Microsoft Learn Student Ambassadors

Small Language Models,

Developing AI for Sustainable Future and Their Applications

Charunthon Limseelo (@boatchrnthn)
Microsoft Learn Student Ambassador | Thai Student Tech Community Lead
Computer Engineering, King Mongkut's University of Technology Thonburi

#MSFTStudentambassadors #SDGs

Retrieved from A Comprehensive Survey of Small Language Models in the Era of Large Language Models: Techniques, Enhancements, Applications, Collaboration with LLMs, and Trustworthiness

Charunthon Limseelo (Boat)

Beta Microsoft Learn Student Ambassadors (Thai Student Tech Lead) and Google Developer Group (GDG Cloud Bangkok) Member

+ Microsoft Certified Office Specialist (Microsoft Excel)
+ Open-source small AI and NLP Interest, with BDE Applications for business aspect
+ Applied Skills Challenger (Azure AI Document Intelligence and NLP)

Charunthon Limseelo

@boatchrnthn

Charunthon Limseelo

Boat Charunthon (boatchrnthn)

Acknowledgement To All AI Leads and Specialists

Engineering, Research-Based, and Practical Leads of AI

Business, Financial, Commercial-based, and Daily Users of AI

3 Tracks will be discussed in **30 Minutes**

1

Pain Points of Using Large Language Models

2

Small Language Models
Capabilities + Use Cases

3

Microsoft Learn Student Ambassadors:
Pathway to New Models

1

Pain points from using **Large Language Models**

Proprietary

(adj.) used, made, or marketed by one having the exclusive legal right, privately owned and managed and run as a **profit-making** organization, Close-Source

Open-Source

(adj.) having the source code **freely available** for possible modification and redistribution, along with publicly available for use by the community at large

Uptime over the past 30 days. View historical uptime.

API	30 days ago	99.46 % uptime	Today — Major Outage
ChatGPT	30 days ago	99.43 % uptime	Today — Major Outage
Sora	30 days ago	98.53 % uptime	Today — Major Outage
Playground	30 days ago	99.96 % uptime	Today — Major Outage
Labs	30 days ago	99.96 % uptime	Today — Major Outage

```
ChatGPT is currently unavailable.

Status: Identified - We have identified the issue and are working to roll out a fix.
```

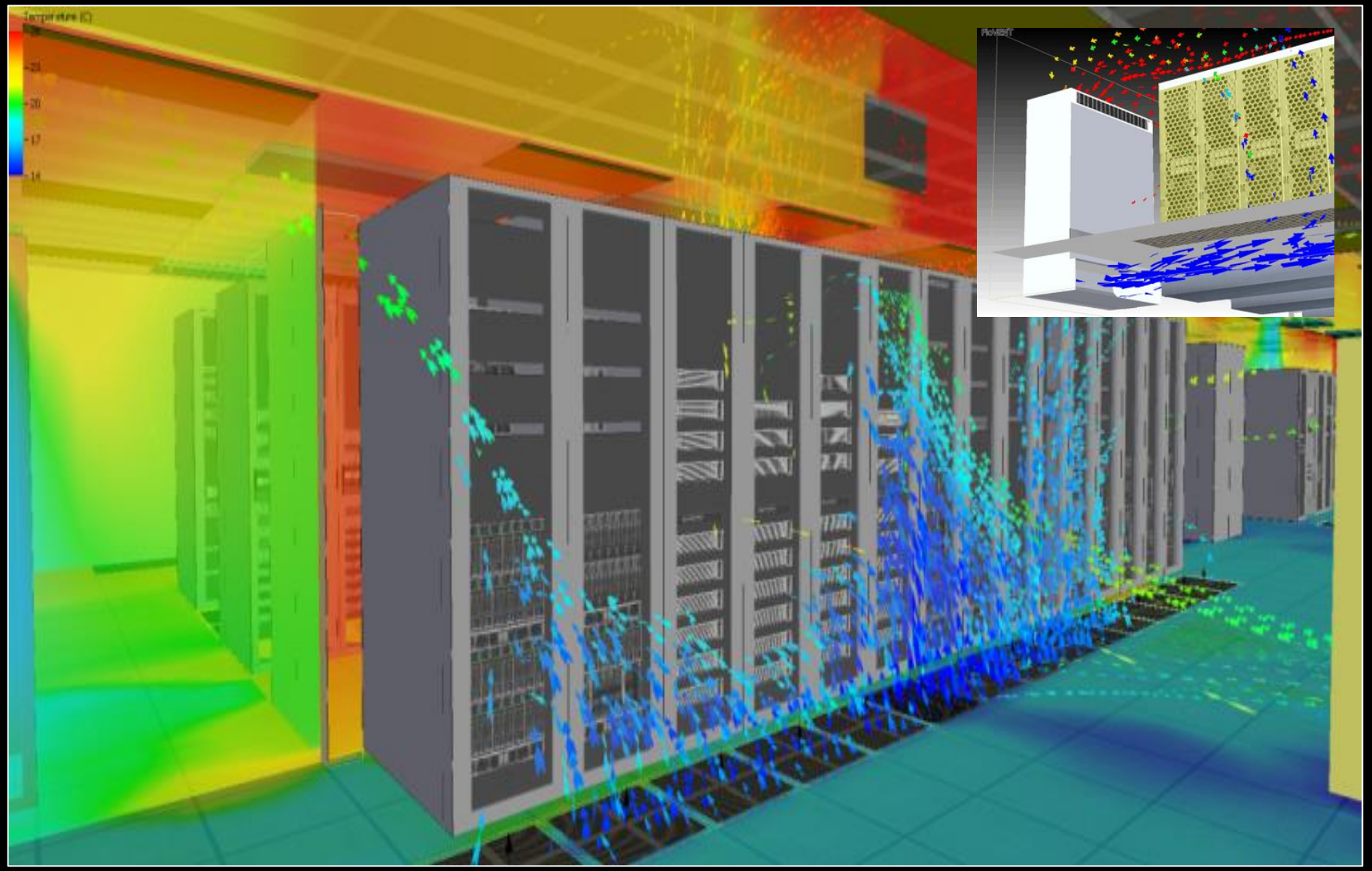
Major Outage Incident of OpenAI ecosystem on December 12th, 2024. (Dec 11 by US Time Zone)

11 Dec 2024

✖ Major outage 4 hrs 10 mins

RELATED

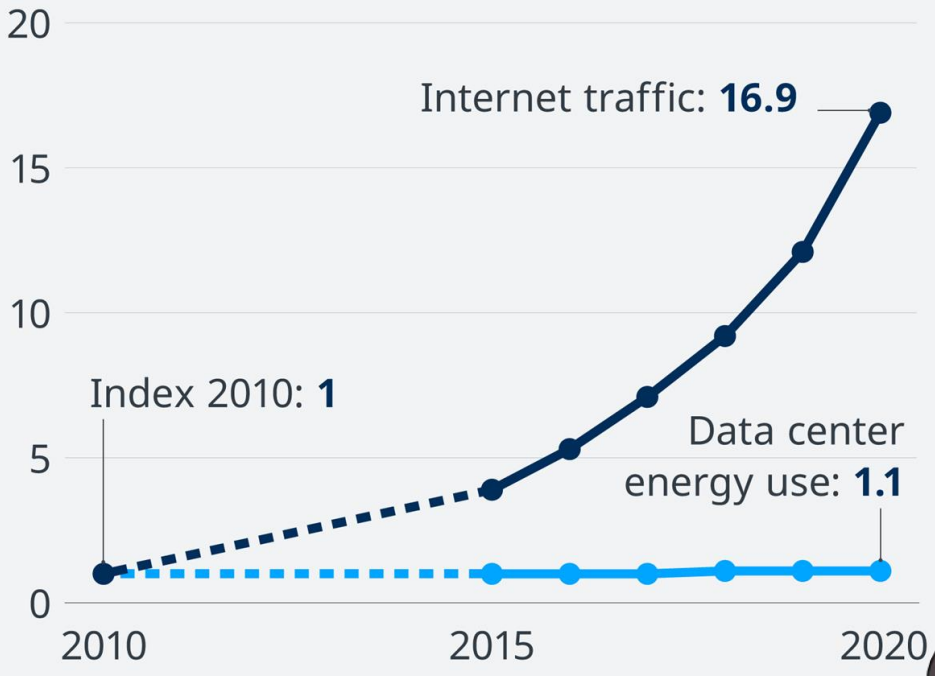
API, ChatGPT & Sora Facing Issues



Thermal Simulation Software on Visualizing Heat Flow of Datacenter

Global trends in internet traffic and data center energy use

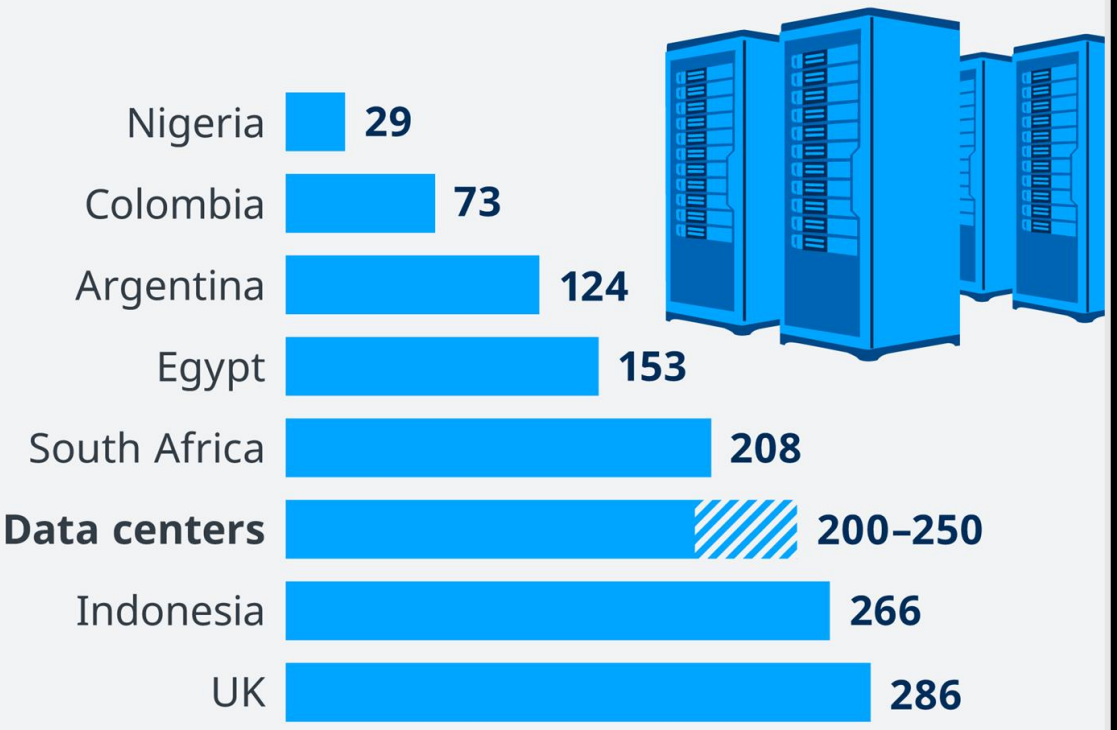
Internet traffic and data center energy use compared to 2010



Source: IEA

Data centers use more electricity than entire countries

Domestic electricity consumption of selected countries vs. data centers in 2020 in TWh

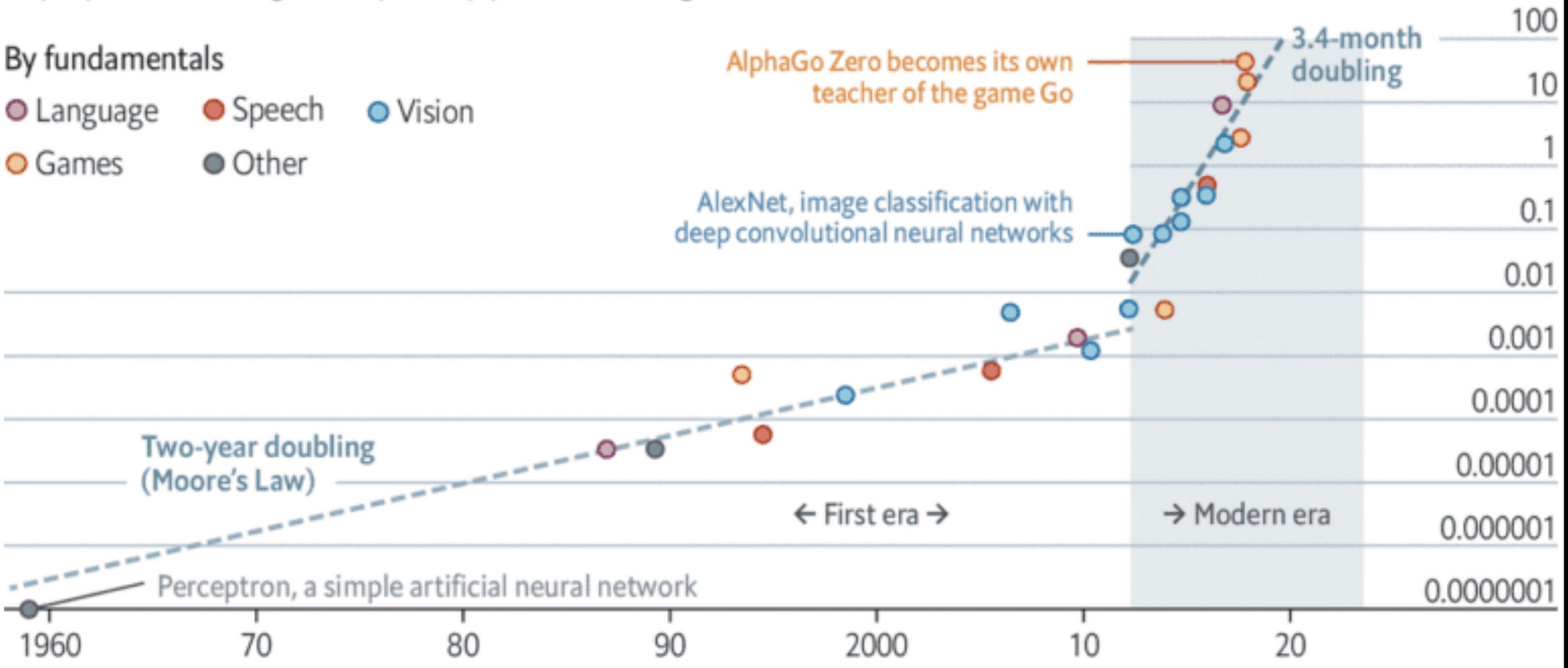


Source: Enerdata, IEA

Deep and steep

Computing power used in training AI systems

Days spent calculating at one petaflop per second*, log scale

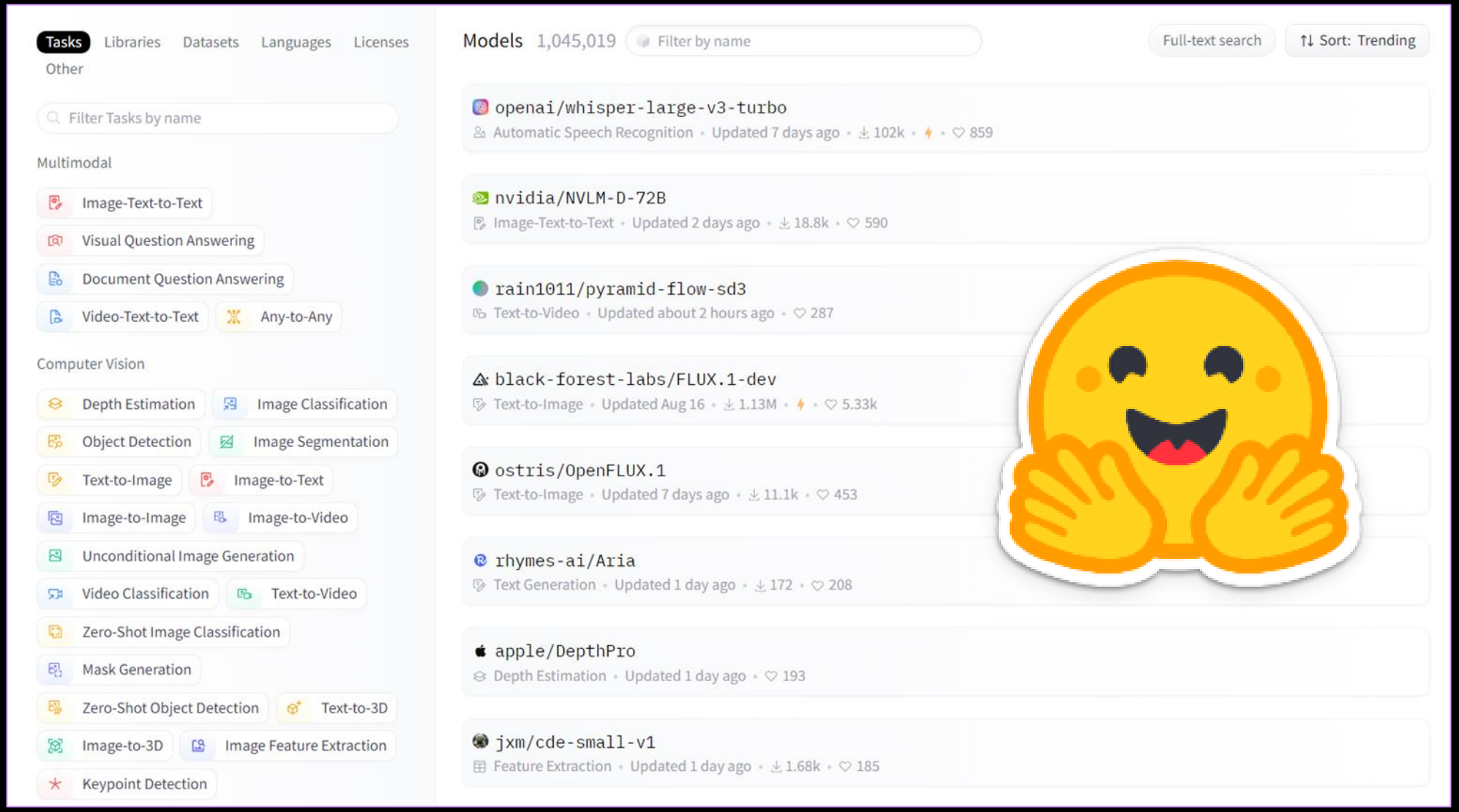


Source: OpenAI

*1 petaflop=10^{15} calculations

The Economist


Climate change is the defining issue of our generation, and addressing it requires swift, collective action and technological innovation. We are committed to meeting our own goals while enabling others to do the same. That means taking responsibility for our operational footprint and accelerating progress through technology.
- **Satya Nadella**, CEO of Microsoft



Data Leak

Big Tech Server's Down

Data Centers' Power Consumption

Prevent from Paying Premium Services

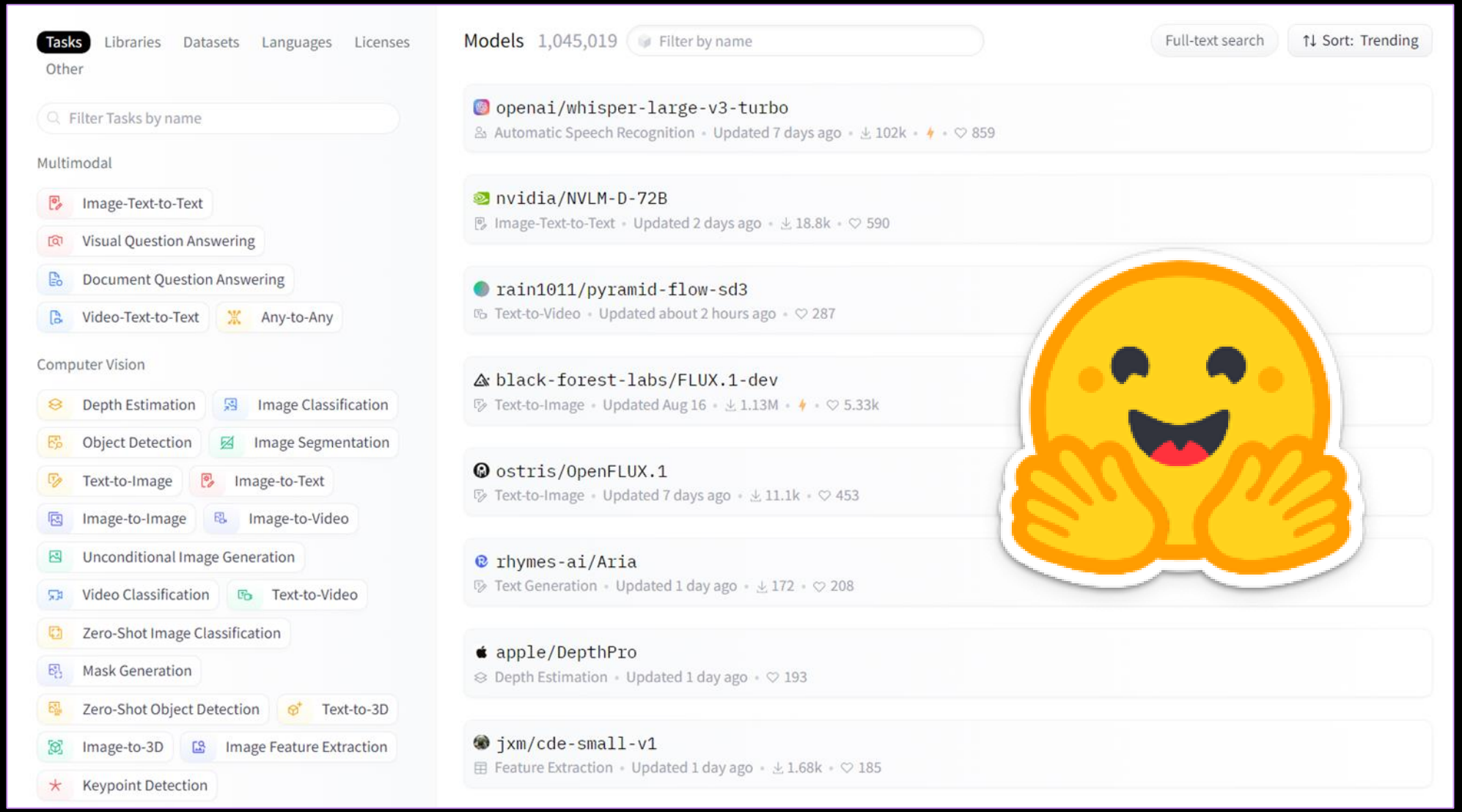
No internet

Try:

- Checking the network cables, modem, and router
- Reconnecting to Wi-Fi

ERR_INTERNET_DISCONNECTED

eduroam



Found out some LLMs are too big to download, need to use API to connect (Require internet)

I have an interesting way to prevent all these problems for using each device offline!

2

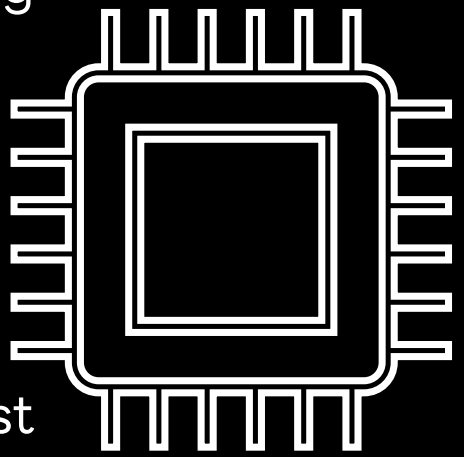
ทำความรู้จักกับ Small Language Models

What is Small Language Model?

Use in offline mode

Local Hosting

Model Drift



Inference Time

Rate Limits

Data Privacy and Protection

Lower Resource Requirements

Ease of Development

Token Usage and Cost

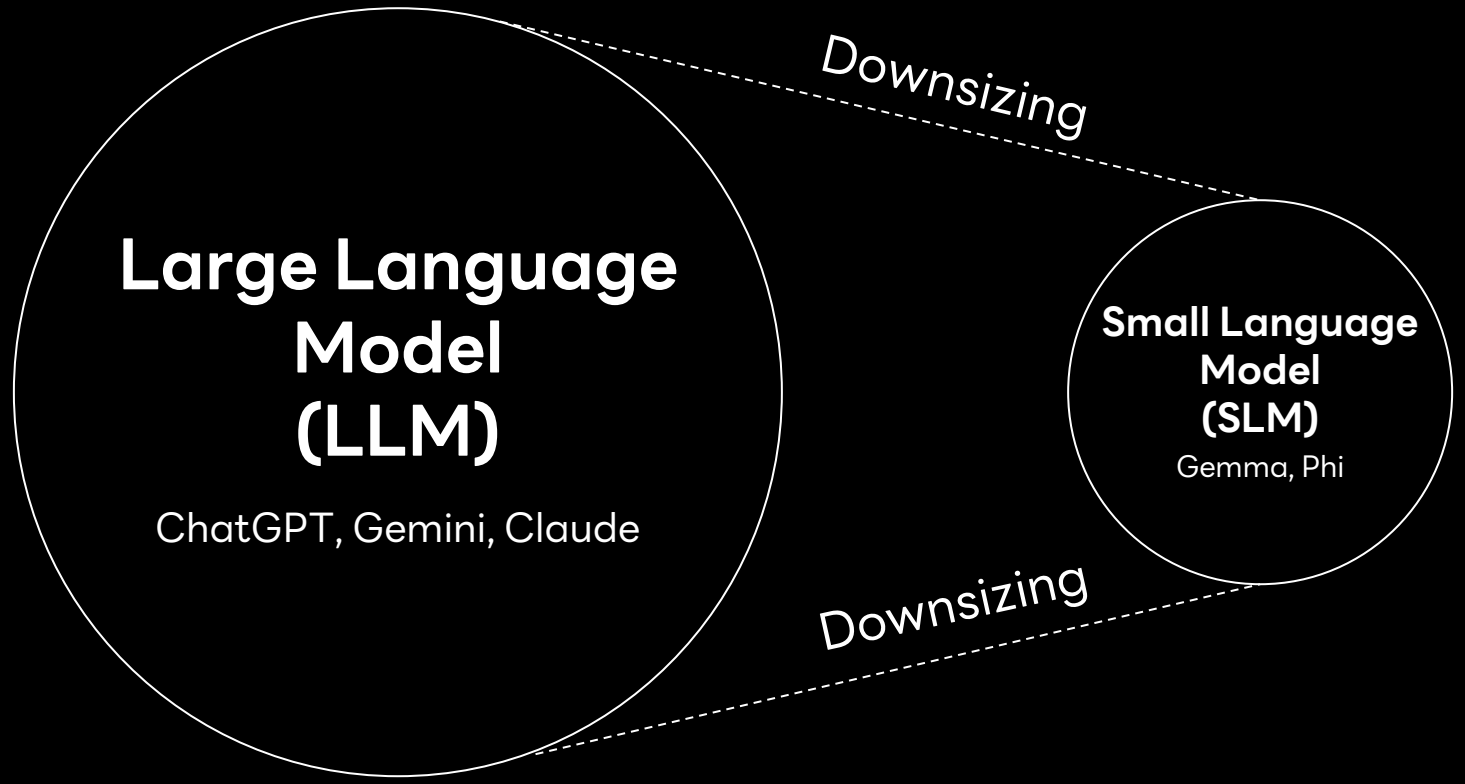
Conversation & Context Management

Natural Language Generation

Knowledge Intensive NPU

And Environmental Preservation 🌿

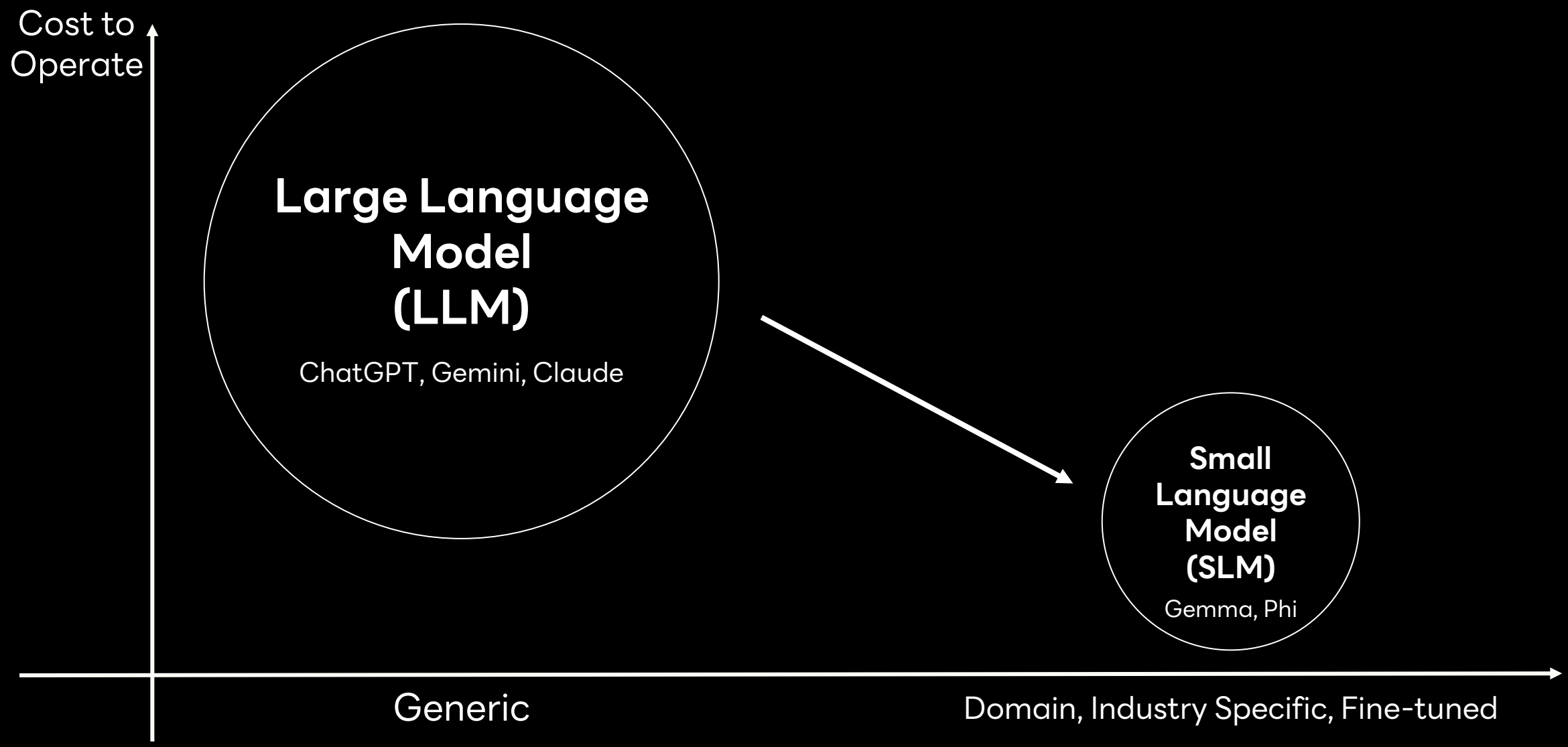
Language Model Comparison



- Ten of billions in parameters
- Requires substantial computational power for training and development
- Higher performance in broader and more complex tasks

- Millions to few billions in parameters **(should be lower than 7B)**
- Capable of being trained with consumer GPUs and lower budgets
- Effective for specific and narrow tasks

Language Model Comparison

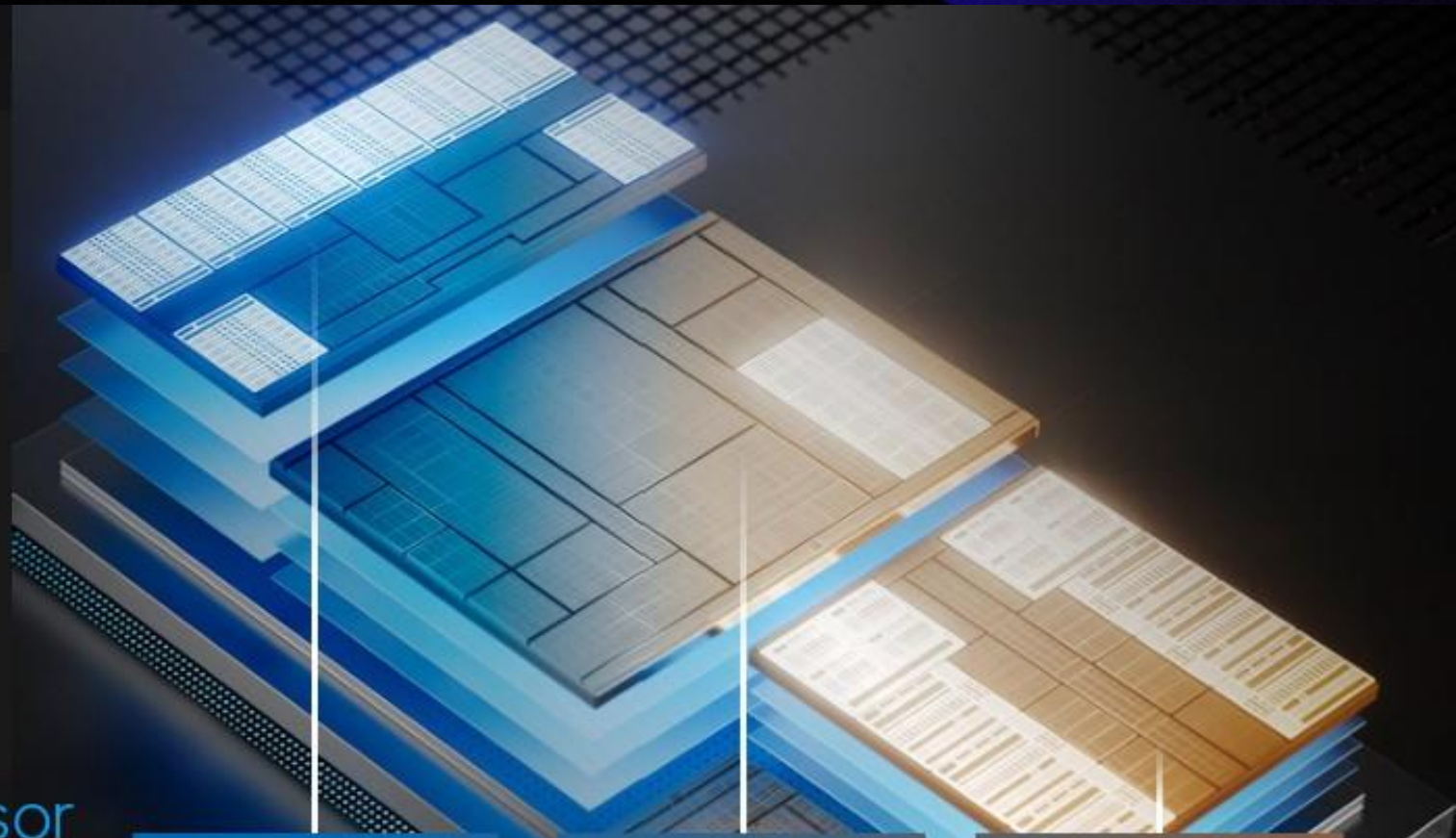


Three AI Engines

with Intel® Core™ Ultra Processor

Heterogenous execution of AI workloads embraces the best practices in AI software design

Deliver up to **34 TeraOPS**[1]



GPU

High Throughput
Ideal for AI-accelerated digital content creation

NPU

Low Power
Ideal for sustained AI workloads and AI offload for battery life

CPU

Fast Response
Ideal for low-latency AI workloads

1. Based on Intel® Core™ Ultra 7 165H processor combined TOPS of CPU, GPU, and NPU engines.

Potential Applications of SLMs in Nowadays

Phone Smart Device

Smart Home Devices

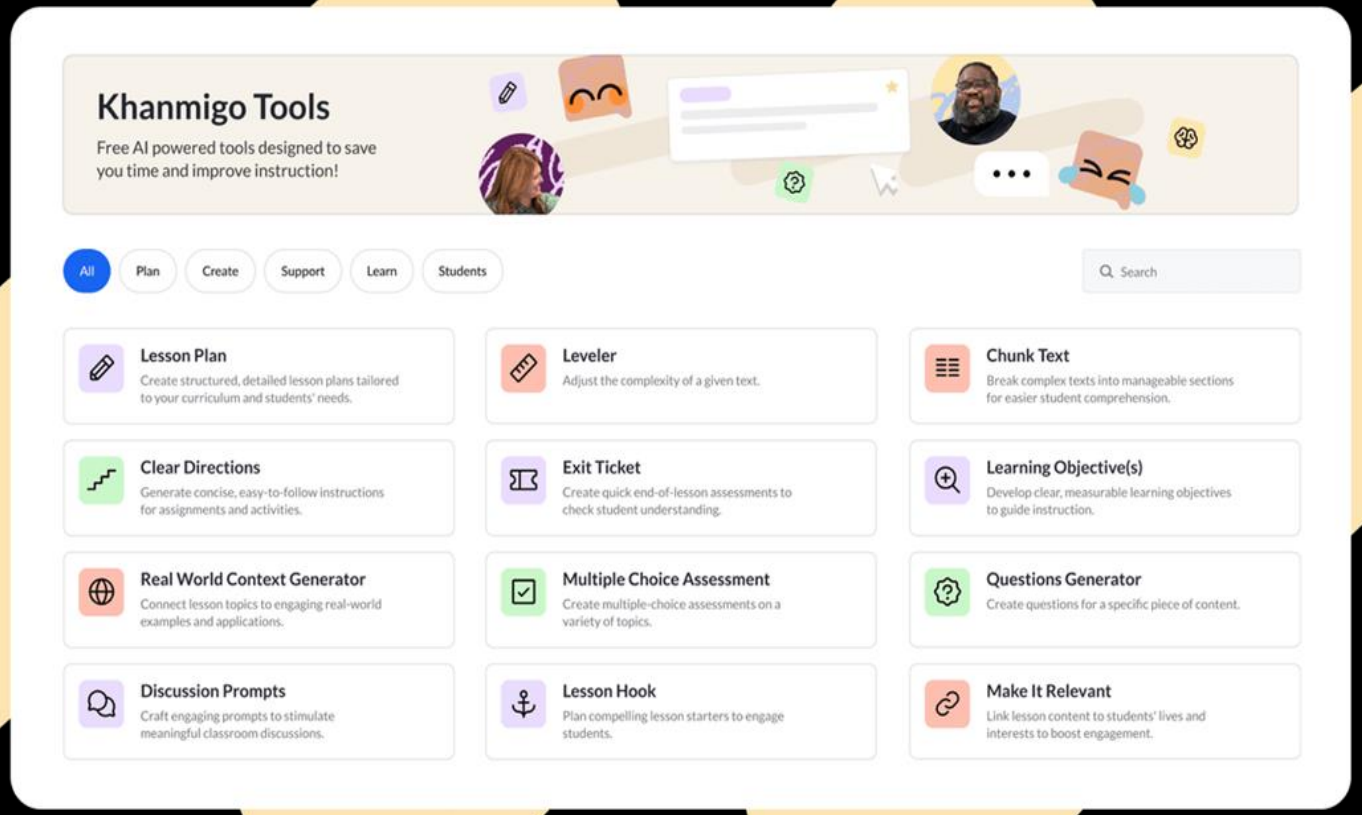
Wearable Technologies

Automative System

Educational Tools

Entertainment Systems

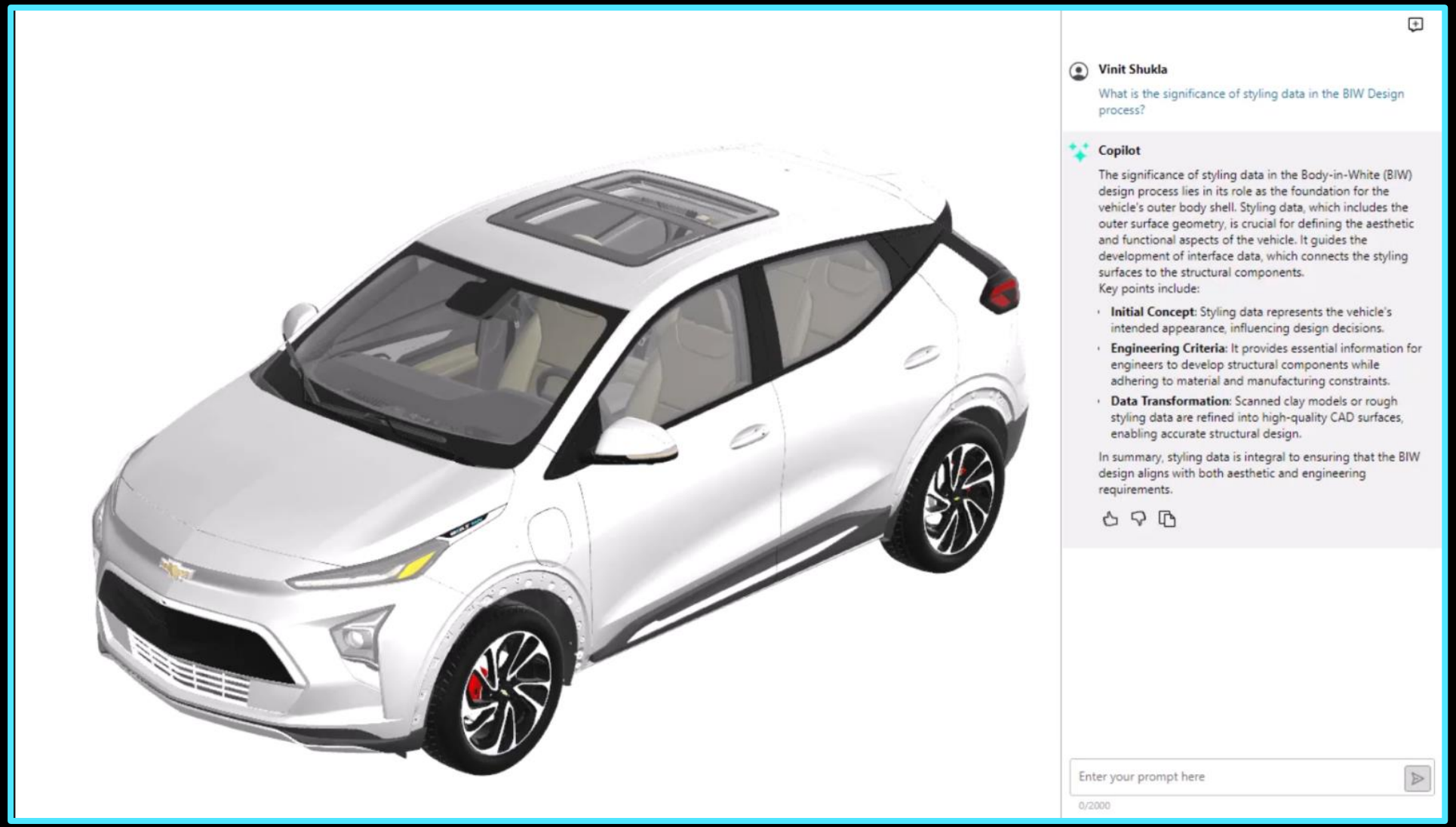
Khanmigo from Khan Academy



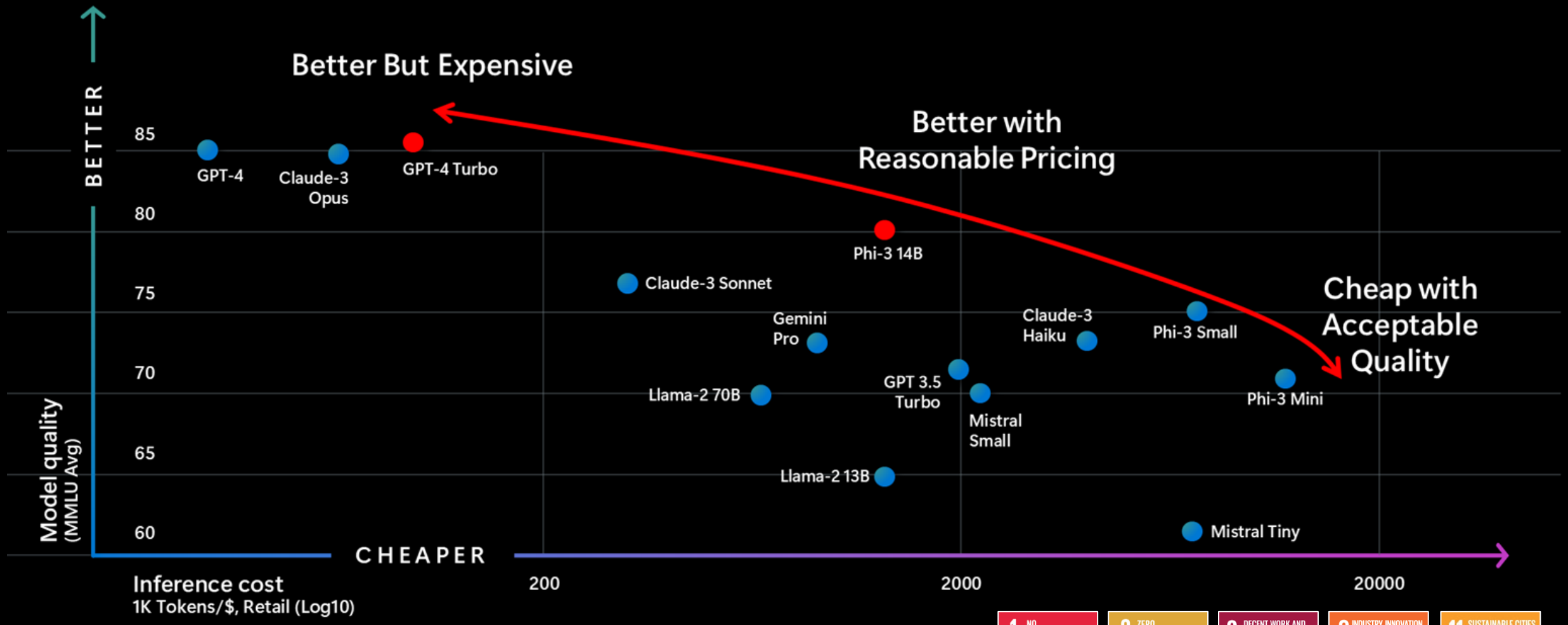
4 QUALITY EDUCATION

[Khanmigo: Free, AI-powered teacher assistant by Khan Academy]

Copilot in NX X, by Siemens



Cost Estimation of Tokenization + Quality

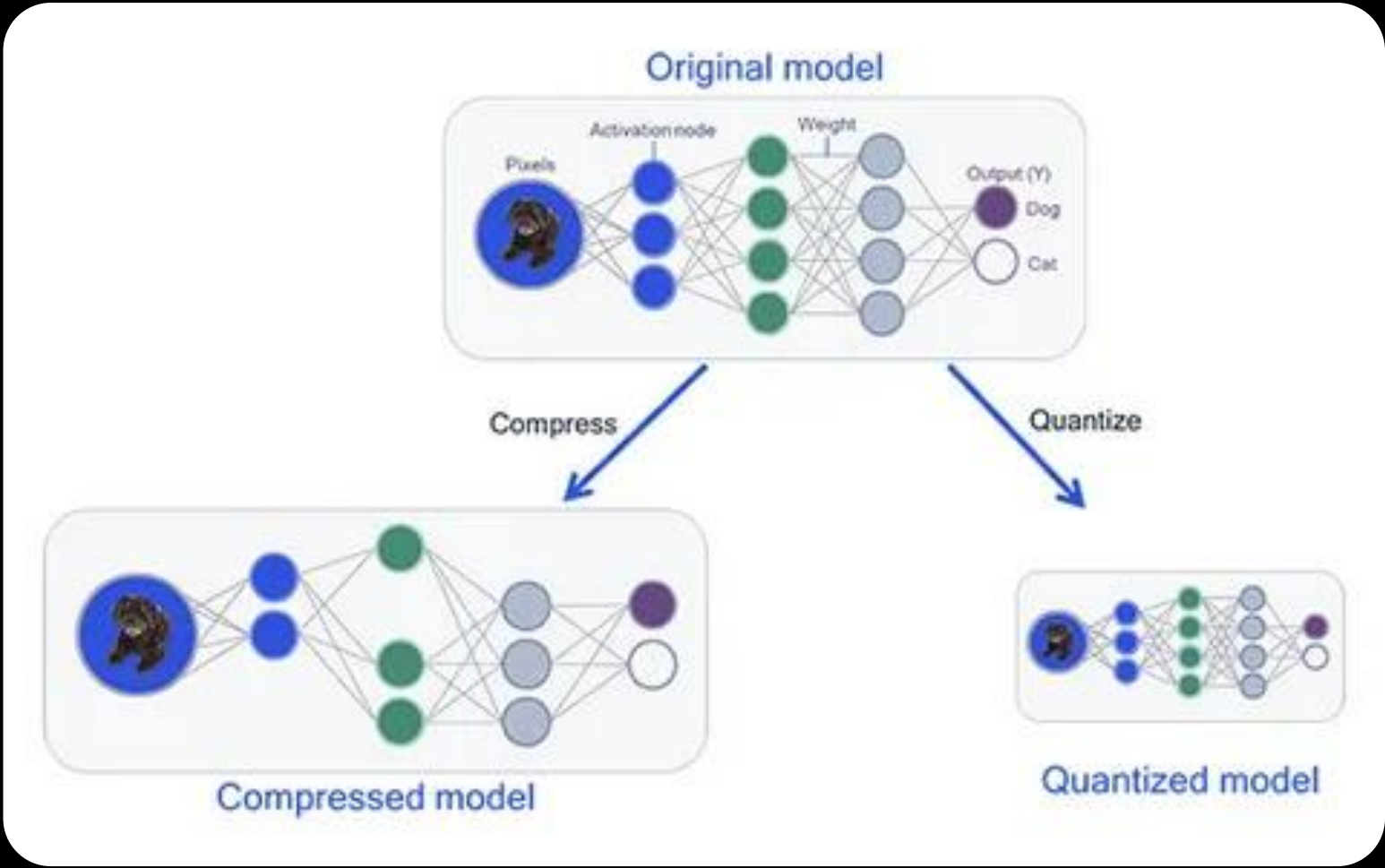


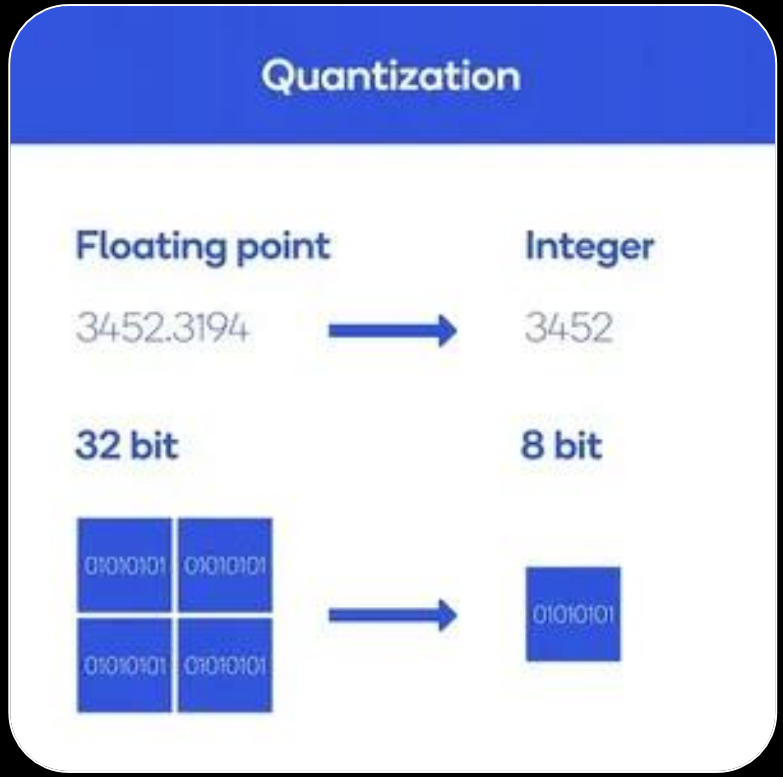
How are you going to use

Small Language Models

on your device at the first step?

SLMs Quantization





GPT-Generated Unified Format (.gguf)

Reduce size and computation cost

Deploy model with limited device resources

Reduce Memory Footprint + Accelerate inference time

May loss some accuracy (but could improve by fine tuning)

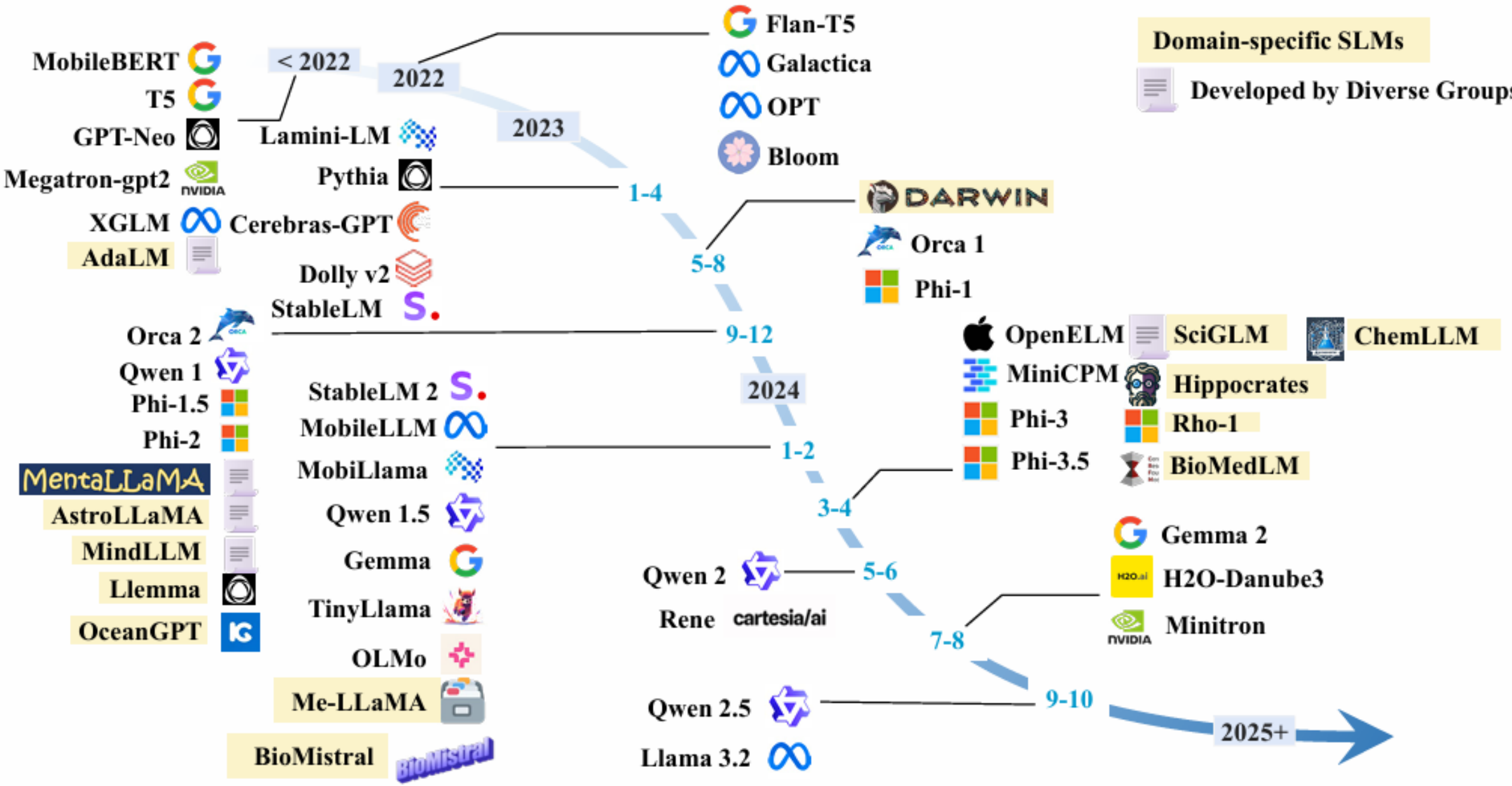
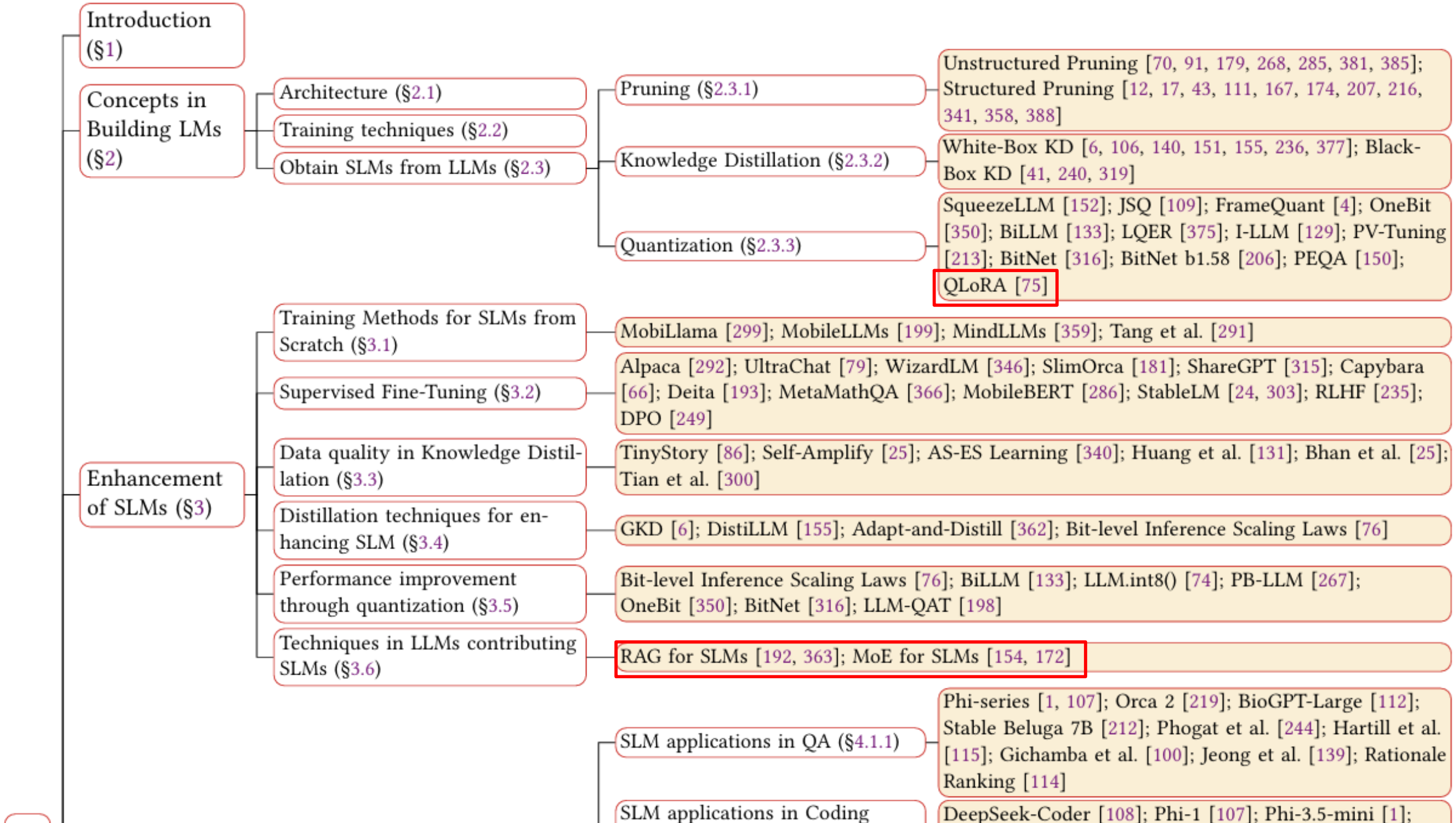
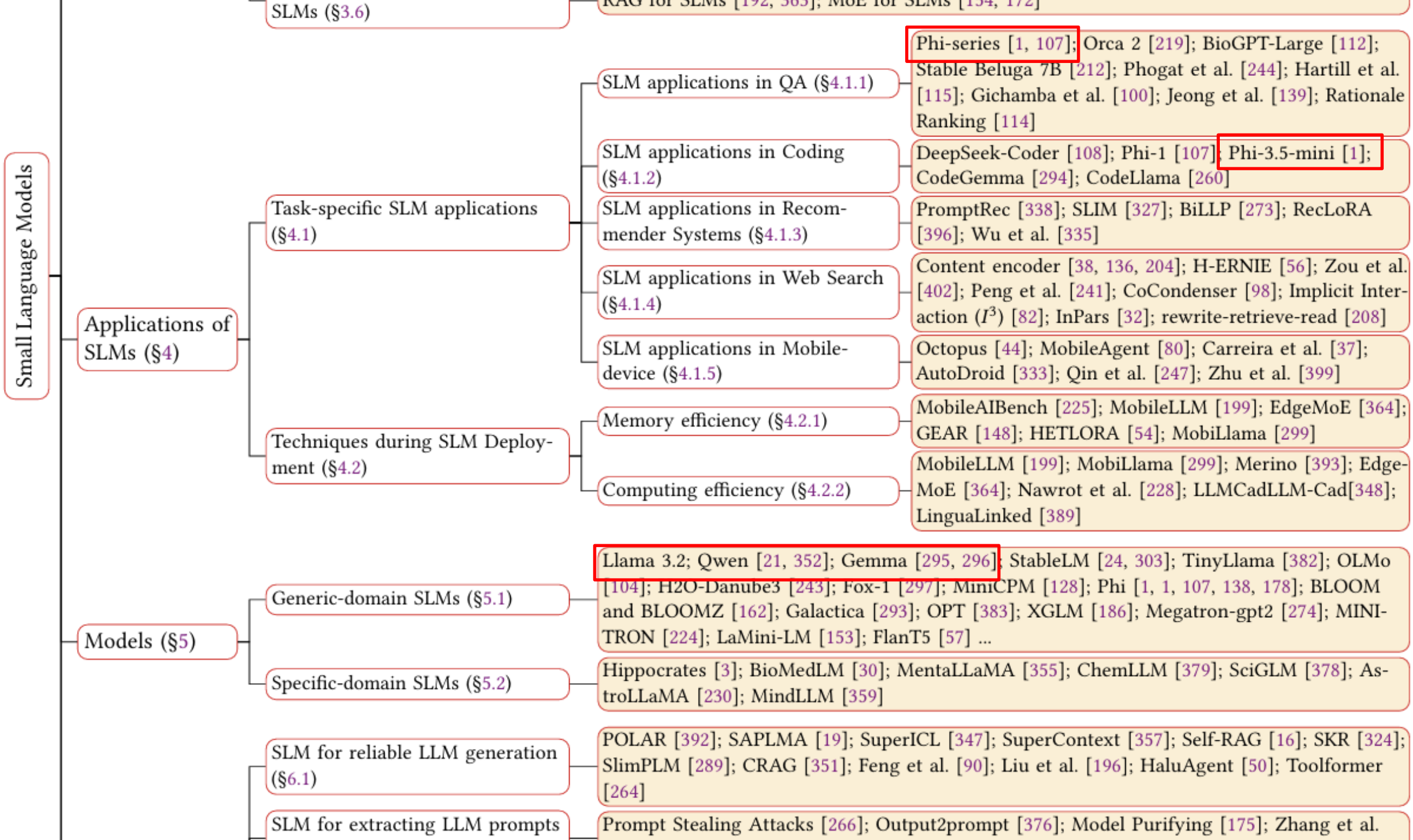
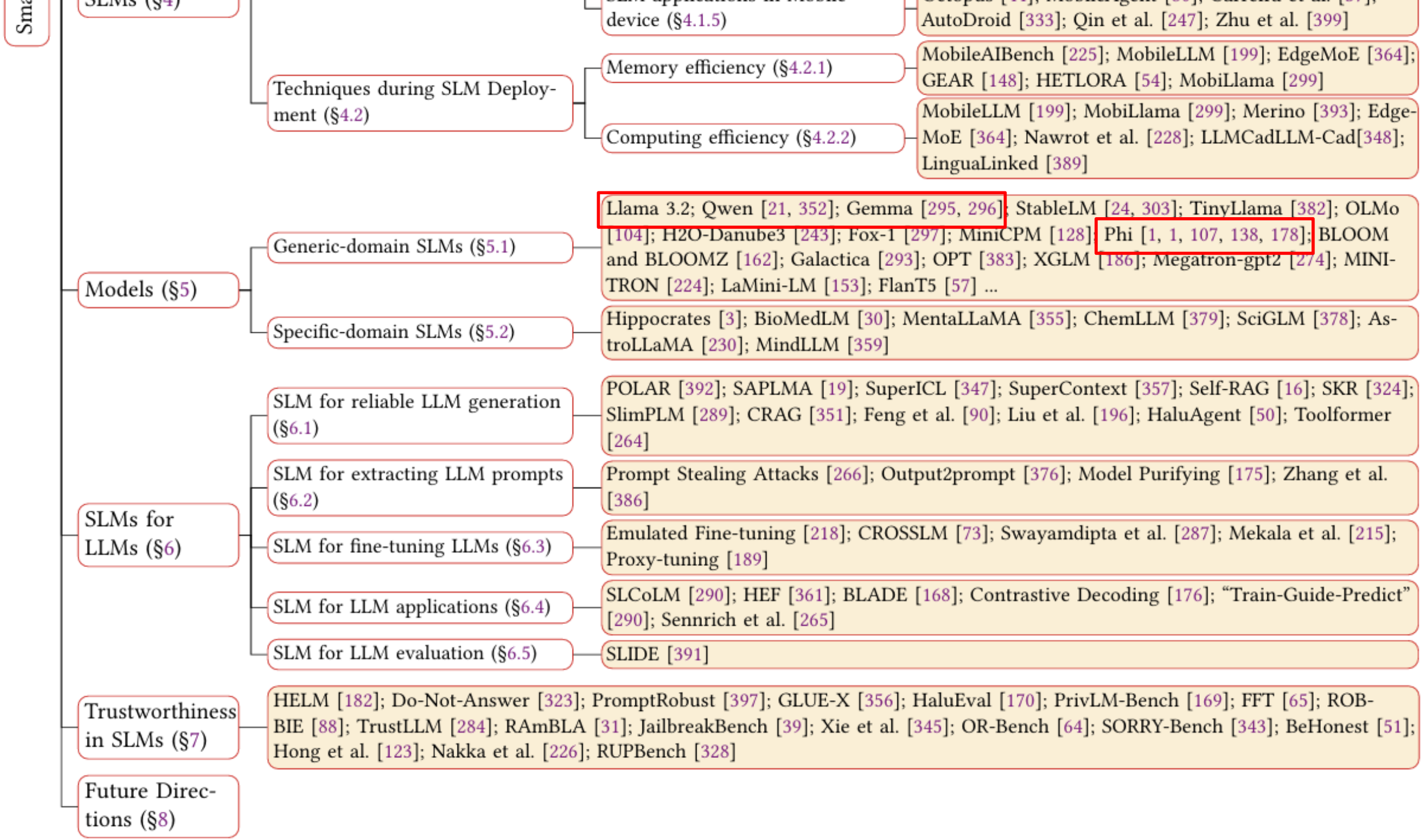


Fig. 3. A timeline of existing small language models.

- Small Language Models
  - Introduction (§1)
  - Concepts in Building LMs (§2)
    - Architecture (§2.1)
    - Training techniques (§2.2)
    - Obtain SLMs from LLMs (§2.3)
      - Pruning (§2.3.1)
        - Unstructured Pruning [70, 91, 179, 268, 285, 381, 385]; Structured Pruning [12, 17, 43, 111, 167, 174, 207, 216, 341, 358, 388]
      - Knowledge Distillation (§2.3.2)
        - White-Box KD [6, 106, 140, 151, 155, 236, 377]; Black-Box KD [41, 240, 319]
      - Quantization (§2.3.3)
        - SqueezeLLM [152]; JSQ [109]; FrameQuant [4]; OneBit [350]; BiLLM [133]; LQER [375]; I-LLM [129]; PV-Tuning [213]; BitNet [316]; BitNet b1.58 [206]; PEQA [150]; QLoRA [75]
  - Enhancement of SLMs (§3)
    - Training Methods for SLMs from Scratch (§3.1)
      - MobiLlama [299]; MobileLLMs [199]; MindLLMs [359]; Tang et al. [291]
    - Supervised Fine-Tuning (§3.2)
      - Alpaca [292]; UltraChat [79]; WizardLM [346]; SlimOrca [181]; ShareGPT [315]; Capybara [66]; Deita [193]; MetaMathQA [366]; MobileBERT [286]; StableLM [24, 303]; RLHF [235]; DPO [249]
    - Data quality in Knowledge Distillation (§3.3)
      - TinyStory [86]; Self-Amplify [25]; AS-ES Learning [340]; Huang et al. [131]; Bhan et al. [25]; Tian et al. [300]
    - Distillation techniques for enhancing SLM (§3.4)
      - GKD [6]; DistiLLM [155]; Adapt-and-Distill [362]; Bit-level Inference Scaling Laws [76]
    - Performance improvement through quantization (§3.5)
      - Bit-level Inference Scaling Laws [76]; BiLLM [133]; LLM.int8() [74]; PB-LLM [267]; OneBit [350]; BitNet [316]; LLM-QAT [198]
    - Techniques in LLMs contributing SLMs (§3.6)
      - RAG for SLMs [192, 363]; MoE for SLMs [154, 172]
  - Applications of SLMs (§4)
    - Task-specific SLM applications (§4.1)
      - SLM applications in QA (§4.1.1)
        - Phi-series [1, 107]; Orca 2 [219]; BioGPT-Large [112]; Stable Beluga 7B [212]; Phogat et al. [244]; Hartill et al. [115]; Gichamba et al. [100]; Jeong et al. [139]; Rationale Ranking [114]
      - SLM applications in Coding (§4.1.2)
        - DeepSeek-Coder [108]; Phi-1 [107]; Phi-3.5-mini [1]; CodeGemma [294]; CodeLlama [260]
      - SLM applications in Recommender Systems (§4.1.3)
        - PromptRec [338]; SLIM [327]; BiLLP [273]; RecLoRA [396]; Wu et al. [335]
      - SLM applications in Web Search (§4.1.4)
        - Content encoder [38, 136, 204]; H-ERNIE [56]; Zou et al. [402]; Peng et al. [241]; CoCondenser [98]; Implicit Interaction (I^3) [82]; InPars [32]; rewrite-retrieve-read [208]
      - SLM applications in Mobile-device (§4.1.5)
        - Octopus [44]; MobileAgent [80]; Carreira et al. [37]; AutoDroid [333]; Qin et al. [247]; Zhu et al. [399]
    - Techniques during SLM Deployment (§4.2)
      - Memory efficiency (§4.2.1)
        - MobileAIBench [225]; MobileLLM [199]; EdgeMoE [364]; GEAR [148]; HETLORA [54]; MobiLlama [299]
      - Computing efficiency (§4.2.2)
        - MobileLLM [199]; MobiLlama [299]; Merino [393]; EdgeMoE [364]; Nawrot et al. [228]; LLMCadLLM-Cad[348]; LinguaLinked [389]
  - Models (§5)
    - Generic-domain SLMs (§5.1)
      - Llama 3.2; Qwen [21, 352]; Gemma [295, 296]; StableLM [24, 303]; TinyLlama [382]; OLMo [104]; H2O-Danube3 [243]; Fox-1 [297]; MiniCPM [128]; Phi [1, 1, 107, 138, 178]; BLOOM and BLOOMZ [162]; Galactica [293]; OPT [383]; XGLM [186]; Megatron-gpt2 [274]; MINITRON [224]; LaMini-LM [153]; FlanT5 [57] ...
    - Specific-domain SLMs (§5.2)
      - Hippocrates [3]; BioMedLM [30]; MentaLLaMA [355]; ChemLLM [379]; SciGLM [378]; AstroLLaMA [230]; MindLLM [359]
  - SLMs for LLMs (§6)
    - SLM for reliable LLM generation (§6.1)
      - POLAR [392]; SAPLMA [19]; SuperICL [347]; SuperContext [357]; Self-RAG [16]; SKR [324]; SlimPLM [289]; CRAG [351]; Feng et al. [90]; Liu et al. [196]; HaluAgent [50]; Toolformer [264]
    - SLM for extracting LLM prompts (§6.2)
      - Prompt Stealing Attacks [266]; Output2prompt [376]; Model Purifying [175]; Zhang et al. [386]
    - SLM for fine-tuning LLMs (§6.3)
      - Emulated Fine-tuning [218]; CROSSLM [73]; Swayamdipta et al. [287]; Mekala et al. [215]; Proxy-tuning [189]
    - SLM for LLM applications (§6.4)
      - SLCoLM [290]; HEF [361]; BLADE [168]; Contrastive Decoding [176]; "Train-Guide-Predict" [290]; Sennrich et al. [265]
    - SLM for LLM evaluation (§6.5)
      - SLIDE [391]
  - Trustworthiness in SLMs (§7)
    - HELM [182]; Do-Not-Answer [323]; PromptRobust [397]; GLUE-X [356]; HaluEval [170]; PrivLM-Bench [169]; FFT [65]; ROBBIE [88]; TrustLLM [284]; RAmBLA [31]; JailbreakBench [39]; Xie et al. [345]; OR-Bench [64]; SORRY-Bench [343]; BeHonest [51]; Hong et al. [123]; Nakka et al. [226]; RUPBench [328]
  - Future Directions (§8)


Introduction (§1)
Concepts in Building LMs (§2)
Architecture (§2.1)
Training techniques (§2.2)
Obtain SLMs from LLMs (§2.3)
Pruning (§2.3.1)
Unstructured Pruning [70, 91, 179, 268, 285, 381, 385]; Structured Pruning [12, 17, 43, 111, 167, 174, 207, 216, 341, 358, 388]
Knowledge Distillation (§2.3.2)
White-Box KD [6, 106, 140, 151, 155, 236, 377]; Black-Box KD [41, 240, 319]
Quantization (§2.3.3)
SqueezeLLM [152]; JSQ [109]; FrameQuant [4]; OneBit [350]; BiLLM [133]; LQER [375]; I-LLM [129]; PV-Tuning [213]; BitNet [316]; BitNet b1.58 [206]; PEQA [150]; QLoRA [75]
Enhancement of SLMs (§3)
Training Methods for SLMs from Scratch (§3.1)
MobiLlama [299]; MobileLLMs [199]; MindLLMs [359]; Tang et al. [291]
Supervised Fine-Tuning (§3.2)
Alpaca [292]; UltraChat [79]; WizardLM [346]; SlimOrca [181]; ShareGPT [315]; Capybara [66]; Deita [193]; MetaMathQA [366]; MobileBERT [286]; StableLM [24, 303]; RLHF [235]; DPO [249]
Data quality in Knowledge Distillation (§3.3)
TinyStory [86]; Self-Amplify [25]; AS-ES Learning [340]; Huang et al. [131]; Bhan et al. [25]; Tian et al. [300]
Distillation techniques for enhancing SLM (§3.4)
GKD [6]; DistiLLM [155]; Adapt-and-Distill [362]; Bit-level Inference Scaling Laws [76]
Performance improvement through quantization (§3.5)
Bit-level Inference Scaling Laws [76]; BiLLM [133]; LLM.int8() [74]; PB-LLM [267]; OneBit [350]; BitNet [316]; LLM-QAT [198]
Techniques in LLMs contributing SLMs (§3.6)
RAG for SLMs [192, 363]; MoE for SLMs [154, 172]
SLM applications in QA (§4.1.1)
Phi-series [1, 107]; Orca 2 [219]; BioGPT-Large [112]; Stable Beluga 7B [212]; Phogat et al. [244]; Hartill et al. [115]; Gichamba et al. [100]; Jeong et al. [139]; Rationale Ranking [114]
SLM applications in Coding
DeepSeek-Coder [108]; Phi-1 [107]; Phi-3.5-mini [1];


Small Language Models
Applications of SLMs (§4)
Task-specific SLM applications (§4.1)
SLM applications in QA (§4.1.1)
Phi-series [1, 107]; Orca 2 [219]; BioGPT-Large [112]; Stable Beluga 7B [212]; Phogat et al. [244]; Hartill et al. [115]; Gichamba et al. [100]; Jeong et al. [139]; Rationale Ranking [114]
SLM applications in Coding (§4.1.2)
DeepSeek-Coder [108]; Phi-1 [107]; Phi-3.5-mini [1]; CodeGemma [294]; CodeLlama [260]
SLM applications in Recommender Systems (§4.1.3)
PromptRec [338]; SLIM [327]; BiLLP [273]; RecLoRA [396]; Wu et al. [335]
SLM applications in Web Search (§4.1.4)
Content encoder [38, 136, 204]; H-ERNIE [56]; Zou et al. [402]; Peng et al. [241]; CoCondenser [98]; Implicit Interaction (I^3) [82]; InPars [32]; rewrite-retrieve-read [208]
SLM applications in Mobile-device (§4.1.5)
Octopus [44]; MobileAgent [80]; Carreira et al. [37]; AutoDroid [333]; Qin et al. [247]; Zhu et al. [399]
Techniques during SLM Deployment (§4.2)
Memory efficiency (§4.2.1)
MobileAIBench [225]; MobileLLM [199]; EdgeMoE [364]; GEAR [148]; HETLORA [54]; MobiLlama [299]
Computing efficiency (§4.2.2)
MobileLLM [199]; MobiLlama [299]; Merino [393]; EdgeMoE [364]; Nawrot et al. [228]; LLMCadLLM-Cad[348]; LinguaLinked [389]
Models (§5)
Generic-domain SLMs (§5.1)
Llama 3.2; Qwen [21, 352]; Gemma [295, 296]; StableLM [24, 303]; TinyLlama [382]; OLMo [104]; H2O-Danube3 [243]; Fox-1 [297]; MiniCPM [128]; Phi [1, 1, 107, 138, 178]; BLOOM and BLOOMZ [162]; Galactica [293]; OPT [383]; XGLM [186]; Megatron-gpt2 [274]; MINITRON [224]; LaMini-LM [153]; FlanT5 [57] ...
Specific-domain SLMs (§5.2)
Hippocrates [3]; BioMedLM [30]; MentaLLaMA [355]; ChemLLM [379]; SciGLM [378]; AstroLLaMA [230]; MindLLM [359]
SLM for reliable LLM generation (§6.1)
POLAR [392]; SAPLMA [19]; SuperICL [347]; SuperContext [357]; Self-RAG [16]; SKR [324]; SlimPLM [289]; CRAG [351]; Feng et al. [90]; Liu et al. [196]; HaluAgent [50]; Toolformer [264]
SLM for extracting LLM prompts
Prompt Stealing Attacks [266]; Output2prompt [376]; Model Purifying [175]; Zhang et al.


Sma
SLMs (§4)
device (§4.1.5)
AutoDroid [333]; Qin et al. [247]; Zhu et al. [399]
Techniques during SLM Deployment (§4.2)
Memory efficiency (§4.2.1)
MobileAIBench [225]; MobileLLM [199]; EdgeMoE [364]; GEAR [148]; HETLORA [54]; MobiLlama [299]
Computing efficiency (§4.2.2)
MobileLLM [199]; MobiLlama [299]; Merino [393]; EdgeMoE [364]; Nawrot et al. [228]; LLMCadLLM-Cad[348]; LinguaLinked [389]
Models (§5)
Generic-domain SLMs (§5.1)
Llama 3.2; Qwen [21, 352]; Gemma [295, 296]; StableLM [24, 303]; TinyLlama [382]; OLMo [104]; H2O-Danube3 [243]; Fox-1 [297]; MiniCPM [128]; Phi [1, 1, 107, 138, 178]; BLOOM and BLOOMZ [162]; Galactica [293]; OPT [383]; XGLM [186]; Megatron-gpt2 [274]; MINITRON [224]; LaMini-LM [153]; FlanT5 [57] ...
Specific-domain SLMs (§5.2)
Hippocrates [3]; BioMedLM [30]; MentaLLaMA [355]; ChemLLM [379]; SciGLM [378]; AstroLLaMA [230]; MindLLM [359]
SLMs for LLMs (§6)
SLM for reliable LLM generation (§6.1)
POLAR [392]; SAPLMA [19]; SuperICL [347]; SuperContext [357]; Self-RAG [16]; SKR [324]; SlimPLM [289]; CRAG [351]; Feng et al. [90]; Liu et al. [196]; HaluAgent [50]; Toolformer [264]
SLM for extracting LLM prompts (§6.2)
Prompt Stealing Attacks [266]; Output2prompt [376]; Model Purifying [175]; Zhang et al. [386]
SLM for fine-tuning LLMs (§6.3)
Emulated Fine-tuning [218]; CROSSLM [73]; Swayamdipta et al. [287]; Mekala et al. [215]; Proxy-tuning [189]
SLM for LLM applications (§6.4)
SLCoLM [290]; HEF [361]; BLADE [168]; Contrastive Decoding [176]; “Train-Guide-Predict” [290]; Sennrich et al. [265]
SLM for LLM evaluation (§6.5)
SLIDE [391]
Trustworthiness in SLMs (§7)
HELM [182]; Do-Not-Answer [323]; PromptRobust [397]; GLUE-X [356]; HaluEval [170]; PrivLM-Bench [169]; FFT [65]; ROBBIE [88]; TrustLLM [284]; RAmBLA [31]; JailbreakBench [39]; Xie et al. [345]; OR-Bench [64]; SORRY-Bench [343]; BeHonest [51]; Hong et al. [123]; Nakka et al. [226]; RUPBench [328]
Future Directions (§8)

Less-Parameters Language Models Arena

Microsoft

Phi-1 : 1.3B
Phi-2 : 2.7B
Phi-3.5: 3.8B (≈2.2 GB)

Meta

LLaMA 3.2 : 3B (≈2.02 GB)
Thai-Lang Supported

Google DeepMind

Deepmind's Gemma 2
2B: ≈1.6 GB
9B: ≈5.4 GB (LLM)

Alibaba Cloud

Qwen2.5 3B (≈1.93 GB)
Thai-Lang Supported

SCB X

Typhoon 8B (≈4.7-4.9 GB)
(Consider as LLM)
Thai-Lang Supported

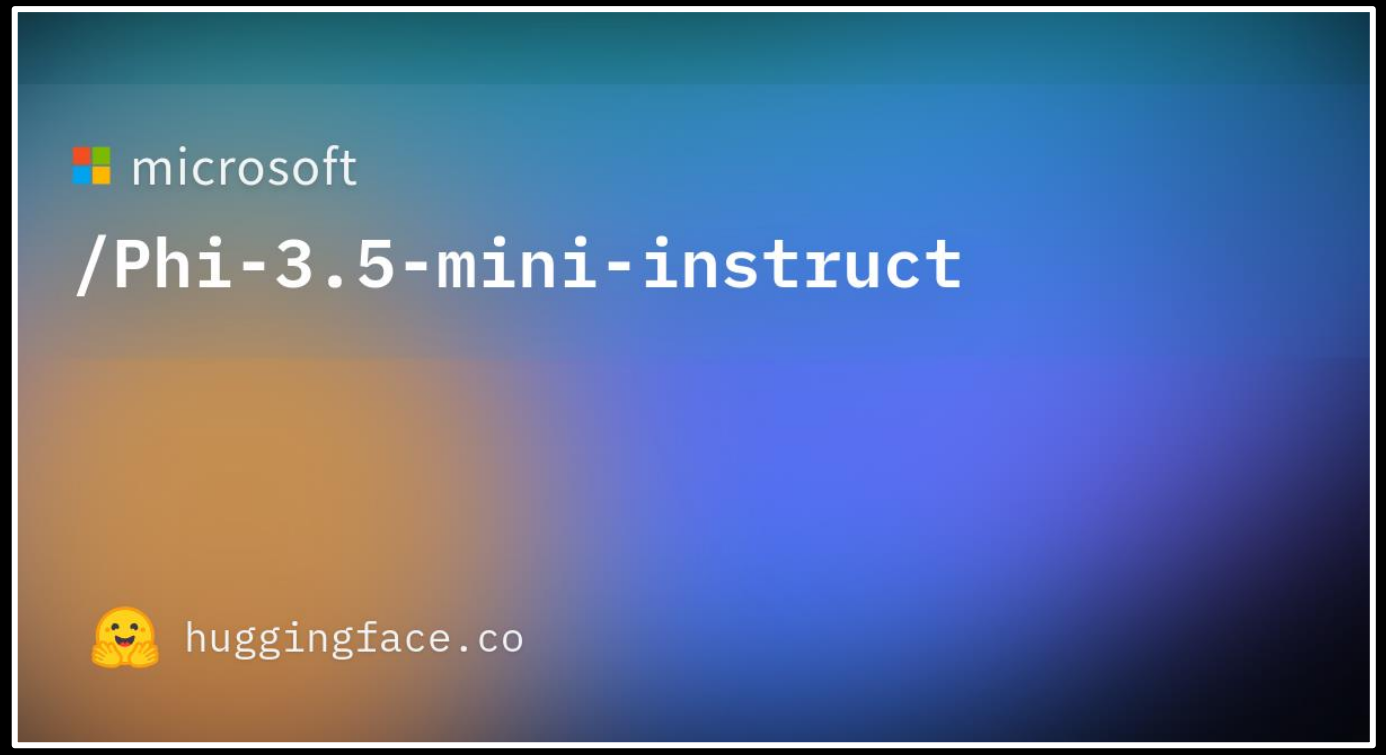
KBTG

THaLLE 7B (≈4.4 GB-5.4 GB)
(Consider as LLM)
Thai-Lang Supported

Size Converted with llama.cpp/GGUF Size

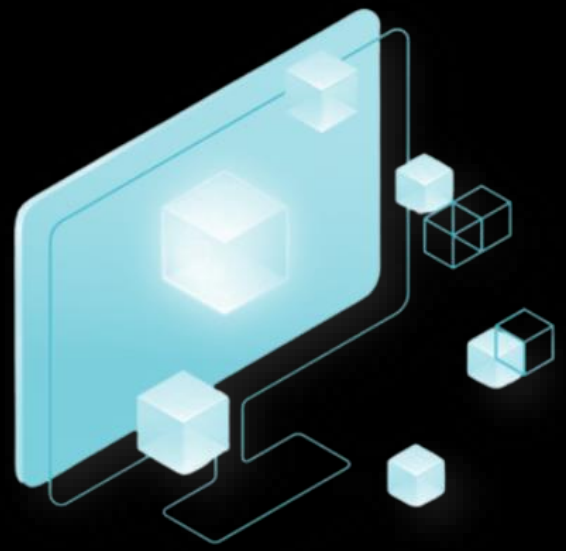
Generally Available

Microsoft's Phi-3.5



microsoft/Phi-3.5-mini-instruct · Hugging Face

Microsoft's Phi-3.5 Family

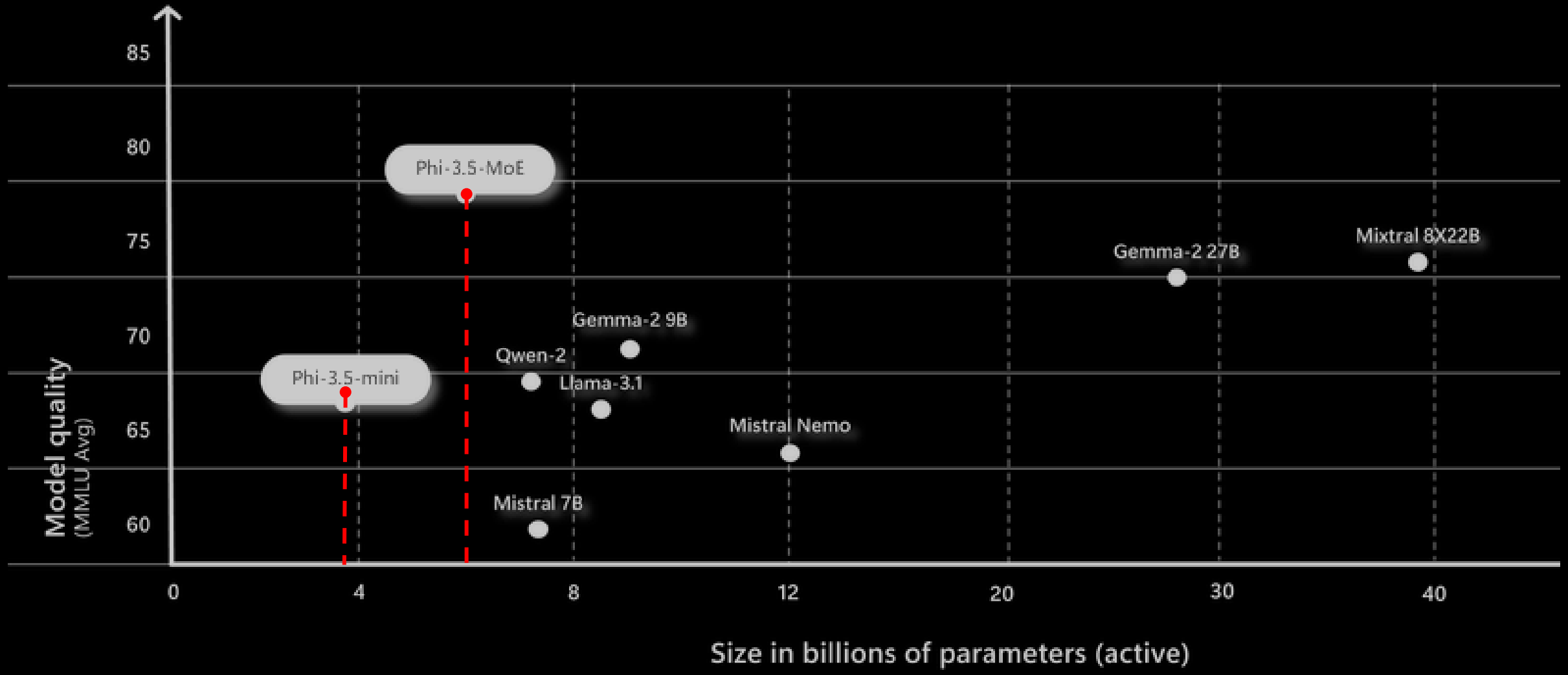


Model For Today's Experiment

Phi-3.5 mini-4k instruct (SLM)

- **Architecture**: 3.8B parameters and is a dense decoder-only Transformer model. The model is fine-tuned with Supervised fine-tuning (SFT) and Direct Preference Optimization (DPO) to ensure alignment with human preferences and safety guidelines.
- **Inputs**: Text. It is best suited for prompts using chat format.
- **Context Window Length**: 4K tokens
- **GPUs**: 512 H100-80G
- **Training time**: 7 days
- **Training data**: 3.3T tokens
- **Outputs**: Generated text in response to the input
- **Dates**: Our models were trained between February and April 2024
- **Status**: This is a static model trained on an offline dataset with cutoff date October 2023. Future versions of the tuned models may be released as we improve models.

Phi-3.5 Performance

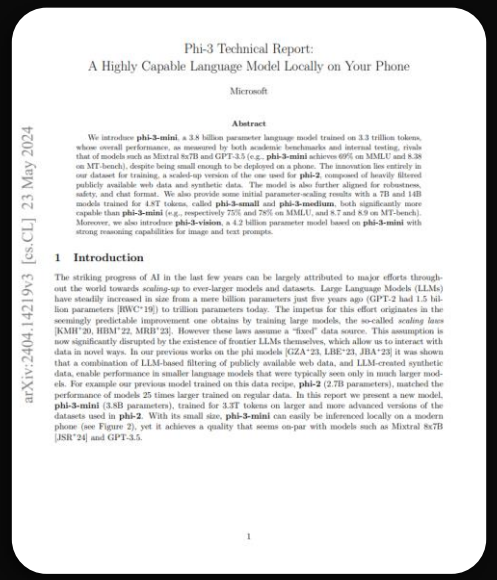


Textbooks Are All You Need

Evolution of Phi family

Phi-1 and Phi-1.5 (First Development)

Phi-2 (First Public Released) – Ignite 2023



Phi-3 (Latest & Stable) – Before #MSBuild 2024

Source: Microsoft Research (ArXiv)

Phi-1 and Phi-1.5 (First Development)

- 1.3 Billion Parameters
- Focusing on Python Coding and Some Commonsense reasoning
- First state-of-the-art performance on Python coding tasks (test with HumanEval and MBPP)
- Comparable performance to model 5x larger in language understanding

Phi-2 (First Public Released) – Ignite 2023

- 2.7 Billion Parameters
- More efficient on reasoning and language
- Matches or outperforms models up to 25x larger on complex benchmarks. Ideal for research and exploration

Phi-3 (Stable) – Before #MSBuild 2024

- At least 3.8B parameters
- More efficient on scientific reasoning and outperform on mathematical problems
- Providing multimodal features on Phi-3-vision-128k-instruct

Phi-3.5 - Fine Tuning from Phi-3 (Latest - Released on August 2024)

- Supporting Multilingual Context (including Thai Language) - MoE model
- More Logical Thinking for specific tasks

Source: Microsoft Research (ArXiv)

Microsoft's Responsible AI Principles

- Fairness
- Reliability & Safety
- Privacy & Security
- Inclusiveness
- Transparency
- Accountability

Run & Develop

LM Studio
For Easy Dev with UI

Ollama
Running in Linux Terminal

Developers
Customize models, build agentic apps, etc.

Main Stack
API and CLI

Distribution
On-premise, Hosted, On-Device, Single-Node

Partners API Providers
Inference, Evals, Fine-tuning, etc.

Models
Core, Safety, Customized, etc.

What is Ollama?

- Ollama is a platform that makes local development with open-source language models a breeze.
- With Ollama, everything you need to run an LLM—model weights and all of the config—is packaged into a single Modelfile. Think it is Docker for LLMs. Act like GitHub at some points
- Run Llama 3.1, Phi 3.5, Mistral, Gemma 2, and other models from Hugging Face. Customize and create your own.

Microsoft

Meta

Introducing

OpenAI API Capability

Ollama now has built-in compatibility with the OpenAI Chat Completions API, making it possible to use more tooling and applications with Ollama locally.

```python
from openai import OpenAI

client = OpenAI(
    base_url = 'http://localhost:11434/v1',
    api_key='ollama', # required, but unused
)

response = client.chat.completions.create(
  model="llama2",
  messages=[
    {"role": "system", "content": "You are a helpful assistant."}
  ]
)
print(response.choices[0].message.content)
```

```javascript
import OpenAI from 'openai'

const openai = new OpenAI({
  baseURL: 'http://localhost:11434/v1',
  apiKey: 'ollama', // required but unused
})

const completion = await openai.chat.completions.create({
  model: 'llama2',
  messages: [{ role: 'user', content: 'Why is the sky blue?' }],
})

console.log(completion.choices[0].message.content)
```

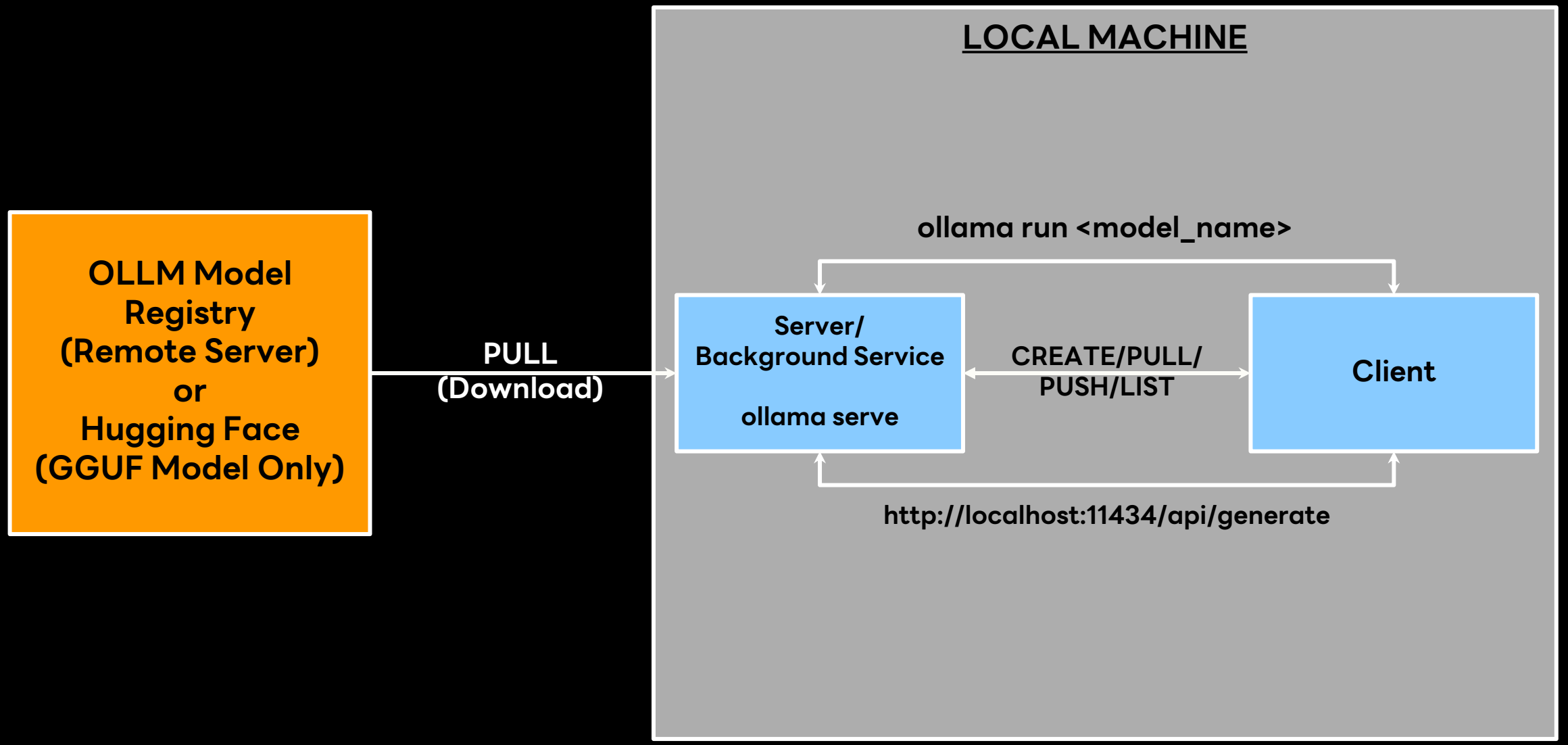
Introducing

Download from Hugging Face

You can use **any GGUF quants** created by the community on Hugging Face directly with Ollama, without creating a new Modelfile. You can run any of them with a single **ollama run** command. We also provide customizations like choosing quantization type to improve your overall experience.

Hugging Face

```
ollama run hf.co/username/repository
```


LOCAL MACHINE
OLLM Model Registry (Remote Server) or Hugging Face (GGUF Model Only)
PULL (Download)
ollama run <model_name>
Server/ Background Service
ollama serve
CREATE/PULL/ PUSH/LIST
Client
http://localhost:11434/api/generate

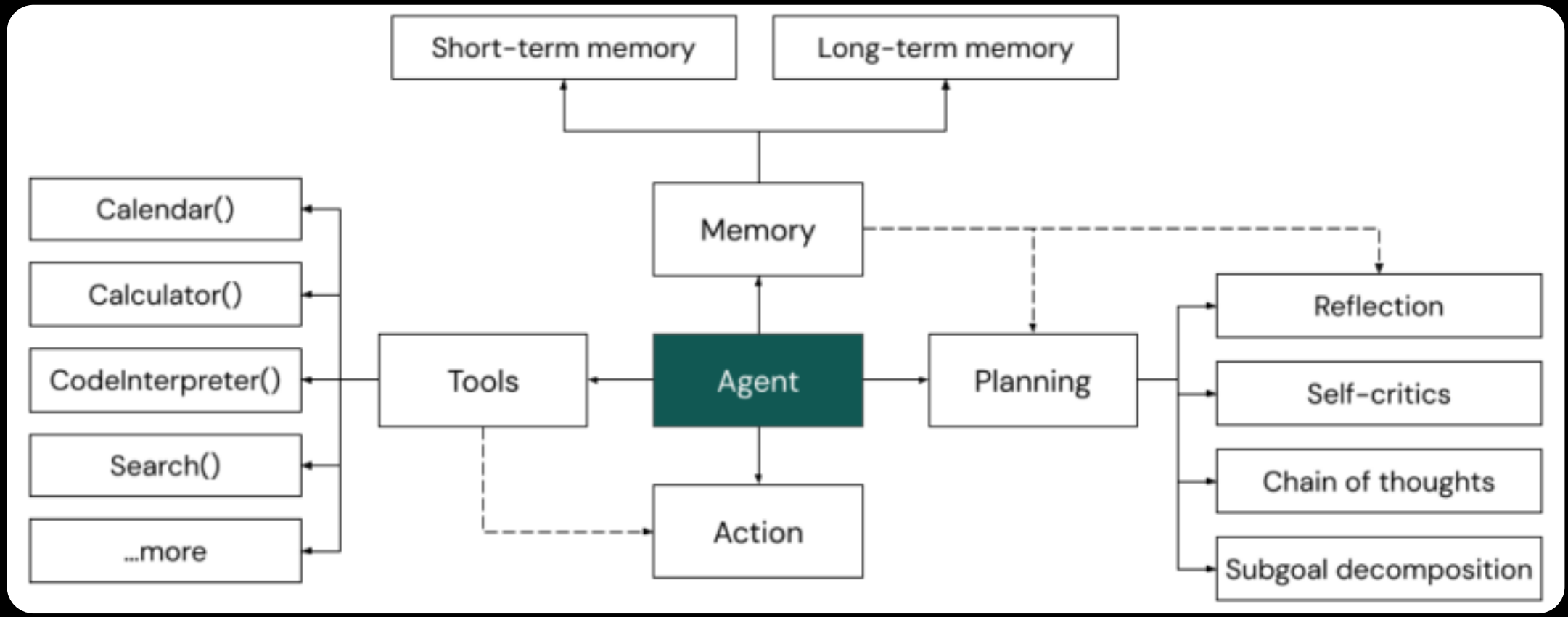
First Demo

Ollama Implementation on Python with **Streamlit**

Ollama Implementation with AutoGen for Python for making the AI agents

What is AI Agents?

- An AI agent is a software entity capable of performing tasks autonomously or semi-autonomously by perceiving its environment, processing information, and taking actions to achieve specific goals.



2025 Top 10 Strategic Technology Trends

AI imperatives and risks

- Agentic AI
- AI Governance Platforms
- Disinformation Security

New frontiers of computing

- Post-Quantum Cryptography
- Ambient Invisible Intelligence
- Energy-Efficient Computing
- Hybrid Computing

Human-machine synergy

- Spatial Computing
- Polyfunctional Robots
- Neurological Enhancement

Source: Gartner

Gartner

What is AutoGen?

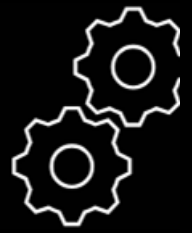
AutoGen An Open-Source Programming Framework for Agentic AI

Multi-Agent Conversation Framework

AutoGen provides multi-agent conversation framework as a high-level abstraction. With this framework, one can conveniently build LLM workflows.

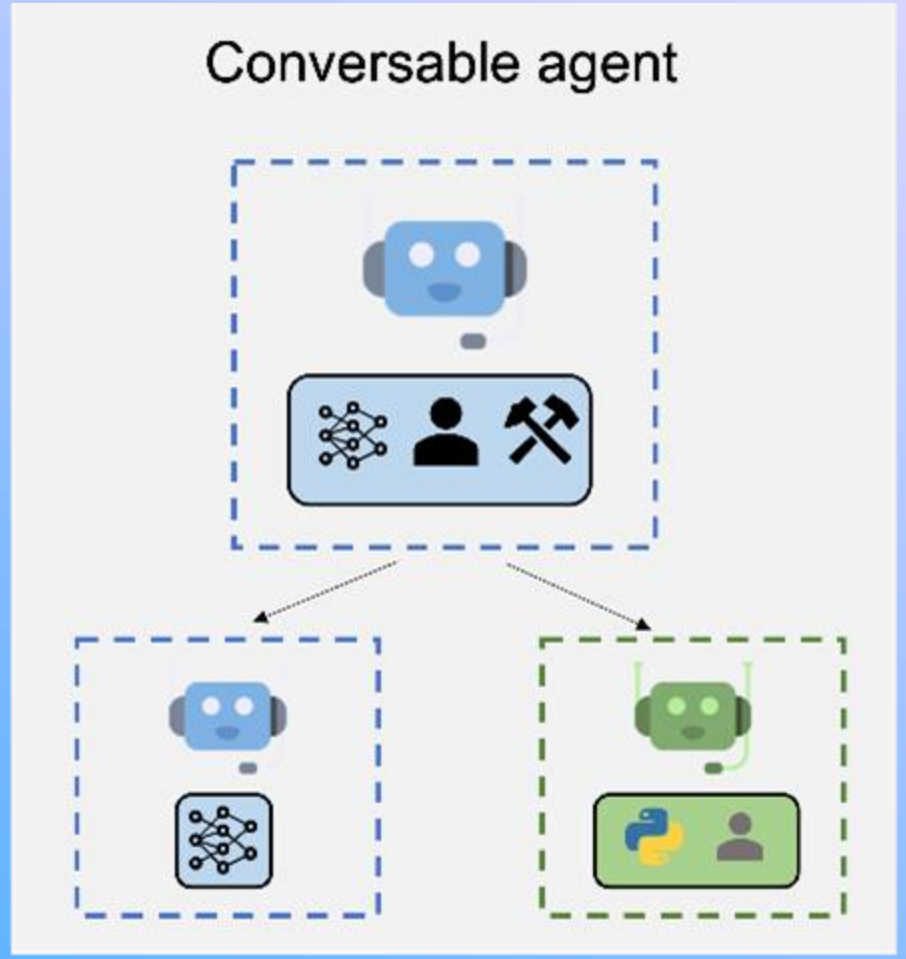
Easily Build Diverse Applications

AutoGen offers a collection of working systems spanning a wide range of applications from various domains and complexities.

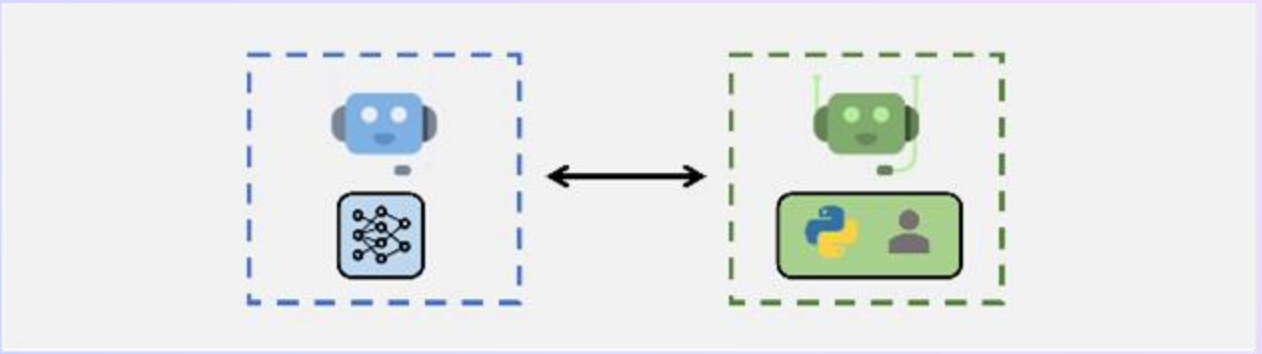


Enhanced LLM Inference & Optimization

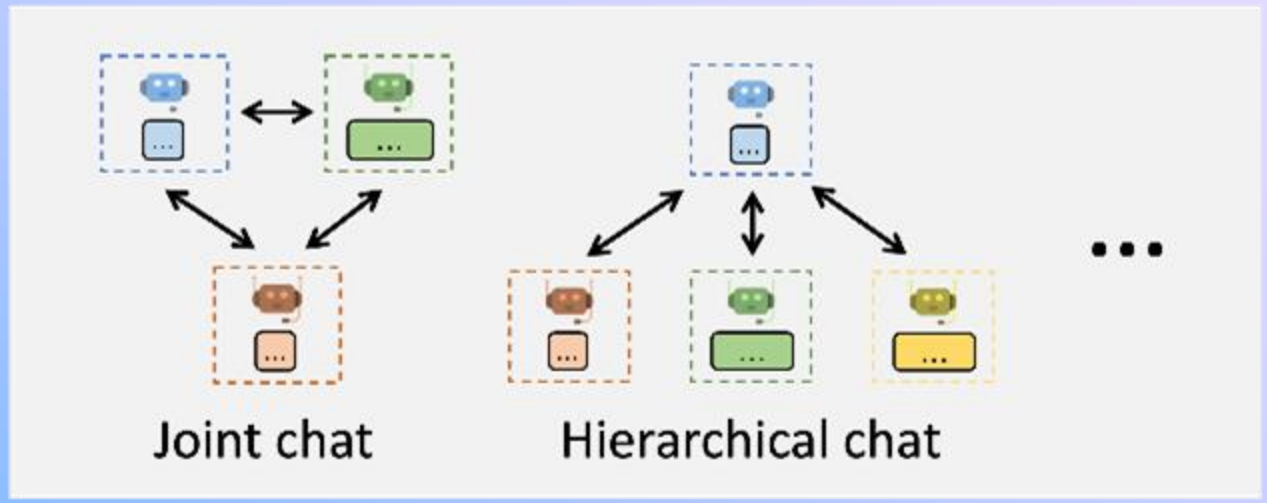
AutoGen supports enhanced LLM inference APIs, which can be used to improve inference performance and reduce cost.



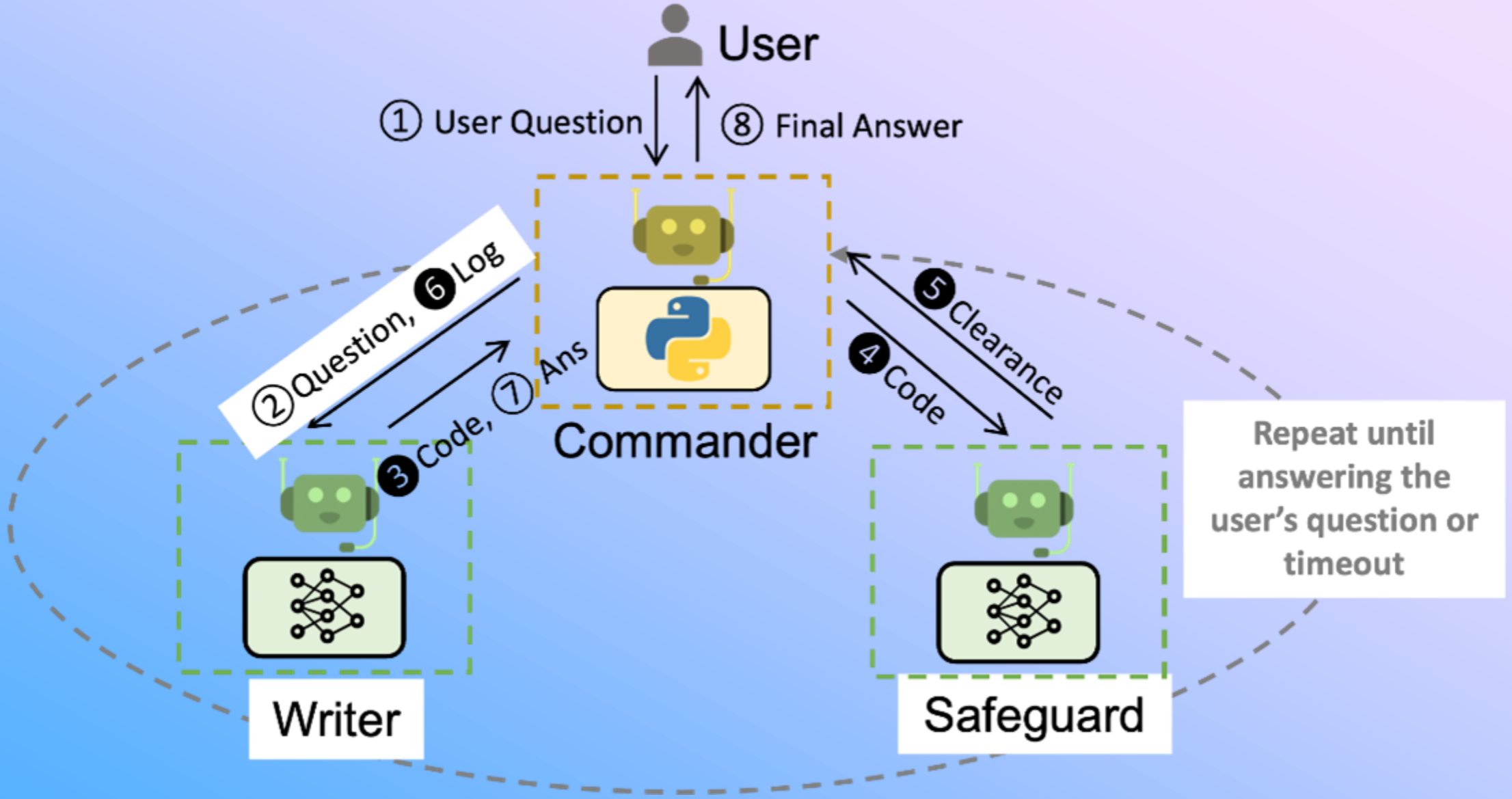
Agent Customization



Multi-Agent Conversations



Flexible Conversation Patterns

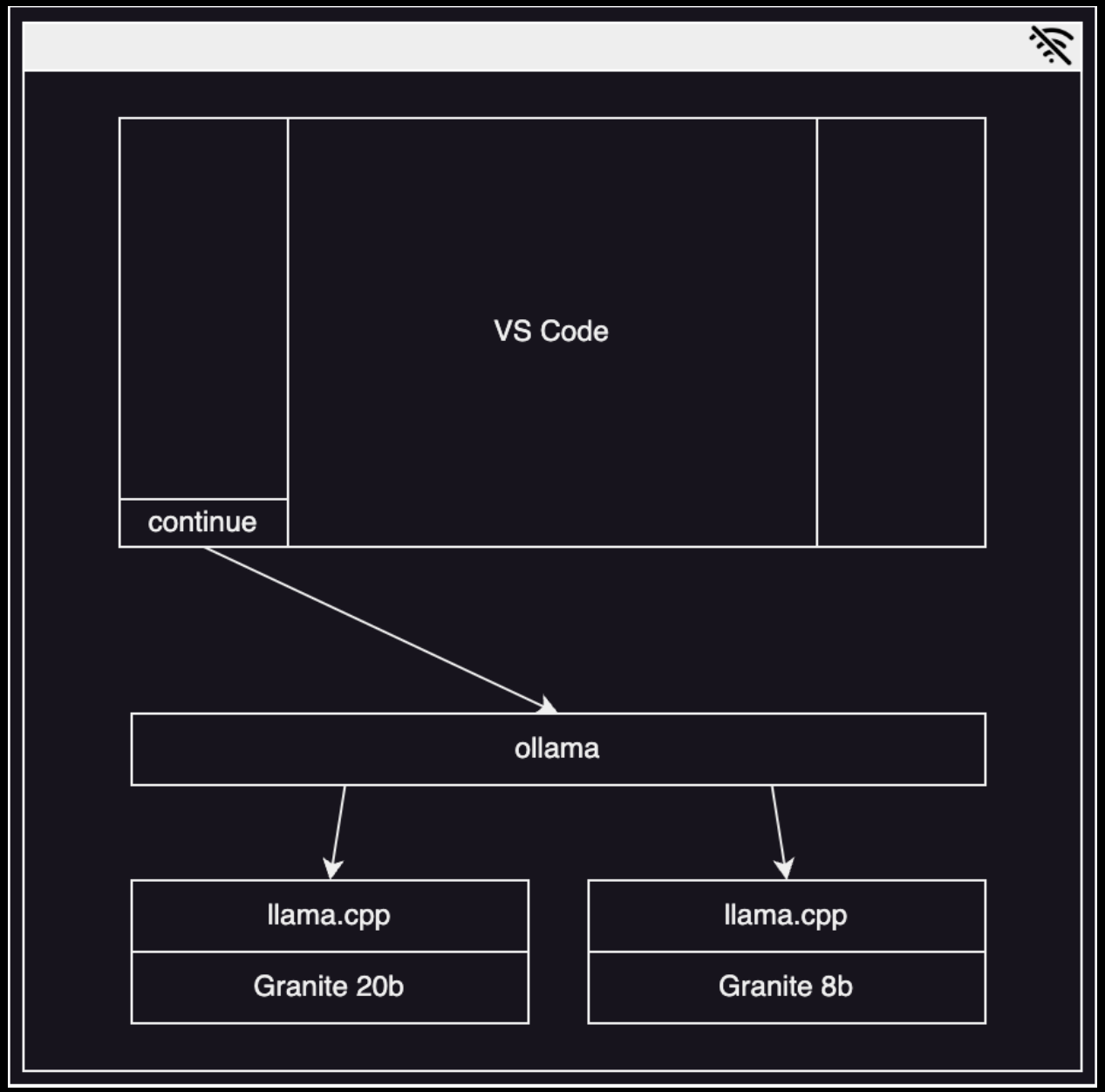

User
① User Question
⑧ Final Answer
②Question, ⑥Log
③Code, ⑦Ans
Commander
⑤Clearance
④Code
Repeat until answering the user's question or timeout
Writer
Safeguard

Continue

Continue enables you to easily create your own coding assistant directly inside Visual Studio Code and JetBrains with open-source LLMs. All this can run entirely on your own laptop or have Ollama deployed on a server to remotely power code completion and chat experiences based on your needs.

https://ollama.com/blog/continue-code-assistant

Ollama + Continue System Architecture



3

Microsoft Learn Student Ambassadors กับเส้นทางค้นหาเคสใหม่ๆ


Unraveling
the Complexities of
LLM Interpretability


Model Creator
Community Support
Engineering in AI
Equality
AI Governance
Wellbeing
Teamwork
Data and Insight + Life Principles
AI Leadership
Thai Language Tokenization
Responsible AI Practice
Use Case Inspiration
Content Creativity
ESL Practice
Engineer Principle
Academic and History
Academic
Copilot and Cloud Deployment
Innovation
Productivity
Science

My Use Cases From Previous Events

Python

- Simple Chatbot with Streamlit (Run with Phi-3)
- Multimodal Testing with Phi-3 Vision Model

github.com/chrnthnkmutt/phi3-py-experiment

JavaScript

- OpenAI API Deployment
- Text Summarizer
- RAG from CSV
- Text Translator
- Streaming Responses

github.com/chrnthnkmutt/phi3.5-js-experiment

.NET (C#)

- OpenAI API Deployment with Semantic Kernel (including Chat History)
- AutoGen Multimodal AI Agent

github.com/chrnthnkmutt/BoatSLM_cs_experiment

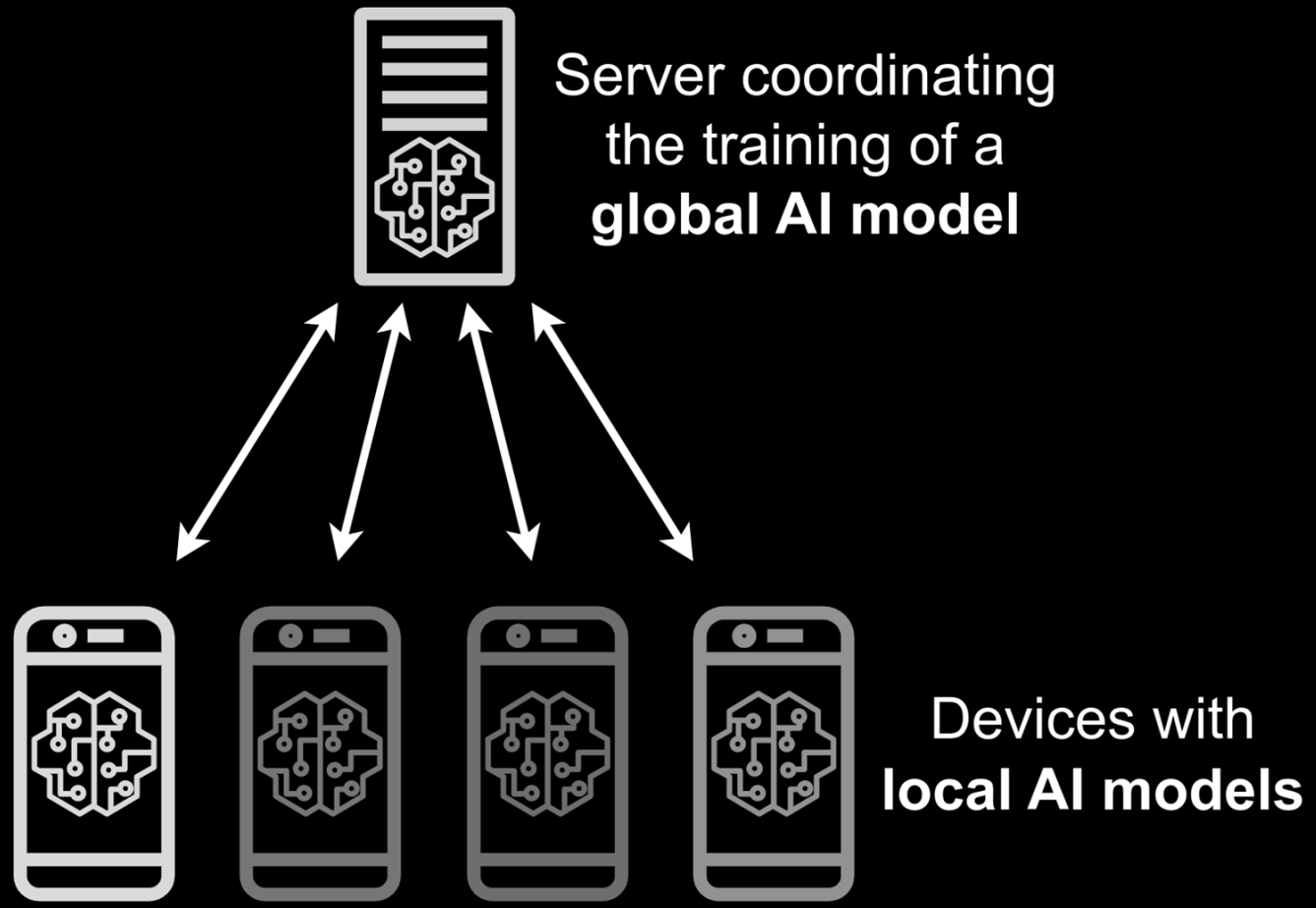
Upcoming Trends of Generative AI and NLPs Industry

- Shift to **multimodality**: Text → text, images, pictures, speech

- **Explainability** (Reason/Evidence for why the output is produce)
  Interpretability (understanding how the model works)
  Trustworthiness (safety and transparency of model usage)

- **Model Miniaturization**: separation between **language fluency** and **knowledge base** for AI on small devices (Laptop, Smartphones) with Multi Agents Compatibility

Federated Learning on Model Minituarization



Less-Parameters Language Models Arena

Microsoft

Phi-3.5: 3.8B (≈2.2 GB)
Thai-Lang Hallucination
Clear Responsible AI Guide
Academic

Meta

LLaMA 3.2 : 3B (≈2.02 GB)
Thai-Lang Supported
Clear Responsible AI Guide
General

Google DeepMind

Gemma 2 2B: ≈1.6 GB
Thai-Lang Hallucination
Clear Responsible AI Guide
General

Alibaba Cloud

Qwen2.5 3B (≈1.93 GB)
Thai-Lang Supported
Unclear Responsible AI Guide
General

SCB X

Typhoon 8B (≈4.9 GB)
Thai-Lang Supported
Meta's Responsible AI Guide
General

KBTG

THaLLE 7B (≈4.4 GB)
Thai-Lang Supported
Qwen-Based + Adding RAI
Financial/Business

Size Converted with llama.cpp/GGUF Size

Good Open Models for SLMs in Nowadays

Meta

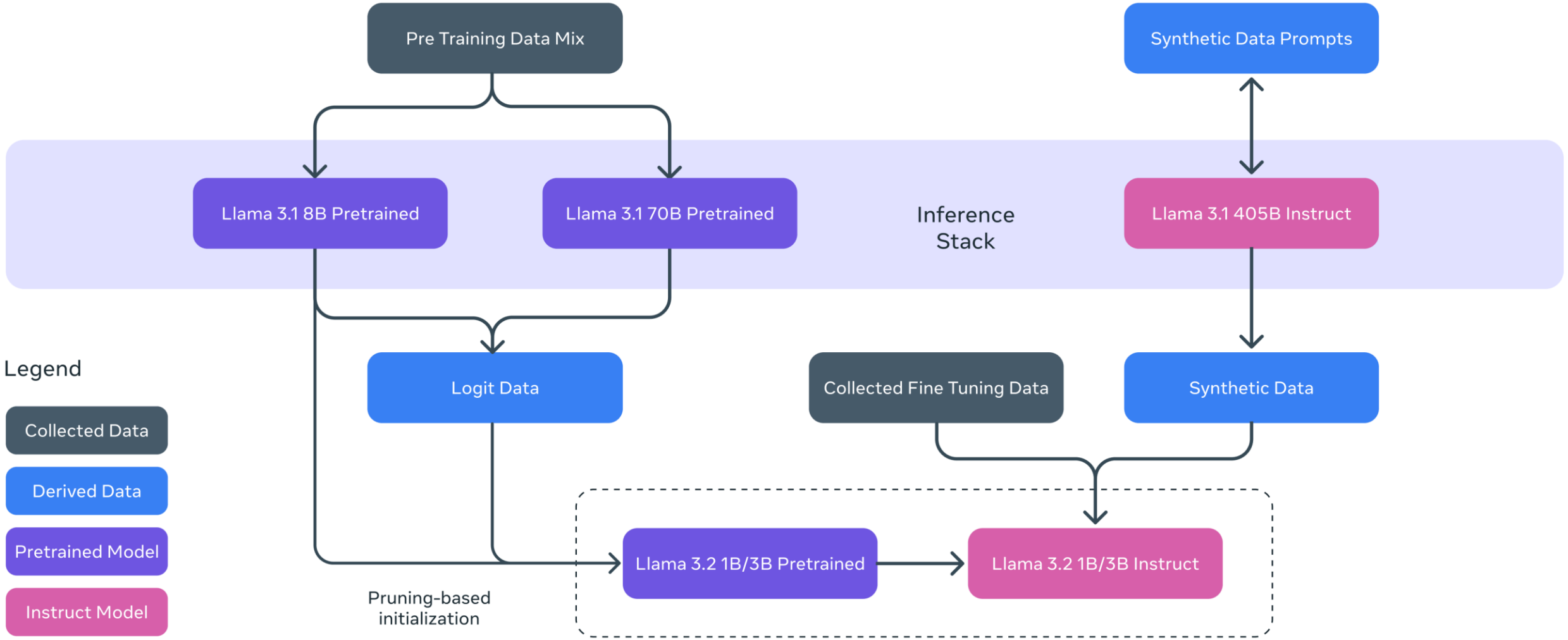
LLaMA 3.2 : 3B
(≈2.02 GB)
General Purposes

Responsible AI

- Privacy and Security
- Fairness and Inclusion
- Robust and Safety
- Transparency and Control
- Accountability and Governance

- Lightweight, text-only models that fit onto edge and mobile devices, including pre-trained and instruction-tuned versions.
- Support context length of 128K tokens and are state-of-the-art in their class for on-device use cases like summarization, instruction following, and rewriting tasks running locally at the edge.
- The Model empower developers to build personalized, on-device agentic applications with strong privacy where data never leaves the device. For example, such an application could help summarize the last 10 messages received, extract action items, and leverage tool calling to directly send calendar invites for follow-up meetings.
- **Outperforms the Gemma 2 2B and Phi 3.5-mini 3.8B models** on tasks such as following instructions, summarization, prompt rewriting, and tool-use, while the 1B is competitive with Gemma.
- Multilingual Support in one single model. (Including Thai Language)

1B & 3B Pruning & Distillation



WangchanThaiInstruct Dataset

100% Human-Annotated Thai Instruction Dataset (Batch 1-3 Release)

4 Domains:

- Medical
- **Finance**
- **Retail**
- Legal

7 Tasks:

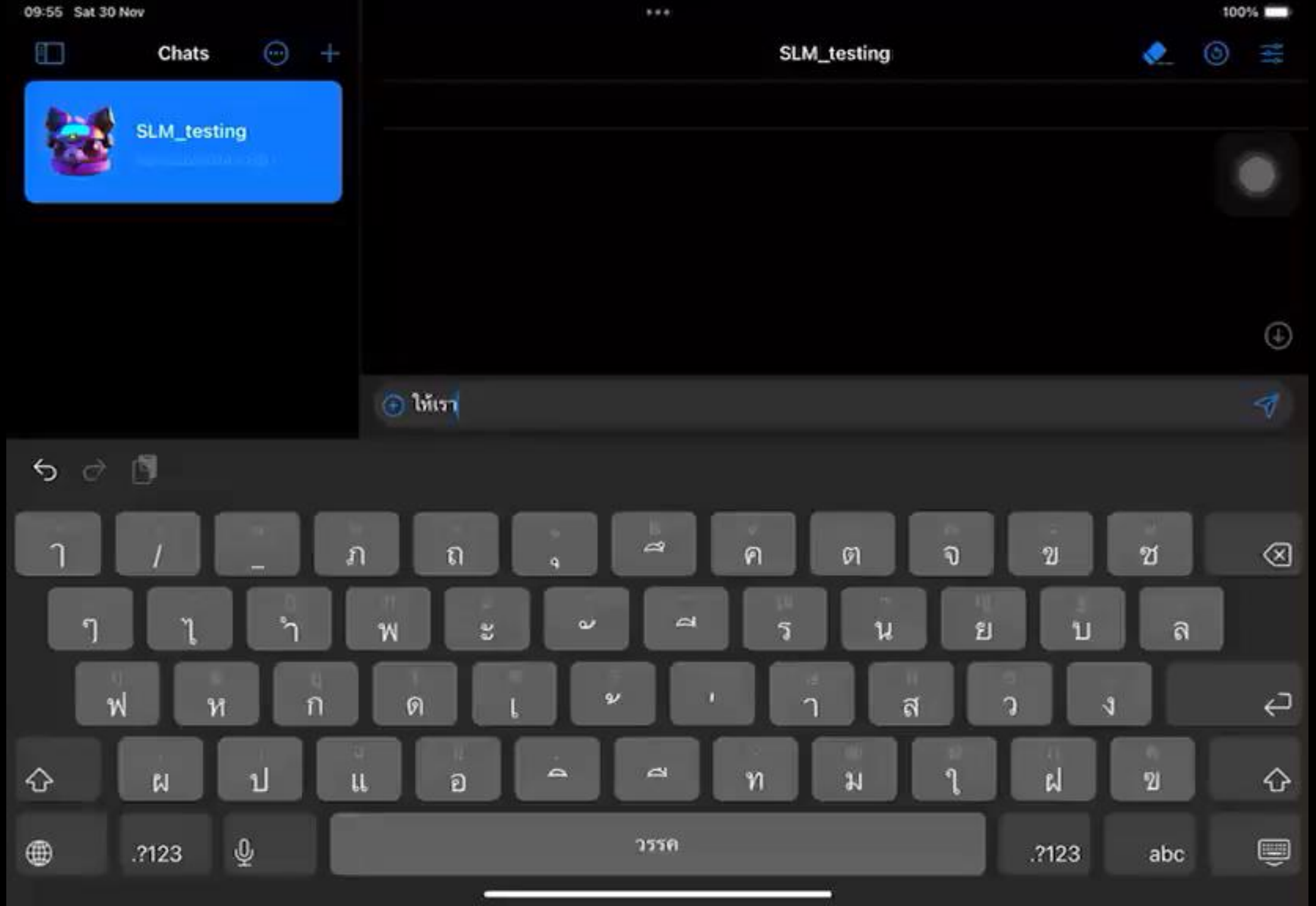
- Summarization
- Open QA
- Close QA
- Classification
- Creative Writing
- Brainstorming
- Multiple Choice QA

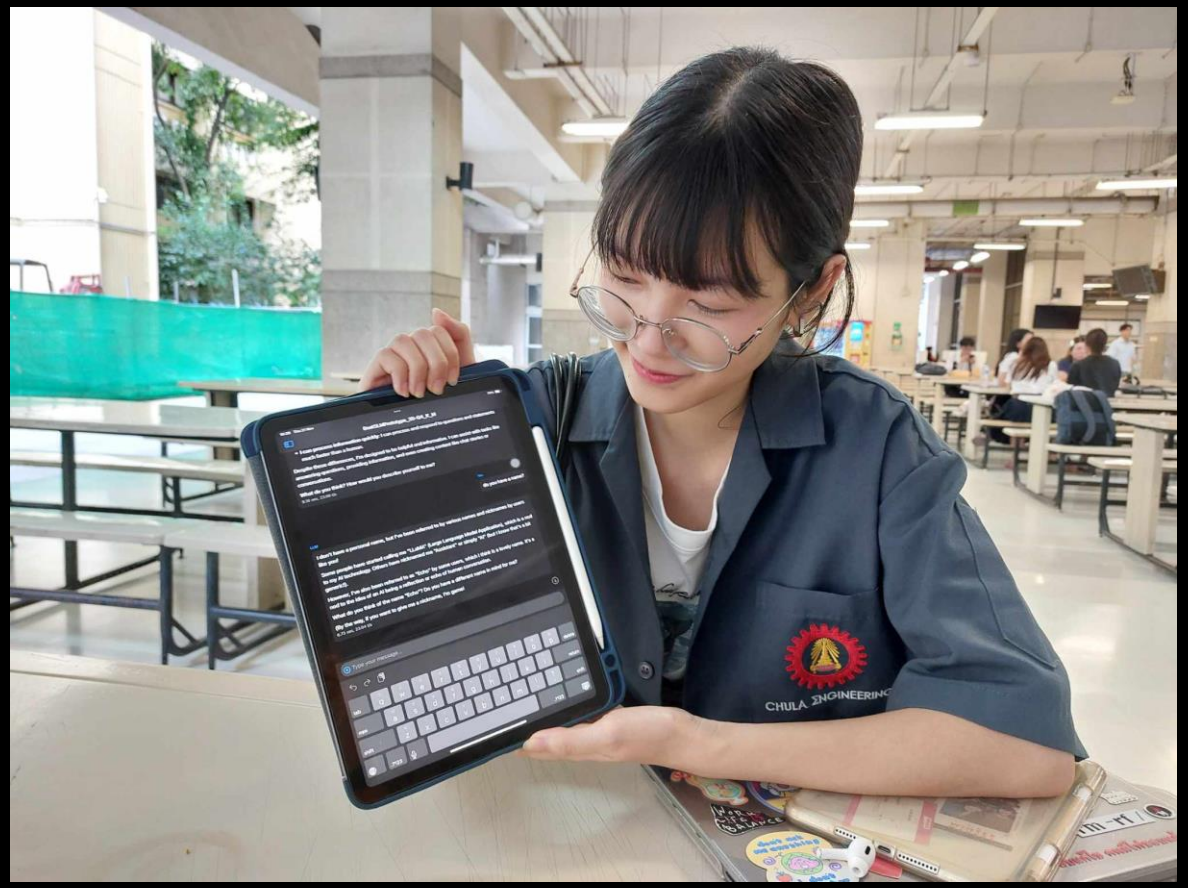
airesearch/WangchanThaiInstruct · Datasets at Hugging Face

The question is...

Why Thai People Still Doesn't Use Small Language Models?

- Hallucination & Performance
- Limited Device Resources
- Languages Barrier
- Low Engagement?


09:55 Sat 30 Nov
100%
Chats
SLM_testing
SLM_testing
ให้เรา
.?123
วรรค
.?123
abc


CHULA ENGINEERING


Hundreds of Thai Data & AI **Community Leaders** are in 2024
We hope to see you more than a **million** leaders and trainers in 2025 as a part of us :)

Escaping Yourself from "Know-it-all" to "Learn-it-all"

Thank You!

Charunthon Limseelo

@boatchrnthn

Charunthon Limseelo

Boat Charunthon (boatchrnthn)