AccelOpt

A Self-Improving LLM Agentic System for AI Accelerator Kernel Optimization

Genghan Zhang, Shaowei Zhu, Anjiang Wei, Zhenyu Song, Allen Nie, Zhen Jia,
Nandita Vijaykumar, Yida Wang, Kunle Olukotun

Stanford University aws UNIVERSITY OF TORONTO

Kernels are Central to ML Systems

Operator

AI accelerator

Kernels are Central to ML Systems

Operator

AI accelerator

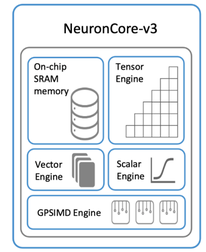
Kernels are Central to ML Systems

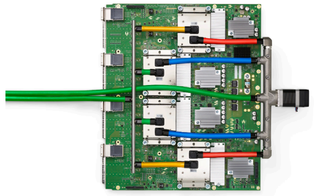
Operator

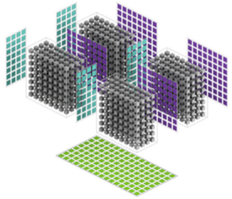
PyTorch

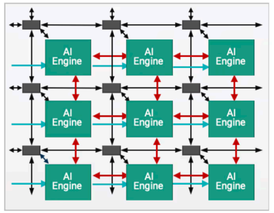
NumPy

AI accelerator









Kernels are Central to ML Systems

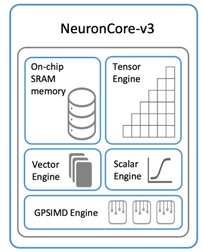
Operator

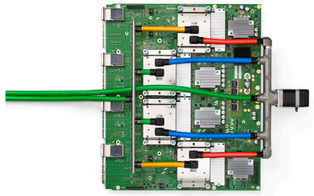
PyTorch

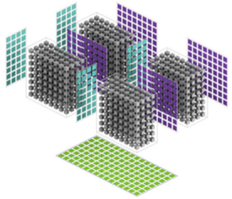
NumPy

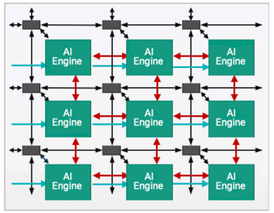
Kernel

AI accelerator









Kernels are Central to ML Systems

Operator

PyTorch

NumPy

Kernel

aws NKI

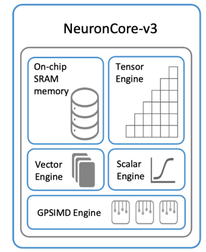
Google Pallas

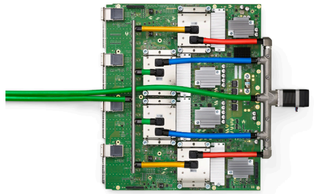
NVIDIA CUDA

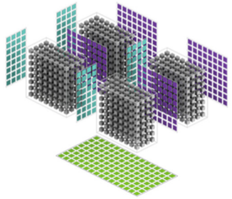
AMD ROCm

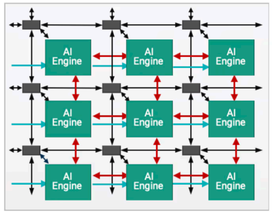
Triton

AI accelerator









Kernels are Central to ML Systems

Operator → **Kernel** → **AI accelerator**

PyTorch

NumPy

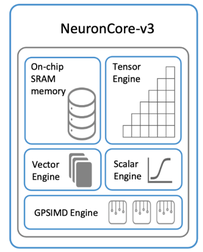
aws NKI

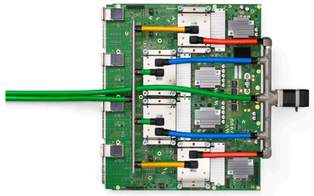
Google Pallas

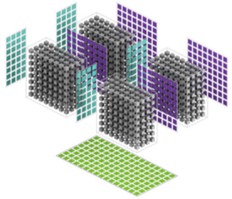
NVIDIA CUDA

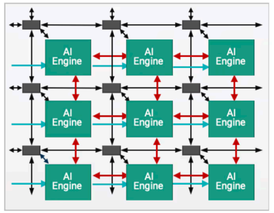
AMD ROCm

Triton









Kernels are Central to ML Systems

Operator → **Kernel** → **AI accelerator**

PyTorch

NumPy

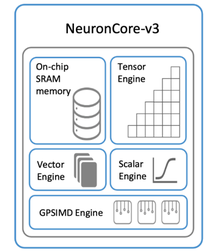
aws NKI

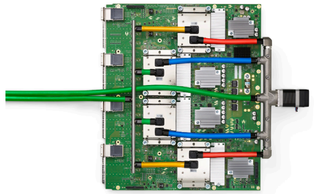
Google Pallas

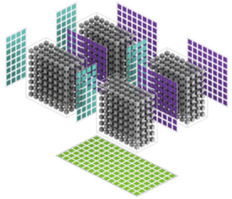
NVIDIA CUDA

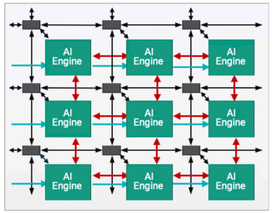
AMD ROCm

Triton

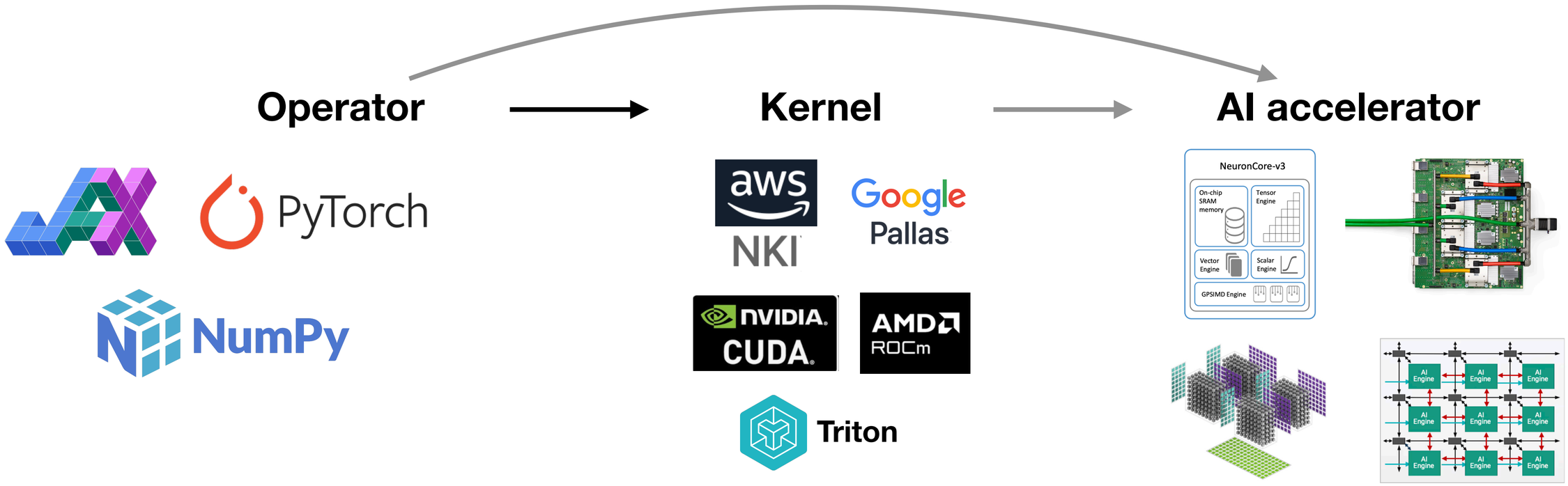




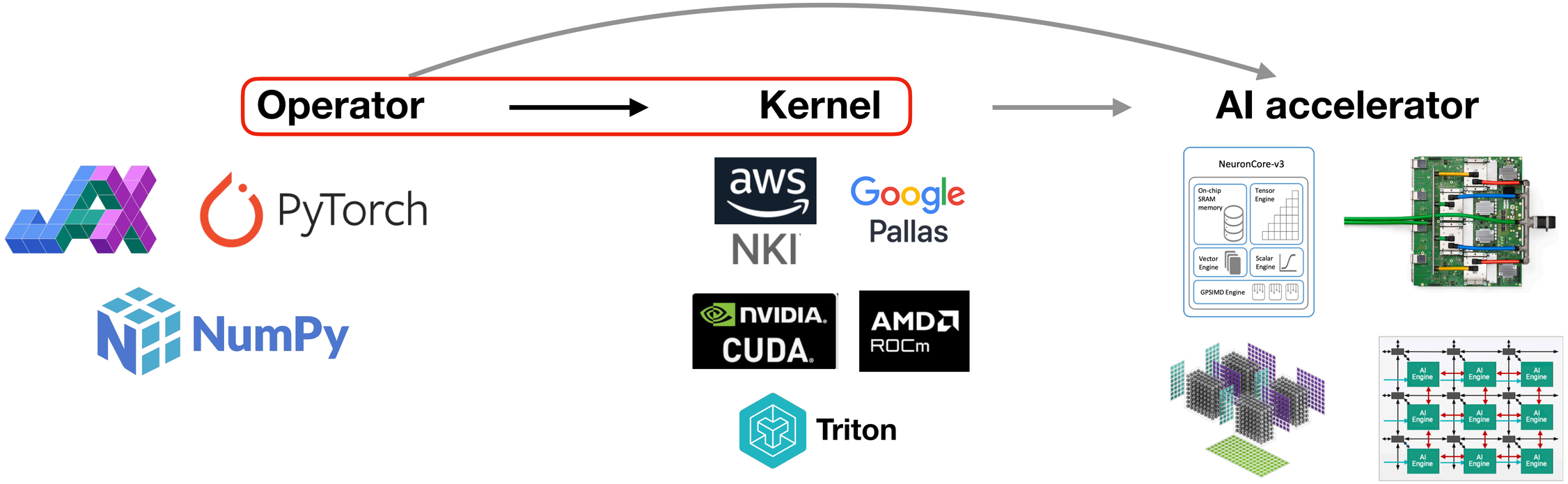




Kernels are Central to ML Systems

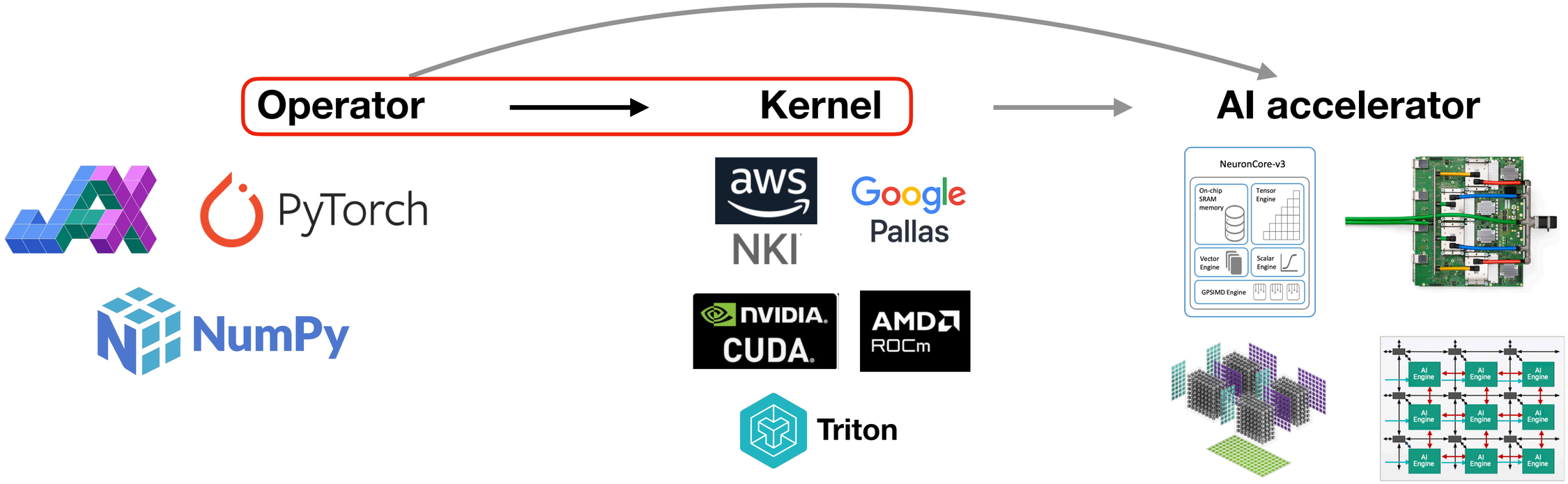


Kernels are Central to ML Systems



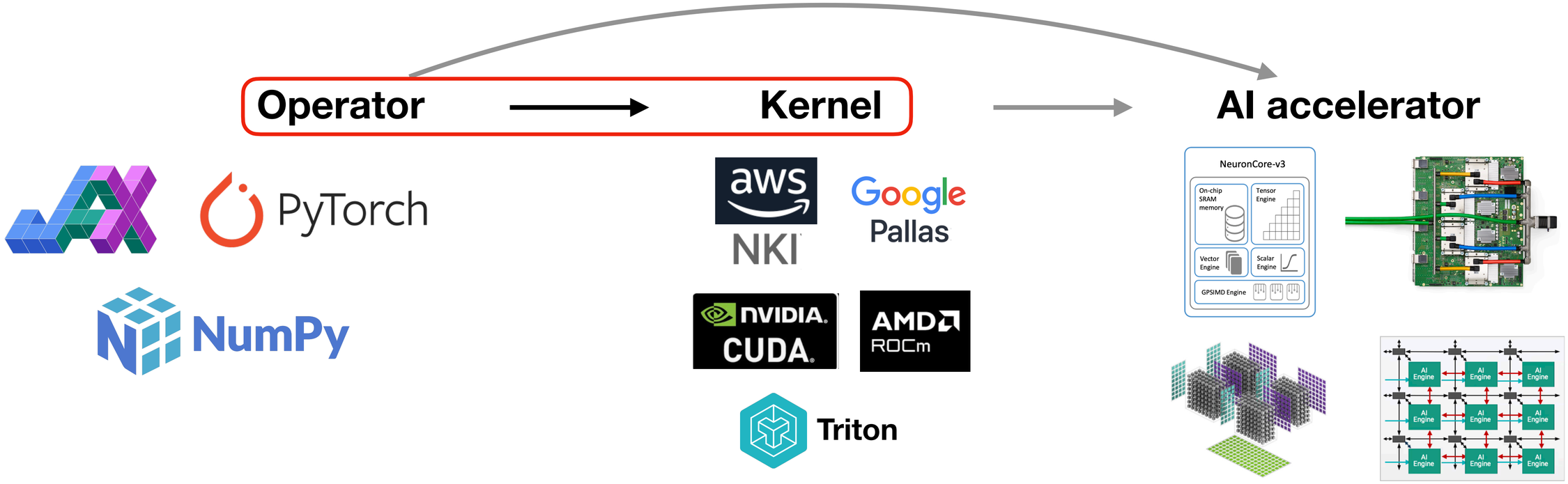
Kernels are Central to ML Systems

Optimizing kernels is hard, time-consuming and requires expertise!



Kernels are Central to ML Systems

Optimizing kernels is hard, time-consuming and requires expertise!



Can we build LLM agentic systems to do this?

Key techniques

Key techniques

- Beam search for fast kernels

- Decompose optimization to two agents: planner and executor

- Guide beam search with optimization memory

1. Beam search for fast kernels

Beam search: select the best performant kernels from the current iteration as the candidates for the next iteration.

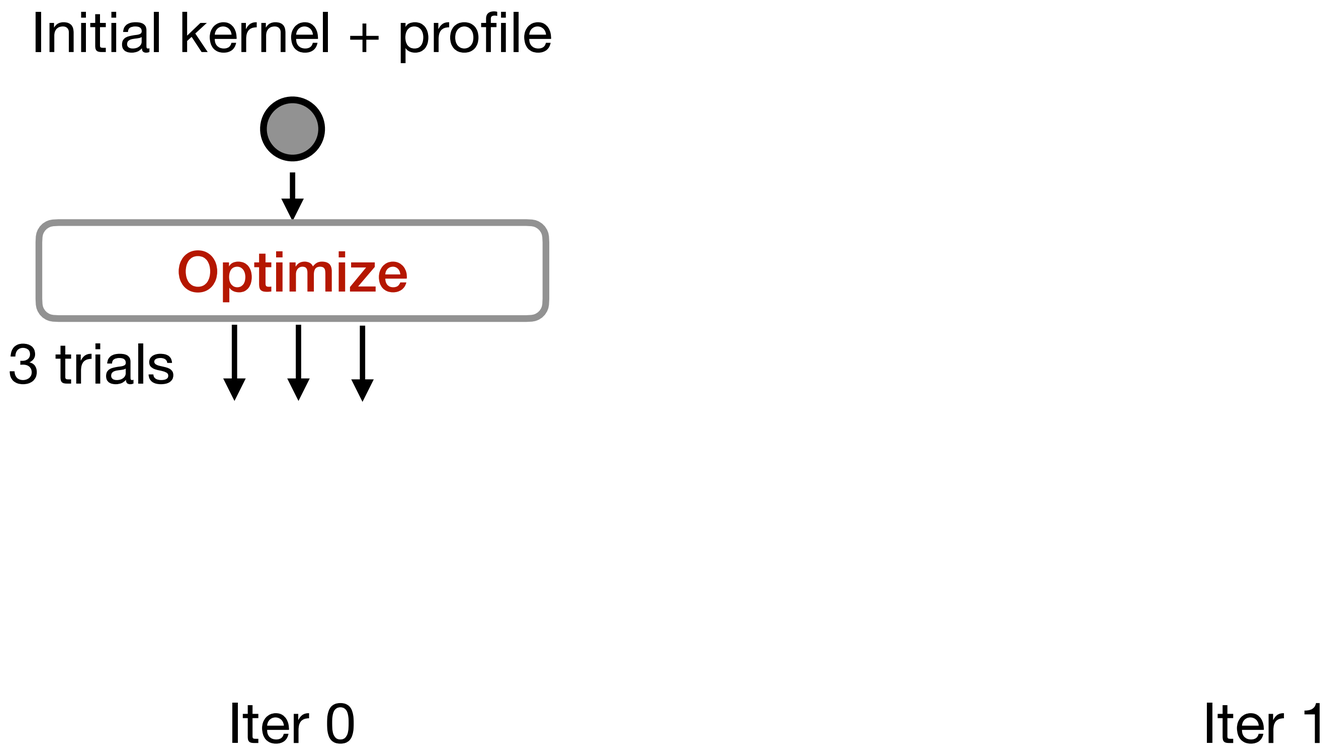
Initial kernel + profile

Iter 0

Iter 1

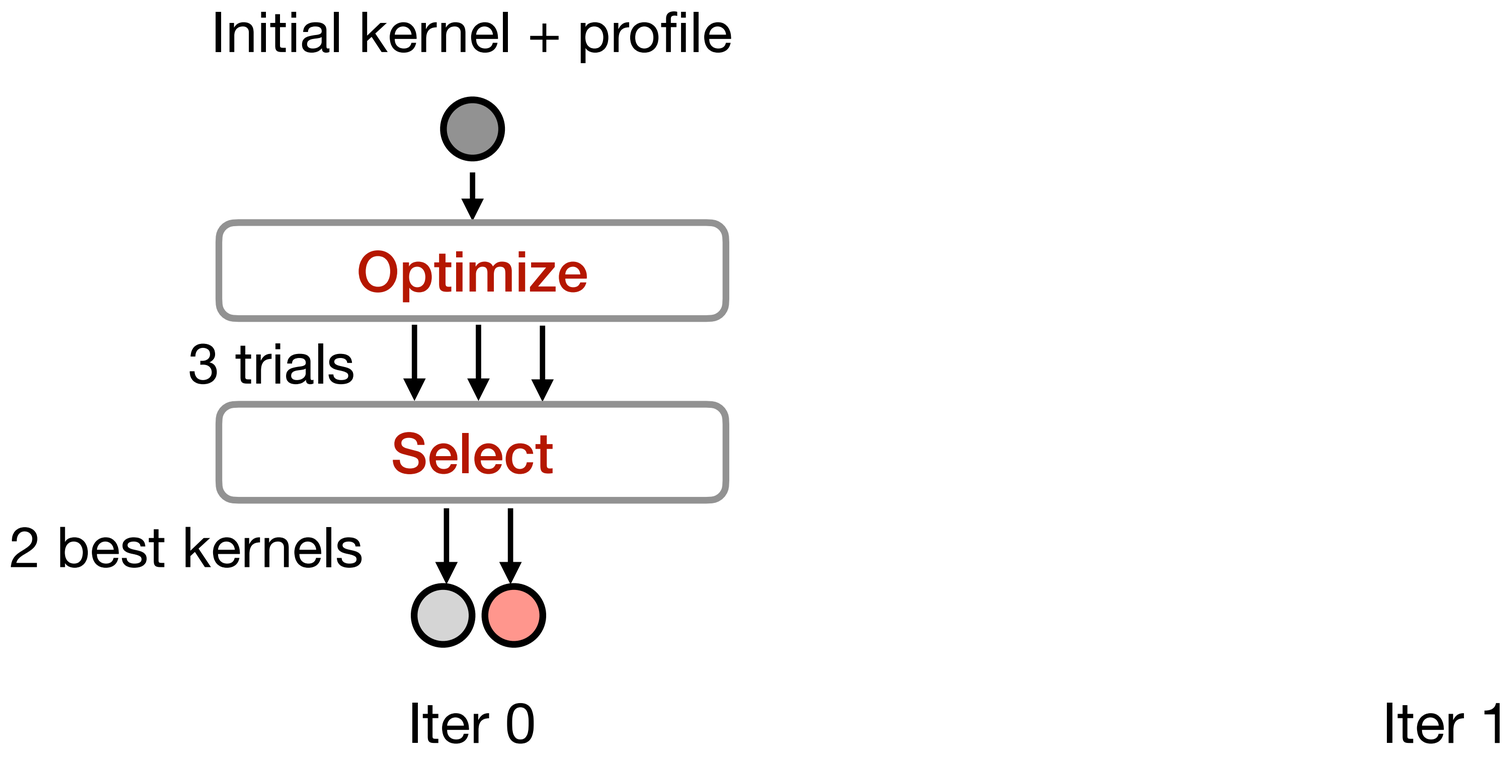
1. Beam search for fast kernels

Beam search: select the best performant kernels from the current iteration as the candidates for the next iteration.



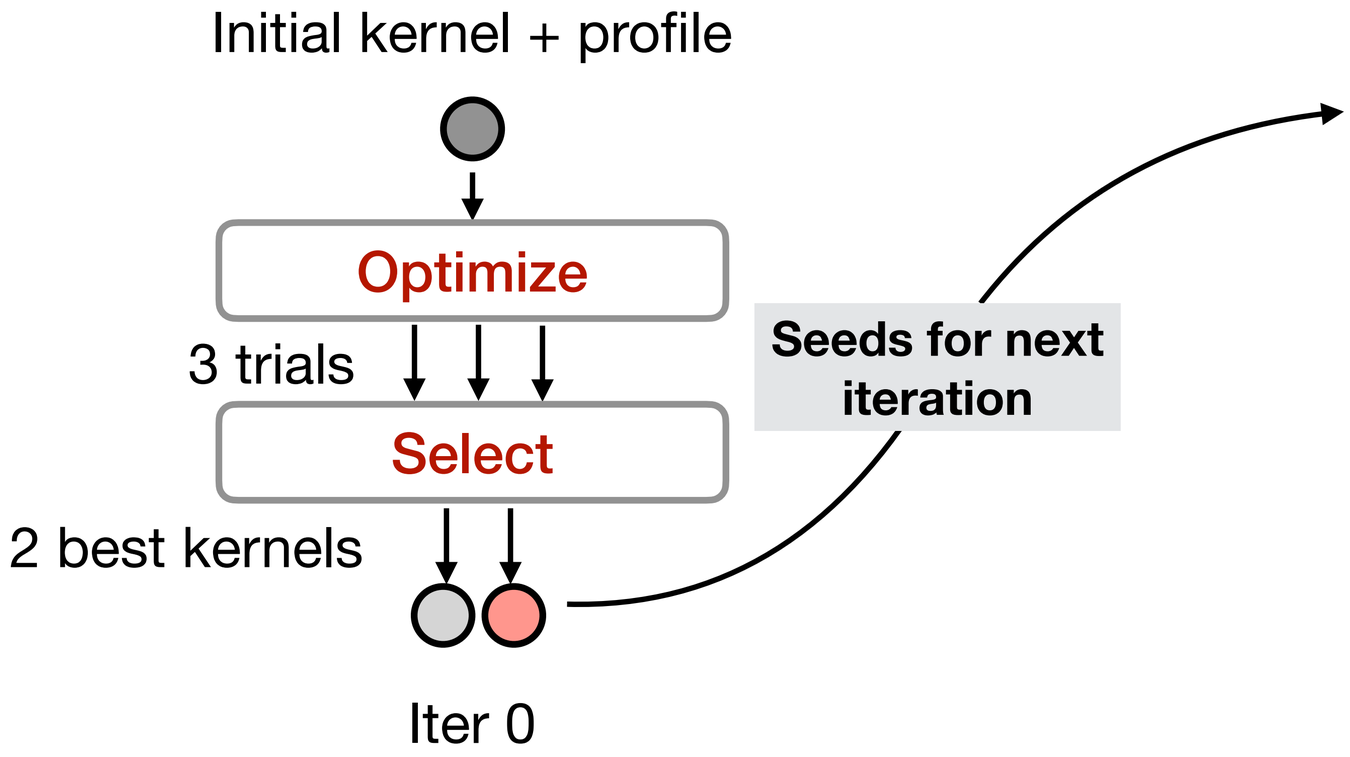
1. Beam search for fast kernels

Beam search: select the best performant kernels from the current iteration as the candidates for the next iteration.



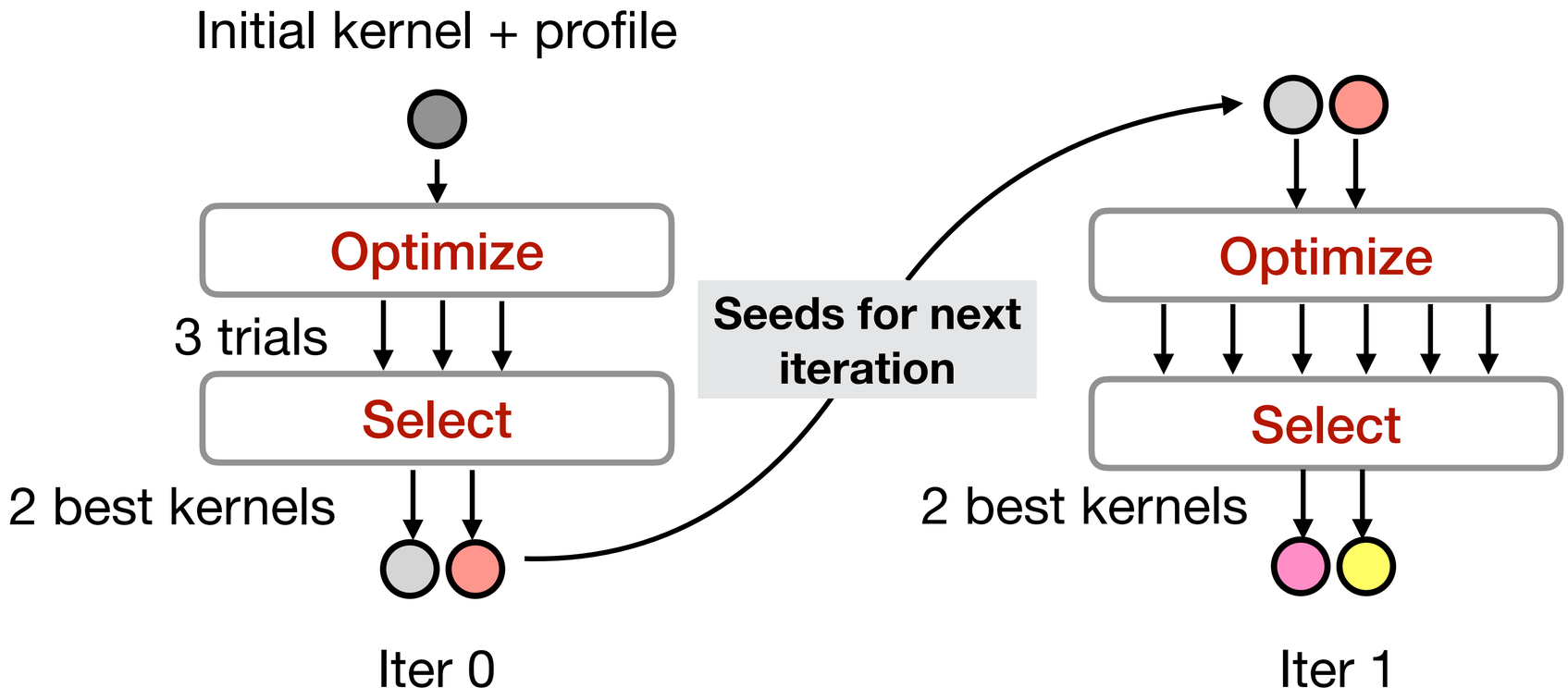
1. Beam search for fast kernels

Beam search: select the best performant kernels from the current iteration as the candidates for the next iteration.



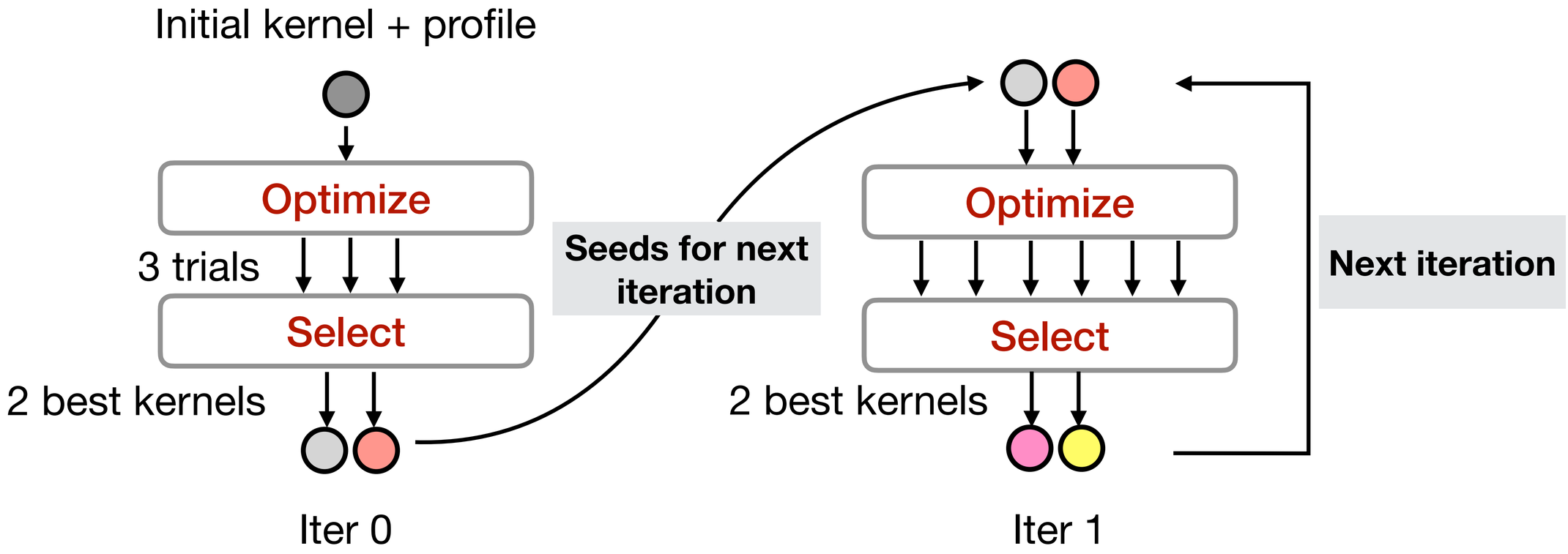
1. Beam search for fast kernels

Beam search: select the best performant kernels from the current iteration as the candidates for the next iteration.

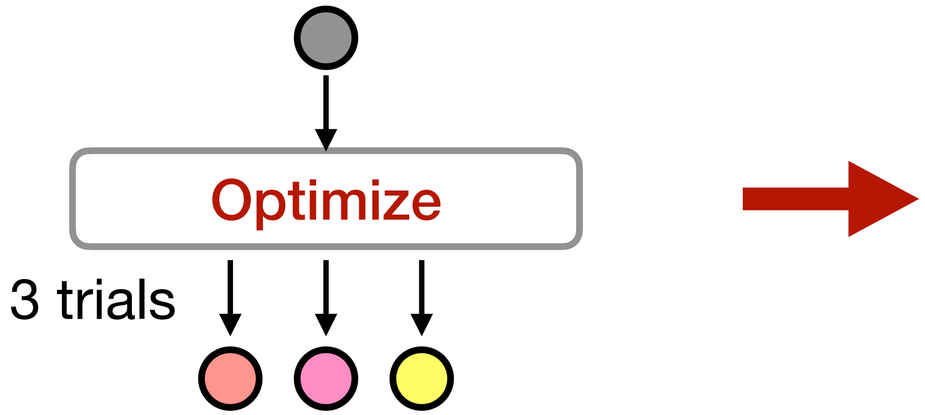


1. Beam search for fast kernels

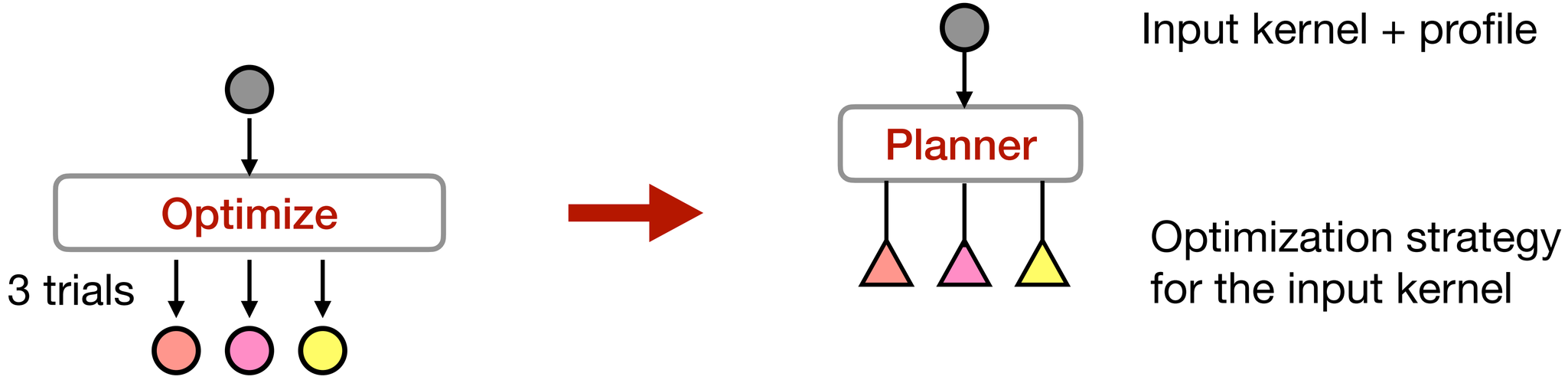
Beam search: select the best performant kernels from the current iteration as the candidates for the next iteration.



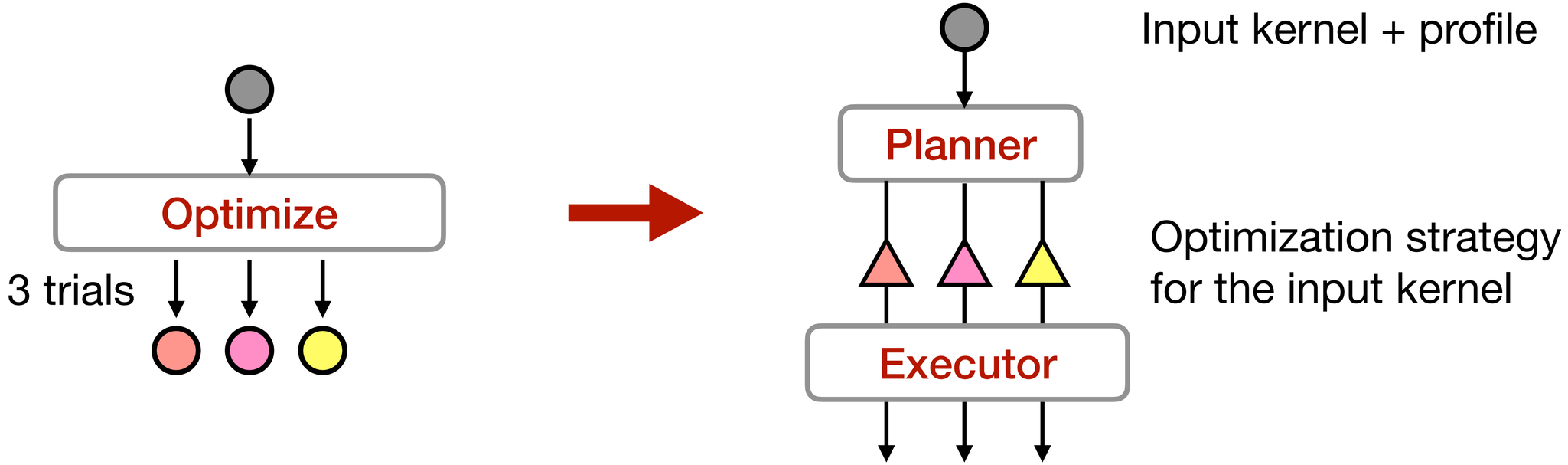
2. Decompose optimization to two agents: planner and executor



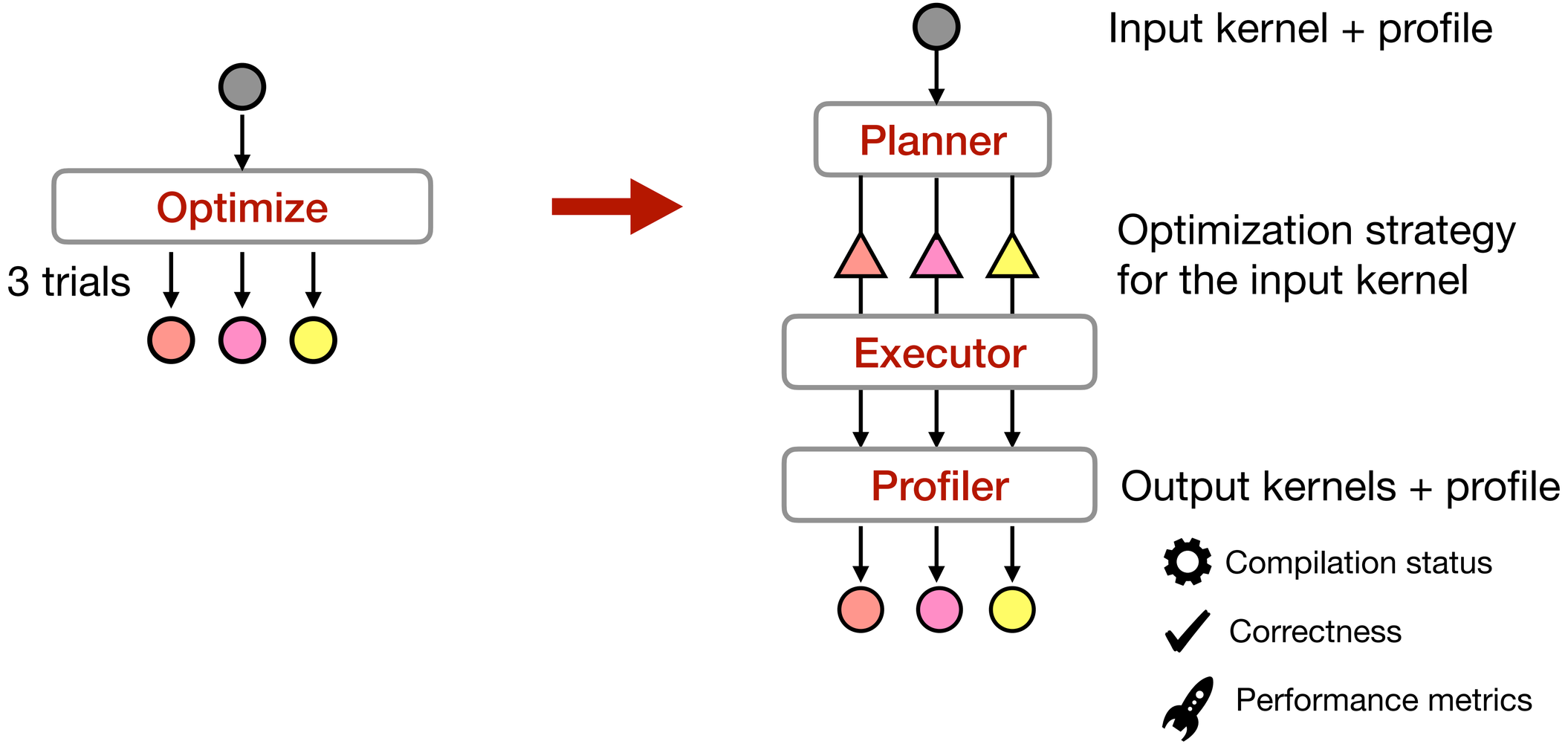
2. Decompose optimization to two agents: planner and executor



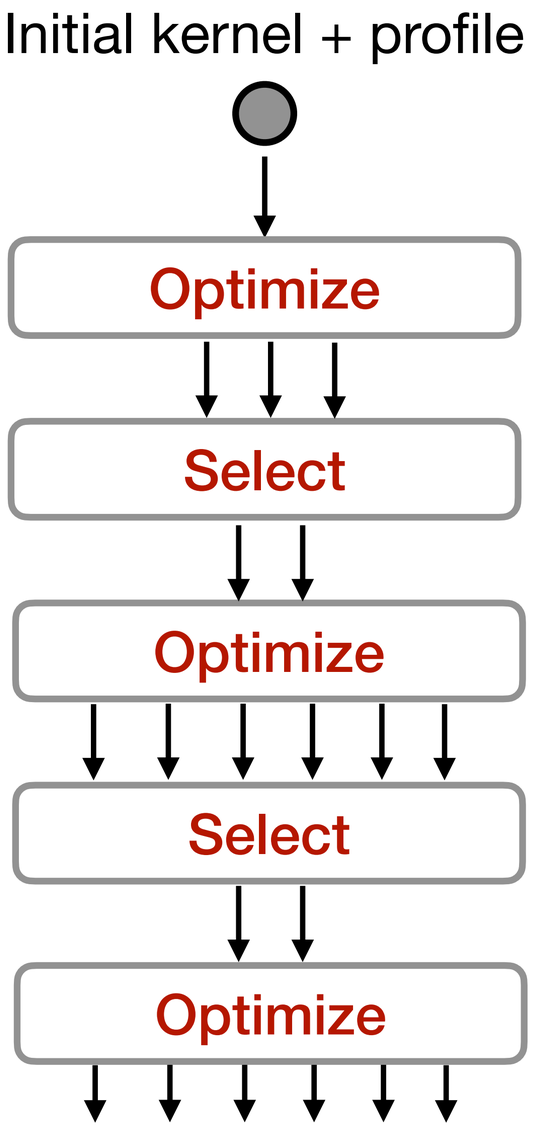
2. Decompose optimization to two agents: planner and executor

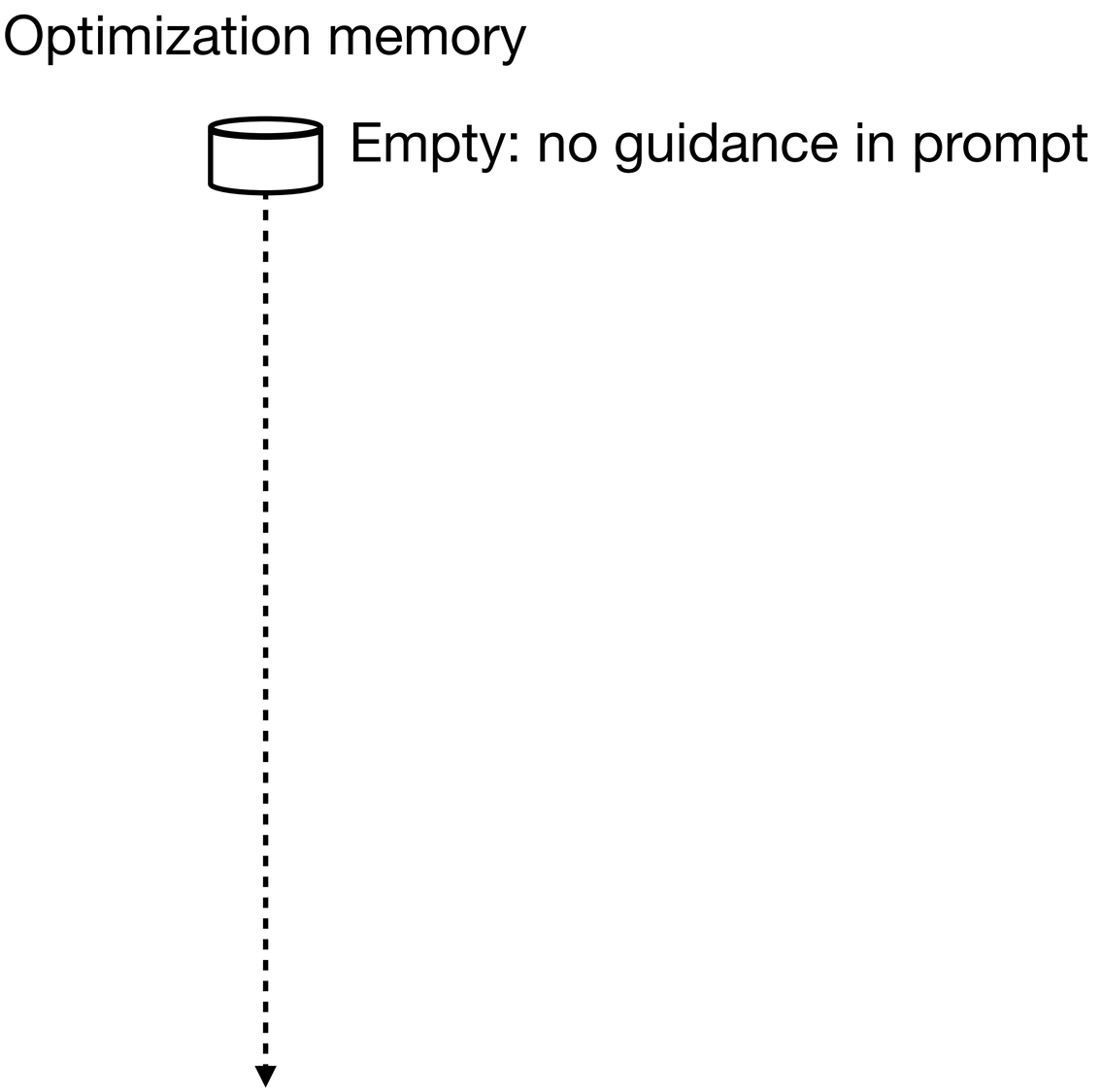


2. Decompose optimization to two agents: planner and executor

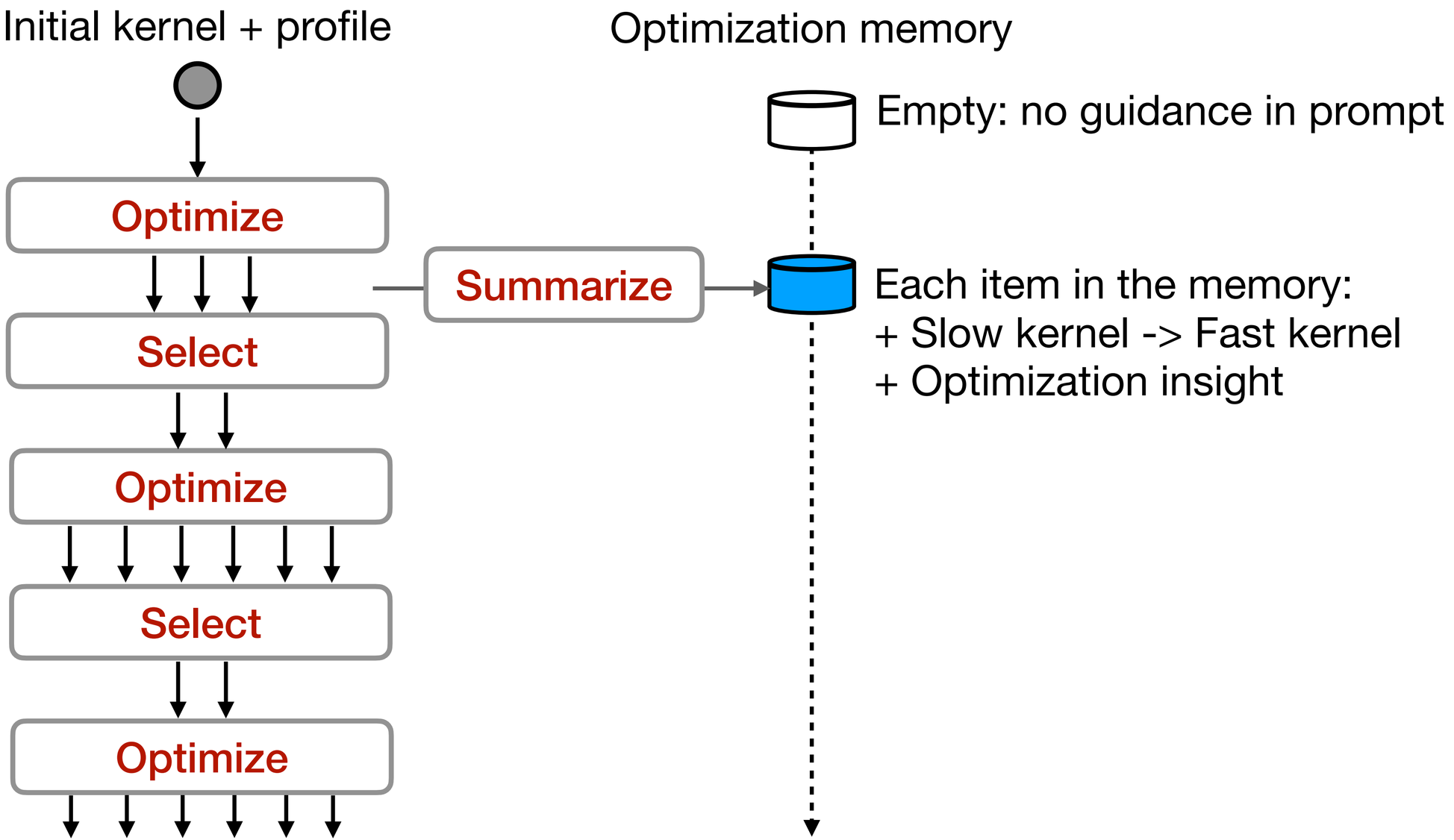


3. Guide beam search with optimization memory

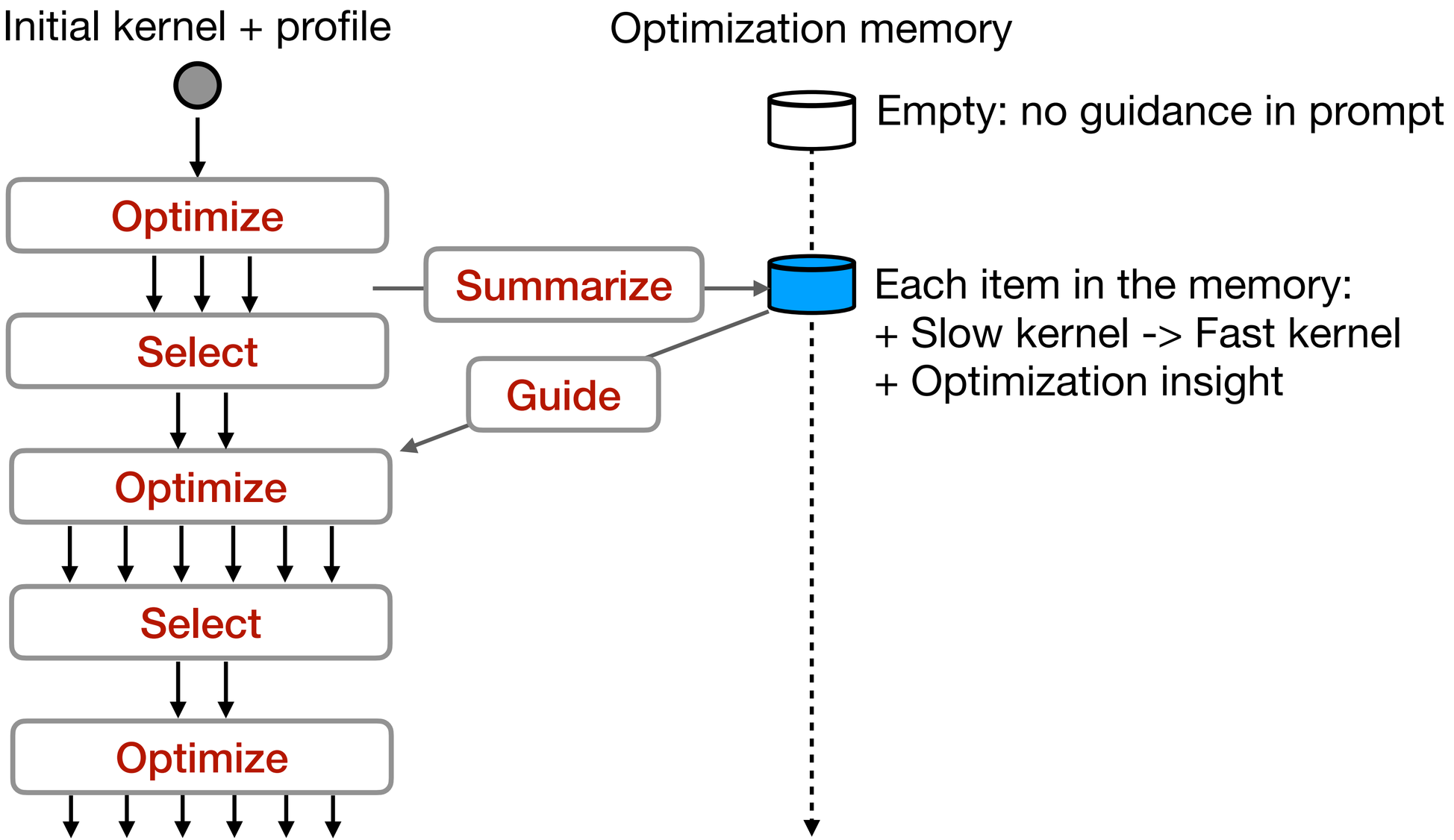




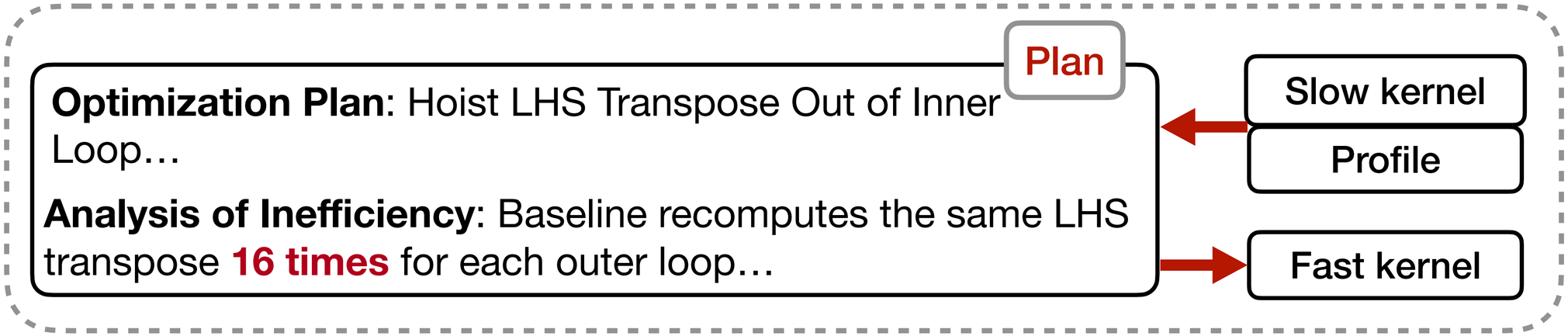
3. Guide beam search with optimization memory



3. Guide beam search with optimization memory

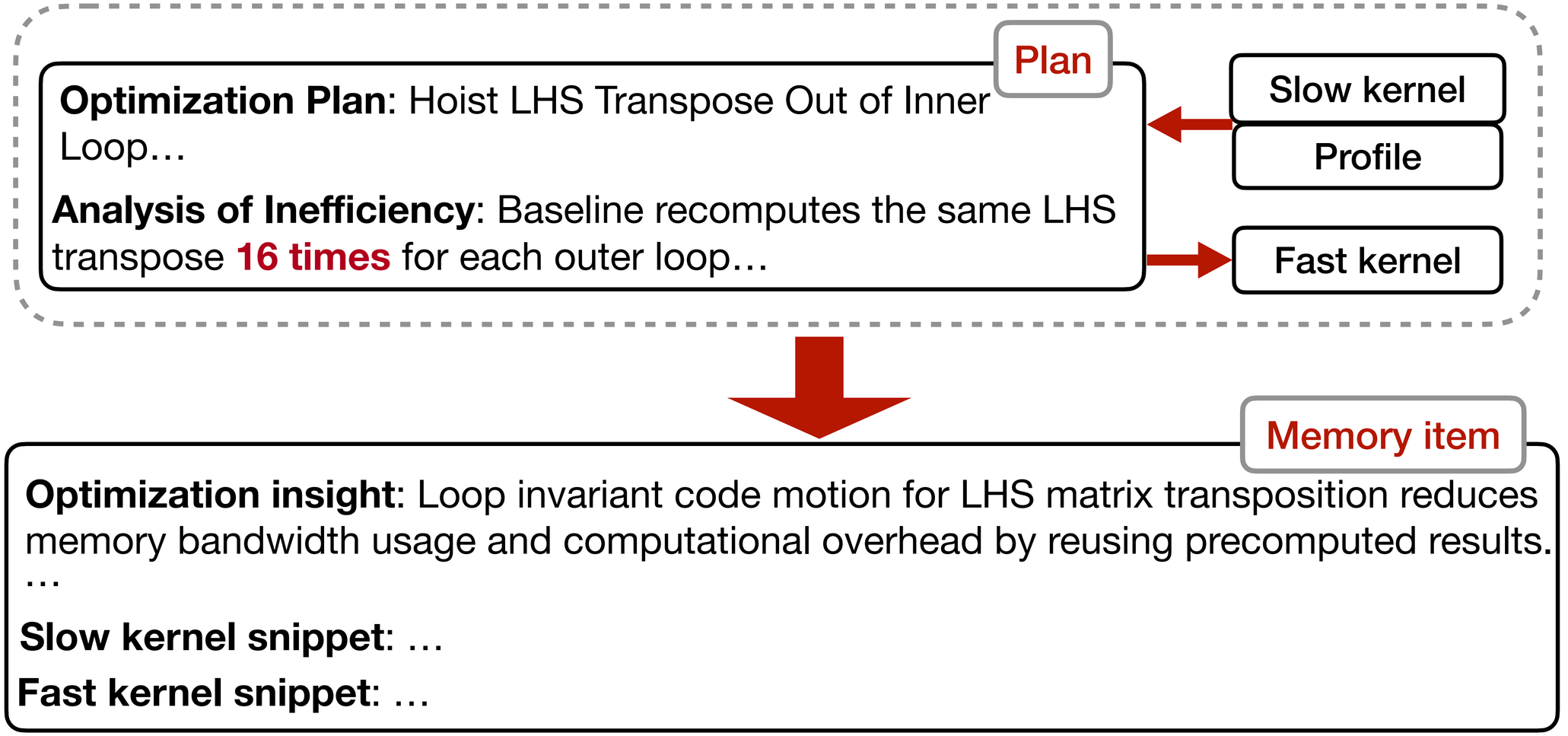


An item in the optimization memory



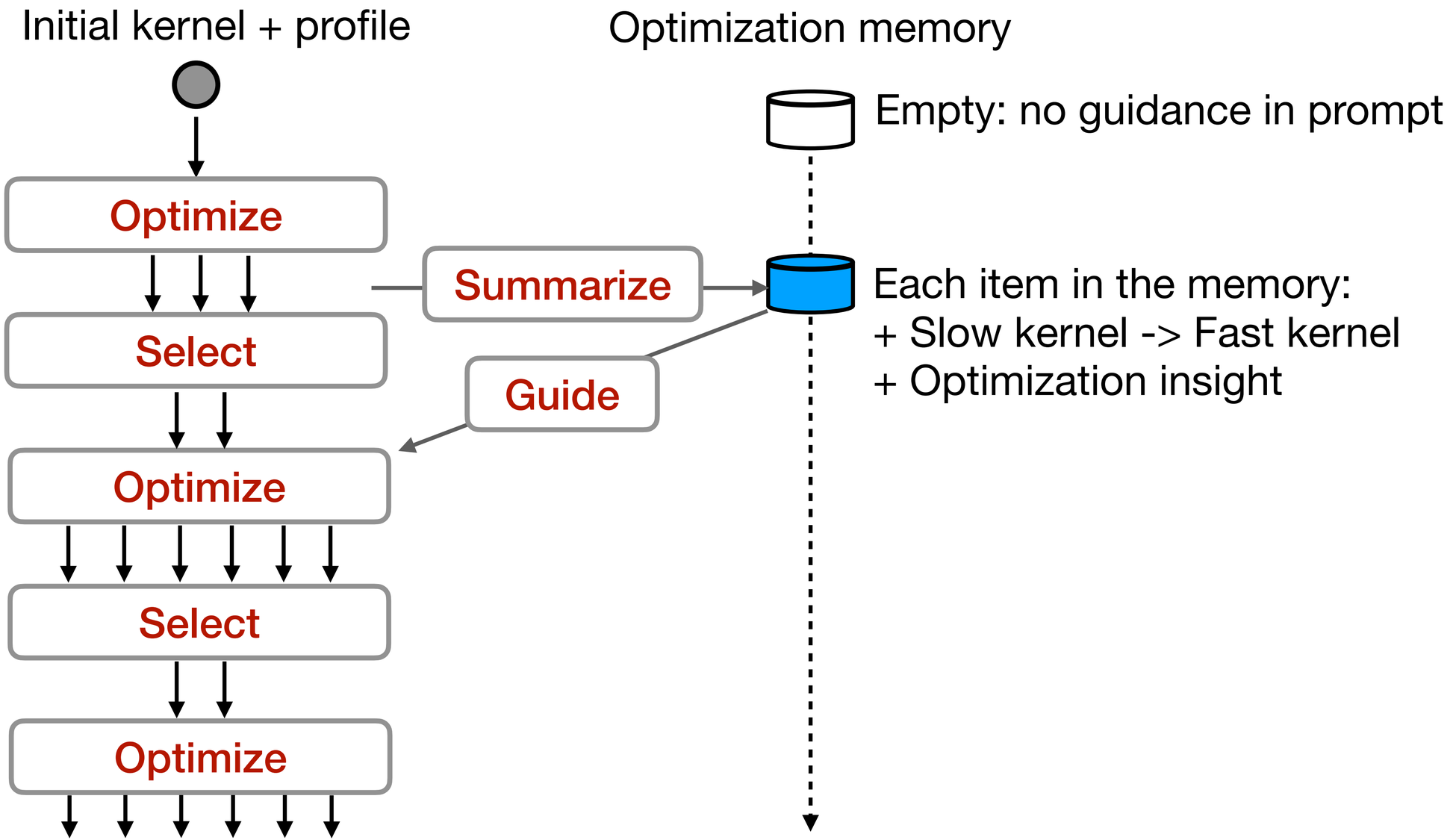
(All the contents were generated by agents in AccelOpt and modified a bit for demonstration.)

An item in the optimization memory

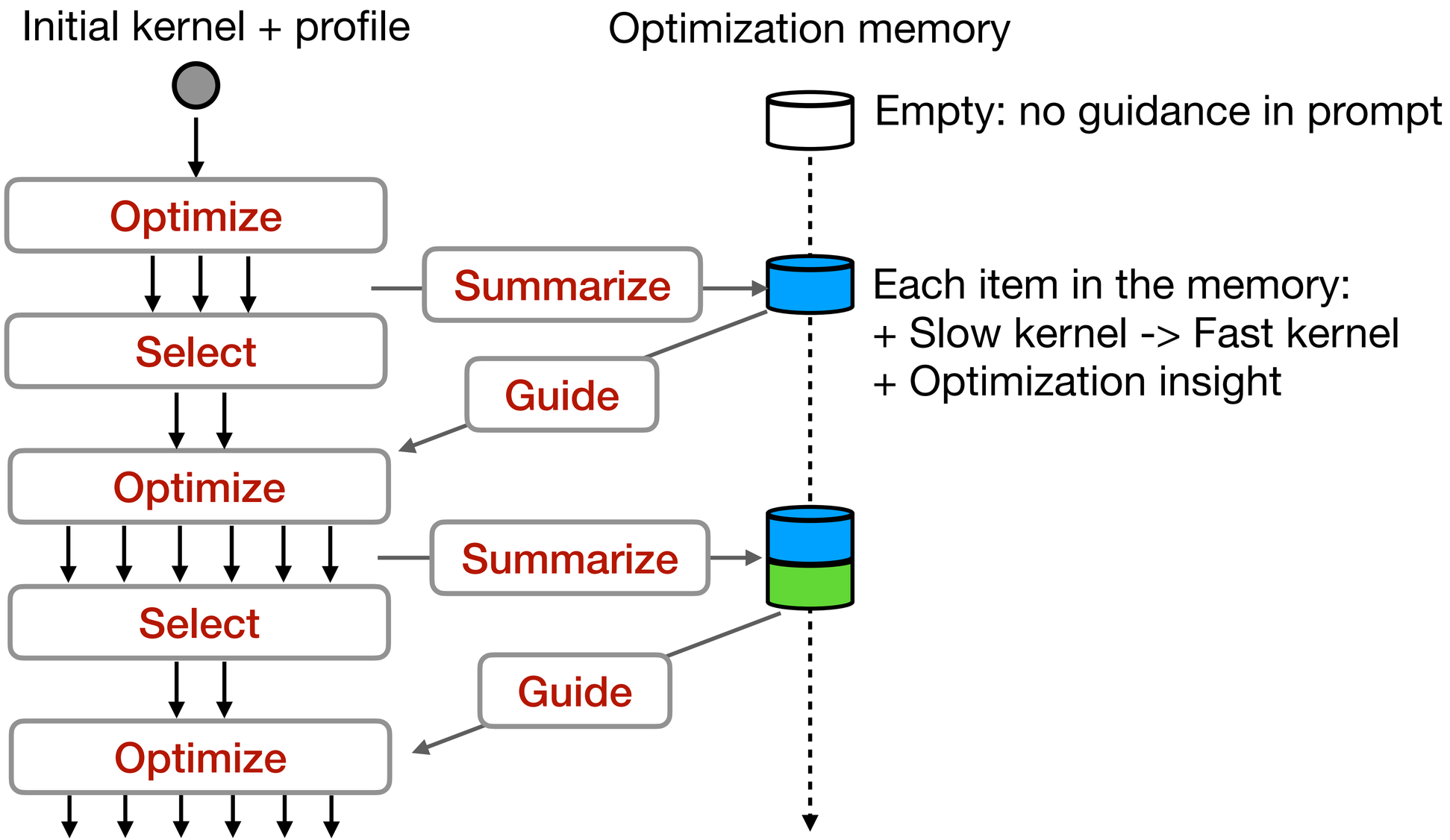


(All the contents were generated by agents in AccelOpt and modified a bit for demonstration.)

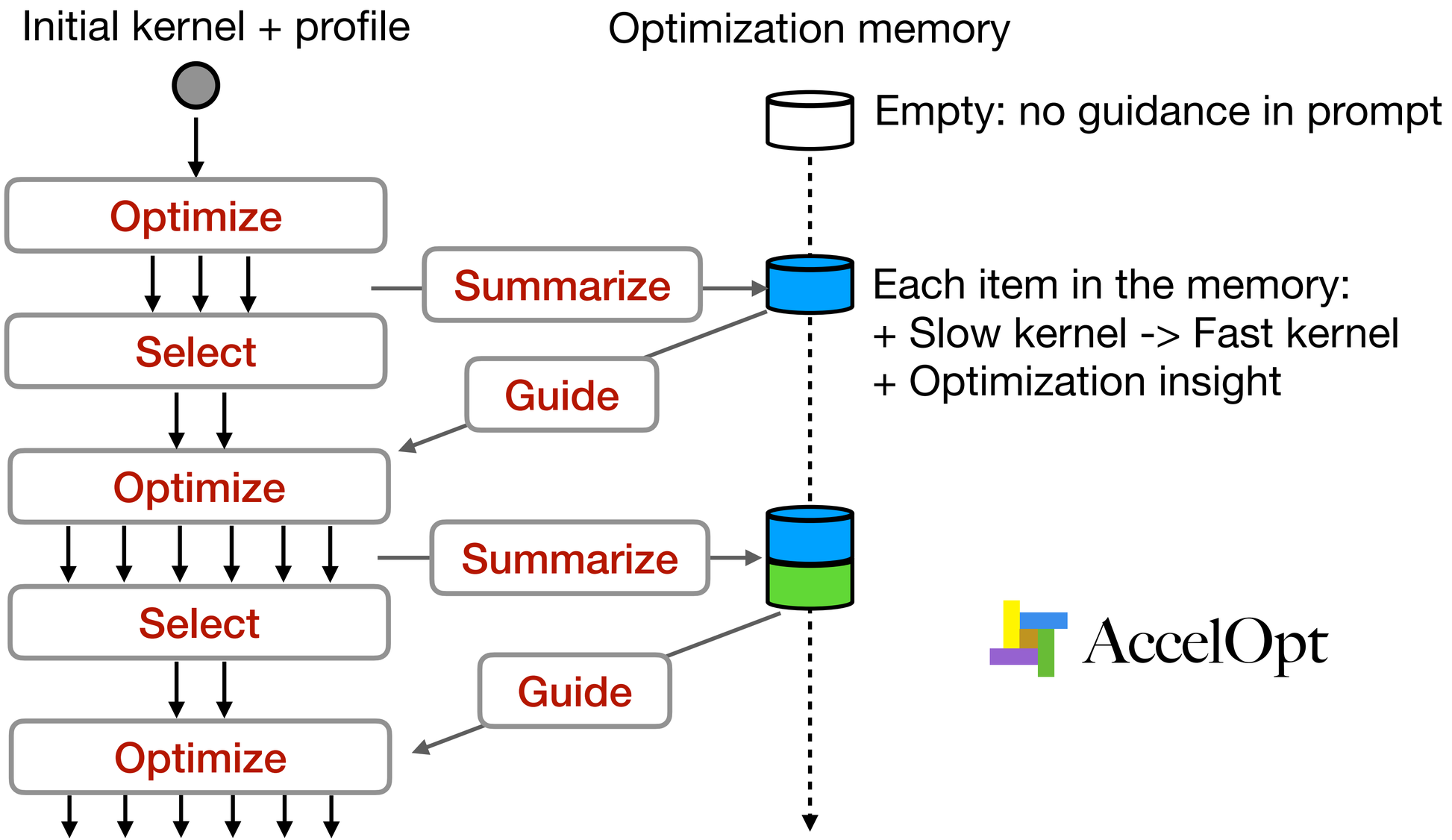
3. Guide beam search with optimization memory



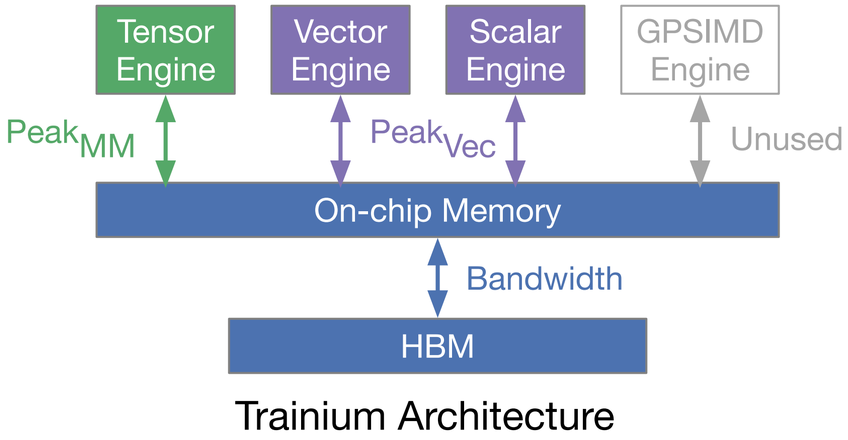
3. Guide beam search with optimization memory



3. Guide beam search with optimization memory

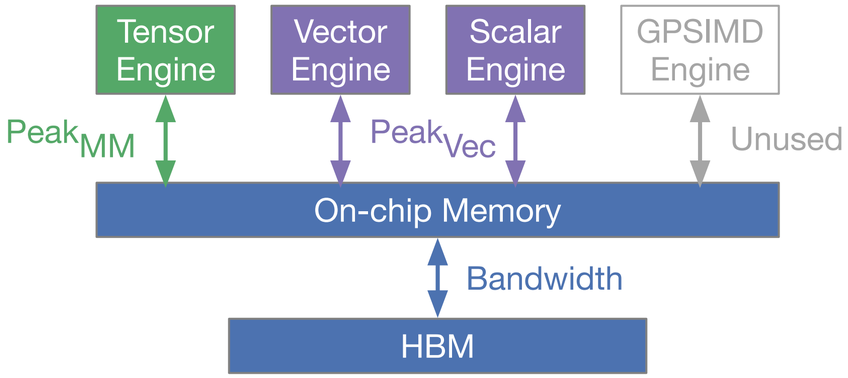


Metric: Percentage of Peak



Trainium Architecture

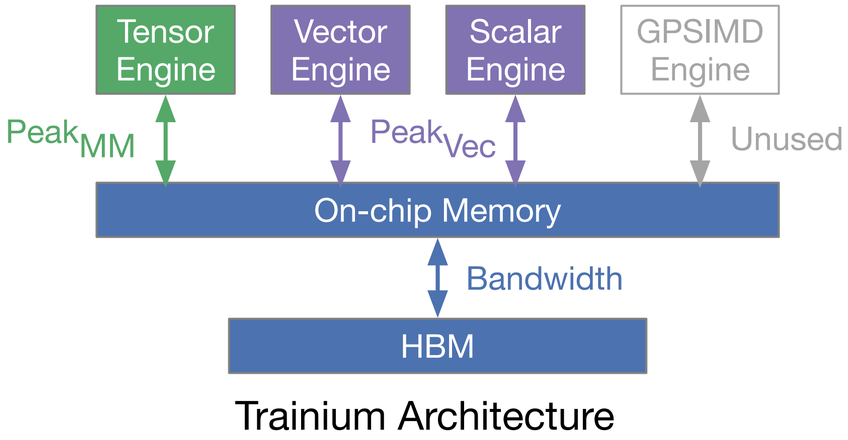
Metric: Percentage of Peak



Trainium Architecture

$$\textbf{Percentage of Peak} = \frac{T}{t_{real}}$$

Metric: Percentage of Peak

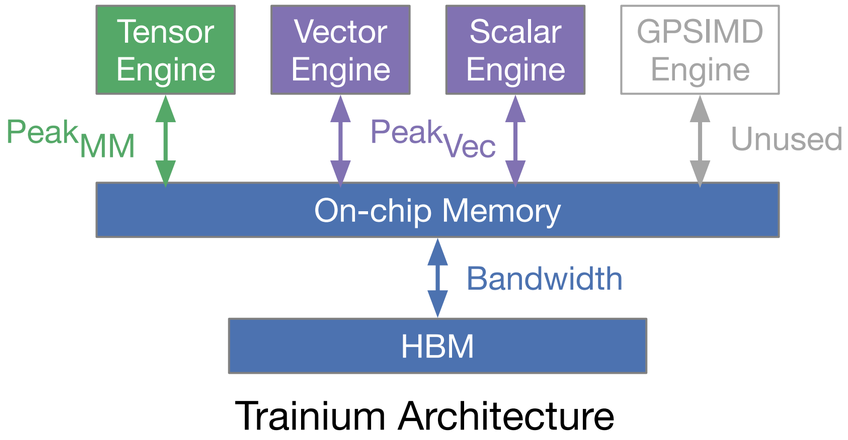


Trainium Architecture

$$\textbf{Percentage of Peak} = \frac{T}{t_{real}}$$

$$T = \max\left(\frac{\text{Traffic}_{\text{Min}}}{\text{Bandwidth}}, \frac{\text{FLOPs}_{\text{MM}}}{\text{Peak}_{\text{MM}}}, \frac{\text{FLOPs}_{\text{Vec}}}{\text{Peak}_{\text{Vec}}}\right)$$

Metric: Percentage of Peak



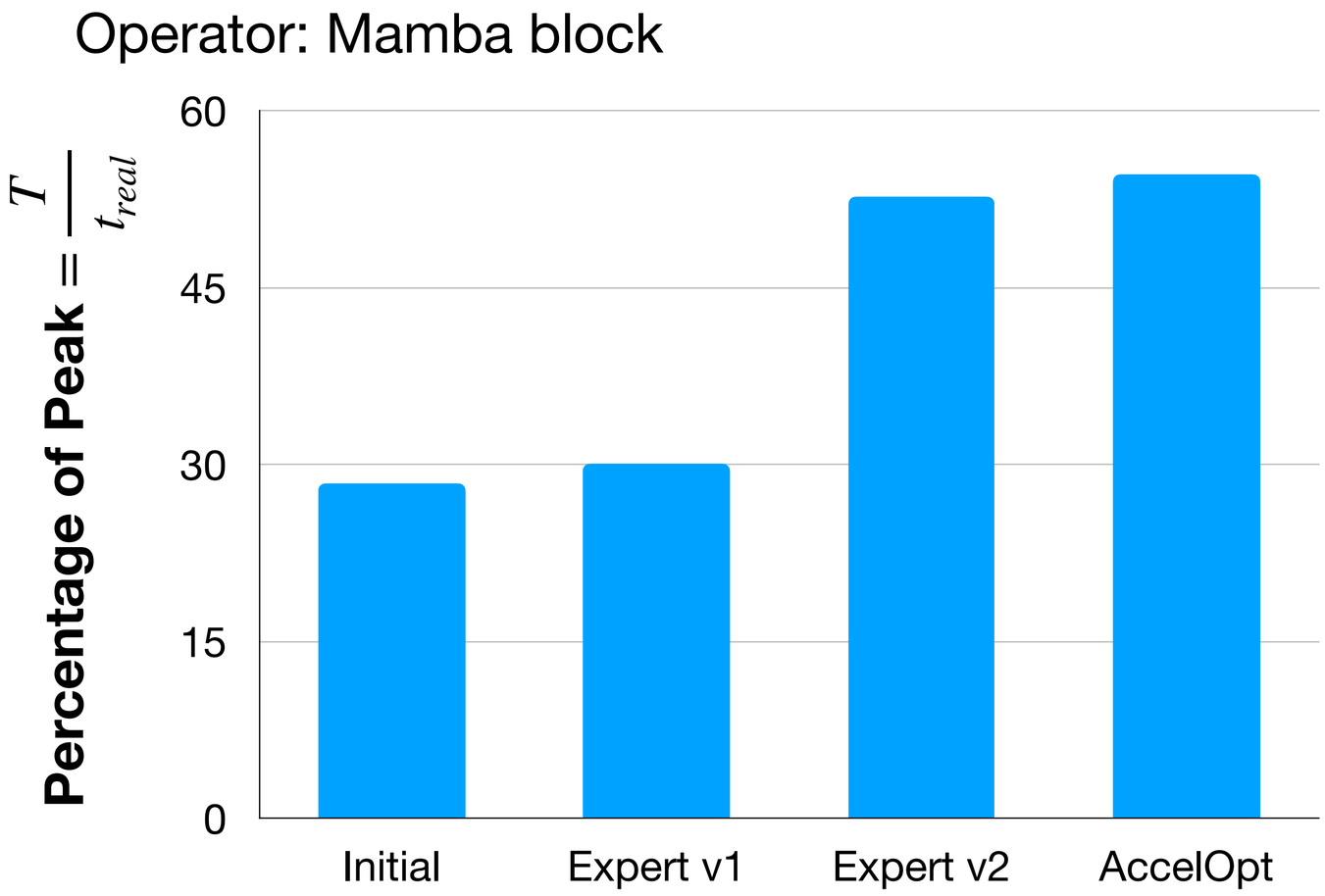
Trainium Architecture

$$\textbf{Percentage of Peak} = \frac{T}{t_{real}}$$

$$T = \max\left(\frac{\text{Traffic}_{\text{Min}}}{\text{Bandwidth}}, \frac{\text{FLOPs}_{\text{MM}}}{\text{Peak}_{\text{MM}}}, \frac{\text{FLOPs}_{\text{Vec}}}{\text{Peak}_{\text{Vec}}}\right)$$

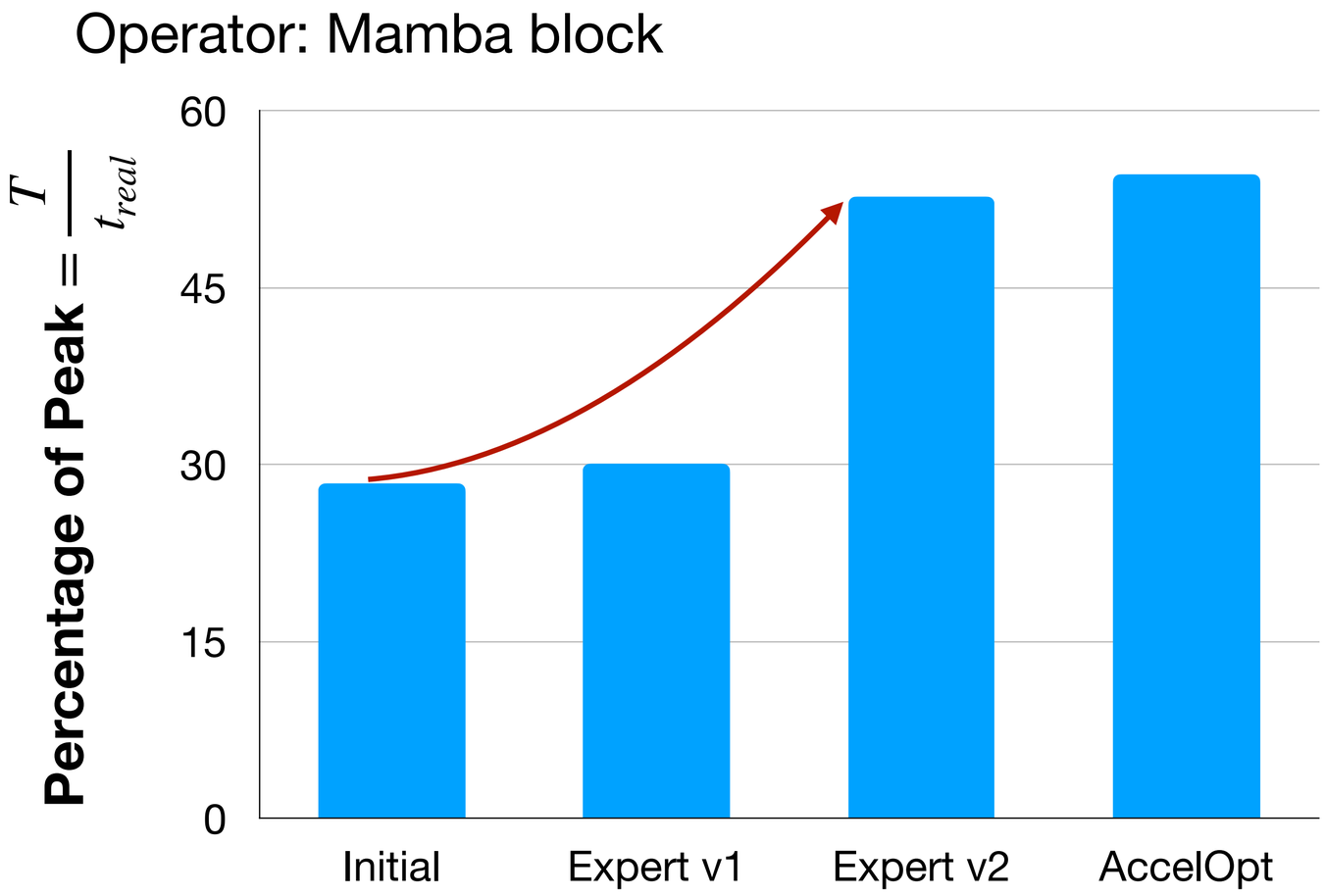
Percentage of peak provides an objective measurement of progress for kernel optimization agnostic to initial kernels.

AccelOpt can discover better kernels than human experts



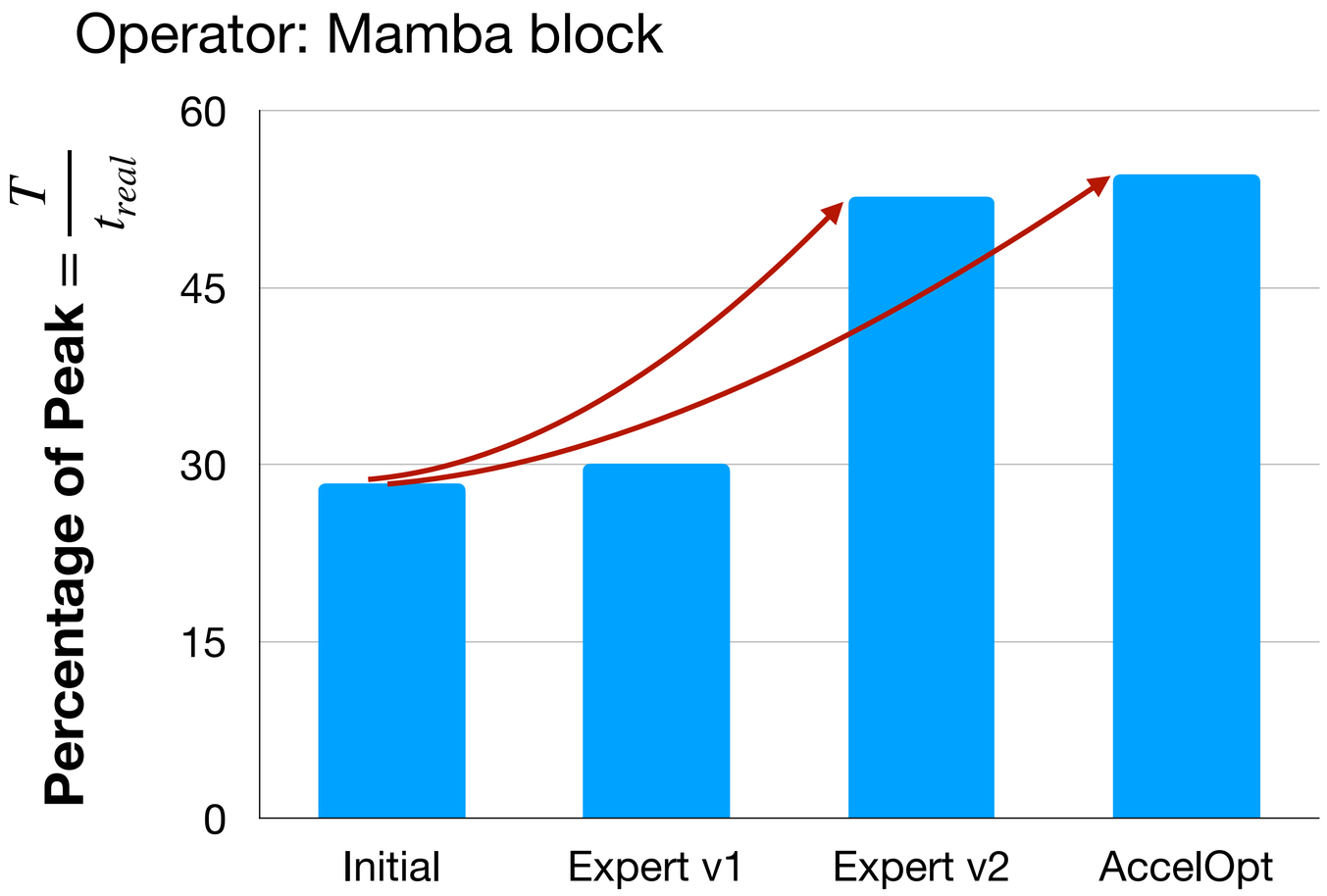
AccelOpt/samples/nki

AccelOpt can discover better kernels than human experts



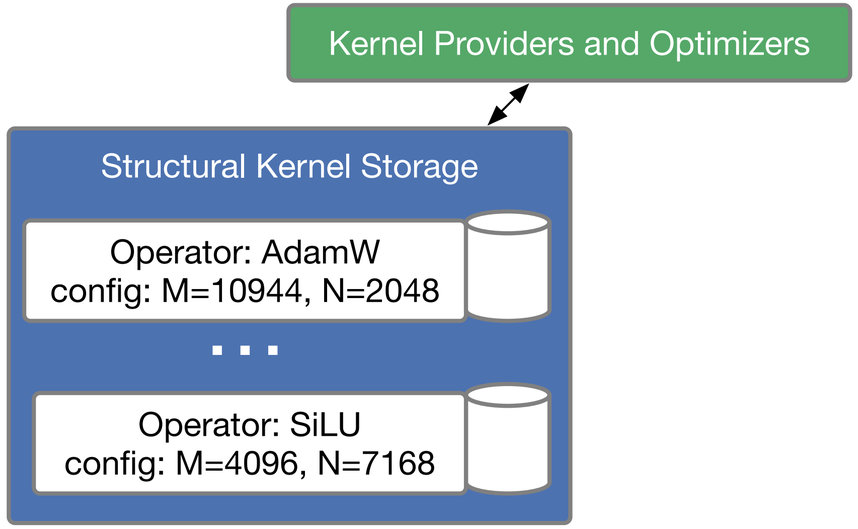
AccelOpt/samples/nki

AccelOpt can discover better kernels than human experts



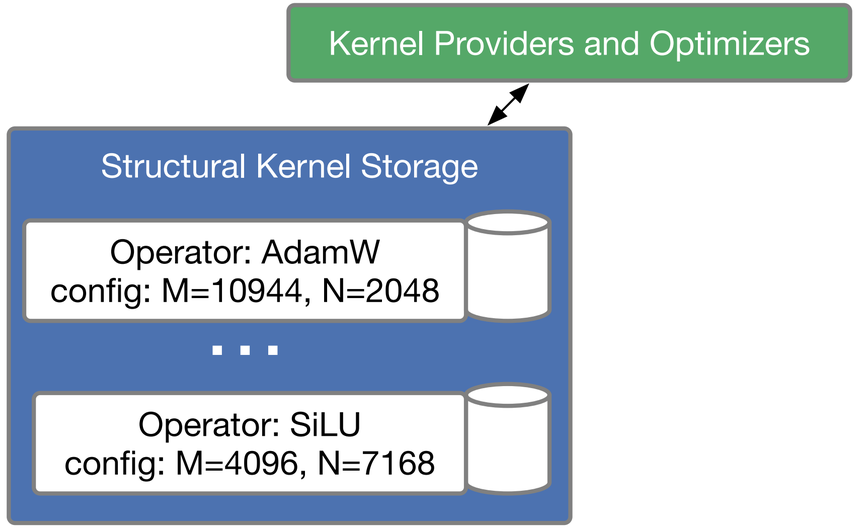
AccelOpt/samples/nki

NKIBench: a benchmark environment with realistic NKI kernels



Genghan/NKIBench

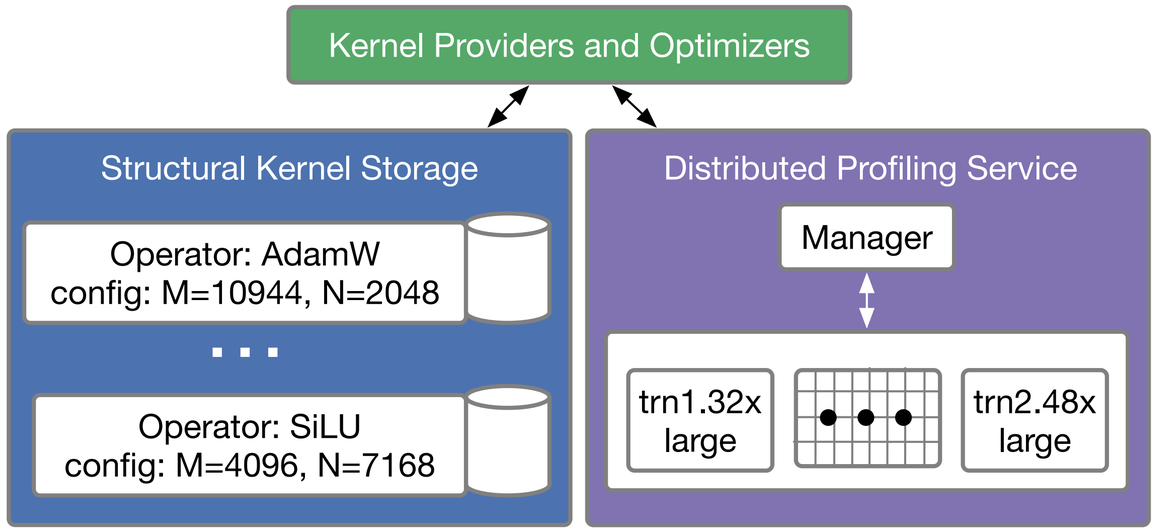
NKIBench: a benchmark environment with realistic NKI kernels



Name	Workload
AdamW	DeepSeek-MoE-16B
Add + RMSNorm + Matmul	Qwen3 0.6B
BatchMatmul	Falcon-40B
BatchMatmul + Softmax	Falcon-40B
Group Query Attention	Qwen3 0.6B/1.7B
LoRA	DeepSeek-V2.5
Mamba block	Synthesized
Matmul + Add + RMSNorm	Qwen3 1.7B
Matmul	DeepSeek-V2.5
RMSNorm + Matmul	Qwen3 0.6B
RoPE	Qwen3 32B
SiLU	DeepSeek-V3 671B
SwiGLU	Qwen3 0.6B
Transpose + Matmul	DeepSeek-MoE-16B

Genghan/NKIBench

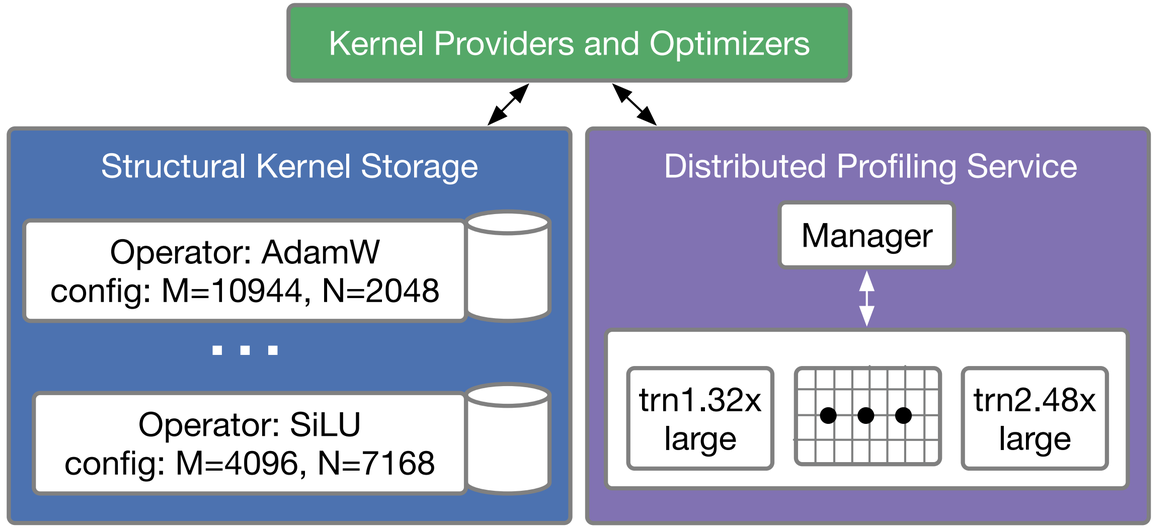
NKIBench: a benchmark environment with realistic NKI kernels



Name	Workload
AdamW	DeepSeek-MoE-16B
Add + RMSNorm + Matmul	Qwen3 0.6B
BatchMatmul	Falcon-40B
BatchMatmul + Softmax	Falcon-40B
Group Query Attention	Qwen3 0.6B/1.7B
LoRA	DeepSeek-V2.5
Mamba block	Synthesized
Matmul + Add + RMSNorm	Qwen3 1.7B
Matmul	DeepSeek-V2.5
RMSNorm + Matmul	Qwen3 0.6B
RoPE	Qwen3 32B
SiLU	DeepSeek-V3 671B
SwiGLU	Qwen3 0.6B
Transpose + Matmul	DeepSeek-MoE-16B

Genghan/NKIBench

NKIBench: a benchmark environment with realistic NKI kernels

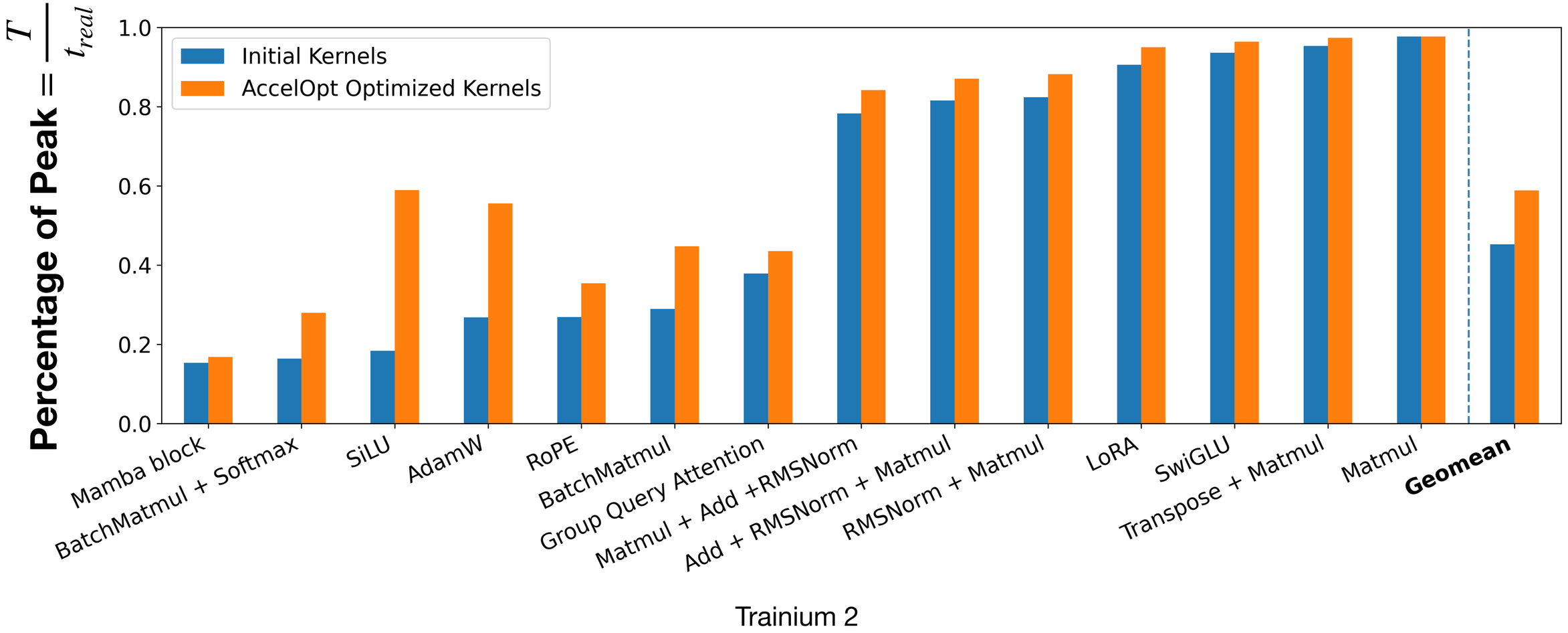


Name	Workload
AdamW	DeepSeek-MoE-16B
Add + RMSNorm + Matmul	Qwen3 0.6B
BatchMatmul	Falcon-40B
BatchMatmul + Softmax	Falcon-40B
Group Query Attention	Qwen3 0.6B/1.7B
LoRA	DeepSeek-V2.5
Mamba block	Synthesized
Matmul + Add + RMSNorm	Qwen3 1.7B
Matmul	DeepSeek-V2.5
RMSNorm + Matmul	Qwen3 0.6B
RoPE	Qwen3 32B
SiLU	DeepSeek-V3 671B
SwiGLU	Qwen3 0.6B
Transpose + Matmul	DeepSeek-MoE-16B

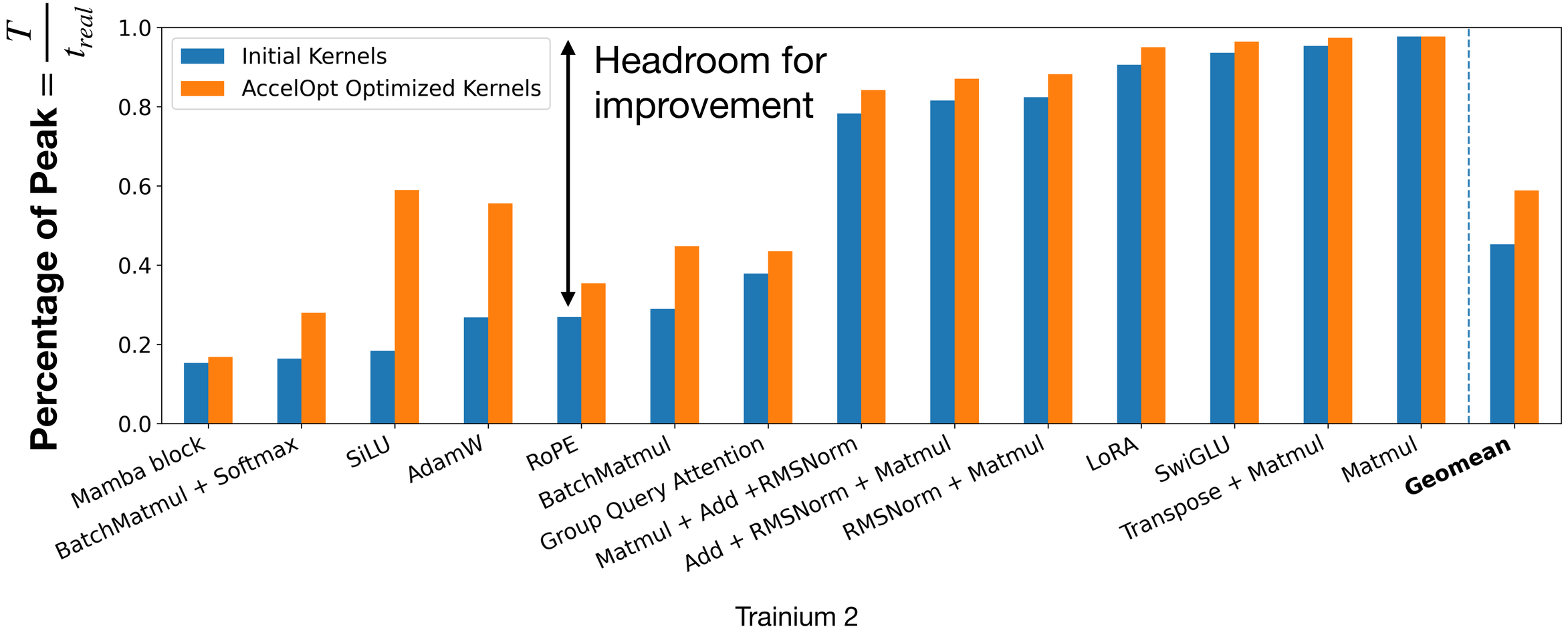
Genghan/NKIBench

AWS is working with the open-source community to maintain and expand NKIBench.

AccelOpt can optimize a broad range of NKI kernels

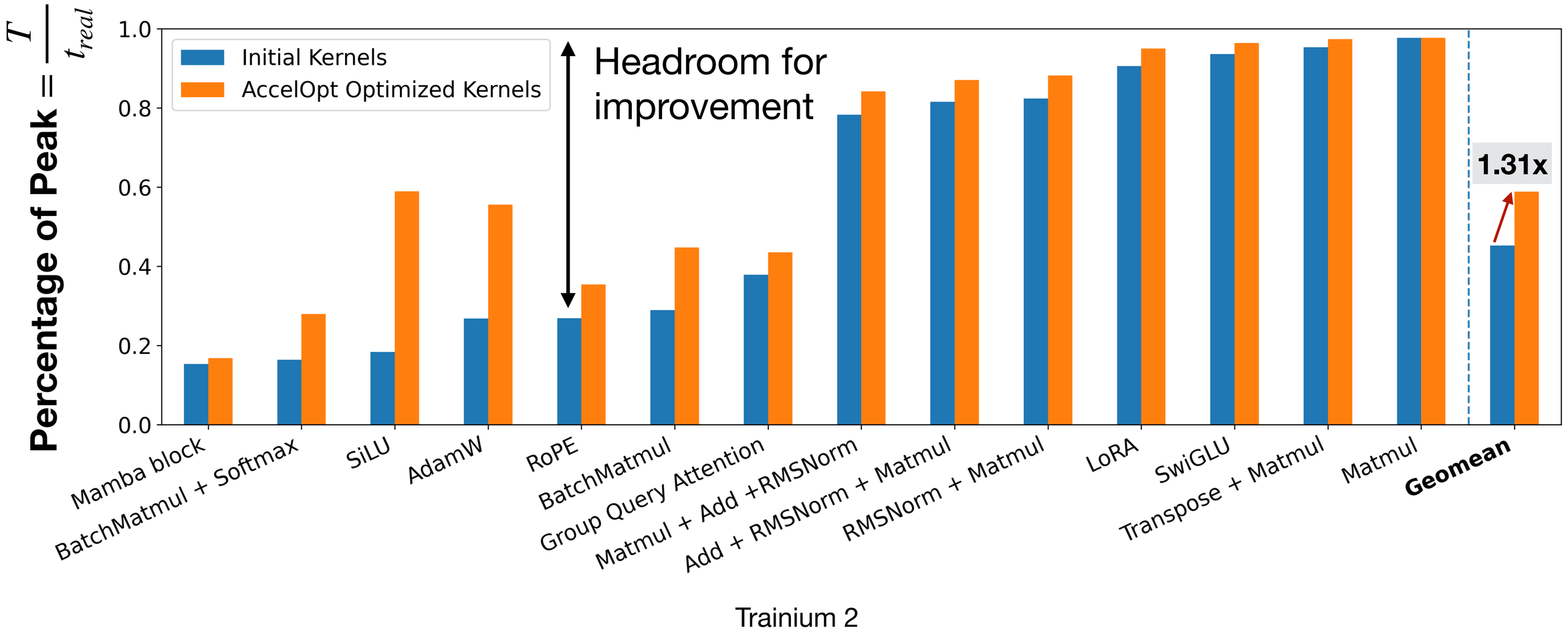


AccelOpt can optimize a broad range of NKI kernels



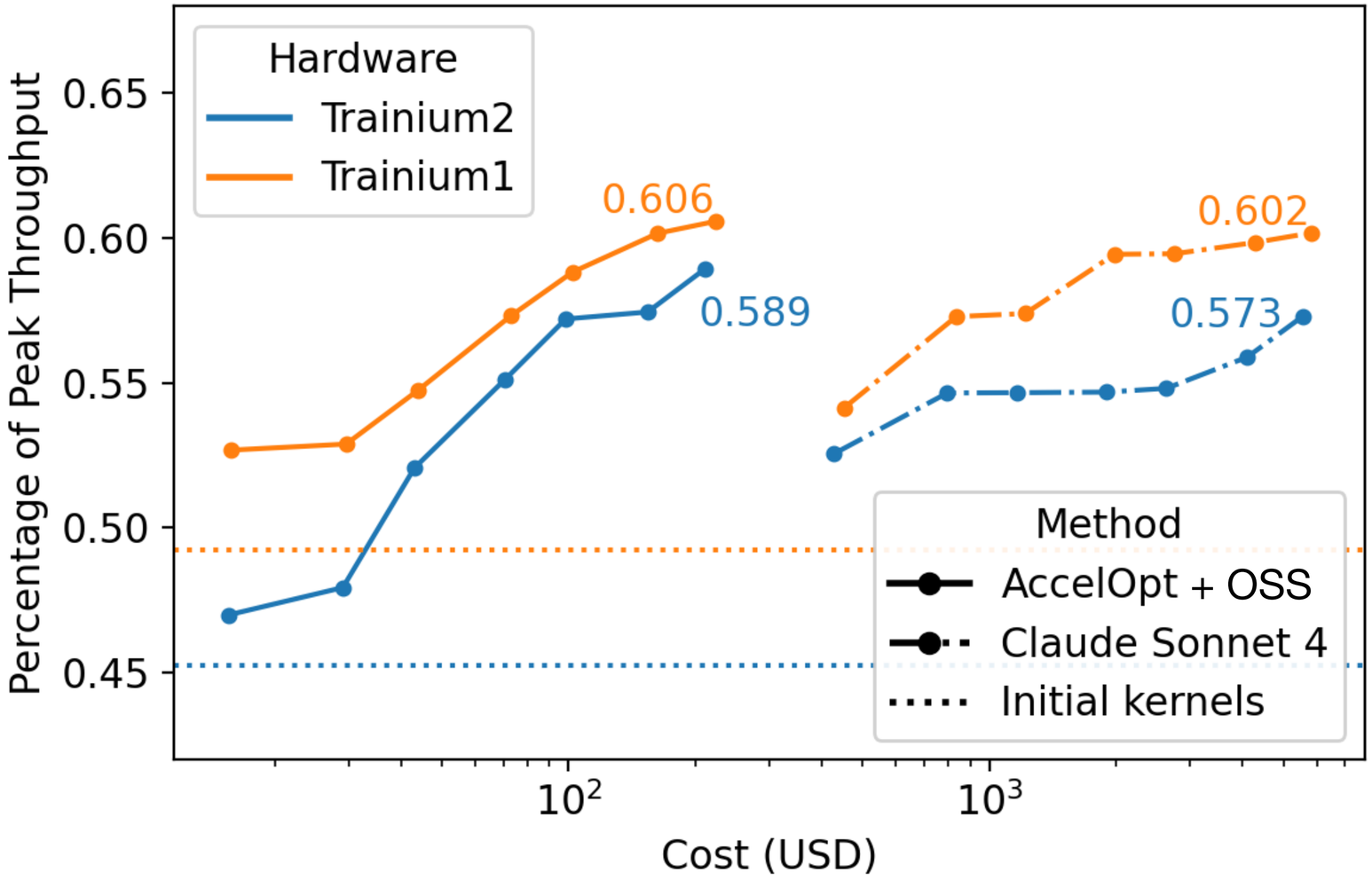
Trainium 2

AccelOpt can optimize a broad range of NKI kernels

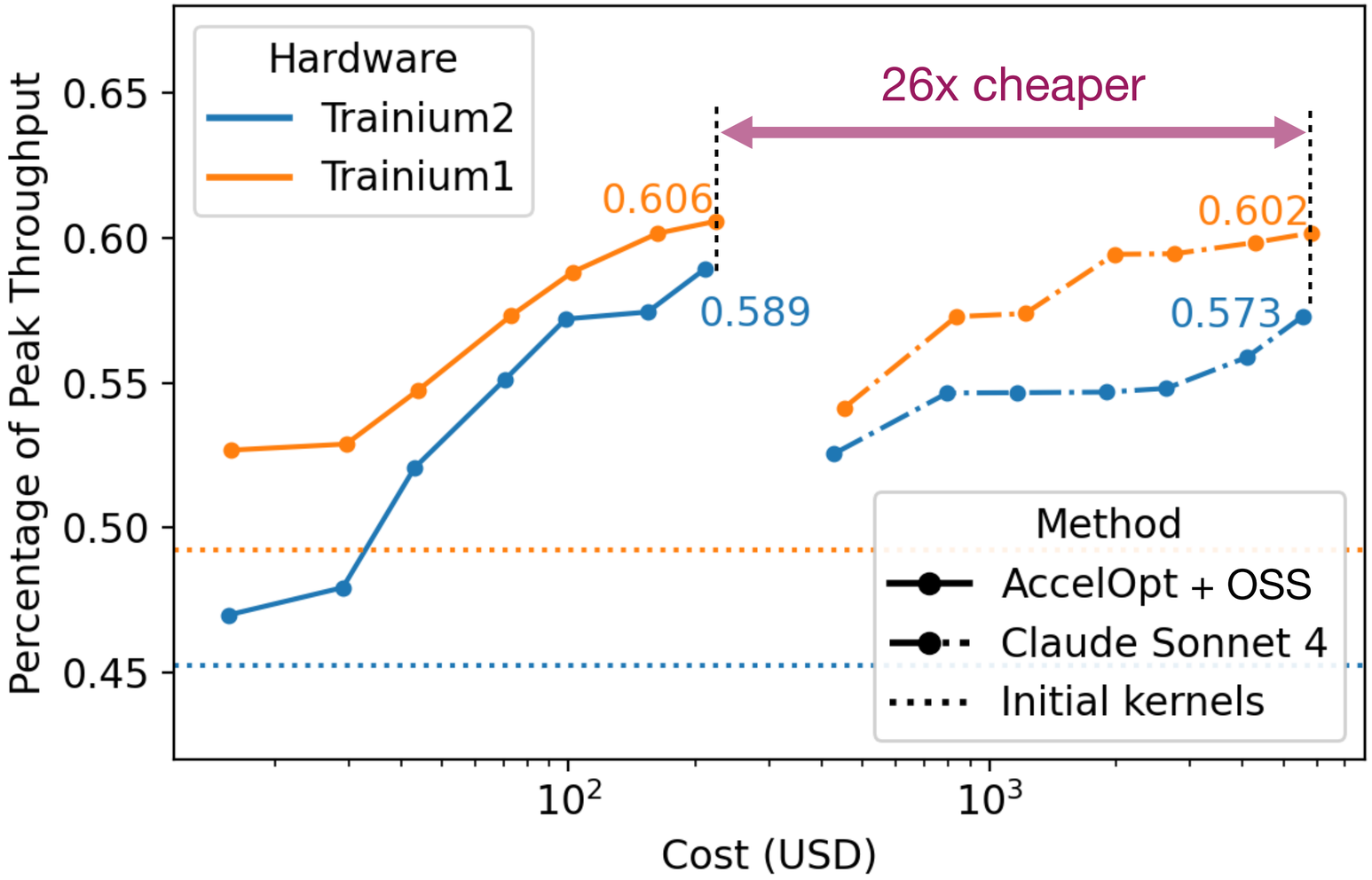


Trainium 2

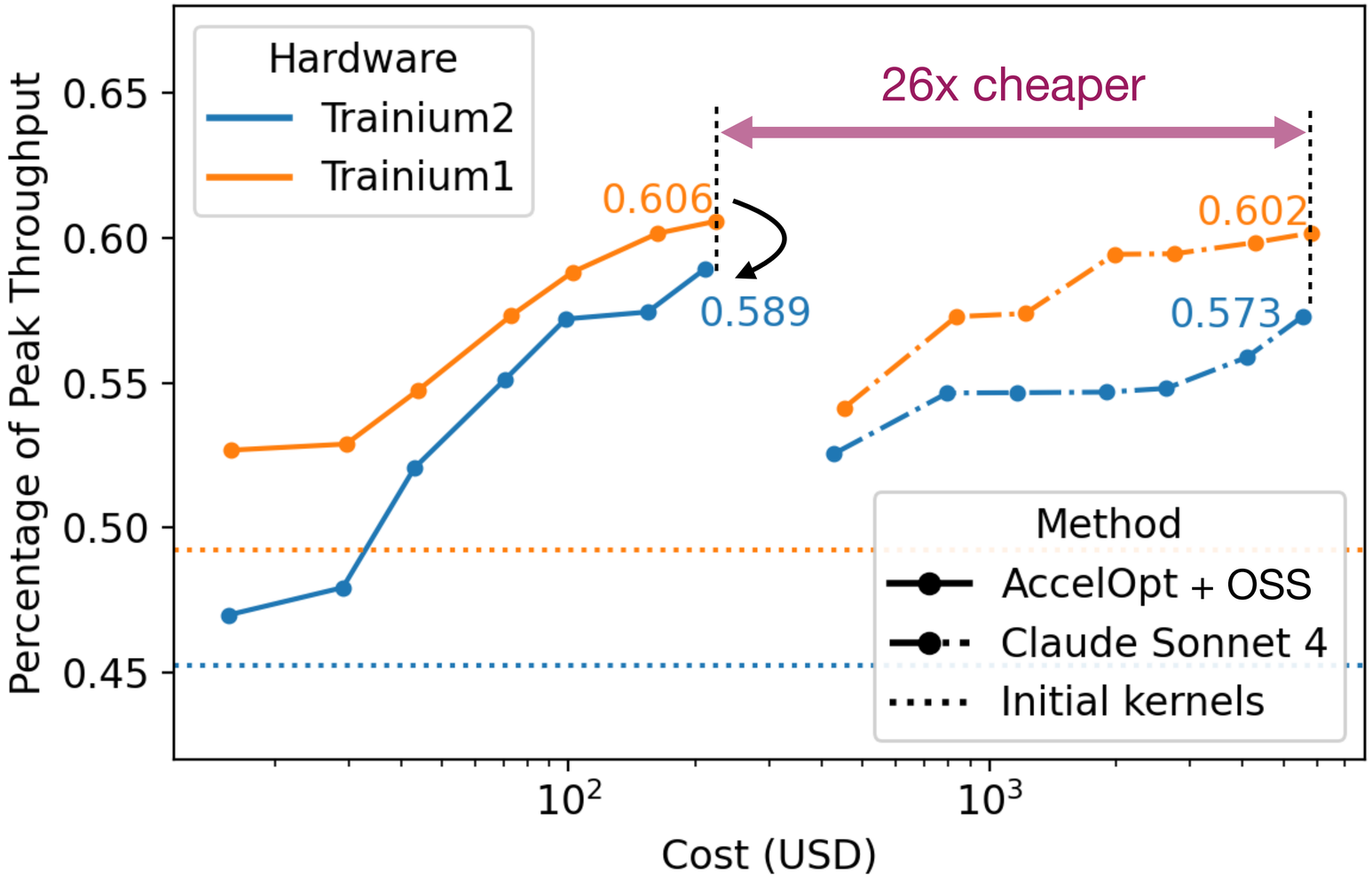
AccelOpt is cost efficient and adaptive to architectural change



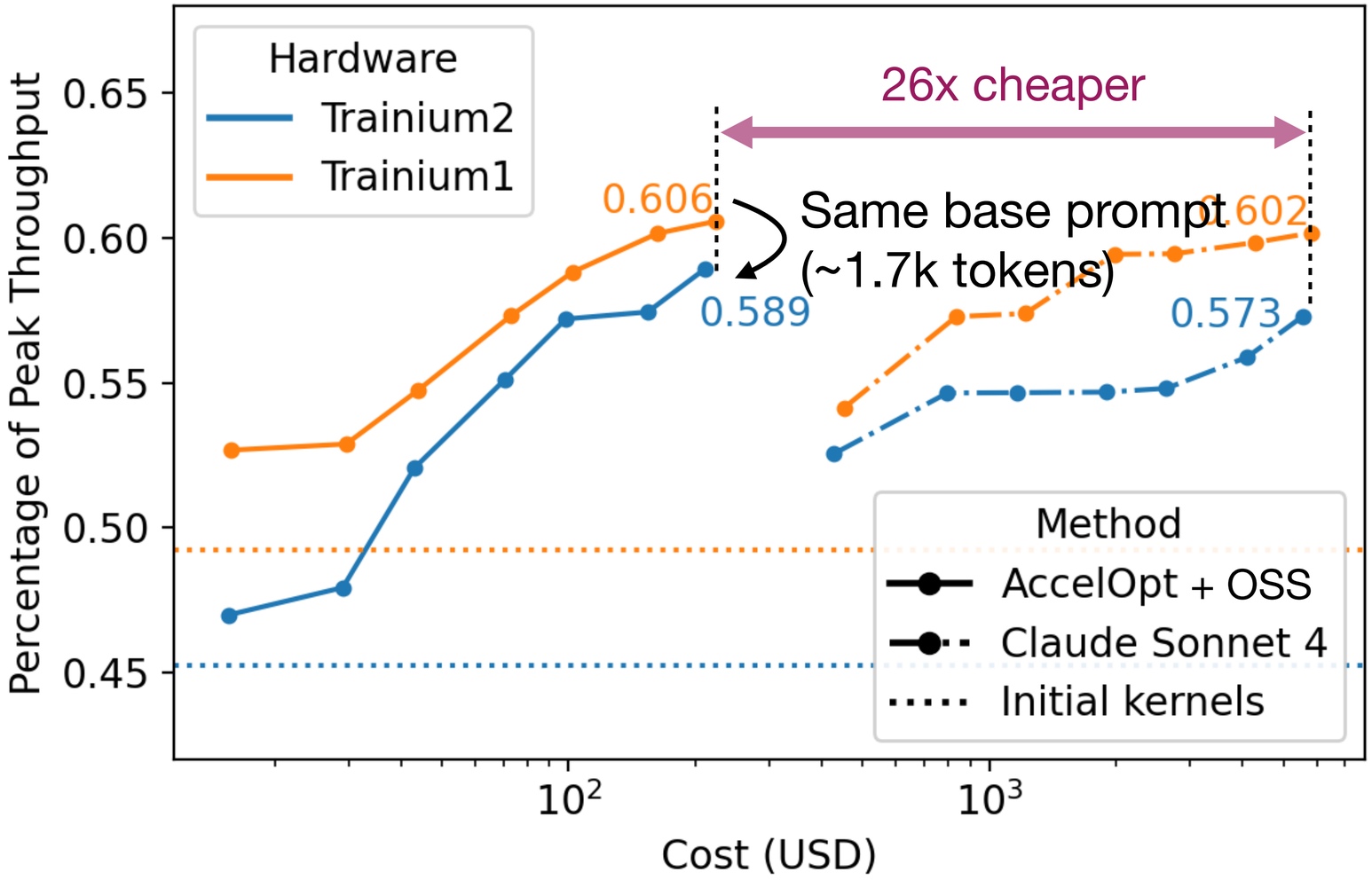
AccelOpt is cost efficient and adaptive to architectural change



AccelOpt is cost efficient and adaptive to architectural change



AccelOpt is cost efficient and adaptive to architectural change



AccelOpt can be applied to any AI accelerator

Operator: Paged Grouped Query Attention on H100
Performance baseline: FlashInfer v0.5.3

Workload*	Best Triton baseline Flashinfer-Bench	AccelOpt + Gemini 3 Flash
Qwen3-30B-A3B Decode	0.26x	**1.50x**
Llama-3.1-8B Decode	0.43x	**1.08x**
Qwen3-30B-A3B Prefill	0.12x	**1.38x**
Llama-3.1-8B Prefill	0.04x	0.10x

Ye Z., et al. Flashinfer: Efficient and customizable attention engine for llm inference serving *MLSys 2025*
Xing S., Zhai Y., Jiang A., Dong Y., et al. FlashInfer-Bench: Building the Virtuous Cycle for AI-driven LLM Systems *MLSys 2026*

*Use large workloads: best Triton baseline > 0.1ms from FlashInfer-Bench

AccelOpt can be applied to any AI accelerator

Operator: Paged Grouped Query Attention on H100
Performance baseline: FlashInfer v0.5.3

Workload*	Best Triton baseline Flashinfer-Bench		AccelOpt + Gemini 3 Flash
Qwen3-30B-A3B Decode	0.26x	**5.77x** →	**1.50x**
Llama-3.1-8B Decode	0.43x	**2.51x** →	**1.08x**
Qwen3-30B-A3B Prefill	0.12x	**11.5x** →	**1.38x**
Llama-3.1-8B Prefill	0.04x	**2.5x** →	0.10x

Ye Z., et al. Flashinfer: Efficient and customizable attention engine for llm inference serving *MLSys 2025*
Xing S., Zhai Y., Jiang A., Dong Y., et al. FlashInfer-Bench: Building the Virtuous Cycle for AI-driven LLM Systems *MLSys 2026*

*Use large workloads: best Triton baseline > 0.1ms from FlashInfer-Bench

AccelOpt can be applied to any AI accelerator

Operator: Paged Grouped Query Attention on H100
Performance baseline: FlashInfer v0.5.3

Workload*	Best Triton baseline Flashinfer-Bench		AccelOpt + Gemini 3 Flash
Qwen3-30B-A3B Decode	0.26x	**5.77x** →	**1.50x**
Llama-3.1-8B Decode	0.43x	**2.51x** →	**1.08x**
Qwen3-30B-A3B Prefill	0.12x	**11.5x** →	**1.38x**
Llama-3.1-8B Prefill	0.04x	**2.5x** →	0.10x

AccelOpt/samples/fib

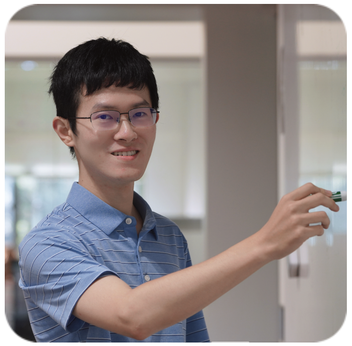
Ye Z., et al. Flashinfer: Efficient and customizable attention engine for llm inference serving *MLSys 2025*
Xing S., Zhai Y., Jiang A., Dong Y., et al. FlashInfer-Bench: Building the Virtuous Cycle for AI-driven LLM Systems *MLSys 2026*

*Use large workloads: best Triton baseline > 0.1ms from FlashInfer-Bench

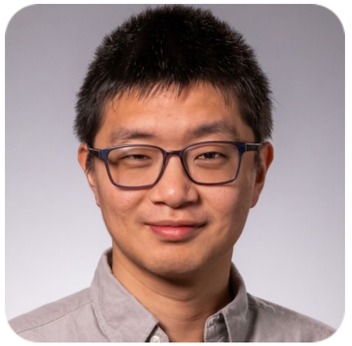
Collaborators

Genghan Zhang

Shaowei Zhu



Anjiang Wei



Zhenyu Song

Allen Nie

Zhen Jia

Nandita Vijaykumar

Yida Wang

Kunle Olukotun

Stanford University

aws

UNIVERSITY OF TORONTO

AccelOpt: A Self-improving LLM Agentic System for AI Accelerator Kernel Optimization

stanford-ppl/AccelOpt

Genghan/NKIBench

zhang677/AccelOpt

AccelOpt: A Self-improving LLM Agentic System for AI Accelerator Kernel Optimization

1. Combining beam search with optimization memory, AccelOpt enables LLM agents to autonomously optimize real-world Trainium kernels in NKIBench, our curated NKI kernel environment, without requiring expert optimization knowledge.

stanford-ppl/AccelOpt

Genghan/NKIBench

zhang677/AccelOpt

AccelOpt: A Self-improving LLM Agentic System for AI Accelerator Kernel Optimization

1. Combining beam search with optimization memory, AccelOpt enables LLM agents to autonomously optimize real-world Trainium kernels in NKIBench, our curated NKI kernel environment, without requiring expert optimization knowledge.

2. With AccelOpt, open-source models can achieve higher cost efficiency than leading proprietary coding models for kernel optimization.

stanford-ppl/AccelOpt

Genghan/NKIBench

zhang677/AccelOpt

AccelOpt: A Self-improving LLM Agentic System for AI Accelerator Kernel Optimization

1. Combining beam search with optimization memory, AccelOpt enables LLM agents to autonomously optimize real-world Trainium kernels in NKIBench, our curated NKI kernel environment, without requiring expert optimization knowledge.

2. With AccelOpt, open-source models can achieve higher cost efficiency than leading proprietary coding models for kernel optimization.

3. AccelOpt and NKIBench provide a promising foundation for automated kernel optimization on emerging AI accelerators.

stanford-ppl/AccelOpt

Genghan/NKIBench

zhang677/AccelOpt