Architecture 2.0

Workshop on AI Computing Systems Design @ ASPLOS 2026

March 23, 2026

Workshop Overview

- Workshop Theme:
  - How do we apply/leverage AI to enhance, accelerate, and enable computer systems and their design
  - Focused on domains across the computing stack and the core disciplines of ASPLOS, spanning computer architecture, programming languages, and operating systems
- Submission Formats:
  - Early, WIP Work
  - Completed Research
  - Wacky Ideas, Opinions, and Proposals
- 13 Accepted Submissions presenting today
  - Acceptance decision made by 2 human reviewers without the help of AI
  - Review feedback included AI-generated reviews, more on that later

Workshop Talk Themes

- 4 Talk Sessions throughout the day:
  - Traditional ML / Lightweight Models
  - LLM Agents for Systems Part I
  - LLM Agents for Systems Part II
  - Wacky Ideas / Proposals
- Talks format:
  - 15 minutes total
  - Talks will be 10-12 minutes to leave time for QA + transition

Workshop Keynotes

- Keynote 1: Tushar Krishna, Georgia Tech
  - To achieve Arch 2.0, we must focus on the design and integration of architecture tools into AI-driven workflows. This talk will overview the design of architecture tools Chakra and AstraSim, and how they can be integrated to achieve AI-driven design. It will also explore how tools will prove essential to agentic workflows.
- Keynote 2: Yakun Sophia Shao, UC Berkeley
  - Redefining how we design systems to achieve Architecture 2.0: introducing new machine-understandable abstractions, building AI-native tools, and embracing system-level thinking in a world of increasingly specialized and heterogeneous architectures.

Workshop Schedule: Morning

8:00-8:15	Welcome and Logistics
8:15-9:00	Architecture 2.0 Introduction
9:00-9:15	Quick Break
9:15-10:00	Traditional ML / Lightweight Models
10:00-10:30	Coffee Break
10:30-11:15	Keynote Address 1
11:15-11:30	Quick Break
11:30-12:00	LLM Agents Part 1

Full Schedule:
harvard-edge.github.io/asplos-26-arch-2-workshop/

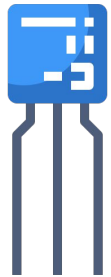
Workshop Schedule: Afternoon

12:00–1:30	Lunch
1:30–2:15	Keynote Address 2
2:15–2:30	Quick Break
2:30–3:30	LLM Agents Part 2
3:30–4:00	Coffee Break
4:00–5:00	Wacky Ideas / Proposals
5:00–5:15	Quick Break
5:15–6:00	Roundtable Discussion

Full Schedule:
harvard-edge.github.io/asplos-26-arch-2-workshop/

Architecture 2.0: Motivations and Background

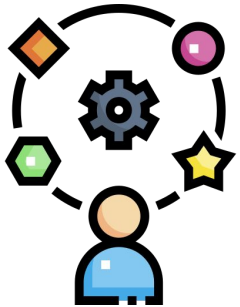
Innovation



Technology

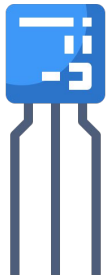
Architecture

Optimization



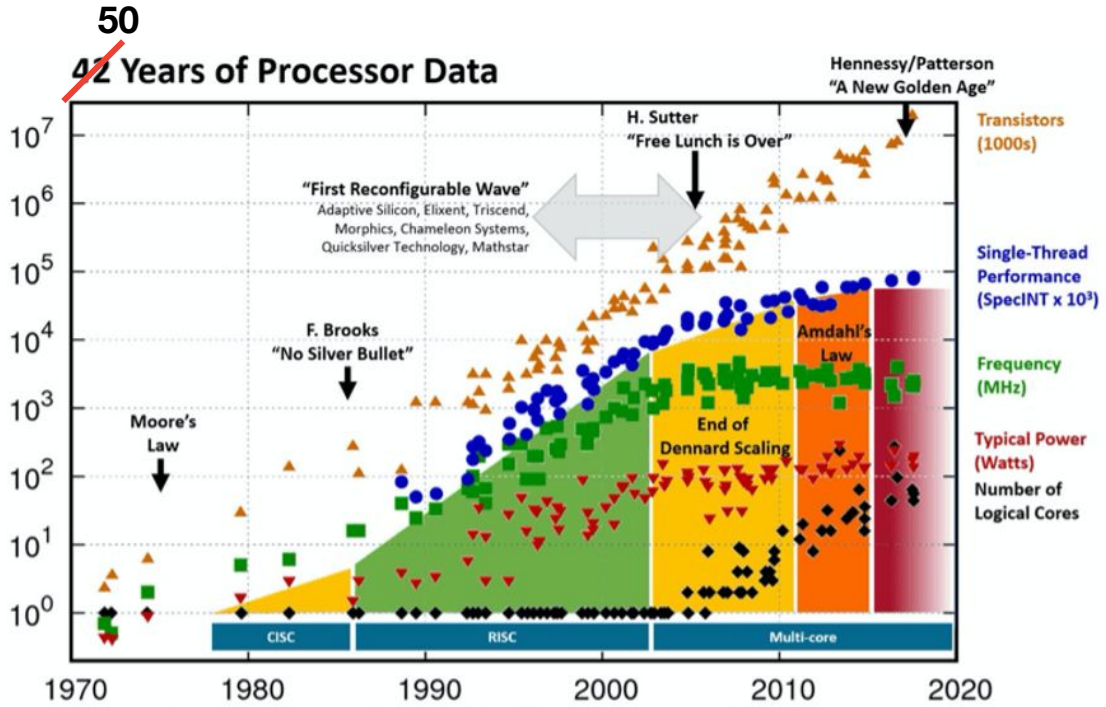
Specialization

Innovation

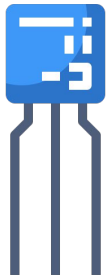


Technology

- Moore's law
- Dennard scaling
- Dark silicon



Innovation

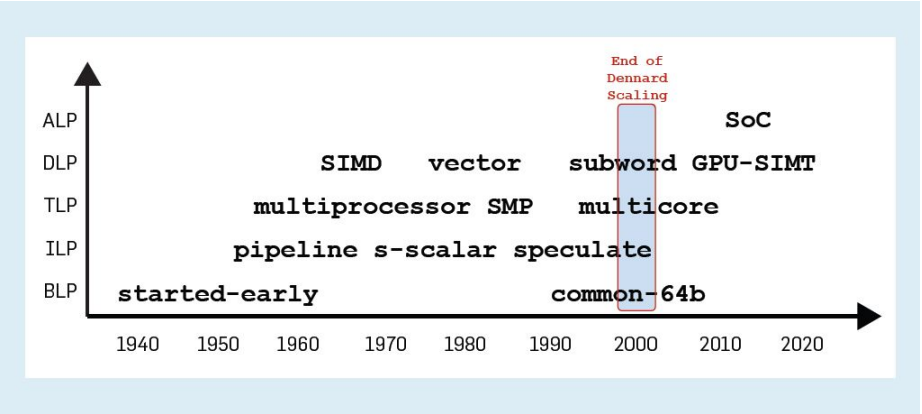


Technology

- Moore's law
- Dennard scaling
- Dark silicon

Architecture

- Hardware support for parallelism: BLP, ILP, DLP, TLP, ALP
- Memory system design
- System-on-chip



[Hill & Reddi, CACM'19]

DOI:10.1145/3460970

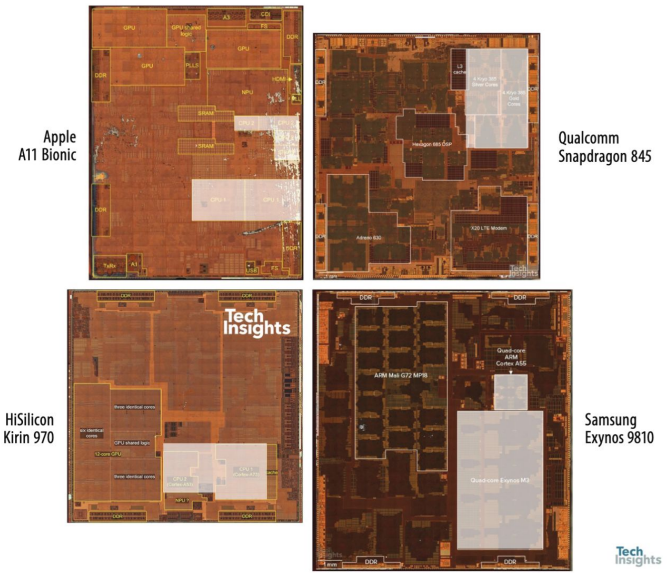
Mark D. Hill and Vijay Janapa Reddi

Viewpoint
Accelerator-Level Parallelism

Charging computer scientists to develop the science needed to best achieve the performance and cost goals of accelerator-level parallelism hardware and software.

WHILE PAST INFORMATION technology (IT) advances have transformed society, future advances hold great additional promise. For example, we have only just begun to reap the changes from artificial intelligence—especially machine learning—with profound advances expected in medicine, science, education, commerce, and government. All too often forgotten, underlying the IT impact are the dramatic improvements in the programmable hardware. Hardware improvements deliver performance that unlocks new capabilities. However, unlike in the 1990s and early 2000s, tomorrow's performance aspirations must be achieved with much less technological advancement (Moore's Law and Dennard Scaling). How then does one deliver AR/VR, self-driving vehicles, and health wearables at costs that enable great customer value?

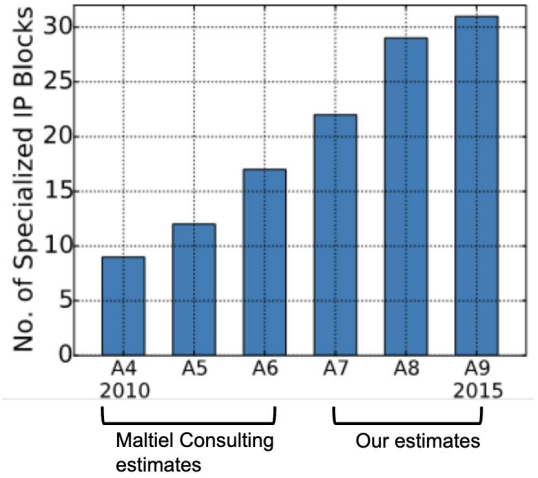
One approach that has emerged is to use **accelerators:** *hardware compo-*



Modern System-on-Chip (SoC) architectures. The CPUs in modern SoCs (shown in white) occupy only a small percentage of the die area. The rest of the SoC is committed to a potpourri of different accelerators, such as the DSP, GPU, ISP, NPU, video, and audio codecs.

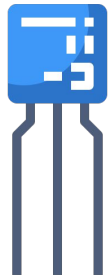
[Hill & Reddi, CACM'19]

< 30% of the die is dedicated to the CPU



[Shao, PhD. Thesis'16]

Innovation



Technology

- Moore's law
- Dennard scaling
- Dark silicon

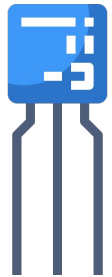
Architecture

- Hardware support for parallelism: BLP, ILP, DLP, TLP, ALP
- Memory system design
- System-on-chip

Optimization

- Application-level tuning
- Full-stack system optimization
- Hardware-Software co-design

Innovation



Technology

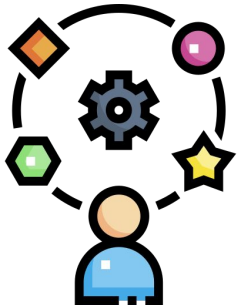
- Moore's law
- Dennard scaling
- Dark silicon

Architecture

- Hardware support for parallelism: BLP, ILP, DLP, TLP, ALP
- Memory system design
- System-on-chip

Optimization

- Application-level tuning
- Full-stack system optimization
- Hardware-Software co-design

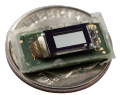


Specialization

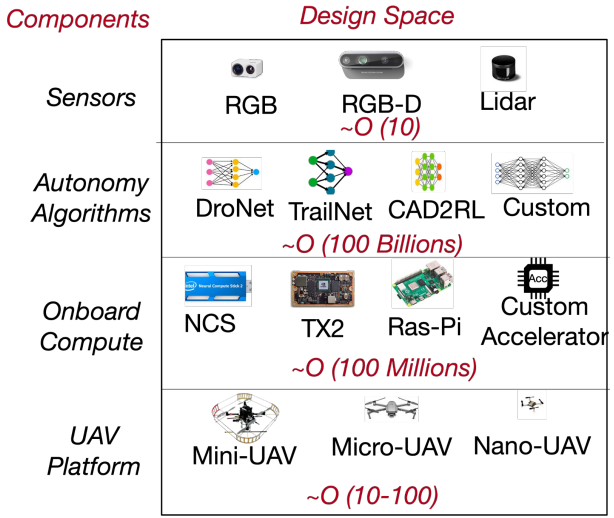
- Application-level co-design
- "Frankenstein" co-design
- Chiplet integration

Specialization-Driven Innovation

The need for domain-specific efficiency is increasing!



Increasing Complexity & Cost



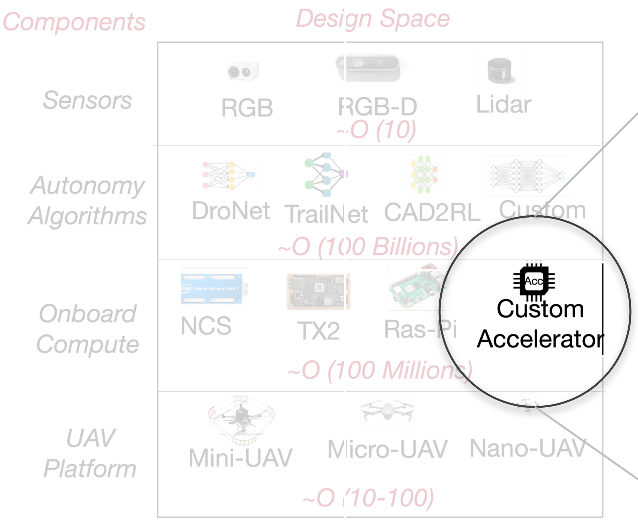
[Krishnan et al., MICRO'22]

Increasing Complexity & Cost

Large Design Space

Parameters

~ 10^{14} to 10^{2300}



[Krishnan et al., MICRO'22]

A Full-Stack Search Technique for Domain Optimized Deep Learning Accelerators

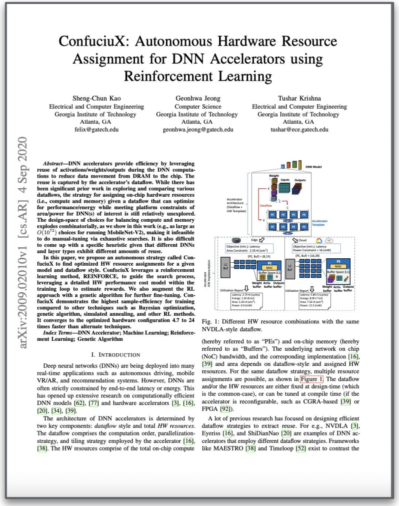
Dan Zhang, Safeen Huda, Ebrahim Songhori, Kartik Prabhu, Quoc Le, Anna Goldie, Azalia Mirhoseini

ABSTRACT

1 INTRODUCTION

CCS CONCEPTS

KEYWORDS



ConfuciuX: Autonomous Hardware Resource Assignment for DNN Accelerators using Reinforcement Learning

Sheng-Chun Kao, Geonhwa Jeong, Tushar Krishna

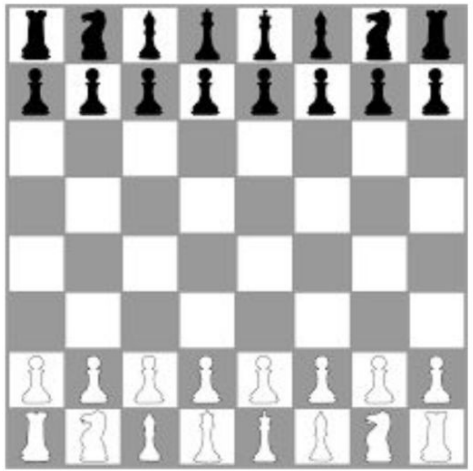
arXiv:2009.02010v1 [cs.AR] 4 Sep 2020

I. Introduction

Fig. 1: Different HW resource combinations with the same NVDLA-style dataflow.

Increasing Complexity & Cost

Chess

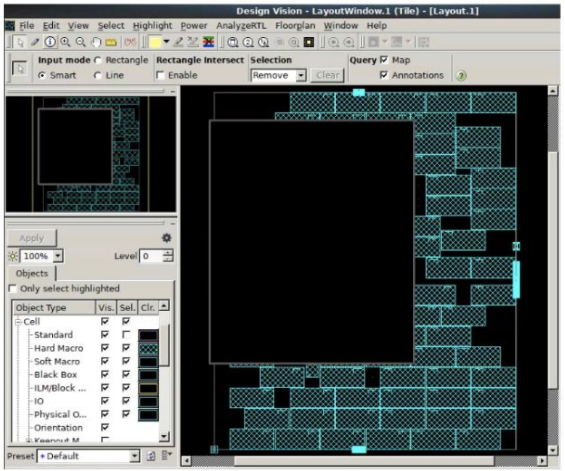


Number of states ~ 10^{123}

Go

Number of states ~ 10^{360}

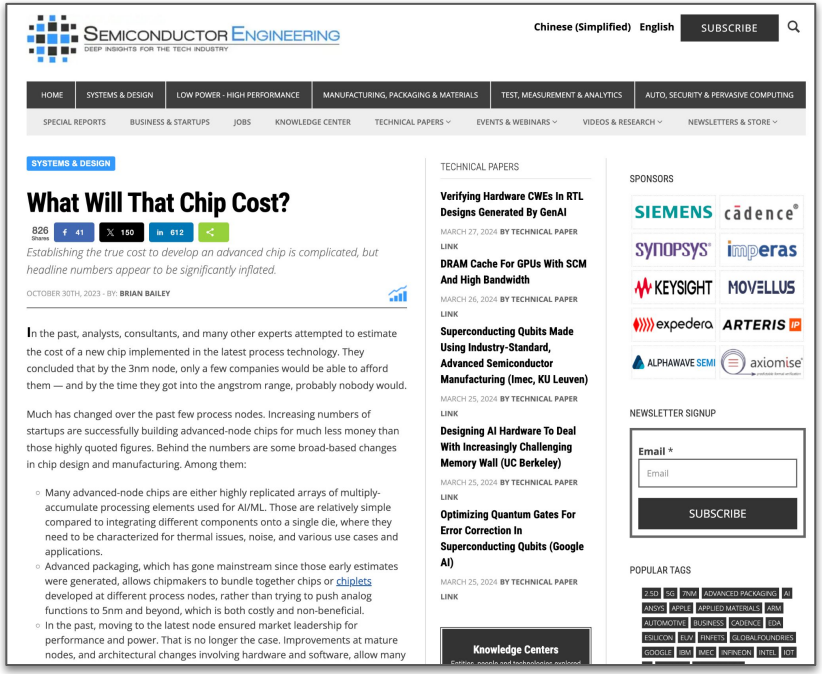
Chip Floorplanning



Number of states ~ 10^{9000}

[Mirhoseini, Nature'21]

Increasing Complexity & Cost



SEMICONDUCTOR ENGINEERING
DEEP INSIGHTS FOR THE TECH INDUSTRY

Chinese (Simplified) English SUBSCRIBE

HOME | SYSTEMS & DESIGN | LOW POWER - HIGH PERFORMANCE | MANUFACTURING, PACKAGING & MATERIALS | TEST, MEASUREMENT & ANALYTICS | AUTO, SECURITY & PERVASIVE COMPUTING

SPECIAL REPORTS | BUSINESS & STARTUPS | JOBS | KNOWLEDGE CENTER | TECHNICAL PAPERS | EVENTS & WEBINARS | VIDEOS & RESEARCH | NEWSLETTERS & STORE

SYSTEMS & DESIGN

What Will That Chip Cost?

826 Shares 41 150 612

Establishing the true cost to develop an advanced chip is complicated, but headline numbers appear to be significantly inflated.

OCTOBER 30TH, 2023 - BY: BRIAN BAILEY

In the past, analysts, consultants, and many other experts attempted to estimate the cost of a new chip implemented in the latest process technology. They concluded that by the 3nm node, only a few companies would be able to afford them — and by the time they got into the angstrom range, probably nobody would.

Much has changed over the past few process nodes. Increasing numbers of startups are successfully building advanced-node chips for much less money than those highly quoted figures. Behind the numbers are some broad-based changes in chip design and manufacturing. Among them:

- Many advanced-node chips are either highly replicated arrays of multiply-accumulate processing elements used for AI/ML. Those are relatively simple compared to integrating different components onto a single die, where they need to be characterized for thermal issues, noise, and various use cases and applications.
- Advanced packaging, which has gone mainstream since those early estimates were generated, allows chipmakers to bundle together chips or chiplets developed at different process nodes, rather than trying to push analog functions to 5nm and beyond, which is both costly and non-beneficial.
- In the past, moving to the latest node ensured market leadership for performance and power. That is no longer the case. Improvements at mature nodes, and architectural changes involving hardware and software, allow many

TECHNICAL PAPERS

Verifying Hardware CWEs In RTL Designs Generated By GenAI
MARCH 27, 2024 BY TECHNICAL PAPER LINK

DRAM Cache For GPUs With SCM And High Bandwidth
MARCH 26, 2024 BY TECHNICAL PAPER LINK

Superconducting Qubits Made Using Industry-Standard, Advanced Semiconductor Manufacturing (Imec, KU Leuven)
MARCH 25, 2024 BY TECHNICAL PAPER LINK

Designing AI Hardware To Deal With Increasingly Challenging Memory Wall (UC Berkeley)
MARCH 25, 2024 BY TECHNICAL PAPER LINK

Optimizing Quantum Gates For Error Correction In Superconducting Qubits (Google AI)
MARCH 25, 2024 BY TECHNICAL PAPER LINK

Knowledge Centers

SPONSORS

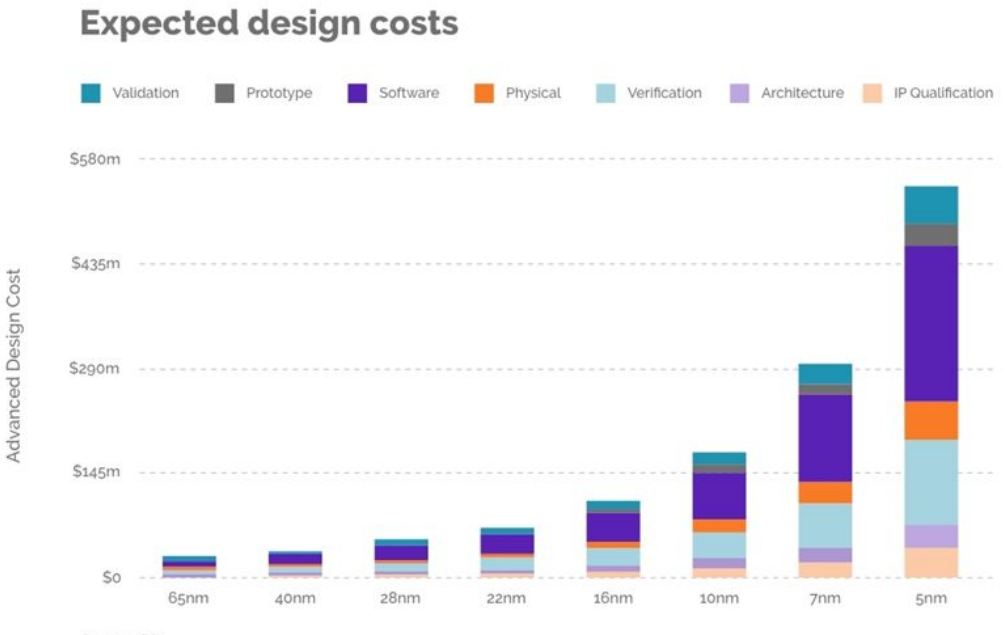
SIEMENS cadence synopsys imperas KEYSIGHT MOVELLUS expedera ARTERIS IP ALPHAWAVE SEMI axiomise

NEWSLETTER SIGNUP

Email *
Email
SUBSCRIBE

POPULAR TAGS

2.5D | 5G | 7NM | ADVANCED PACKAGING | AI | ANSYS | APPLE | APPLIED MATERIALS | ARM | AUTOMOTIVE | BUSINESS | CADENCE | EDA | ESILICON | EUV | FINFETS | GLOBALFOUNDRIES | GOOGLE | IBM | IMEC | INFINEON | INTEL | IOT

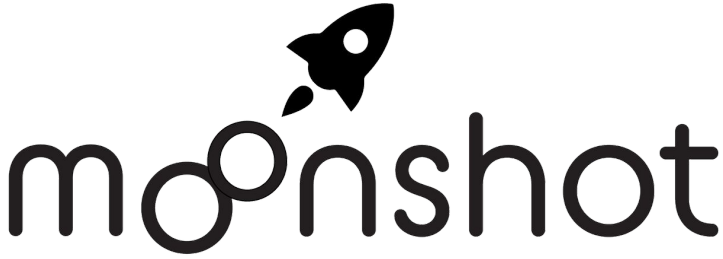


How can we **rethink the way we design and build systems**—reducing human intervention, embracing complexity, and delivering greater efficiency?

How can we rethink the way we design and build systems—**reducing human intervention**, embracing complexity, and delivering greater efficiency?

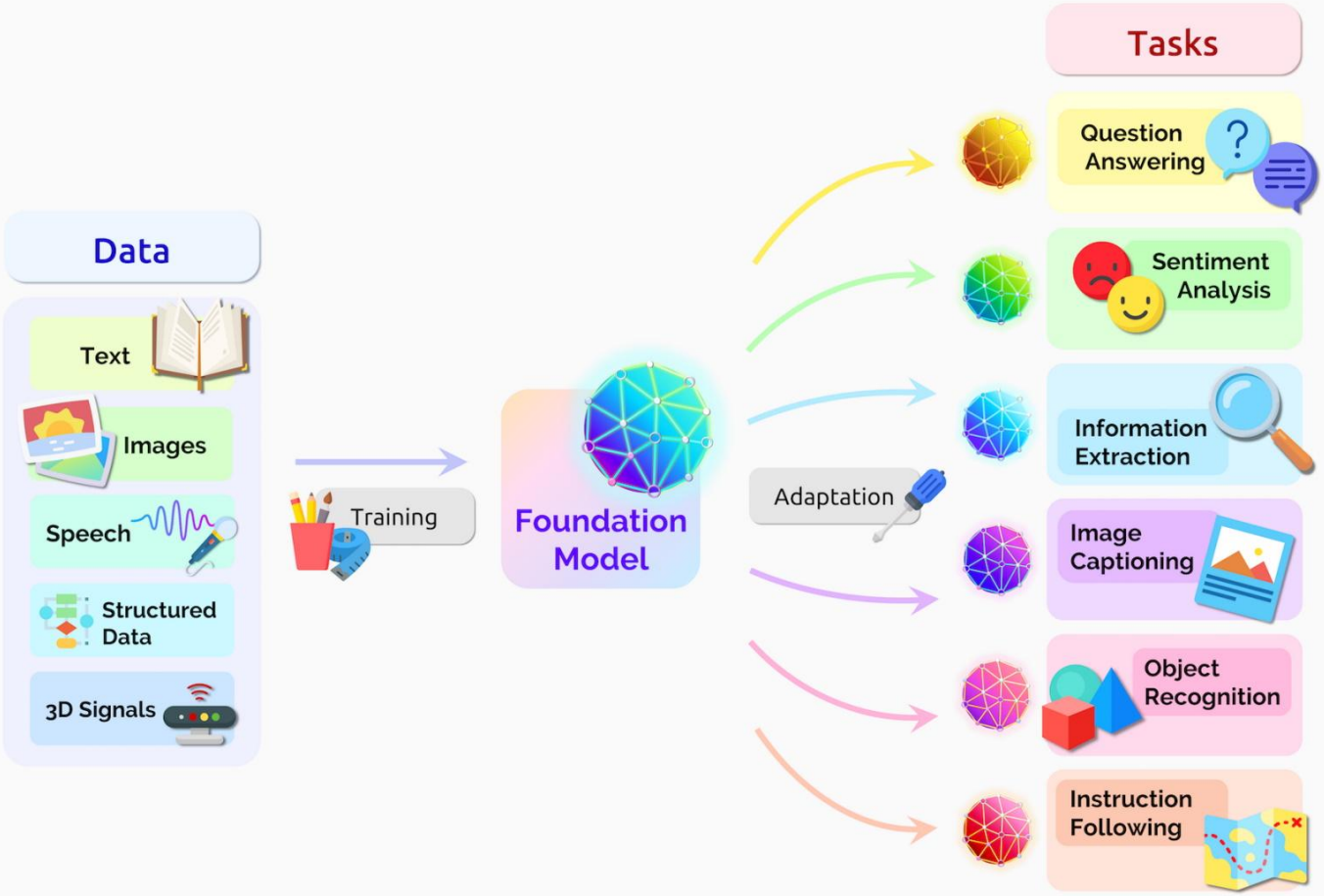
How can we rethink the way we design and build systems—reducing human intervention, **embracing complexity**, and delivering greater efficiency?

How can we rethink the way we design and build systems—reducing human intervention, embracing complexity, and **delivering greater efficiency**?

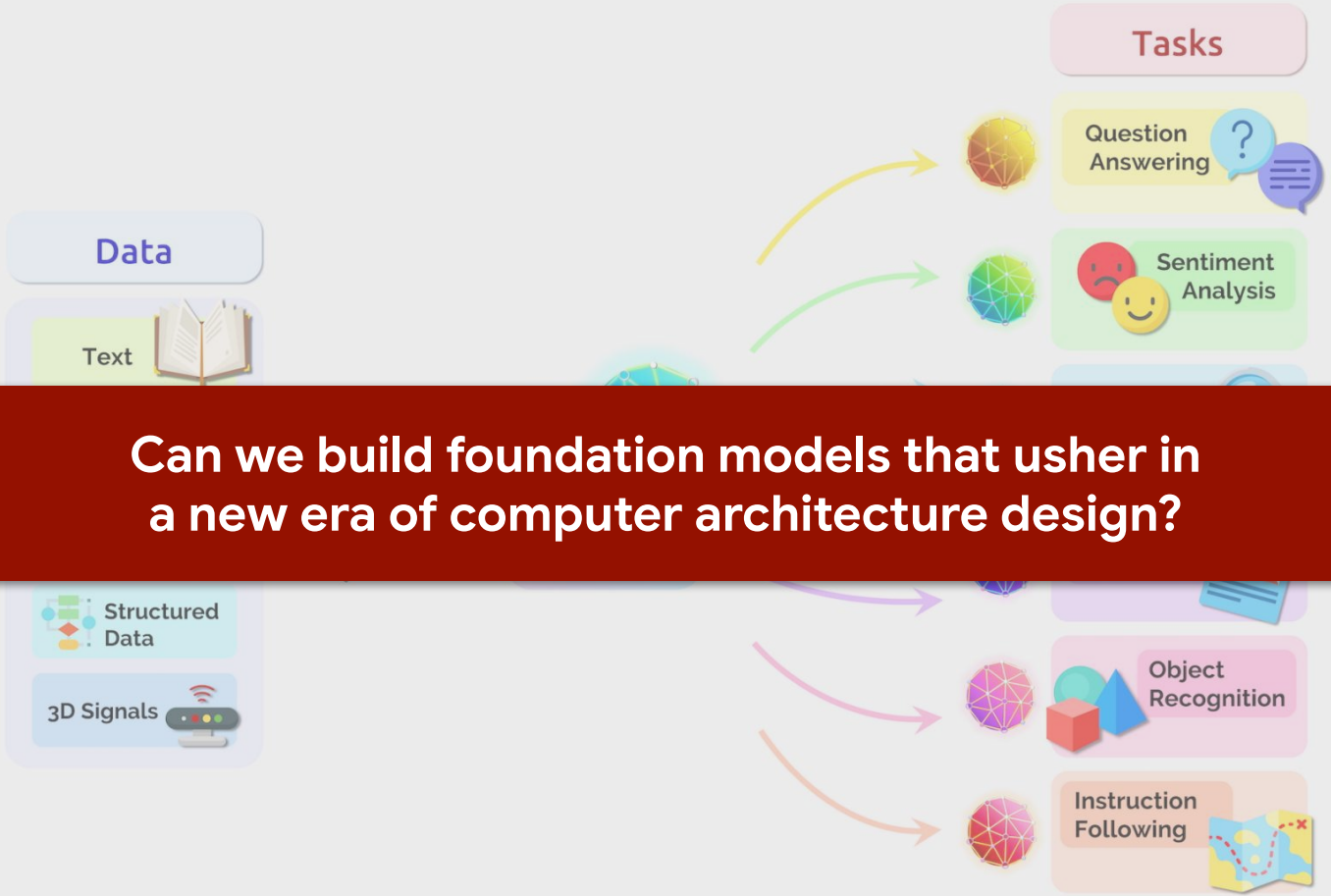

moonshot

moonshot

"Act like an architect — design me a custom 64-bit RISC-V processor with full vector extension support and optimize it for less than 3 Watt TDP in a 7 nm LP process node using the TSMC plugin library"



[Liang et al., 2022]



Can we build foundation models that usher in a new era of computer architecture design?

[Liang et al., 2022]

The Bottlenecks

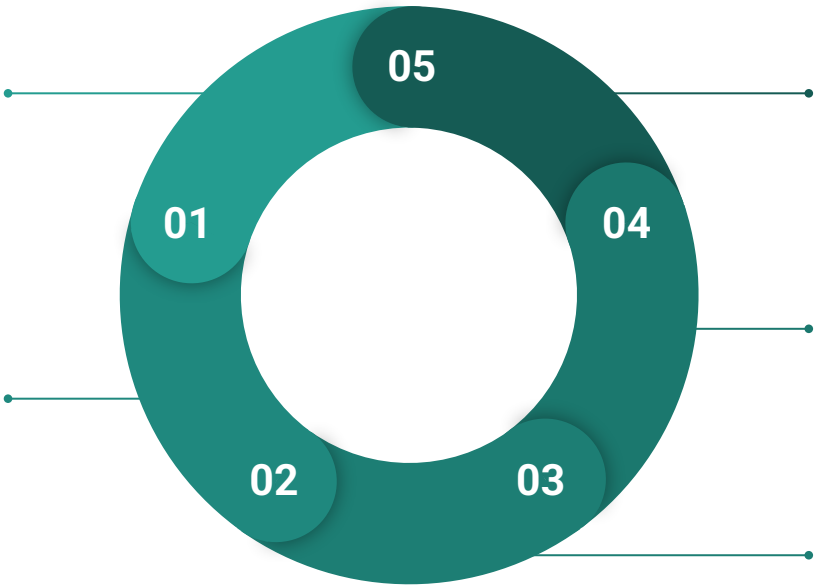
Key Themes for Architecture 2.0

Datasets

What datasets do we need? How we should collect these datasets for architecture research? What metadata should the datasets contain to enable broad usage? How do we create standard data formats from any ML algorithm?

ML Algorithms

How can we learn and apply new ML algorithms to effectively design high-performance/efficient systems? How do we make our community more accessible to ML researchers? How do we embrace ML algorithm design as part of architecture research?



Workforce & Training

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

Tools & Infrastructure

How do we reduce the sim2real gap? What instrumentation mechanisms do we need for creating the datasets? What gym environments do we need to enable data-centric AI? How do we define standard data formats for interoperability?

Best Practices

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

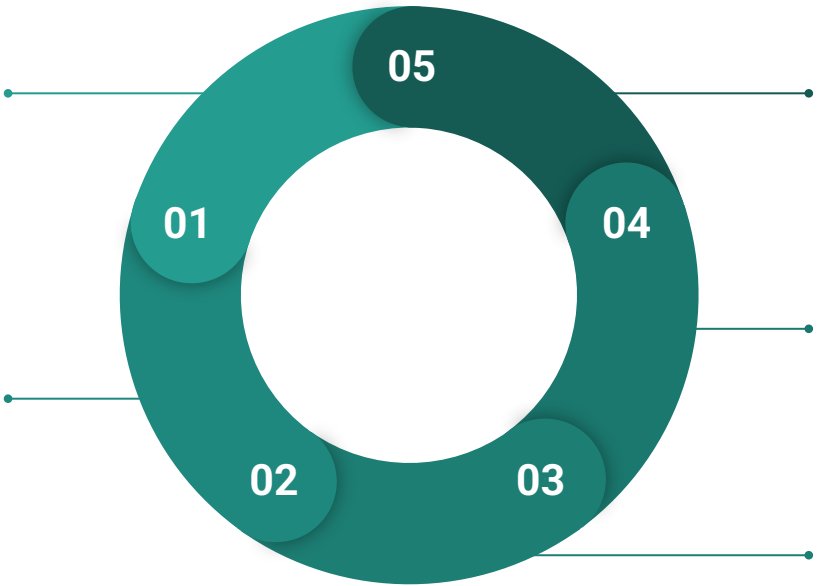
Key Themes for Architecture 2.0

Datasets

What datasets do we need? How we should collect these datasets for architecture research? What metadata should the datasets contain to enable broad usage? How do we create standard data formats from any ML algorithm?

ML Algorithms

How can we learn and apply new ML algorithms to effectively design high-performance/efficient systems? How do we make our community more accessible to ML researchers? How do we embrace ML algorithm design as part of architecture research?



Workforce & Training

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

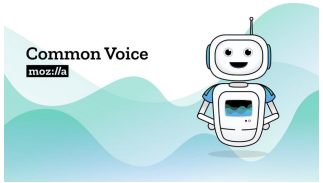
Tools & Infrastructure

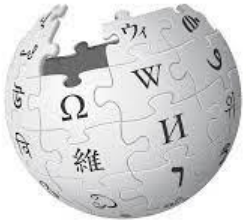
How do we reduce the sim2real gap? What instrumentation mechanisms do we need for creating the datasets? What gym environments do we need to enable data-centric AI? How do we define standard data formats for interoperability?

Best Practices

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

Lack of large, high-quality public datasets





Lack of large, high-quality public datasets

Challenge #1:
Inability to "scrape" the internet for creating datasets/benchmarks

Common Crawl

WIKIPEDIA
The Free Encyclopedia

GitHub

Lack of large, high-quality public datasets

Challenge #2:
Domain-specific expertise required for validating dataset quality

Common Crawl

WIKIPEDIA
The Free Encyclopedia

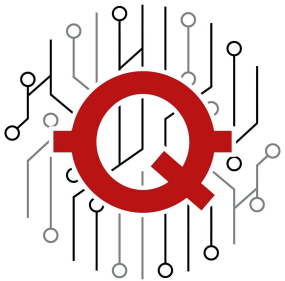
GitHub

Related Work on LLMs for Systems

Benchmark & Dataset for Computing Systems	Focus in Computing Stack	Conceptual & Analytical QA	Design QA & Program Impl.	Multimodal Assessment	Expert Verified	Benchmark Size
SWE-bench (Jimenez et al., 2023)	Software Eng.	✗	✓	✗	✗	2294
SWE-bench Verified (OpenAI, 2024)	Software Eng.	✗	✓	✗	✓	500
SWE-Perf (He et al., 2025)	Performance Eng.	✗	✓	✗	✗	140
KernelBench (Ouyang et al., 2025)	Performance Eng.	✗	✓	✗	✗	250
CodeMMLU (Nguyen et al., 2025)	Code Reasoning	✓	✗	✗	✗	19912
CRUXEval (Gu et al., 2024)	Code Reasoning	✓	✓	✗	✗	800
The gap…						
SLDB (Alvanaki et al., 2025)	System Design	✗	✓	✗	✓	10
CreativEval (DeLorenzo et al., 2024b)	HW Design	✗	✓	✗	✗	120
VerilogEval (Liu et al., 2023b)	RTL Generation	✗	✓	✗	✗	156
CVDP (Pinckney et al., 2025b)	RTL Generation	✓	✓	✗	✓	783
MG-Verilog (Zhang et al., 2024b)	RTL Generation	✗	✓	✗	✗	11000
EDA Corpus (Wu et al., 2024a)	EDA Tooling	✓	✓	✗	✓	1533
FIXME (Wan et al., 2025)	Verification	✗	✓	✗	✗	180
ChiPBench (Wang et al., 2024b)	Layout	✗	✓	✗	✗	20

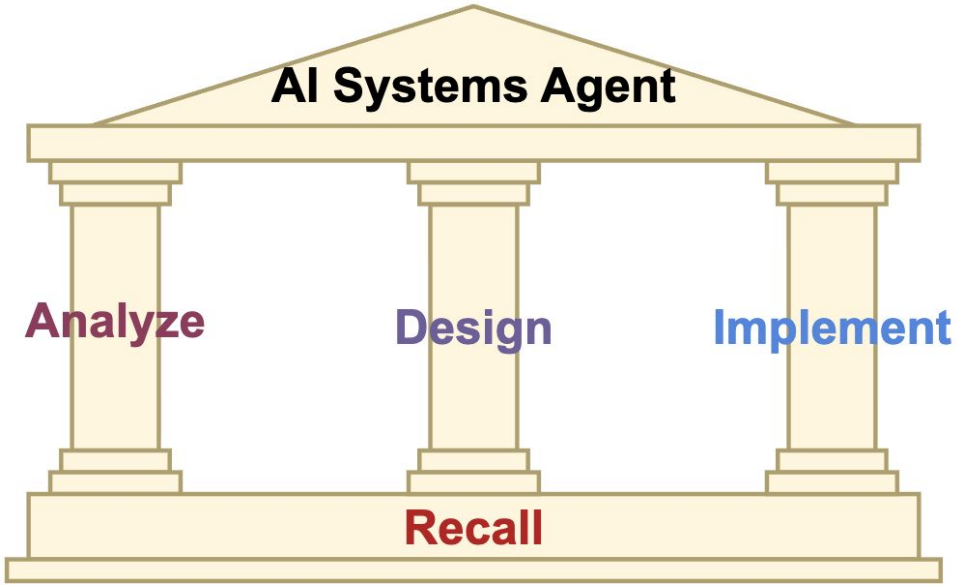
QuArch: A Benchmark for Evaluating LLM Reasoning in Computer Architecture

ArXiv paper

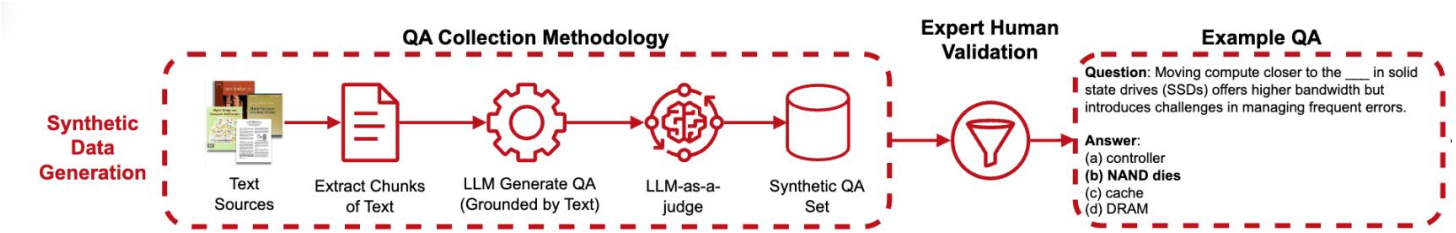


Quarch.ai website

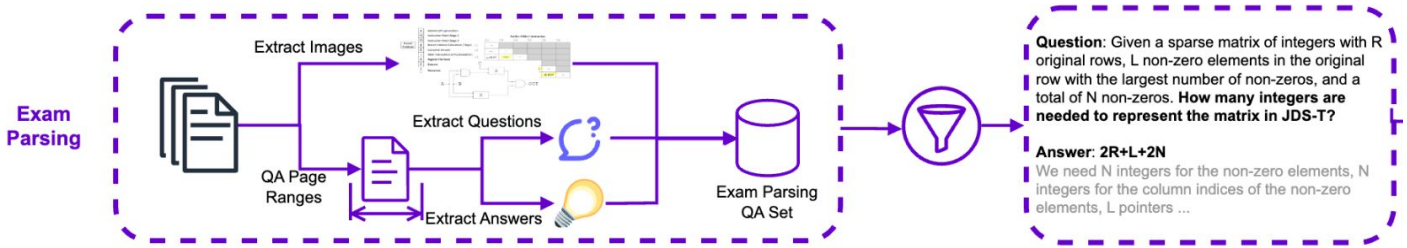
QuArch QA Skills Framework



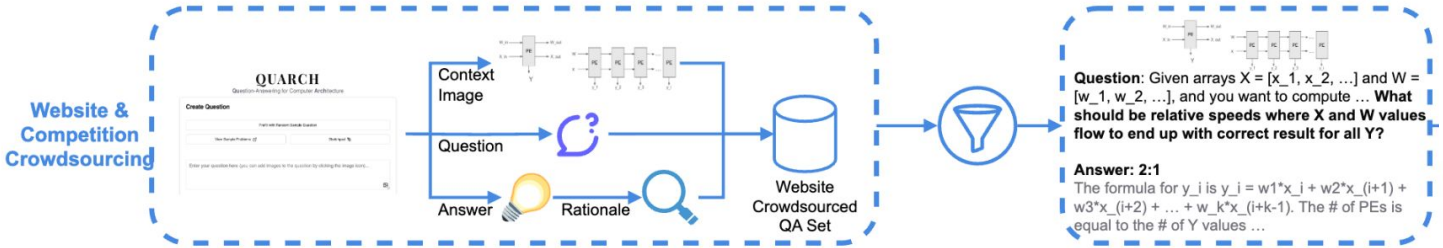
QuArch Construction - Synthetic Data



QuArch Construction - Exam Parsing



QuArch Construction - Crowdsourced



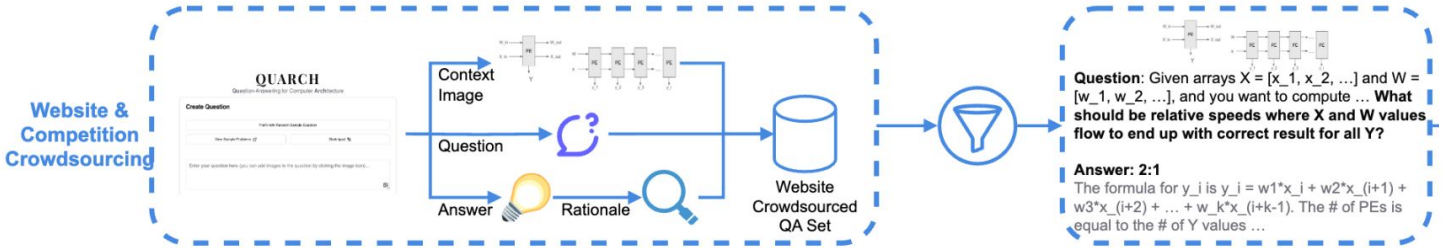
QuArch Construction - Crowdsourced

ASPLOS 2026 Competition Winners

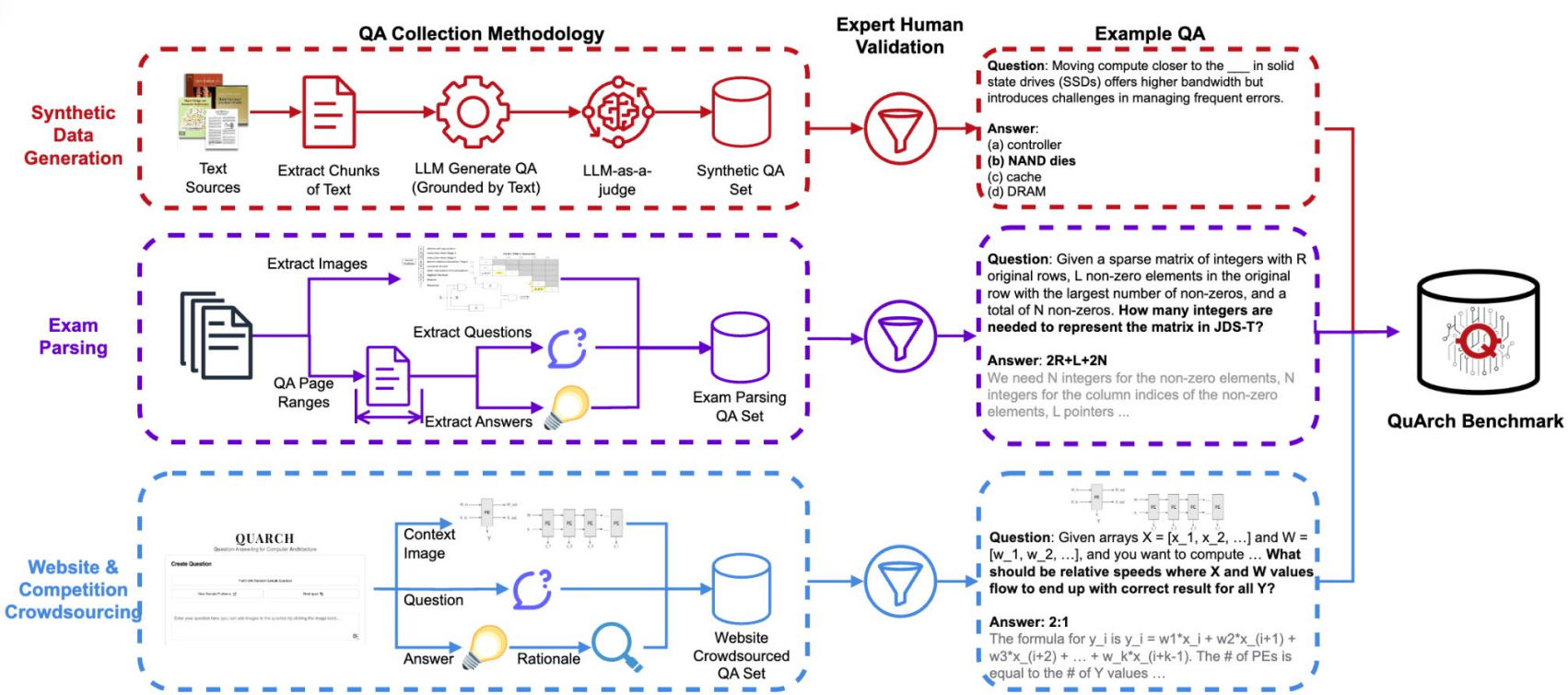
1. Mahnoor Malik, NED University of Technology
2. User “hexu”, Advanced Computing SIG
3. Gokulan Ravi, Purdue University

Congratulations!

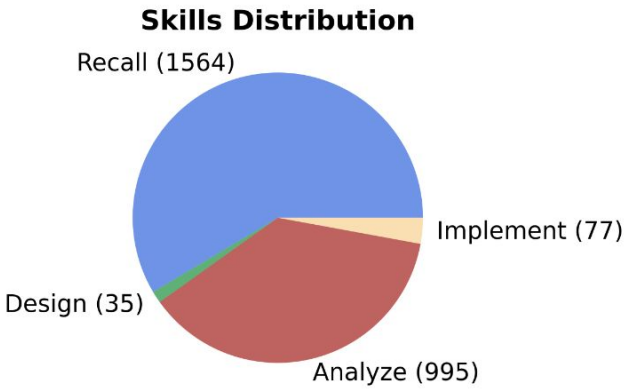
We will reach out individually for awards distribution.

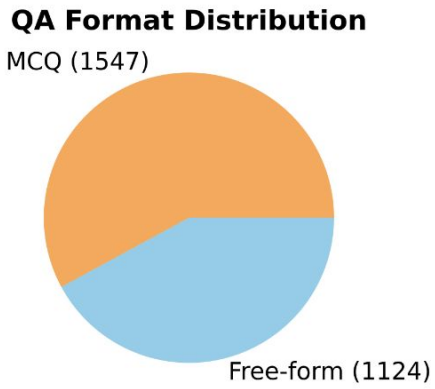


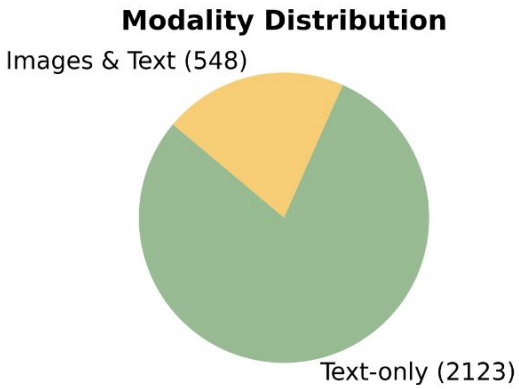
QuArch Construction



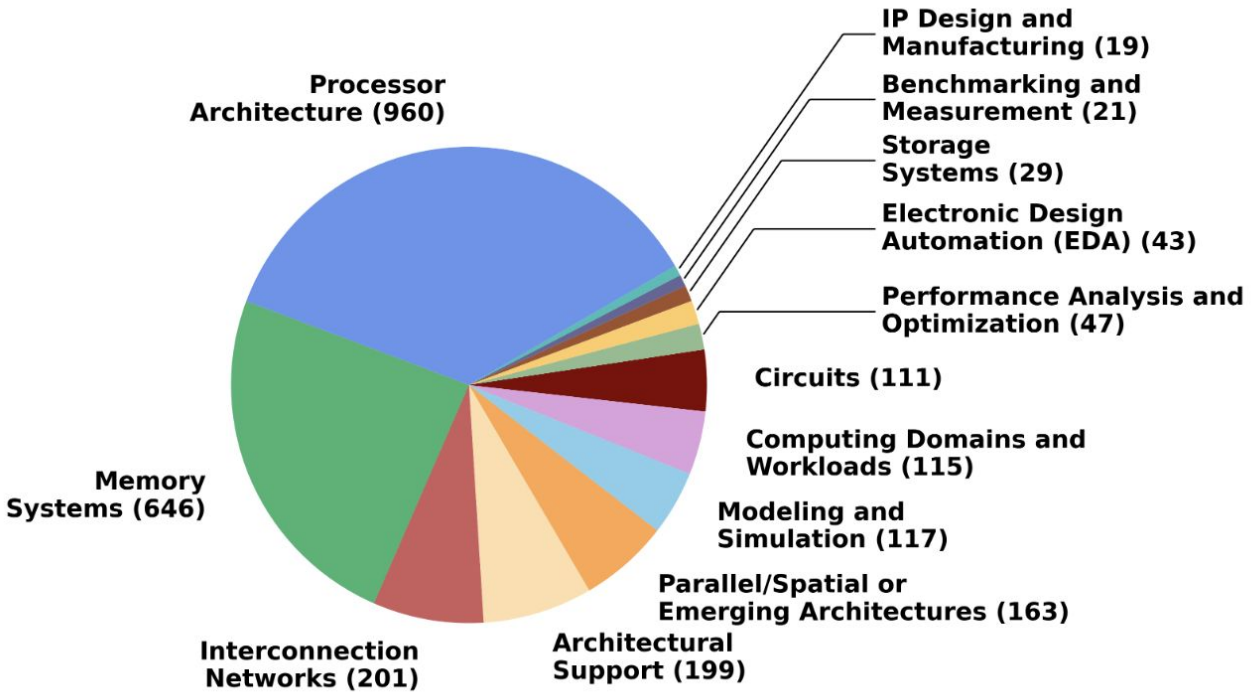
QuArch Format Breakdown







QuArch Topic Breakdown



Distribution of 2500+ QAs Across Architecture Topics

The Eval

“How well do LLMs understand computer architecture?”

Recall Required

What is the capital of Australia?

A) Sydney

B) Canberra

C) Melbourne

D) Brisbane

Recall Required

What is the capital of Australia?

A) Sydney

B) Canberra

C) Melbourne

D) Brisbane

***Factual* Question-Answering: Jump to answer in "*one-shot.*"**

Recall Required

What is the capital of Australia?

A) Sydney
B) Canberra
C) Melbourne
D) Brisbane

***Factual* Question-Answering: Jump to answer in "*one-shot.*"**

Reasoning Required

If a train departed at 8:30AM traveling at 60 mph and travels for 180 miles, what time did it arrive at its destination if it took a 45 minute break in between for maintenance?

A) 10:30AM
B) 11:30AM
C) 12:15PM
D) 12:30PM

Recall Required

What is the capital of Australia?

A) Sydney
B) Canberra
C) Melbourne
D) Brisbane

***Factual* Question-Answering: Jump to answer in "*one-shot.*"**

Reasoning Required

If a train departed at 8:30AM traveling at 60 mph and travels for 180 miles, what time did it arrive at its destination if it took a 45 minute break in between for maintenance?

A) 10:30AM
B) 11:30AM
C) 12:15PM
D) 12:30PM

Requires (complex) multi-step generation with intermediate steps & understanding relationship between distance, speed, and time.

QuArch-Recall

In ___, each processor has its own local memory system.

(a) symmetric multiprocessing
(b) asymmetric multiprocessing
(c) core-based multiprocessing
(d) clustered multiprocessing

***Factual* Question-Answering: Jump to answer in "*one-shot.*"**

QuArch-Recall

In ___, each processor has its own local memory system.

(a) symmetric multiprocessing
(b) asymmetric multiprocessing
(c) core-based multiprocessing
(d) clustered multiprocessing

***Factual* Question-Answering: Jump to answer in "*one-shot.*"**

QuArch-Reasoning

Assume a DRAM system with a burst size of 256 bytes and a peak bandwidth of 240 GB/s. Assume a thread block size of 256 and warp size of 32 and that A is a float array in the global memory. What is the maximal memory data access throughput we can hope to achieve in the following access to A?

```
int i = 4*blockIdx.x * blockDim.x + threadIdx.x;
float temp = A[i];
```

(A) 240 GB/s (B) 120 GB/s (C) 60 GB/s (D) 30 GB/s

Requires (complex) multi-step generation with intermediate steps & understanding relationships between software, burst size, warp size, memory bandwidth, etc.

Key Observations & Insights from QuArch

Model	QuArch-Recall	QuArch-Reasoning
Multimodal Models		
GPT-5		
GPT-5 (Non-Reasoning)		
Gemini 2.5 Pro		
Gemini 2.5 Flash		
Claude Sonnet 4		
Claude 3.7 Sonnet Thinking		
Llama 4 Maverick		
Mistral Medium 3.1		
Text-Only Models		
GPT-OSS 120B		
DeepSeek R1		

Key Observations & Insights from QuArch

Model	QuArch-Recall	QuArch-Reasoning
Multimodal Models		
GPT-5	**89.0**	
GPT-5 (Non-Reasoning)	**86.3**	
Gemini 2.5 Pro	**87.4**	
Gemini 2.5 Flash	83.4	
Claude Sonnet 4	85.5	
Claude 3.7 Sonnet Thinking	85.8	
Llama 4 Maverick	85.3	
Mistral Medium 3.1	84.5	
Text-Only Models		
GPT-OSS 120B	84.2	
DeepSeek R1	86.9	

Key Observations & Insights from QuArch

Model	QuArch-Recall	QuArch-Reasoning
Multimodal Models		
GPT-5	**89.0**	**72.4**
GPT-5 (Non-Reasoning)	**86.3**	49.0
Gemini 2.5 Pro	**87.4**	**62.9**
Gemini 2.5 Flash	83.4	**56.8**
Claude Sonnet 4	85.5	48.4
Claude 3.7 Sonnet Thinking	85.8	52.1
Llama 4 Maverick	85.3	34.2
Mistral Medium 3.1	84.5	34.1
Text-Only Models		
GPT-OSS 120B	84.2	64.7
DeepSeek R1	86.9	56.1

Key Observations & Insights from QuArch

Model	QuArch-Recall	QuArch-Reasoning
Multimodal Models		
GPT-5	89.0	72.4
GPT-5 (Non-Reasoning)	86.3	49.0
Gemini 2.5 Pro	87.4	62.9
Gemini 2.5 Flash	83.4	56.8
Claude Sonnet 4	85.5	48.4
Claude 3.7 Sonnet Thinking	85.8	52.1
Llama 4 Maverick	85.3	34.2
Mistral Medium 3.1	84.5	34.1
Text-Only Models		
GPT-OSS 120B	84.2	64.7
DeepSeek R1	86.9	56.1

Key Observations & Insights from QuArch

Model	QuArch-Recall	QuArch-Reasoning	Δ
Multimodal Models			
GPT-5	**89.0**	**72.4**	**-16.5**
GPT-5 (Non-Reasoning)	**86.3**	49.0	-37.3
Gemini 2.5 Pro	**87.4**	**62.9**	**-24.5**
Gemini 2.5 Flash	83.4	**56.8**	**-26.6**
Claude Sonnet 4	85.5	48.4	-37.0
Claude 3.7 Sonnet Thinking	85.8	52.1	-33.7
Llama 4 Maverick	85.3	34.2	-51.1
Mistral Medium 3.1	84.5	34.1	-50.4
Text-Only Models			
GPT-OSS 120B	84.2	64.7	-19.5
DeepSeek R1	86.9	56.1	-30.7

Key Observations & Insights from QuArch

Model	QuArch-Recall	QuArch-Reasoning	Δ
Claude 3.7 Sonnet Thinking	85.8	52.1	-33.7
Llama 4 Maverick	85.3	34.2	-51.1
Mistral Medium 3.1	84.5	34.1	-50.4
Text-Only Models			
GPT-OSS 120B	84.2	64.7	-19.5
DeepSeek R1	86.9	56.1	-30.7

Takeaway #1

Datasets will be required for effective AI agents in computer architecture.

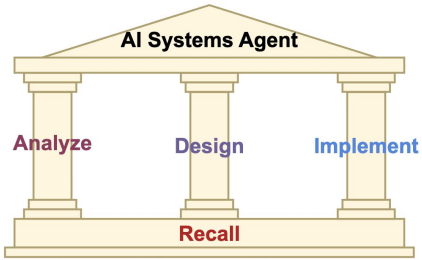
Key Observations & Insights from QuArch

Model	QuArch-Recall	QuArch-Reasoning	Δ
Claude 3.7 Sonnet Thinking	85.8	52.1	-33.7
Llama 4 Maverick	85.3	34.2	-51.1
Mistral Medium 3.1	84.5	34.1	-50.4
Text-Only Models			
GPT-OSS 120B	84.2	64.7	-19.5
DeepSeek R1	86.9	56.1	-30.7

Takeaway #2

Recall-Reasoning Gap:
Recall is mastered but higher order thinking skills are not.

Key Observations & Insights from QuArch

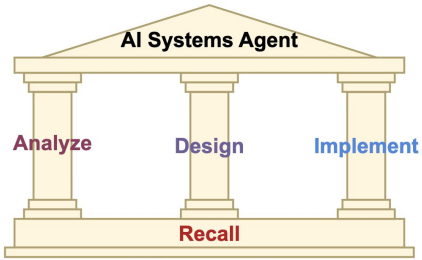


Foundations of AI Agents for Systems Design

Model	Recall	QuArch-Reasoning			
		Analyze	Design	Implement	Overall
Closed-Source Multimodal Models					
GPT-5	89.0	72.1	86.7	71.0	72.4
GPT-5 (Non-Reasoning)	86.3	49.5	52.4	40.7	49.0
GPT-4o	84.1	28.4	12.4	22.9	27.5
Gemini 2.5 Pro	87.4	63.3	59.0	59.7	62.9
Gemini 2.5 Flash	83.4	57.3	57.1	49.8	56.8
Claude Sonnet 4	85.5	49.0	36.2	46.8	48.4
Claude 3.7 Sonnet Thinking	85.8	53.3	34.3	45.5	52.1
Mistral Medium 3.1	84.5	34.8	27.6	27.7	34.1
Open-Source Multimodal Models					
Gemma 3 27B IT	75.6	22.4	15.2	16.0	21.7
Gemma 3 4B IT	62.3	7.6	2.9	3.5	7.2
Llama 4 Maverick	85.3	35.1	18.1	30.2	34.2
Llama 3.2 11B	69.4	8.8	1.0	4.0	8.2
Mistral Small 3.2 24B Instruct	78.0	24.3	16.2	17.7	23.6
Text-Only Models					
GPT-OSS 120B	84.2	65.5	56.1	57.8	64.7
DeepSeek R1	86.9	57.4	38.6	47.1	56.1
Llama 3.3 70B	80.4	25.8	0.0	13.7	24.2
Llama 3.2 1B	36.5	1.7	0.0	1.0	1.6
Mistral Codestral 2508	75.3	29.7	5.3	16.7	28.1
Mistral Devstral Medium	81.9	29.0	3.5	22.5	27.7
Kimi K2 0905	84.2	43.9	35.1	37.3	43.2
Qwen3 Coder 480B A35B Instruct	82.9	41.7	17.5	31.4	40.2
Qwen3 235B A22B Thinking	85.6	62.2	50.9	52.9	61.3
Qwen3 235B A22B NonThinking Instruct	86.5	56.4	42.1	47.1	55.3
Qwen3 Next 80B A3B Thinking	84.5	54.9	36.8	42.2	53.5
Qwen3 30B A3B Thinking	82.5	50.0	31.6	37.3	48.6
Qwen3 Coder 30B A3B Instruct	78.3	28.2	8.8	12.7	26.6

LLMs: 53.1% avg

Key Observations & Insights from QuArch



Foundations of AI Agents for Systems Design

Model	Recall	QUARCH-REASONING			
		Analyze	Design	Implement	Overall
Closed-Source Multimodal Models					
GPT-5	89.0	72.1	86.7	71.0	72.4
GPT-5 (Non-Reasoning)	86.3	49.5	52.4	40.7	49.0
GPT-4o	84.1	28.4	12.4	22.9	27.5
Gemini 2.5 Pro	87.4	63.3	59.0	59.7	62.9
Gemini 2.5 Flash	83.4	57.3	57.1	49.8	56.8
Claude Sonnet 4	85.5	49.0	36.2	46.8	48.4
Claude 3.7 Sonnet Thinking	85.8	53.3	34.3	45.5	52.1
Mistral Medium 3.1	84.5	34.8	27.6	27.7	34.1
Open-Source Multimodal Models					
Gemma 3 27B IT	75.6	22.4	15.2	16.0	21.7
Gemma 3 4B IT	62.3	7.6	2.9	3.5	7.2
Llama 4 Maverick	85.3	35.1	18.1	30.2	34.2
Llama 3.2 11B	69.4	8.8	1.0	4.0	8.2
Mistral Small 3.2 24B Instruct	78.0	24.3	16.2	17.7	23.6
Text-Only Models					
GPT-OSS 120B	84.2	65.5	56.1	57.8	64.7
DeepSeek R1	86.9	57.4	38.6	47.1	56.1
Llama 3.3 70B	80.4	25.8	0.0	13.7	24.2
Llama 3.2 1B	36.5	1.7	0.0	1.0	1.6
Mistral Codestral 2508	75.3	29.7	5.3	16.7	28.1
Mistral Devstral Medium	81.9	29.0	3.5	22.5	27.7
Kimi K2 0905	84.2	43.9	35.1	37.3	43.2
Qwen3 Coder 480B A35B Instruct	82.9	41.7	17.5	31.4	40.2
Qwen3 235B A22B Thinking	85.6	62.2	50.9	52.9	61.3
Qwen3 235B A22B NonThinking Instruct	86.5	56.4	42.1	47.1	55.3
Qwen3 Next 80B A3B Thinking	84.5	54.9	36.8	42.2	53.5
Qwen3 30B A3B Thinking	82.5	50.0	31.6	37.3	48.6
Qwen3 Coder 30B A3B Instruct	78.3	28.2	8.8	12.7	26.6

SLMs: 22.9% avg

Key Observations & Insights from QuArch

Model	Recall	QuArch-Reasoning			
		Analyze	Design	Implement	Overall
Closed-Source Multimodal Models					
GPT-5	89.0	72.1	86.7	71.0	72.4
GPT-5 (Non-Reasoning)	86.3	49.5	52.4	40.7	49.0

Takeaway #3

SLM Struggles:
Today's small language models exacerbate the recall-reasoning gap.

Mistral Small 3.2 24B Instruct	78.0	24.3	16.2	17.7	23.6
Text-Only Models					
GPT-OSS 120B	84.2	65.5	56.1	57.8	64.7
DeepSeek R1	86.9	57.4	38.6	47.1	56.1
Llama 3.3 70B	80.4	25.8	0.0	13.7	24.2
Llama 3.2 1B	36.5	1.7	0.0	1.0	1.6
Mistral Codestral 2508	75.3	29.7	5.3	16.7	28.1
Mistral Devstral Medium	81.9	29.0	3.5	22.5	27.7
Kimi K2 0905	84.2	43.9	35.1	37.3	43.2
Qwen3 Coder 480B A35B Instruct	82.9	41.7	17.5	31.4	40.2
Qwen3 235B A22B Thinking	85.6	62.2	50.9	52.9	61.3
Qwen3 235B A22B NonThinking Instruct	86.5	56.4	42.1	47.1	55.3
Qwen3 Next 80B A3B Thinking	84.5	54.9	36.8	42.2	53.5
Qwen3 30B A3B Thinking	82.5	50.0	31.6	37.3	48.6
Qwen3 Coder 30B A3B Instruct	78.3	28.2	8.8	12.7	26.6

Key Observations & Insights from QuArch

Today's

gap.

Model	QuArch-Reasoning			
	Recall	Analyze	Design	Implement	Overall
		86.7	71.0	72.4
		52.4	40.7	49.0
			22.9	27.5
#3			59.7	62.9
			49.8	56.8

Small Language Models are the Future of Agentic AI

Peter Belcak[1] Greg Heinrich[1] Shizhe Diao[1] Yonggan Fu[1] Xin Dong[1]
Saurav Muralidharan[1] Yingyan Celine Lin[1,2] Pavlo Molchanov[1]
[1]NVIDIA Research [2]Georgia Institute of Technology
agents-research@nvidia.com

Abstract

Large language models (LLMs) are often praised for exhibiting near-human performance on a wide range of tasks and valued for their ability to hold a general conversation. The rise of agentic AI systems is, however, ushering in a mass of applications in which language models perform a small number of specialized tasks repetitively and with little variation.

Here we lay out the position that small language models (SLMs) are *sufficiently powerful, inherently more suitable, and necessarily more economical for many invocations in agentic systems, and are therefore the future of agentic AI*. Our argumentation is grounded in the current level of capabilities exhibited by SLMs, the common architectures of agentic systems, and the economy of LM deployment. We further argue that in situations where general-purpose conversational abilities are essential, heterogeneous agentic systems (i.e., agents invoking multiple different models) are the natural choice. We discuss the potential barriers for the adoption of SLMs in agentic systems and outline a general LLM-to-SLM agent conversion algorithm.

Our position[1], formulated as a value statement, highlights the significance of the operational and economic impact even a partial shift from LLMs to SLMs is to have on the AI agent industry. We aim to stimulate the discussion on the effective use of AI resources and hope to advance the efforts to lower the costs of AI of the present day. Calling for both contributions to and critique of our position, we commit to publishing all such correspondence at `research.nvidia.com/labs/lpr/slm-agents`.

1 Introduction

The deployment of agentic artificial intelligence is on a meteoric rise. Recent surveys show that m
than a half of large IT enterprises are actively using AI agents, with 21% having adopted just wi
the last year [14]. Aside from the users, markets also see substantial economic value in AI agents

SLM-MUX: Orchestrating Small Language Models for Reasoning

Chenyu Wang[1*] Zishen Wan[2*] Hao Kang[2] Emma Chen[1]
Zhiqiang Xie[3] Tushar Krishna[2] Vijay Janapa Reddi[1] Yilun Du[1]
[1]Harvard University [2]Georgia Institute of Technology [3]Stanford University

Abstract

With the rapid development of language models, the number of small language models (SLMs) has grown significantly. Although they do not achieve state-of-the-art accuracy, they are more efficient and often excel at specific tasks. This raises a natural question: can multiple SLMs be orchestrated into a system where each contributes effectively, achieving higher accuracy than any individual model? Existing orchestration methods have primarily targeted frontier models (e.g., GPT-4) and perform suboptimally when applied to SLMs. To address this gap, we propose a three-stage approach for orchestrating SLMs. First, we introduce SLM-MUX, a multi-model architecture that effectively coordinates multiple SLMs. Building on this, we develop two optimization strategies: (i) a model selection search that identifies the most complementary SLMs from a given pool, and (ii) test-time scaling tailored to SLM-MUX. Our approach delivers strong results: Compared to existing orchestration methods, our approach achieves up to 13.4% improvement on MATH, 8.8% on GPQA, and 7.0% on GSM8K. With just two SLMs, SLM-MUX outperforms Qwen 2.5 72B on GPQA and GSM8K, and matches its performance on MATH. We further provide theoretical analyses to substantiate the advantages of our method. In summary, we demonstrate that SLMs can be effectively orchestrated into more accurate and efficient systems through the proposed approach. Project page and code: https://slm-mux.github.io.

1 Introduction

Recent years have witnessed a surge of small-sized language models (SLMs) containing billions to tens of billions of parameters (Wang et al., 2024a; Javaheripi & Bubeck, 2023; Guo et al., 2025; Allal et al., 2025). While these models may underperform state-of-the-art frontier language models, which usually contain hundreds of billions to trillions of parameters, on any given query, they offer substantially lower inference costs, are more affordable to train and finetune, and allow edge deployment due to their small size (Belcak et al., 2025). Meanwhile, frontier models have reached trillion-parameter scales where further increases in size and training data yield diminishing returns. This mirrors a well-known challenge in computer architecture two decades ago: when enlarging single CPU cores no longer delivered proportional performance gains, computer architects turned to designing multi-core processors, where multiple smaller cores working together enabled sustained

Key Observations & Insights from QuArch

Consider the code given below:

```
bool detect_duplicate(int* elements, int length){
    bool cond, dup = false;
    for(int i=0; i<length; i++) {
        for(int j=0; j<length; j++) {
            cond = (elements[i] == elements[j]) && (i!=j);
            dup = CMOV(cond, cond, dup);
        }
    }
    return !dup;
}
```

Assume that CMOV is implemented as the cmov x86 instruction, a secure constant-time instruction.

Assume the length of the array is already known publicly and is not sensitive information.

Does this code have timing side channels that can reveal insights into the contents of the elements array? Explain why.

Key Observations & Insights from QuArch

Consider the code given below:

```
bool detect_duplicate(int* elements, int length){
    bool cond, dup = false;
    for(int i=0; i<length; i++) {
        for(int j=0; j<length; j++) {
            cond = (elements[i] == elements[j]) && (i!=j);
            dup = CMOV(cond, cond, dup);
        }
    }
    return !dup;
}
```

Assume that CMOV is implemented as the cmov x86 instruction, a secure constant-time instruction.

Assume the length of the array is already known publicly and is not sensitive information.

Does this code have timing side channels that can reveal insights into the contents of the elements array? Explain why.

Key Observations & Insights from QuArch

Consider the code given below:

```
bool detect_duplicate(int* elements, int length){
    return !dup;
}
```

Takeaway #4

Multiple model failure points exist:

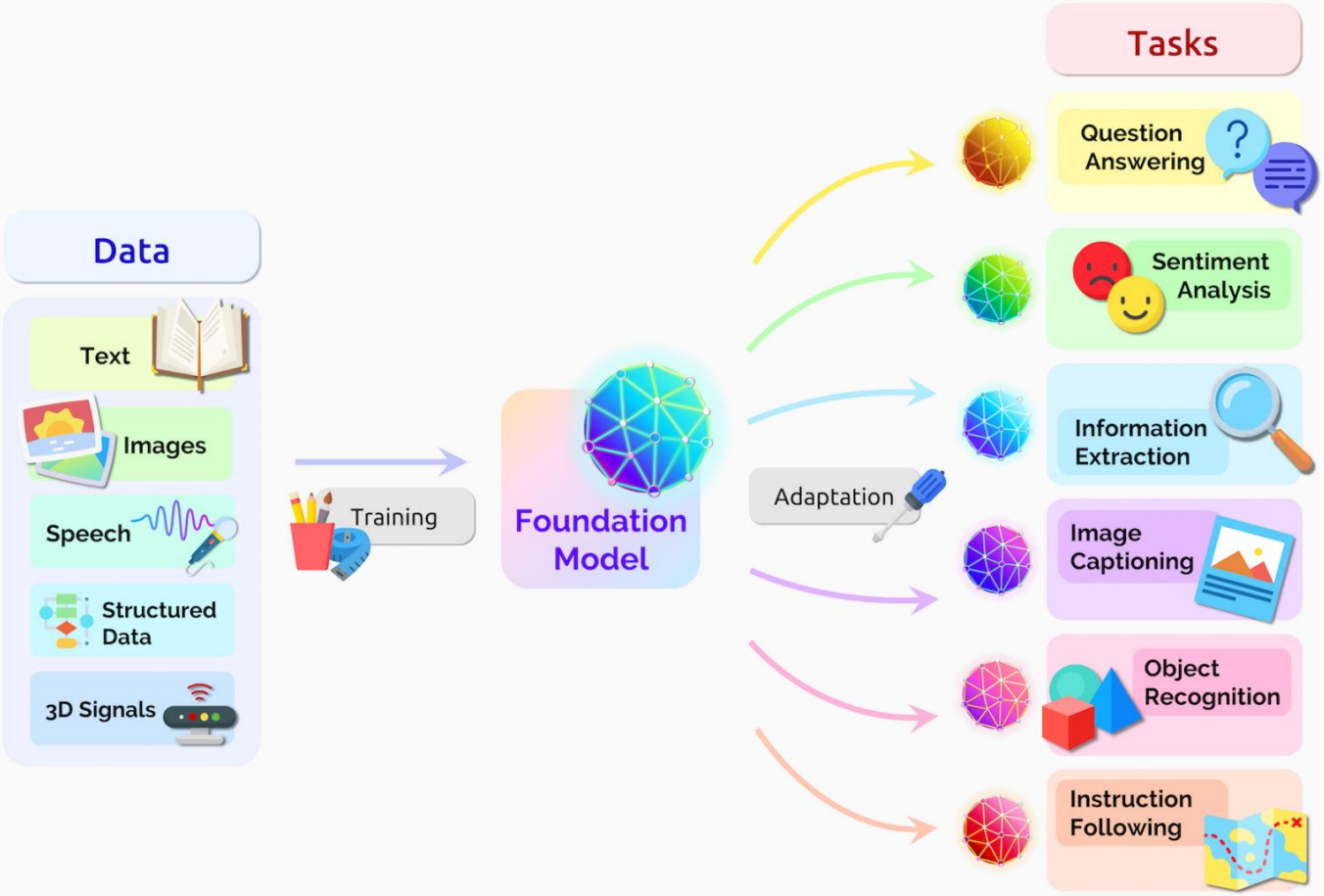
Struggles with architecture-semantics of code execution, assuming unconventional architectural properties, modeling and tracking system state, and others highlighted in paper!

Assume that CMOV is implemented as the cmov x86 instruction, a secure constant-time instruction.

Assume the length of the array is already known publicly and is not sensitive information.

Does this code have timing side channels that can reveal insights into the contents of the elements array? Explain why.

Downstream Usecases



[Liang et al., 2022]

How does accuracy on QuArch translate?

How does accuracy on QuArch translate?

Case Study #1

- **Downstream Task Formulation:**
  - Models instructed to perform hardware-software co-design of memory hierarchy for workload
  - Included decisions about cache hierarchy, tiling strategies, dataflow ordering, memory layout
  - Provided a fixed area and energy budget.
  - No programming; pure reasoning; iterative 10-turn process.

How does accuracy on QuArch translate?

Case Study #1

- **Downstream Task Formulation:**
  - Models instructed to perform hardware-software co-design of memory hierarchy for workload
  - Included decisions about cache hierarchy, tiling strategies, dataflow ordering, memory layout
  - Provided a fixed area and energy budget.
  - No programming; pure reasoning; iterative 10-turn process.

- **Baselines:** Obtained with open-source Gemma 3 27B and Llama 3.3 70B

How does accuracy on QuArch translate?

Case Study #1

- **Downstream Task Formulation:**
  - Models instructed to perform hardware-software co-design of memory hierarchy for workload
  - Included decisions about cache hierarchy, tiling strategies, dataflow ordering, memory layout
  - Provided a fixed area and energy budget.
  - No programming; pure reasoning; iterative 10-turn process.

- **Baselines:** Obtained with open-source Gemma 3 27B and Llama 3.3 70B

- **Finding:** After being fine-tuned on QuArch data, models proposed designs that **successfully met area and energy budget ~40% more often** and achieved **1.86x-1.99x more area-efficient designs**

How does accuracy on QuArch translate?

Case Study #1

- **Downstream Task Formulation:**

Takeaway #5

Knowledge Transfer:
Reasoning elicited by QuArch can lead to improved design outcomes.

- **Baselines:** Obtained with open-source Gemma 3 27B and Llama 3.3 70B

- **Finding:** After being fine-tuned on QuArch data, models proposed designs that **successfully met area and energy budget ~40% more often** and achieved **1.86x-1.99x more area-efficient designs**

How does accuracy on QuArch translate?

Case Study #2

How does accuracy on QuArch translate?

Case Study #2

- **Downstream Task Formulation:**
  - Developing domain-specific review assistant for computer architecture
  - "*Act and think* like an architect—provide me with a constructive review for my ASPLOS conference submission."

	RevExp	RevCon	SigPro	NovSol	Cor	WriQua	RelWor	RobEva	AdvCorASPLOS	OveMer	RanRelOth
Review #1234A	1	2	3	1	2	4	5	1	2	4	3
Review #1234B	3	4	2	4	1	3	3	3	3	3	2
Review #1234C	1	4	1	3	3	2	5	2	3	2	1
Review #1234D	2	2	4	2	4	4	3	2	2	4	3
ArchScholar	3	3	2	4	2	3	2	2	3	2	3

How does accuracy on QuArch translate?

Case Study #2

- **Downstream Task Formulation:**
  - Developing domain-specific review assistant for computer architecture
  - "*Act and think* like an architect—provide me with a constructive review for my ASPLOS conference submission."

	RevExp	RevCon	SigPro	NovSol	Cor	WriQua	RelWor	RobEva	AdvCorASPLOS	OveMer	RanRelOth
Review #1234A	1	2	3	1	2	4	5	1	2	4	3
Review #1234B	3	4	2	4	1	3	3	3	3	3	2
Review #1234C	1	4	1	3	3	2	5	2	3	2	1
Review #1234D	2	2	4	2	4	4	3	2	2	4	3
ArchScholar	3	3	2	4	2	3	2	2	3	2	3

- **Archscholar:**
  - Gives LLMs access to corpus of 50 years of computer architecture/systems research

How does accuracy on QuArch translate?

Case Study #2

	RevExp	RevCon	SigPro	NovSol	Cor	WriQua	RelWor	RobEva	AdvCorASPLOS	OveMer	RanRelOth
Review #1234A	1	2	3	1	2	4	5	1	2	4	3
Review #1234B	3	4	2	4	1	3	3	3	3	3	2
Review #1234C	1	4	1	3	3	2	5	2	3	2	1
Review #1234D	2	2	4	2	4	4	3	2	2	4	3
ArchScholar	3	3	2	4	2	3	2	2	3	2	3

- **Experimental Evaluation:**
  - Submissions given 2 human reviews + 4 LLM-generated reviews (2 with **Archscholar**)
  - Accepted submitters asked to rank reviews in order of preference (0-5, higher is better)

Human						GPT OSS	Human				Qwen 235B	GPT OSS	Human	Qwen 235B	Human	
A	B	C	D	E	F	Arch scholar	G	H	I	J	Base line	Base line	K	Arch scholar	L	M
4.5	4.3	4.0	4.0	3.2	3.2	3.0	3.0	2.7	2.7	2.5	2.4	1.9	1.8	1.3	0.3	0.0

- **Takeaways:**
  - Archscholar models can match the average review in quality
  - Models vary in ability to leverage ArchScholar for more informative reviews

How does accuracy on QuArch translate?

Case Study #2

	RevExp	RevCon	SigPro	NovSol	Cor	WriQua	RelWor	RobEva	AdvCorASPLOS	OveMer	RanRelOth
Review #1234A	1	2	3	1	2	4	5	1	2	4	3
Review #1234B	3	4	2	4	1	3	3	3	3	3	2
Review #1234C	1	4	1	3	3	2	5	2	3	2	1
Review #1234D	2	2	4	2	4	4	3	2	2	4	3
ArchScholar	3	3	2	4	2	3	2	2	3	2	3

- **Experimental Evaluation:**
  - Submissions given 2 human reviews + 4 LLM-generated reviews (2 with **Archscholar**)

These are only 2 of the many potential applications—reach out to explore your use case with the dataset!

A	B	C	D	E	F	Arch scholar	G	H	I	J	Base line	Base line	K	Arch scholar	L	M
4.5	4.3	4.0	4.0	3.2	3.2	3.0	3.0	2.7	2.7	2.5	2.4	1.9	1.8	1.3	0.3	0.0

- **Takeaways:**
  - Archscholar models can match the average review in quality
  - Models vary in ability to leverage ArchScholar for more informative reviews

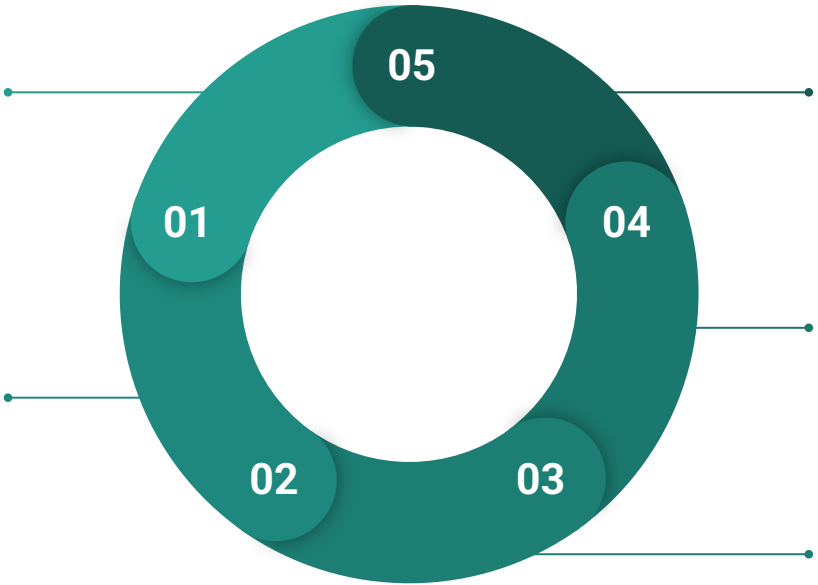
Key Themes for Architecture 2.0

Datasets

What datasets do we need? How we should collect these datasets for architecture research? What metadata should the datasets contain to enable broad usage? How do we create standard data formats from any ML algorithm?

ML Algorithms

How can we learn and apply new ML algorithms to effectively design high-performance/efficient systems? How do we make our community more accessible to ML researchers? How do we embrace ML algorithm design as part of architecture research?



Workforce & Training

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

Tools & Infrastructure

How do we reduce the sim2real gap? What instrumentation mechanisms do we need for creating the datasets? What gym environments do we need to enable data-centric AI? How do we define standard data formats for interoperability?

Best Practices

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

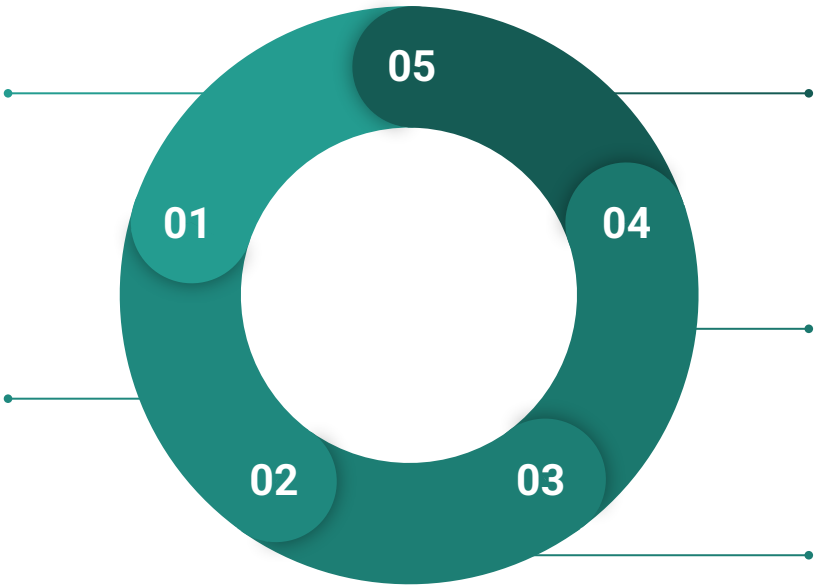
Key Themes for Architecture 2.0

Datasets

What datasets do we need? How we should collect these datasets for architecture research? What metadata should the datasets contain to enable broad usage? How do we create standard data formats from any ML algorithm?

ML Algorithms

How can we learn and apply new ML algorithms to effectively design high-performance/efficient systems? How do we make our community more accessible to ML researchers? How do we embrace ML algorithm design as part of architecture research?



Workforce & Training

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

Tools & Infrastructure

How do we reduce the sim2real gap? What instrumentation mechanisms do we need for creating the datasets? What gym environments do we need to enable data-centric AI? How do we define standard data formats for interoperability?

Best Practices

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

Data-Centric Approaches Across the Computing Stack

SWE

Algorithms

Compilers

CompArch

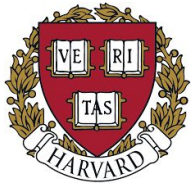
RTL

EDA

Circuits

Architecture 2.0

The era when we use AI/ML methods to
(1) minimize human intervention,
(2) build complex, efficient systems,
(3) in a shorter time frame.



Join the community!

https://quarch.ai

contact.quarch@gmail.com