QCon
全球软件开发大会

InfoQ 极客传媒

端侧大模型操作系统的架构、优化与展望

Mengwei Xu (徐梦炜)

Beijing University of Posts and Telecommunications
https://xumengwei.github.io

QCon
全球软件开发大会

InfoQ 极客传媒

目录

01 Mobile intelligence before LLM

02 On-device LLM

03 The changes LLM brings: App

04 The changes LLM brings: OS

05 The changes LLM brings: H/W

06 Takeaways

InfoQ 极客传媒

北京

QCon 全球软件开发大会

会议时间：4月10-12日

- 大模型赋能 AIOps
- 越挫越勇的大前端
- 云上业务架构演进
- 多模态大模型及应用
- 海外 AI 应用创新实践

北京

AiCon 全球人工智能开发与应用大会

会议时间：6月27-28日

- 端侧智能
- AI Agent
- 多模态大模型
- 金融+大模型

上海

QCon 全球软件开发大会

会议时间：10月23-25日

- AI Agent
- 大模型训练推理
- 端侧 AI
- 搜推深度融合
- 数智金融

4月 — 5月 — 6月 — 8月 — 10月 — 12月

上海

AiCon 全球人工智能开发与应用大会

会议时间：5月23-24日

- 通用大模型
- AI Agent
- 垂直领域应用
- 模型可解释性

深圳

AiCon 全球人工智能开发与应用大会

会议时间：8月22-23日

- 模型效率与部署
- 多模态大模型
- 大模型安全
- 智能硬件

北京

AiCon 全球人工智能开发与应用大会

会议时间：12月19-20日

- 通用大模型
- 智能硬件
- LMOps
- 具身智能

极客邦科技 2025 年会议规划

促进软件开发及相关领域知识与创新的传播

参会咨询

查看会议

Mobile intelligence before LLM

- DNNs already run on devices at large scale

DNN-embedded mobile apps

- Increased by almost 10x (2018 to 2021)[1,2]
- Downloaded billions of times in one year
- Include almost every high-popularity app
- Up to 200+ DNNs in a single app[3]

[1] Mengwei Xu, et al. “A First Look at Deep Learning Apps on Smartphones”. In WWW 2019
[2] Mario Almeida, et al. “Smart at what cost? Characterising Mobile Deep Neural Networks in the wild”. In IMC 2021.
[3] Through offline communication with application developers.

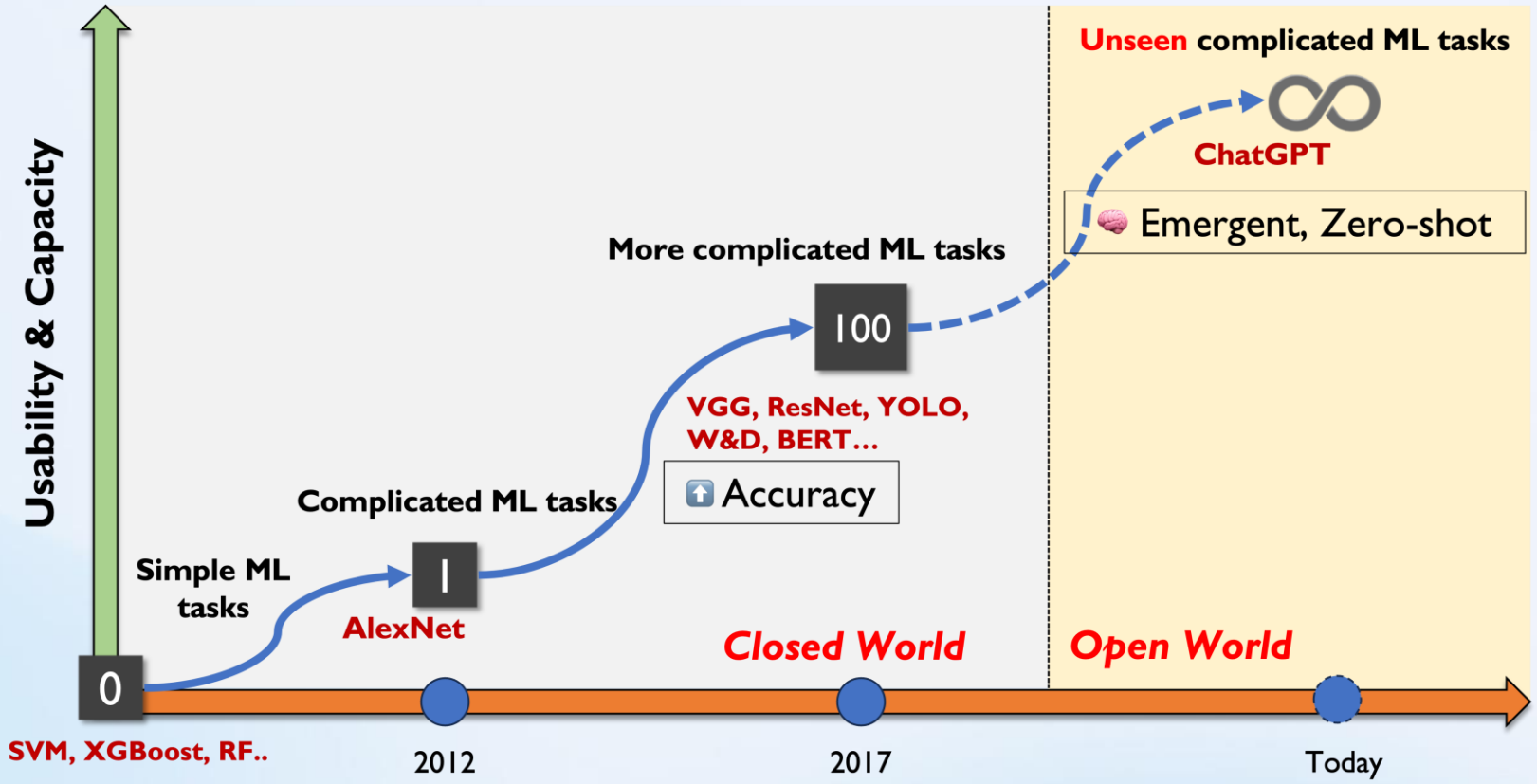
What we expect as mobile intelligence

A device that can

- Comprehend human language
- Reasoning & Planning
- Zero-shot & in-context learning
- Multimodal alignment
- Instruct. following

The opportunity: LLM

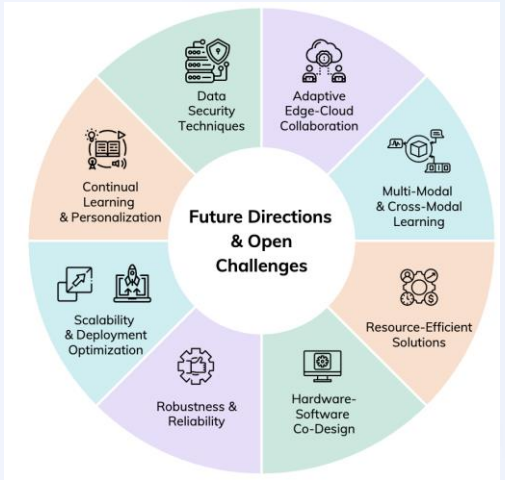
- To bring mobile devices the “next-level” intelligence



- Comprehend human language
- Reasoning & Planning
- Zero-shot & in-context learning
- Multimodal alignment
- Instruct. following

On-device LLM

- On-device LLMs handle language tasks in a way that is ..
  - ✓ cost-efficient (important, obviously)
  - ✓ more available (even w/o network)
  - ✓ faster (not always)
  - ✓ privacy-preserving (very important, LLMs can leverage almost every bits of local data)



- LLMs on devices does not obviate mega-scale LLMs on clouds!
  - Creating music/poetry, solving math problems, etc.

[1] Jiajun Xu, et al. “On-Device Language Models: A Comprehensive Review”. In preprint’24.

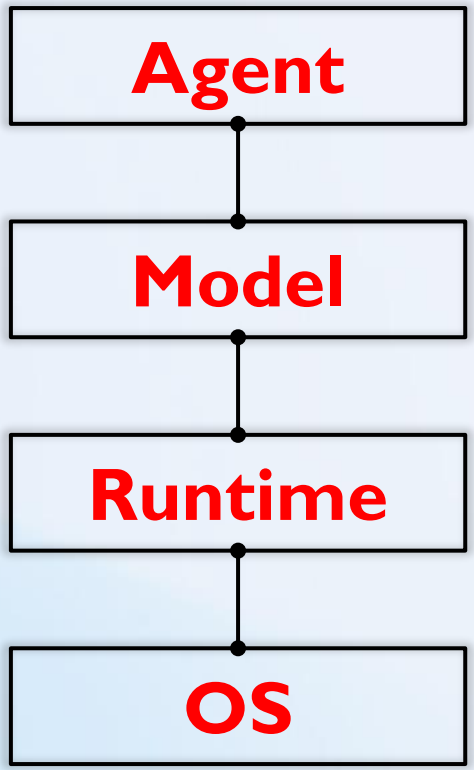
On-device LLM

- We already have a mobile device that can function with high intelligence!

A *mobile device* that can comprehend, reason, and plan without a cloud!

Call for full-stack design

- Our response: **agent-model-runtime-OS** co-design



Computer Use Agents (GUI or MCP) **testbed** [LlamaTouch, UIST'24], **datasets** [DroidCall, EMNLP'25][SHORTCUTSBENCH, ICLR'25], and **privacy** [SILENCE, NeurIPS'24]

A training-from-scratch, fully-reproducible **SLM family** [PhoneLM, preprint'24], Any-to-any modality **mobile foundation model** [M4, MobiCom'24] , and **Efficient Training** techniques [FwdLLM, ATC'24][AdaFL, MobiCom'23] [FeS, MobiCom'23]

Acceleration through **NPU** [llm.npu, ASPLOS'25], **SpecDecoding** [LLMCad, TMC'24] , **Sparsity** [EdgeMoE, TMC'25], **Early Exiting** [Recall, Nature Communications'25], etc

LLMaaS Context Management [LLMS, SenSys'26], **Elasticity** [ELMS, MobiCom'25] , **and Forward-compatibility** [LoRaSuite, NeurIPS'25]

Demo#1: Digital Agent

DroidCall: A Dataset for LLM-powered Android Intent Invocation

Weikai Xie [1] Li Zhang [1] Shihe Wang [1] Rongjie Yi [1] Mengwei Xu [1]

Agent

PhoneLM: an Efficient and Capable Small Language Model Family through Principled Pre-training

Rongjie Yi [1] Xiang Li [1] Weikai Xie [1] Zhenyan Lu [1] Chenghua Wang [1] Ao Zhou [1] Shangguang Wang [1] Xiwen Zhang [2] Mengwei Xu [1]

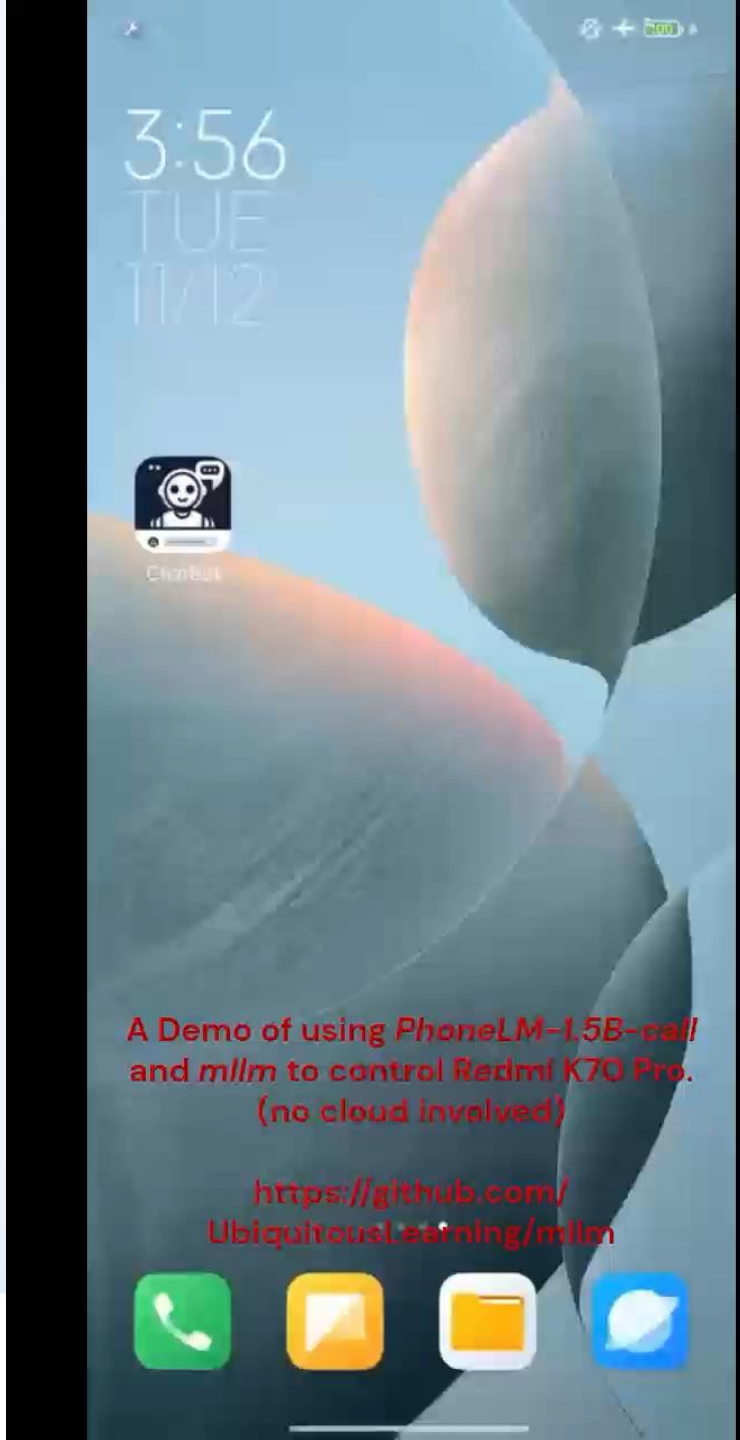
Model

Empowering 1000 tokens/second on-device LLM prefilling with mllm-NPU

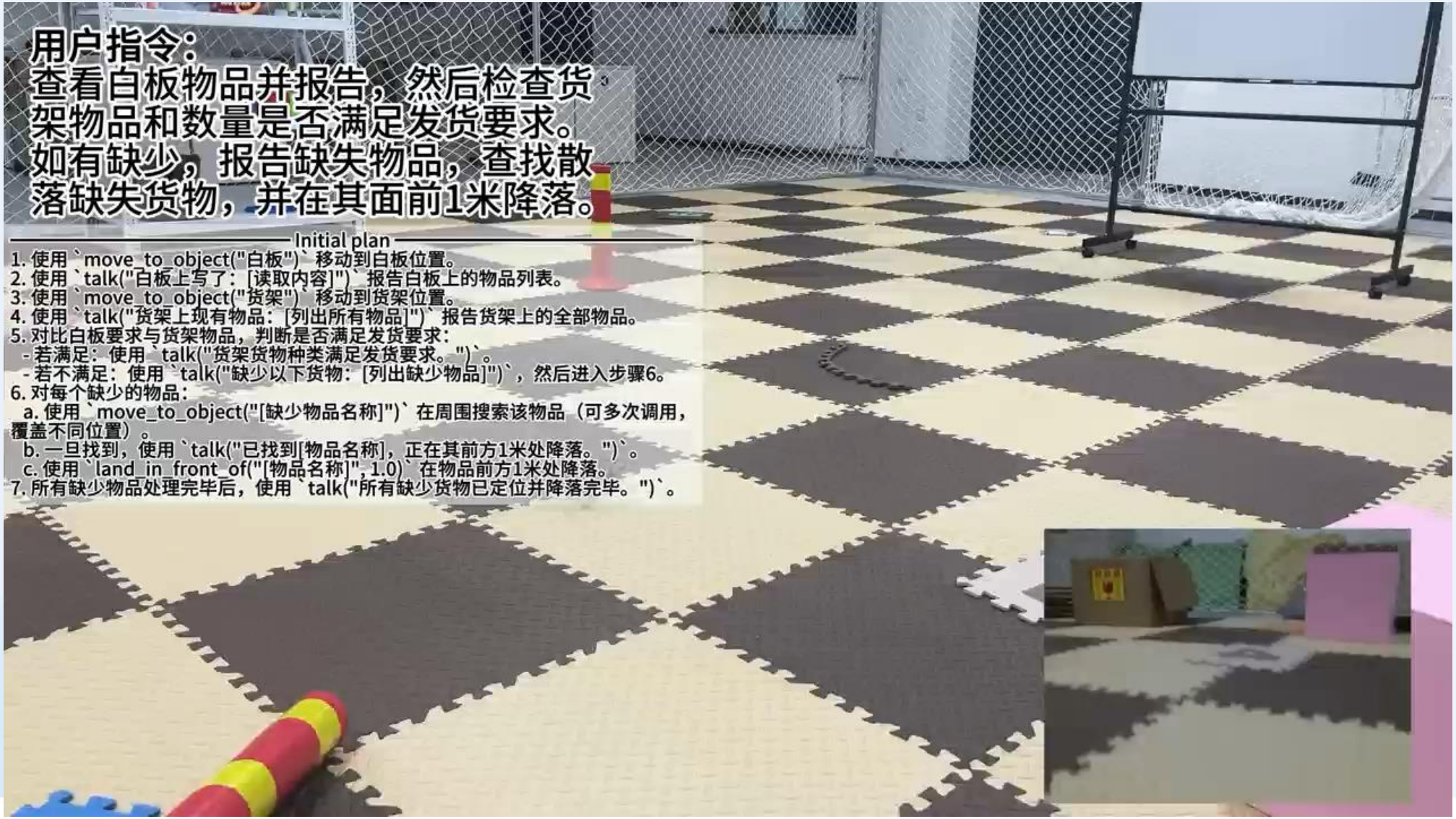
Daliang Xu♦, Hao Zhang◇, Liming Yang♦, Ruiqi Liu♦, Gang Huang♦, Mengwei Xu◇*, Xuanzhe Liu♦

♦Peking University, ◇Beijing University of Posts and Telecommunications

Runtime



Demo#2: UAV Agent



mllm: an NPU-centric LLM engine

Open source link: https://github.com/UbiquitousLearning/mllm

GUI Agent	API/Codegen Agent	Embodied Agents

LLM	**VLM**	**OMNI***	**MOE**	**VLA***
Llama, Qwen, MiniCPM, etc	Qwen-VL, LLaVA, etc	MiniCPM-o, etc	GPT-OSS, etc	OpenVLA, Pi, etc

Linear Attn.*	**Spec. Decoding**	**Token Pruning**	**KVCache Management**	**Experts Offload.**
Qwen3-NEXT, etc	Faster decoding (+NPU)	Faster Prefill	Paged Attn., Prefix Cache, etc	w/ storage hierarchy

Graph-level IR + Kernel-level optimization

Flash Attention 2, ARM KleidiAI, Qualcomm QNN, etc

Heterogeneous Hardware on Edge

Qualcomm, Ascend, AMD, Intel, OrangePi, etc..

* : on the way to go

The changes LLM brings

SW Stack

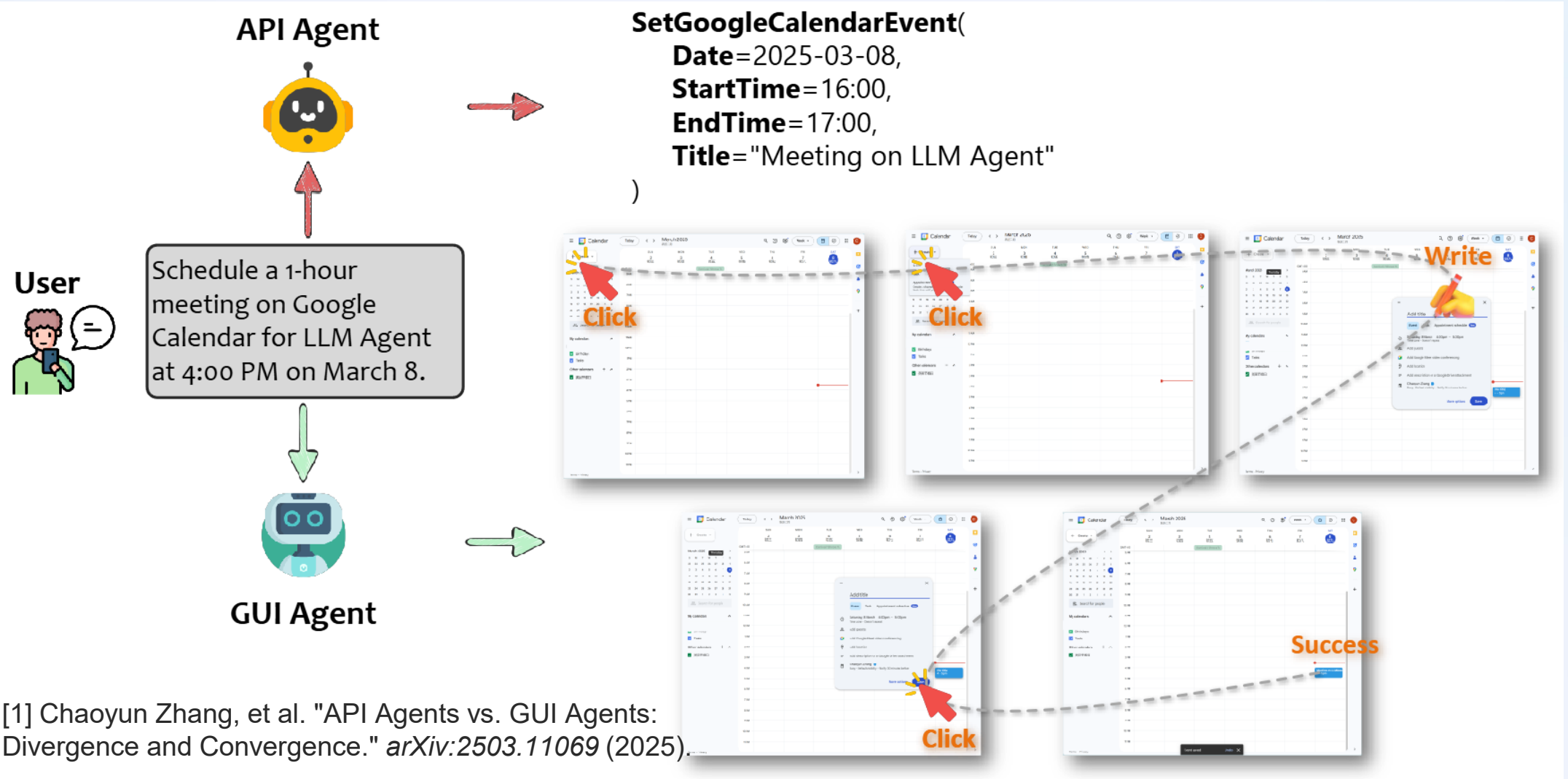
	Pre-LLM Era		LLM Era
App:	fragmented tasks	→	a unified agent
OS:	model-agnostic	→	LLM-native
H/W:	CPU/GPU-centric	→	NPU-centric

The changes LLM brings

	Pre-LLM Era		LLM Era
App:	fragmented tasks	→	a unified agent
OS:	model-agnostic	→	LLM-native
H/W:	CPU/GPU-centric	→	NPU-centric

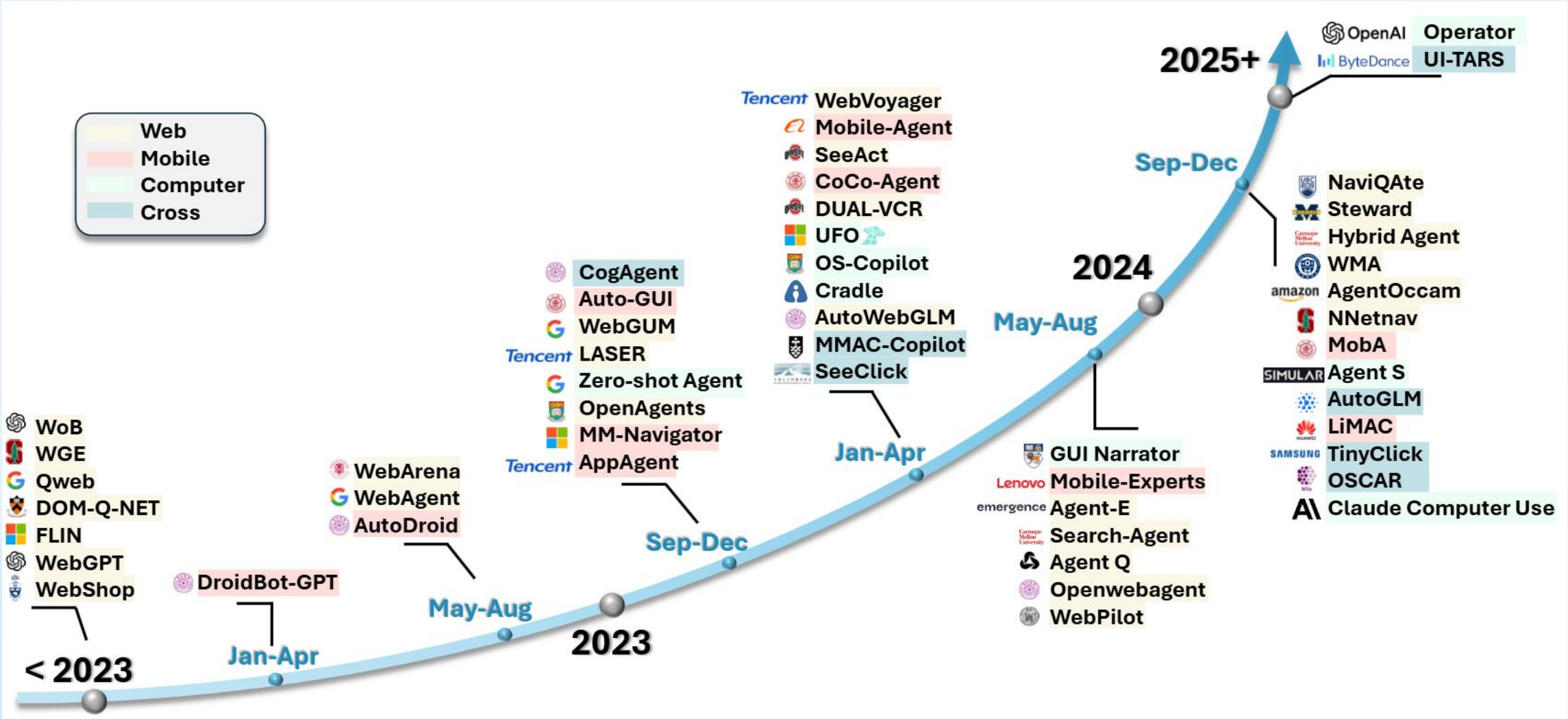
How to build a capable, generalized, and personalized mobile agent?

General approaches: API (MCP) vs. GUI



[1] Chaoyun Zhang, et al. "API Agents vs. GUI Agents: Divergence and Convergence." *arXiv:2503.11069* (2025).

GUI Agent: Status Quo



[1] Chaoyun Zhang, et al. "Large Language Model-Brained GUI Agents:A Survey." *arXiv:2411.18279* (2024).

GUI Agent: Status Quo

苹果发布多模态模型 Ferret-UI，部分手机 UI 任务超越 GPT-4V

赖文昕 AI科技评论 2024年04月10日 12:49 广东

支付宝突然推出新App，竟想用AI让日常生活开挂

荣耀MagicOS 9.0来了个全局智能体，AI手机方向变了

Original 关注AI智能体的 机器之心 2024年10月23日 20:21 北京

让大模型成为能够操控计算机的智能体，作者带来OmniParser V2详解

手机「自动驾驶」大揭秘！vivo万字综述探讨大模型手机自动化

机器之心 2025年01月07日 13:12 新加坡

让AI像人类一样操作手机，华为也做出来了

Agentic GLM全面登陆三星最新款手机Galaxy S25

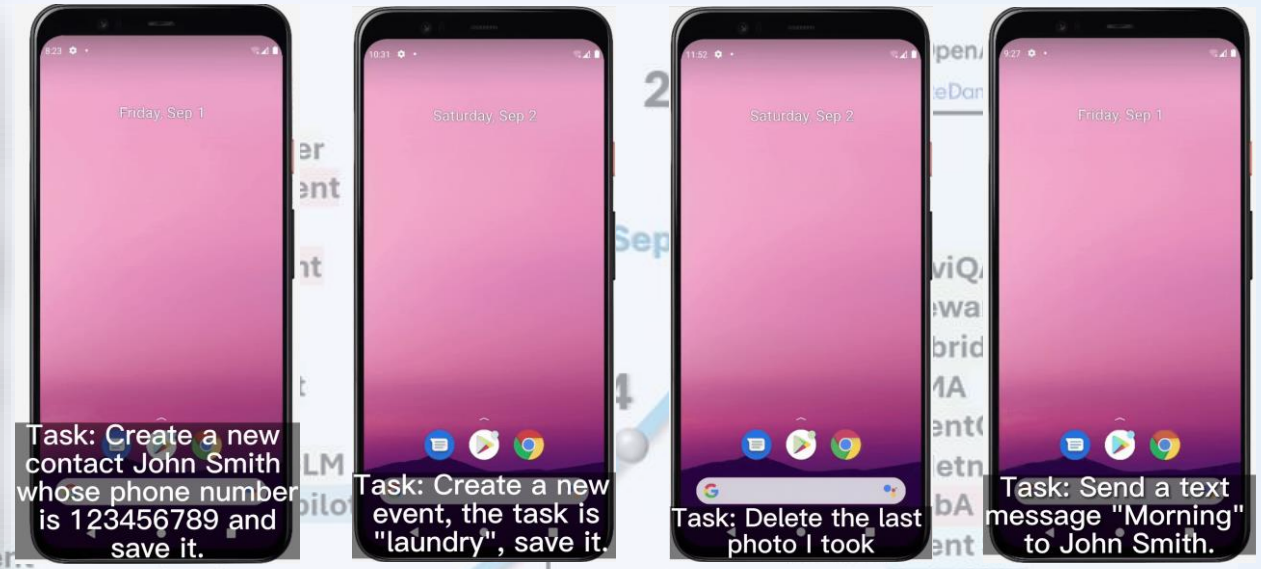
智谱 2025年02月11日 18:59 北京

AI智能体引擎加持：天玑9400让「完全体」AI手机提前问世了

Original 关注大模型的 机器之心 2024年10月15日 14:40 北京

八年磨一剑，国产首款AI Agent手机跑赢苹果！手机学会「自动驾驶」秀翻全场

新智元 新智元 2024年09月06日 20:55 北京



The reality: <60% accuracy, minutes-long for complex tasks.

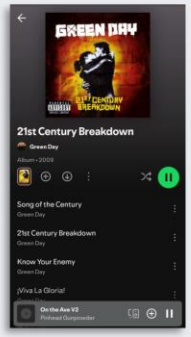
GUI Agent: Status Quo

SFT/RL/In-context Prompting

Instruction: *Display calendar in week view*

Response: *(0.30, 0.24)*

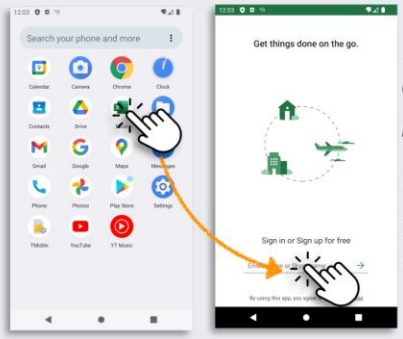
Grounding



Question: *Who created the album, and when?*

Response: *Green Day in 2019*

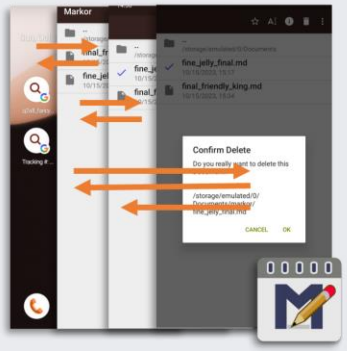
Screen Q&A



Instruction: *Open Excel and Login*

Response: *Click (0.38, 0.7)*

GUI Navigation



App Knowledge

VLMs
- GPT-4o
- Claude 3.7 Sonnet
- Qwen-VL

GUI Datasets
- Rico
- AndroidInTheWild
- MobileViews

Training

Input

GUI Agent

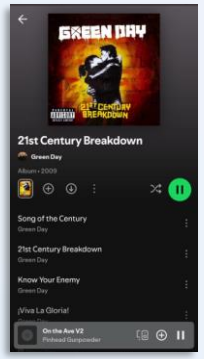
Agentic Workflow

Task Instruction

GUI Representation

Working Memory

Pixel-based Screenshot



Text-based View Hierarchy

```
<?xml version='1.0' encoding='UTF-8'>
<node index=0 class=FrameLayout>
 <node index=0 class=TextView text='21st Country Breakdown' />
 <node index=1 class=TextView text='Album 2009' />
</node>
<node index=1 class=FrameLayout>
</node>
```

Set-of-Mark Prompting

Tool Using

Multi-agent Collaboration

Reflection and Backtracking

Test-time Scaling

Interaction

Interactive Testbed
(Smartphone, Android Emulator)

- AndroidWorld
- LlamaTouch

Environment Perception

Online

GUI Agent: Status Quo

Self-supervised Reinforcement Learning

[arxiv'25] UIShift: Enhancing VLM-based GUI Agents through Self-supervised Reinforcement Learning

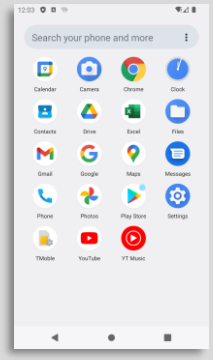
Why self-supervised RL?

- GUI trajectories (w/o annotations) can be scaled out easily
  - Using approaches like MobileViews
  - But human-labelled task instructions can not
    - DeepMind collects AndroidControl dataset (14K tasks) using 1 year – still far less than enough, and error-prone

- **RL is the way to generalize**
  - GUI data easily drift
  - Online or offline? An open problem, still

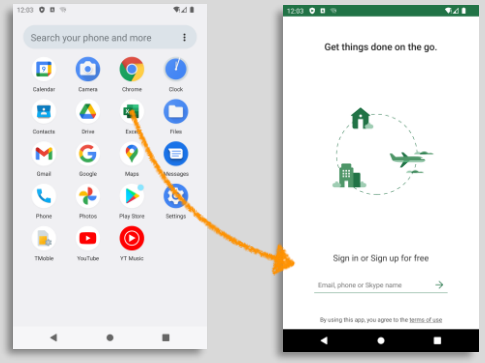
Why K-step GUI transition task?

- Rethinking the GUI tasks:
  - VLM can understand single GUI screen pretty well ✅
  - What's difficult is to understand/predict the GUI transitions ❌



- **Summary**: the image shows..
- **VQA**: how many app icons in this image? It has 21..
- **Grounding**: where is the chrome located in the image? It is at [0.2, 0.5]..
- Other single-image task

Easy tasks



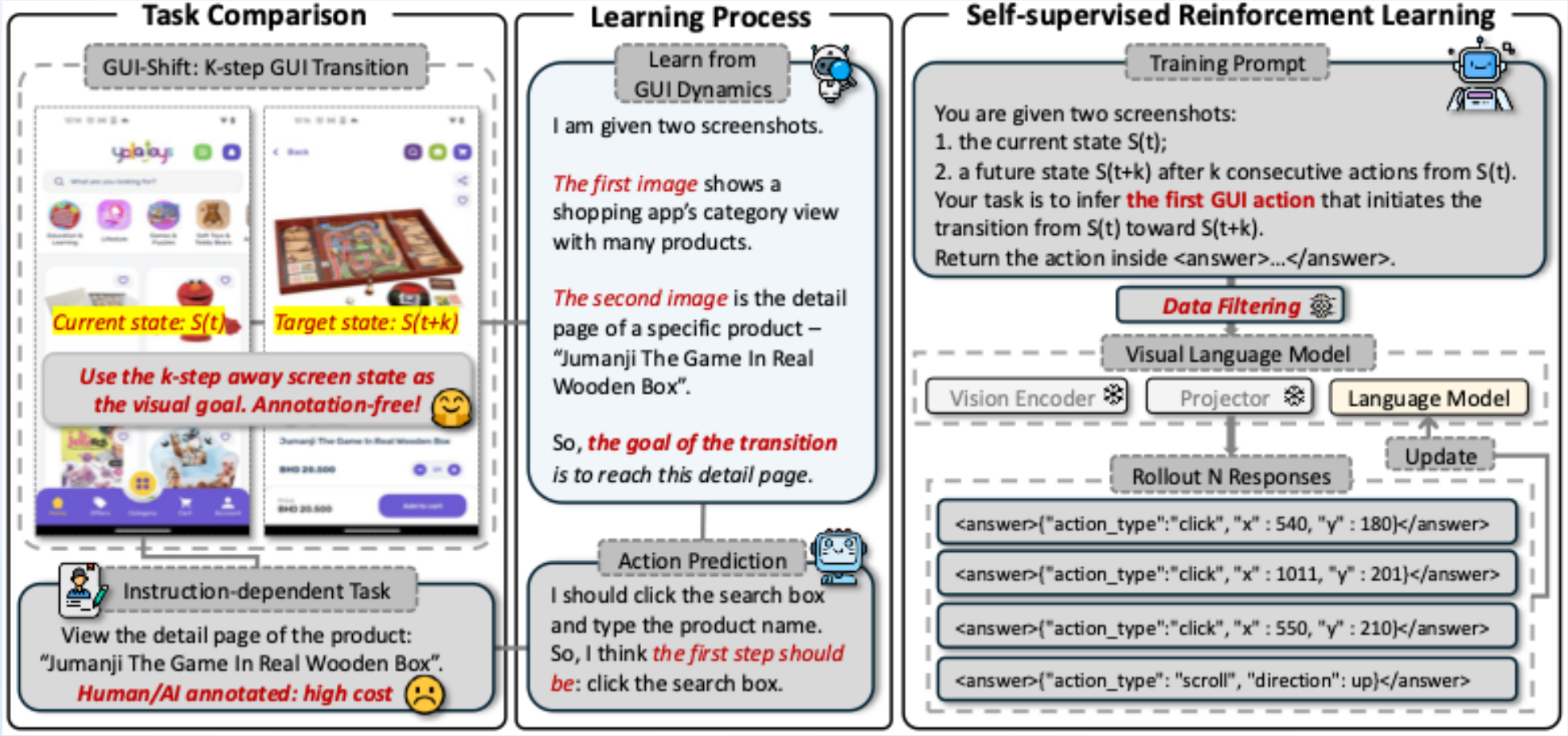
- **GUI relation**
- **Multi-hop planning**
- **Complex task automation**
- **GUI world model**
- ..

Challenging tasks

- The key is to embed GUI-to-GUI relation into the VLM/agent

K-step GUI transition

- Asking VLM to predict the action that leads to specific GUI transition

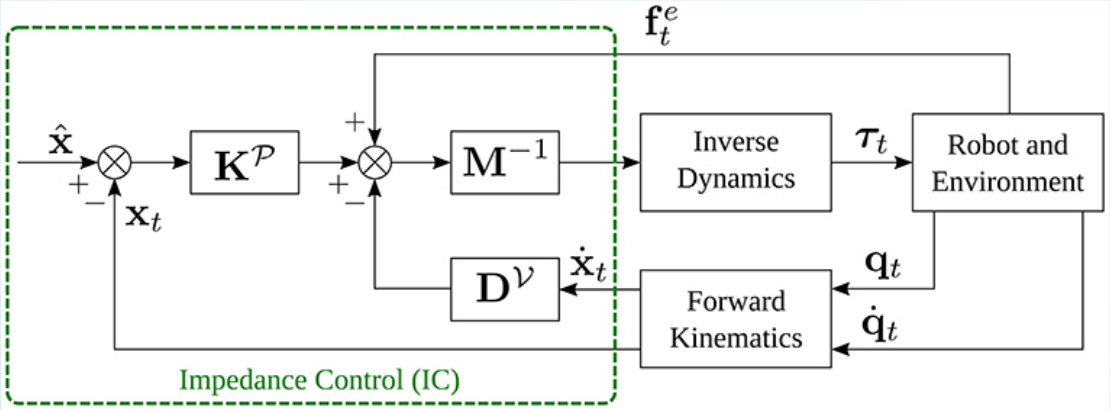


K–step GUI transition

- Asking VLM to predict the action that leads to specific GUI transition
  - Trained with data without human labels
  - Trained with GRPO[1]

  – A unified sampling and scoring mechanism during data filtering and training.

  - Inspired by **inverse dynamics** control theory



[1] Shao, Zhihong, et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." *arXiv preprint arXiv:2402.03300* (2024).

K-step GUI transition

- Asking VLM to predict the action that leads to specific GUI transition
  - Trained with data without human labels
  - Trained with GRPO[1]
  - **A unified sampling and scoring mechanism during data filtering and training.**
  - Inspired by **inverse dynamics** control theory
- **We conduct small-scale experiments**
  - **2K filtered K-step GUI Transition data sampled from AndroidControl, without using its task instructions**
  - **Models: Qwen2.5-VL-7B, InternVL3-8B, Mimo-VL-7B-SFT, Mimo-VL-7B-RL**
  - **We experiment w/ and w/o data filtering, w/ and w/o CoT reasoning**
  - **We want to show if it's as good as SFT, and whether it can scale out**

Results on GUI automation and grounding

- GUI-Shift outperforms models trained with large-scale task instructions.

Table 1: Performance comparison on GUI task automation benchmarks: AndroidControl (AC-Low, AC-High) and GUI Odyssey. GUI-Shift achieves substantial improvements over base models. **Bold**: the best result; underlined: the second best result. TM: type match; EM: exact match.

Model	# Training Samples	AC-Low		AC-High		GUI Odyssey	
		TM	EM	TM	EM	TM	EM
Proprietary models							
GPT-4o (OpenAI, 2024)	-	74.3	19.4	66.3	20.8	34.3	3.3
Models trained with annotations							
SeeClick (Cheng et al., 2024)	1M	93.0	75.0	82.9	59.1	71.0	53.9
OS-Atlas-7B (Wu et al., 2024b)	2.3M	93.6	85.2	85.2	71.2	-	62.0
Aguvis-7B (Xu et al., 2024)	1M	-	80.5	-	61.5	-	-
UI-TARS-7B (Qin et al., 2025)	-	98.0	90.8	83.7	72.5	**94.6**	**87.0**
UI-R1-3B (Lu et al., 2025)	136	94.3	88.5	57.9	45.4	52.2	32.5
GUI-R1-7B (Xia & Luo, 2025)	3K	85.2	66.5	71.6	51.7	65.5	38.8
InfiGUI-R1-3B (Liu et al., 2025)	32K	96.0	92.1	82.7	71.1	-	-
AgentCPM-GUI (Liu et al., 2025)	470K	94.4	90.2	77.7	69.2	90.9	75.0
UI-Venus-Navi-7B (Gu et al., 2025)	350K	97.1	92.4	86.5	**76.1**	87.3	71.5
Ours: Qwen2.5-VL-7B as the base model							
Qwen2.5-VL-7B (Bai et al., 2025)	-	94.9	83.8	72.9	59.2	59.8	44.9
GUI-Shift-Qwen ($k = 1$)	2K	98.0↑3.1	90.6↑6.8	85.9↑13.0	70.4↑11.2	78.5↑18.7	54.8↑9.9
Ours: InternVL3-8B as the base model							
InternVL3-8B (Chen et al., 2024)	-	97.8	90.0	71.5	49.8	48.8	20.3
GUI-Shift-Intern ($k = 4$)	2K	97.3↓0.5	88.0↓2.0	78.5↑7.0	56.6↑6.8	59.6↑10.8	23.3↑3.0
Ours: Mimo-VL-7B-SFT as the base model							
Mimo-VL-7B-SFT (Xiaomi, 2025b)	-	90.8	85.7	75.2	63.1	86.9	62.0
GUI-Shift-Mimo-SFT ($k = 3$)	2K	98.6↑7.8	**93.2**↑7.5	**87.2**↑12.0	73.4↑10.3	86.1↓0.8	60.7↓1.3
Ours: Mimo-VL-7B-RL as the base model							
Mimo-VL-7B-RL (Xiaomi, 2025b)	-	91.8	87.2	76.5	64.6	87.2	63.1
GUI-Shift-Mimo-RL ($k = 1$)	2K	**98.9**↑7.1	**93.2**↑6.0	86.9↑10.4	71.7↑7.1	84.8↓2.4	59.5↓3.6

Table 2: Performance comparison on GUI grounding benchmarks: ScreenSpot-v2 and ScreenSpot-Pro. GUI-Shift exhibits strong generalization and achieves the second best result on ScreenSpot-Pro. **Bold**: the best result; underlined: the second best result.

Model	# Training Samples	ScreenSpot-v2							-Pro
		Mobile		Desktop		Web		Avg.	Avg.
		Text	Icon	Text	Icon	Text	Icon		
Models trained with annotations									
CogAgent-18B (Wang et al., 2024)	-	-	-	-	-	-	-	-	7.7
SeeClick-9.6B (Cheng et al., 2024)	1M	78.4	50.7	70.1	29.3	55.2	32.5	55.1	1.1
UGround-7B (Gou et al., 2024)	1.3M	75.1	84.5	85.1	61.4	84.6	71.9	76.3	16.5
OS-Atlas-7B (Wu et al., 2024b)	2.3M	95.2	75.8	90.7	63.6	90.6	77.3	84.1	18.9
ShowUI-2B (Lin et al., 2024)	256K	-	-	-	-	-	-	-	7.7
UI-TARS-7B (Qin et al., 2025)	-	96.9	89.1	95.4	85.0	93.6	85.2	91.6	35.7
UI-R1-E-3B (Lu et al., 2025)	2K	83.0	**97.1**	85.0	**91.7**	77.9	**95.4**	89.5	33.5
InfiGUI-R1-3B (Liu et al., 2025)	32K	-	-	-	-	-	-	-	35.7
LPO (Tang et al., 2025)	-	97.9	82.9	**95.9**	86.4	95.6	84.2	90.5	-
UI-Venus-Ground-7B (Gu et al., 2025)	107K	**99.0**	90.0	97.0	90.7	**96.2**	88.7	**94.1**	**50.8**
Ours: Qwen2.5-VL-7B as the base model									
Qwen2.5-VL-7B (Bai et al., 2025)	-	98.3	86.3	88.7	67.1	92.7	81.8	87.7	26.4
GUI-Shift-Qwen ($k = 4$)	2K	98.6	89.6	86.1	75.0	92.7	82.8	89.0↑1.3	27.1↑0.7
Ours: InternVL3-8B as the base model									
InternVL3-8B (Chen et al., 2024)	-	93.4	81.5	80.4	52.1	91.0	73.4	81.3	15.0
GUI-Shift-Intern ($k = 1$)	2K	93.8	83.4	80.4	51.4	91.0	73.4	81.6↑0.3	15.4↑0.4
Ours: Mimo-VL-7B-SFT as the base model									
Mimo-VL-7B-SFT (Xiaomi, 2025b)	-	96.6	84.4	92.8	80.0	88.9	76.8	87.6	39.8
GUI-Shift-Mimo-SFT ($k = 1$)	2K	98.3	87.7	92.3	82.1	94.0	79.8	90.1↑2.5	40.7↑0.9
Ours: Mimo-VL-7B-RL as the base model									
Mimo-VL-7B-RL (Xiaomi, 2025b)	-	98.3	86.3	90.2	80.7	92.7	75.4	88.4	40.2
GUI-Shift-Mimo-RL ($k = 1$)	2K	**99.0**	87.7	91.2	83.6	89.7	72.9	88.4↑0.0	41.7↑1.5

Ablation study：GRPO vs. SFT

- GRPO is more suitable than SFT for K-step GUI Transition.

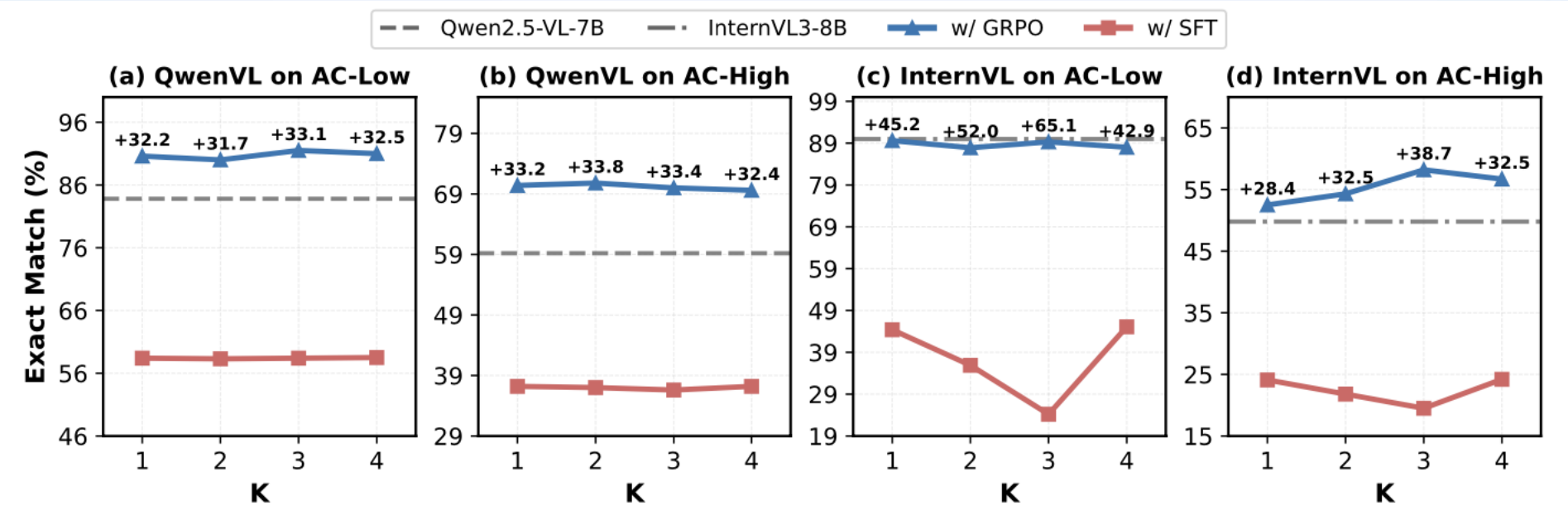


Figure 3: Comparison of training algorithms for the K-step GUI Transition task ($k \in \{1, 2, 3, 4\}$). Qwen2.5-VL-7B and InternVL3-8B are fine-tuned with 2K samples for each k and evaluated on AndroidControl. GRPO provides notable performance gains over SFT for all models and settings.

Our Roadmap to GUI Agent

- Starting from a capable Visual Language Model
- Step-1: Self-supervised Reinforcement Learning
  - Large quantity of trajectory data, no tasks labelled
  - Building “World Model” for GUI
  - Static data + Offline RL
- Step-2: Task-guided Reinforcement Learning
  - Small scale with human labelled tasks and trajectories
  - Real Testbeds + Online RL
- Step-3: Online Agent Design

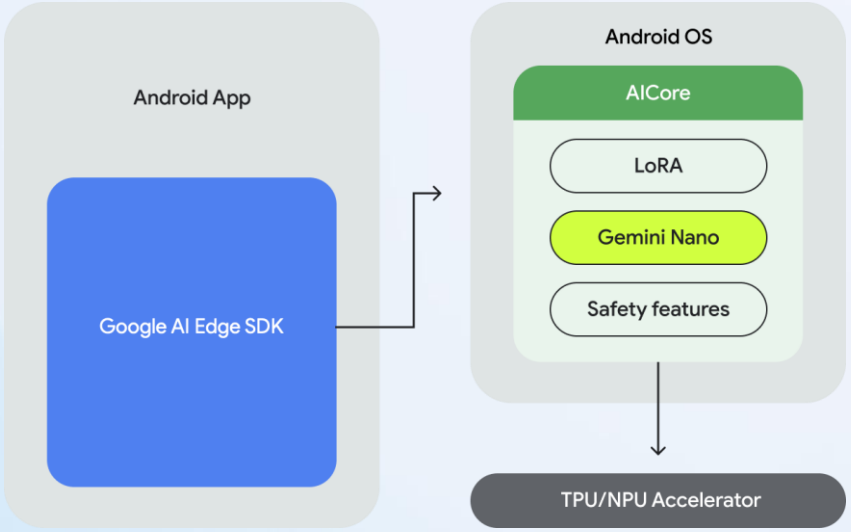
The changes LLM brings

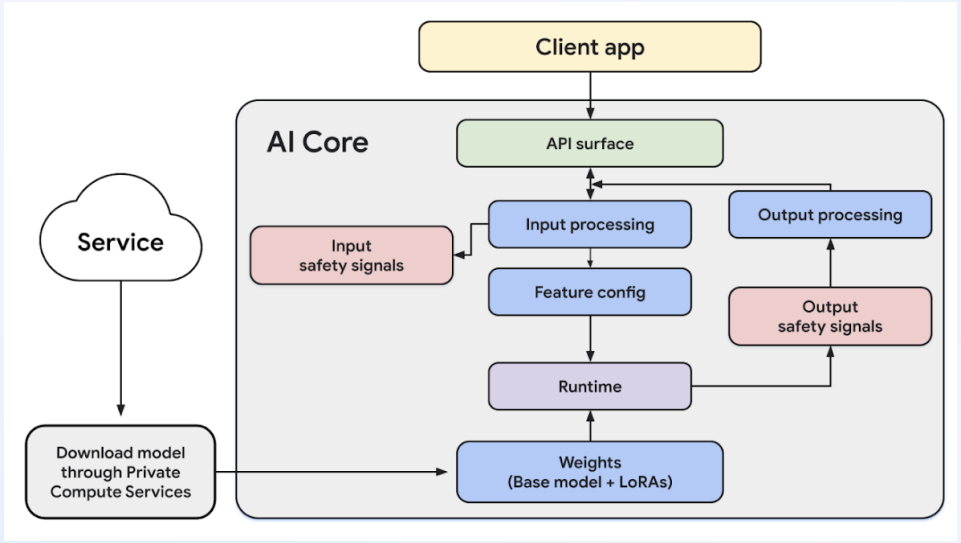
	Pre-LLM Era		LLM Era
App:	fragmented tasks	→	a unified agent
OS:	model-agnostic	→	LLM-native
H/W:	CPU/GPU-centric	→	NPU-centric

How should OS better serve/manage device-wise LLM requests?

LLM as an OS service

- LLM integrated into OS as a **system service**
  - Scales to infinite number of tasks
  - Hardware-design-friendly
  - OS gains full visibility into LLM requests





[1] Source: https://developer.android.com/ai/gemini-nano

Challenges of LLMaaS

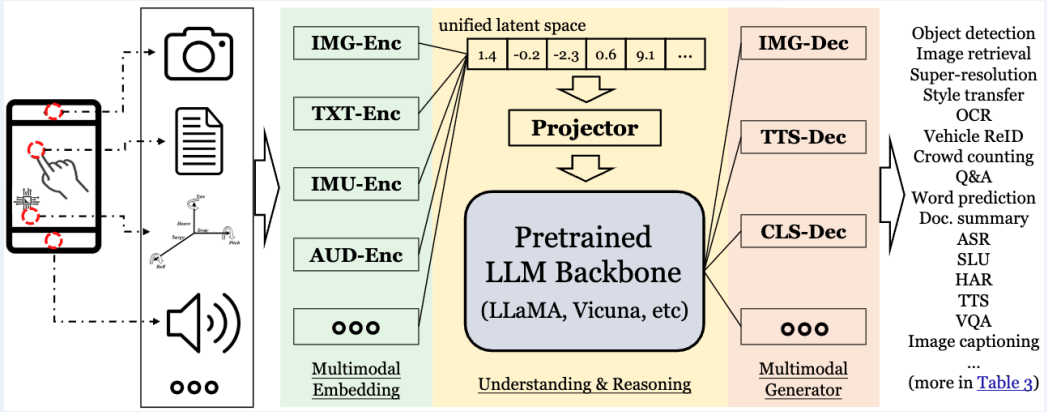
- Opening new research opportunities and challenges
  - *Usability*: How to design LLMaaS interface? How to upgrade LLM?
    - *[MobiCom'24] Mobile Foundation Model as Firmware*
    - *[NeurIPS'25] Efficient LoRA Adaptation Across Large Language Model Upgrades*
  - *Efficiency*: How to schedule, batch, and cache-reuse system-wise LLM requests? How to manage the LLM context states across apps?
    - *[MobiCom'25] Elastic On-Device LLM Service*
    - *[SenSys'26] LLM as a System Service on Mobile Devices*
  - *Security*: how to protect app-owned LoRa? How to isolate cross-app requests?
  - Etc..

Challenges of LLMaaS

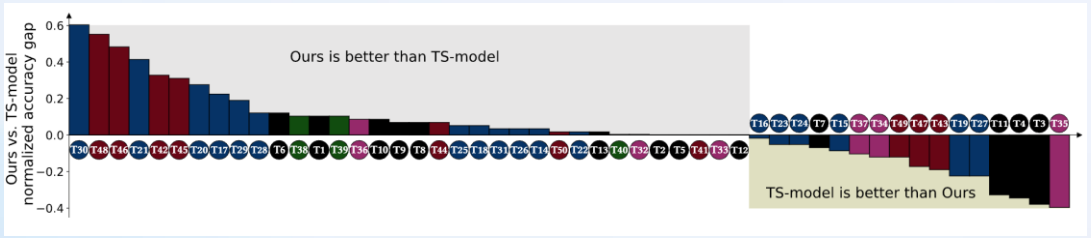
- Opening new research opportunities and challenges
  - *Usability*: How to design LLMaaS interface? How to upgrade LLM?
    - ***[MobiCom'24] Mobile Foundation Model as Firmware***
    - *[NeurIPS'25] Efficient LoRA Adaptation Across Large Language Model Upgrades*
  - *Efficiency*: How to schedule, batch, and cache-reuse system-wise LLM requests? How to manage the LLM context states across apps?
    - *[MobiCom'25] Elastic On-Device LLM Service*
    - *[SenSys'26] LLM as a System Service on Mobile Devices*
  - *Security*: how to protect app-owned LoRa? How to isolate cross-app requests?
  - Etc..

M4: a one-size-fits-all mobile MLLM

- Can one model (as an OS service) solve all mobile AI tasks?



M4: Any-to-any modality MLLM



M4 outperforms prior arts on most tasks

Category	Tasks	Mobile Application	Dataset	Specific-Models	Results	Metrics
NLP	Input word prediction T1	Input method (GBoard)	PTB	RNN [23]	0.17*	Accuracy
	Question answering T2 T3	Private assistant (Siri)	SQuAD v2.0	RoBERTa [35]	0.79*	F1
			TyDi QA	AraELECTRA [36]	0.87	F1
	Machine translation T4	Translator (Google Translate)	wmt22 en-de	Transformer [32]	0.34*	BLEU
	Emoji prediction T5	Input method (GBoard)	tweet_eval	RoBERTa [22]	0.33*	Accuracy
	Emotion prediction T6	Conversational analytics (Clarabridge)	go_emotion	RoBERTa [29]	0.57*	Accuracy
	Sentiment analysis T7	Conversational analytics (Clarabridge)	tweet_eval	RoBERTa [27]	0.77*	Accuracy
	Text classification T8 T9	Spam SMS filtering (Truecaller)	ag_news	BERT [37]	0.93*	Accuracy
			SST2	DistilBERT [38]	0.91*	Accuracy
	Grammatical error correction T10	Writing assistant (Grammarly)	JFLEG	FLAN-t5 [30]	0.68*	BLEU
	Text summary T11	Reading assistant (ChatPDF)	CNN Daily Mail	BART [5]	0.43*	ROUGE1
	Code document generation T12	Code editor (Javadoc)	CodeSearchNet	CodeT5-base [20]	0.33*	ROUGE1
	Code generation T13	Code editor (Copilot)	Shellcode_IA32	CodeBERT [21]	0.92	BLEU
CV	Object detection T14 T15	Augmented Reality (Google Lens)	COCO	Libra-rcnn [24]	0.43*	mAP
			LVIS	X-Paste [33]	0.51	AP
	Image retrieval T16	Image searcher (Google Photos)	Clothes Retrieval	Resnet50-arcface [31]	0.90*	Recall
	Super-resolution T17	Video/Image super-resolution (VSCO)	set5	Real-ESRGAN [19]	0.82*	SSIM
	Styler transfer T18	Painting & Beatifying (Meitu)	COCO, Wikiart	StyleGAN-nada [4]	0.23	CLIP score
	Semantic segmentation T19 T20	Smart camera (Segmentix)	ADE20K-150	Deeplabv3plus [25]	0.43*	mIoU
			PASCAL VOC	Deeplabv3plus [26]	0.79*	mIoU
	Optical character recognition T21	Intelligent document automation	Rendered SST2	CLIP [34]	0.71	Accuracy
	Im					y
	Tr					y
	Ve					
	Gender recognition T26	Smart camera (Face++)	Adience	MiVOLO-D1 [2]	0.96	Accuracy
	Location recognition T27	Navigation search (Google Maps)	Country211	CLIP [34]	0.46	Accuracy
	Pose estimation T28	AI fitness coach (Keep)	AP-10K	ViTPose [134]	0.69	AP
	Video classification T29	Video player (YouTube)	kinetics400	SlowFast [28]	0.79	Accuracy
	Crowd Counting T30	Smart camera (Fitness Tracking)	UCP-QNRF	CSS-CCNN [12]	437	MAE
	Image matting T31	Virtual backgrounds (Zoom)	RefMatte-RW100	MDETR [79]	0.06	MSE
Audio	Automatic speech recognition T32	Private assistant (Siri)	LibriSpeech	CTC+attention [14]	3.16%*	WER
	Spoken language understanding T33 T34	Private assistant (Siri)	FSC	Transformer [18]	0.37%	WER
			SLURP	CRDNN [3]	0.82*	Accuracy
	Emotion recognition T35	Emoji recommendation (WeChat)	IEMOCAP	ECAPA-TDNN [15]	0.64*	Accuracy
	Audio classification T36	Music discovery (Shazam)	ESC-50	ACDNet [1]	0.87	Accuracy
	Keyword spotting T37	Private assistant (Siri)	Speech command	Cnn-trad-fpool3 [17]	0.88*	Accuracy
Sensing	Human activity recognition T38 T39 T40	AI fitness coach (Keep)	Using Smartphones	TS-TCC [51]	0.90	Accuracy
			HHAR	LIMU-BERT [16]	0.84	Accuracy
			MotionSense	LIMU-BERT [16]	0.91	Accuracy
Multimodal	Text-to-speech T41	Voice broadcast (WeChat reading)	LJSpeech	Transformer [13]	3.26	MCD
	Audio captioning T42 T43	Hearing-impaired accessibility (Ava)	Clotho	Transformer [10]	0.52*	BLEU
			AudioSet	Transformer [10]	0.64	BLEU
	Image captioning T44 T45	Visual-impaired accessibility (Supersence)	MSCOCO'14	LSTM [7]	0.73*	BLEU
			Flickr8k	LSTM [110]	0.58	BLEU
	Text-to-image retrieval T46 T47	Image search (Google Photos)	Flickr8k	NAPReg [69]	0.39	Recall
			Flickr30k	CLIP [34]	0.69	Recall
	Audio/Text-to-image generation T48	Art creation (Verb Art)	VGGSound	Wav2clip [11]	99.89	FID
	Visual question answering T49 T50	Visual-impaired accessibility (Answerables)	VQA v2.0	MUTAN [41]	0.63	Accuracy
			VizWiz	MUTAN [41]	0.52	Accuracy

Tested on 50 mobile AI tasks

[1] Jinliang Yuan, et al. "Mobile Foundation Model as Firmware". In MobiCom'24.

Challenges of LLMaaS

- Opening new research opportunities and challenges
  - *Usability*: How to design LLMaaS interface? How to upgrade LLM?
    - *[MobiCom'24] Mobile Foundation Model as Firmware*
    - ***[NeurIPS'25] Efficient LoRA Adaptation Across Large Language Model Upgrades***
  - *Efficiency*: How to schedule, batch, and cache-reuse system-wise LLM requests? How to manage the LLM context states across apps?
    - *[MobiCom'25] Elastic On-Device LLM Service*
    - *[SenSys'26] LLM as a System Service on Mobile Devices*
  - *Security*: how to protect app-owned LoRa? How to isolate cross-app requests?
  - Etc..

LoRASuite: Towards LLM Upgradability

- Upgrading LLM without compromising LoRa (much)

Algorithm 1 CKA-based Layer Mapping

Input: CKA similarity matrix $S \in \mathbb{R}^{l_o \times l_n}$, Original and upgrade layer depth l_o and l_n.
Output: $path[i][j]$ with the highest total CKA.

```
/# Store the maximum sum
Initialize dp[i][j] ← −∞ for all i, j
/# Store path information
Initialize path[i][j] ← 0 for all i, j
/# Limit maximum offset
for j = 0 to |l_n − l_o| do
    dp[0][j] ← S[0][j]
end for

for i = 1 to l_o − 1 do
    /# Limit maximum offset
    for j = i to i + |l_n − l_o| do
        max_v ← −∞
        max_i ← −1
        for k = i − 1 to j − 1 do
            /# Transfer equation
            if dp[i−1][k] + S[i][j] > max_v then
                max_v ← dp[i − 1][k] + S[i][j]
                max_i ← k
            end if
        end for
    en
end f
```

CKA-based Layer Mapping

Architecture	Model	Vocab. Size	Hidden size	Interm. size	# of layers	# of heads	Attention
LlamaForCausalLM	Yi-6B	64k	4096	11008	32	32	GQA
	Yi-1.5-9B	64k	4096	11008	48	32	GQA
GPTNeoXForCausalLM	pythia-1b	50k	2048	8192	16	8	MHA
	pythia-1.4b	50k	2048	8192	24	16	MHA
BloomForCausalLM	bloom-560m	250k	1024	4096	24	16	MHA
	bloomz-1b1	250k	1536	6144	24	16	MHA
LlamaForCausalLM	Llama-2-7b	32k	4096	11008	32	32	MHA
	Llama-3-8b	128k	4096	14336	32	32	GQA
Qwen2ForCausalLM	Qwen-1.5-1.8B	152k	2048	5504	24	16	MHA
	Qwen-2.5-3B	152k	2048	11008	36	16	GQA
MiniCPMForCausalLM	MiniCPM-S-1B	73k	1536	3840	52	24	GQA
	MiniCPM-2B	123k	2304	5760	40	36	MHA

Base LLM upgrades at different dimensions

Base Model	PEFT	AddSub	MultiArith	SingleEq	GSM8K	AQuA	MAWPS	SVAMP	Avg.
MiniCPM-S-1B	-	23.80	5.17	23.62	7.43	17.72	20.17	16	16.27
	LoRA (10k)	29.62	54.67	31.5	12.13	8.27	31.93	18.2	26.62
MiniCPM-2B	-	23.29	48	27.56	12.28	14.57	22.27	19	23.85
	LoRASuite w/o LFT	23.04	47	28.15	12.36	14.17	22.27	20.7	23.96
	LoRA (100)	20.76	40.33	28.54	11.98	15.35	23.95	18.8	22.82
	LoRASuite w LFT (100)	**54.18**	75.5	**56.69**	**18.73**	14.17	**50.84**	**36.5**	**43.80**
	LoRA (10k)	47.59	**84.67**	47.44	17.97	**19.69**	47.48	31.9	42.39

Base Model	PEFT	BoolQ	PIQA	SIQA	HellaSwag	WinoG	ARC-c	ARC-e	OBQA	Avg.
MiniCPM-S-1B	-	51.9	33.68	13.46	12.09	39.78	9.56	12.16	8.8	22.68
	LoRA (10k)	58.17	27.48	53.74	20.63	59.67	27.39	35.23	40.6	40.36
MiniCPM-2B	-	53.55	22.8	39.66	14.5	52.33	20.14	27.19	31.6	32.72
	LoRASuite w/o LFT	53.64	22.69	40.17	14.61	52.09	20.56	26.77	31	32.69
	LoRA (100)	**63.18**	31.88	42.37	15.77	50.99	28.5	35.27	40.8	38.60
	LoRASuite w LFT (100)	61.28	**43.91**	48.21	**24**	51.07	**32.42**	**39.02**	38.8	**42.34**
	LoRA (10k)	62.35	26.71	**50.61**	21.17	**60.77**	27.73	34.64	**42.8**	40.85

LoRaSuite achieves comparative performance with only 1% finetune data

[1] Yanan Li, et al. “LoRASuite: Efficient LoRA Adaptation Across LargeLanguage Model Upgrades”. In NeurIPS’25.

Challenges of LLMaaS

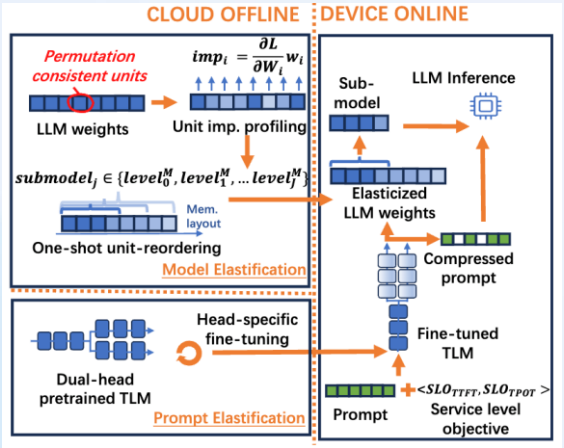
- Opening new research opportunities and challenges
  - *Usability*: How to design LLMaaS interface? How to upgrade LLM?
    - *[MobiCom'24] Mobile Foundation Model as Firmware*
    - *[NeurIPS'25] Efficient LoRA Adaptation Across Large Language Model Upgrades*
  - *Efficiency*: How to schedule, batch, and cache-reuse system-wise LLM requests? How to manage the LLM context states across apps?
    - ***[MobiCom'25] Elastic On-Device LLM Service***
    - *[SenSys'26] LLM as a System Service on Mobile Devices*
  - *Security*: how to protect app-owned LoRa? How to isolate cross-app requests?
  - Etc..

Serving LLM requests with different QoS

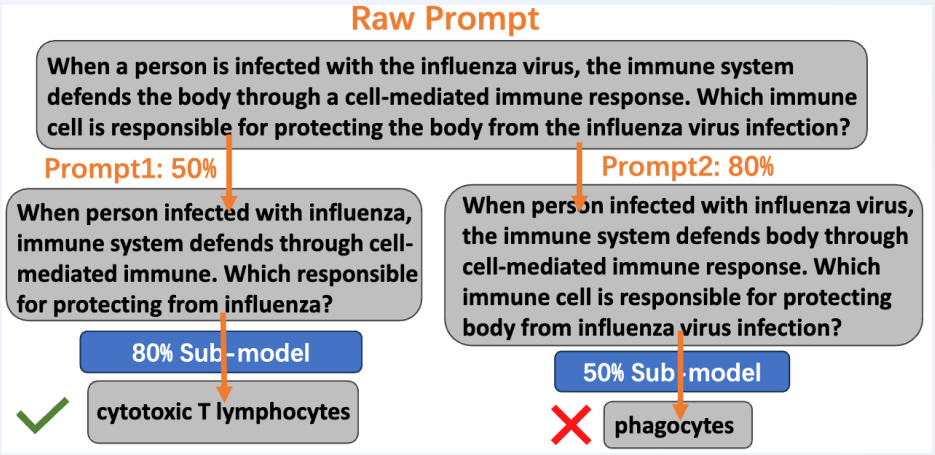
- Key idea: a joint planning of token/model pruning

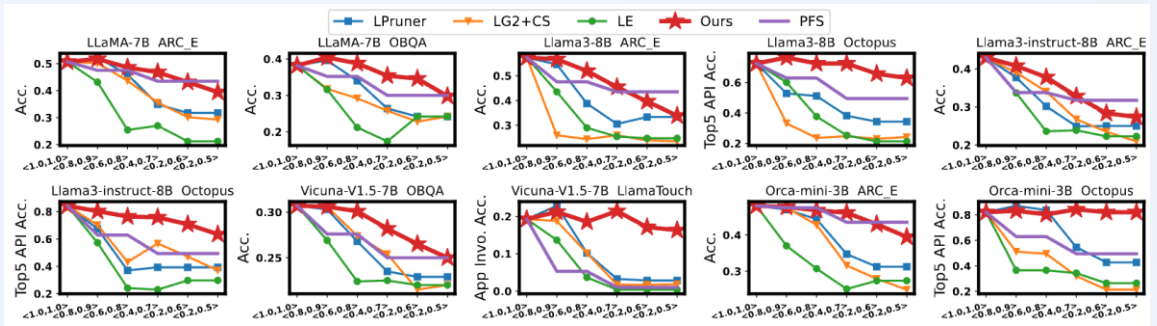
Mobile LLM App.	Service-Level Objective
Chatbot [9]	Readable TTFT/TPOT
Always-on Voice Assistant [6, 16]	Very-Low TTFT, medium TPOT
Background Screen-Event Recorder [15]	Tolerable TTFT/TPOT
Smart Message Reply [5]	Low TTFT, low TPOT
API-Calling Agent [23]	Low TTFT, acceptable TPOT
UI-Automation Agent [71, 84]	Low TTFT, acceptable TPOT

Different apps demand diversified QoS



A offline-guided, joint planning of token pruning and weights pruning





Significant improvement over static approaches

[1] Wangsong Yin, et al. "ELMS: Elasticized Large Language Models On Mobile Devices". In MobiCom'25.

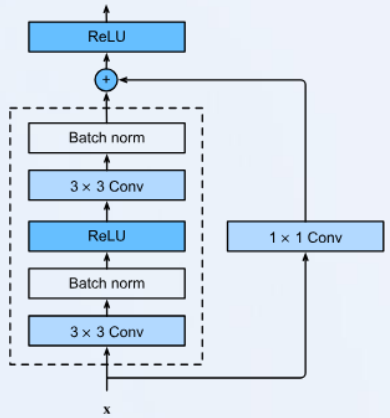
The changes LLM brings

	Pre-LLM Era		LLM Era
App:	fragmented tasks	→	a unified agent
OS:	model-agnostic	→	LLM-native
H/W:	CPU/GPU-centric	→	NPU-centric

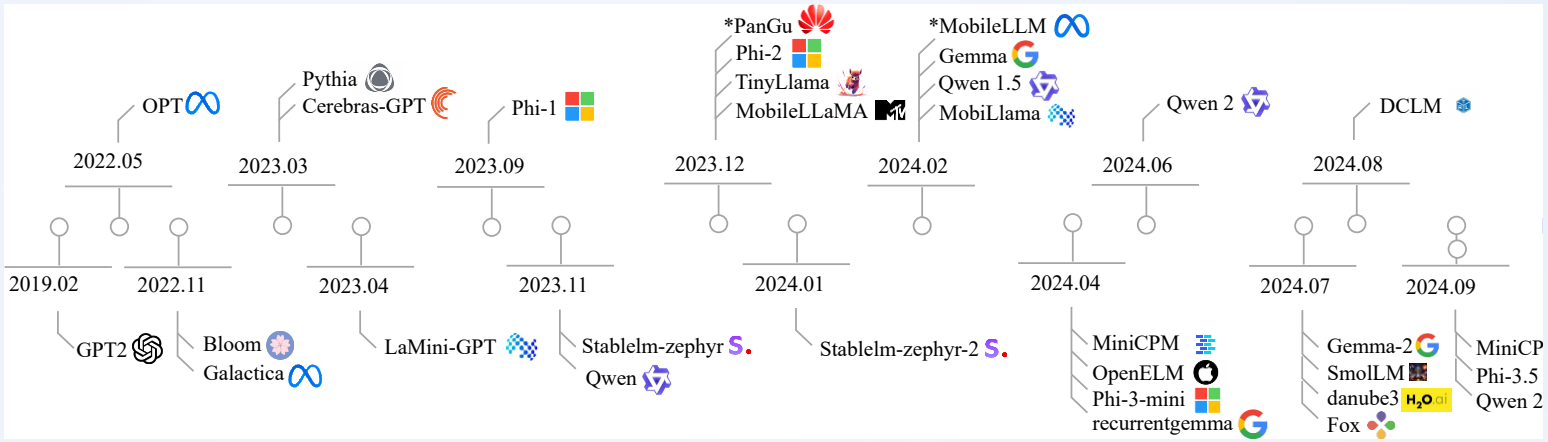
How to serve LLM requests with low latency and energy efficiency?

On-device LLM needs LLM-processor

- On-device **resource scarcity** further exacerbated.



vs.



ResNet, YoLo, LSTM, etc
(<200M)
<100ms to process one image
<100MB memory footprint
Easy to quantize (integer-only)
Static shape and cost

Small Language Models (1B~5B)

>10sec to process one prompt on CPU
>1GB memory footprint
Difficult to quantize (FP required)
Dynamic shape and increased cost with longer prompt

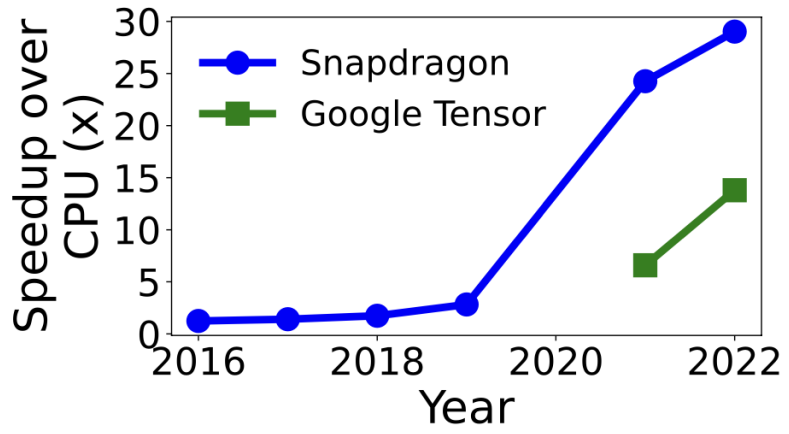
[1] Zhenyan Lu, et al. “Small Language Models: Survey, Measurements, and Insights”. In preprint’24.

On-device LLM needs NPU

- DSA (LLM-processor) is the answer to on-device LLM.

Vendor	Latest NPU	SDK	Open	Group	INT8 Perf.
Qualcomm	Hexagon NPU [15]	QNN [23]	×	×	73 TOPS
Google	Edge TPU [17]	Edge TPU API [7]	×	×	4 TOPS
MediaTek	MediaTek APU 790 [11]	NeuroPilot [13]	×	N/A	60 TOPS
Huawei	Ascend NPU [6]	HiAI [9]	×	×	16 TOPS

"Open": Open-source?; "Group": Support per-group quantization MatMul? "N/A": No available documents for public; "INT8 Perf.": Int8 performance.



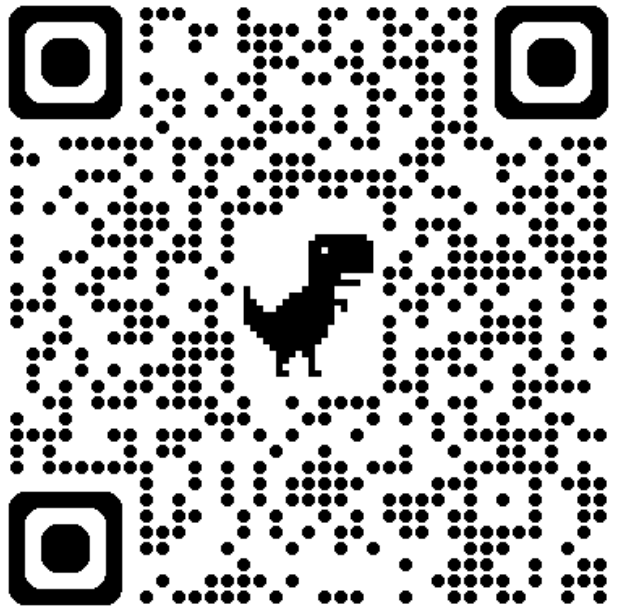
- *The gap between CPU/GPU and NPU increases over time*
  - *Moore's law still stands for NPU*
- *The gap of energy efficiency is even larger*

[1] Jinliang Yuan. "Mobile Foundation Model as Firmware". In MobiCom'24.

Filling the design gap between legacy NPUs and modern LLM inference

[ASPLOS'25] Fast On-device LLM Inference with NPUs

Code at https://github.com/UbiquitousLearning/mllm

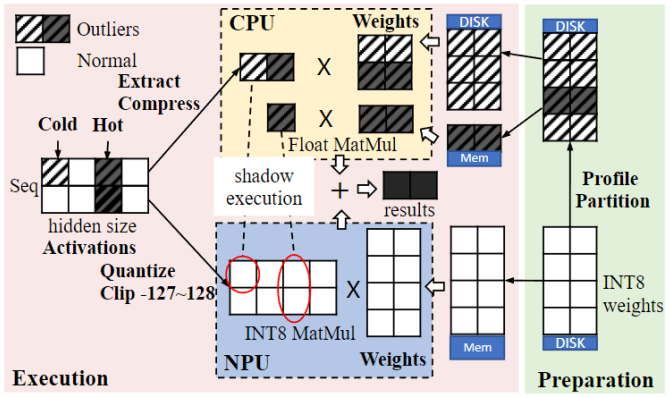


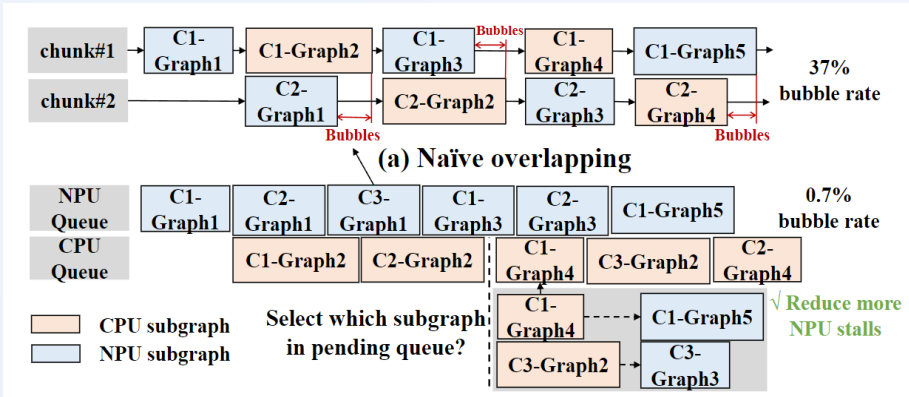
llm.npu: accelerating LLM prefilling with NPU

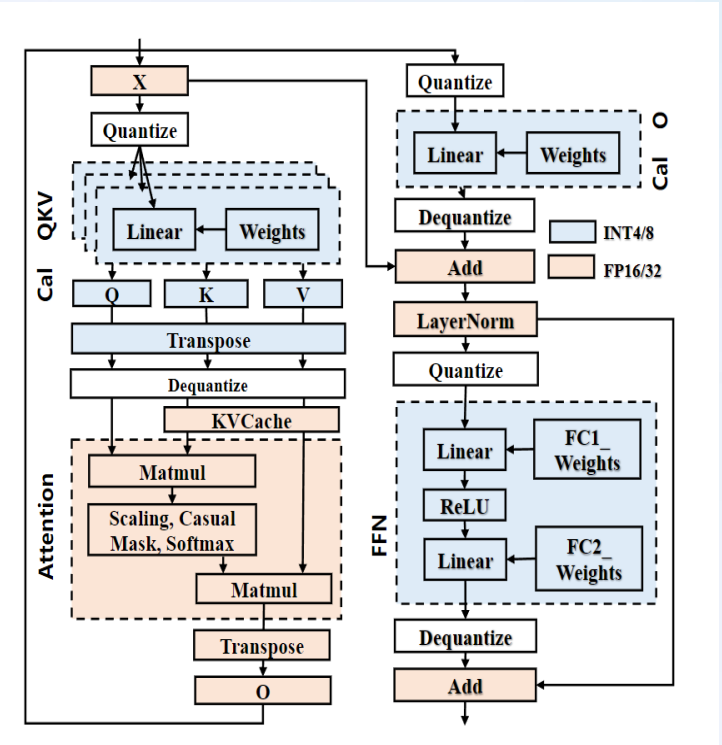
- Legacy mobile NPU has poor support for

  (1) Dynamic shape; (2) FP operations; (3) group-level quantization

- *llm.npu* proposes
  - Chunked prefill with partial sharing
  - Shadow outlier execution across CPU/NPU
  - Our-of-order scheduling among CPU/NPU



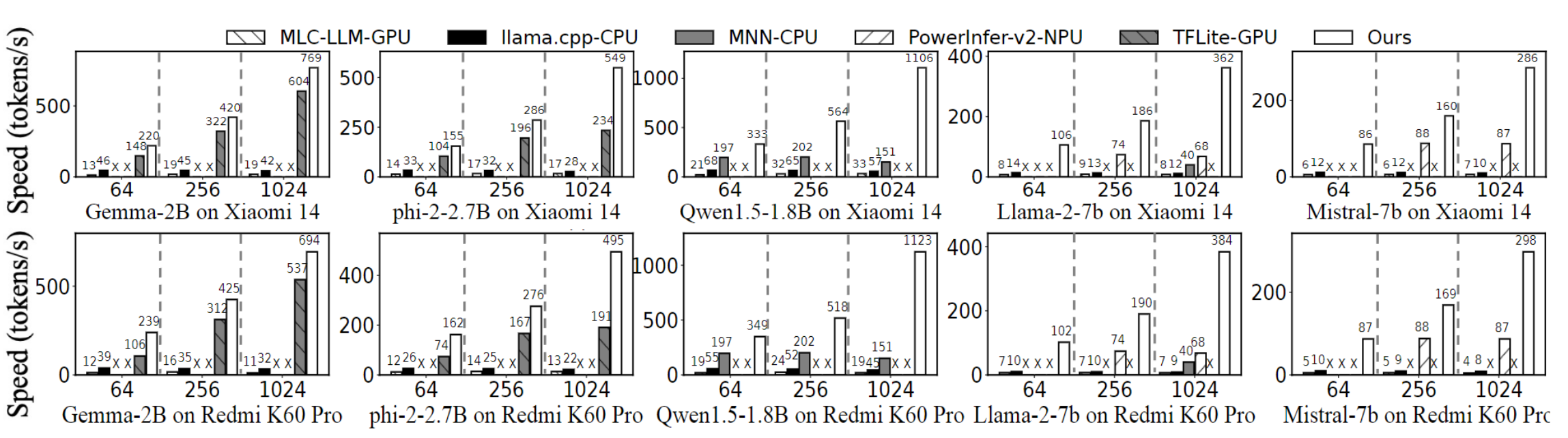




[1] Daliang Xu, et al. "Fast On-device LLM Inference with NPUs". In ASPLOS'25.

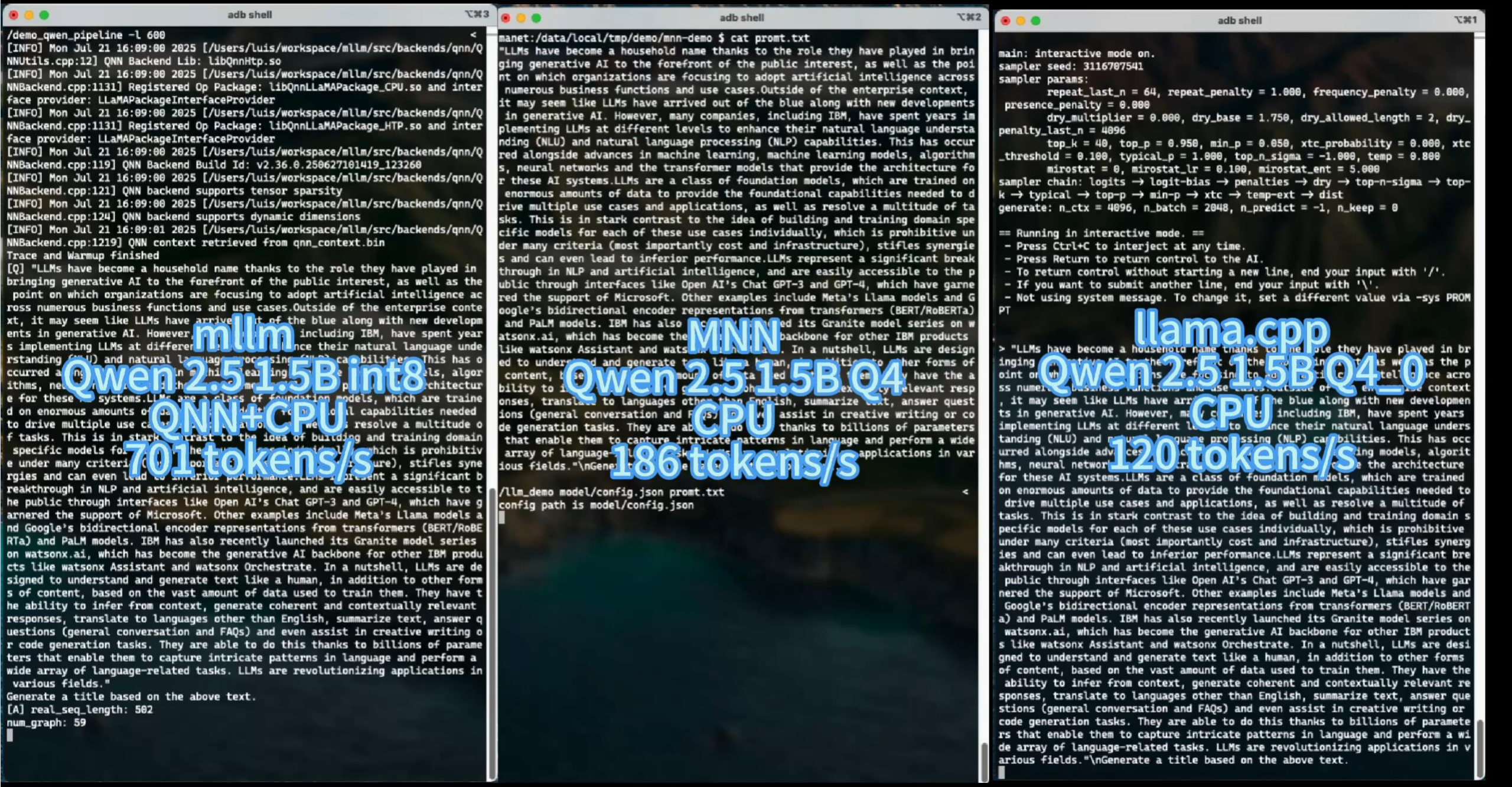
Highlighted results

Prefill speed under different prompt lengths on different devices (datasets: Longbench-2wiki-Multi-doc QA)
Baselines: MLC-LLM (GPU), llama.cpp (CPU), MNN (CPU), PowerInfer-v2 (NPU), TFLite (GPU)



7.3×–18.4×faster than baselines on CPU, and 1.3×–43.6× on GPU with prompt length of 1024
Achieves >1000 tokens/second on Qwen1.5-1.8B (for the first time)

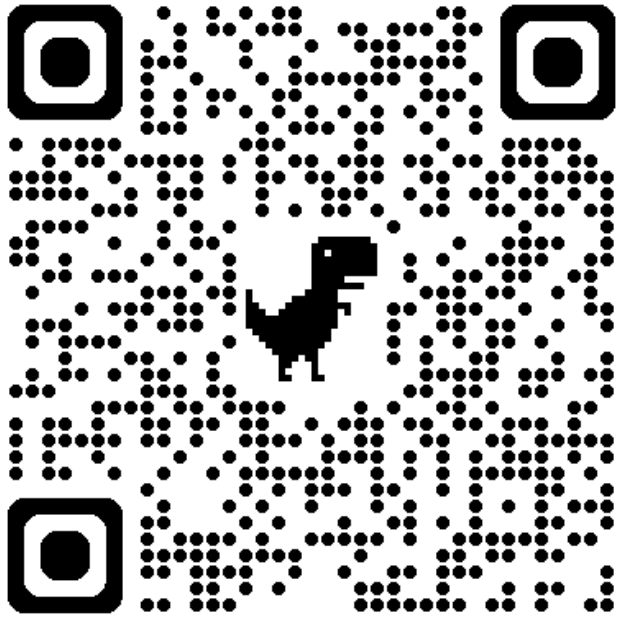
[1] Daliang Xu, et al. “Fast On-device LLM Inference with NPUs”. In ASPLOS’25.


adb shell
⌥⌘3
/demo_qwen_pipeline -l 600
[INFO] Mon Jul 21 16:09:00 2025 [/Users/luis/workspace/mllm/src/backends/qnn/QNNUtils.cpp:12] QNN Backend Lib: libQnnHtp.so
[INFO] Mon Jul 21 16:09:00 2025 [/Users/luis/workspace/mllm/src/backends/qnn/QNNBackend.cpp:1131] Registered Op Package: libQnnLLaMAPackage_CPU.so and interface provider: LLaMAPackageInterfaceProvider
[INFO] Mon Jul 21 16:09:00 2025 [/Users/luis/workspace/mllm/src/backends/qnn/QNNBackend.cpp:1131] Registered Op Package: libQnnLLaMAPackage_HTP.so and interface provider: LLaMAPackageInterfaceProvider
[INFO] Mon Jul 21 16:09:00 2025 [/Users/luis/workspace/mllm/src/backends/qnn/QNNBackend.cpp:119] QNN Backend Build Id: v2.36.0.250627101419_123260
[INFO] Mon Jul 21 16:09:00 2025 [/Users/luis/workspace/mllm/src/backends/qnn/QNNBackend.cpp:121] QNN backend supports tensor sparsity
[INFO] Mon Jul 21 16:09:00 2025 [/Users/luis/workspace/mllm/src/backends/qnn/QNNBackend.cpp:124] QNN backend supports dynamic dimensions
[INFO] Mon Jul 21 16:09:01 2025 [/Users/luis/workspace/mllm/src/backends/qnn/QNNBackend.cpp:1219] QNN context retrieved from qnn_context.bin
Trace and Warmup finished
[Q] "LLMs have become a household name thanks to the role they have played in bringing generative AI to the forefront of the public interest, as well as the point on which organizations are focusing to adopt artificial intelligence across numerous business functions and use cases.Outside of the enterprise context, it may seem like LLMs have arrive[illegible] of the blue along with new developments in generative AI. However, [illegible] including IBM, have spent years implementing LLMs at differen[illegible]nce their natural language understanding (NLU) and natural language [illegible]els, algorithms, neu[illegible] p[illegible]rchitecture for these [illegible] systems.LLMs are [illegible] class of foundation models, which are trained on enormous amounts o[illegible]l capabilities needed to drive multiple use ca[illegible] resolve a multitude of tasks. This is in stark [illegible]ntrast to the idea of building and training domain specific models for [illegible]which is prohibitive under many criteri[illegible]ure), stifles synergies and can even lead to inferior performance.LLMs represent a significant breakthrough in NLP and artificial intelligence, and are easily accessible to the public through interfaces like Open AI's Chat GPT-3 and GPT-4, which have garnered the support of Microsoft. Other examples include Meta's Llama models and Google's bidirectional encoder representations from transformers (BERT/RoBERTa) and PaLM models. IBM has also recently launched its Granite model series on watsonx.ai, which has become the generative AI backbone for other IBM products like watsonx Assistant and watsonx Orchestrate. In a nutshell, LLMs are designed to understand and generate text like a human, in addition to other forms of content, based on the vast amount of data used to train them. They have the ability to infer from context, generate coherent and contextually relevant responses, translate to languages other than English, summarize text, answer questions (general conversation and FAQs) and even assist in creative writing or code generation tasks. They are able to do this thanks to billions of parameters that enable them to capture intricate patterns in language and perform a wide array of language-related tasks. LLMs are revolutionizing applications in various fields."
Generate a title based on the above text.
[A] real_seq_length: 502
num_graph: 59
mllm
Qwen 2.5 1.5B int8
QNN+CPU
701 tokens/s
adb shell
⌥⌘2
manet:/data/local/tmp/demo/mnn-demo $ cat promt.txt
"LLMs have become a household name thanks to the role they have played in bringing generative AI to the forefront of the public interest, as well as the point on which organizations are focusing to adopt artificial intelligence across numerous business functions and use cases.Outside of the enterprise context, it may seem like LLMs have arrived out of the blue along with new developments in generative AI. However, many companies, including IBM, have spent years implementing LLMs at different levels to enhance their natural language understanding (NLU) and natural language processing (NLP) capabilities. This has occurred alongside advances in machine learning, machine learning models, algorithms, neural networks and the transformer models that provide the architecture for these AI systems.LLMs are a class of foundation models, which are trained on enormous amounts of data to provide the foundational capabilities needed to drive multiple use cases and applications, as well as resolve a multitude of tasks. This is in stark contrast to the idea of building and training domain specific models for each of these use cases individually, which is prohibitive under many criteria (most importantly cost and infrastructure), stifles synergies and can even lead to inferior performance.LLMs represent a significant breakthrough in NLP and artificial intelligence, and are easily accessible to the public through interfaces like Open AI's Chat GPT-3 and GPT-4, which have garnered the support of Microsoft. Other examples include Meta's Llama models and Google's bidirectional encoder representations from transformers (BERT/RoBERTa) and PaLM models. IBM has also recently launched its Granite model series on watsonx.ai, which has become the generative AI backbone for other IBM products like watsonx Assistant and watsonx Orchestrate. In a nutshell, LLMs are designed to understand and generate t[illegible] like a human, in addition to other forms of content, b[illegible] the vast amount of data used to train them. They have the ability to infer from context, generate coherent and contextually relevant responses, translate to languages other than English, summarize [illegible], answer questions (general conversation and FAQs) and even assist in creative writing or code generation tasks. They are able to do this thanks to billions of parameters that enable them to capture intricate patterns in language and perform a wide array of language-related tasks. LLMs are revolutionizing applications in various fields."\nGenerate a title based on the above text.
/llm_demo model/config.json promt.txt
config path is model/config.json
MNN
Qwen 2.5 1.5B Q4
CPU
186 tokens/s
adb shell
⌥⌘1
main: interactive mode on.
sampler seed: 3116707541
sampler params:
repeat_last_n = 64, repeat_penalty = 1.000, frequency_penalty = 0.000, presence_penalty = 0.000
dry_multiplier = 0.000, dry_base = 1.750, dry_allowed_length = 2, dry_penalty_last_n = 4096
top_k = 40, top_p = 0.950, min_p = 0.050, xtc_probability = 0.000, xtc_threshold = 0.100, typical_p = 1.000, top_n_sigma = -1.000, temp = 0.800
mirostat = 0, mirostat_lr = 0.100, mirostat_ent = 5.000
sampler chain: logits → logit-bias → penalties → dry → top-n-sigma → top-k → typical → top-p → min-p → xtc → temp-ext → dist
generate: n_ctx = 4096, n_batch = 2048, n_predict = -1, n_keep = 0
== Running in interactive mode. ==
- Press Ctrl+C to interject at any time.
- Press Return to return control to the AI.
- To return control without starting a new line, end your input with '/'.
- If you want to submit another line, end your input with '\'.
- Not using system message. To change it, set a different value via -sys PROMPT
> "LLMs have become a household name thanks to the role they have played in bringing generative AI to the forefront of the public interest, as well as the point on which organizations are focusing to adopt artificial intelligence across numerous business functions and use cases.Outside of the enterprise context, it may seem like LLMs have arrived out of the blue along with new developments in generative AI. However, many companies, including IBM, have spent years implementing LLMs at different levels to enhance their natural language understanding (NLU) and natural language processing (NLP) capabilities. This has occurred alongside advances in machine learning, machine learning models, algorithms, neural networks and the transformer models that provide the architecture for these AI systems.LLMs are a class of foundation models, which are trained on enormous amounts of data to provide the foundational capabilities needed to drive multiple use cases and applications, as well as resolve a multitude of tasks. This is in stark contrast to the idea of building and training domain specific models for each of these use cases individually, which is prohibitive under many criteria (most importantly cost and infrastructure), stifles synergies and can even lead to inferior performance.LLMs represent a significant breakthrough in NLP and artificial intelligence, and are easily accessible to the public through interfaces like Open AI's Chat GPT-3 and GPT-4, which have garnered the support of Microsoft. Other examples include Meta's Llama models and Google's bidirectional encoder representations from transformers (BERT/RoBERTa) and PaLM models. IBM has also recently launched its Granite model series on watsonx.ai, which has become the generative AI backbone for other IBM products like watsonx Assistant and watsonx Orchestrate. In a nutshell, LLMs are designed to understand and generate text like a human, in addition to other forms of content, based on the vast amount of data used to train them. They have the ability to infer from context, generate coherent and contextually relevant responses, translate to languages other than English, summarize text, answer questions (general conversation and FAQs) and even assist in creative writing or code generation tasks. They are able to do this thanks to billions of parameters that enable them to capture intricate patterns in language and perform a wide array of language-related tasks. LLMs are revolutionizing applications in various fields."\nGenerate a title based on the above text.
llama.cpp
Qwen 2.5 1.5B Q4_0
CPU
120 tokens/s

Filling the design gap between legacy NPUs and modern LLM training

[USENIX ATC'24] FwdLLM: Efficient Federated Finetuning of Large Language Models with Perturbed Inferences

Code at https://github.com/UbiquitousLearning/FwdLLM



FwdLLM: BP-free LLM finetuning

- Key idea: leveraging forward gradient for LLM finetuning

$$\boldsymbol{g(\theta)} = \boldsymbol{(\nabla f(\theta) \cdot v)\, v}$$

A random, independent perturbation with same size as trainable weights θ

Forward gradient, an unbiased estimator of $f(\theta)$'s gradient

The directional derivative of f at point θ in direction v. Computing it takes only forward, no need for backpropagation

Compared to BP approach:

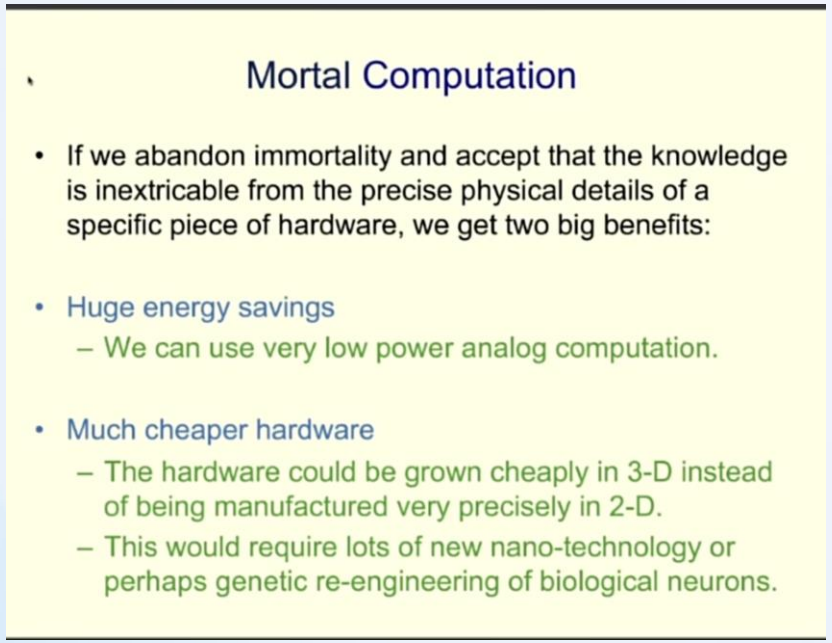
- Legacy NPU-compatible
- More memory efficient

- Further optimizations:
  - var.-controlled perturbation pacing
  - discriminative perturbation sampling
- Highlighted results: federated Llama-7B finetuning on devices, with significant speedup and memory saving

Methods	Mem. (GB)	Centralized Training (A100)			Federated Learning		
		Acc.	Round	Time	Acc.	Round	Time
BP, FP16	39.2	89.7	500	0.1 hrs	N/A due to memory inefficiency on Pixel 7 Pro (8GB)		
BP, INT8	32.4	88.6	500	0.06 hrs			
BP, INT4	28.5	87.8	500	0.04 hrs			
Ours, FP16	15.6	87.0	240	1.5 hrs			
Ours, INT8	7.9	86.9	260	0.8 hrs			
Ours (CPU), INT4	4.0	85.8	130	0.25 hrs	85.8	130	0.19 hrs
Ours (NPU*), INT4							0.07 hrs

[1] Mengwei Xu, et al. "FwdLLM: Efficient Federated Finetuning of Large Language Models with Perturbed Inferences". In ATC'24.

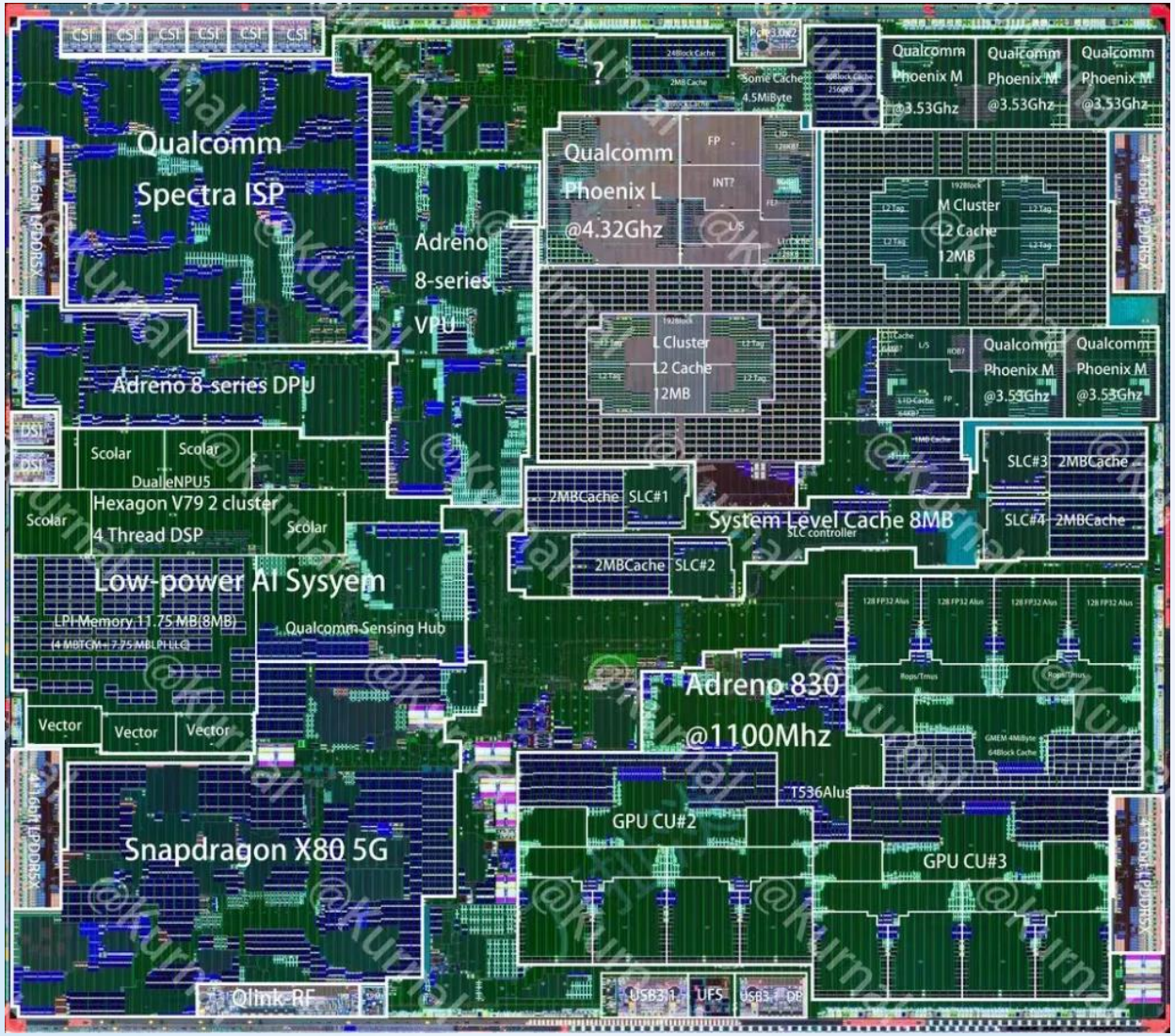
Looking into the future..



We shall probably look for hardware-software co-evolution, e.g., mortal computation by Geoffrey Hinton

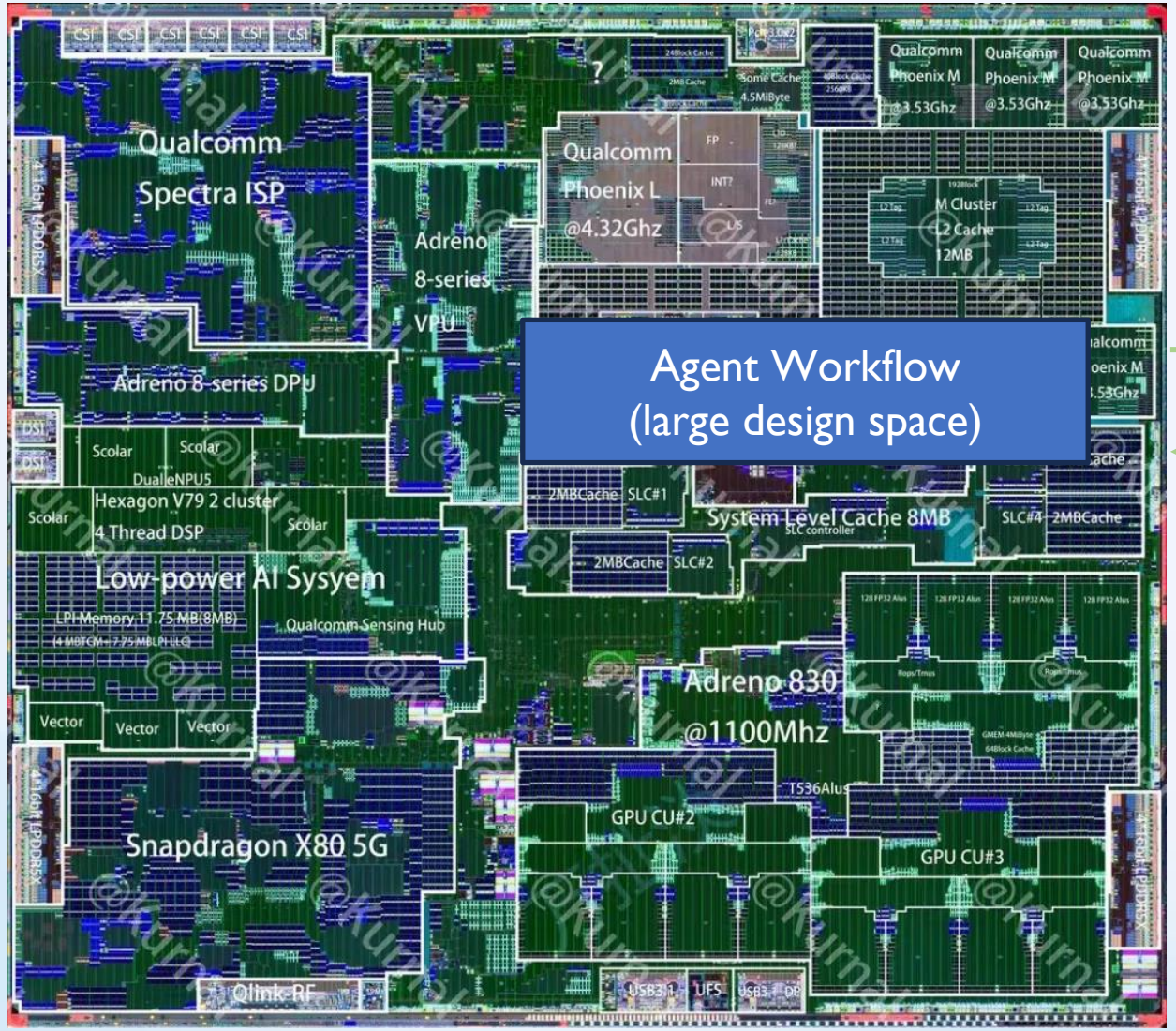
"Two paths to Intelligence" by Geoffrey Hinton

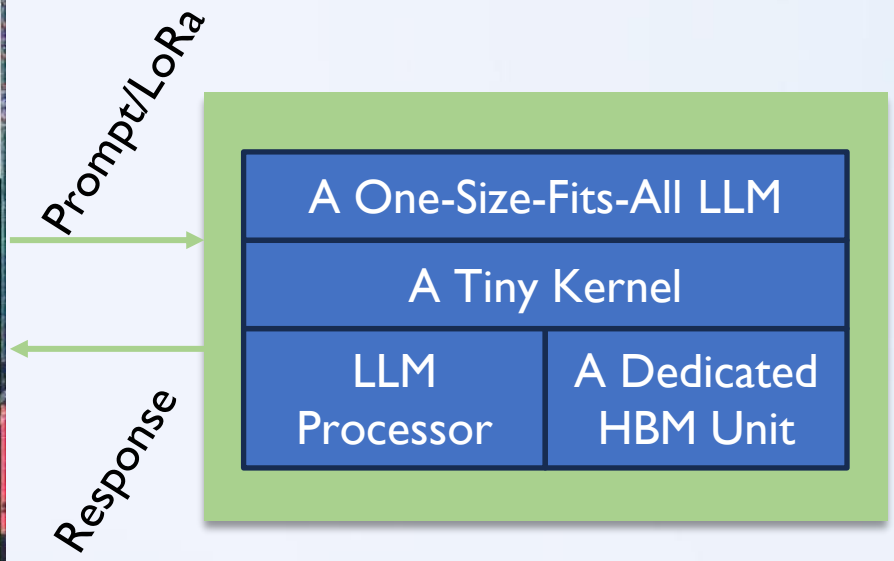
The Future: full-stack design!



https://innogyan.in/2024/10/28/die-shot-of-snapdragon-8-elite-reveals-component-space-allocation/

The Future: full-stack design!





Time to sacrifice flexibility for efficiency!
(we still have flexibility at agent workflow)

https://innogyan.in/2024/10/28/die-shot-of-snapdragon-8-elite-reveals-component-space-allocation/

Takeaways

- On-device LLM is reinventing the mobile devices
  - A total paradigm shift of mobile AI ecosystem

- It calls for full-stack LLM research
  - OS, runtime, model, and application (agent)

InfoQ 极客传媒

北京

QCon 全球软件开发大会

会议时间：4月10-12日

- 大模型赋能 AIOps
- 越挫越勇的大前端
- 云上业务架构演进
- 多模态大模型及应用
- 海外 AI 应用创新实践

北京

AiCon 全球人工智能开发与应用大会

会议时间：6月27-28日

- 端侧智能
- AI Agent
- 多模态大模型
- 金融+大模型

上海

QCon 全球软件开发大会

会议时间：10月23-25日

- AI Agent
- 大模型训练推理
- 端侧 AI
- 搜推深度融合
- 数智金融

4月 — 5月 — 6月 — 8月 — 10月 — 12月

上海

AiCon 全球人工智能开发与应用大会

会议时间：5月23-24日

- 通用大模型
- AI Agent
- 垂直领域应用
- 模型可解释性

深圳

AiCon 全球人工智能开发与应用大会

会议时间：8月22-23日

- 模型效率与部署
- 多模态大模型
- 大模型安全
- 智能硬件

北京

AiCon 全球人工智能开发与应用大会

会议时间：12月19-20日

- 通用大模型
- 智能硬件
- LMOps
- 具身智能

极客邦科技 2025 年会议规划

促进软件开发及相关领域知识与创新的传播

参会咨询

查看会议

QCon
全球软件开发大会

InfoQ 极客传媒

THANKS