



SIGGRAPH 2025
Vancouver+ 10-14 August

KHRONOS[®]
GROUP

Accelerating AI with Open Standards and Ecosystem Insights

August 2025

Markus Tavenrath
ML Council Chair, NVIDIA

David Kanter
Real World Insights
“Mr. MLPerf”



Outline for Today's BoF

- **Motivation & Purpose**

The Khronos AI Ecosystem Research Project

- **Exploring the Problem**

Why is the AI Acceleration ecosystem so complex?

- **Technical Understanding**

Why is AI Acceleration performance portability hard?

- **Solution Exploration**

How can open standards help?

- **Open Discussion**

We welcome your thoughts and feedback!





Introduction

Khronos and its ML acceleration APIs

Khronos ML Council Group Activities

- Discussing challenges in cross-platform ML development
- Promoting exchange of knowledge in the ML-domain
- Building industry consensus on effective ML acceleration directions
- Input to the evolution of existing API standards and extensions
- Providing feedback to Khronos members and API implementers
- Conducting market research!



Machine Learning Acceleration Complexity

Open-Source Frameworks

Compilers, Runtimes and Libraries

Acceleration APIs

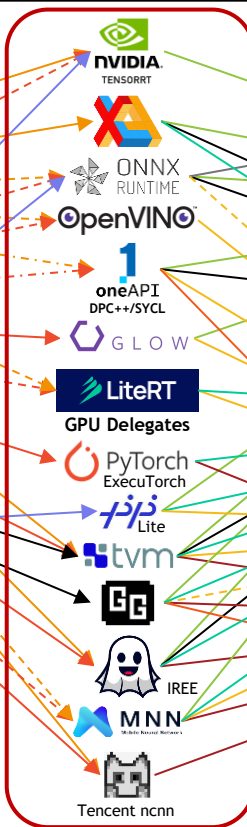
- Direct Integration
- - - - - Custom Kernels
- - - - - File Formats

TensorFlow

PyTorch

LLaMA

PaddlePaddle
Baidu



DirectML

NVIDIA
CUDA

AMD
ROCm



OpenCL

Vulkan

SYCL

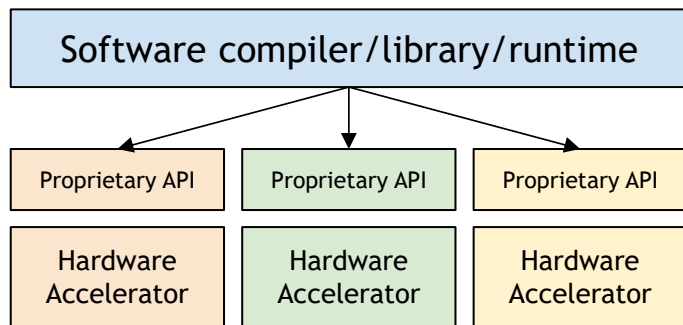
Proprietary
APIs

Open
standard
APIs

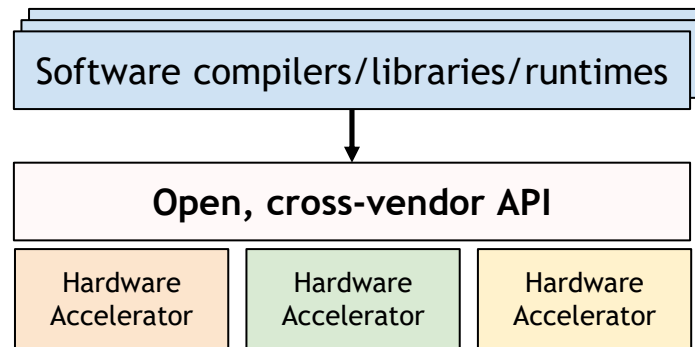
The complex ML landscape of compiler,
runtimes and libraries results from the
search for acceleration flexibility,
customization, and optimization

Can Open Interoperability Standards Help?

- **Open cross-vendor APIs benefit software AND hardware vendors**
 - Software vendors have reduced costs and broader market reach
 - Hardware vendors have access to a broader software ecosystem
 - Robust API design and support with input and investment from multiple companies
- **BUT acceleration standards must be designed with care**
 - Not too soon - only standardize interoperability that is **proven** and **stable**
 - **Extensible** to accommodate technology advances



Custom API per hardware vendor
increases software porting and support costs



Cross-vendor API streamlines software
development and widens market reach

Khronos ML Open Standards Activity



NNEF - Portable ML model format and operator set

- Shape inference and runtime computation, including dynamic control flow
- **NEW** SkriptND to define custom operators to build models
- NNEF 2.0 operators' computation formulas can be lowered into tensor languages



Slang - modern shading language and open-source compiler

- Type reflection, specialization, and auto differentiation - perfect for 'Neural Graphics'



SPIR-V - hardware-agnostic GPU IR compiler target

- Intrinsic maps matrix ops efficiently onto Tensor Cores
- **NEW** extended graph - tensor dataflow with TOSA operators



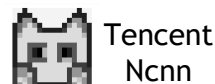
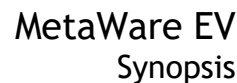
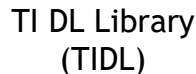
Vulkan and OpenCL - cross-platform execution APIs

- Portable management of shared memory, synchronization, tiling
- **NEW** expanding Tensor intrinsics, e.g., cooperative matrices
 - **NEW** ML datatypes like bfloat16, fp8, int8, ...



Aim for single 'HAL' IR and API for hardware vendors to support a complete ML stack?

ML Adoption of Khronos Open Standards



Multiple ML runtimes and compilers ship today leveraging Khronos API backend acceleration

Khronos ML Research Project

Khronos APIs *are* supported by many ML runtimes and frameworks

But we think we can do more to streamline the ecosystem!

BUT we need real-world feedback to guide our next steps...

Are the current Khronos APIs providing effective acceleration *in practice*?

What are the emergent *effective* directions for the ML acceleration stack?

What can open standards do to provide *genuine* ecosystem benefit?

Can we finally crack functional **AND** performance portability?

So, Khronos has initiated the AI Ecosystem Research Project
to better understand market realities

Many thanks to Futurewei for sponsoring the project!





Motivation and Purpose

Of the Khronos AI Ecosystem Research Project

Real World Insights Team

[Victor Bittorf](#) AI and open-source experience at Google Brain and Meta PyTorch AI, focus on optimization across NVIDIA, AMD, and other accelerators (e.g., SiMa AI). Experience with PyTorch, TensorFlow, ONNX, AI hardware-software interactions, benchmarking (e.g., MLPerf and customer applications), product management, competitive analysis

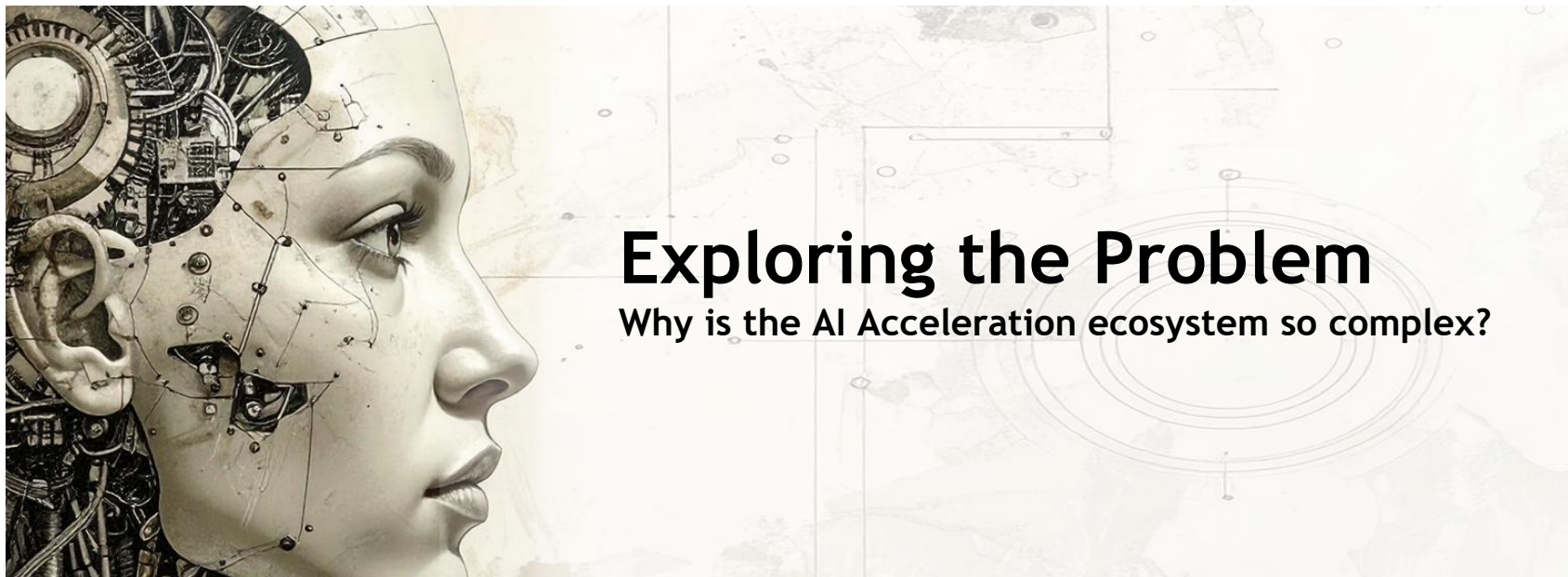
[Peter Glaskowsky](#) Previously a system architect at Esperanto Systems. Co-founded x86 startup Montalvo Systems, held senior roles at Intel, Microsoft, and Microprocessor Report. Co-inventor on 30 U.S. patents, authored two books on computer graphics, former chair of the Marketing Research committee for the RISC-V Foundation

[David Kanter](#) Co-founder, leading MLPerf benchmarks at MLCommons. Founded HW/SW co-design startup, drafted patents for Cerebras Systems, 16 years expertise in computer architecture, ML, market research with extensive consulting for Khronos Group members

Project Goals

Understand the hardware and software ecosystem for machine learning acceleration and the role and opportunity for standards bodies such as the Khronos Group to advance pervasive acceleration of machine learning





Exploring the Problem

Why is the AI Acceleration ecosystem so complex?

Who We've Talked To

Seek out a holistic view from 1-1 interviews with many experts and decision makers in the ecosystem

AI Model Experts - people who work with models every day

Product Builders - people whose products depend on AI

Hardware Makers - people who build and support new AI accelerators

AI Compilers & Tools - people make the compiler tool chain and ecosystem

Community & Standards - people at the biggest projects and orgs

Across these different groups, we sought out people from technical, product, and business roles to understand:

What problems does the industry face, and what solutions are viable?

The BIG Problem

*Launching AI products & features
requires too much manual expert work*

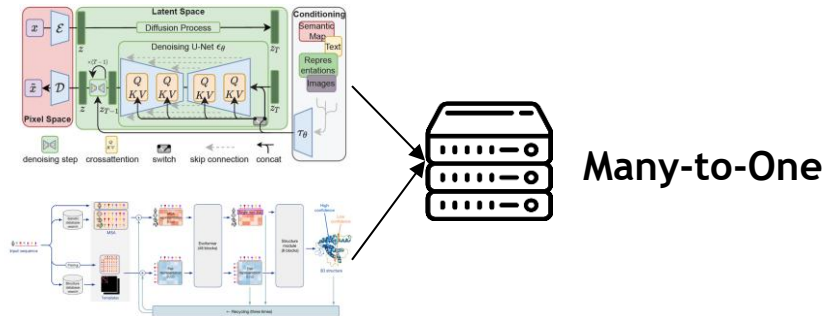
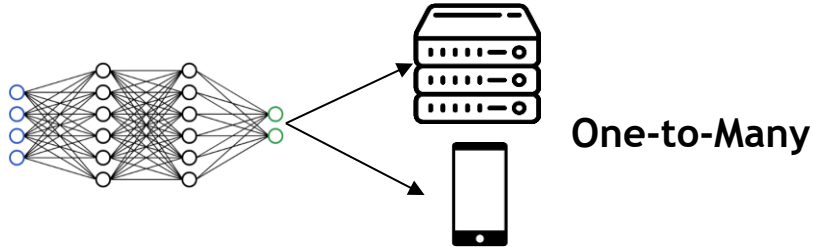
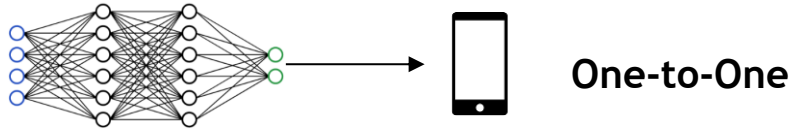
Based on discussions so far:

85% relates to manual expert work

15% relates to “other” (regulation, access to hardware, etc.)

Portability Problems

Most issues with running AI models fit into one of three use cases:



These problems are all solvable by expert manual work today

The real difficulty is no one has enough experts to to complete all the work they need done

Training v. Inference

Currently, we see inference as much more significant to our research than training

- **Training is reportedly less performance sensitive**
 - E.g. Reducing 28 days to 26 days is not a significant win for researchers
 - Most training efforts are exceptionally highly resourced already
- **Training is less likely to adopt a standard**
 - VERY fast moving target, speculative and experimental
 - More bespoke systems and narrow range of technology
- **Inference will be a larger market than training in the long term**
- **Inference has significantly more players and need for agreement**
 - More software stacks, hardware players; diverse power envelopes and systems scales

Speed of Light

Speed of light is the best possible performance of a model on a given system

Every inference value chain eventually looks at speed of light

If the person you're talking to doesn't think about speed of light, then they're delegating lower in the stack or delaying into the future

The Business of Speed of Light (SoL)

Competition on cost or KPIs drives the market towards SoL

Cost

"I want YOLOv3 @ 30FPS"

10% SoL w/ \$100 Chip
99% SoL w/ \$10 Chip

Same performance,
10x cost savings

KPIs

"I want new feature X"

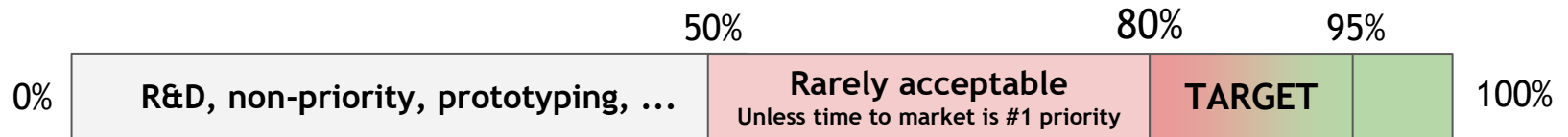
"I want my inference to be
better than competitors"

"I want to reduce power"

Over time, libraries approach
the speed of light through
competitive pressure.

Approaching the Inference Speed of Light

What % of Speed of Light is needed?



- **<50% is seen in R&D and prototyping use cases**
 - Note: some view portability, even <50% as a win!
- **50%-80% is rarely viewed as acceptable**
 - May be enough when time to market is the #1 priority
- **80% is most often cited as a minimum**
- **>95% is viewed as very difficult to achieve even by the best practitioners**

Why Is It Hard to Reach the Speed of Light?

- **Pre & Post Processing**

- Non-AI ops (e.g., image scaling) are sometimes >50% of task speed of light

- **Data Movement**

- Planning and managing data movement (especially to and from memory, or network) is a major challenge, often done by experts

- **Graph Rewriting for an Architecture**

- Each architecture has preferences for how to do operations; rewrites often needed
- E.g.: $\text{CONV}(\text{input}, \text{weights}) \Leftrightarrow \text{GEMM}(\text{im2col}(\text{input}), \text{weights})$

- **Custom Operators**

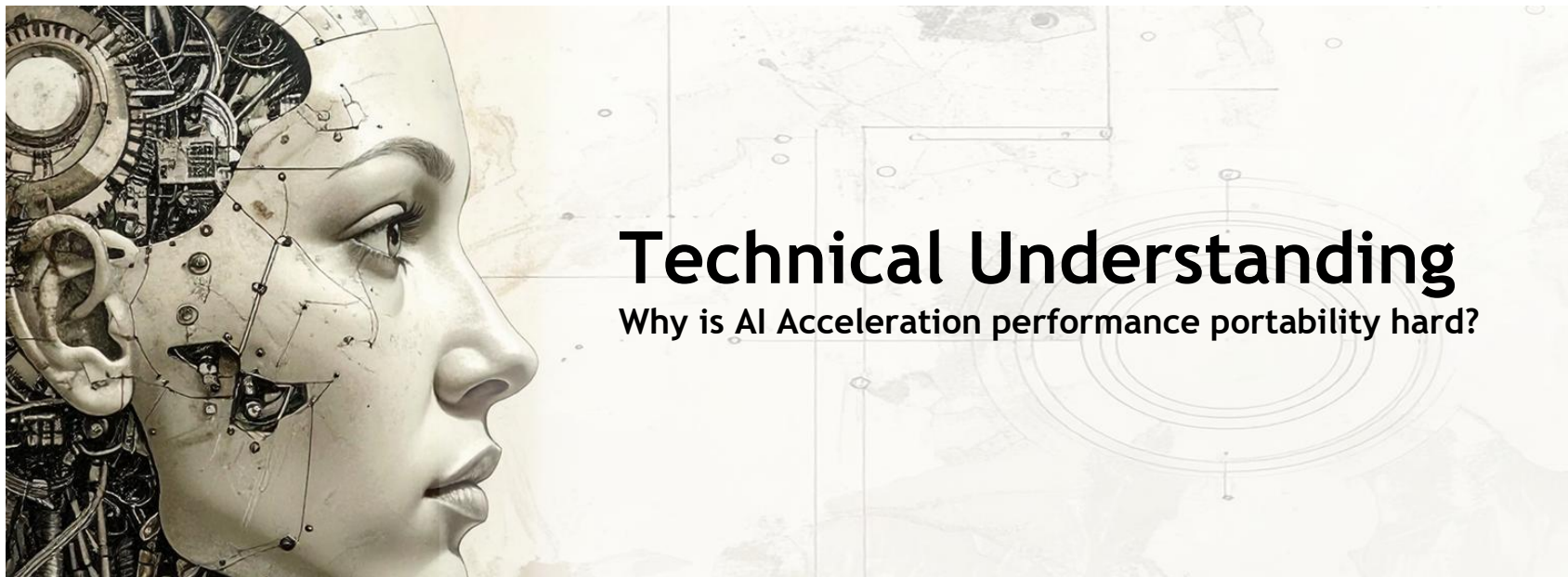
- Specialized logic specific to a model (e.g., new attention variant) or platform
- Compiler and systems support can take a year, if at all

- **Quality**

- Quality (e.g., accuracy of object detection) determines capabilities; difficult to quantify
- Many optimizations (e.g., reduced precision, layer fusion) can impact quality

Problem Restatement

**How to achieve portability for AI models,
in the face of the desire for speed-of-light,
without demanding significant expert work?**



Technical Understanding

Why is AI Acceleration performance portability hard?

What Does “Performance Portability” Mean?

Let's consider the popular YOLO computer vision models

Take this model trained on 1 computer, and run it on a different system

Let's consider a single common feature of such models which are Activation Functions and Convolutions:

ReLU + CONV

While there are many more layers, let's consider just this simplified slice:

Suppose you have an ONNX graph which contains ReLU + CONV

Same Math, Radically Different Performance

- **Do you have a fused conv-relu op?**
 - $A = \text{FusedConvReLU}(X, K, b)$
- **Do you have Relu & Conv?**
 - $A = \text{ReLU}(\text{Conv}(X, K, b))$
- **Do you have a Relu?**
 - $A = \text{ReLU}(K * X + b)$
- **Just Basic Linear Algebra?**
 - $A = \max(0, KX + b)$
- **Do you need to use a GEMM?**
 - $A = \max(0, \text{reshape}(\text{GEMM}(\text{unroll}(K), \text{im2col}(X)) + b))$

**An Expert Human or
Compiler Needs to Choose**

Custom Operators

On new popular models...

20-30% of ops may be custom operators

Meaning the humans or compilers from the previous slide will not have seen them before, requiring manual implementation

Inference operators are typically inherited from training R&D

Triton Operator Implementations

OpenAI Triton (and other frameworks) can be used to write operators:

Operator performance is not portable

Even if the operator is functionally portable, operator performance is not portable

Writing every op in something like Triton may help with functional porting but will not solve for performance portability

Assignment & Scheduling

"How do you assign and schedule ops on one or more accelerators"

Given a specific set of ops, the optimal assignment and schedule onto hardware is specific to the system

There are general principles to use...but the process requires detailed knowledge of the memory, network, accelerators, and CPUs

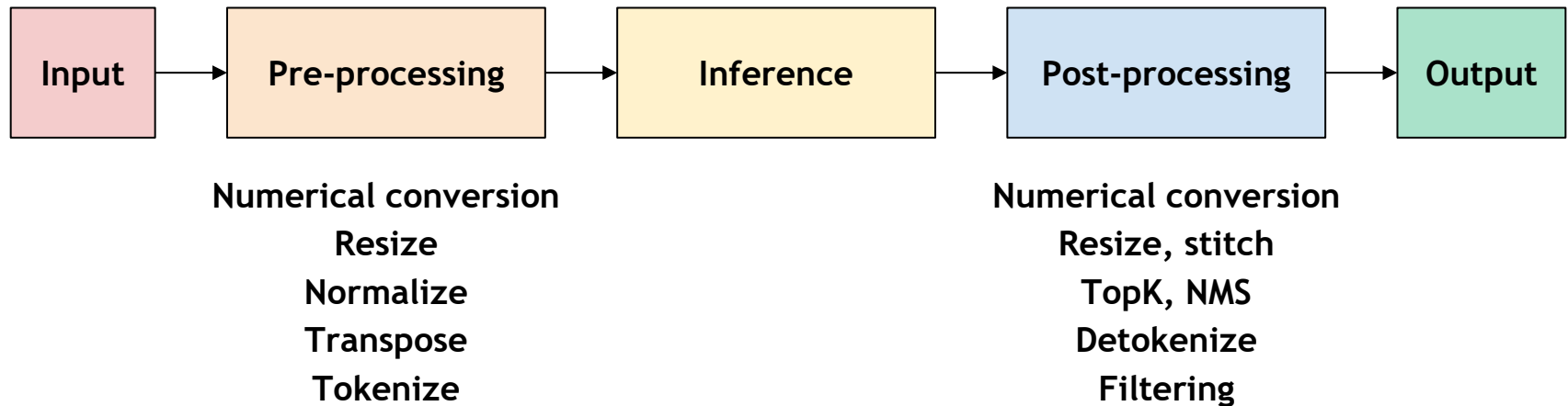
**An Expert Human or
Compiler Needs to Choose**

Many Hardware Makers = Many Special Cases

Graph -> Op Choice -> Scheduling

Because operator selection and scheduling is so specific to the system hardware, system expertise resides only at the hardware vendor

Pre- and Post-Processing



Model- (e.g., tokenize) and platform-specific (e.g., transpose) steps

Some existing libraries, e.g., OpenCV for vision, but many types of data/model

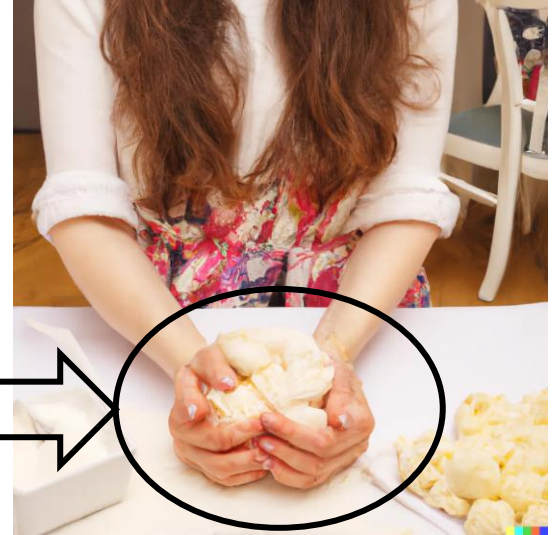
Often performed on host processor → **opportunity to advance the field**

Quality

As in graphics, AI accuracy is not exact
Accuracy determines capabilities

Prompt:
Kneading dough, DALL-
E 2

Fast and wrong is still wrong :)



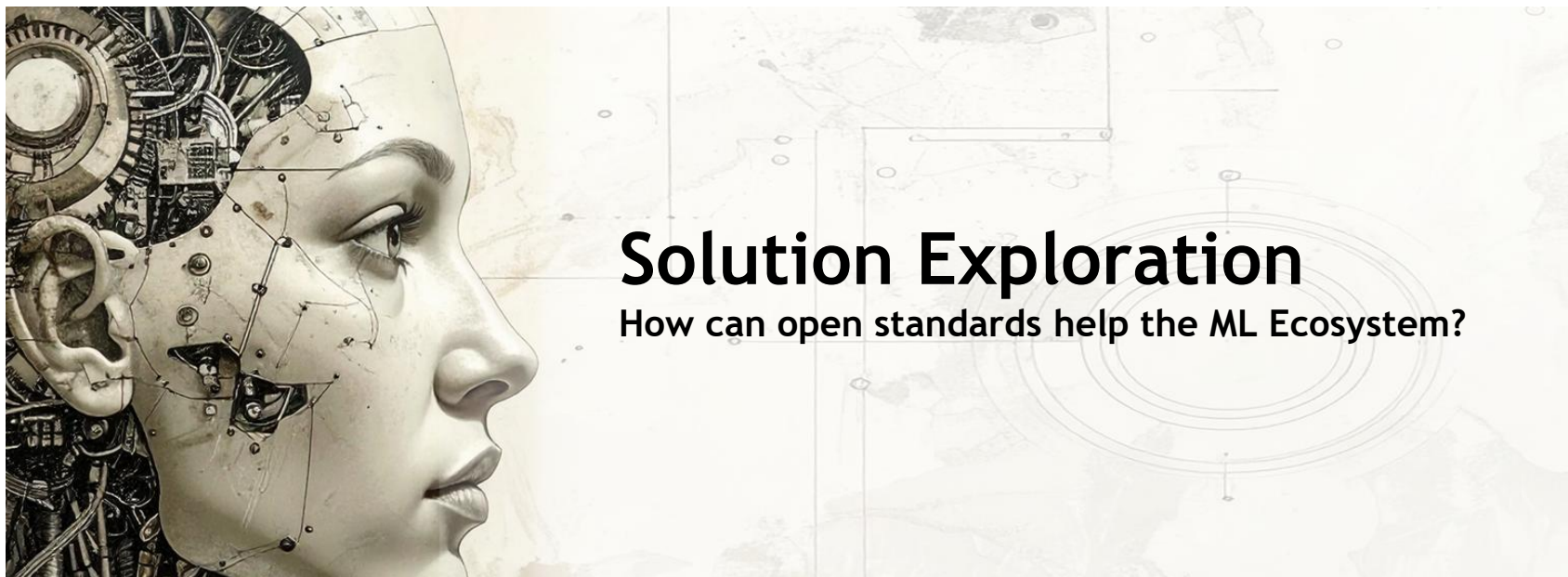
Evaluating correctness is exceptionally difficult and expensive for application developers... acceleration mustn't make that more difficult

Market Research Input for New ML Standards

- **Don't become a time capsule**
 - AI moves very quickly - need to adapt and adopt new technology in weeks, not years
- **Large operator sets that are too hard to support will struggle for support**
 - Too much effort to onboard new systems
- **Do not union functionality of many technologies**
 - Specifically called out as non-workable as too broad
- **Be clear and opinionated**
 - There are already too many ways to do things

The next ML standards need to reduce confusion, not add to it

Productive standardization can help the industry converge on market-enabling solutions

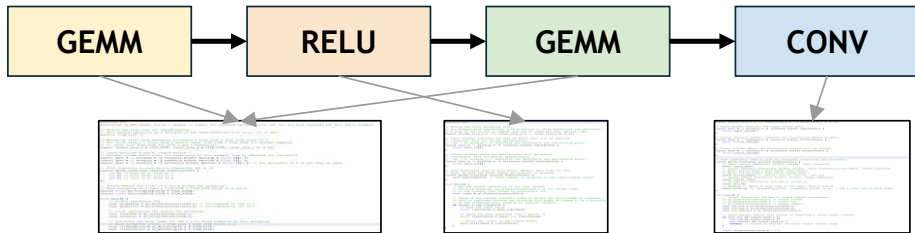


Solution Exploration

How can open standards help the ML Ecosystem?

Compute Shaders

- Compute is widely supported in Khronos APIs: Vulkan, OpenCL and SYCL
- Vulkan is now widely supported by GGML, TVM, IREE, ExecuTorch, ...
- Vulkan has a series of extensions to implement ML operations
 - VK_KHR_cooperative_matrix | VK_NV_cooperative_matrix2
 - VK_KHR_shader_integer_dot_product | VK_NV_cooperative_vector
- And to support a variety of data types
 - VK_KHR_shader_float16_int8 | VK_KHR_shader_bfloat16 | VK_EXT_shader_float8
- BUT developers need to do a LOT of work!
 - For each framework M, each operation must be implemented
 - For each hardware platform N, each operation must be optimized
 - For each IHV K, one more more platforms can be released each year
 - Worst case $M*N*K$ permutations of the same operation have to be updated in the ML ecosystem every year

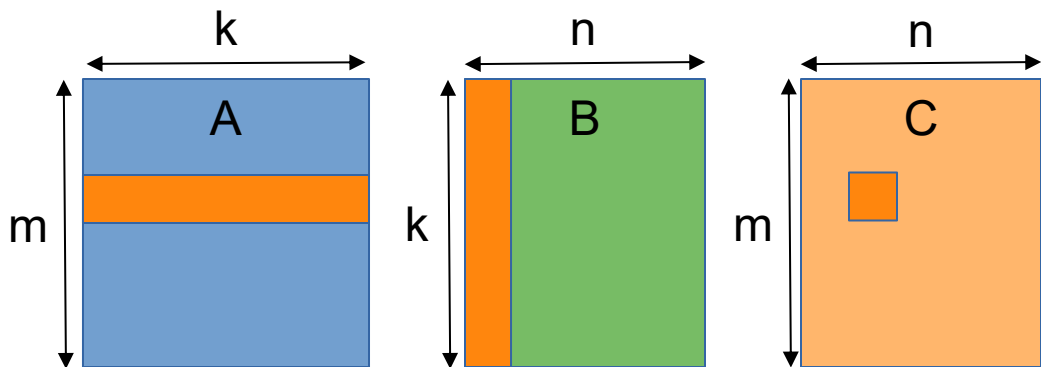


Optimal performance depends on

- Register Limits
- Shared Memory Limits
- Number of Cores
- Threads per core

...

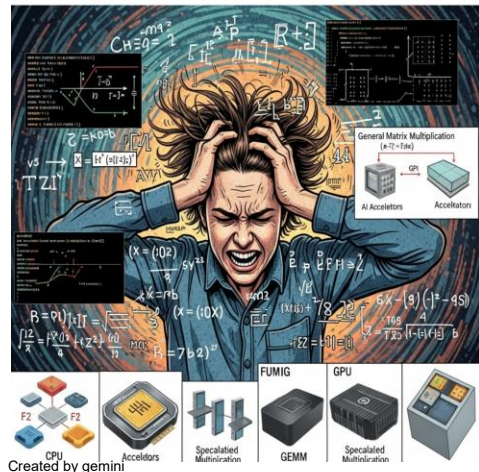
What is GEMM and why it is hard?



$$C = A \times B$$

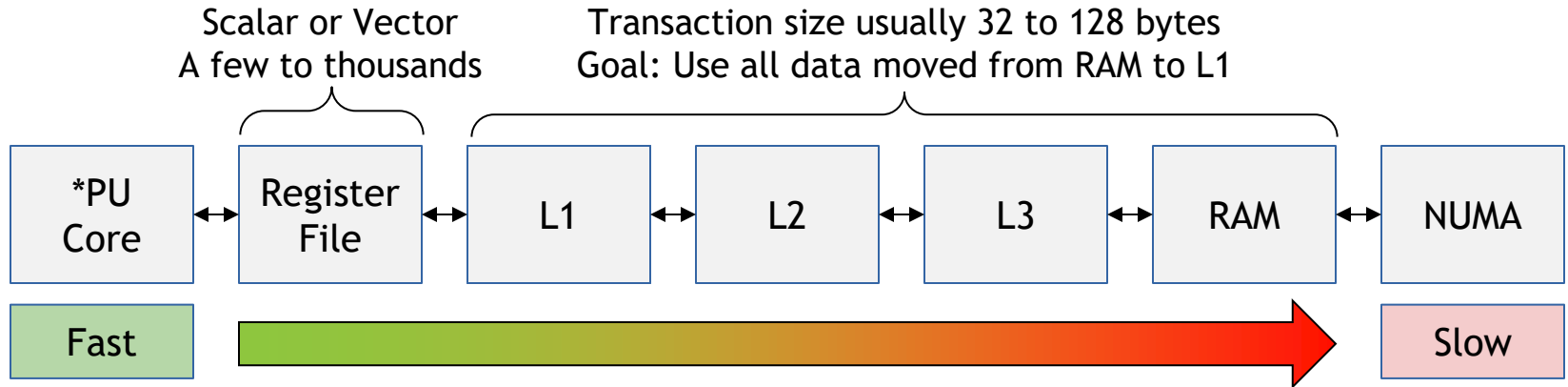
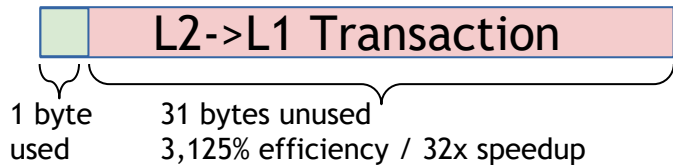
- GEMM stands for GEneral Matrix Multiply
- $C_{i,j} = \text{dot}(A_{i,*}, B_{*,j})$
- Complexity
 - Math ops: $m*n*k$ multiply and add
 - Memory ops best: $m*k + n*k$ reads, $m*n$ writes
 - Memory ops worst $(m*n) * k$ reads, $m*n$ writes
- Optimize memory access pattern to optimize GEMM

The never-ending quest to optimize GEMM for every possible hardware

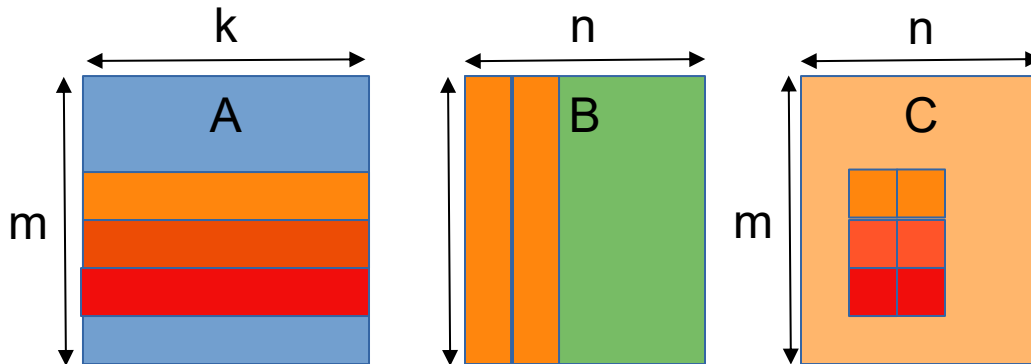


Memory Hierarchy

- Memory architectures are vastly different across different GPU architectures
 - It's impossible to write a single shader optimal for all architectures
- Goal: Minimize transactions and maximize usage of transaction data
 - E.g. do not read 32 bytes from RAM to L1 and use only 1 byte



Matrix Memory Layout



- **Multiple possible matrix layouts**

- Row-major: Elements of a row are contiguous in memory
- Column-Major: Elements of a column are contiguous in memory
- Blocked: $m \times n$ tiles are stored in implementation dependent order

- **Minimize number of memory transactions**

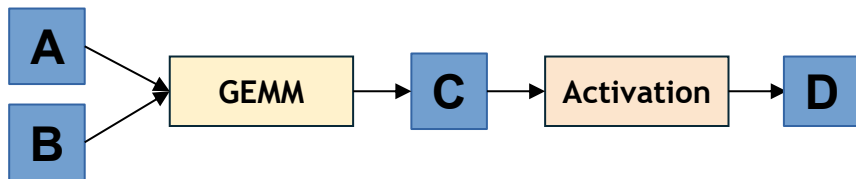
- For row-major ensure that multiple column are computed at once within a workgroup
- For col-major use vectorized loads
- How to handle random swizzled memory layouts?

Implementation Strategies

- **Min Power vs. Max Perf implementation strategy might differ**
 - How to handle attached to power vs. running on battery
- **How to efficiently assign and schedule work across multiple cores?**
 - Use all cores for lowest latency?
 - Run on subset of cores to enable background task?
 - Do partially redundant work to make use of more cores?
- **More complex for heterogeneous platforms!**

Meta-command-like ML Operation HAL?

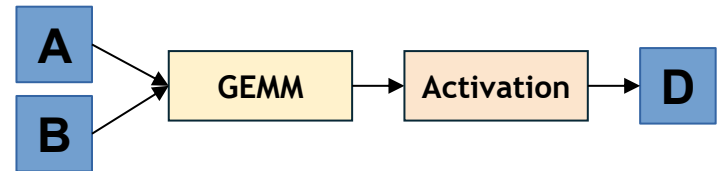
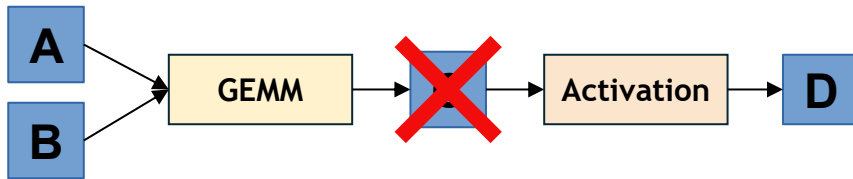
- Should we expose ML ops through a HAL?
- HAL knows platform best → big win?
- Naive approach: Each op is independent
 - GEMM requires A and B as input
 - GEMM writes C as output
- What is the best layout for A, B, and C?
- How to handle Fusion to save Memory Bandwidth
 - C has to be written even if it's just a temporary
 - Do not want to have combinatorial explosion of op combinations



Activation:
 $D_{i,j} = f(C_{i,j})$
E.g. ReLU

Graph HAL

- Pass full graph to the driver
 - Only the graph inputs & outputs have fixed layout
 - All intermediates are controlled by HAL
- HAL can now happily optimize full graph!
- What happens if an op is not supported by a graph?
 - E.g., new custom operator
- What happens if data type is not supported?
 - How to specify decoding?



Compute HAL vs Graph HAL

Compute HAL

Library / App Developers

High

Not Portable & Future Proof

No

Developer Work

Yes

Yes

Feature / Usage

Expertise Required

Performance

NPU support

Optimization / Fusion
New Operator support

Predictable
Performance

Custom Operator Set

Graph HAL

IHV / Driver Developers

Medium to Low

Portable & Future Proof

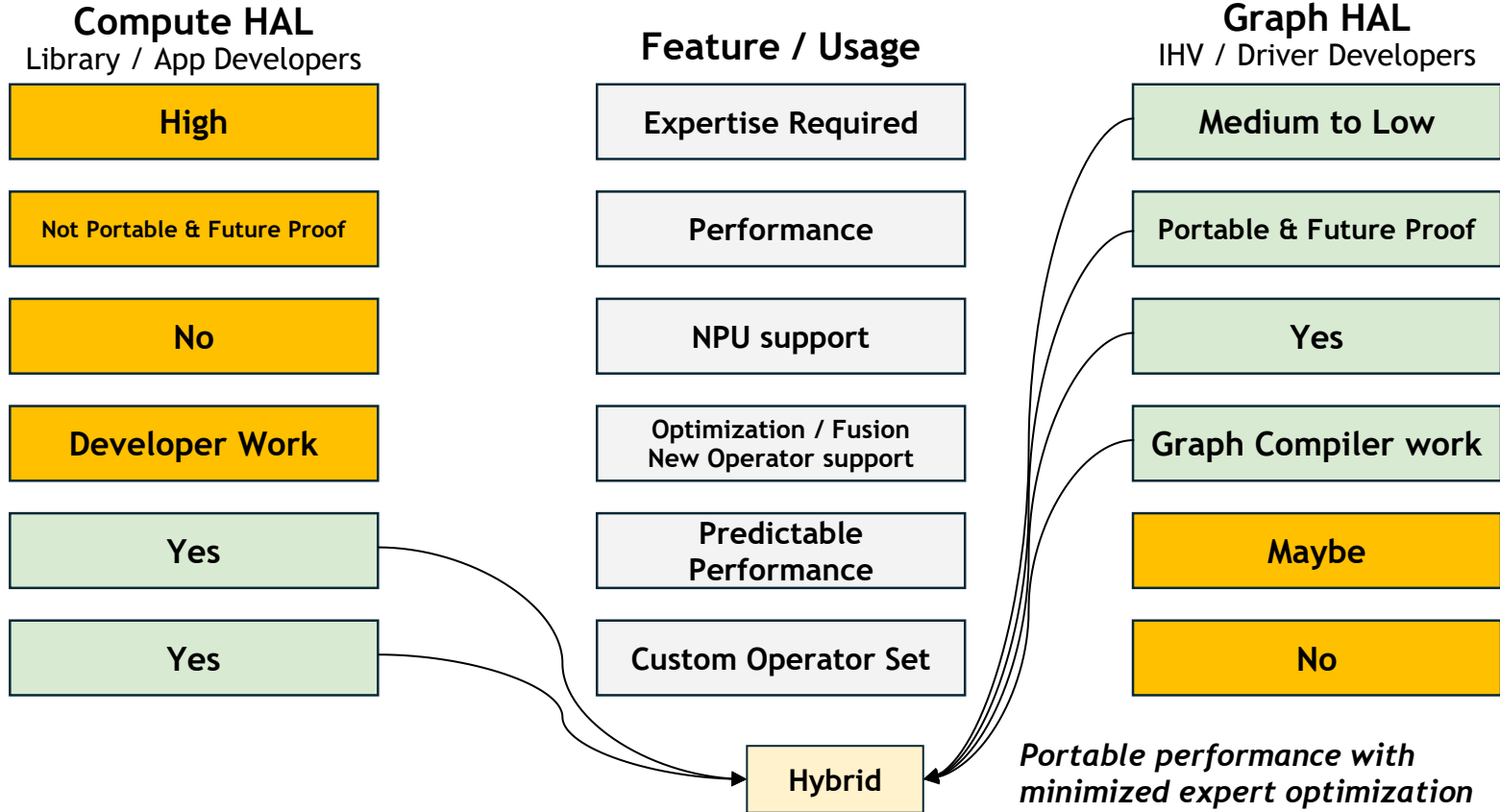
Yes

Graph Compiler work

Maybe

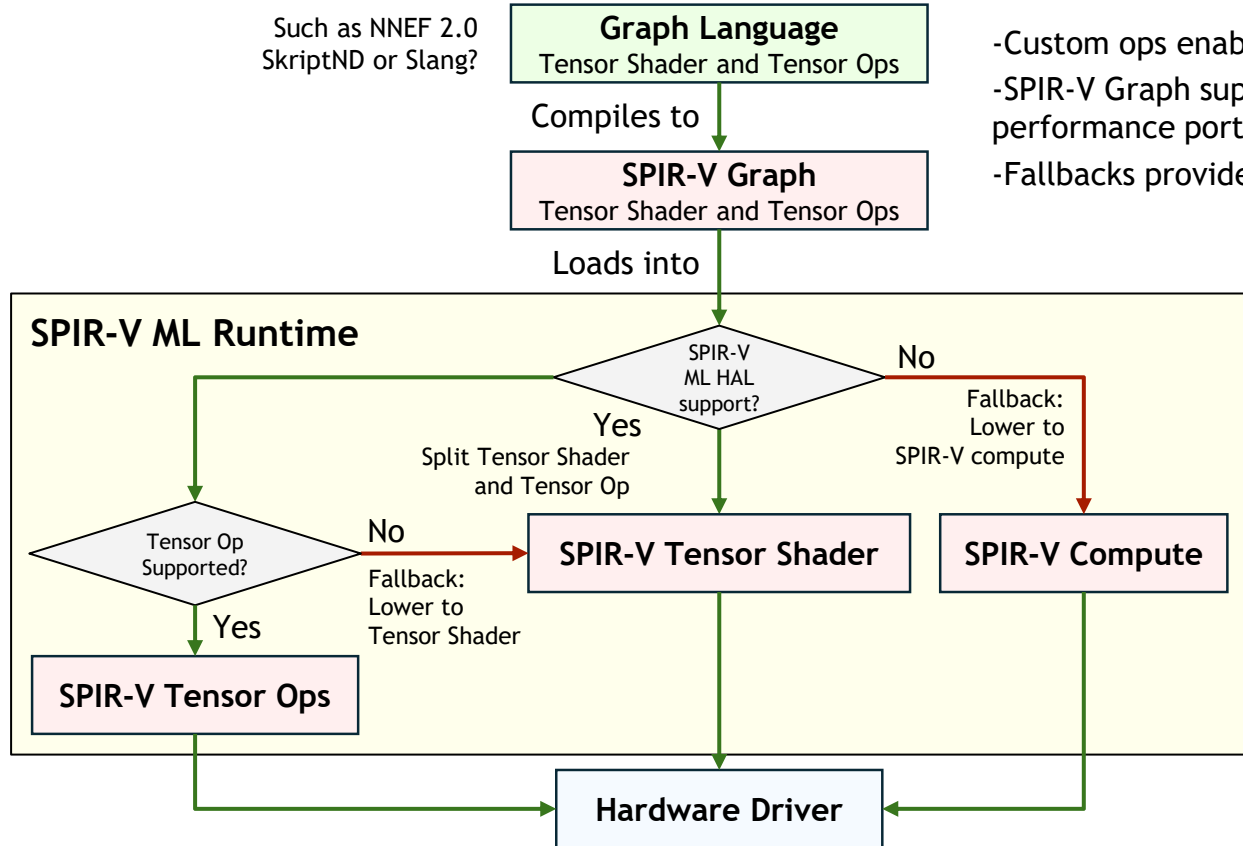
No

Compute HAL vs Graph HAL



Hybrid API Vision

Such as NNEF 2.0
SkriptND or Slang?



- Custom ops enabled by tensor shaders
- SPIR-V Graph support enables performance portability and NPU support
- Fallbacks provide platform portability

Key

Khronos Open Standard or External Standard
Khronos Open Standard
Khronos Open Source
IHV specific



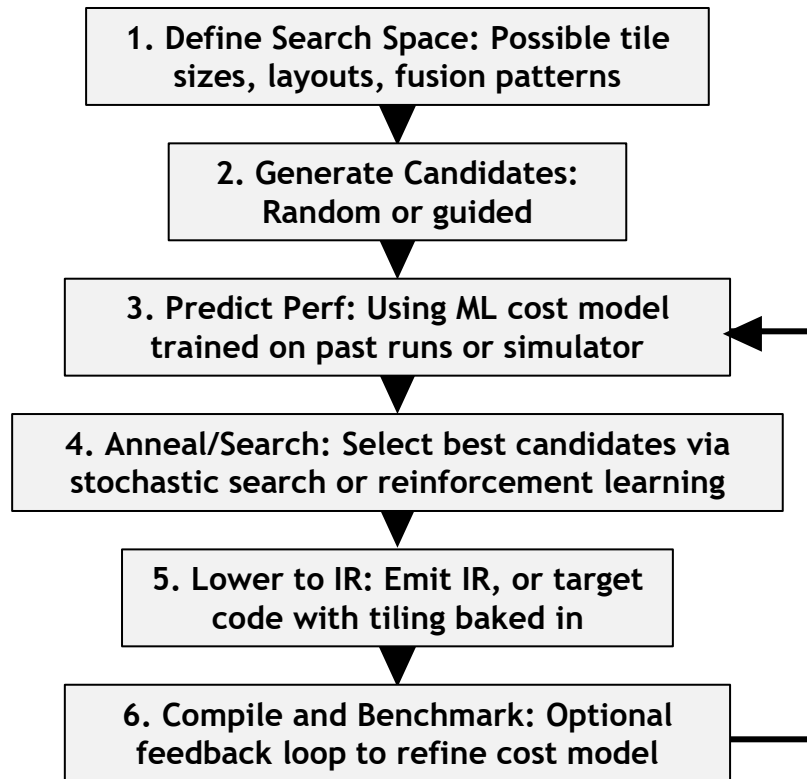
SkriptND on GitHub

Use AI to Drive Performance Optimizations?

AI-Driven annealing performance optimization being actively explored (TVM, IREE, etc.)

AI could (finally) provide a solution to performance portability?

What data should hardware provide to the optimizer?



HW Metadata for Compiler Optimization?

Potential area for exploration?

A consistent method for hardware to describe itself to compilers, enabling easier compiler design and onboarding of new hardware

Standardized optimization metadata to be used through the compilation stack could include:

Tile grid size: e.g., 8×8 grid

Tile compute capacity:

ALUs, tensor cores, MAC units per tile

Local memory per tile:

SRAM size, cache hierarchy, bandwidth

Interconnect topology:

Mesh, NoC latency/cost matrix between tiles

DMA / bandwidth limits:

Bandwidth to global memory or tile neighbors

Execution constraints:

Allowed tile group sizes, sync scopes

Occupancy limits:

Max ops/tensors resident per tile

What does the compiler need to know about the hardware?

How is the information provided to the compiler?

How to achieve critical mass to drive adoption?

Get Involved!

- **We are collecting online survey responses!**
 - Use the QR code below to take a few minutes to give us your views
 - We appreciate your input!
- **Indicate in the survey if you would be open to a 1-1 interview**
 - Your chance to directly influence the evolution of Khronos ML acceleration
 - Particularly if you are an ML developer 🙏
- **Join the Khronos ML Forum for ongoing updates**
 - <https://www.khronos.org/machine-learning>

Thank you for your time today!



[Khronos Group AI
Opportunity Survey](#)



Questions and Discussion

We welcome your thoughts and feedback!