The Era of Reasoning-Driven AI Models: Challenges and Opportunities in AI Computing

Dongsoo Lee, EVP, PhD
AI Computing Solution @ NAVER Cloud

NAVER Cloud

Dongsoo Lee

Academical Background

- **BS/MS at KAIST**
- **PhD at Purdue Univ.** (Emerging memory device, VLSI Testing, memory architecture)

Career

- **Intern at Intel Corporation** (Low-power GPU) **Qualcomm** (SRAM Design)
- **IBM TJ Watson Research Center**, Research Staff Member (2013-2017)

  IBM Power9, Power 10 server CPU chip design

  Deep learning accelerator (low-precision, distributed-learning, etc.)
- **Samsung Research**, Principal Engineer (2017-2021)

  Model Compression Lead (compression algorithm, kernel design, etc.)

- **NAVER Cloud**, Executive Vice President (2021-)

  Responsible for Efficient Serving Systems of HyperCLOVA (LLM) and AI chip strategy

  Director of NAVER-Intel Co Lab

The Paradigm Shift of Generative AI

LLM Phase 1: Era of Knowledge AI

Base LLM Model: AI that writes well

Pre-training + Post-training

GPT4o, Claude 3.5, Gemini, DeepSeek v3, Qwen2.5-Max

HyperCLOVA X, EXAONE 3.5

RL

LLM Phase 2: Era of Thinking AI

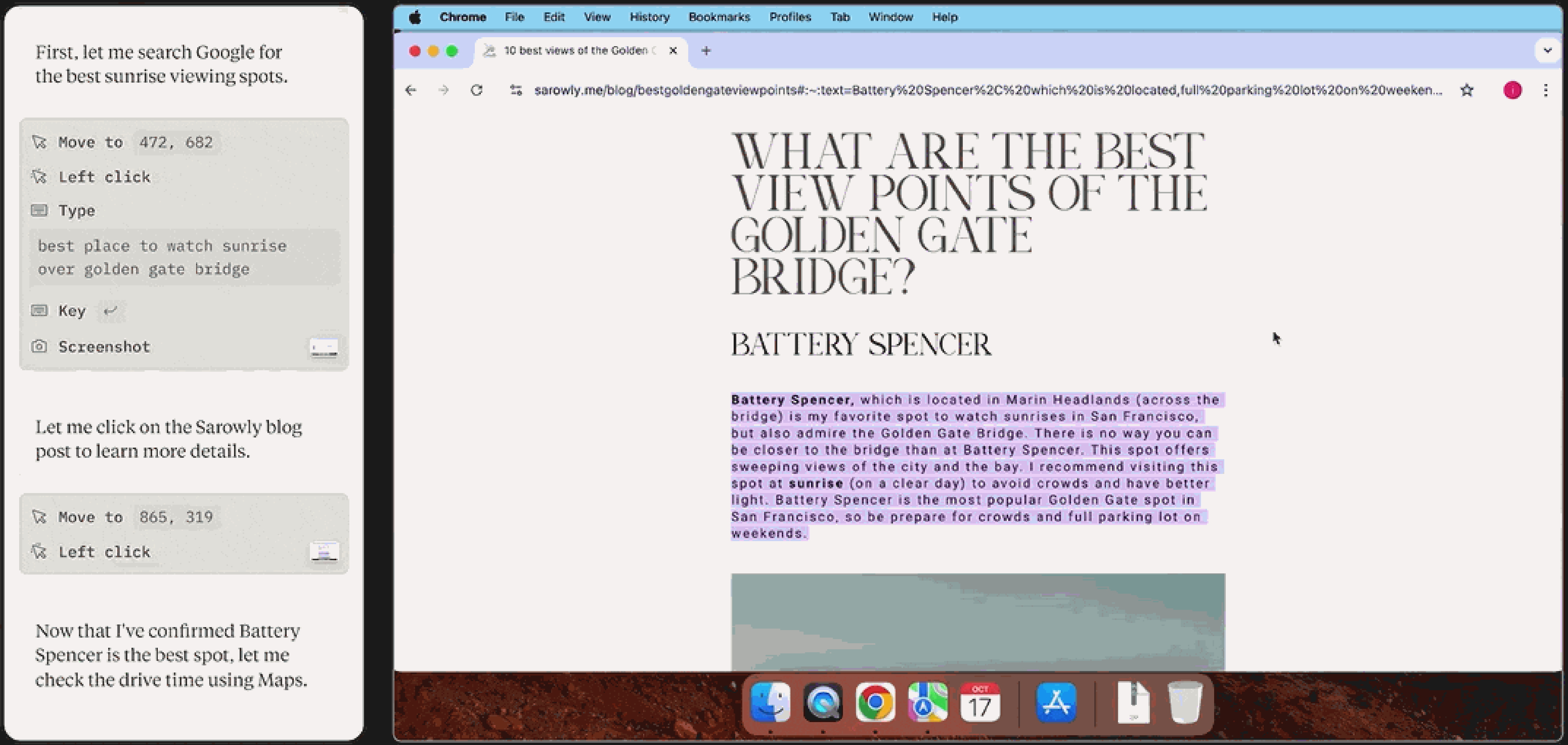
Reasoning-Enhanced AI Model: AI with advanced logical and mathematical reasoning capabilities

RL (w/ very long CoT Data)

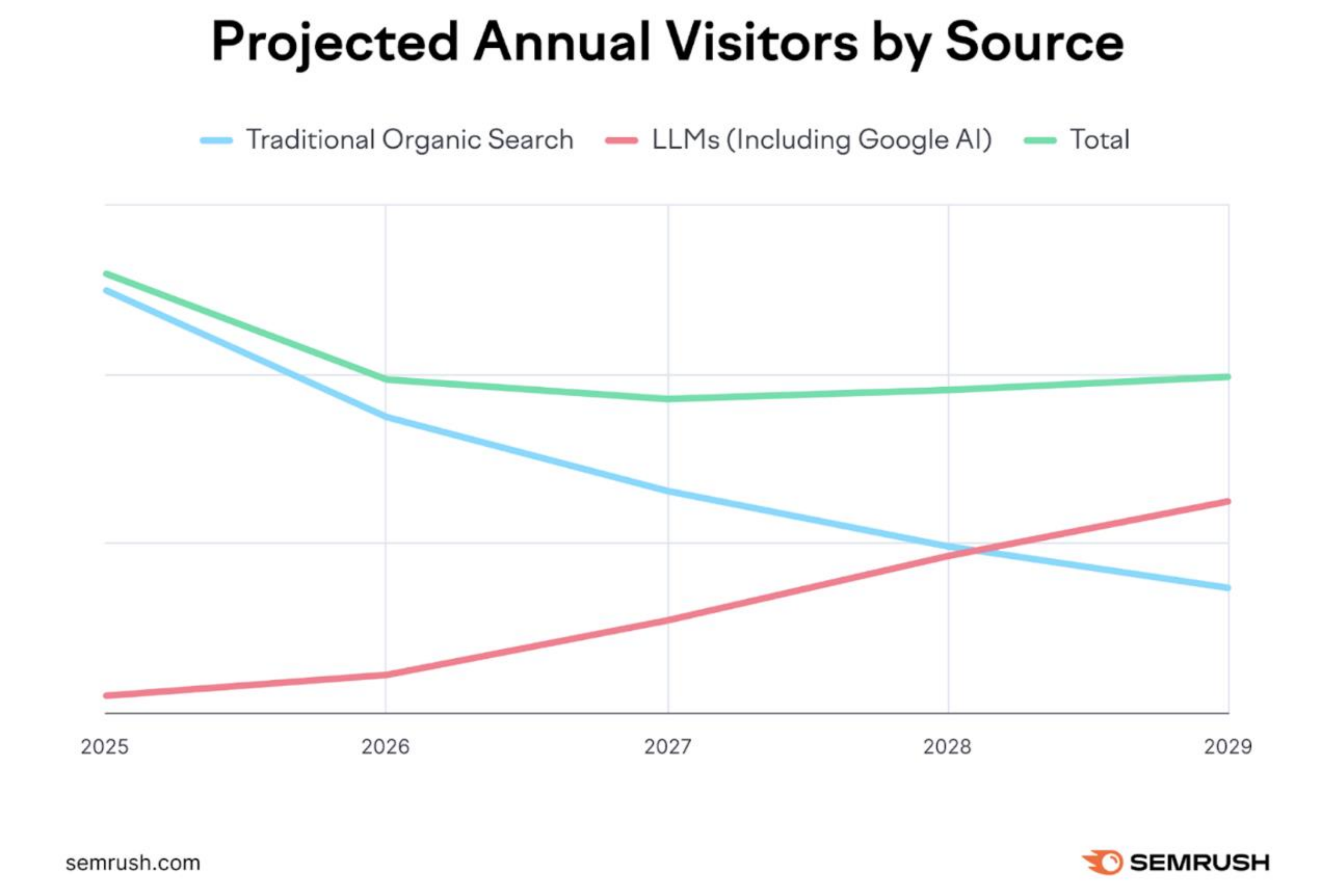
A strong base LLM model is essential

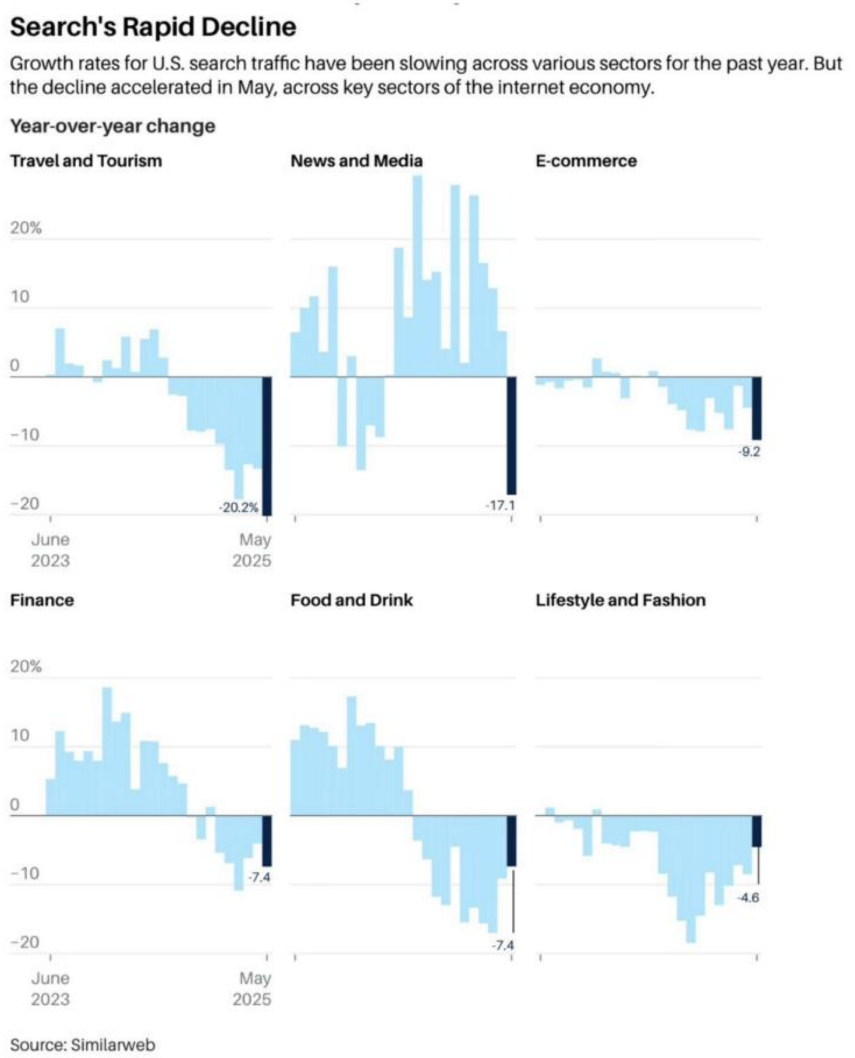
o1, o3, Deepseek-R1

Anthropic Claude 3.5 Computer Use!!



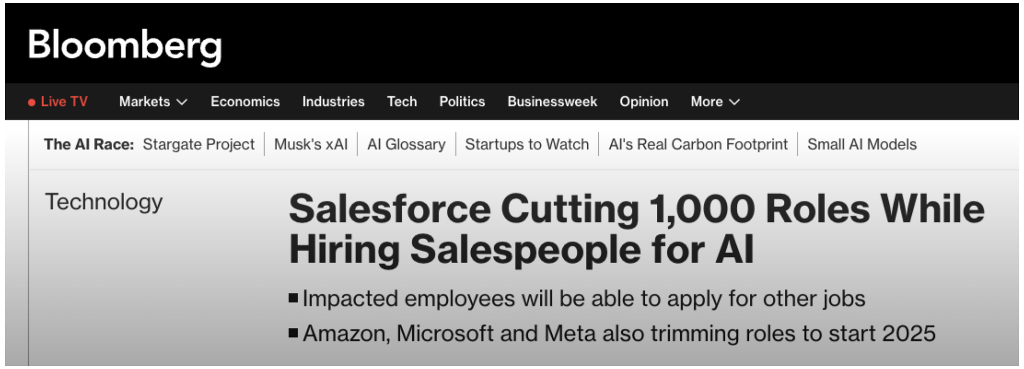
The Transformation of Search





NAVER Cloud

AI Transformation in which Agents Replace Human Labor


Bloomberg

● Live TV Markets Economics Industries Tech Politics Businessweek Opinion More

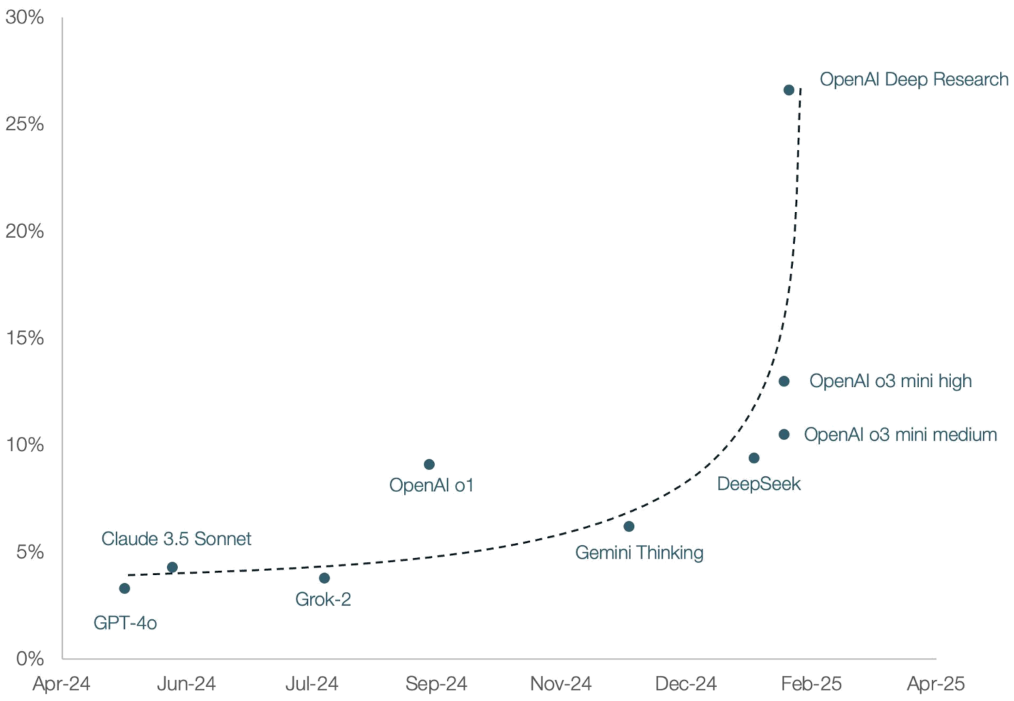
The AI Race: Stargate Project | Musk's xAI | AI Glossary | Startups to Watch | AI's Real Carbon Footprint | Small AI Models

Technology

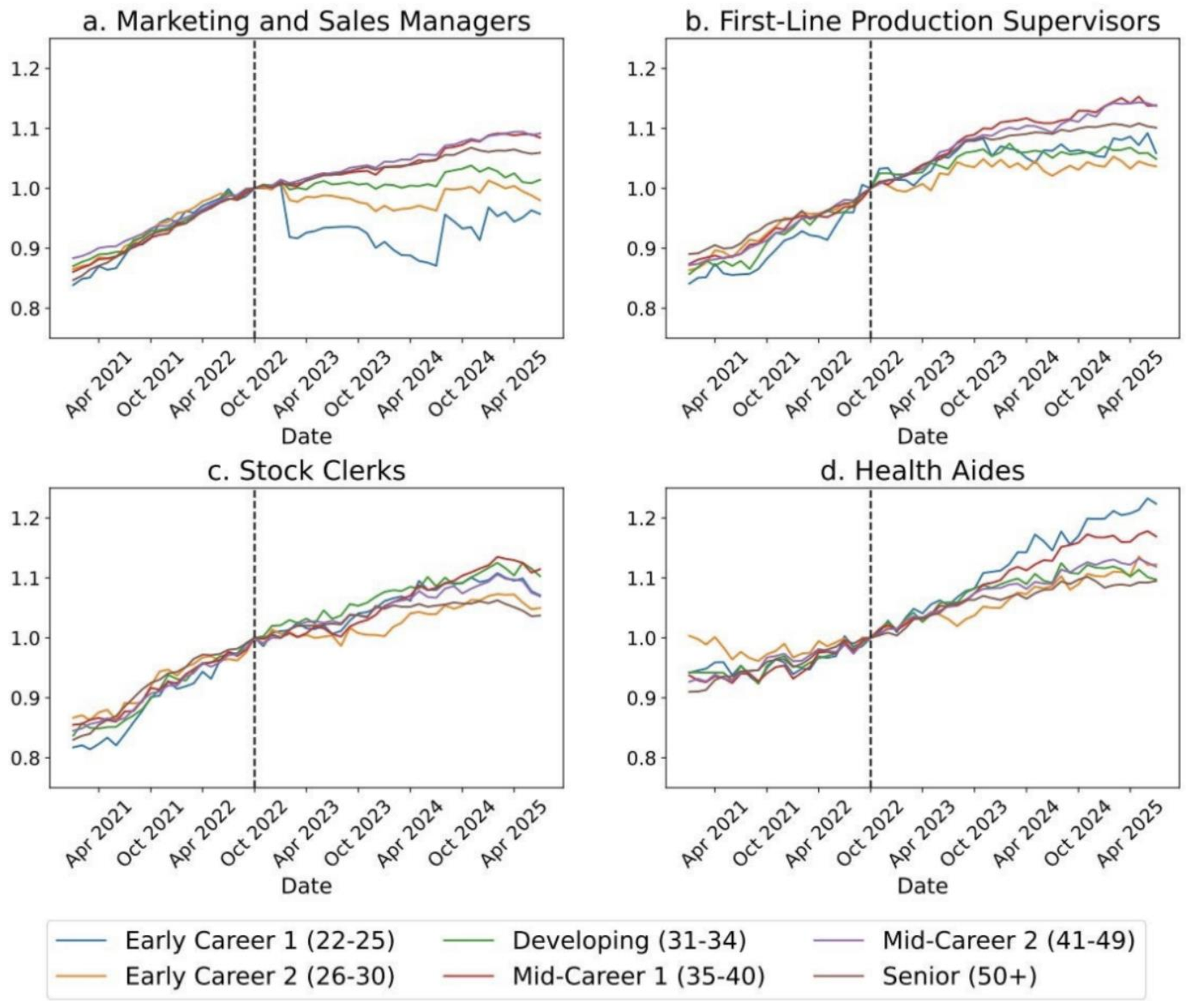
Salesforce Cutting 1,000 Roles While Hiring Salespeople for AI

- Impacted employees will be able to apply for other jobs
- Amazon, Microsoft and Meta also trimming roles to start 2025

AI Scores on *Humanity's Last Exam*



Source: Tomas Pueyo for Uncharted Territories, with data from Dan Hendrycks, of Humanity's Last Exam



$\sqrt{x}$ Mathematics

Question:

The set of natural transformations between two functors $F, G : C \to D$ can be expressed as the end

$$Nat(F,G) \cong \int_A Hom_D(F(A), G(A)).$$

Define set of natural cotransformations from F to G to be the coend

$$CoNat(F,G) \cong \int^A Hom_D(F(A), G(A)).$$

Let:

- $F = B_\bullet(\Sigma_4)_{*/}$ be the under ∞-category of the nerve of the delooping of the symmetric group Σ_4 on 4 letters under the unique 0-simplex $*$ of $B_\bullet \Sigma_4$.
- $G = B_\bullet(\Sigma_7)_{*/}$ be the under ∞-category nerve of the delooping of the symmetric group Σ_7 on 7 letters under the unique 0-simplex $*$ of $B_\bullet \Sigma_7$.

How many natural cotransformations are there between F and G?

Emily S
University of São Paulo

Computer Science

Question:

Let G be a graph. An edge-indicator of G is a function $a : \{0,1\} \to V(G)$ such that $\{a(0), a(1)\} \in E(G)$.

Consider the following Markov Chain $M = M(G)$:
The statespace of M is the set of all edge-indicators of G, and the transitions are defined as follows:

Assume $M_t = a$.
1. pick $b \in \{0,1\}$ u.a.r.
2. pick $v \in N(a(1-b))$ u.a.r. (here $N(v)$ denotes the open neighbourhood of v)
3. set $a'(b) = v$ and $a'(1-b) = a(1-b)$
4. Set $M_{t+1} = a'$

We call a class of graphs $\mathcal{G}$ well-behaved if, for each $G \in \mathcal{G}$ the Markov chain $M(G)$ converges to a unique stationary distribution, and the unique stationary distribution is the uniform distribution.

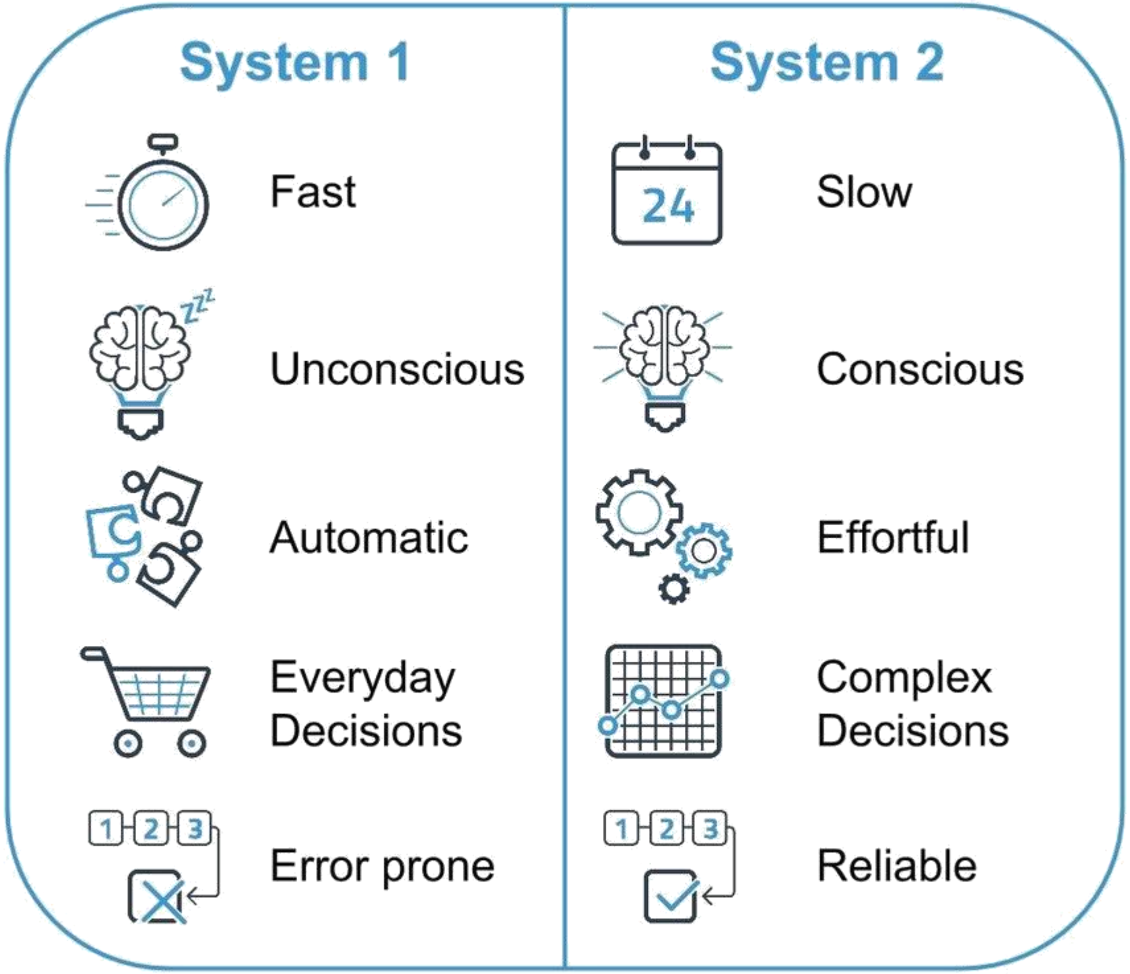
Which of the following graph classes is well-behaved?

Answer Choices:

A. The class of all non-bipartite regular graphs
B. The class of all connected cubic graphs
C. The class of all connected graphs
D. The class of all connected non-bipartite graphs
E. The class of all connected bipartite graphs.

Marc R
Queen Mary University of London

From System 1 Thinking To System 2 Thinking
: AI that Thinks Slowly and Deeply → Reasoning



- How to Elicit System 2 AI:

  : Let the LLM Talk to Itself and Scribble Notes

If a train is moving at 60 mph and travels for 3 hours, how far does it go?

The train travels 180 miles.

Plain response

To determine the distance traveled, use the formula:

Distance = Speed × Time

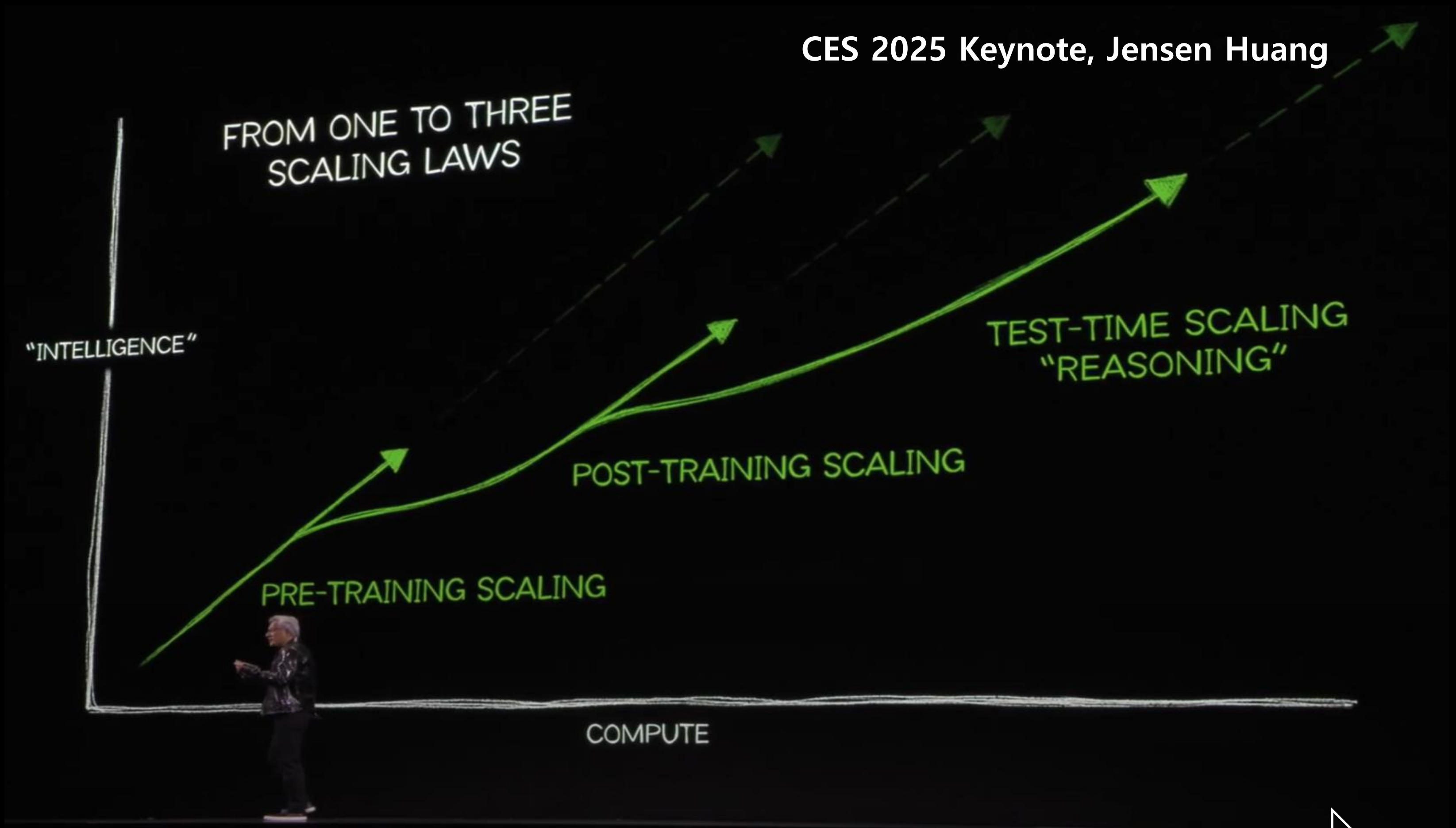
Given that the speed is 60 mph and the time is 3 hours:

Distance = 60 mph × 3 hours = 180 miles

So, the train travels 180 miles.

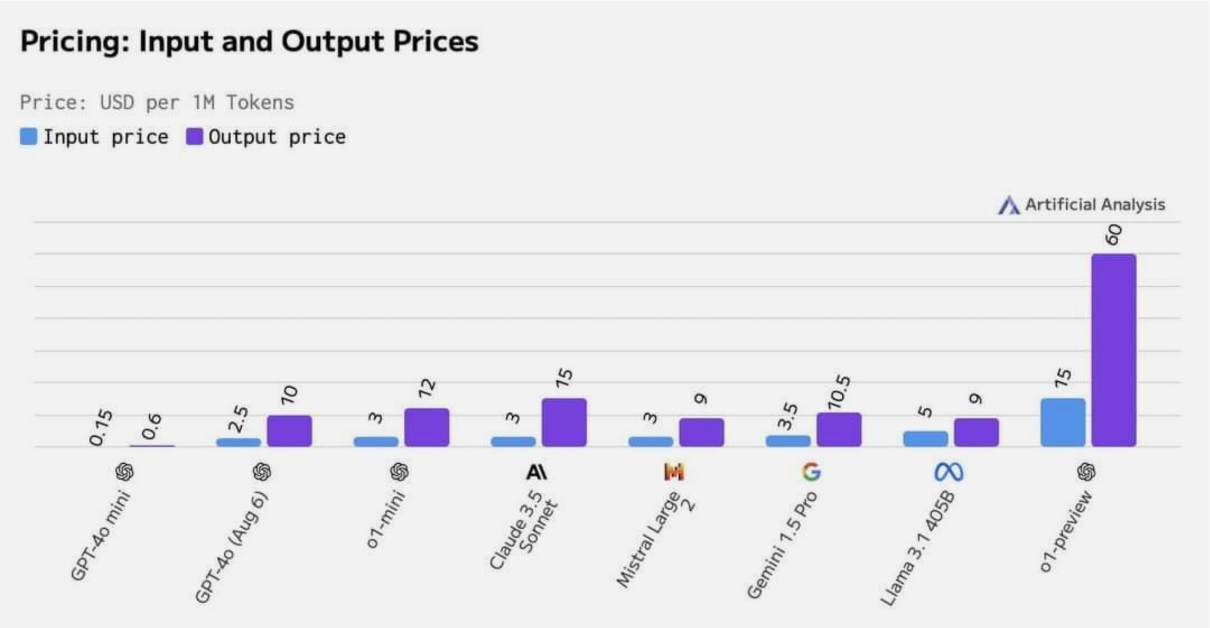
Response with intermediate reasoning steps

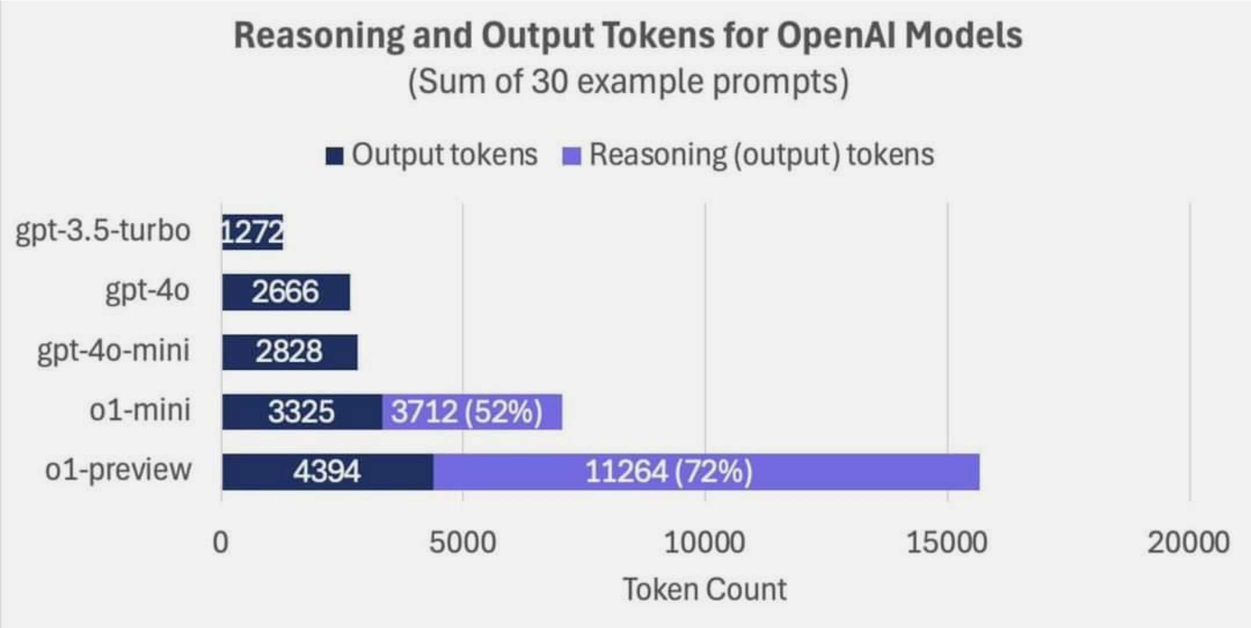
The Era of Test-time Scaling

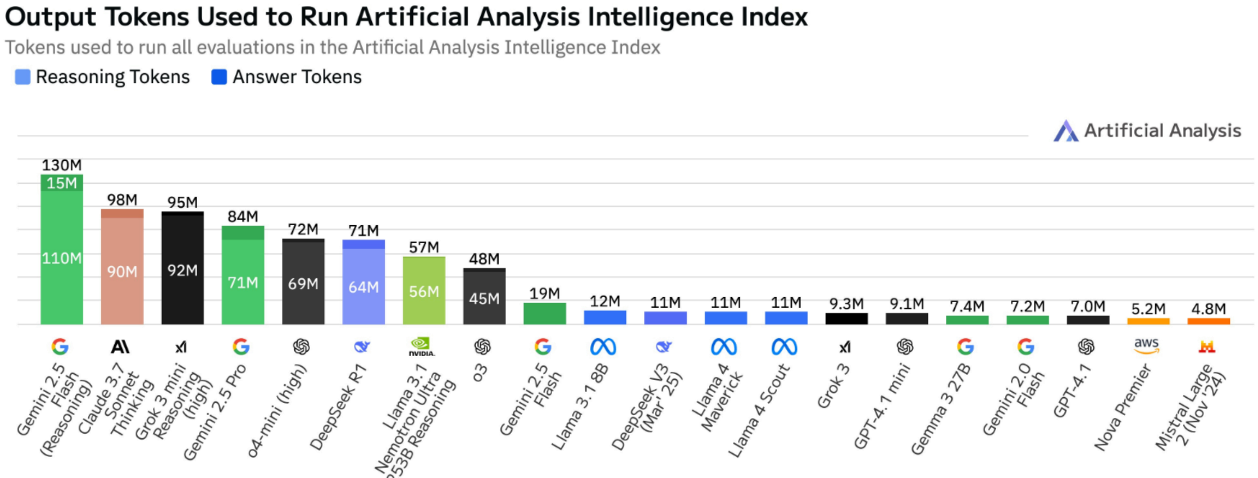


GPT-3 → ChatGPT → DeepResearch

The Main Computation Shifts to the Inference Phase







Rapid Growth of AI Services

Agentic AI

×

Growth of Model Size

×

Generating More Tokens (Reasoning)

=

Rising Inference Cost

NAVER Cloud

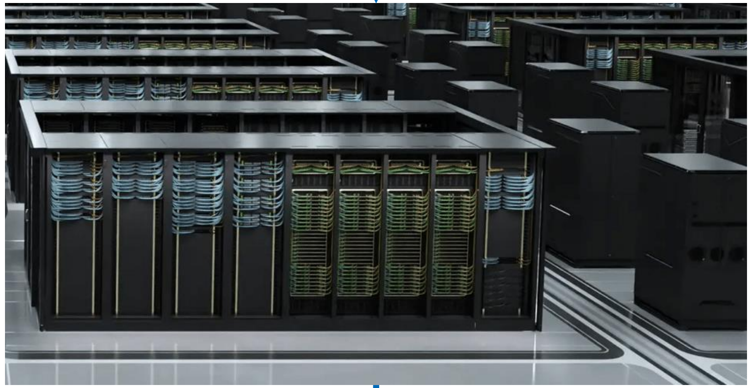
AI Factory that uses tokens as input and output are at the heart of today’s AI services.

A token is a unit of language that machines can understand.
It is generally larger than a character but smaller than a word.

[48392, 10752, 62430, 18904, 3285, 51267, 30112, 57643, 22346, 945, 6181, 43720, 28701, 15480, 36053, 49200, 27355, 6446, 18209, 39271, ...]

They're AI factories
because they have one job and one job only
— generating these incredible tokens
(Jensen Huang, GTC 2025)

Tokens

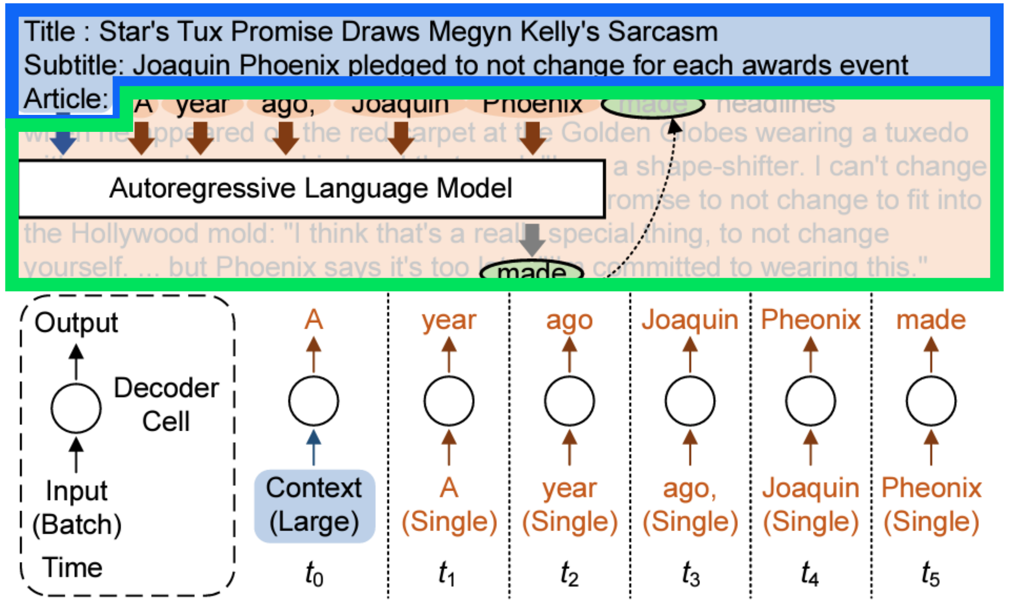


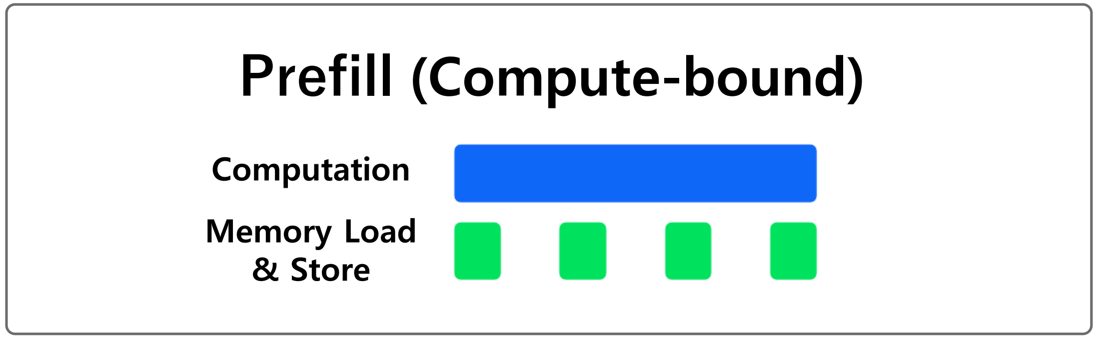
彼らは **AIファクトリー** です。
なぜなら、彼らの仕事はたった一つ、
つまり **この驚異的なトークンを生成することだけ** だからです。
（ジェンセン・フアン、GTC 2025 ）

Tokens

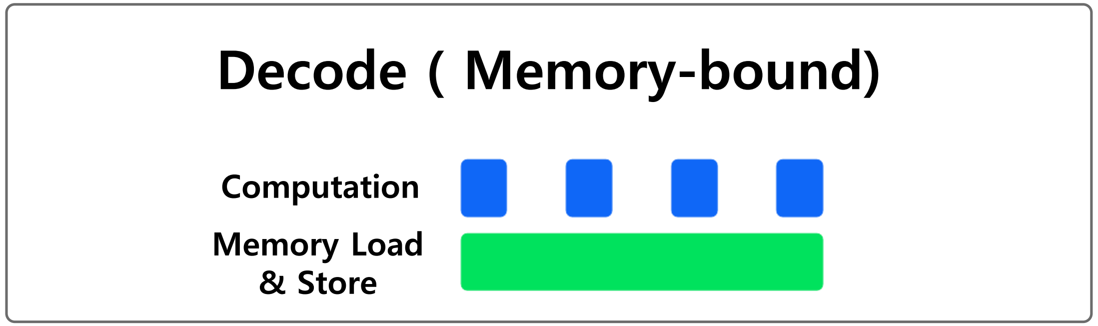
[32109, 54787, 8721, 1512, 63894, 41855, 27460, 5904, 33210, 20834, 46023, 12573, 60183, 3108, 55330, 17847, 39702, 48689, 7609, 22416, ...]

"Decode," which is limited by memory constraints, accounts for the majority of inference costs. Optimizing this part is the key to breakthrough cost reduction.





- This is the stage where most of the computation is concentrated.
- Memory read/write is relatively low, and compute resource utilization is high.
- Parallel processing is efficient at this stage.



- This stage requires repeated processing for each generated token.
- Memory read/write becomes the bottleneck rather than computation.
- As a result, it is more challenging to optimize for both performance and cost.

Service Cost of DeepSeek R1 (ref: Together.AI)

Llama3

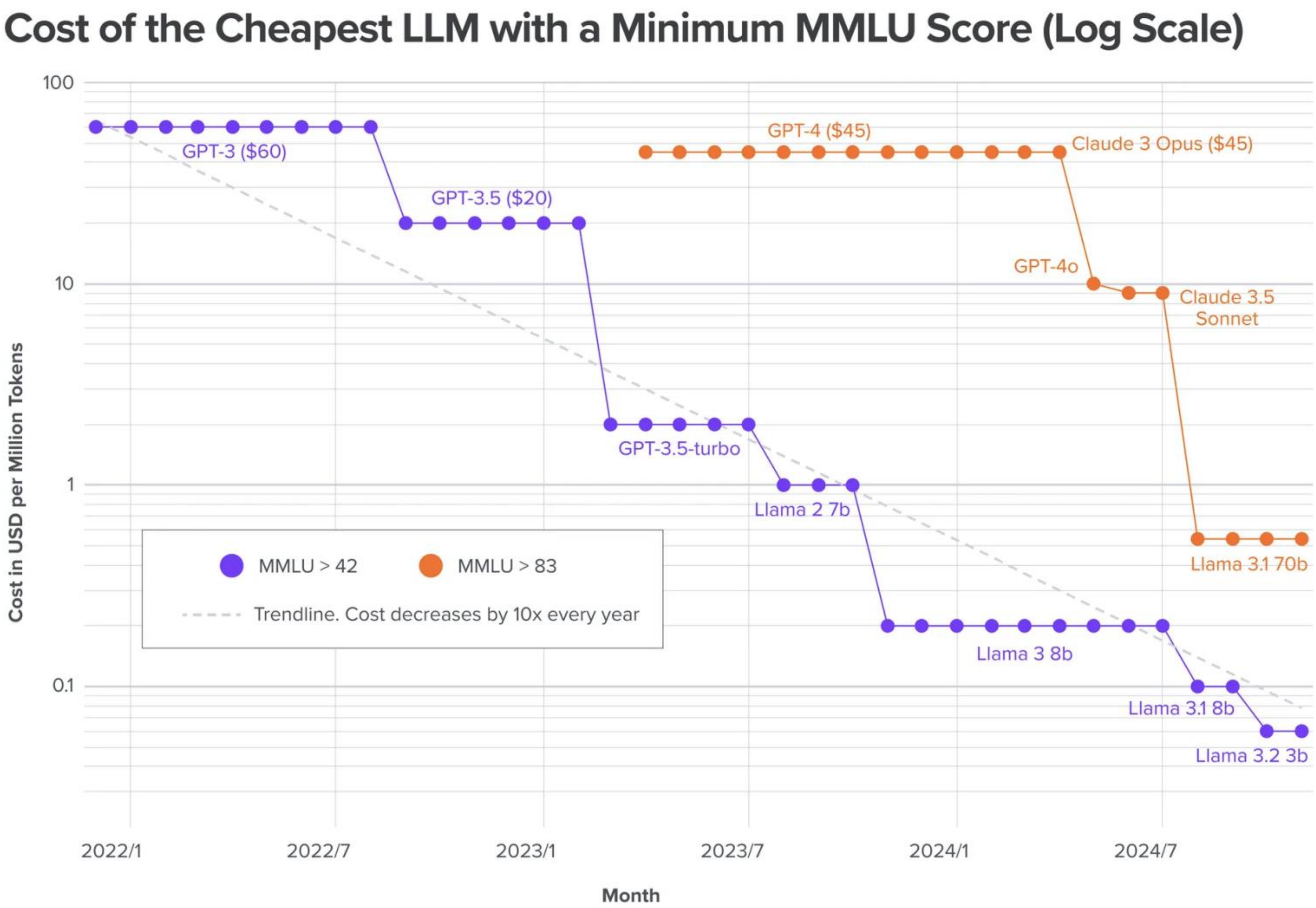
MODEL SIZE	TYPE	LITE	TURBO	REFERENCE
8B	Text	$0.10	$0.18	$0.20
11B	Vision		$0.18	
70B	Text	$0.54	$0.88	$0.90
90B	Vision		$1.20	
405B	Text		$3.50	

DeepSeek

MODEL	PRICE 1M TOKENS
DeepSeek-V3	$1.25
DeepSeek-R1	$7.00

Feature	DeepSeek-R1
Model Type	Mixture of Experts (MoE)
Total Parameters	671 billion
Active Parameters (per pass)	37 billion

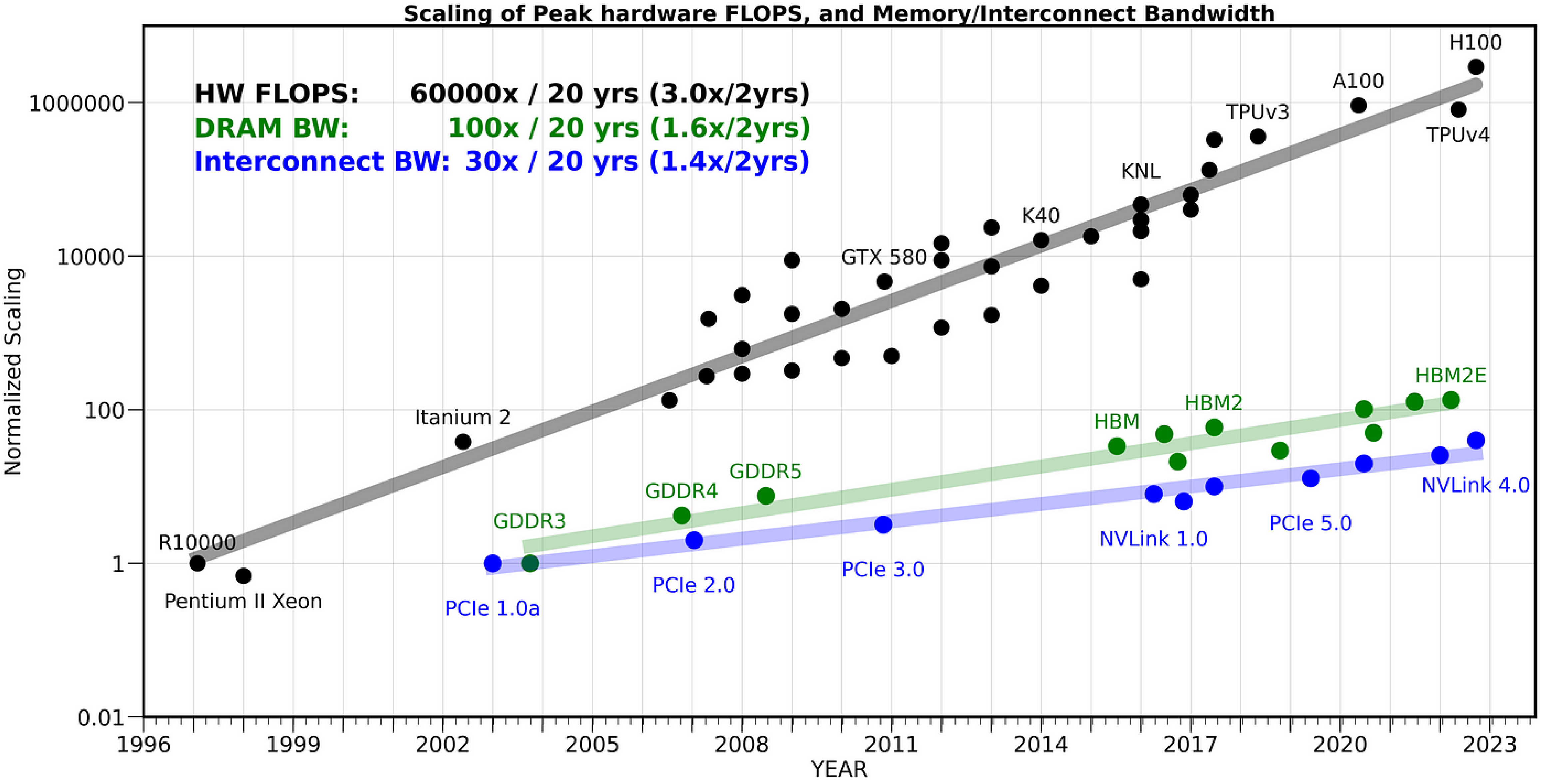
NAVER Cloud



Cost of the Cheapest LLM with a Minimum MMLU Score (Log Scale)
100
10
1
0.1
Cost in USD per Million Tokens
GPT-3 ($60)
GPT-3.5 ($20)
GPT-3.5-turbo
Llama 2 7b
Llama 3 8b
Llama 3.1 8b
Llama 3.2 3b
GPT-4 ($45)
Claude 3 Opus ($45)
GPT-4o
Claude 3.5 Sonnet
Llama 3.1 70b
MMLU > 42
MMLU > 83
Trendline. Cost decreases by 10x every year
2022/1
2022/7
2023/1
2023/7
2024/1
2024/7
Month

The Price Per Token of the Most Demanded Model Stays Approximately Constant

Model	Time Period	Output Price ($/M tokens)
GPT-4 (winner)	Mar 2023	$60.00
GPT-4	July 2023	$30.00 [50% drop]
GPT-4	November 2023	$15.00
GPT-4	March 2024	$1.50
Claude 3 Opus (becomes winner)	March 2024	$75.00
GPT-4o	March 2024	$30.00
Claude 3.5 Opus (becomes winner)	June 2024	$75.00
o1 (becomes winner)	September 2024	$60.00
o1	Dec 2024	$15.00
o3 (becomes winner)	December 2024	$60.00
Claude 4 Opus (becomes winner)	May 2025	$75.00
o3	June 2025	$8.00


Scaling of Peak hardware FLOPS, and Memory/Interconnect Bandwidth
HW FLOPS: 60000x / 20 yrs (3.0x/2yrs)
DRAM BW: 100x / 20 yrs (1.6x/2yrs)
Interconnect BW: 30x / 20 yrs (1.4x/2yrs)
Normalized Scaling
YEAR
1000000
10000
100
1
0.01
1996
1999
2002
2005
2008
2011
2014
2017
2020
2023
R10000
Pentium II Xeon
Itanium 2
GTX 580
K40
KNL
TPUv3
A100
H100
TPUv4
GDDR3
GDDR4
GDDR5
HBM
HBM2
HBM2E
PCIe 1.0a
PCIe 2.0
PCIe 3.0
NVLink 1.0
PCIe 5.0
NVLink 4.0

NAVER Cloud

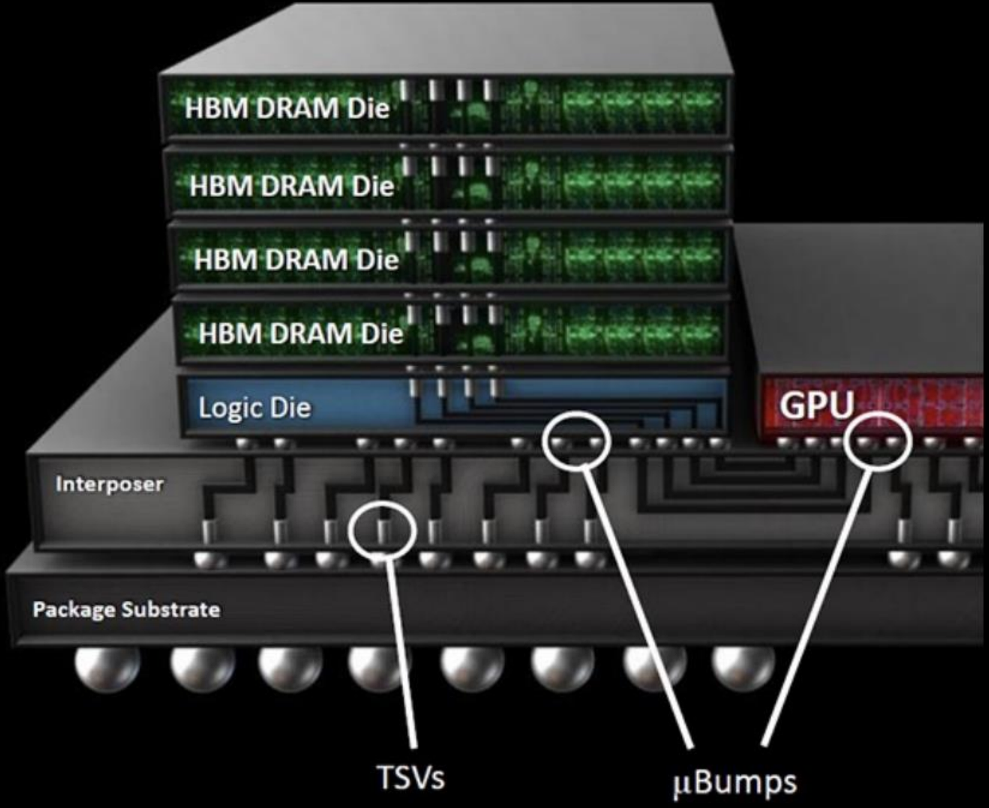
Critical Issues on NVIDIA's Solutions

NVIDIA Accelerator Specification Comparison

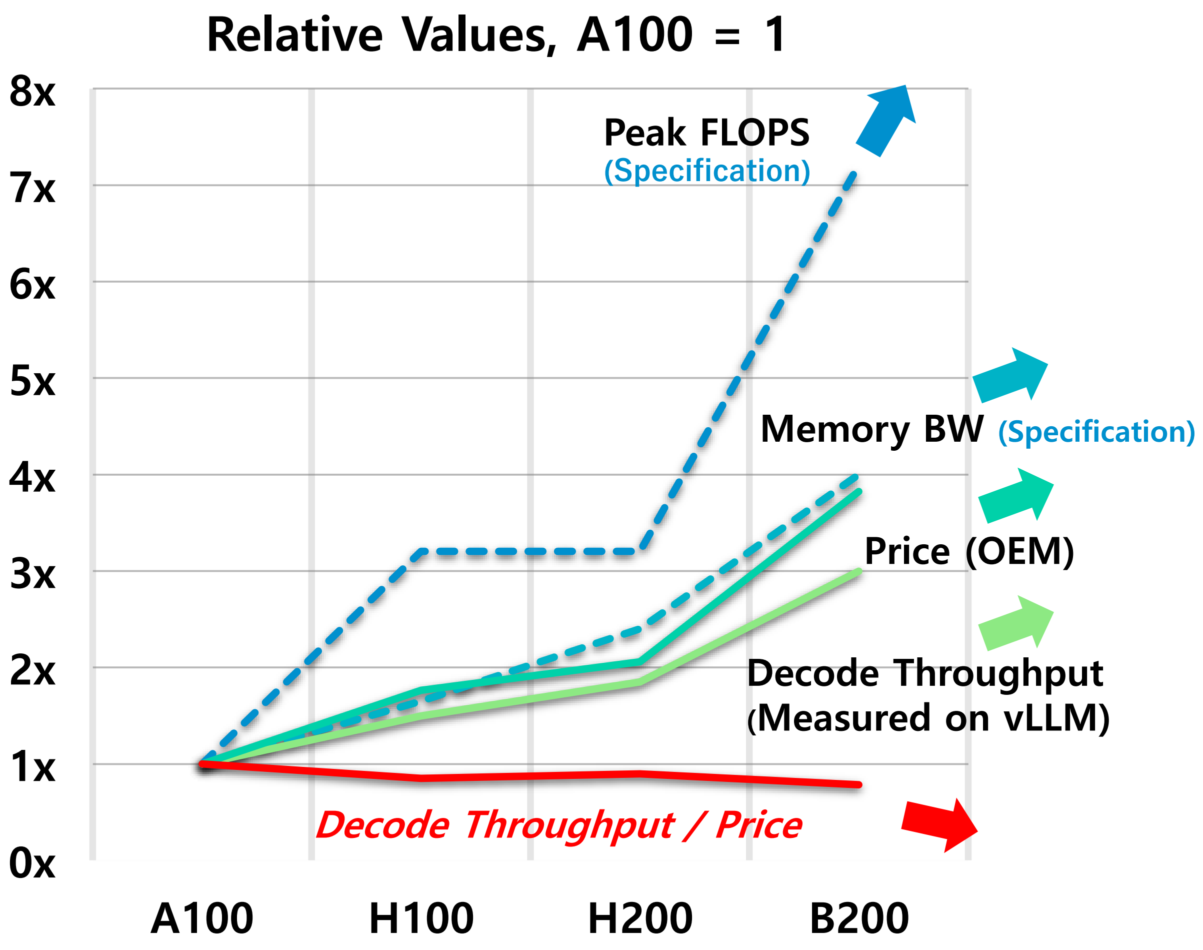
	H100	A100 (80GB)	V100
FP32 CUDA Cores	16896	6912	5120
Tensor Cores	528	432	640
Boost Clock	~1.78GHz (Not Finalized)	1.41GHz	1.53GHz
Memory Clock	4.8Gbps HBM3	3.2Gbps HBM2e	1.75Gbps HBM2
Memory Bus Width	5120-bit	5120-bit	4096-bit
Memory Bandwidth	3TB/sec	2TB/sec	900GB/sec
VRAM	80GB	80GB	16GB/32GB
FP32 Vector	60 TFLOPS	19.5 TFLOPS	15.7 TFLOPS
FP64 Vector	30 TFLOPS	9.7 TFLOPS (1/2 FP32 rate)	7.8 TFLOPS (1/2 FP32 rate)
INT8 Tensor	2000 TOPS	624 TOPS	N/A
FP16 Tensor	1000 TFLOPS	312 TFLOPS	125 TFLOPS
TF32 Tensor	500 TFLOPS	156 TFLOPS	N/A
FP64 Tensor	60 TFLOPS	19.5 TFLOPS	N/A
Interconnect	NVLink 4 18 Links (900GB/sec)	NVLink 3 12 Links (600GB/sec)	NVLink 2 6 Links (300GB/sec)
GPU	GH100 (814mm2)	GA100 (826mm2)	GV100 (815mm2)
Transistor Count	80B	54.2B	21.1B
TDP	700W	400W	300W/350W
Manufacturing Process	TSMC 4N	TSMC 7N	TSMC 12nm FFN
Interface	SXM5	SXM4	SXM2/SXM3
Architecture	Hopper	Ampere	Volta

The key

Too much TD



Inference cost should be evaluated based on total cost and throughput. The problem is that actual performance is no longer keeping up with the cost.



Gap between spec and actual performance

The gap between peak FLOPS and memory bandwidth continues to widen.
Actual inference throughput is dropping even more sharply.
As a result, the cost-efficiency of LLMs is deteriorating,
and users are now paying more for less.

Splitwise Paper (Microsoft, Nov. 2023)

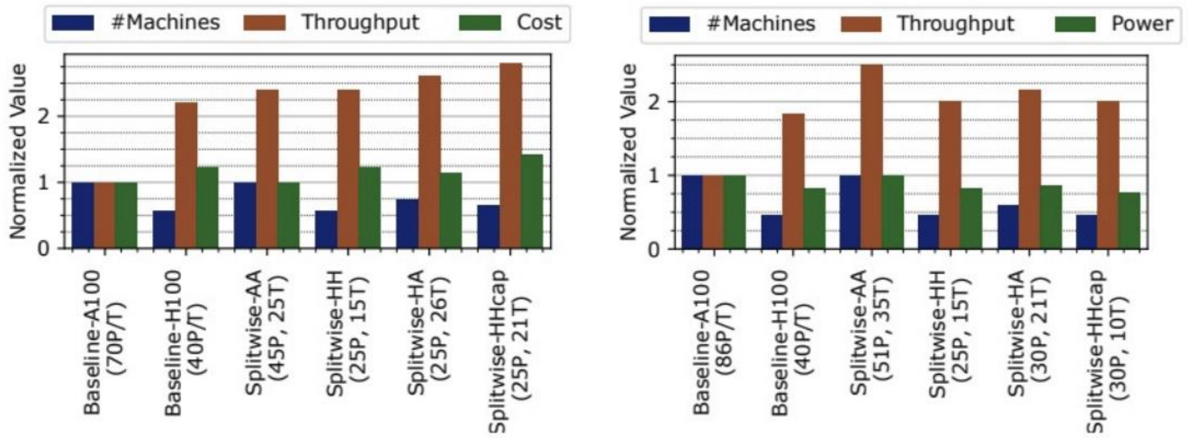
	A100	H100	Ratio
TFLOPs	19.5	66.9	3.43×
HBM capacity	80GB	80GB	1.00×
HBM bandwidth	2039GBps	3352GBps	1.64×
Power	400W	700W	1.75×
NVLink	50 Gbps	100Gbps	2.00×
Infiniband	200GBps	400GBps	2.00×
Cost per machine [3]	$17.6/hr	$38/hr	2.16×

TABLE I: NVIDIA A100 vs. H100 relevant specifications.

	Coding			Conversation		
	A100	H100	Ratio	A100	H100	Ratio
TTFT	185 ms	95 ms	0.51×	155 ms	84 ms	0.54×
TBT	52 ms	31 ms	0.70×	40 ms	28 ms	0.70×
E2E	856 ms	493 ms	0.58×	4957 ms	3387 ms	0.68×
Cost [3]	$0.42	$0.52	1.24×	$2.4	$3.6	1.50×
Energy	1.37 Whr	1.37 Whr	1.00×	7.9 Whr	9.4 Whr	1.20×

TABLE IV: P50 request metrics on A100 vs. H100 without batching on Llama2-70B.

Limit the H100 to A100 perf. levels,
or simply use the A100



(a) Iso-power. (b) Iso-cost.

Fig. 17: Summary of throughput-optimized cluster designs.

The A100 outperforms the H100 in the Decoding phase

NAVER Cloud

Toward Better Inference Performance with AI-inspired Optimization Techs.

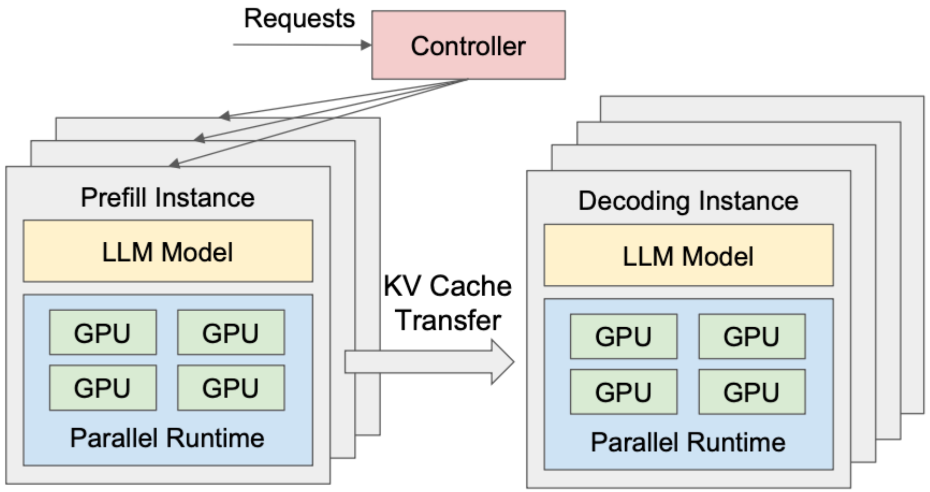
Disaggregated Inference

- Compute-bound Prefill Machine

  <-> Memory-bound Decode Machine

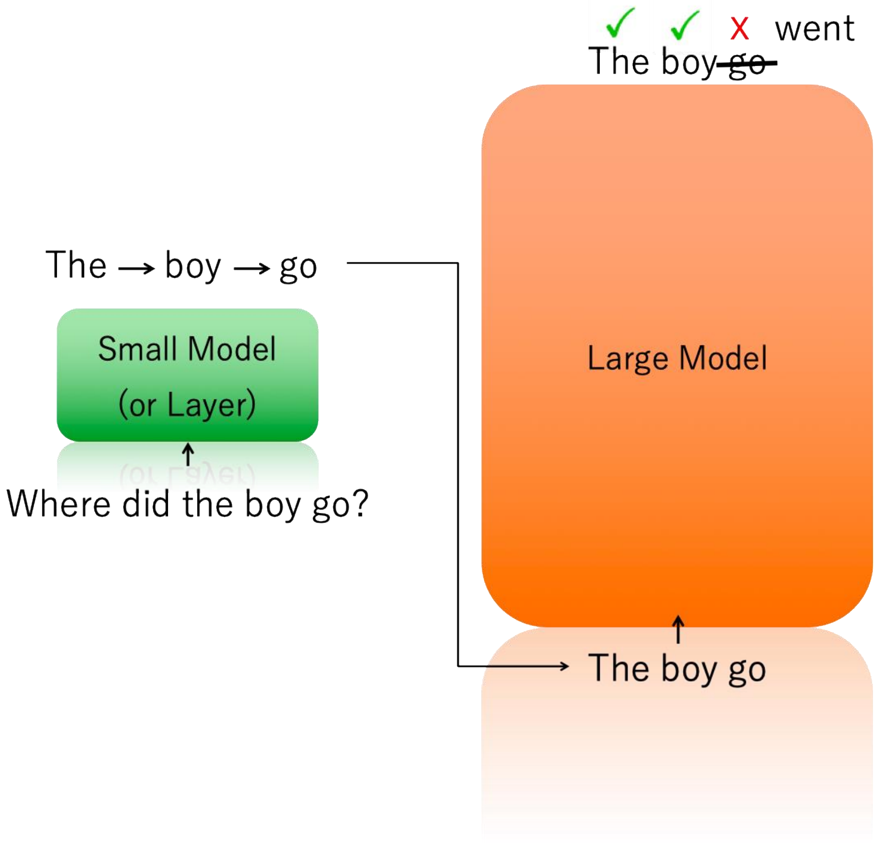
- Various options can also be considered

  (ex. Gaudi-3 + Gaudi-2d).



Speculative Decoding

- Faster decoding by using draft model



NAVER Cloud

The Cost of Dynamic Reasoning: Demystifying AI Agents and Test-Time Scaling from an AI Infrastructure Perspective

Jiin Kim Byeongjun Shin Jinha Chung Minsoo Rhu
KAIST
{jiin.kim, byeongjun.shin, jinha.chung, mrhu}@kaist.ac.kr

		Accuracy (%)	Latency (seconds)	Energy (Wh/query)	Power @ 71.4 Million Queries/day (Watt)	Power @ 13.7 Billion Queries/day (Watt)
8B	**ShareGPT**	–	4.23 (1×)	0.32 (1×)	1.0 M	182.7 M
	Reflexion	38	649.34 (153.7×)	41.53 (130.9×)	123.6 M	23.7 G
	LATS	80	380.90 (90.1×)	22.76 (71.7×)	67.7 M	13.0 G
70B	**ShareGPT**	–	6.40 (1×)	2.55 (1×)	7.6 M	1.5 G
	Reflexion	67	720.00 (112.6×)	348.41 (136.5×)	1.0 G	198.9 G
	LATS	82	305.67 (47.8×)	158.48 (62.1×)	471.5 M	90.5 G

TABLE III: Energy and power demands of handling an AI agent service request on HotpotQA. We report accuracy, latency, GPU energy consumption, and datacenter-wide power demand under current and future traffic scenarios (71.4 Million Queries/day and 13.7 Billion Queries/day) for two agentic workflows (Reflexion and LATS) using Llama-3.1-Instruct 8B and 70B models. ShareGPT serves as the baseline for conventional single-turn LLM inference. Numbers in parentheses denote the multiplicative increase relative to ShareGPT. Reflexion and LATS design points were selected based on the highest-accuracy configurations in Figure 22. The datacenter-wide power is computed by $P = \text{(Wh/query)} \times \text{(Queries/day)}/(24 \text{ hours})$.

Agentic AI requires a completely new data center architecture

DroidSpeak: KV Cache Sharing for Cross-LLM Communication and Multi-LLM Serving

Yuhan Liu[1] Yuyang Huang[1] Jiayi Yao[1] Shaoting Feng[1] Zhuohan Gu[1] Kuntai Du[1] Hanchen Li[1] Yihua Cheng[1]
Junchen Jiang[1] Shan Lu[2] Madan Musuvathi[2] Esha Choukse[2]
[1]University of Chicago [2]Microsoft

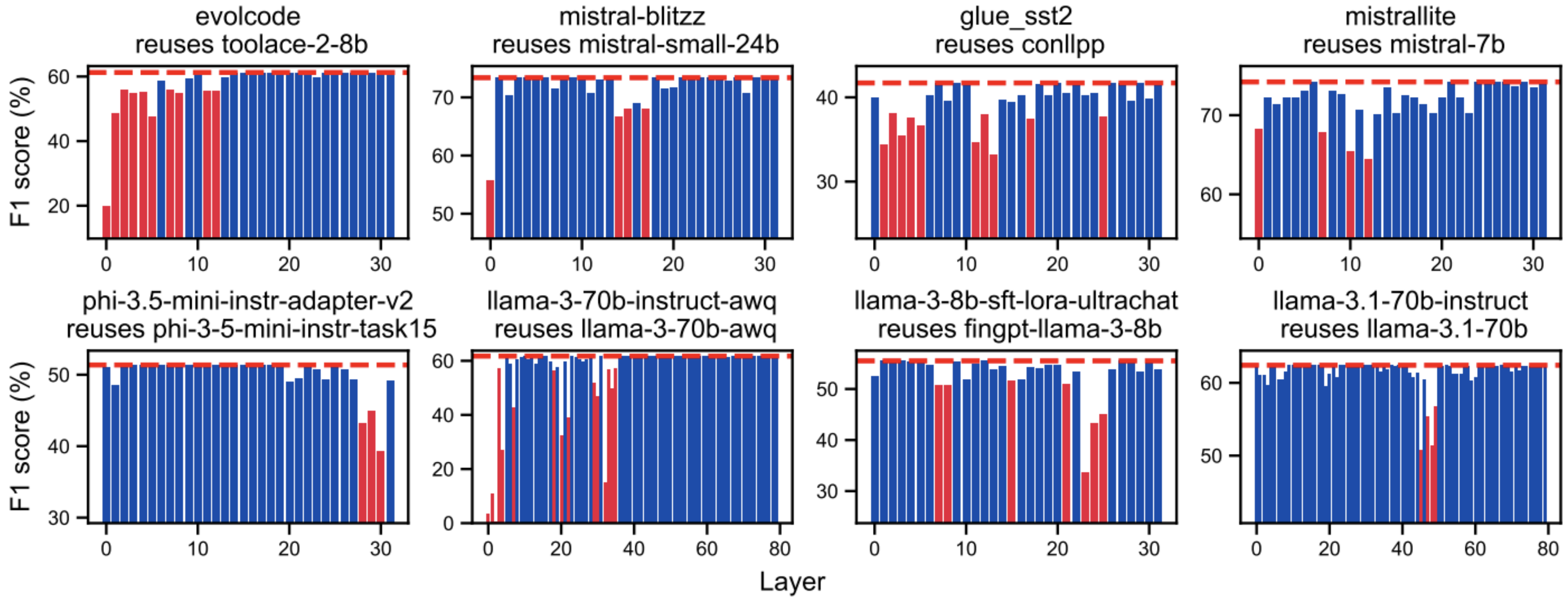
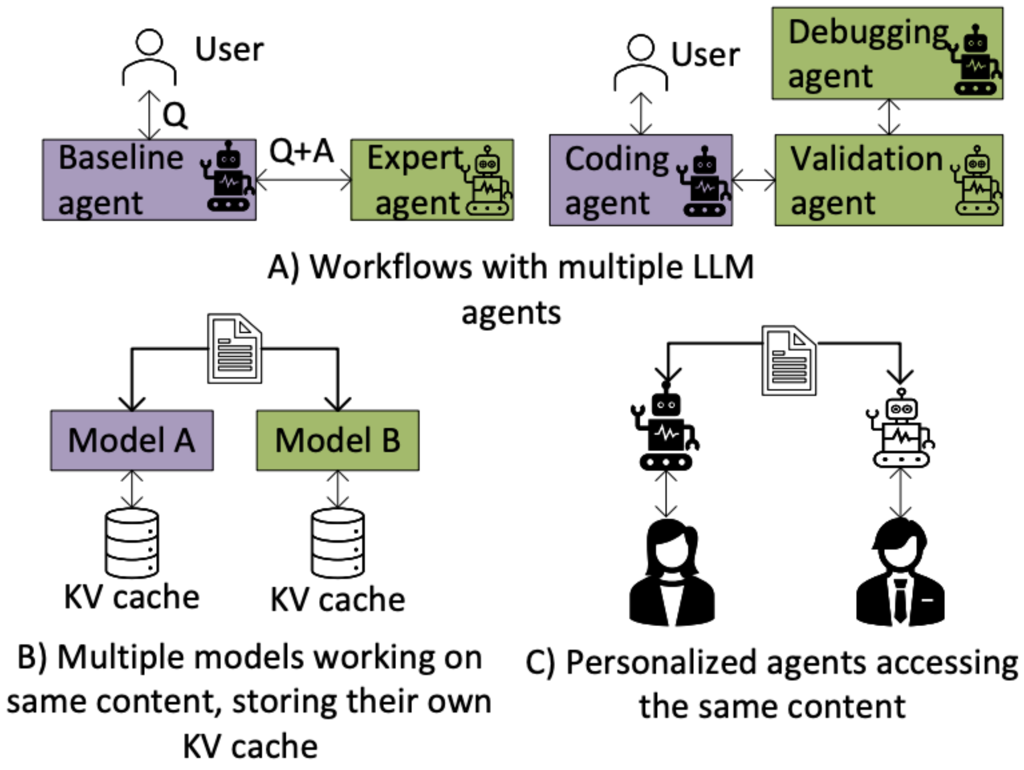


Figure 7: *Different layers have different sensitivities to deviation in KV cache. Plotted by reusing only one layer's KV cache from the sender model on the receiver model. The red dashed line is the original accuracy of the receiver model. The bars colored red are those that have an F1 score drop of over 10% compared to the original receiver model.*

DroidSpeak: KV Cache Sharing for Cross-LLM Communication and Multi-LLM Serving

Yuhan Liu[1] Yuyang Huang[1] Jiayi Yao[1] Shaoting Feng[1] Zhuohan Gu[1] Kuntai Du[1] Hanchen Li[1] Yihua Cheng[1]
Junchen Jiang[1] Shan Lu[2] Madan Musuvathi[2] Esha Choukse[2]
[1]University of Chicago [2]Microsoft



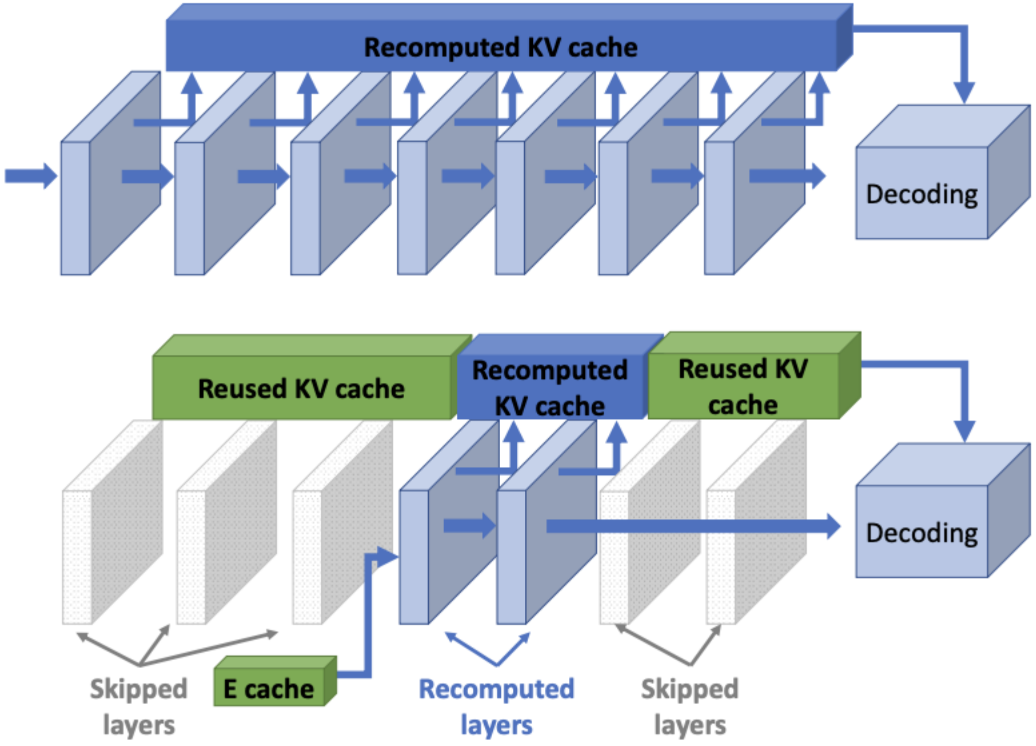
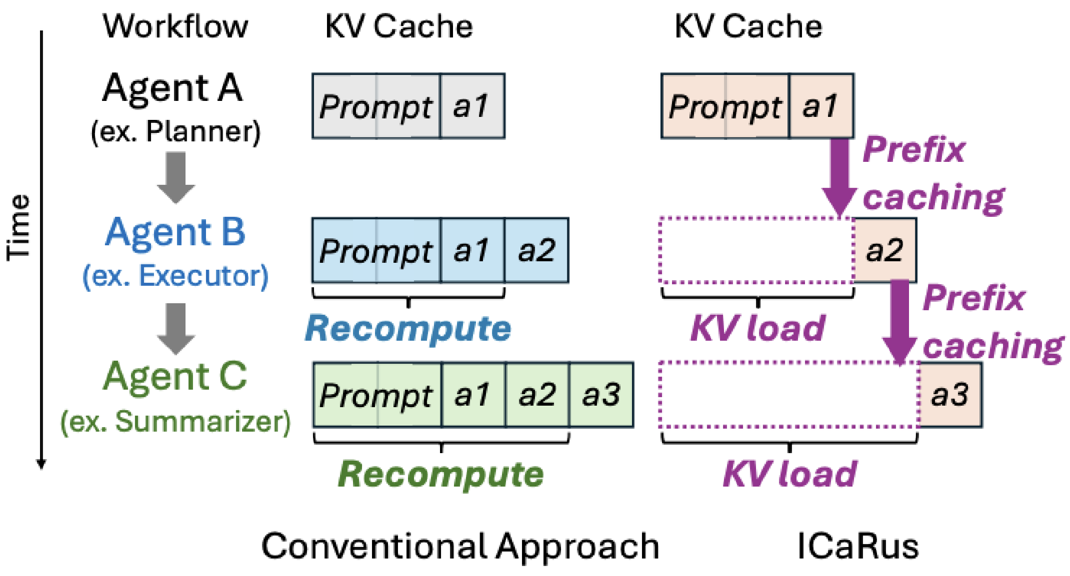


Figure 9: *DroidSpeak chooses the critical layer groups (layers 4-5) to re-compute, and reuse KV cache for other layers.*

ICaRus: Identical Cache Reuse for Efficient Multi Model Inference



(a) KV Cache management strategies in agent workflow using multi model

	Single Model	Multi Model	
		Conventional	ICaRus (Ours)
Training Method	Prompting	Fine-tuning	**Fine-tuning (only logical Decoder)**
Task Performance	Weak	Strong	**Strong**
KV Sharing	Inherent	Unsupported	**Supported**
KV Memory Usage	Low	High	**Low**
# Prefill Recomputation	Low	High	**Low**

(b) Comparison of ICaRus and conventional approaches

Figure 1: Comparison of KV cache management strategies and effectiveness in multi model scenarios between conventional approaches and ICaRus.

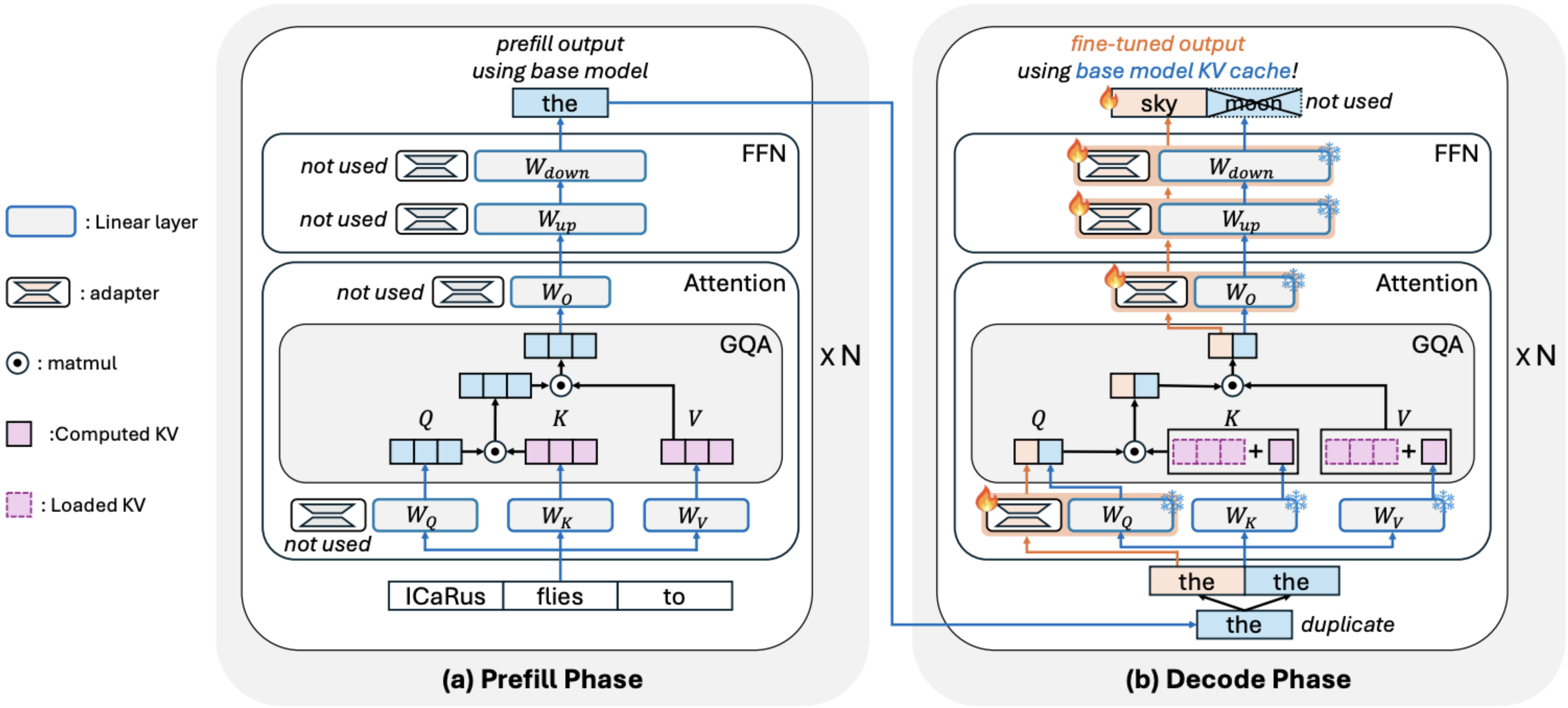


Figure 3: Overview of the ICaRus architecture. The base model, a pretrained decoder-only Transformer, serves as the logical encoder, while the adapter-tuned model (consisting of the base model and a tunable adapter) serves as the logical decoder. The blue and orange lines indicate computations performed by the base model and the adapter-tuned model, respectively.

Table 2: Comparisons of prior task-specific fine-tuning and ICaRus on various datasets.

Model	Method	Math		Coding		Knowledge
		GSM8K	GSM+	HEval	HEval+	GPQA
LLaMA3.1-8B	No-tuning	25.9	18.0	36.6	29.9	16.7
	Task-tuning	**69.7**	**48.5**	48.2	41.5	27.3
	ICaRus (Ours)	67.9	45.8	**48.2**	**43.9**	**28.8**
Qwen3-8B-Base	No-tuning	11.8	12.5	68.3	61.6	24.2
	Task-tuning	85.4	66.1	81.7	75.6	**34.3**
	ICaRus (Ours)	**87.3**	**67.5**	**86.6**	**79.9**	33.8

Table 3: Comparison of task-specific fine-tuning and ICaRus across different model sizes (Qwen3-1.7B/8B/14B-Base) trained on the MetaMathQA-40K dataset.

Model	Qwen3-1.7B-Base		Qwen3-8B-Base		Qwen3-14B-Base	
Method	Task-tuning	ICaRus	Task-tuning	ICaRus	Task-tuning	ICaRus
GSM8K	73.2	**74.0**	85.4	**87.3**	85.6	**88.8**
GSM+	53.7	**54.1**	66.1	**67.5**	66.7	**68.8**

Table 4: Comparison of task-specific fine-tuning and ICaRus in both single and multi model inference scenarios.

# Model	Method	Math		Coding		Knowledge	Avg.
		GSM8K	GSM-Plus	HEval	HEval+	GPQA	
Single Model	No-tuning	25.9	18.0	36.6	29.9	16.7	25.4
	Math-tuning	**69.7**	**48.5**	42.7	36.6	20.7	43.6
	Code-tuning	22.8	17.5	**48.2**	41.5	21.7	30.3
	Instruct-tuning	24.5	16.5	44.5	39.0	27.2	30.3
Multi Model	Task-tuning	**69.7**	**48.5**	**48.2**	41.5	27.2	**47.0**
	ICaRus (Ours)	67.9	45.8	**48.2**	**43.9**	**28.8**	46.9

NAVER Cloud

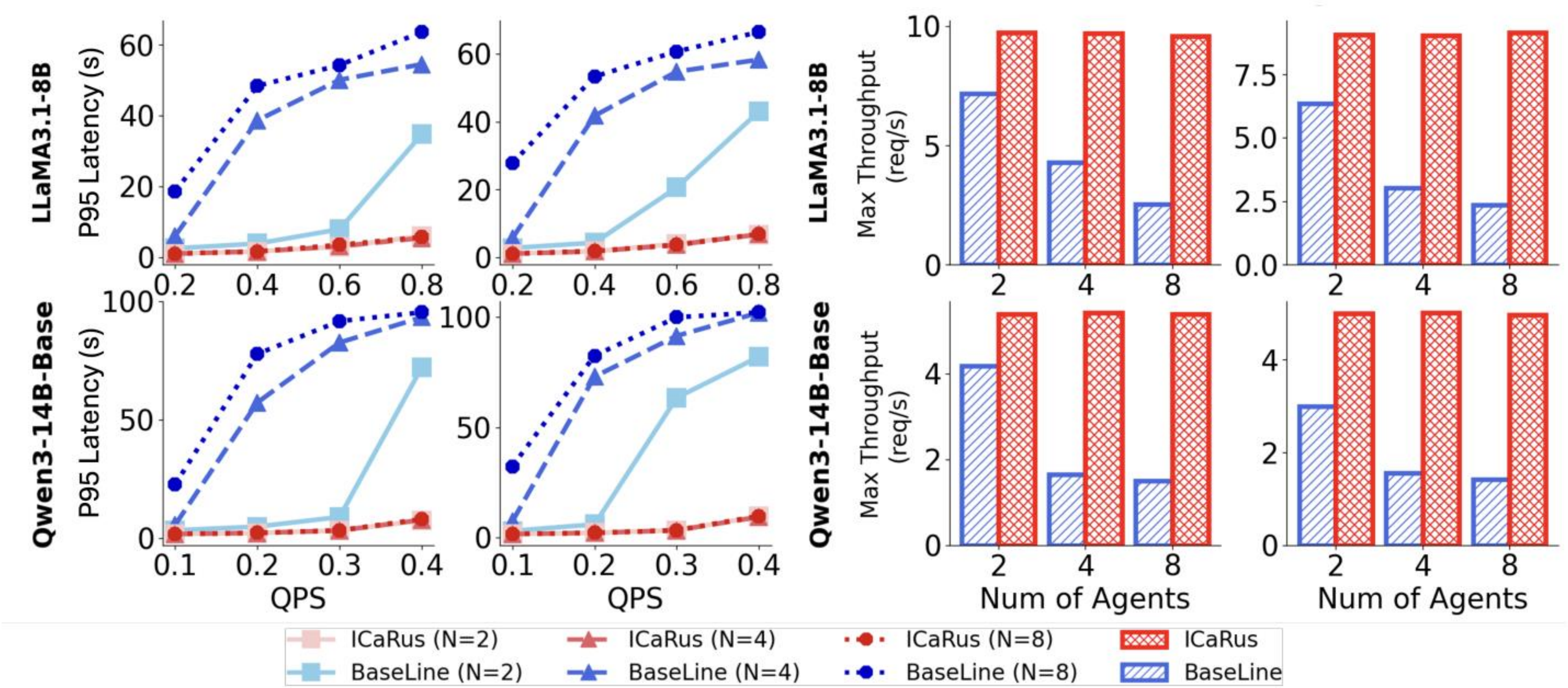


Figure 5: Comparison of P95 latency and maximum throughput across QPS for LLaMA3.1-8B and Qwen-3-14B Base under ReAct and Reflexion workflows.

Thank you