

Declarative Optimization for AI Workloads

Michael Cafarella
MIT CSAIL

March 31, 2025



MIT AI Hardware Program



School of
Engineering



MIT Schwarzman
College of Computing

AI Models are Full of Promise..

- Chat is fun, but foundation models are incredible potential building blocks for apps that fluidly mix AI and data processing



...And Are Still Underexploited

- Chat is fun, but foundation models are incredible potential building blocks for apps that fluidly mix AI and data processing

Data Integration	Multimodal Document Compliance
Data Cleaning	Next-Generation Dashboards
Information Extraction	Log-Driven System Diagnosis
Long Document Understanding	Data-Driven Digital Twins
Multimodal Scientific Discovery	... and many others

- All of these have traditionally been very difficult to engineer



AI Programs are Thrilling...

Such as:

“Find all the materials science papers that talk about EV batteries”

“Double-check all the facts in this mortgage application”

“Find all US banks’ SEC filings in 2022 and extract footnotes that talk about solvency”

“Extract a video of the winning touchdown from every Super Bowl”



...but AI Programming is a Drag



...but AI Programming is a Drag

- Make it fast, cheap, and high quality



...but AI Programming is a Drag

- Make it fast, cheap, and high quality
- While models, GPUs, and AI methods change every day



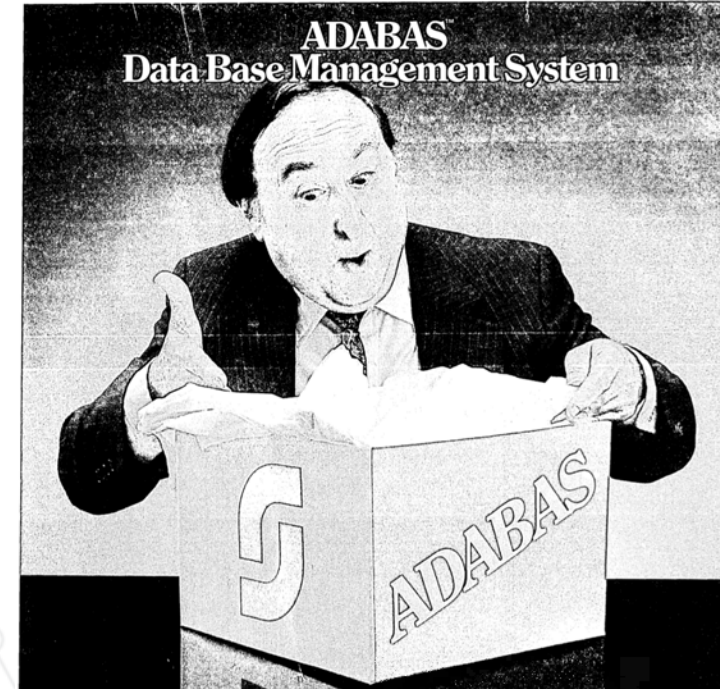
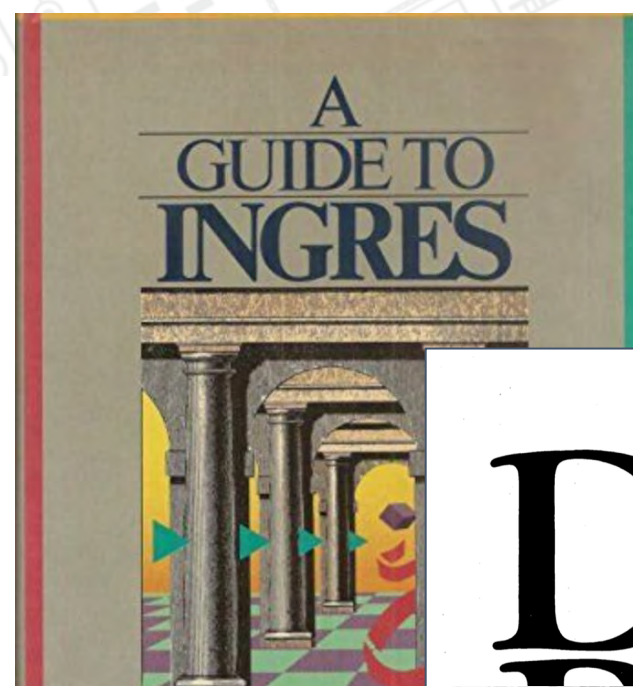
...but AI Programming is a Drag

- Make it fast, cheap, and high quality
- While models, GPUs, and AI methods change every day
- While project needs change over time



The Good News

- We've solved a problem like this before!
- In the mid-1970s, database programmers had to write custom code for every query
- Declarative queries allowed them to write succinct programs while also obtaining good performance in a rapidly-changing technological environment
- Let's do the same for AI applications



DATA
BASE
AT.

IT, FROM THE LEADER
IN TRANSACTION PROCESSING.



Our System: Palimpzest

- Python package that lets users implement AI tasks in little code
- Behind the scenes, it hypothesizes and tests ways to use AI models to implement user's goal
- User specifies an optimization goal and constraints. E.g., "maximize quality subject to runtime < 10 seconds" and system attempts to implement it



<https://github.com/mitdbg/palimpzest>



Palimpzest Internals

Python Code

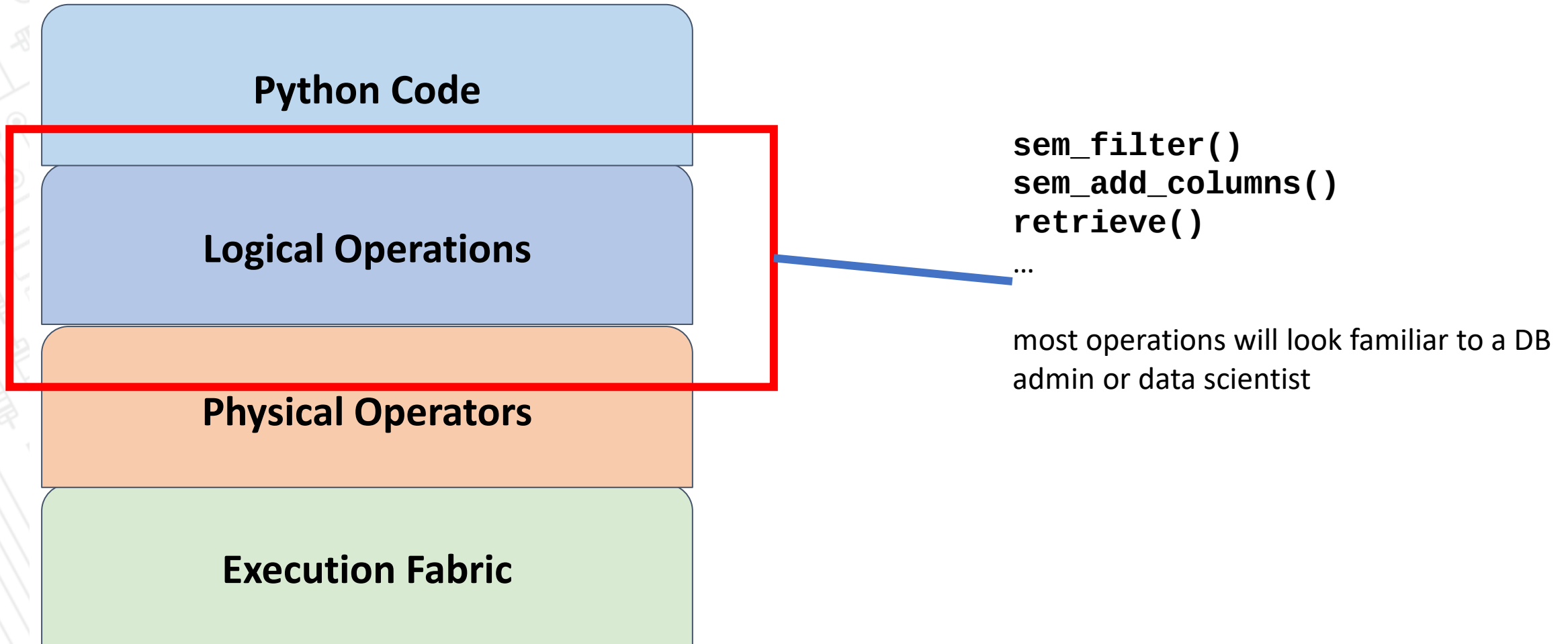
Logical Operations

Physical Operators

Execution Fabric



Palimpzest Internals



Palimpzest Internals

Python Code

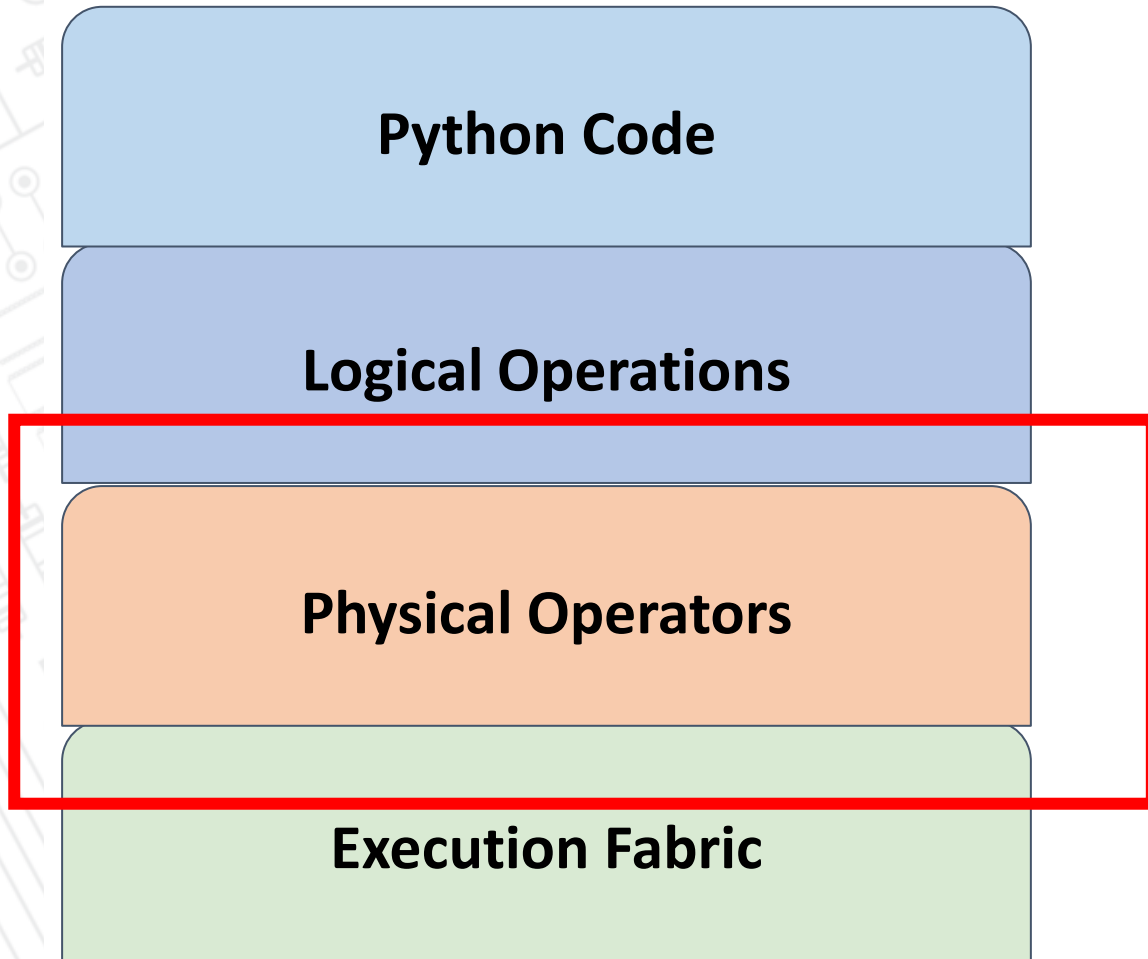
Logical Operations

Physical Operators

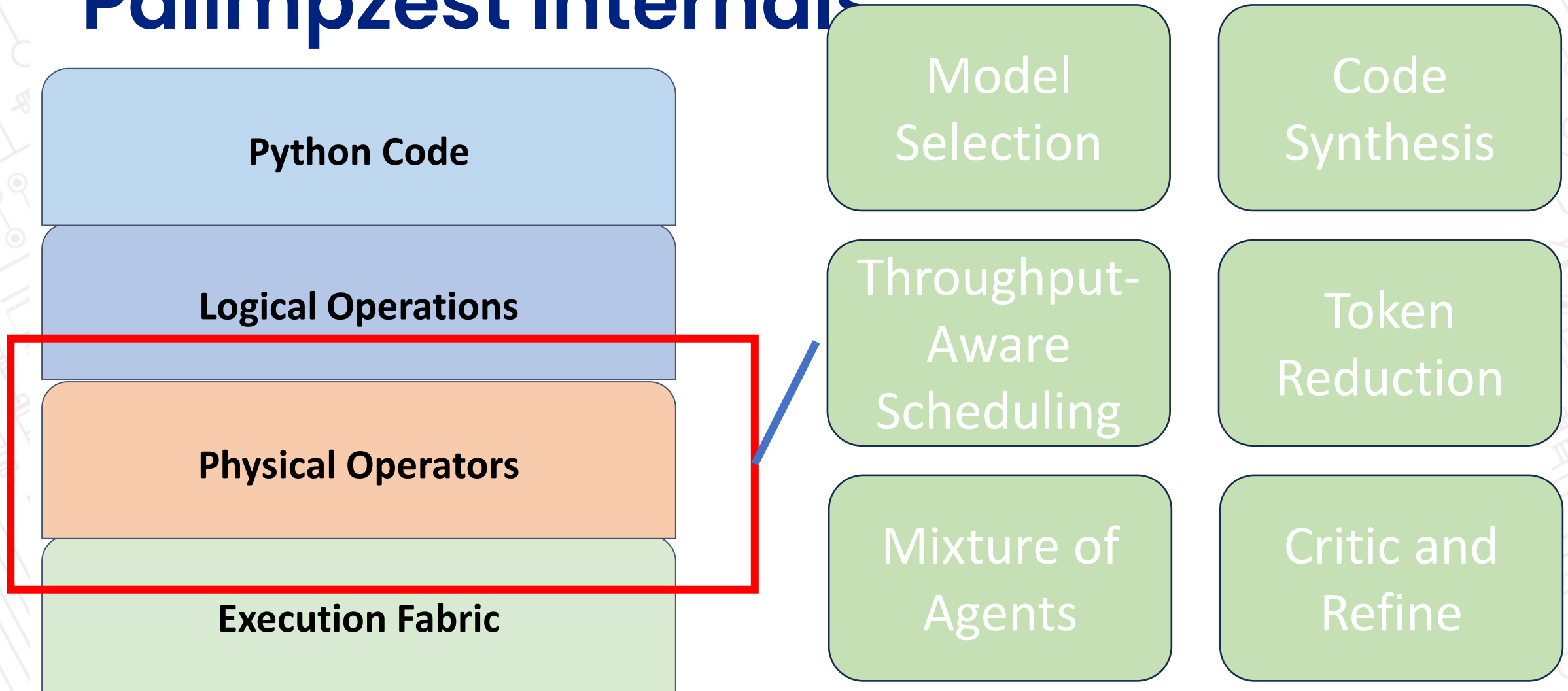
Execution Fabric



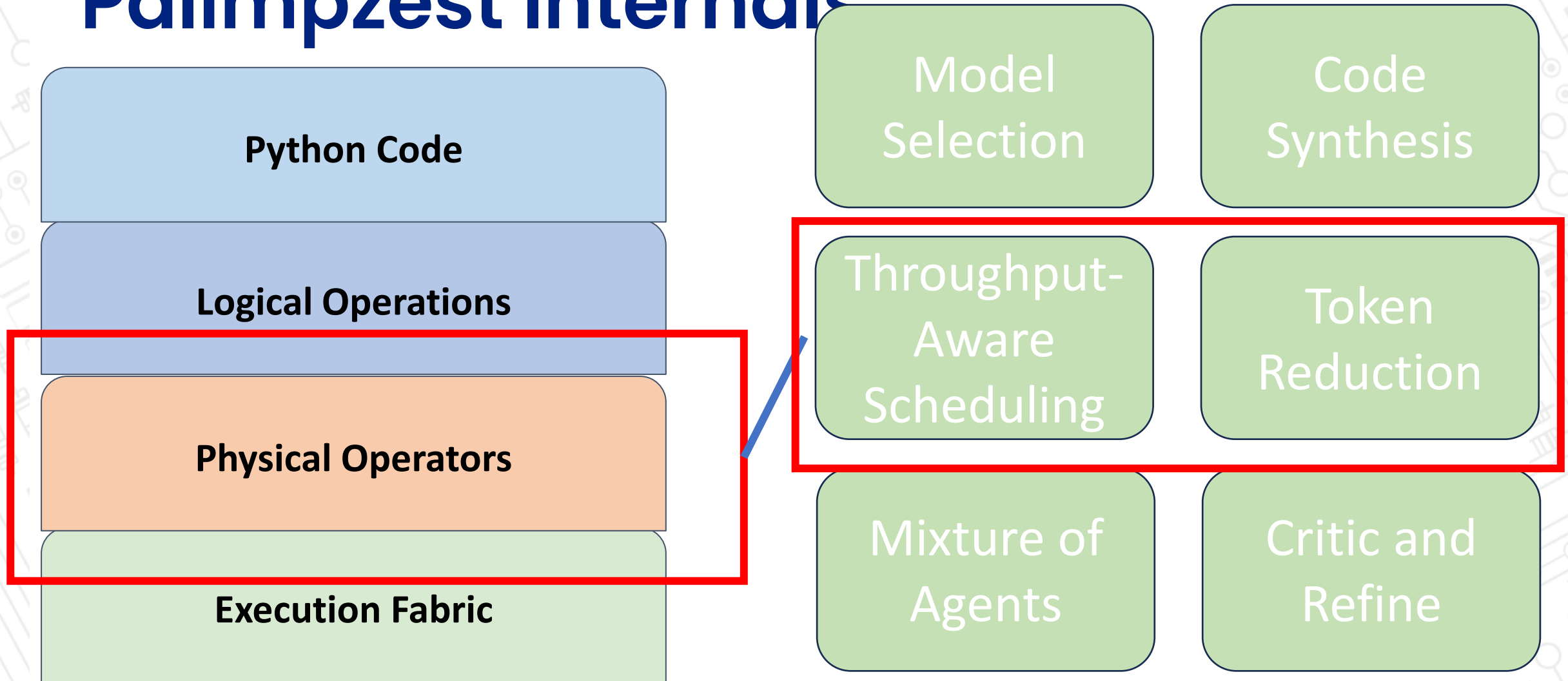
Palimpzest Internals



Palimpzest Internals



Palimpzest Internals



Palimpzest Internals

High-level information about user's AI workload should enable many novel hardware optimizations

Model

Code
Analysis

on

Mixture of
Agents

Critic and
Refine

Execution Fabric



BioDEX Classification Results

	Optimization Financial Cost	Plan Financial Cost	Total Financial Cost	# LLM Calls	Plan Runtime (s)	RP@5
LOTUS (as reported in March 1, 2025 paper release)	?	?	?	32,026	647	0.289
DocETL (as reported in 8 December, 2024 paper relase)	\$2.37	\$3.65	\$6.02	~12,000	463	0.281
Palimpzest	\$0.119	\$0.789	\$0.907	2,514	369	0.292

BioDEX comprises 250 biomedical papers, to be labeled with 24,300 possible labels.

To remove any unfair advantage, we disabled Palimpzest’s model choice optimization method for this experiment. It is limited to GPT-4o-mini.

Palimpzest user optimization goal is “MAX QUALITY”



The Students Who Did The Work



Matt Russo



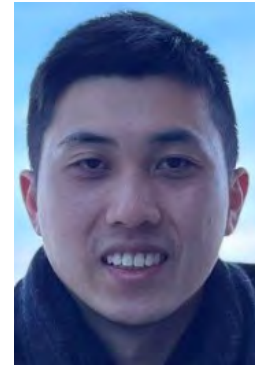
Chunwei Liu



Gerardo
Vitagliano



Sivaprasad
Sudhir



Peter Baille
Chen



Rana Shahout



Zui Chen



Thanks



Thanks to our other collaborators



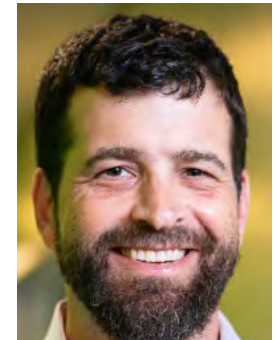
Lei Cao



Tim Kraska



Mike Franklin



Sam Madden



MIT AI Hardware Program



School of
Engineering



MIT Schwarzman
College of Computing