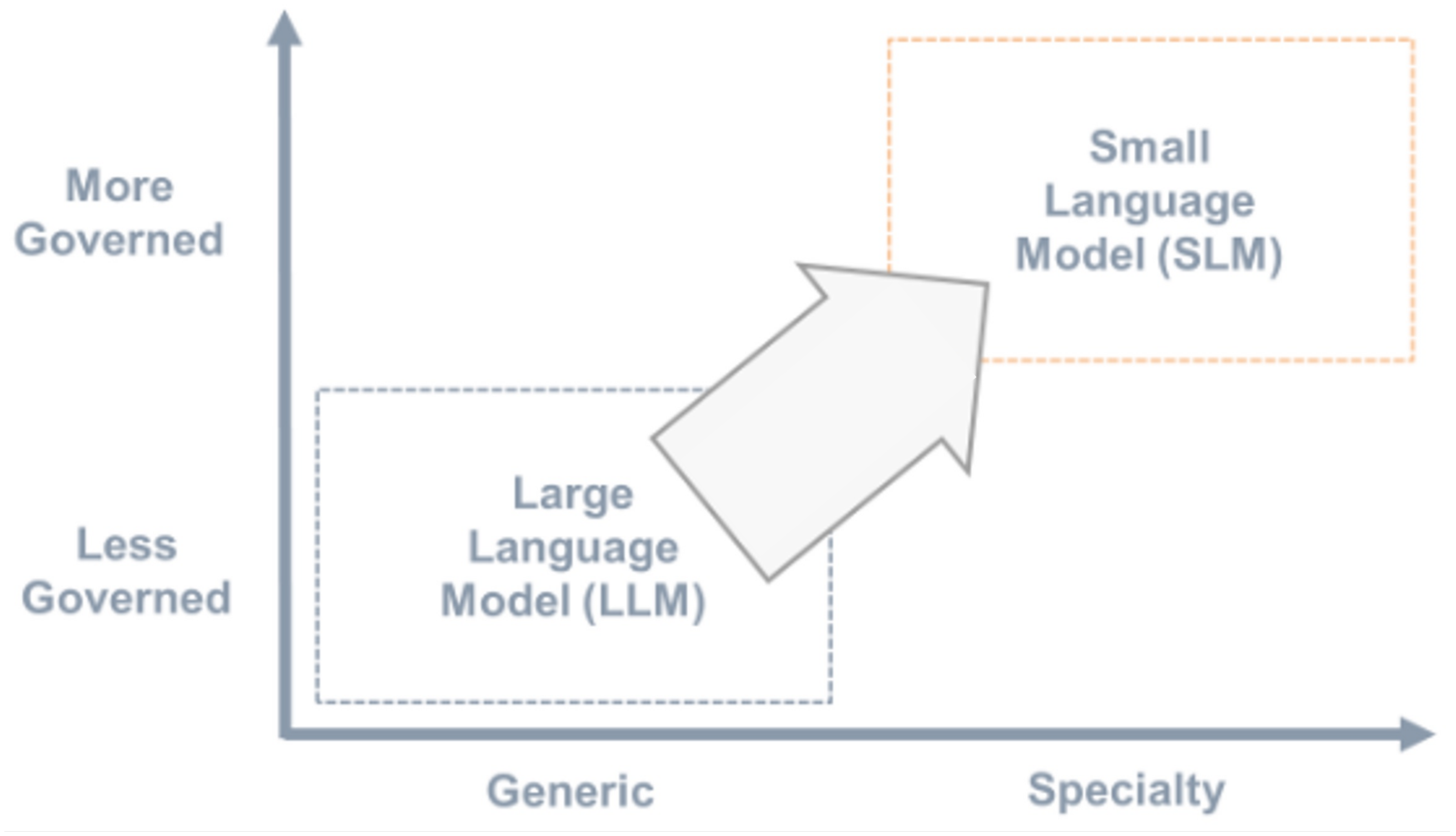
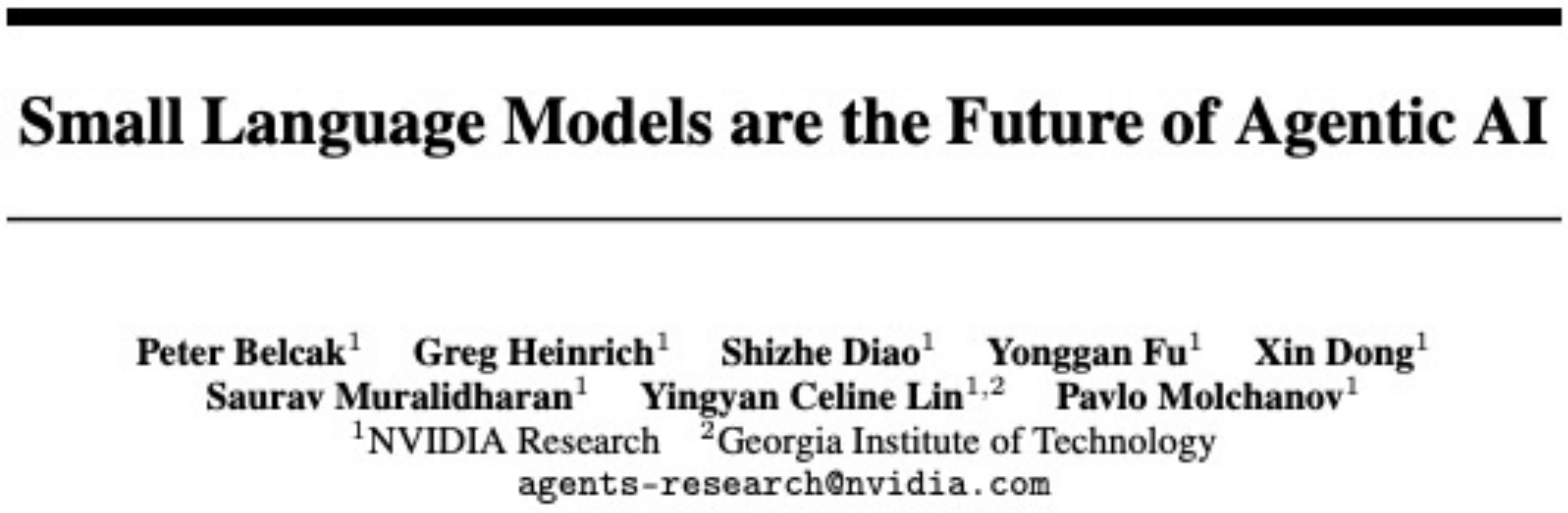
Small Language Models are the Future of Agentic AI



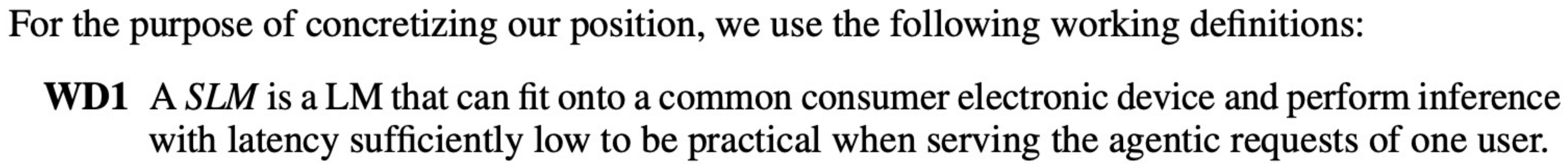


Small Language Models are the Future of Agentic AI

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

It is a position paper. You may agree or disagree.
The point is the discussion that ensues.

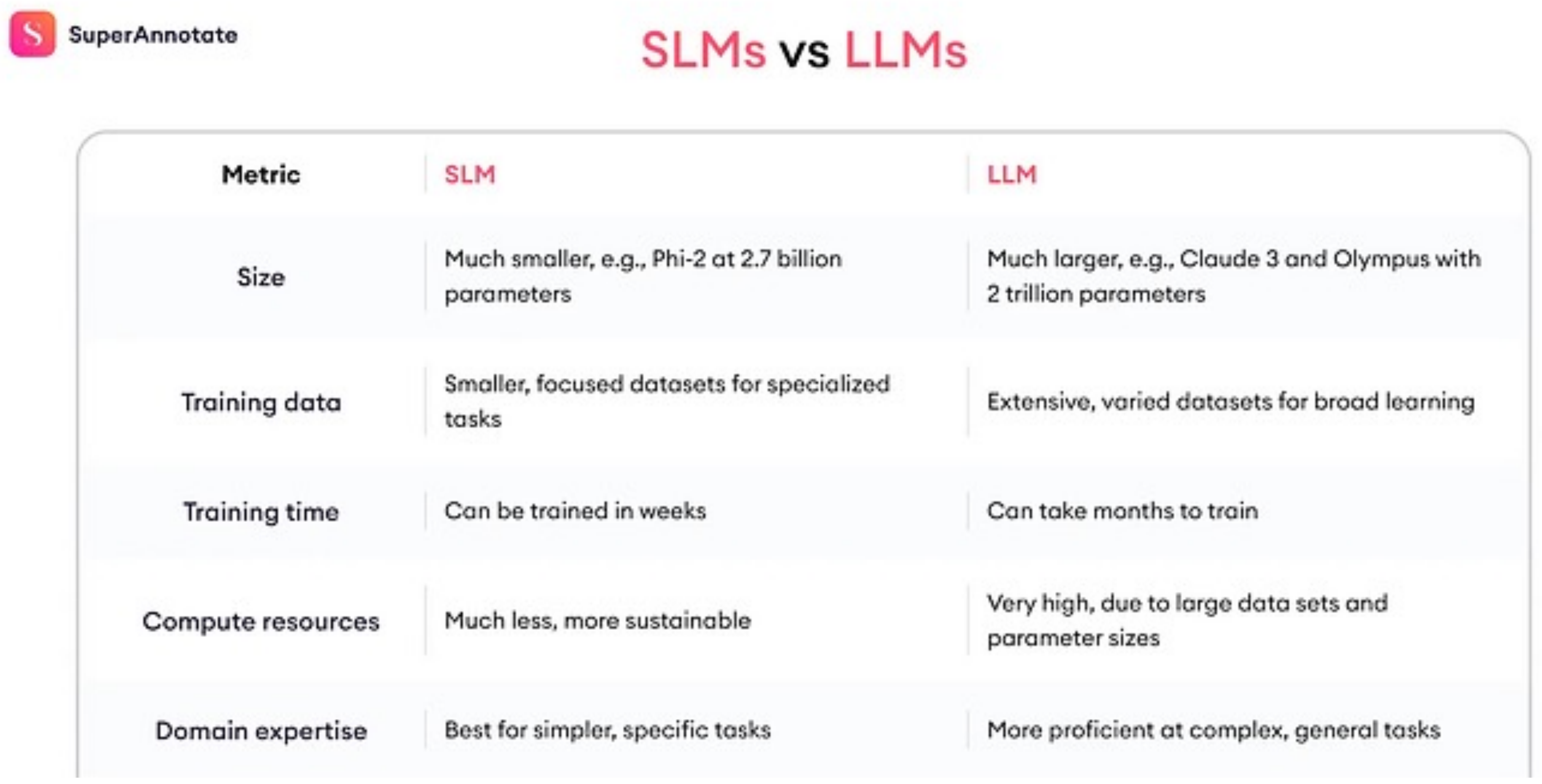
SLM: Working Definition



For the purpose of concretizing our position, we use the following working definitions:

WD1 A *SLM* is a LM that can fit onto a common consumer electronic device and perform inference with latency sufficiently low to be practical when serving the agentic requests of one user.

We note that as of 2025, we would be comfortable with considering most models **below 10B** parameters in size to be SLMs.



SuperAnnotate

SLMs vs LLMs

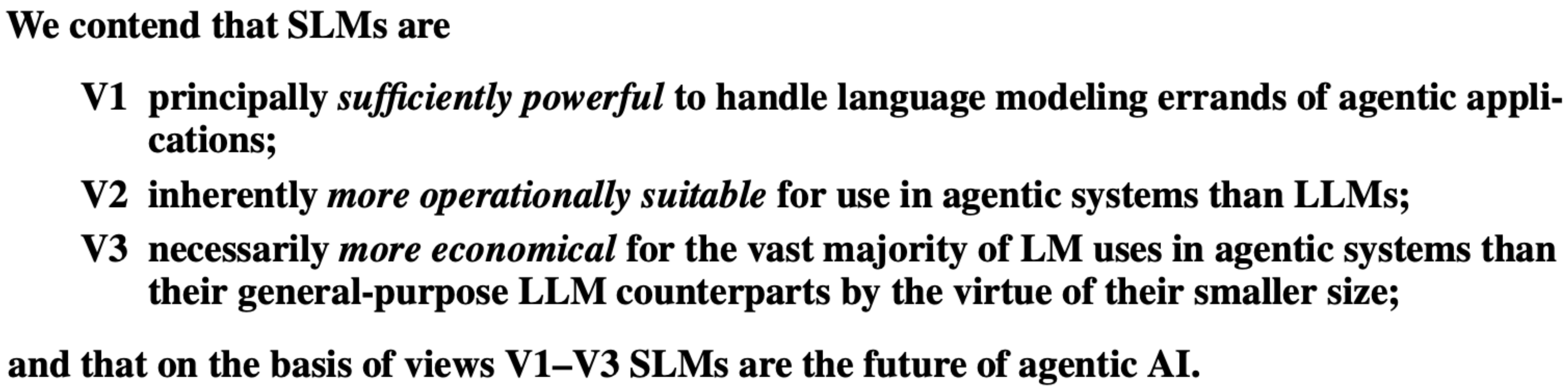
Metric	SLM	LLM
Size	Much smaller, e.g., Phi-2 at 2.7 billion parameters	Much larger, e.g., Claude 3 and Olympus with 2 trillion parameters
Training data	Smaller, focused datasets for specialized tasks	Extensive, varied datasets for broad learning
Training time	Can be trained in weeks	Can take months to train
Compute resources	Much less, more sustainable	Very high, due to large data sets and parameter sizes
Domain expertise	Best for simpler, specific tasks	More proficient at complex, general tasks

SuperAnnotate

SLMs vs LLMs

Metric	SLM	LLM
Inference	Can run locally on devices like Raspberry Pi, no internet needed	Requires specialized hardware like GPUs, often needs internet
Latency	Lower, responds quickly due to smaller size	Higher, can be slow depending on the task
Cost	Lower, cheaper to operate	Higher, due to larger model size and compute needs
Control	Easier to manage, can be run and updated on personal servers	Dependent on model builders, potential for model drift
Performance	Adequate for less complex tasks, may struggle with harder ones	Excels in handling complex tasks, offering broader capabilities

SLM: The Position



We contend that SLMs are

V1 **principally *sufficiently powerful* to handle language modeling errands of agentic applications;**

V2 **inherently *more operationally suitable* for use in agentic systems than LLMs;**

V3 **necessarily *more economical* for the vast majority of LM uses in agentic systems than their general-purpose LLM counterparts by the virtue of their smaller size;**

and that on the basis of views V1–V3 SLMs are the future of agentic AI.

Motivation for the Position

- Feedback from enterprises pioneering the use of AI
  - Companies are looking for ways to increase their margins without compromising the performance of their agents (”same performance but cheaper”)
  - The majority of agentic subtasks in deployed agentic systems are **repetitive, scoped, and non-conversational**—calling for models that are efficient, predictable, and inexpensive. We need specialist than generalist.
  - Edge AI on device

Why SLMs are preferable?

- Lower latency (10-30X lower)
- Reduced memory and computational requirements
- Significantly lower operational costs
- Promoting responsible and sustainable AI deployment

all while maintaining adequate task performance in constrained domains.

Two Modes of Agency

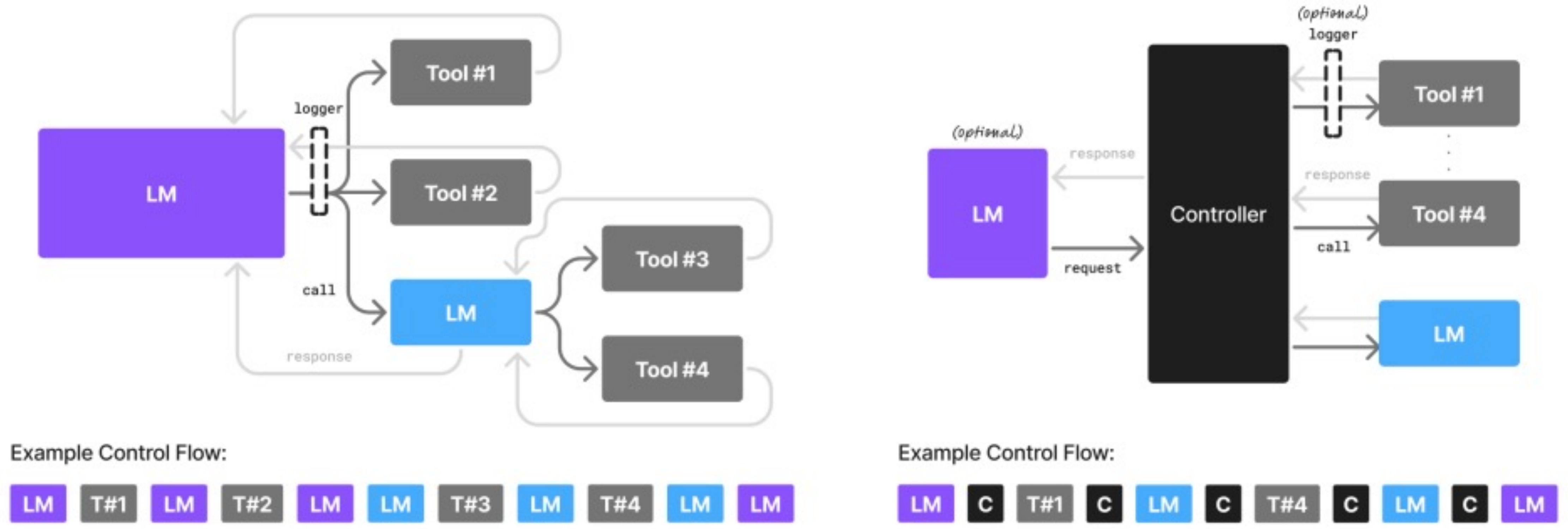
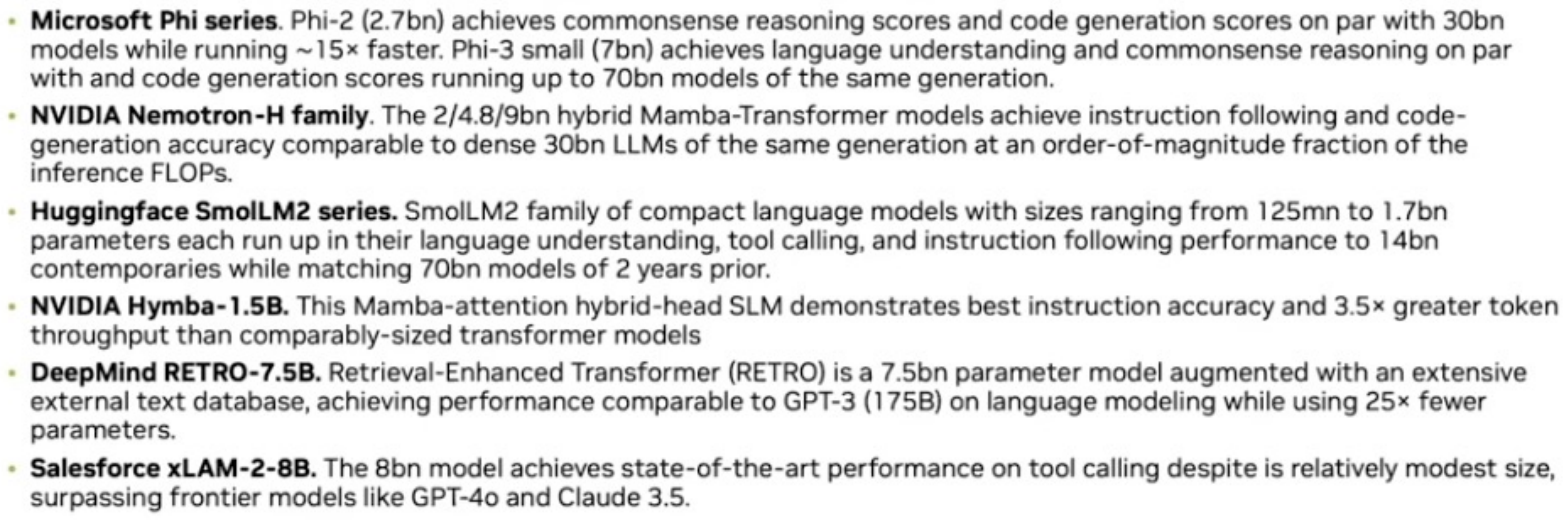


Figure 1: An illustration of agentic systems with different modes of agency. *Left: Language model agency.* The language model acts both as the HCI and the orchestrator of tool calls to carry out a task. *Right: Code agency.* The language model fills the role of the HCI (optionally) while a dedicated controller code orchestrates all interactions.

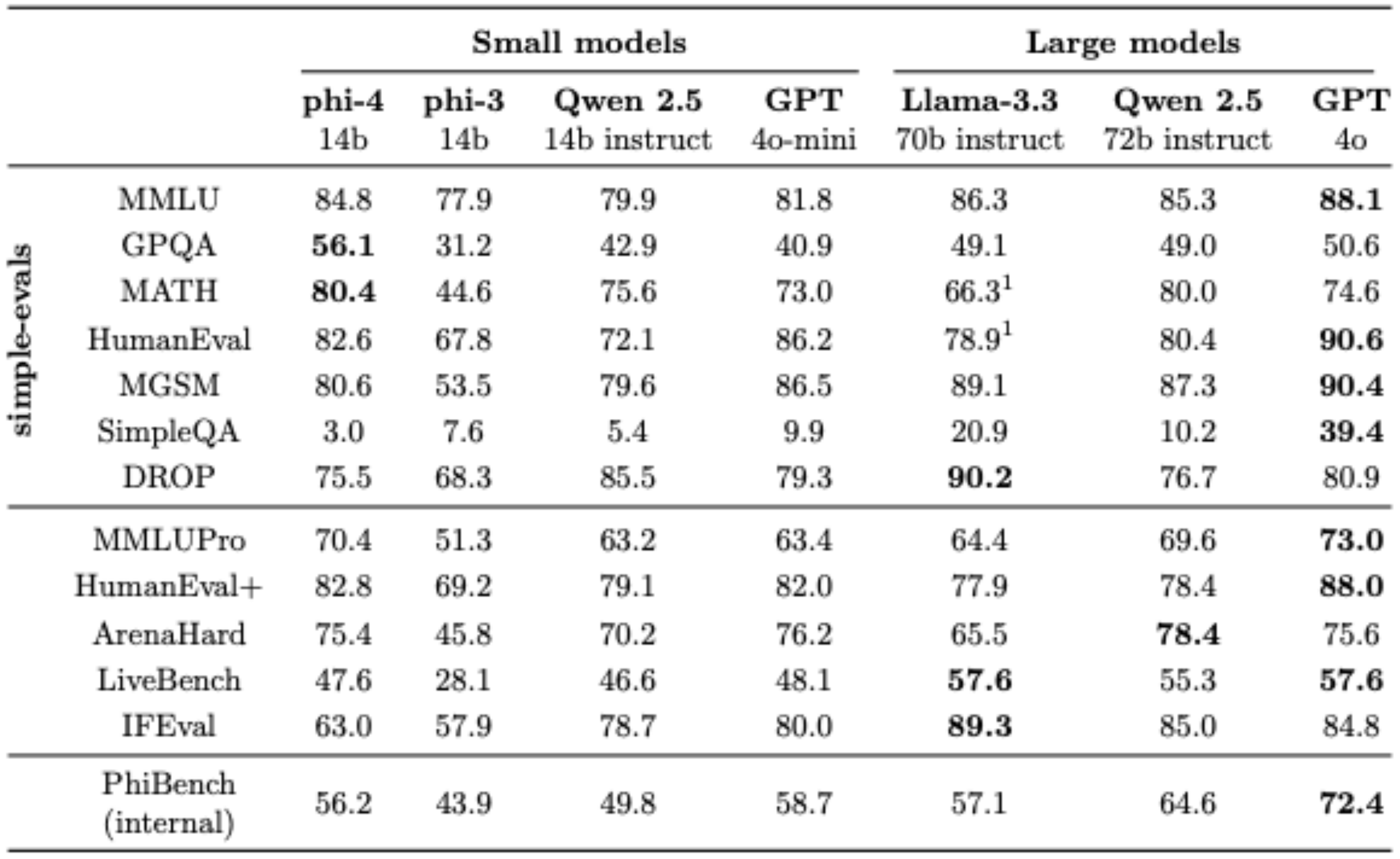
SLMs are “sufficiently powerful”

SLMs are sufficiently powerful to take the place of LLMs in agentic systems:



- **Microsoft Phi series**. Phi-2 (2.7bn) achieves commonsense reasoning scores and code generation scores on par with 30bn models while running ~15× faster. Phi-3 small (7bn) achieves language understanding and commonsense reasoning on par with and code generation scores running up to 70bn models of the same generation.
- **NVIDIA Nemotron-H family**. The 2/4.8/9bn hybrid Mamba-Transformer models achieve instruction following and code-generation accuracy comparable to dense 30bn LLMs of the same generation at an order-of-magnitude fraction of the inference FLOPs.
- **Huggingface SmolLM2 series.** SmolLM2 family of compact language models with sizes ranging from 125mn to 1.7bn parameters each run up in their language understanding, tool calling, and instruction following performance to 14bn contemporaries while matching 70bn models of 2 years prior.
- **NVIDIA Hymba-1.5B.** This Mamba-attention hybrid-head SLM demonstrates best instruction accuracy and 3.5× greater token throughput than comparably-sized transformer models
- **DeepMind RETRO-7.5B.** Retrieval-Enhanced Transformer (RETRO) is a 7.5bn parameter model augmented with an extensive external text database, achieving performance comparable to GPT-3 (175B) on language modeling while using 25× fewer parameters.
- **Salesforce xLAM-2-8B.** The 8bn model achieves state-of-the-art performance on tool calling despite is relatively modest size, surpassing frontier models like GPT-4o and Claude 3.5.

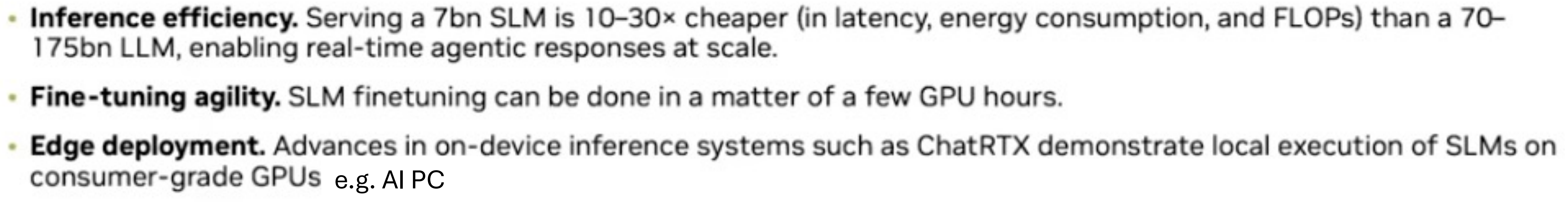
Phi-4 Performance



			Small models				Large models		
			phi-4 14b	**phi-3** 14b	**Qwen 2.5** 14b instruct	**GPT** 4o-mini	**Llama-3.3** 70b instruct	**Qwen 2.5** 72b instruct	**GPT** 4o
	simple-evals	MMLU	84.8	77.9	79.9	81.8	86.3	85.3	**88.1**
Scientific		GPQA	**56.1**	31.2	42.9	40.9	49.1	49.0	50.6
		MATH	**80.4**	44.6	75.6	73.0	66.3[1]	80.0	74.6
Coding		HumanEval	82.6	67.8	72.1	86.2	78.9[1]	80.4	**90.6**
Math		MGSM	80.6	53.5	79.6	86.5	89.1	87.3	**90.4**
		SimpleQA	3.0	7.6	5.4	9.9	20.9	10.2	**39.4**
Reading reasoning		DROP	75.5	68.3	85.5	79.3	**90.2**	76.7	80.9
		MMLUPro	70.4	51.3	63.2	63.4	64.4	69.6	**73.0**
Coding		HumanEval+	82.8	69.2	79.1	82.0	77.9	78.4	**88.0**
Chatbot arena		ArenaHard	75.4	45.8	70.2	76.2	65.5	**78.4**	75.6
Maths, coding, reasoning		LiveBench	47.6	28.1	46.6	48.1	**57.6**	55.3	**57.6**
Instruction-following		IFEval	63.0	57.9	78.7	80.0	**89.3**	85.0	84.8
		PhiBench (internal)	56.2	43.9	49.8	58.7	57.1	64.6	**72.4**

Table 1: Performance of phi-4 on a set of standard benchmarks. The first set of benchmarks uses OpenAI's SIMPLE-EVALS framework [Ope24b], specifying the prompts/extraction/temperature=0.5. We compare to small models of similar inference cost, as well as to larger models.

SLMs are more economical



- **Inference efficiency.** Serving a 7bn SLM is 10–30× cheaper (in latency, energy consumption, and FLOPs) than a 70–175bn LLM, enabling real-time agentic responses at scale.
- **Fine-tuning agility.** SLM finetuning can be done in a matter of a few GPU hours.
- **Edge deployment.** Advances in on-device inference systems such as ChatRTX demonstrate local execution of SLMs on consumer-grade GPUs

e.g. AI PC

What's an AI PC?

AI PCs tackle complex AI tasks running on your device or in the cloud while boosting performance for your daily work by:

Accelerated performance for demanding AI and other applications

Professional graphic design and visual effects

Rapid rendering and large-scale simulation

AI PCs are engineered to handle data-intensive AI operations such as:

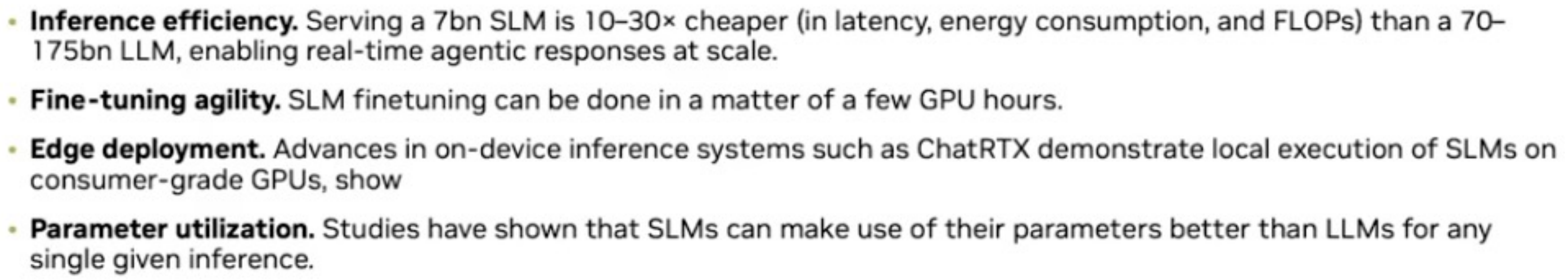
Machine learning and LLM training

High-end graphics processing

Generative AI creation

Analyzing large datasets

SLMs are more economical



- **Inference efficiency.** Serving a 7bn SLM is 10–30× cheaper (in latency, energy consumption, and FLOPs) than a 70–175bn LLM, enabling real-time agentic responses at scale.
- **Fine-tuning agility.** SLM finetuning can be done in a matter of a few GPU hours.
- **Edge deployment.** Advances in on-device inference systems such as ChatRTX demonstrate local execution of SLMs on consumer-grade GPUs, show
- **Parameter utilization.** Studies have shown that SLMs can make use of their parameters better than LLMs for any single given inference.

Parameter Utilization

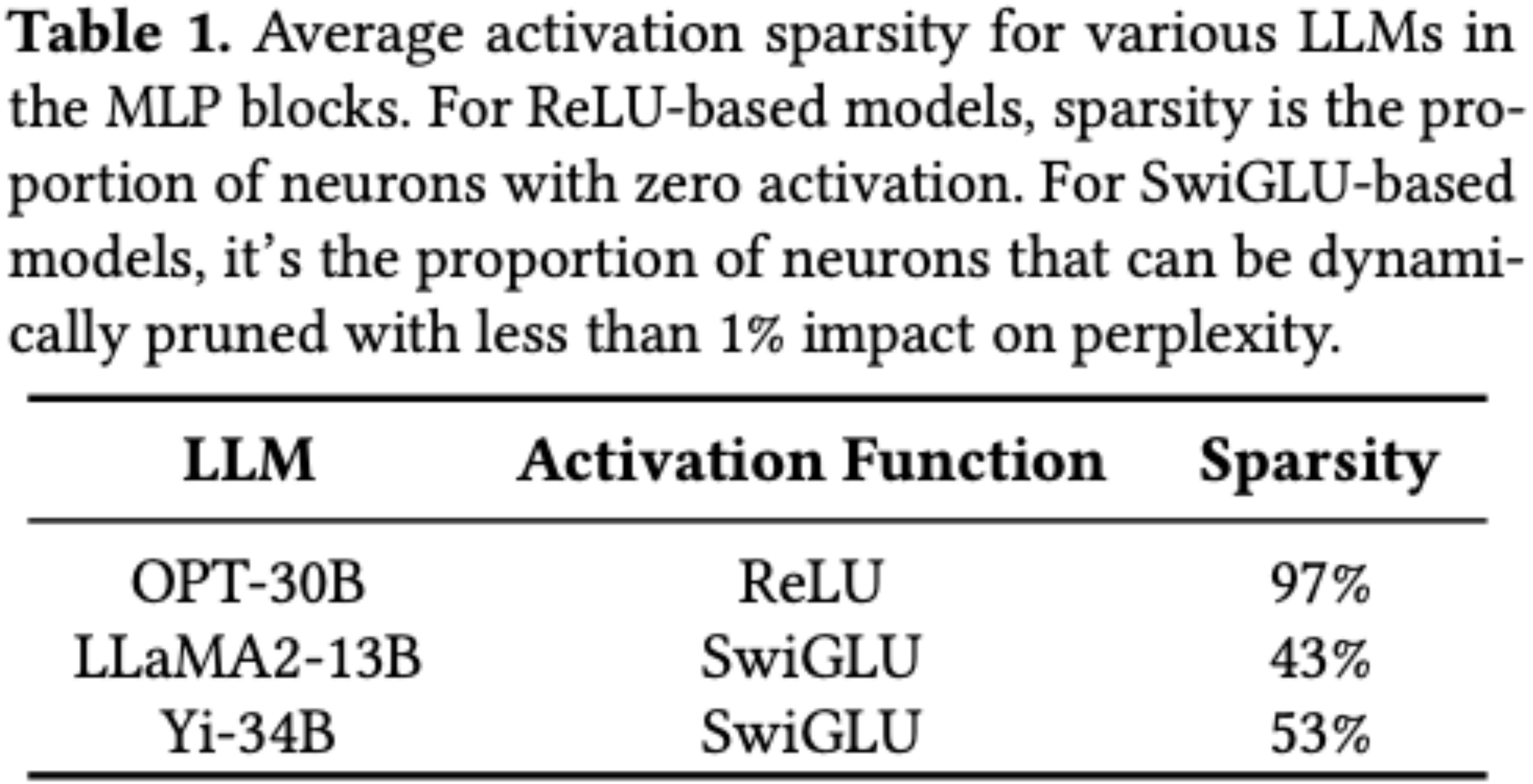
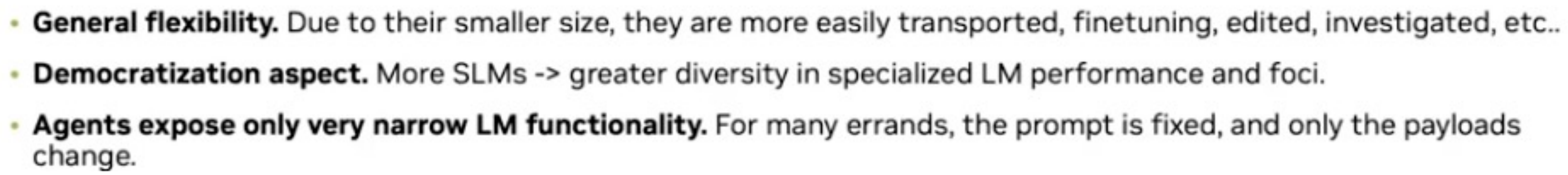


Table 1. Average activation sparsity for various LLMs in the MLP blocks. For ReLU-based models, sparsity is the proportion of neurons with zero activation. For SwiGLU-based models, it's the proportion of neurons that can be dynamically pruned with less than 1% impact on perplexity.

LLM	**Activation Function**	**Sparsity**
OPT-30B	ReLU	97%
LLaMA2-13B	SwiGLU	43%
Yi-34B	SwiGLU	53%

SLMs are “operational more suitable”



- **General flexibility.** Due to their smaller size, they are more easily transported, finetuning, edited, investigated, etc..
- **Democratization aspect.** More SLMs -> greater diversity in specialized LM performance and foci.
- **Agents expose only very narrow LM functionality.** For many errands, the prompt is fixed, and only the payloads change.

Agents expose only very narrow LM functionality

- **Task specialization:** Agentic systems are designed for specific tasks like automating workflows, interacting with tools, or generating specific kinds of data, rather than general conversation.
- **Structured workflows:** Many agentic interactions involve structured data formats, predefined workflows, and predetermined actions, requiring less complex reasoning than a general-purpose LLM provides.

SLMs are “operational more suitable”

- **General flexibility.** Due to their smaller size, they are more easily transported, finetuning, edited, investigated, etc..
- **Democratization aspect.** More SLMs -> greater diversity in specialized LM performance and foci.
- **Agents expose only very narrow LM functionality.** For many errands, the prompt is fixed, and only the payloads change.
- **Agentic interactions necessitate close behavioral alignment.** Any code/structured output needs to be correct more than ever.
- **Agentic systems are naturally heterogeneous.** Any agent is comprised of many blocks.

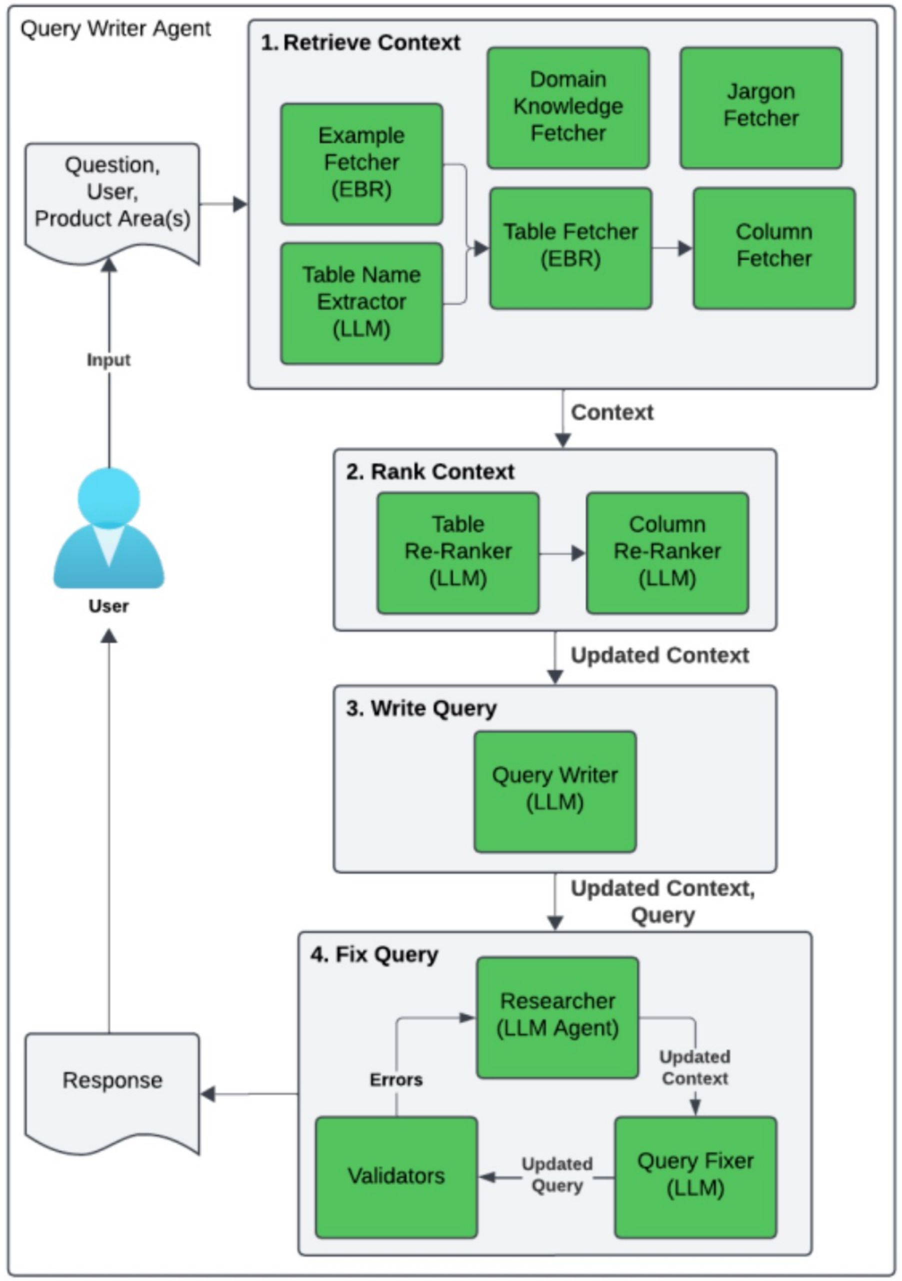
Agentic systems are naturally heterogeneous

- **Multi-agent architecture**: More complex agentic systems can be composed of multiple individual AI agents that work together to solve a problem. These agents may have different levels of expertise or be arranged in a hierarchical structure to handle specialized tasks, with a central "conductor" agent for orchestration.
- **Diverse tools and APIs**: To act in the real world, agents connect to various external tools and APIs, such as web search, databases, or third-party software. The ability to call on these diverse, real-world capabilities is a critical aspect of an agent's heterogeneous nature.

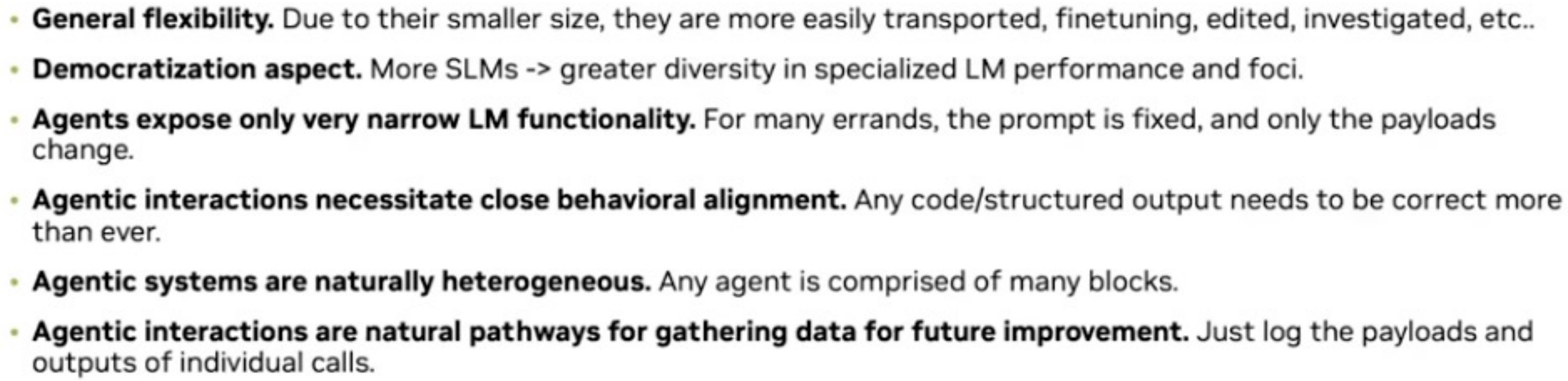
Agentic systems are naturally heterogeneous

Let's look at Query Writer Agent architecture

Table name extractor: SLM
Re-ranker: SLM
Query writer: text to SQL -> SLM
Researcher: LLM
Query Fixer: SLM



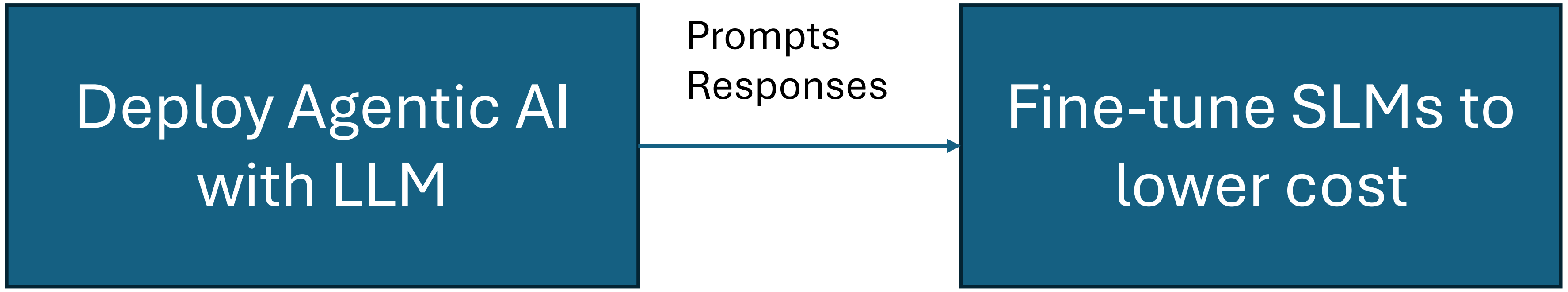
SLMs are “operational more suitable”



- **General flexibility.** Due to their smaller size, they are more easily transported, finetuning, edited, investigated, etc..
- **Democratization aspect.** More SLMs -> greater diversity in specialized LM performance and foci.
- **Agents expose only very narrow LM functionality.** For many errands, the prompt is fixed, and only the payloads change.
- **Agentic interactions necessitate close behavioral alignment.** Any code/structured output needs to be correct more than ever.
- **Agentic systems are naturally heterogeneous.** Any agent is comprised of many blocks.
- **Agentic interactions are natural pathways for gathering data for future improvement.** Just log the payloads and outputs of individual calls.

Agentic interactions are natural pathways for gathering data

A listener decorating the tool/model call interface can **gather specialized instruction data** that can later be used to produce a fine-tune an expert SLM and lower the cost of that call in the future



Alternative Views on LLM

LLM generalists will always have the advantage of more general language understanding

-> TRUE, but a SLM is sufficient for a narrow scope of tasks to be performed

LLM inference will still be cheaper because of their centralization

-> Could be true, but depending on ROI per use case (e.g. payload)

Alternative Views

Both the agentic world utilizing SLMs and agentic world utilizing LLMs are equally possible worlds

➔The most pragmatic future likely involves the heterogeneous architecture, where SLMs handle the "heavy lifting" of routine tasks, with LLMs reserved for complex exceptions.

➔This balances the strengths of both approaches and makes agentic AI more scalable and economical for enterprises.

Barriers to Adoption of SLMs

- **Sunk cost:** Large amounts of upfront investments into centralized LLM inference infrastructure
- **Lack of specialized benchmarks:** Use of generalist benchmarks in SLM training, design and evaluation
- **Lack of popular awareness**

Six Steps: LLM-to-SLM Agent Conversion

- **Step 1. Secure usage data collection:** Log the payloads and output for every LM call.
  - Input prompts
  - Output responses
  - Contents of tool calls
  - Latency metrics
- **Step 2.** Data curation and filtering: Anonymize all data and filter out cancelled/failed LM calls
  - 10K-100K for fine-tuning SLM
  - Remove application-specific sensitive data such as PPI or PHI to prevent data leakage

Six Steps: LLM-to-SLM Agent Conversion

- **Step 3.** Task clustering: Subdivide your data into blocks based on tasks performed
  - Unsupervised clustering on prompts and actions to identify recurring patterns of requests or internal agent operation
  - E.g. user intention classification, data extraction, summarization of specific document types, code generation

Six Steps: LLM-to-SLM Agent Conversion

- **Step 4.** SLM selection: Choose the appropriate SLM for your computational budget and required generalist performance. Criteria for selection:
  - Instruction following
  - Reasoning
  - Context window size
  - Performance on relevant benchmark for specific task types
  - License
  - Deployment footprint

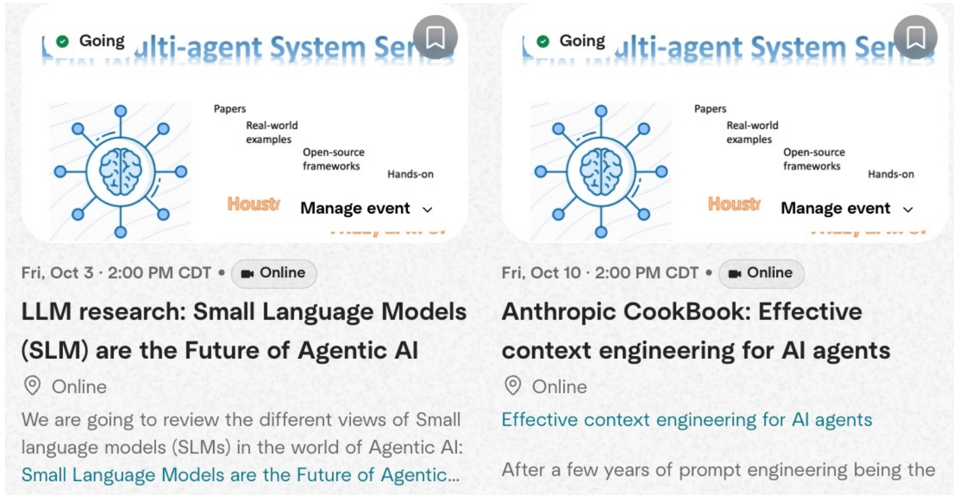
Six Steps: LLM-to-SLM Agent Conversion

- **Step 5.** Specialized SLM fine-tuning: Fine-tune on your curated data
  - PEFT techniques: LoRA and QLoRA
  - Knowledge distillation from a powerful generalist LLM on task-specific data
- **Step 6.** Iteration and refinement: “Jump back to step 2 – Data Curation if needed”
  - Retrain the SLMs periodically with new data to maintain performance and adapt to evolving usage patterns

A heterogeneous architecture involving LLMs and SLMs are the Future of Agentic AI

SLMs handle the "heavy lifting" of routine tasks
LLMs reserved for complex exceptions.

Upcoming meetups – Open to proposals and guest speakers!



Fri, Oct 3 · 2:00 PM CDT • Online

LLM research: Small Language Models (SLM) are the Future of Agentic AI

Online

We are going to review the different views of Small language models (SLMs) in the world of Agentic AI: Small Language Models are the Future of Agentic...

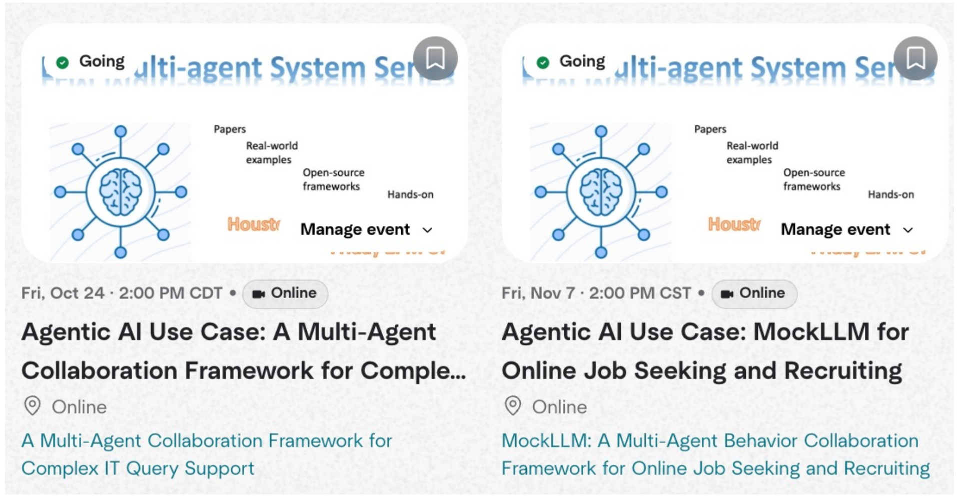
Fri, Oct 10 · 2:00 PM CDT • Online

Anthropic CookBook: Effective context engineering for AI agents

Online

Effective context engineering for AI agents

After a few years of prompt engineering being the



Fri, Oct 24 · 2:00 PM CDT • Online

Agentic AI Use Case: A Multi-Agent Collaboration Framework for Comple...

Online

A Multi-Agent Collaboration Framework for Complex IT Query Support

Fri, Nov 7 · 2:00 PM CST • Online

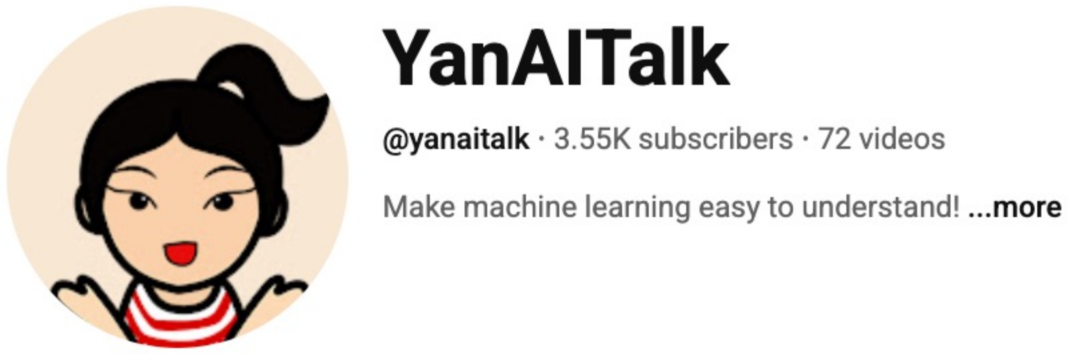
Agentic AI Use Case: MockLLM for Online Job Seeking and Recruiting

Online

MockLLM: A Multi-Agent Behavior Collaboration Framework for Online Job Seeking and Recruiting

Slides posted at:
https://github.com/YanXuHappygela/LLM-reading-group

Recordings posted at:



YanAITalk

@yanaitalk · 3.55K subscribers · 72 videos

Make machine learning easy to understand! ...more