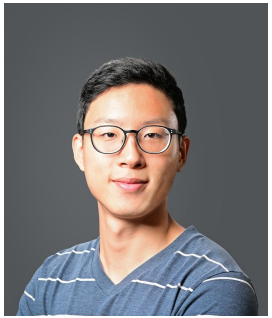
ASPLOS 2026 Tutorial:
AI-Driven Accelerator Programming with LLMLift and Autocomp

Monday, March 23, 2026

Pittsburgh, PA, USA

Organizers



Charles Hong
PhD Candidate
UC Berkeley

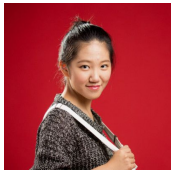
Sahil Bhatia
PhD Candidate
UC Berkeley

Alvin Cheung
Associate Prof
UC Berkeley

Sophia Shao
Associate Prof
UC Berkeley

Berkeley
UNIVERSITY OF CALIFORNIA

SLICE

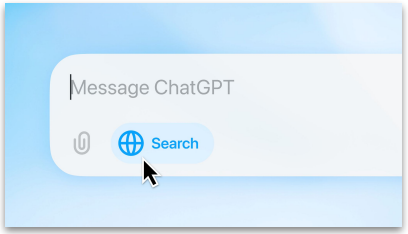



w/ **Jie Qiu**, **Huijae An**, **Boru Chen**, **Jack Toubes**

Special thanks to…

AWS for providing support and the instances you will be using today!

AI is everywhere



Chatbots

Robots

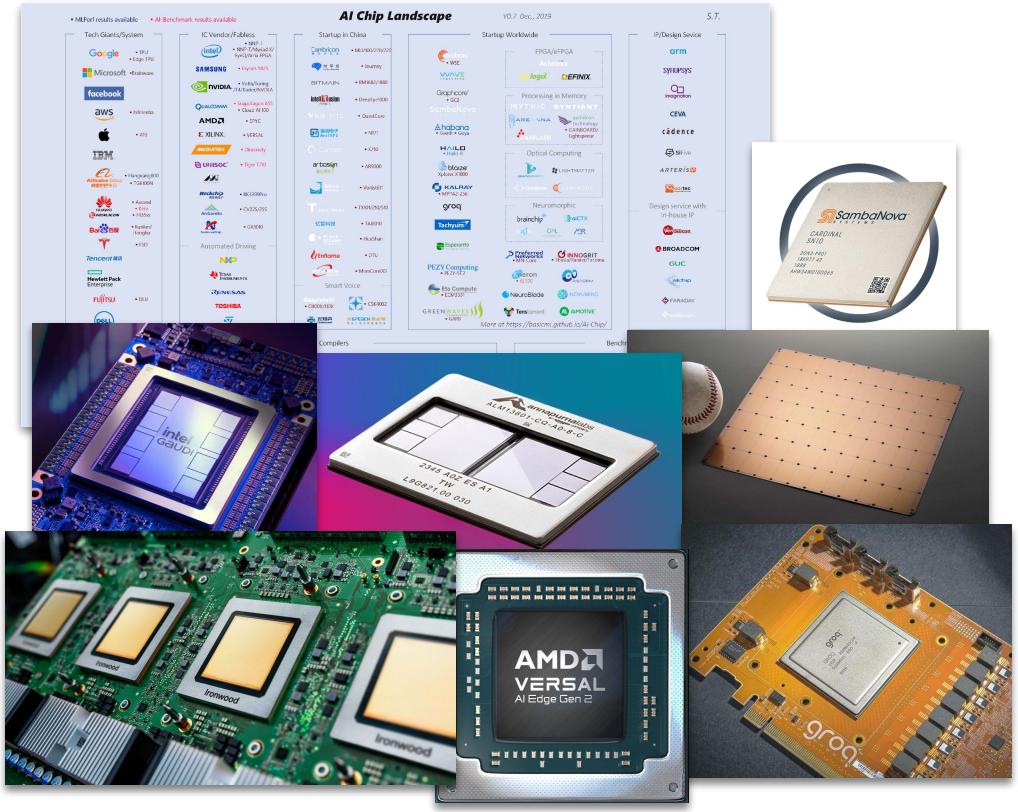
Mobile phones

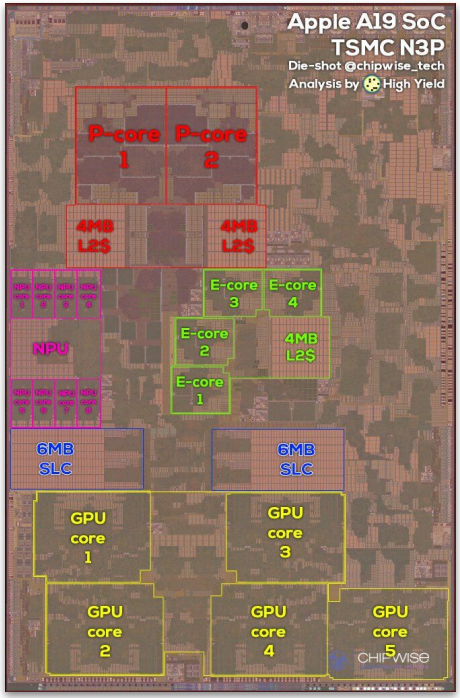
Augmented Reality

Autonomous Vehicles

Genomics

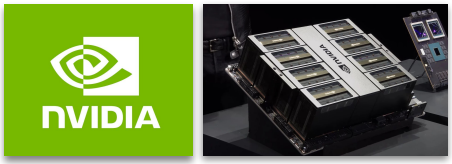
AI chips are everywhere





The real AI chip landscape

The Big Players



Google

🔒 **The Moat: Software (CUDA)**

NVIDIA wins thanks to **decades of hand-tuning** and **customer lock-in** to tooling and infrastructure. Only deep vertical integration (Google) can compete.

The Challengers

AMD

Qualcomm

intel

cerebras

groq

annapurnalabs an amazon company

SambaNova SYSTEMS

tenstorrent

ETCHED

+ many more

⚠️ **The Pain Point**

Can be technically superior (faster/cheaper per watt), but fail to match software user experience in practice.
“Hardware rich, software poor.”

Why is AI software so difficult?

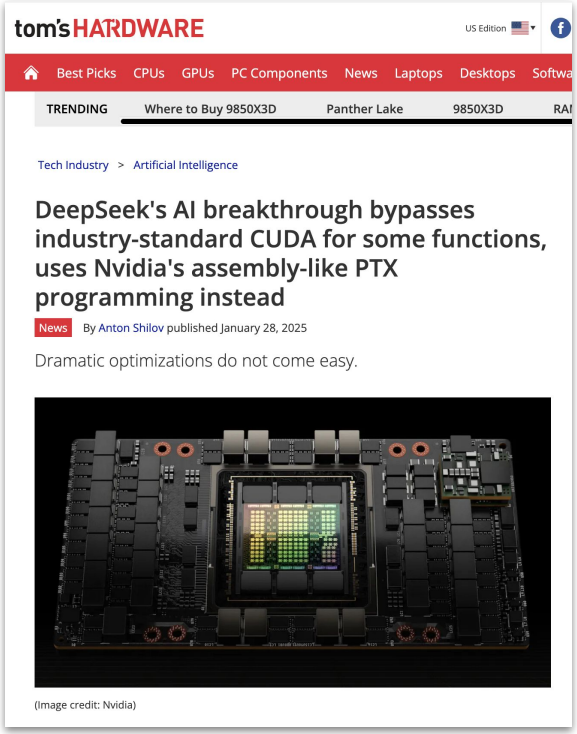
Traditional Compilers

- Decades of development
- CPU ISA and micro-architecture rarely change
- Well-established boundary between HW/SW

AI Software

- Software is a moving target (MLPs→CNNs→LLMs→?)
- Hardware is a moving target (Ampere→Hopper→Blackwell)
- No clear ISA boundary
- HW/SW co-design is common (quantization, sparsity)

The end result: hand-written, low-level code



Disclaimer: No PTX in this presentation.

AI Software

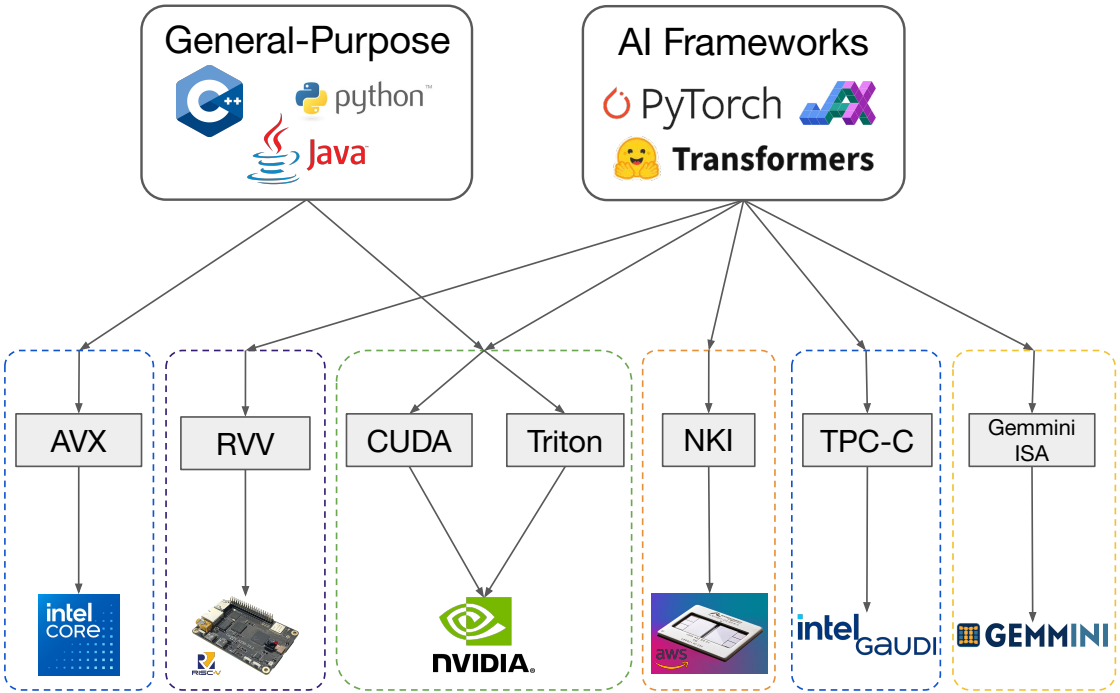
- Hardware is a moving target (Ampere→Hopper→Blackwell)
- No clear ISA boundary
- HW/SW co-design is common (quantization, sparsity)
- Relatively new, with requirements rapidly changing (LLMs!)

Lots of sources, lots of targets…

High-Level Languages

Low-Level Performance Programming

Heterogeneous Accelerators

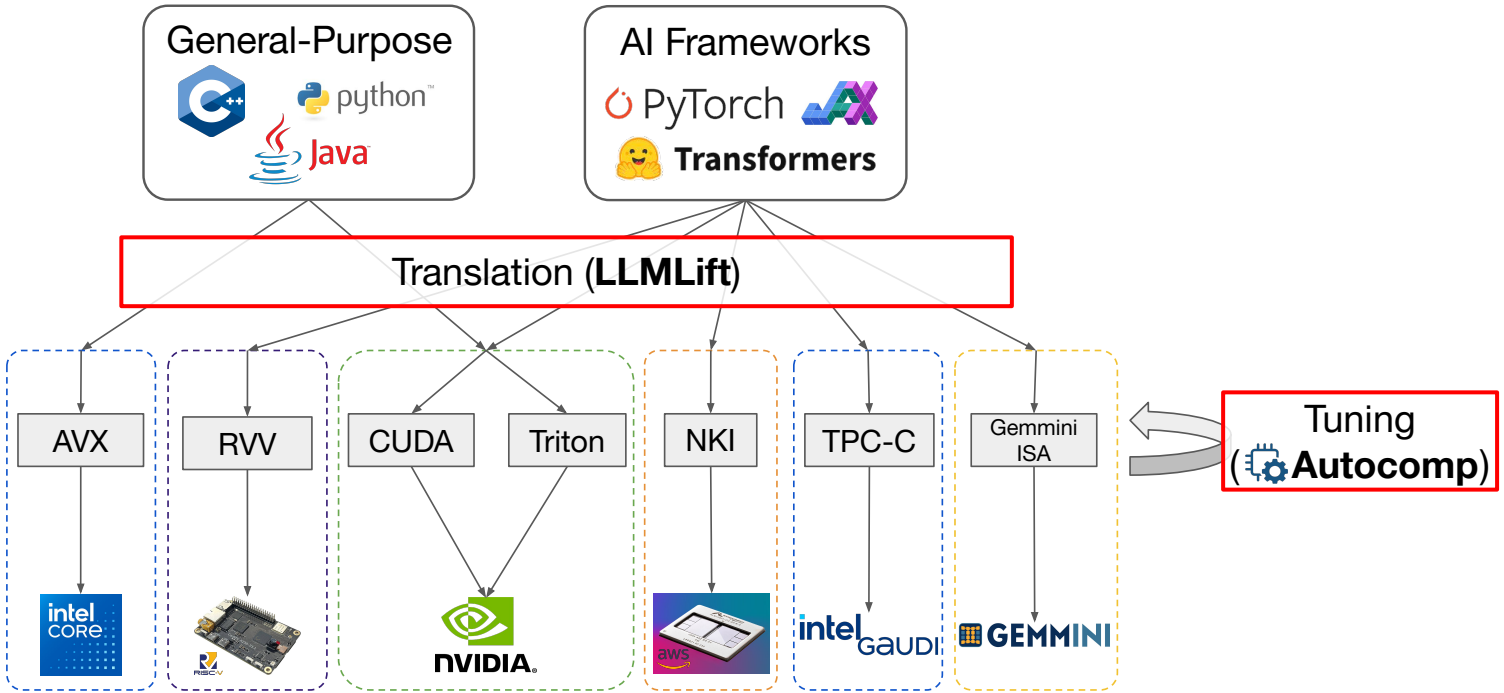


Our solution: LLMLift + Autocomp

High-Level Languages

Low-Level Performance Programming

Heterogeneous Accelerators



Agenda

8:30–8:45 Introduction

8:45–10:00 LLMLift

- 8:40–9:00 Talk: LLMLift: Verified Code Transpilation with LLMs
- 9:00–10:00 Hands-on: Translating Python to Accelerator DSL

10:00–10:30 Coffee Break ☕

10:30–11:50 Autocomp

- 10:30–10:50 Talk: Autocomp: A Portable Code Optimizer for Tensor Accelerators
- 10:50–11:50 Hands-on: Building a Trainium Code Optimizer

11:50–12:00 Q&A and Future Directions