc. and their enabling distributed computing environments are expanding just as rapidly. This is the spinning ML flywheel where all its components – usage models, engineering, science, and development – are fully synergistic. A shared infrastructure further accelerates the flywheel because it enables wider usage by eliminating the startup cost that entrants would otherwise have to bear, thereby reducing barriers to entry. Wider and deeper participation means more data and innovation, and therefore a faster flywheel. It is safe to say that any significant business value may

be realized by an enterprise only once its ecosystem's ML flywheel gets in full motion.

## 1.3 What To Share? What Not To Share?

The term "shared infrastructure" refers to an "open" approach where individuals and organizations have free access to a centralized repository of data that follows standard protocols and interfaces. This allows for greater collaboration and knowledge-sharing among members of the community, which can lead to more efficient development and deployment of machine learning techniques for EDA.

However, there are challenges when it comes to confidential and commercial data, which are "non-shareable". Certain components of the design process, such as Process Design Kits (PDKs) and Intellectual Property (IPs), are often proprietary and protected by copyright and non-disclosure agreements. In addition, export controls may limit the sharing of data across international borders. To tackle non-shareable data, one can develop effective proxy methods that allow machine learning models to learn from the available data without directly accessing the confidential or commercial components. For example, one approach is to use synthetic data generated through simulation or other techniques to augment the available data and enable more robust machine learning models. Another approach is to use transfer learning, where pre-trained models are adapted to new datasets without accessing the confidential components. The proxies may also include open-source PDKs, open-source EDA tools, public domain benchmarks and generative models.

We believe that it is critical to address the following issues in a shared and maintained infrastructure.

- **Data for ML Model Training**. Data of high-quality and standard format is the most important recipe for ML EDA.
- **Interface with Machine Learning Infrastructures**. PyTorch, Caffee, TensorFlow, Keras, MXNet; Platforms: H2O.ai, Databricks, Dataiku, Alteryx, RapidMiner; Programming and Environment: Julia, Jupyter Notebook, Colab.
- **ML Models for Multiple EDA Problems**. It is essential to establish unified frameworks for benchmarking, training, and hyper-parameter tuning. This framework should also include a standardized format and interface for data, models, and workflows to ensure seamless integration and compatibility across different tools and platforms.
- **Trustworthiness and Explainability**. The ability to interpret and understand the decisions made by the machine learning models is essential for ensuring the safety and reliability of the resulting chip designs.
- **Distributed Computing**. Distributed computing handles larger datasets and more complex tasks. Techniques such as data parallelism and model parallelism can be used to distribute the computation across multiple machines or GPUs.
- **Testcases, Benchmarks and Validation Systems.** Shared testcases, benchmarks and validation systems are essential for facilitating easy reproducibility and comparison among newly developed ML EDA techniques. This is crucial for obtaining meaningful feedback and advancing the field.

## 1.4 Examples of Successes

We list below a set of examples of successful open-source EDA infrastructures that are released under permissive licenses such as BSD-3.

- OpenROAD [6–8] is an RTL-to-GDSII flow for digital ASIC designs. It is a collaborative effort between industry and academia, with contributors from major semiconductor companies,

leading research institutions, and government agencies. OpenROAD provides a comprehensive set of tools and workflows that enable users to design, optimize, and verify custom ICs with greater efficiency and accuracy. The framework is built on top of industrial-strength open-source elements, such as OpenDB and OpenSTA, and the Python programming language, and is designed to be highly modular and extensible.

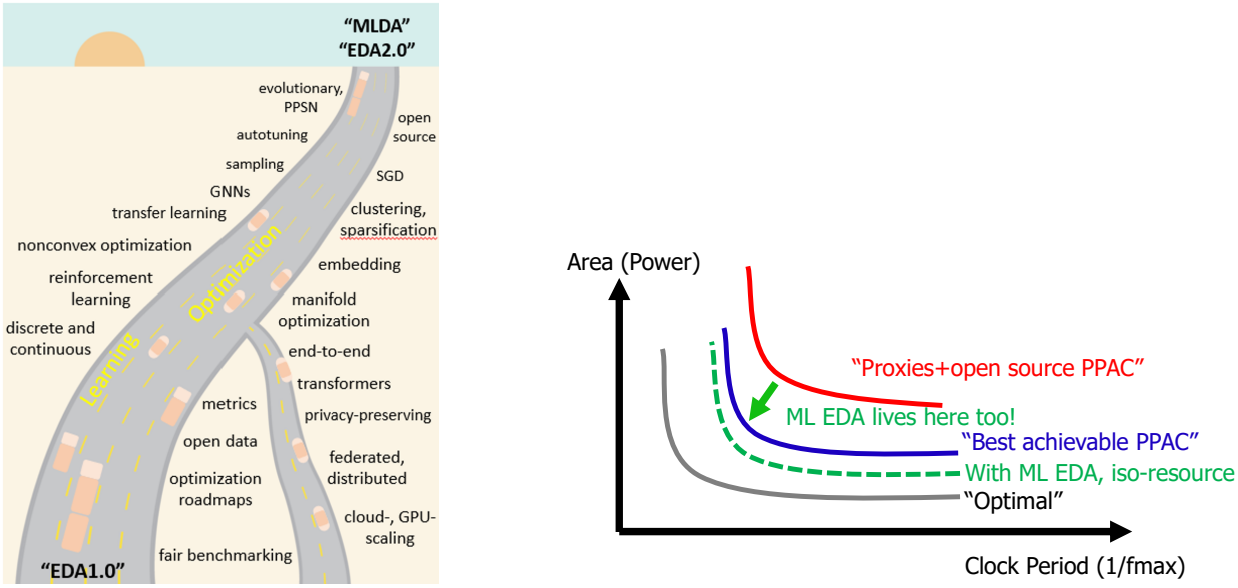
- ALIGN [9–11] is an automated layout generator for FinFET and bulk designs that automatically translates an unannotated (or partially annotated) SPICE netlist of an analog circuit to a GDSII layout. ALIGN can build analog layouts for multiple classes of circuits: low-frequency components (e.g., analog-to-digital converters, amplifiers, and filters); wireline components for high-speed links (e.g., equalizers, clock/data recovery circuits, and phase interpolators); RF/wireless components (e.g., components of RF transmitters and receivers), and power delivery components (e.g., capacitor- and inductor-based DC–DC converters and low dropout (LDO) regulators).
- MAGICAL [12, 13] generates layouts for analog designs from a netlist, focusing on bulk technology nodes. Designer insights and expertise are strategically embedded into MAGICAL through pattern matching, heuristics, and deep learning techniques.
- BAG2 [14, 15] is framework for the development of process-portable analog/mixed-signal circuit generators based on designer-coded scripts. Such generators are parameterized design procedures that produce schematics, layouts, and verification testbenches for a circuit with given input specifications.
- FreePDK [16] is an open-source process design kit for semiconductor technology nodes.
- OpenRAM [17] is an open-source memory compiler for digital ASIC designs.
- OpenCores [18] is an online community for the development of IP cores.
- Yosys [19] is an open-source logic synthesis software.
- OpenABC [20] is a large-scale labeled dataset generated by synthesizing open-source hardware IPs.
- Verilog-to-Routing (VTR) [21] is an open-source framework for FPGA synthesis and layout.

### 1.5 Need a "Base Version" of Infrastructure to Start Spinning EDA's AI Flywheel

EDA's ML flywheel needs a base version of an open ecosystem of large data, trained models, and supporting automation. The ML EDA ecosystem, not uniquely but especially given the required pace of semiconductor innovation, stands to benefit from vigorous, simultaneous, and synergistic participation from academic and industry researchers and practitioners. The ML flywheel runs on the "exhaust" of data – and only with an open ecosystem of data, models, methodology and more (which basically requires open-source EDA) can we hope for this kind of progress. Unquestionably, the need of the hour is a shared infrastructure for ML EDA to unblock new ML modeling and EDA optimization opportunities and set the ML EDA flywheel in motion. The graphic from a 2022 ISPD keynote [2] in Fig. 5a shows how the field might advance over the next 5+ years – with some items further along than others. Fig. 5b shows the "hockey stick" Pareto tradeoff between power and performance (clock period) for a digital design. The best possible power/performance/area/cost (PPAC) tradeoff today is known to be sub-optimal, and we know ML EDA can show great value in shifting this "hockey stick" downwards. Shared infrastructure is a vital piece for unblocking barriers and speeding up progress in ML EDA.

The shared infrastructure for ML EDA ideally needs to satisfy the following requirements.

- **Data quality**. Refining and obtaining high-quality data is crucial in any machine learning project. By ensuring that the data is accurate, consistent, and complete, researchers can




(a) Pathway for ML-enabled EDA

(b) Promise of ML EDA in delivering better PPAC

Figure 5: EDA roadmap and efficiency with ML EDA [2].

avoid wasting time and resources on cleaning and transforming data. Additionally, high-quality data can help researchers to identify patterns and relationships more easily, as well as reduce the risk of making incorrect conclusions due to errors in the data. Therefore, it is essential to invest time and effort in ensuring that the data used for machine learning EDA applications is of the highest quality possible.

- **Repeatibility**. Repeatability ensures that the research results can be independently verified and reproduced by others, including methods, datasets, and models. By promoting repeatability, researchers can help ensure that the EDA community, along with its technology leading edge, grows and develops in a sustainable and efficient manner.
- **Data Volume**. The availability of large and diverse datasets is crucial for the development and application of advanced machine learning models, particularly in EDA. By leveraging large datasets, EDA researchers can gain insights into the design process, identify patterns, and develop advanced ML models to automate and optimize chip design. Large datasets also enable the development of more robust ML models that can handle more complex tasks. In other domains, they have provided more diverse sets of images for image recognition and improved the accuracy of language models for natural language processing. Overall, the availability of large and diverse datasets is critical for the growth and advancement of machine learning, and in the context of EDA can help address the challenges of designing complex and efficient circuits.

# 2 Shared Infrastructure for ML EDA: Challenges and Solutions

*Many hands make light work*
*– Unknown*

*The journey of a thousand miles begins with one step.*
*– Lao Tzu*

As discussed in Section 1, a shared infrastructure is needed to enable the ML EDA flywheel and unlock the massive potential of innovation in EDA. A straightforward interpretation of shared infrastructure in the context of ML EDA includes code, data, models and testcases. Before proceeding to address the challenges to the creation and maintenance of shared infrastructure and potential solution space, let us first establish some basic terms. Anything that is shareable must be open, as in more permissive, in order to enable an uptick in productivity. In contrast, elements of the infrastructure that are not shareable would hold back productivity and would require the creation and growth of proxies to complete the flywheel. Challenges that must be overcome may be in the form of *bars* and *barriers*. Bars are thresholds such as critical mass of functionality, critical quality, baseline volume of user base, and a number of silicon proofs. Barriers block the flywheel, e.g., an IP that is under proprietary control, or license agreements that prevent researchers from openly sharing work. Next, we will list each major challenge accompanied by its potential solution space.

## 2.1 Challenge #0: Missing/Misplaced Incentives

**Challenge**: The development of shared open-source ML EDA software requires collaboration between academia and industry (EDA and Chip Design), which may have different priorities and incentives.

**Solution Space**: The incentive models for academia, EDA, and chip design companies need to be studied and a win-win-win model must be proposed that takes into account values and priorities. For instance, the creation and maintenance of infrastructure is tedious by nature and contains a big share of "non-researchy" components. The incentives in such a case for academia must be tailored to establish one's path to career progress. For instance, open-source contributions, such as user support and bug-fixing, need to be included in academic hiring and promotion criteria. It will also be helpful to have industry support for student internships and a path for assimilation upon graduation. The incentive model for EDA companies may involve a clear path for import of new research insights and collateral (models, applications). The support of government agencies will be vital in enabling such a model. In addition, philanthropic investment will be a valuable resource that can be leveraged.

## 2.2 Challenge #1: Access and Permissions

**Challenge**: EDA is data-rich (diversity and magnitude). In principle, industry research and development groups have access to a wide swath of advanced circuits that are implemented in cutting edge technologies. Chip design and EDA companies generate nightly runs *en masse* ranging from hundreds to hundreds of thousands. However, this apparent "problem of plenty" hides the fact that there are major barriers to sharing this data with the community at large: there are often non-disclosure agreements associated with design data; sharing data involves the labor-intensive process of curating information from design databases and overcoming internal barriers across groups within companies; etc.

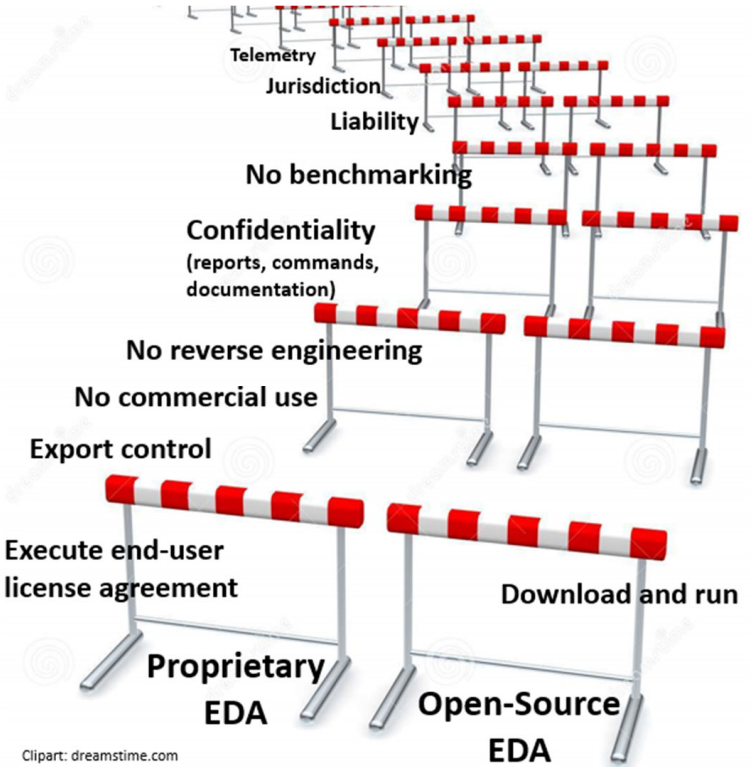



Figure 6: A pictorial view of traditional EDA barriers [22].

Designs are large and diverse, but again they remain under proprietary control as company-owned intellectual property. In some cases, a design/EDA company may be working on designs that they are not at liberty to share as the design is owned by another company. Process Design Kits (PDKs) are proprietary too, with strict NDAs imposed by foundries such as TSMC, Intel, Global Foundries. Tool reports are plentiful and diverse, but many that are generated with commercial EDA tools remain largely locked or under license/EULA restrictions as illustrated in Fig. 6. This poses a significant challenge for EDA vendors, who are hesitant to share such data with third parties, especially their competitors. There are several reasons why EDA vendors are reluctant to share their proprietary data. First, sharing such data could lead to the loss of competitive advantage, as other vendors may be able to replicate or improve upon their products using this data. Second, there are legal and contractual restrictions on sharing such data, such as non-disclosure agreements and intellectual property rights. Finally, there are concerns about data privacy and security, especially given the sensitive nature of the data involved. Recent developments, where EDA companies have permitted tool scripts to be shared publicly for research purposes, are a small step in the right direction, but much more remains to be done to enable an open and shareable ecosystem.

Overall, large and diverse data, IPs, and supporting infrastructure associated with chip design and verification are largely locked away or disallowed from being shared. The process of investigating whether sharing can occur is mired in company legal layers and by the time legal approval would arrive, the purported benefit is far outweighed by the risk of being obsolete.

**Solution Space**: To address these challenges, we must find ways to protect EDA vendors' confidentiality of their data while still enabling ML-based EDA development. There will also be massive opportunities for EDA vendors to collaborate with academic researchers and government agencies to develop standardized datasets and models for use across the industry. This would enable EDA

vendors to contribute to the development of ML-based EDA while still protecting their proprietary data. A shareable infrastructure must be open, and open-source tools, flows, methodologies, and design/process data are essential to make discernible positive impacts. However, in the foreseeable future, not all components of this infrastructure can be made open-source: in this environment, any non-shareable components of a shared infrastructure must be represented by proxies. For instance, proxy PDKs and enablements (e.g. ASAP7 [23]), proxy EDA tools (e.g. OpenROAD, ALIGN, MAGICAL) that we can benchmark and record data from, relevant proxy testcases and design drivers (e.g. RISC-V, NVDLA [24], OpenPiton [25], Chipyard [26], etc.) represent potentially acceptable proxies for industry-strength components of the shared infrastructure. In cases where today's available proxies are not good enough, it is essential to invest in making them good enough to meet and remain above the bar.

In addition to design infrastructure, it is also important to have a shareable corpus of design data for training EDA ML algorithms. Some efforts have attempted to harvest available data from open-source databases [18, 27], but the quality of these designs is often suboptimal. Generating a large amount of training data can be challenging due to the complexity and diversity of the data involved. Traditional approaches to data collection and labeling can be time-consuming and costly, making it difficult to obtain the quantity of data required for ML-based EDA development. One solution to this challenge is to use generative and synthetic data to produce large quantities of training data. Generative models can learn the underlying distribution of the data and then use this distribution to generate new samples of data that are similar to the original data; an early approach is described in [28]. Synthetic data, on the other hand, is created by modeling the underlying physics of the system being designed and using this model to generate new data.

Generative and synthetic data have several advantages over traditional data collection approaches. Large quantities of training data can be produced more quickly and at a lower cost, with potential coverage of a wider range of design types or operating conditions. However, generative and synthetic data also have some limitations. One challenge is ensuring that the generated data is representative of the original data and covers the relevant operating conditions. Another challenge is ensuring that the generated data is of sufficient quality to be useful for training ML models. Finally, there is a risk that the ML models trained on generative and synthetic data may not generalize well to real-world data, which can limit their effectiveness in practice. To address these challenges, it is important to carefully design and validate the generative and synthetic data generation processes using testbenches and validation systems.

### 2.3 Challenge #2: Model Training

**Challenge**: The creation and maintenance of large, diverse, and sophisticated trained models are key to the success of shared infrastructure, but this requires a large volume of data, which itself suffers from the issues described in Challenge #1 above. EDA vendors report that even the models that only have weights from training on customers' designs are not allowed to "leave the company floor" due to legal restrictions.

**Solution Space**: Aside from the obvious solution of organically growing the open-source base of the shared infrastructure with large, diverse, and sophisticated trained models, one may also seek potential methods of model obfuscation. Learnings and techniques from recent research related to the generation of ML datasets for digital and analog EDA with GANs and obfuscation may be sought in open-source adaptation of trained models that would otherwise not be allowed to be shared.

## 2.4 Challenge #3: Relevance and Quality

**Challenge**: Shared infrastructure must start from a position of credibility and maintain it thereafter. Two important attributes in this context are relevance and quality. Relevance may be measured in functionality (scope, diversity) and how current it is with the times, and applies to both the inception and maintenance over-time. Quality is measured both in results as well as in support, and needs to be ensured with significant investment. One of the most important factors in building reliable and effective ML models for EDA is the quality of the underlying data. The accuracy, completeness, and representativeness of the data can have a significant impact on the performance and practicality of the resulting ML solutions.

Commercial EDA tools rely on large amounts of proprietary data, such as PDKs, testbenches, and other design and verification data, which are often subject to strict data confidentiality agreements. Additionally, different EDA vendors may use different data formats and structures, which can make it challenging to develop standardized ML models that can be used across multiple platforms. Academic, open-source tools and IP blocks typically have limited support and documentation, and lack of standardization. Therefore, quality should cover data format consistency and usability as well.

One challenge is that the data may be incomplete or inconsistent, making it difficult to develop accurate models. For example, if a model is trained on incomplete or biased data, it may not generalize well to new designs and verification scenarios. Another challenge is the need for constant retraining of models to prevent "model drift." Design IPs and technology nodes (feature size, design rules, etc.) are advancing at such a rapid pace that changes are significant between generations, necessitating frequent and costly retraining to avoid model drift.

**Solution Space**: Addressing these challenges requires more open and standardized data formats and structures, as well as tools and platforms for sharing and accessing data across different EDA vendors. In addition, efforts are needed to open up the black box of EDA tools to enable the generation and collection of more representative data. This can involve creating APIs and other interfaces inside commercial EDA tools that allow data to be generated and collected programmatically. Another approach is to encourage and make use of open-source EDA tools that are designed with data generation and collection in mind. Where needed, we must seek to create acceptable proxies for industry-strength components of the shared infrastructure. If the proxies are not good enough today, then we must invest in making them good enough to meet and remain above the bar. The decision-making process of ML models must be transparent and understandable to designers and engineers to ensure the reliability and trustworthiness of EDA tools. Models must undergo training and retraining to prevent drift so as to maintain relevance. Finally, the first version of any offering need not be perfect, but good enough to enable enthusiastic community participation to get the flywheel spinning.

## 2.5 Challenge #4: Continuous Improvement

**Challenge**: Improvements in ML EDA must be tightly coupled with tool improvements. ML can find ways for an EDA tool to set better defaults [29–32]. For example, ML can discover better defaults for the router or help refine the routing recipe. But this can only go so far when the EDA tool is a black box. To fully unleash the power of ML EDA, greater visibility within this black box is crucial in order to access features that can be used by an ML model.

**Solution Space**: In principle, it is possible to access data in commercial tool flows through Tcl interfaces. However, this access is often slow, incomplete, and requires clunky translation to the

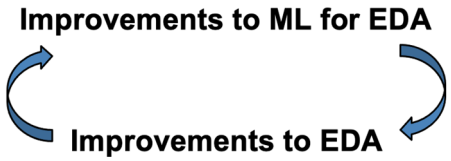



Figure 7: Regenerative loop for continuous improvement.

Python world of ML algorithms. Open-source design flows overcome these issues by providing full visibility – including source code – to the innards of the EDA algorithm. Today, open-source digital design flows are not at the level of capability of commercial digital flows, but the EDA community must come together and build these capabilities up to the point where they are at industry strength. There is reason for optimism here: while commercial EDA flows have evolved over $\sim$40 years, open-source efforts have made progress at a steeper rate over the last five years. On the analog design side, current commercial offerings are very limited, and open-source design tools offer great promise. Substantial investment in open-source digital and analog EDA infrastructure is essential to trigger the regenerative loop shown in Fig. 7 within a shared ecosystem.

### 2.6 Challenge #5: Analog Design Automation

**Challenge**: Unlike digital design, analog design is largely unautomated: much of analog design is still performed with intense human designer involvement. Therefore, there is a large potential for ML solutions to make an impact by helping replicate, or even improve upon, the capabilities of the human designer. In short, analog EDA is a distinct area of interest as it is a still-developing field where there are few mature industry EDA solutions, and high potential for ML.

ML infrastructure for analog design automation faces its own challenges. Analog design is a highly specialized field that involves manual effort and custom simulations to ensure the performance and reliability of the analog circuits. While analog circuits may only occupy a small portion of the overall real estate on an design, they are critical to the overall functionality of the system, especially for data conversion, amplification, communications, I/O interfaces, and special circuits such as thermal sensors and clock generators/PLLs. Thus, ML infrastructure for analog design automation faces at least two challenges: (1) analog design automation is not yet a mature field; (2) unlike digital ICs, which use uniform PPA (power, performance, area) metrics across designs, the figures of merit for analog circuits are different for each design. This has conventionally invited manual design effort due to the specialized knowledge required to design each circuit.

**Solution Space**: Unlike in digital domains where machine learning has improved existing automation techniques, in analog design automation, machine learning has the potential to play a transformative and enabling role, as has been shown in early efforts in this field [11, 13]. Hence, the development of the ML infrastructure must be closely integrated with the development of analog design automation techniques. Open-source analog design automation tools, such as ALIGN and MAGICAL, are valuable resources that should be leveraged effectively. The challenge of diversified design metrics can be addressed by intimate collaborations with analog designers. Training data is a little easier to obtain in the analog (as opposed to digital) domain as there is a higher volume of analog design activities in academia. There is also a plethora of synthetic data in the analog domain (e.g., UT-AnLay for analog performance prediction for OTA). Continued investment in shared infrastructure for ML-based analog design automation will be crucial for solving the long-standing challenge of analog design efficiency and productivity.

# 3 Characteristics and Attributes of the Shared Infrastructure

*Change what you can; accept what you can't; have the wisdom to know the difference.*

*– Serenity Prayer (paraphrased)*

*The greatest transformation brought about by technology is when you bring the various pieces and have them work together in combination. It's the synergies that bring about the greatest changes in the world.*

*– Malcolm Gladwell*

## 3.1 Desired and Achievable Scope for the Shared Infrastructure

In the big picture, the shared infrastructure need not be differentiating and simply needs to exist with a high bar for quality (both initial and sustained) in robustness, usability, and extensibility. As an analogy, the data models, database, and features such as standards support, PDK support, logging, scripting, GUI, etc. should resemble plumbing and utilities, which we take for granted. In this regard, the notion of "don't need to think about it" is the appropriate bar to aim for, so that researchers' bandwidth can be focused on innovation with algorithms and models. Tools need not beat commercial versions, but should be good enough to verify the effectiveness of new ML techniques for enabling proof of concept. Ideally, shared infrastructure must accelerate progress by solving issues that are common and repetitive for the design community. For example, the process of transferring data from design/reports to ML frameworks is a universal commodity need, which is also time-consuming and error-prone.

The shared infrastructure needs not only open-sourced libraries and algorithms (such as TensorFlow Agents) but also distributed training framework recipes due to the need for large-scale simulation and training. In terms of testcases (RTL, netlists, circuits), modern, large-scale designs with high-quality labels generated with industry-strength EDA tools will be highly desirable. Where data sharing is a barrier, tools and techniques such as GNL (GenerateNetlist), GANs or generative AI may be employed to generate highly obfuscated synthetic cases that are relevant to industry, using in-house design benchmark suites with various directives as labels. The goodness of synthetic test cases must be assessed and established. In physical design, for example, the utilization of cells, clock domains, congestion, blockage constraints, region constraints, etc. must be considered to match both the "hardness" and "realness" of typical industry-relevant testcases. In functional verification, testcases/datasets for coverage closure may include stimulus and cover groups/points, whereas testcases/datasets for debugging may manipulate assertions with bug injection and error measurement.

In analog design, the datasets need to cover more types of analog circuits and take advantage of open PDKs such as SKY130, SKY90, and GF180. Training data for analog ML models can be obtained through curating academic analog designs, existing synthetic data and performing open-source software such as ALIGN and MAGICAL in collaboration with analog designers.

Finally, it may be instructive to study the EU's TRISTAN (Together for RISc-V Technology and Applications) project to share key learnings for scope given the parallels. TRISTAN is a multi-year project aimed at expanding and industrializing the European RISC-V ecosystem to compete with existing commercial alternatives. The project defines an European strategy for RISC-V based designs, creates a repository of industrial quality building blocks, and covers both EDA tools and the full software stack. The consortium consists of 46 partners from industry, research organizations, universities, and RISC-V related industry associations from various countries. The project aligns

with the European Commission's strategy to support the digital transformation of all economic and societal sectors, including the development of new semiconductor components to retain technological and digital sovereignty. The TRISTAN approach leverages the open-source community to boost productivity, improve security, increase transparency, allow better interoperability, reduce cost to companies and consumers, and avoid vendor lock-ins.

### 3.2 Data Generation and Representation

The shared infrastructure should include pre-generated data, readily available for immediate use, and scripts that can generate data on demand. Pre-generated data can help to reduce data generation costs significantly, while scripted data generation can provide greater flexibility to meet customized needs. Both should conform to established standards for reproducibility and comparability, ensuring that results can be consistently replicated and compared across different contexts. The field of ML EDA can draw inspiration from ML conferences that have data tracks for training datasets, and establish similar efforts. By creating standardized datasets, researchers and practitioners can compare their models and results more easily.

In addition, a shareable database (e.g., OpenDB) or circuit/netlist presentation can greatly simplify the repetitive engineering efforts needed to do ML EDA research while also making datasets extensible. The open EDA environment of the OpenROAD project offers the OpenDB database for physical design. The structure of OpenDB is based on the text-file LEF (library) and DEF (design) formats and their implicit physical design data model. OpenDB also supports a binary file format to save and load the designs much faster than using LEF and DEF. Meanwhile, the data format must be suitable for downstream ML tasks and environments. An example is graphML [33], an XML-based file format used for storing and exchanging graph data structures. It is commonly used in machine learning and data science for representing graph-based models, such as social networks, chemical structures, or knowledge graphs. Similar data formats, developed for the EDA context, can help turn the ML EDA flywheel.

### 3.3 Software Interface between ML and EDA Tools

Conversion from the EDA tool format, which is Tcl-friendly in commercial EDA environment, to a Python-friendly form that is consumable in the PyTorch environment will be greatly beneficial to data engineer/data scientist productivity by lowering the barrier to entry. State-of-the-art commercial EDA tools utilize Tcl and Scheme as their primary interface. While in and of themselves, Tcl and Scheme are deemed powerful and used successfully by the EDA and design community, they are widely acknowledged to present a low bandwidth interface and are unfamiliar to the ML community.

### 3.4 Shared Infrastructure Sustainability and Extensibility

The sustainability and extensibility of shared infrastructure will be crucial for achieving efficient ML EDA in the long run. ML is making inroads enabling efficiency improvements in EDA solutions but as noted during the workshop sessions, both technology and design complexity are growing fast as well. The infrastructure used for ML models must be sustainable and extensible to ensure that the benefits can be maintained and scaled over time. Sustenance may be enabled as long as the shared infrastructure and its constituents continue to scale and provide value to stakeholders with quality, user support, and omnipresent recognition that keep stakeholders engaged. The shared infrastructure must not be rigid or brittle, in order to maintain extensibility.

Sustainability in shared infrastructure means that the infrastructure can maintain its quality and performance over time, even as the data and requirements change. This requires careful attention to the design of the infrastructure, as well as ongoing monitoring, code inspection and bug fixing. Additionally, sustainability requires meticulous attention to details and quality, industry-relevant solutions, a high degree of customer orientation in support, and a keen eye toward ensuring that incentives remain intact to encourage continued participation from all players throughout the ecosystem that the infrastructure serves.

Extensibility in shared infrastructure means that the infrastructure can be easily modified and expanded to accommodate new ML models, data sources, and requirements. This requires a flexible and modular design, as well as open standards and APIs that enable interoperability and integration with other systems. A "crawl-walk-run" design and implementation plan will go a long distance in ensuring incremental and continuous success. Additionally, to maintain extensibility, we can set up frameworks or repositories that allow others to contribute to the infrastructure. For instance, Metrics4ML [34] provides a repository for design metrics data and scripts that can reproduce the experiments using OpenROAD. Extensibility also requires careful documentation and version control, so changes can be tracked and replicated.

### 3.5 Testcases, Benchmarks, and Validation Systems

The shared infrastructure intends to facilitate the reproducibility and comparability of newly published ML EDA techniques. Without this infrastructure, it would be challenging to distinguish truly effective and practical techniques from those with hidden and significant drawbacks. Such ambiguity poses a risk that further research and exploration might be based on shaky grounds and misled in deviated directions. Public domain testcases and benchmarks are fundamental for achieving reproducibility and comparability. A successful example is the placement and routing benchmarks that were released in physical design contests in collaboration with the industry.

The infrastructure for ML EDA is generally more complicated than that of conventional EDA due to the involvement of ML platforms. As a result, additional elements, such as scripts and configuration files, are necessary to form complete validation systems in addition to testcases. Building such validation systems will require collaboration between academia and industry.

### 3.6 Security and Privacy

Security and privacy are a serious and complex problem for open-source software and repositories, and ML EDA is not immune from this challenge. There are valid concerns about the possibility of reverse-engineering and model extraction attacks in EDA. Such security risk must be well addressed in the shared infrastructure so that community members, especially industrial companies, feel confident in contributing data and models, even if they are obfuscated. To achieve this, open-source commit/contributions must be regulated via established policies. Concurrent with the first step in establishing the shared infrastructure is the initiative to seek security and privacy protocols. For example, ML models can be trained through federated learning so that sensitive design data can be utilized without being directly exposed. The EDA community is new to such practice and would be well served to take inspiration and adopt best-known methods from longstanding, successful open-source initiatives in the software and ML community. Engagement with such bodies, such as the Linux Foundation and its subsidiary, CHIPS Alliance, will be beneficial in implementing known best practices, rather than reinventing the wheel.

# 4 Recommendations

*What we do in life echoes in eternity.*

*– Gladiator (the movie)*

*If you look at history, innovation doesn't come just from giving people incentives; it comes from creating environments where their ideas can connect.*

*– Steven Johnson*

Based on the discussions in this report, we make the following recommendations calling for action.

## 4.1 Jumpstart the ML EDA Flywheel

NSF should play a coordinating and enabling role in bringing together academia, EDA, and chip design companies with the goal of removing obstacles that have stymied the ML EDA flywheel from spinning. The appropriate levers of incentives and strategic alignment should be exercised to overcome the challenges noted in previous sections. Dr. Eric Schmidt, formerly the CEO of Google and Executive Chairman of Alphabet, attended the workshop in his capacity as Executive Chairman of Steel Perlot, a venture firm that has interests in ML EDA. He suggested the seeds of what might be needed to enable the flywheel:

> "Build an open-source interoperable platform that everyone agrees to collaborate on and then you compete with proprietary modules on top or proprietary value add, services, and so forth. Getting there is difficult and the incumbents will oppose it but the incumbents must embrace it because it gives them higher interoperability and therefore good for the incumbents too. We need to enable new designs. It is difficult to take an open-source AI module and plug it into a proprietary system. The new AI modules are architected in a shared service model that are cloud-based. The end goal should be that anybody using these tools can build cloud-based something that is easy to assemble with diverse pieces, is interesting, and that sells. We also cannot do it with a few grad students so we need industry, government, academia partnerships to enable this."

## 4.2 Facilitate Open-Source Ecosystem Learning from AI, SW, and Healthcare

We suggest that the EDA community learns from the open-source models in AI, software and healthcare communities. The barrier of entry for open-source models in AI has been amazingly low as evidenced by the explosive growth of data sourced from all walks of society, and models created and curated by a burgeoning community of data scientists and ML experts. The recent and rapid emergence of companies leveraging ML is a testimony to this trend (example: Landing.ai). The software and healthcare communities have facilitated ecosystems for strong and collaborative research in academia, whether it be from sustained incentives or in response to crisis (e.g. Operation Warp Speed in search of a vaccine to fight the COVID-19 pandemic). Key players in the EDA community include EDA vendors and chip design companies, for whom the incentive model and urgency for shared infrastructure is unclear. It would be helpful to dig deeper to understand the similarities and differences between the EDA ecosystem and other ecosystems that sustain the AI, software and healthcare industries, then crystallize a viable incentive model that brings together the EDA ecosystem.

### 4.3 Promote the Establishment of Standards and Best Practices

To ensure the quality and reliability of ML-based EDA solutions, it is essential to establish standards and best practices. This can be facilitated through the development of industry standards and guidelines, along with the establishment of benchmark datasets and performance metrics. Industry partners can also collaborate with academia to create standardized datasets that can be used by researchers to develop and test ML-based EDA solutions.

Shared Intermediate Representation (IR) for data can be helpful. One successful example is the Labeled Property Graph (LPG) data represented by relational tables, shared among multiple Generative AI (GAI) tasks. A unified database with a standard yet extensible data format for storing training data is beneficial.

### 4.4 Leverage Existing Open-Source EDA Resources

To overcome the barriers of proprietary data, design and tools, existing open-source EDA resources ought to be leveraged as proxies. These include open source PDKs such as ASAP7 [23] and SKY130, digital synthesis tools such as OpenROAD [8] and Yosys [19], analog design automation tools such as ALIGN [11] and MAGICAL [13], and public domain benchmarks such as OpenCores [18].

### 4.5 Construct ML-Friendly Software Interfaces

To connect ML platforms, which are typically developed using Python, with traditional EDA infrastructure that mainly relies on Tcl, software interfaces are essential. Open frameworks like OpenROAD and ALIGN already include work in progress to provide a rich Python interface along with Tcl. Additionally, access to internal data structures is a significant value for feature extraction as well as real-time decision-making.

### 4.6 Boost Collaboration between Industry and Academia

The development and maintenance of ML EDA infrastructure require closer collaboration between academia and industry than ever before. Industrial companies possess design data of the latest technology, which may not be curated, and cannot be directly shared. However, these data can be utilized to train public models using federated learning, while preserving data privacy. Alternatively, the data can be obfuscated and released through research contests, like existing placement and routing benchmarks. Moreover, the data can be internally used to validate whether proxy data and results are truly industry-relevant. Additionally, industrial companies can provide incentives to the shared infrastructure in which they stand to benefit. Lastly, the code and data quality, as well as the long-term sustainability of the infrastructure, undoubtedly require assistance from industrial experts.

Reciprocally, collaborations with academia would benefit industrial companies. In the fast-changing field of machine learning, new techniques emerge on a monthly basis, making it difficult for industrial companies to allocate sufficient resources for comprehensive studies while managing daily development operations. Conversely, exploring new ML techniques is a daily routine for many academic research activities, making academia a pioneer in identifying the latest ML techniques relevant to EDA industry. Collaborations can also help train students with both EDA and ML skills, which are highly demanded in the EDA industry. Additionally, EDA and chip design companies can directly leverage the shared infrastructure resulting from these collaborations, improving design efficiency.

We recommend the following action items for the collaboration.

- The semiconductor industry should support effort to build the shared infrastructure through funded research projects, e.g., SRC projects.
- EDA and chip design companies can organize research contests related to ML EDA, like IBM and Intel did for the ISPD contests. These contests can release obfuscated yet industry-relevant data and benchmarks, allowing researchers to develop and test new ML techniques while preserving data privacy.
- Forming an advisory board of industrial experts can provide critical feedback on the quality and sustainability of the shared ML EDA infrastructure.
- Academic researchers can give seminar talks and tutorials to industrial audience, introducing the infrastructure along with showcasing the state-of-the-art ML EDA techniques.
- Students participating in the infrastructure development can do internships at related companies to demonstrate the usage of the infrastructure in industrial settings and how the infrastructure can benefit industrial ML EDA development.

### 4.7 Develop Frameworks to Assess Proxy Quality

As stated earlier, to ensure that the shared infrastructure for ML-based EDA faithfully reflects the industrial level quality while maintaining data confidentiality, it is important to create the appropriate proxies. Proxies are stand-ins for the actual data and tools used in the EDA process, which can be used for research purposes while maintaining the confidentiality of the actual data and tools. One way to assess proxy quality is to compare the performance of ML-based EDA solutions developed using the proxies to those developed using the actual industrial data and tools. This can be done by collaborating with industry partners to obtain access to the actual data and tools, or by comparing the results obtained using the proxies to industry benchmarks or known results. Another way is to evaluate the accuracy and completeness of the proxy data and tools. This can be done by comparing the proxy data and tools to industry standards and specifications, and by conducting thorough testing and validation of the proxy data and tools.

For physical design in generic applications, we may consider diversity in netlist topology (Rent's exponent, hypergraph spectral properties, etc.), layout/floorplanning constraints, and timing constraints applied to all levels of IP design hierarchy. For domain-specific applications, we may consider domain-specific design structures (e.g. designs that pertain solely to ML or math kernels). For functional verification, diversity of design IPs (CPU, Memory, DSP, etc.) is a requirement. Open-source designs for CPU and ASICs are gaining strong adoption and reaching industry strength as many are scaling for the market.

### 4.8 Explore Generative ML for Synthetic Data

Generative AI refers to the use of machine learning algorithms to generate new data or solutions that are optimized for a specific task. In the field of EDA, generative AI has the potential to revolutionize the way designers approach complex problems. By learning from massive amounts of optimized design data, generative AI can directly generate design solutions that are tailored to specific requirements.

One successful example of generative AI in EDA is the use of generative adversarial networks (GANs) for synthetic power delivery network (PDN) data generation (BeGAN) [28]. This approach involves training a GAN on a large dataset of PDN designs to generate new, optimized designs that meet specific power delivery requirements. By using generative AI in this way, designers can solve the scalability challenge of implementing complex PDN designs at scale. Beyond synthetic PDN data generation, there are many other possibilities for using generative AI in EDA. High-level synthesis,

for example, involves automatically generating hardware designs from high-level specifications. By using generative AI, including methods based on large language models (LLMs) such as those used in Chat-GPT, designers could potentially create more efficient and optimized hardware designs with less manual effort. Logic synthesis involves optimizing digital circuits by converting high-level descriptions into low-level representations. Generative AI could be used to improve this process by automatically generating optimized logic circuits based on a given set of design requirements. Lithography, which involves the process of printing complex patterns onto silicon wafers, could also benefit from generative AI. Finally, gate sizing involves determining the optimal sizes of logic gates in a digital circuit to minimize power consumption and improve performance. Generative AI could be used to automate this process, facilitating the generation of a large volume of training data.

Overall, generative AI can serve as a crucial proxy for bypassing the need to deal with the confidentiality of industrial intellectual assets and obtaining industry-grade data.

### 4.9 Enable Scalable Data Management

To effectively manage the vast amounts of data required for successful ML in EDA, we need scalable infrastructure that can accommodate large volumes of data while maintaining data quality. This requires the implementation of new or evolving data management strategies that can efficiently handle the large and complex datasets required for ML EDA. Additionally, we need to increase persistent storage resources to ensure that data is easily accessible and available when needed. Normalized data access methodologies such as DataMesh [35] can also help to simplify and standardize data access, making it easier for ML developers to access and manipulate the data they need. Finally, normalized data transformations using libraries and services can also help to standardize the ML workflow and improve data quality, enabling more accurate and effective ML EDA.

### 4.10 Support Cloud-based Solutions for Improving Training Efficiency

Cloud-based solutions can provide the necessary computational power to facilitate ML EDA solutions. Major cloud service providers, such as Amazon AWS, offer on-demand access to high-performance computing resources. Additionally, there are academic projects like CloudBank (http://cloudbank.org) that aim to simplify the management of cloud resources. Cloud-based model training will play a pivotal role in the shared infrastructure for accelerating the turnaround time of ML EDA research.

### 4.11 Develop Sustainable and Extensible Infrastructure

To achieve shared infrastructure sustainability and extensibility in ML EDA, it is important to follow a few key principles.

- First, the infrastructure should be designed with a focus on modularity, scalability, and interoperability. This means using open standards and APIs and designing for easy integration with other systems.
- Second, quality and user support must remain the prime focus to encourage and retain participation. For instance, the shared infrastructure must provide evidence that the learned model on openly shared data behaves as well as on real commercial (possibly confidential) data, with limited retraining and fine-tuning effort.
- Third, ongoing monitoring and maintenance should be performed to ensure that the infrastructure remains sustainable and extensible over time.

## References


[1] M. Rapp, H. Amrouch, Y. Lin, B. Yu, D. Z. Pan, M. Wolf, and J. Henkel, "MLCAD: A survey of research in machine learning for CAD," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 10, pp. 3162–3181, 2021.

[2] A. B. Kahng, "Leveling up: A trajectory of OpenROAD, TILOS and beyond," in *Proceedings of the ACM/IEEE International Symposium on Physical Design*, 2022.

[3] ——, "MLCAD today and tomorrow: Learning, optimization and scaling," in *Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD*, 2020.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.

[5] "Hugging face: The AI community building the future," https://huggingface.co/.

[6] T. Ajayi, V. A. Chhabria, M. Fogaça, S. Hashemi, A. Hosny, A. B. Kahng, M. Kim, J. Lee, U. Mallappa, M. Neseem, G. Pradipta, S. Reda, M. Saligane, S. S. Sapatnekar, C. Sechen, M. Shalan, W. Swartz, L. Wang, Z. Wang, M. Woo, and B. Xu, "Toward an open-source digital flow: First learnings from the OpenROAD project," in *Proceedings of the ACM/IEEE Design Automation Conference*, 2019.

[7] T. Ajayi, D. Blaauw, T.-B. Chan, C.-K. Cheng, V. A. Chhabria, D. K. Choo, M. Coltella, R. Dreslinski, M. Fogaça, S. Hashemi, A. Ibrahim, A. B. Kahng, M. Kim, J. Li, Z. Liang, U. Mallappa, P. Penzes, G. Pradipta, S. Reda, A. Rovinski, K. Samadi, S. S. Sapatnekar, L. Saul, C. Sechen, V. Srinivas, W. Swartz, D. Sylvester, D. Urquhart, L. Wang, M. Woo, and B. Xu, "OpenROAD: Toward a self-driving, open-source digital layout implementation tool chain," in *Proceedings of the Government Microcircuit Applications and Critical Technology Conference*, 2019.

[8] "OpenROAD," https://github.com/The-OpenROAD-Project/OpenROAD.

[9] K. Kunal, M. Madhusudan, A. K. Sharma, W. Xu, S. M. Burns, R. Harjani, J. Hu, D. A. Kirkpatrick, and S. S. Sapatnekar, "ALIGN: Open-source analog layout automation from the ground up," in *Proceedings of the ACM/IEEE Design Automation Conference*, 2019, pp. 77–80.

[10] T. Dhar, K. Kunal, Y. Li, M. Madhusudan, J. Poojary, A. K. Sharma, W. Xu, S. M. Burns, R. Harjani, J. Hu, D. A. Kirkpatrick, P. Mukherjee, S. S. Sapatnekar, and S. Yaldiz, "ALIGN: A system for automating analog layout," *IEEE Design & Test*, vol. 38, no. 2, pp. 8 – 18, Apr. 2021.

[11] "ALIGN: Analog layout, intelligently generated from netlists," https://github.com/ALIGN-analoglayout/ALIGN-public.

[12] B. Xu, K. Zhu, M. Liu, Y. Lin, S. Li, X. Tang, N. Sun, and D. Z. Pan, "MAGICAL: Toward fully automated analog IC layout leveraging human and machine intelligence," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 2019.

[13] "MAGICAL," https://github.com/magical-eda/MAGICAL.

[14] E. Chang, J. Han, W. Bae, Z. Wang, N. Narevsky, B. NikoliC, and E. Alon, "BAG2: A process-portable framework for generator-based ams circuit design," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, 2018.

[15] "BAG2_cds_ff_mpt," https://github.com/ucb-art/BAG2_cds_ff_mpt.

[16] "FreePDK," https://eda.ncsu.edu/freepdk.

[17] M. R. Guthaus, J. E. Stine, S. Ataei, B. Chen, B. Wu, and M. Sarwar, "OpenRAM: An open-source memory compiler," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 2016.

[18] "OpenCores," https://opencores.org.

[19] "Yosys Open Synthesis Suite," http://yosyshq.net/yosys.

[20] "OpenABC," http://github.com/NYU-MLDA/OpenABC.
[21] "VTR," http://verilogtorouting.org.
[22] A. B. Kahng, "A mixed open-source and proprietary EDA commons for education and prototyping," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 2022.
[23] "ASAP7 PDK," https://asap.asu.edu.
[24] "NVDLA: Open Architecture for Deep Learning Accelerator," http://nvdla.org.
[25] "OpenPiton: Open Source Research Processor," http://parallel.princeton.edu/openpiton/.
[26] "ChipYard: an Agile RISC-V SoC Design Framework," http://github.com/ubc-bar/chipyard.
[27] "Build Custom Silicon With Google," https://developers.google.com/silicon.
[28] V. A. Chhabria, K. Kunal, M. Zabihi, and S. S. Sapatnekar, "BeGAN: Power grid benchmark generation using a process-portable GAN-based methodology," in *Proceedings of the IEEE/ACM International Conference On Computer Aided Design*, 2021.
[29] J. Kwon, M. M. Ziegler, and L. P. Carloni, "A learning-based recommender system for autotuning design flows of industrial high-performance processors," 2019.
[30] A. B. Kahng, L. Wang, and B. Xu, "TritonRoute: The open-source detailed router," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 3, pp. 547–559, 2020.
[31] ——, "The tao of PAO: Anatomy of a pin access oracle for detailed routing," *Proceedings of the ACM/IEEE Design Automation Conference*, 2020.
[32] R. Liang, J. Jung, H. Xiang, L. Reddy, A. Lvov, J. Hu, and G.-J. Nam, "FlowTuner: A multi-stage EDA flow tuner exploiting parameter knowledge transfer," in *Proceedings of the IEEE/ACM International Conference On Computer Aided Design*, 2021.
[33] "GraphML: a Comprehensive and Easy-to-Use File Format for Graphs," http://graphml.graphdrawing.org.
[34] "Metrics4ml," https://github.com/ieee-ceda-datc/datc-rdf-Metrics4ML.
[35] "Datamesh," https://datamesh.com.

# Appendix A – Workshop Agenda

| | |
|---|---|
| **Session 1** | Chairs: Sachin S. Sapatnekar (UMN) and Mike Quinn (TAMU) |
| 10:00 – 10:10 a.m. | Welcome Message<br>Dean Andrew Alleyne, College of Science and Engineering, UMN |
| 10:10 – 10:15 a.m. | Opening<br>Sankar Basu (NSF) |
| 10:15 – 10:35 a.m. | "Bars and Barriers to Overcome for Shared ML EDA Instructure"<br>Andrew B. Kahng (UCSD) |
| 10:35 – 10:55 a.m. | "Engineering the Flywheel of AI for Electronic Design Automation: Present Challenges and Future Opportunities"<br>Ruchir Puri (IBM) |
| 10:55 – 11:15 a.m. | "AI for Chip Design - An Industry Perspective"<br>Thomas Andersen (Synopsys) |
| 11:15 – 11:35 a.m. | "ML for Data-Driven Verification"<br>Dan Yu (Siemens EDA) |
| 11:35 – 11:55 a.m. | "Towards Large, High Quality and Open Datasets for ML4EDA"<br>Siddharth Garg (NYU) |
| **Session 2** | Chair: Jiang Hu (TAMU) |
| 12:30 – 12:50 p.m. | "Generating ML Datasets for Digital and Analog EDA: Opportunities and Challenges"<br>Sachin S. Sapatnekar (UMN) |
| 12:50 – 1:10 p.m. | "Enabling Generative AI and GPU Acceleration for EDA"<br>Mark Ren (Nvidia) |
| 1:10 – 1:30 p.m. | "Practical Considerations for Scaling AI/ML in an EDA Context"<br>Scot Weber (AMD) |
| 1:30 – 2:30 p.m. | **Breakout Session** |

· 1. Data: raw data or scripts? format, scope & pitfalls (S. Garg, T.-W. Huang)
· 2. Software interface between ML/EDA tools: scope & pitfalls (V. Chhabria, M. Robbins)
· 3. Open-source environment and platform extensibility (T. Ansell, C. Yu)
· 4. Testcases, benchmark and validation systems (M. Quinn, I. Bustany)
· 5. Collaboration between industry and academia (Y. Chen, C. Alpert)
· 6. Analog design automation (D. Pan, J. Hu)

| | |
|---|---|
| **Session 3** | Chair: Yiran Chen (Duke) |
| 2:45 – 3:00 p.m. | Eric Schmidt, Michelle Ritter (Steel Perlot) |
| 3:00 – 4:00 p.m. | Summary of breakout discussion |
| 4:00 – 5:00 p.m. | "Panel: Towards Pervasive AI in EDA through a Shared ML Infrastructure"<br>Moderator: Ismail Bustany (AMD)<br>Panelists: Srinivas Bodapati (Intel), Joe Jiang (Google), Sung-Kyu Lim (DARPA), Marcus Pan (SRC), Matt Robbins (Steel Perlot) |

## Appendix B – Workshop Attendees

| | |
|---|---|
| Charles J. Alpert | Cadence |
| Tim Ansell | Google |
| Thomas Andersen | Synopsys |
| Sanmitra Banerjee | Nvidia |
| Sankar Basu | NSF |
| Ben Beaumont | Cadence |
| Srinivas Bodapati | Intel |
| Ismail Bustany | AMD |
| Luca Carloni | Columbia University |
| Norman Chang | Ansys |
| Yiran Chen | Duke University |
| Vidya A. Chhabria | Arizona State University |
| Damian Dechev | NSF |
| Jana Doppa | Washington State University |
| Harry Foster | Siemens EDA |
| Paul Franzon | NC State University |
| Siddharth Garg | New York University |
| Andreas Gerstlauer | University of Texas at Austin |
| Christal Gordon | DARPA |
| Song Han | MIT |
| Cong "Callie" Hao | Georgia Tech |
| Abdelrahman Hosny | Brown University |
| Jiang Hu | Texas A&M University |
| Sharon Hu | NSF |
| Tsung-Wei Huang | University of Utah |
| Joe Jiang | Google |
| Eugene John | University of Texas at San Antonio |
| Lizy John | University of Texas at Austin |
| Jinwook Jung | IBM |
| Andrew B. Kahng | UC San Diego |
| Ramesh Karri | New York University |
| Brucek Khailany | Nvidia |
| Akhilesh Kumar | Ansys |
| Vaibhav Kumar | NXP Semiconductors |
| Peng Li | UC Santa Barbara |
| Rongjian Liang | Nvidia |
| Sung-Kyu Lim | DARPA |
| Frank Liu | Oak Ridge National Lab |
| Ethan Mahintorabi | Google |
| Robert Mains | CHIPS Alliance / Linux Foundation |
| Somdeb Majumdar | Intel |
| Yiorgos Makris | UT Dallas |
| Deepankar Medhi | NSF |
| Siddhartha Nath | Intel |
| Borivoje Nikolic | UC Berkeley |

| | |
|---|---|
| David Z. Pan | University of Texas at Austin |
| Marcus Pan | Semiconductor Research Corporation |
| Partha Pande | Washington State University |
| Massoud Pedram | University of Southern California |
| Ruchir Puri | IBM |
| Mike Quinn | Texas A&M University |
| Sherief Reda | Brown University |
| Mark Ren | Nvidia |
| Michelle Ritter | Steel Perlot |
| Matthew Robbins | Steel Perlot |
| Elyse Rosenbaum | UIUC |
| Mehdi Saligane | University of Michigan |
| Sachin S. Sapatnekar | University of Minnesota |
| Ioannis Savidis | Drexel University |
| Eric Schmidt | Steel Perlot |
| Savithri Sundareswaran | NXP Semiconductors |
| Sheldon Tan | UC Riverside |
| Aakash Tyagi | Texas A&M University |
| Shobha Vasudevan | Google |
| L.-C. Wang | UC Santa Barbara |
| Scot Weber | AMD |
| Marilyn Wolf | University of Nebraska – Lincoln |
| Xiaoqing Xu | Google |
| Cunxi Yu | University of Utah |
| Dan Yu | Siemens EDA |
| Ming Zhang | Self employed |
| Yanqing Zhang | Nvidia |
| Zhiru Zhang | Cornell University |
| Matthew Ziegler | IBM |