Summary Report for the

NSF workshop on Shared Infrastructure for Machine Learning EDA

Workshop Dates:	March 10, 2023
Workshop location:	Kenneth H. Keller Hall 200 Union Street Se, Minneapolis, MN 55455
Sponsors:	National Science Foundation

Acknowledgements

This workshop was sponsored by the National Science Foundation (NSF) CISE/CCF Division under grant *XYZ*. 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.

Yiran Chen, Callie Hao, Vidya Chhabria, Ramesh Harjani, Jiang Hu,[1] Andrew Kahng, Mike Quinn, Sachin Sapatnekar, Aakash Tyagi.

(Members of the workshop organizing committee)

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

Contents

Section 1: 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 accelerate the flywheel of ML systems by enabling wider and deeper participation, eliminating the startup cost that entrants would otherwise have to bear and thereby reducing barriers to entry. This wider participation means more data and innovation, which leads to a faster flywheel and a greater benefit to both the research and practitioner community 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, industry experts, 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? Examples of applications.
- What is an achievable scope for the shared infrastructure? Data based on only academic tools or additionally commercial tools? Data based on only public domain PDKs or beyond?
- How will the shared infrastructure interoperate with existing EDA infrastructure?
- What are major challenges and hurdles to the shared infrastructure? Any solutions to the challenges?
- How will academia and industry collaborate on this?
- How to make the shared infrastructure 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 further dialog on the goals of the workshop. Session 3 brought invited speakers from the VC community and concluded with a panel discussion with the theme of ‘Pervasive AI in EDA through a shared ML infrastructure’.

This report summarizes the views expressed by the speakers as well as the attendees and the panelists. Shared infrastructure in ML EDA includes code, data, models, support, incentives, and mandates. Some of the major challenges in creating and maintaining shared infrastructure include missing/misplaced incentives, access and permissions, model training, and continuous improvement. The potential solution space includes proposing a win-win-win incentive model that considers values and priorities for academia, EDA, and chip design companies, representing non-shareable components with industry-strength proxies, seeking potential methods of model obfuscation, and enabling the EDA community to come together in "picking one horse" for tools and enabling continuous improvement.

Ultimately, sharing means open-source, where more permissive usually means more positive impacts. The goal is to create a high-quality, robust, and usable infrastructure that enables innovation with algorithms and models. The shared infrastructure 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 OpenDB or any other standard form of sharable DB. The shared infrastructure needs to 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. In analog design, the shared infrastructure should cover a wide range of analog circuits and take advantage of open PDKs. DARPA's OpenRoad and EU's TRISTAN projects provide a useful model for the scope and early successes and learnings of the shared infrastructure and are worthy candidates for extension to include adaptors for ML EDA. 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.

The report concludes with the following recommendations to the NSF:

Open-Source ecosystem learnings from AI, SW and Healthcare

NSF should commission a joint industry-academia workshop on debating the learnings of open-source models in AI, Software and Healthcare communities. A workshop with the goal to understand the similarities and differences of the EDA ecosystem with the ones that sustain the AI, SW, and Healthcare industry, and then crystalize the incentive model for the EDA ecosystem will be timely and fruitful.

Warming up the ML EDA Flywheel

NSF should play a coordinating and enabling role in bringing academia, EDA, and chip

design companies with the goal to remove the knots that have stymied the ML EDA flywheel from spinning.

Accelerating and Sustaining the ML EDA Flywheel

NSF should consider funding a larger project on a 10-year scale with clear intermediate milestones, with the aim of encouraging multidisciplinary collaboration and continuity in support of the ML EDA flywheel.

Section 2: The Need for a Shared Infrastructure in ML EDA

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

- African Proverb

Artificial Intelligence (AI), and specifically, Machine Learning (ML) is being employed to solve some of the toughest problems, known and emerging, in practically every field that touches humankind. By its very nature and definition, ML relies on data, and lots of it. It is therefore not surprising that areas with open-source data have benefited the most from the "AI revolution". Take for instance, the fields of computer vision and natural language processing; GPT-3 claims nearly 500B tokens, DALL-E claims 650M images, ImageNet contains 14M images in over 20K categories [REF]. Fei-Fei Li, the founder of 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." Deep learning (DL) models such as convolutional neural networks (CNNs) solved the image recognition problem; generative adversarial networks (GAN) generate impressive real-looking images; reinforcement learning (RL) agents beat the best human GO players; and so forth.

The open-source eco-system of data, models, and (largely) the 'ML infrastructure' in these fields have enabled what Ruchir Puri of IBM describes as the Flywheel of ML Systems shown in Figure 1.

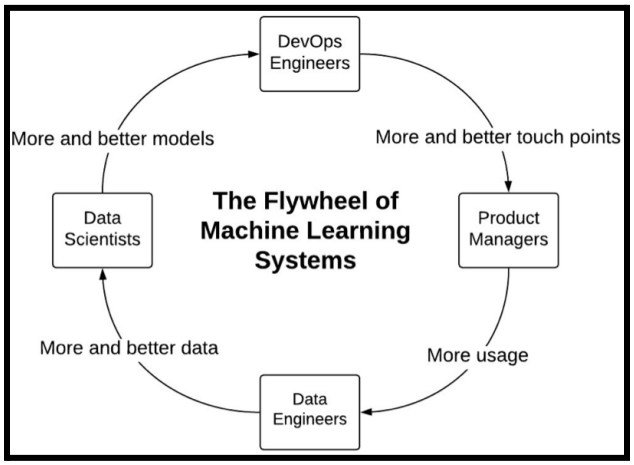

Figure 1. The flywheel of machine learning systems [Courtesy of Ruchir Puri, IBM].

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 [REF] dubbed as the ‘Home of Machine Learning’, the number of models have increased from 69878 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 Tensor Flow, PyTorch, etc. and their enabling distributed computing environments are expanding just as rapidly. This is the ML flywheel in acceleration where all its components - usage models, engineering, science, and devops - 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 and 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.

Electronic design automation (EDA) is a software technology that attempts to let computers undertake chip design and verification tasks whose complexities extend beyond manual design capabilities under given time and resource limitations. Although conventional EDA techniques have led to huge design productivity improvement, they face the fundamental limit that most EDA problems deal with combinatorial optimization and are classified as NP hard, and therefore have no polynomial-time algorithms for optimal solutions. For instance, the runtimes of many EDA tools on System-on-Chip (SOC) designs are now measured in weeks and not hours and days. As semiconductor feature size scales to sub-5nm and the number of devices on a chip increases to dozens of billions of transistors, such limitation has become even more pronounced; take placement and legalizer tools in SOC physical design as an example where they must now also be aware of intricate design rules in real-time. There is a compelling need for innovative changes. Systems designed with multi core CPU’s and GPU Accelerators have helped but they also stack up against the limits of (or imposed by) Moore’s Law, Dennard Scaling, and Amdahl’s Law.

A large number of heuristics employed in EDA rely on classification and decision making. EDA tools also have an abundance of data. See Figure 2 for instance that shares a profile of the data generated during the functional verification stage in chip design. And this is not limited to verification, but is true for all stages in chip logic and physical design and verification.

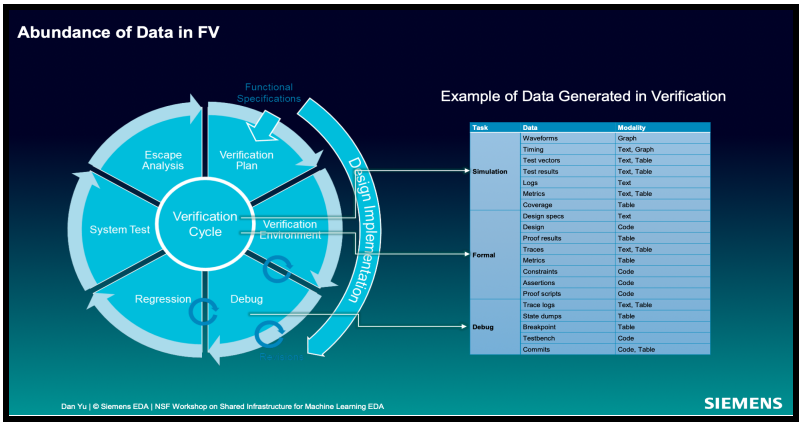

Figure 2. Abundance of Data in FV [REF].

These realizations of *large* data and the *fitting characteristics* of EDA algorithms have motivated the study of ML techniques for advancing the state of the art of EDA technology. This is indicated by both the growing number of related publications (Figure 3) and adoptions in industry EDA tools and design flows. There is no denying that ML techniques are able to extract knowledge from data and achieve knowledge reuse with unprecedented efficiency compared to conventional approaches.

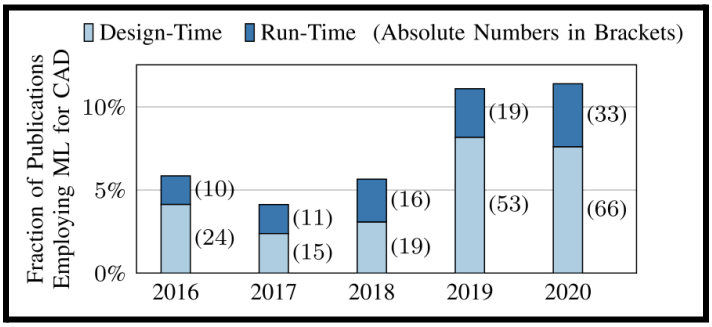

Figure 3. ML EDA publications in TCAD, DAC, ICCAD, ASP-DAC and ESWeek [REF].

Despite the aforementioned advantages, research and development of ML EDA faces a key bottleneck in data preparation which can be very costly. For example, if obtaining a labeled data sample through a design flow takes 3 hours, which is common for modern chip designs, then 1000 data samples would require more than 4 months to generate. It is estimated that about 70% ML EDA development time is expended on data preparation. Currently, each development team, academia and industry alike, individually repeats the same effort on collecting data to vastly varying degrees of effort and success. Although the EDA and chip design companies generate and own a significant volume of design data, the data remains uncurated, fragmented across different teams within a company, and siloed and protected under proprietary control,

and thus far from ready to use to the research and practitioner community at large. The repeated data preparation effort is arguably unnecessary, causes tremendous waste on computing resources and manpower, and largely slows down the development turnaround time. Apart from data, separated efforts on testcases and other infrastructure make ML EDA results difficult to be compared or reproduced. EDA is a fundamental pillar technology for chip design and semiconductor industry. In the flywheel of ML systems for chip designs, data and infrastructure constitute a critical component that pushes forward the progress in dealing with growing chip complexity and design productivity crisis.

Today, the EDA ML flywheel is barely moving, if not at a standstill. Unfortunately, a direct palpable consequence of this state is stymied innovation 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 trained models/weights. To underscore the issue, this has resulted in waste and low productivity of ML efforts as evidenced by impact to:

1. Data Quality: More effort spent in preparing data for research (as much as 70%)
2. Repeatability: Results are not easily verifiable and comparable
3. Data Volume: Too small to train advanced ML models

Needless to say, EDA's ML flywheel needs a base version of an open ecosystem of large data, trained models, and supporting automation. The EDA ML ecosystem, not uniquely but especially given the required pace of semiconductor innovation, stands to benefit from a vigorous, simultaneous, and synergistic participation from academia 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 EDA ML flywheel in motion. The cartoon from ISPD 2022 in Figure 4 shows how we might advance over the next 5-plus years – with some items further along than others.

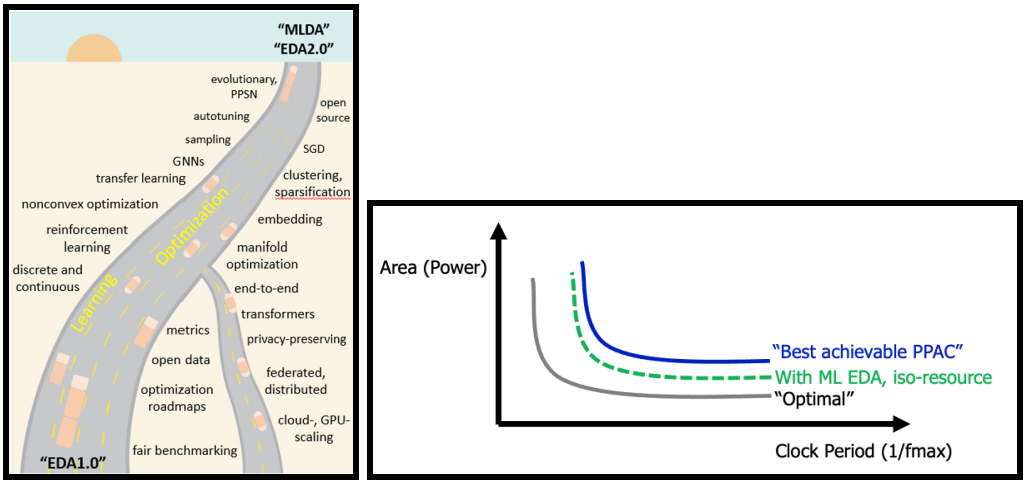

Figure 4. EDA Roadmap and Efficiency with ML EDA [REF].

Figure 4 also shows a cartoon of the hockey stick pareto of clock period on the X axis and area or power on the Y axis. Best possible PPAC today is not optimal – we don't know where optimal is – but we know ML EDA can shift this hockey stick, which has a huge value. There are definitely too many challenges, but too little time. So shared infrastructure is urgent because it unblocks and speeds up progress.

In closing, as Prof. Andrew Kahng noted in his 2020 MLCAD keynote [REF] titled "MLCAD Today and Tomorrow: Learning, Optimization and Scaling", we are well into the last scaling levers where we're trying to claw back inefficiencies and whatever we left on the table while "riding the Moore's Law wave", the existential need is to derive future gains from efficiency improvements. As shown in Figure 5, the difference between today on the left, and 'tomorrow' on the right, is really efficiency: bringing research and practice together and finding synergies between learning, optimization and CAD.

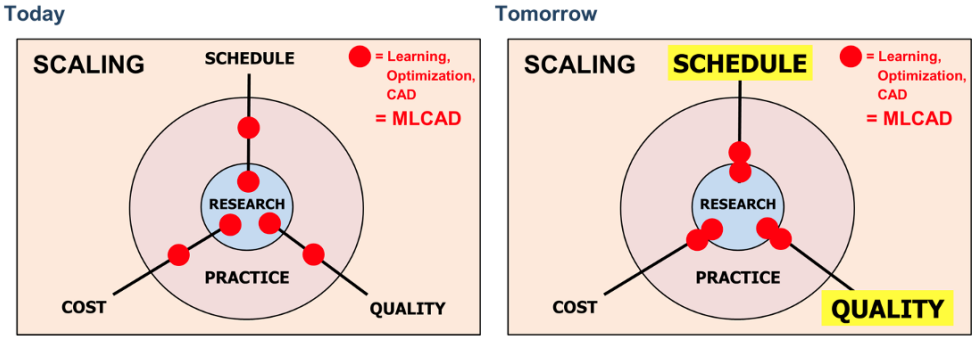

Figure 5. Efficiency Gains with Pervasive ML EDA [REF].

The gains extend beyond efficiency, and to (a) reliability with consistency in data, interface, test scripts and faster fraud detection, and (b) scalability with the ease of

scaling for larger team collaboration. As suggested below in Figure 6, we may well be at the *cusp* of setting forth on the long promise road rich with innovation in EDA.

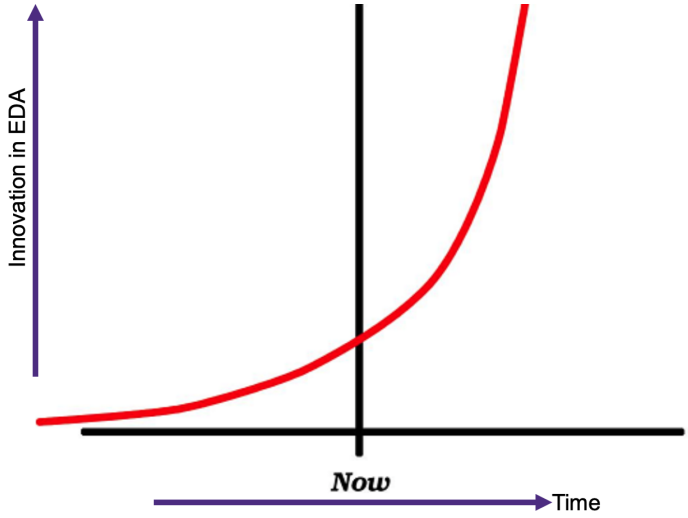

Figure 6. The promise of the ML EDA Flywheel with a shared infrastructure.

The promise may only be realized with the humming ML EDA flywheel and the road traveled *together* with a shared infrastructure.

Section 3: Major Challenges to the Creation and Maintenance of Shared Infrastructure in ML EDA, and Potential Solution Space

The journey of a thousand miles begins with one step.

- Lao Tzu

As claimed in Chapter 2, 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. Here, we extend this to also include support (clarifications, bug fixes, enhancements), incentives, and mandates.

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. Shareable means *open* as in more permissive in order to enable an uptick in productivity. In contrast, not shareable would mean creating and growing *proxies*. Challenges may be in the form of *bars* and *barriers* that must be overcome. Bars are thresholds like critical mass of functionality, critical quality, baseline volume of user base, number of silicon proofs, etc. Barriers are blockers e.g. an IP that is under proprietary control, license agreements that prevent researchers from openly sharing work, etc.

Next, we will list each major challenge accompanied by its potential solution space.

Challenge #0 (Barrier): 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 academic, 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 example, industry support for student internships and assured assimilation upon graduation). The incentive model for EDA companies may involve a

clear path of import of new research insights and collateral (models, applications). The support of Government agencies will be vital in enabling such a model.

Challenge #1 (Barrier): Access and Permissions

Challenge: EDA is data rich (diversity and magnitude). Chip design and EDA companies generate nightly runs *en masse* ranging from hundreds to hundreds of thousands. But there is a non-disclosure agreement attached (*proverbially*) to each of them to prevent it from being shared with the community at large. Designs are large and diverse, but again they remain under proprietary control as company-owned intellectual property. Process Design Kits (PDK's) are proprietary too with foundry providers such as TSMC, Intel, Global Foundries. Tool reports are plentiful and diverse, but all of it generated with commercial EDA tools remain largely locked or under license/EULA restrictions as also illustrated in Figure 7.

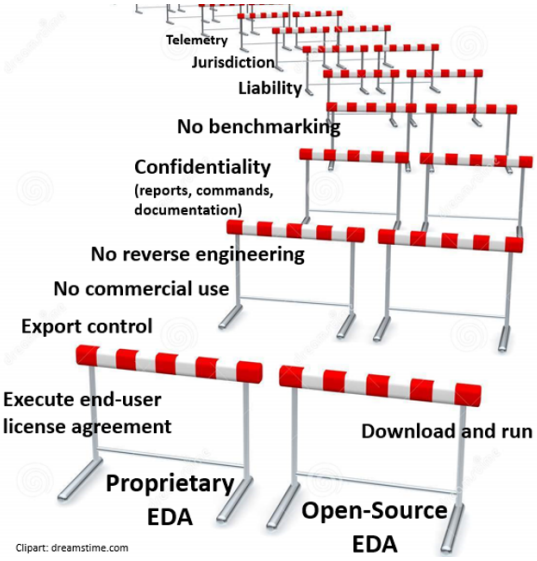

Figure 7. A Pictorial View of Traditional EDA Barriers [REF]

Overall, availability of large, diverse data, IP, and supporting infrastructure belonging to chip design and verification is either locked away or disallowed from being shared. The process of investigation for 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: Ultimately, sharable means open – as in open source, where more permissive usually means more positive impacts. Therefore, any non-sharable components of a shared infrastructure must be represented by proxies. For instance, proxy PDKs and enablements (e.g. ASAP7), proxy EDA tools (e.g. OpenRoad constituents) that we can benchmark and record data from, relevant proxy testcases

and design drivers (e.g. RISC-V, NVDLA, OpenPiton, Chipyard, etc.) represent potentially 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.

Challenge #2 (Barrier): Model Training

Challenge: Creation and maintenance of large, diverse, and sophisticated trained models are key to the success of shared infrastructure. But this requires data, and lots of it which itself suffers from the challenges described in Challenge #1 above. EDA vendors report that even the models that only have weights from training on company 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 [REF] related to the generation of ML datasets for digital and analog EDA with GAN's and obfuscation may be sought in open-source adaptation of trained models that would otherwise not be allowed to be shared. Perhaps the use of generative AI [REF] could be investigated as well in this regard.

Challenge #3 (Barrier): Continuous Improvement

Challenge: Improvements in ML for EDA really need to be tightly coupled with tool improvements. ML can find ways in which the tool can set better defaults, as ably demonstrated with OpenROAD's router [REF], and in general (for example) where 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. So it may be claimed that black-box EDA tools can – by design, for business reasons – be barriers to ML EDA.

Solution Space: The recourse here is rather narrow and that is for the EDA community to come together in 'picking one horse' and enabling it to achieve 'industry strength' from which point it can be possible to trigger the regenerative loop as shown in Figure 8.

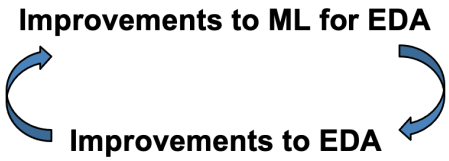

Figure 8. Regenerative Loop for continuous improvement

There is reasonable and growing evidence [REF] that OpenRoad may be the 'right horse to bet on'.

Challenge #4 (Bar): 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 which applies to both at seed and maintenance over-time. Quality is measured both in results as well as in support, and significant investment must be made to ensure that. Academic, open-source tools and IP blocks typically have limited support and documentation, and lack of standardization. Therefore a successful adoption of ML techniques in EDA requires careful consideration of the interpretability and explainability of ML models. The other challenge is in the need for constant retraining of models to prevent 'model drift'. Design IP's and technology nodes (feature size, design rules, etc.) are advancing at a rapid pace that changes are significant between generations, requiring frequent and costly retraining, or otherwise face model drift.

Solution Space: The remedy is similar to the one suggested in response to challenge #1. 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. Proper incentives must be put in place to allow for budgeted and discretionary participation from industry and academia. Finally, the first version of any offering need not be perfect, but good enough to enable enthusiastic community participation to get the flywheel spinning.

Challenge #5 (Bar): Security

Challenge: Open-source supply chain security is a big, complex problem and ML EDA is not immune from this challenge. There is a well-founded concern of reverse engineering of model extraction in EDA. Further, open-source commit/contribution must be regulated via established policies.
Solution Space: Concurrent with the first step in establishing the open, shared infrastructure is the initiative to seek standardization. The hardware community is new to such a structure and would be well served to take inspiration and adopt best-known methods from longstanding, successful open source initiatives in the SW community.

Section 4: Desired and Achievable Scope for the Shared Infrastructure in ML-EDA

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

- Serenity Prayer

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

In the big picture, the shared infrastructure need not be differentiating and simply needs to exist with a high bar for quality (initial and sustained) in robustness, usability, and extensibility. The data models, database, and features like standards support, PDK support, logging, scripting, GUI, etc. should be like plumbing and utilities. In that respect, the notion "*don't need to think about it*" is the appropriate bar to aim for, thereby keeping the focus on innovation with algorithms and models. Tools need not beat commercial versions but they should be good enough to verify effectiveness of new ML techniques for enabling proof of concept (POC).

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 like GNL (GenerateNetlist) and GANN's or generative AI may be employed to generate highly obfuscated synthetic cases of industry relevance that use in-house design benchmark suites with various directives as labels. Goodness of synthetic testcases must be assessed and established. In physical design (placed DB), for example, the utilization of cells, clock domains, congestion, blockage constraints, region constraints etc. must be considered to fit both 'hardness' and 'realness' of a typical industry relevance test case. In functional verification, testcases/datasets for coverage closure may include stimulus and cover groups/points, whereas testcases/datasets for debug may manipulate assertions with bug injection and error measurement.

The format of datasets may conform to OpenDB or any other standard form of sharable DB. Significant time must be invested in preparation of datasets with downstream use in

mind. For instance, conversion from EDA tool format which is TCL friendly to what is consumed in the PyTorch environment will be greatly beneficial to data engineer/data scientist productivity by lowering the barrier to entry.

In analog design, IP's need to cover more types of analog circuits and take advantage of open PDKs like SKY130, 90, GF180. Training data and infrastructure to generate them is a bit easier to come by in the analog domain as there is a higher volume of such activity in academia. There are also plethora of synthetic data in the analog domain (e.g., UT-AnLay for analog performance prediction for OTA). Availability of open-source software like ALIGN, MAGICAL, BAG, FASOC etc. is a boon but getting them to run requires expert knowledge and closer participation from analog designers (human in the loop).

Ideally, shared infrastructure must address issues that are common and repetitive for the design community (for example, the process of transferring data from design/reports to ML frameworks is time consuming and error prone).

Finally, it may be instructive to inquire about the EU's TRISTAN (*Together for RISc-V Technology and ApplicatioNs*) project to share key learnings for scope given the parallels. TRISTAN is a 3-year project aimed at expanding and industrializing the European RISC-V ecosystem to compete with existing commercial alternatives. The project will define a European strategy for RISC-V based designs, create a repository of industrial quality building blocks, and cover 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, increase security, increase transparency, allow better interoperability, reduce cost to companies and consumers, and avoid vendor lock-ins.

The shared infrastructure must remain sustainable and extendable in the long run. Sustenance may be enabled as long as the shared infrastructure and its constituents continue to scale and provide value to stakeholders with quality, customer support, and omnipresent recognition that keep stakeholders engaged. The shared infrastructure must not be rigid or brittle in order to maintain extensibility.

Section 5: Recommendations to the NSF

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 to the National Science Foundation (NSF):

Open-Source ecosystem learnings from AI, SW and Healthcare

NSF should commission a joint industry-academia workshop on debating the learnings of 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 AI and ML is a testimony to this trend (example: Landing.ai). The software and healthcare communities have facilitated ecosystems for strong, collaborative, concurrent research in academia whether it be from creative and sustained incentives, or need created from crisis (e.g. Operation Warp Speed in search of a vaccine to fight the COVID-19 pandemic). EDA community involves key players in EDA Vendors and Chip Design Companies where the incentive model and urgency for shared infrastructure is absent or invisible. A workshop with the goal to understand the similarities and differences of the EDA ecosystem with the ones that sustain the AI, SW, and Healthcare industry, and then crystalize the incentive model for the EDA ecosystem will be timely and fruitful.

Warming up the ML EDA Flywheel

NSF should play a coordinating and enabling role in bringing academia, EDA, and chip design companies with the goal to remove the knots that have stymied the ML EDA flywheel from spinning. The appropriate levers of incentives and strategic alignment should be pulled to overcome the bars and barriers suggested in Section 3. Dr. Eric

Schmidt, CEO of Steel Perlot who also attended the workshop 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 few grad students so we need industry, government, academia partnerships to enable this.*"

Accelerating and Sustaining the ML EDA Flywheel

NSF should consider funding a larger project(s) on a 10-year scale with clear intermediate milestones, with the aim of encouraging multidisciplinary collaboration and continuity in support of the ML EDA flywheel. The NSF should create new funding mechanisms to support high-risk system-level demonstrations with end-to-end expertise and significant funding, incentivizing prototypes and collaborations between industry and academia. Finally, longer-term support for the shared infrastructure should be explored along with an effective governance model to ensure its sustainability and inclusivity.

References [INCOMPLETE - WIP]

1. M. Rapp, H. Amrouch, Y. Lin, B. Yu, D. Z. Pan, M. Wolf, 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, October 2022.
2. Z. Xie, X. Xu, M. Walker, J. Knebel, K. Palaniswamy, N. Hebert, J. Hu, H. Yang, Y. Chen and S. Das, “APOLLO: An Automated Power Modeling Framework for Runtime Power Introspection in High-Volume Commercial Microprocessors,” *IEEE/ACM International Symposium on Microarchitecture*, 2021.
3. 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,” *IEEE/ACM International Conference on Computer-Aided Design*, 2021.
4. C.-C. Chang, J. Pan, T. Zhang, Z. Xie, J. Hu, W. Qi, C.-W. Lin, R. Liang, J. Mitra, E. Fallon and Y. Chen, “Automatic Routability Predictor Development Using Neural Architecture Search,” *IEEE/ACM International Conference on Computer-Aided Design*, 2021.
5. Z. Xie, R. Liang, X. Xu, J. Hu, C.-C. Chang, J. Pan and Y. Chen, “Pre-Placement Net Length and Timing Estimation by Customized Graph Neural Network”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 41, No. 11, pp. 4667- 4680, November 2022.
6. G. Saumil, A. Tyagi, M. Quinn and J. Hu, “Transaction Level Stimulus Optimization in Functional Verification Using Machine Learning Predictors,” *IEEE International Symposium on Quality Electronic Design*, 2022.
7. J. Pan, C.-C. Chang, Z. Xie, A. Li, M. Tang, T. Zhang, J. Hu and Y. Chen, “Towards Collaborative Intelligence: Routability Estimation based on Decentralized Private Data,” *ACM/IEEE Design Automation Conference*, 2022.
8. R. Liang, H. Xiang, J. Jung, J. Hu and G.-J. Nam, “A Stochastic Approach to Handle Non-Determinism in Deep Learning-Based Design Rule Violation Predictions,” *IEEE/ACM International Conference on Computer-Aided Design*, 2022.
9. P. Sengupta, A. Tyagi, Y. Chen and J. Hu, “How Good is Your Verilog RTL Code? A Quick Answer from Machine Learning,” *IEEE/ACM International Conference on Computer-Aided Design*, 2022.
10. J. Pan, C.-C. Chang, Z. Xie, J. Hu and Y. Chen, “Robustify ML-based Lithography Hotspot Detectors,” *IEEE/ACM International Conference on Computer-Aided Design*, 2022.

APPENDIX A1: NSF Workshop on Shared Infrastructure for Machine Learning EDA

Workshop date: Mar 10, 2023, Friday, 10:00 a.m. – 5:00 p.m. CST

***In person attendance location*:** 4-178A Keller Hall, 200 Union Street SE, Minneapolis

Virtual attendance*:* Via Zoom

9:30 – 10:00 a.m.	Breakfast
Session 1	*Chairs*: Sachin Sapatnekar (UMN) and Mike Quinn (TAMU)
10:00 – 10:10 a.m.	Welcome, Dean Andrew Alleyne, College of Science and Engineering, UMN
10:10 – 10:15 a.m.	Opening, Sankar Basu (NSF)
10:15 – 10:35 a.m.	Andrew Kahng (UCSD) "Bars and Barriers to Overcome for Shared ML EDA Instructure"
10:35 – 10:55 a.m.	Ruchir Puri (IBM) "Engineering the Flywheel of AI for Electronic Design Automation: Present Challenges and Future Opportunities"
10:55 – 11:15 a.m.	Thomas Andersen (Synopsys) "AI for chip design - An industry perspective"
11:15 – 11:35 a.m.	Dan Yu (Siemens EDA) "ML for Data-Driven Verification"
11:35 – 11:55 a.m.	Siddharth Garg (NYU) "Towards Large, High Quality and Open Datasets for ML4EDA"

12:00 p.m. – 12:30 p.m. **Break**

Session 2	*Chair*: Jiang Hu (TAMU)
12:30 – 12:50 p.m.	Sachin Sapatnekar (UMN) "Generating ML datasets for digital and analog EDA: Opportunities and challenges"
12:50 – 1:10 p.m.	Mark Ren (Nvidia) "Enabling Generative AI and GPU Acceleration for EDA"
1:10 – 1:30 p.m.	Scot Weber (AMD) "Practical considerations for scaling AI/ML in an EDA context"

1:30 – 2:30 p.m. **Breakout session**

1. Data: raw data or scripts? format, scope & pitfalls (S. Garg, T.-W. Huang) **4-131 Keller**
2. Software interface between ML/EDA tools: Scope/pitfalls (V. Chhabria, M. Robbins) **5-120 Keller**
3. Open source environment and platform extensibility (T. Ansell, C. Yu) **4-178A Keller**
4. Testcases, benchmark and validation systems (M. Quinn, I. Bustany) **4-146 Keller**
5. Collaboration between industry and academia (Y. Chen, C. Alpert) **4-178B Keller**
6. Analog design automation (D. Pan, J. Hu) **3-166 Keller**

2:30 – 2:45 p.m.	**Break**
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 Moderator: Ismail Bustany (AMD) Panelists: Srinivas Bodapati (Intel), Joe Jiang (Google), Sung-Kyu Lim (DARPA), Marcus Pan (SRC), Matt Robbins (Steel Perlot)

Dinner: 6pm, Tea House

APPENDIX A2: List of Registered Attendees [INCOMPLETE]

Name:	Affiliation:
Jiang Hu	Texas A&M University
Marilyn Wolf	University of Nebraska Lincoln
Cunxi Yu	University of Utah
Yiorgos Makris	UT Dallas
Tim Ansell	Google
David Pan	University of Texas at Austin
Tsung-Wei Huang	University of Utah
Peng Li	UCSB
Harry Foster	Siemens EDA
Paul Franzon	NC State University
Srinivas Bodapati	Intel
Norman Chang	Ansys
Ioannis Savidis	Drexel University
Borivoje Nikolic	University of California, Berkeley
Zhiru Zhang	Cornell University
Yanqing Zhang	NVIDIA
Jinwook Jung	IBM Research
Jana Doppa	Washington State University
Partha Pande	Washington State University
Siddharth Garg	NYU
Akhilesh Kumar	Ansys
Luca Carloni	Columbia University
Kumar, Vaibhav	NXP semiconductors
Matthew Robbins	Steel Perlot
Li-C. Wang	University of CA Santa Barbara
Ming Zhang	Self Employed
Robert Mains	CHIPS Alliance / Linux Foundation
Lizy K. John	UT Austin
Song Han	MIT
Elyse Rosenbaum	Univ. Illinois Urbana-Champaign
Ethan MAhintorabi	Google
ramesh karri	nyu

Xiaoqing Xu	X Development LLC
Sheldon Tan	UC Riverside
Siddhartha Nath	Intel Corp
Rongjian Liang	NVIDIA
Ben Beaumont	Cadence Design Systems
Matthew Ziegler	IBM
Charles Alpert	Cadence Design Systems
Frank Liu	Oak Ridge National Lab
Sanmitra Banerjee	NVIDIA Corporation
Shobha Vasudevan	Google
Brucek Khailany	NVIDIA
Abdelrahman Hosny	Brown University
Michelle Ritter	Steel Perlot
Eugene John	University of Texas at San Antonio
Mehdi Saligane	University of Michigan
Savithri Sundareswaran	NXP Semiconductors