RAG, Agents, Finetuning, frameworks

Konrad Handrick

Workshop 3: stepping stone in AI

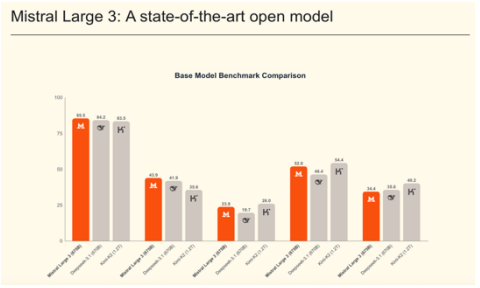
Dezember 2025

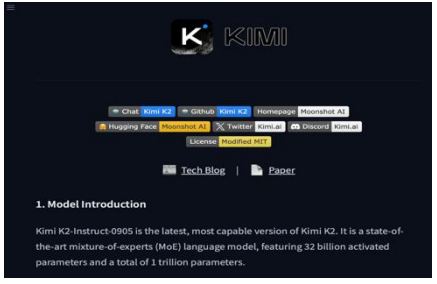
Today’s menu (less theoretical)

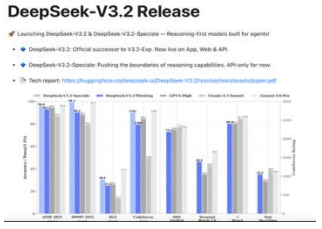
1. Catching up a little
2. Quantization
3. Finetuning
4. RAG
5. Agents

... it’ll become a little more complex though!

A few news first

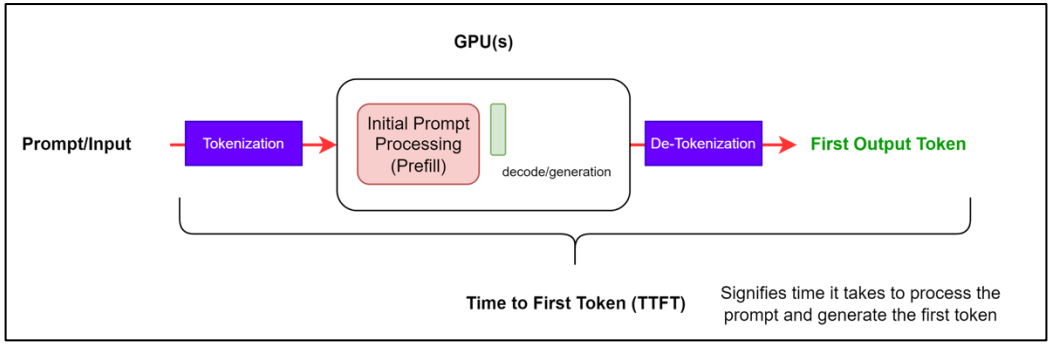




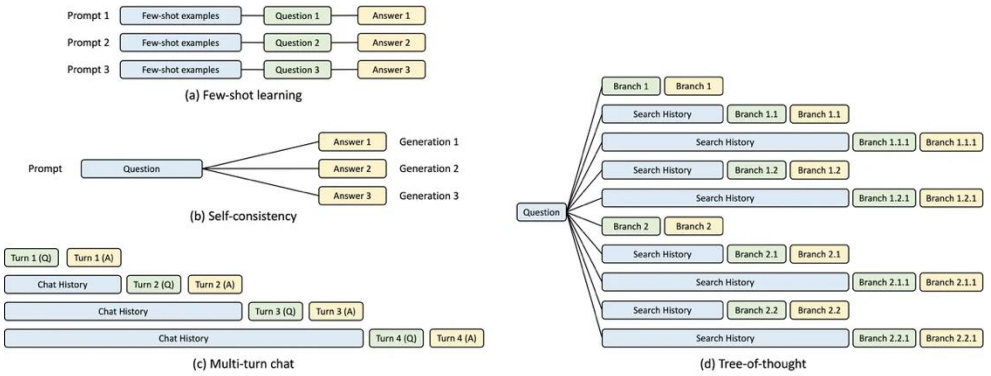


Top left: https://allenai.org/olmo
Top right: https://mistral.ai/news/mistral-3/
Bottom left: https://api-docs.deepseek.com/news/news251201
Bottom right: https://huggingface.co/moonshotai/Kimi-K2-Instruct-0905

Addendum: KV Caches

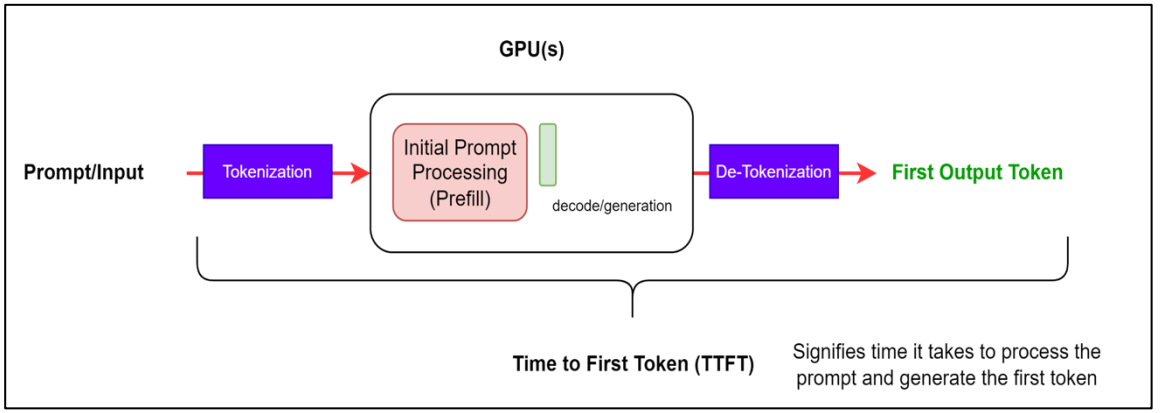


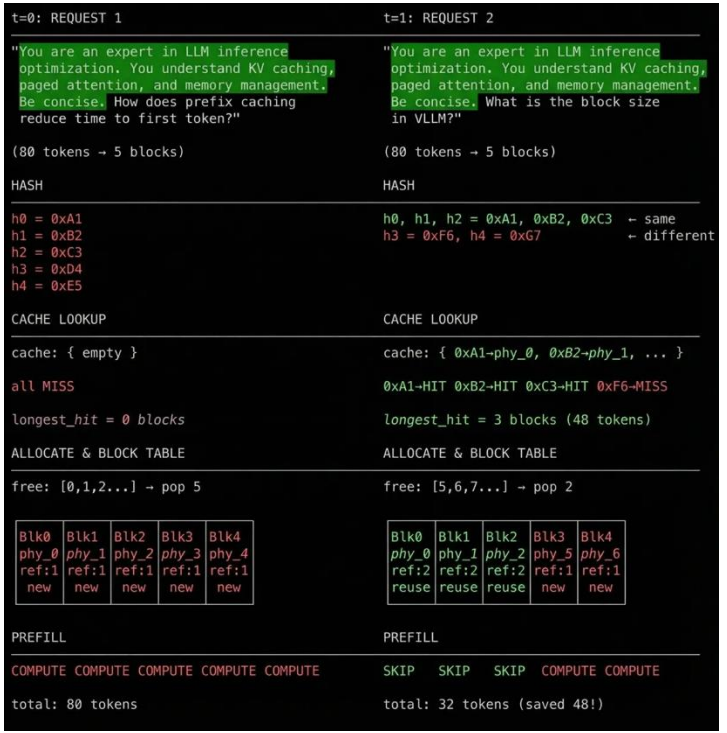
Language redundancy, prefixes, positionally-encoded tokens



Top: https://docs.nvidia.com/nim/benchmarking/llm/latest/metrics.html
Right: https://arxiv.org/abs/2309.06180

Addendum: KV Caches



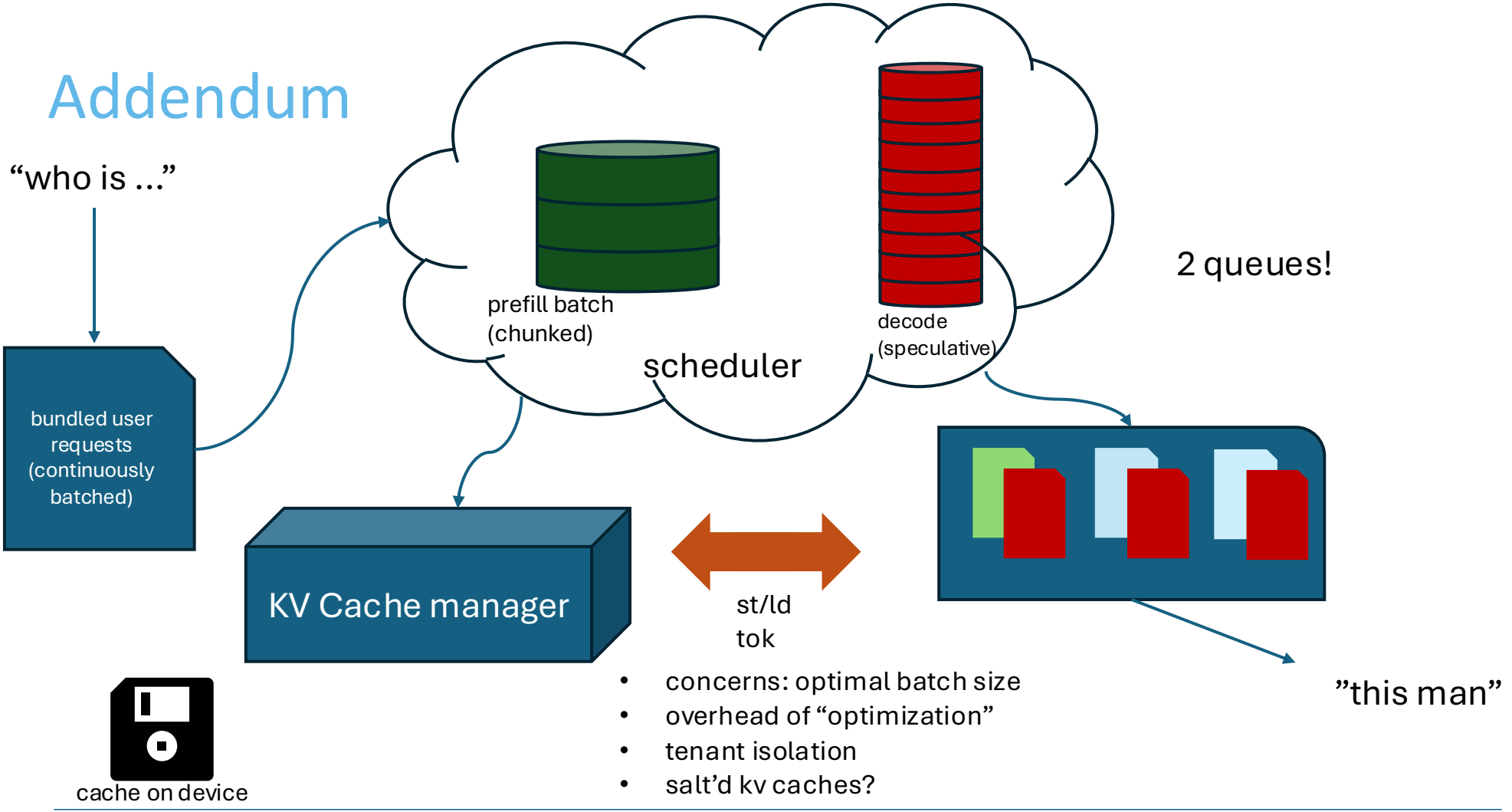


Top: https://docs.nvidia.com/nim/benchmarking/llm/latest/metrics.html
Right: https://sankalp.bearblog.dev/how-prompt-caching-works/

system prompt

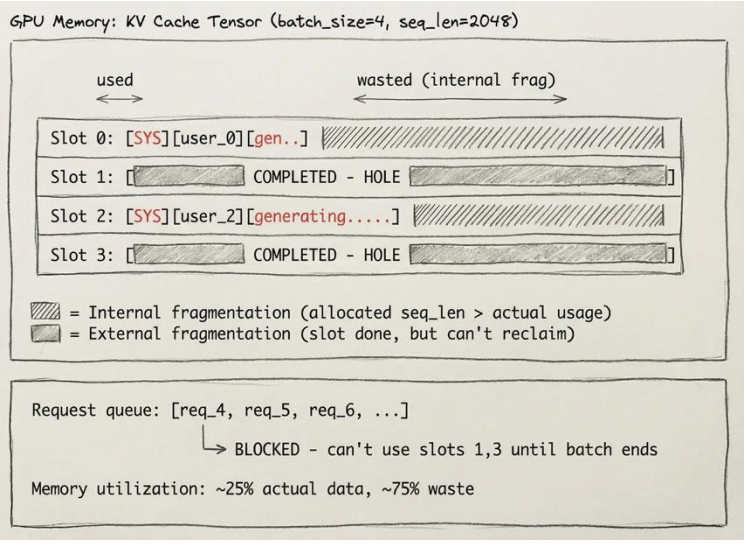
cache eviction

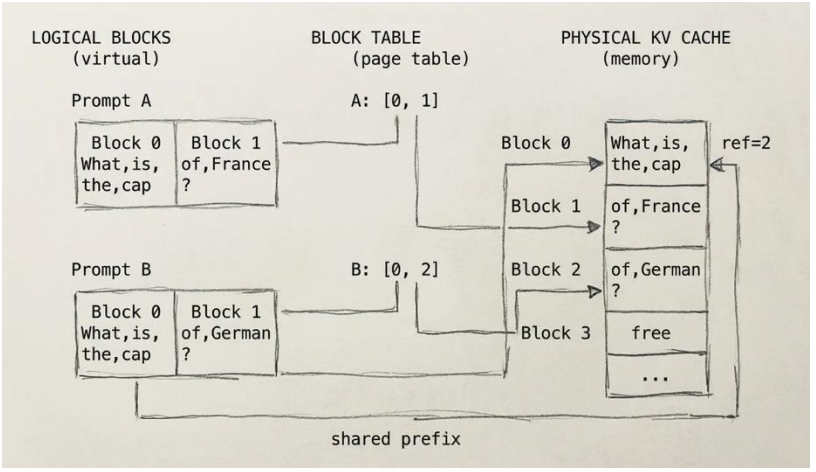
Addendum



- concerns: optimal batch size
- overhead of “optimization”
- tenant isolation
- salt’d kv caches?

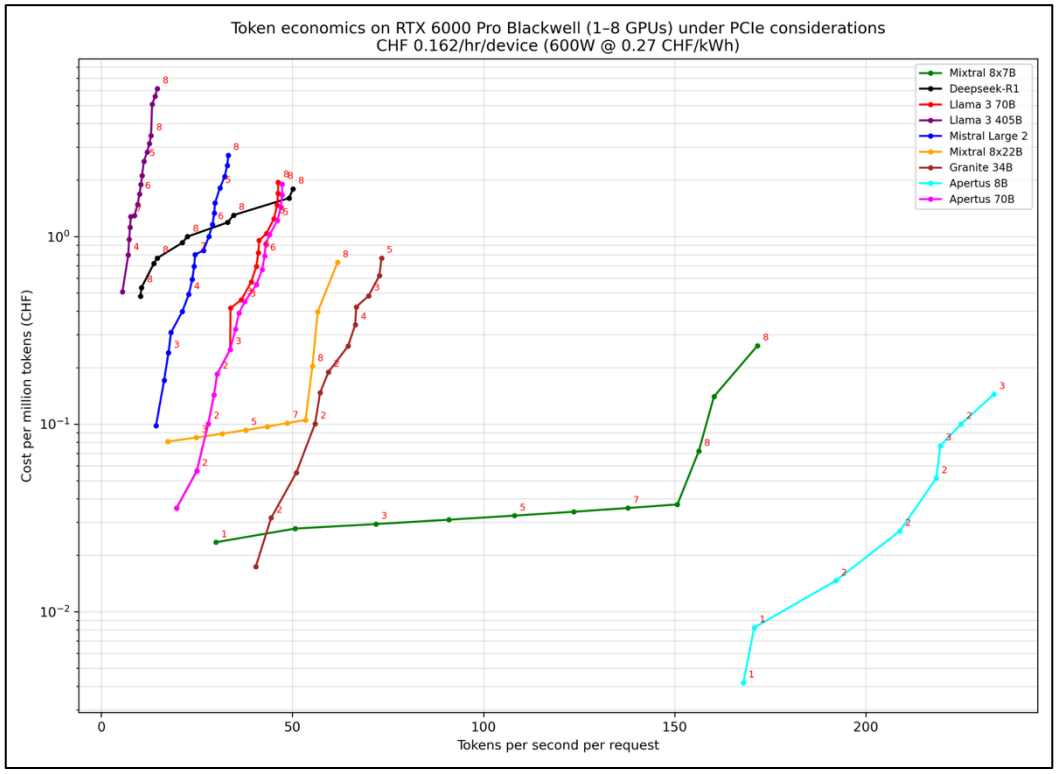
Addendum: KV Caches





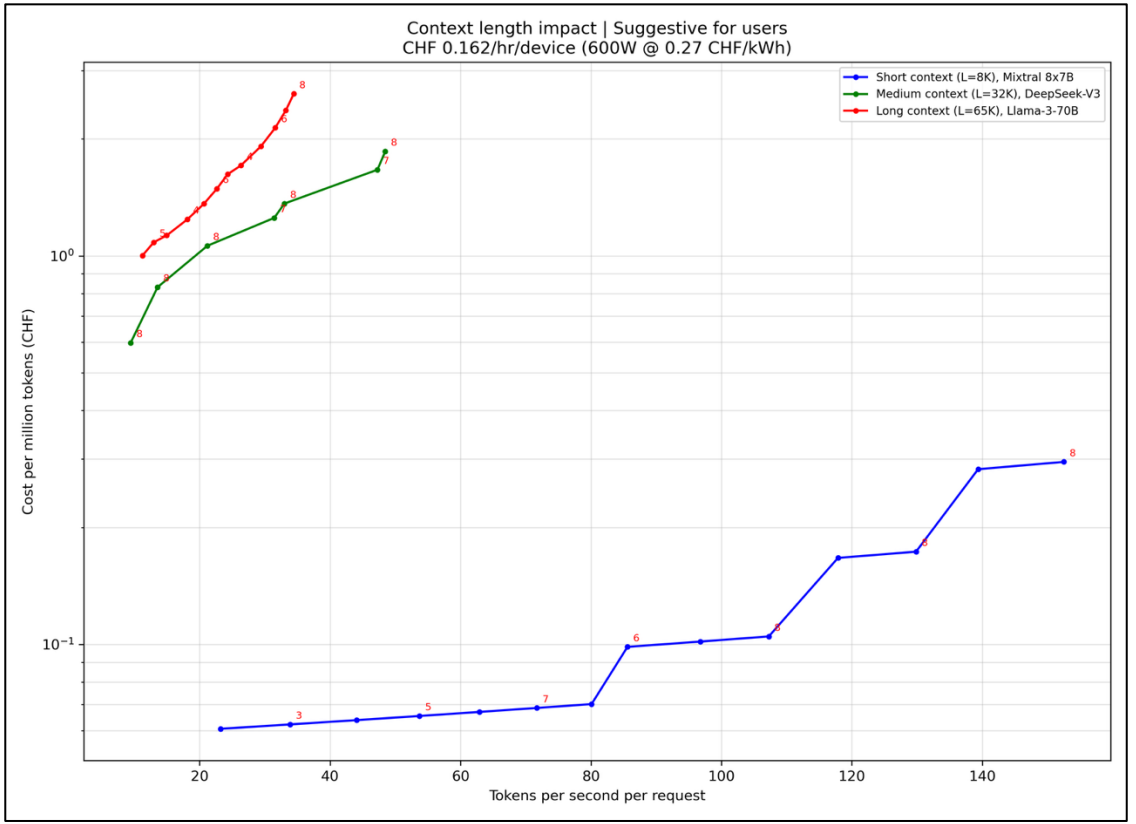
Left and right: https://sankalp.bearblog.dev/how-prompt-caching-works/

Pareto frontiers



Pareto frontiers

repo



The Economics of Large Language Models:

Token Allocation, Fine-Tuning, and Optimal Pricing*

Dirk Bergemann† Alessandro Bonatti‡ Alex Smolin§

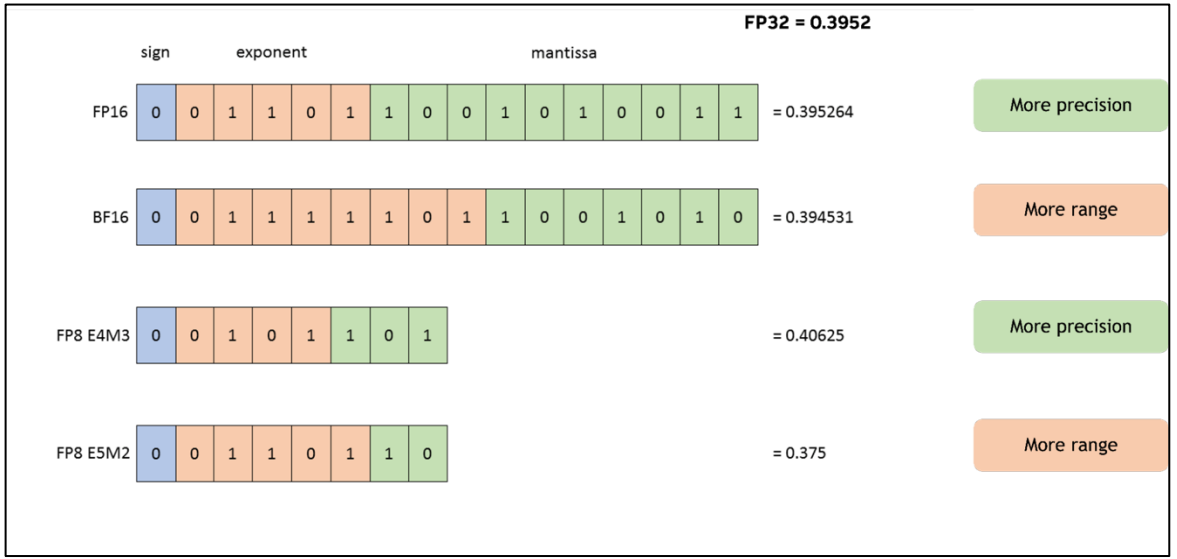
February 12, 2025

INFERENCE ECONOMICS OF LANGUAGE MODELS

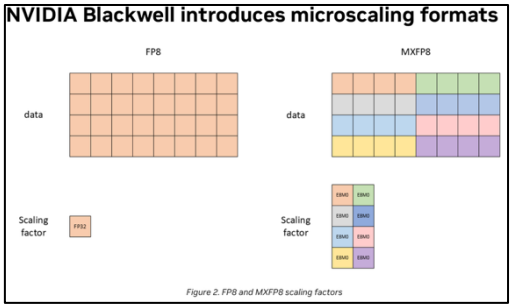
Ege Erdil
Epoch
ege@epoch.ai

Top: https://arxiv.org/abs/2502.07736
Bottom: https://arxiv.org/abs/2506.04645

Quantization

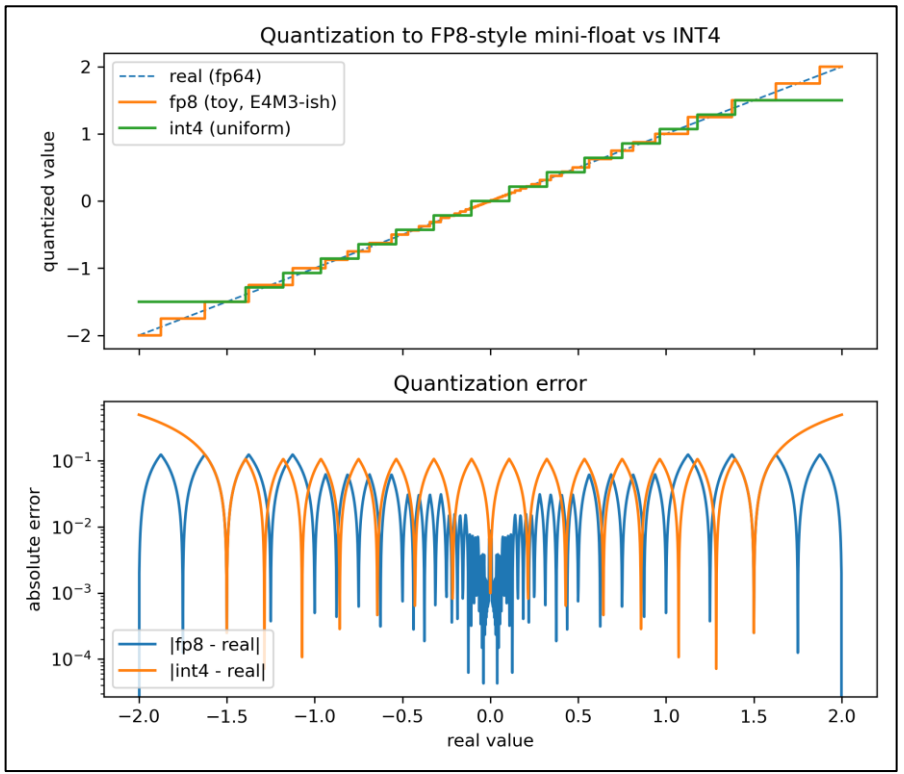


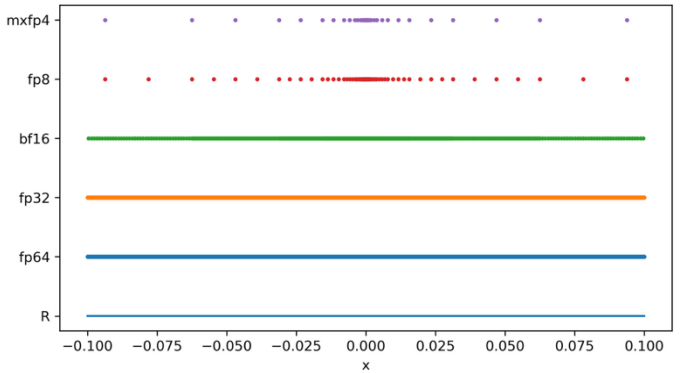
model size = num_bytes_per_weight * num_params
+ extra

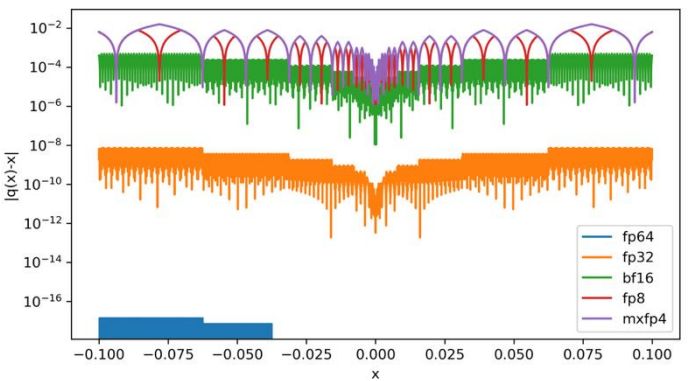


Top and right: https://developer.nvidia.com/blog/floating-point-8-an-introduction-to-efficient-lower-precision-ai-training

Quantization



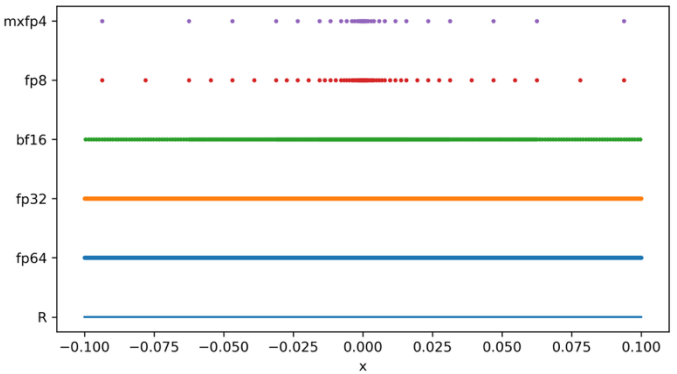


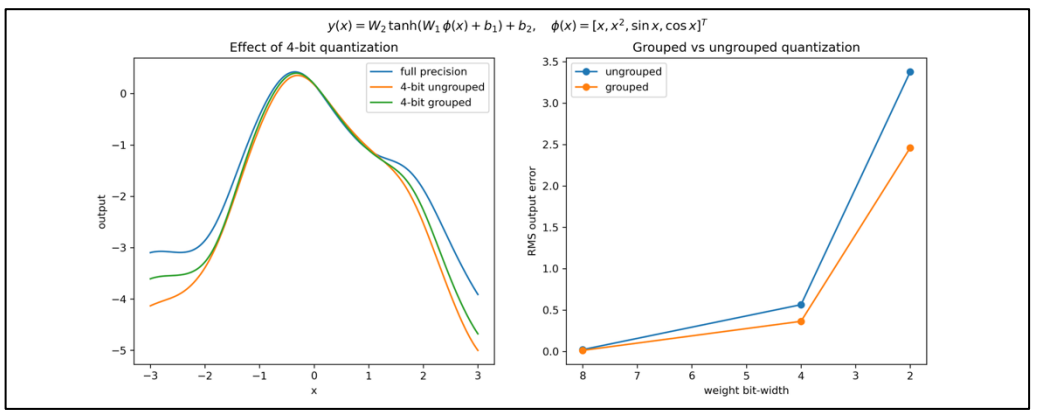


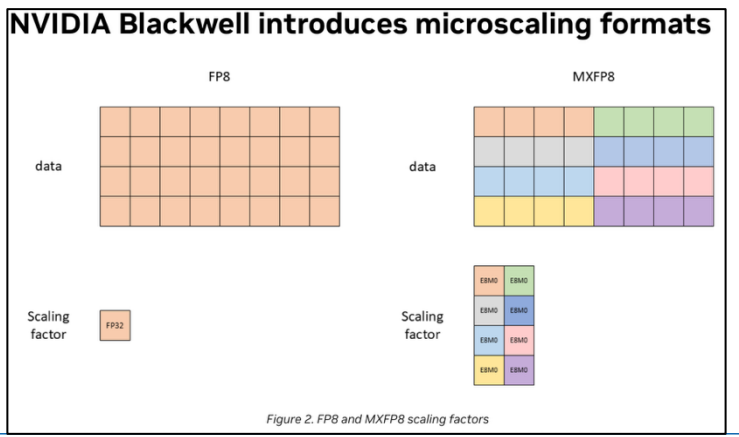
Quantization

model size = num_bytes_per_weight * num_params
+ extra

we’re on vectorized machines and want to save space
-> use groupings and quantize along







Quantization

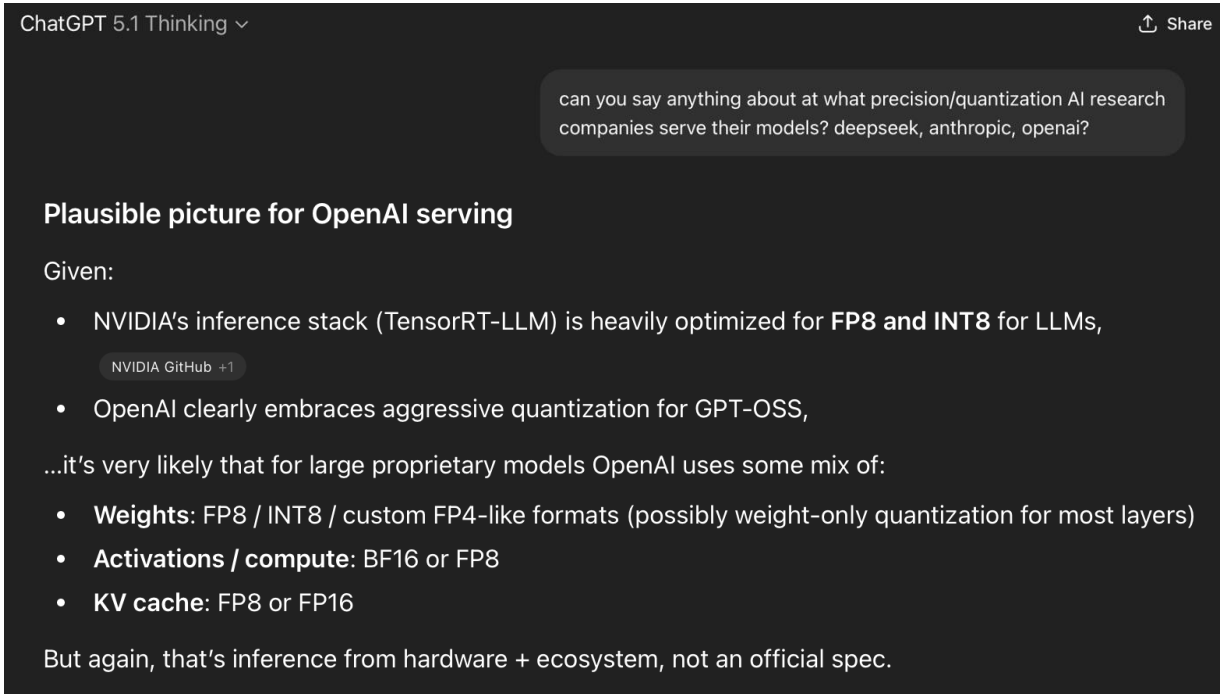
model size = num_bytes_per_weight * num_params
+ extra

training often in fp16

➔ weights in bf16

➔ sucks for us when we need to serve

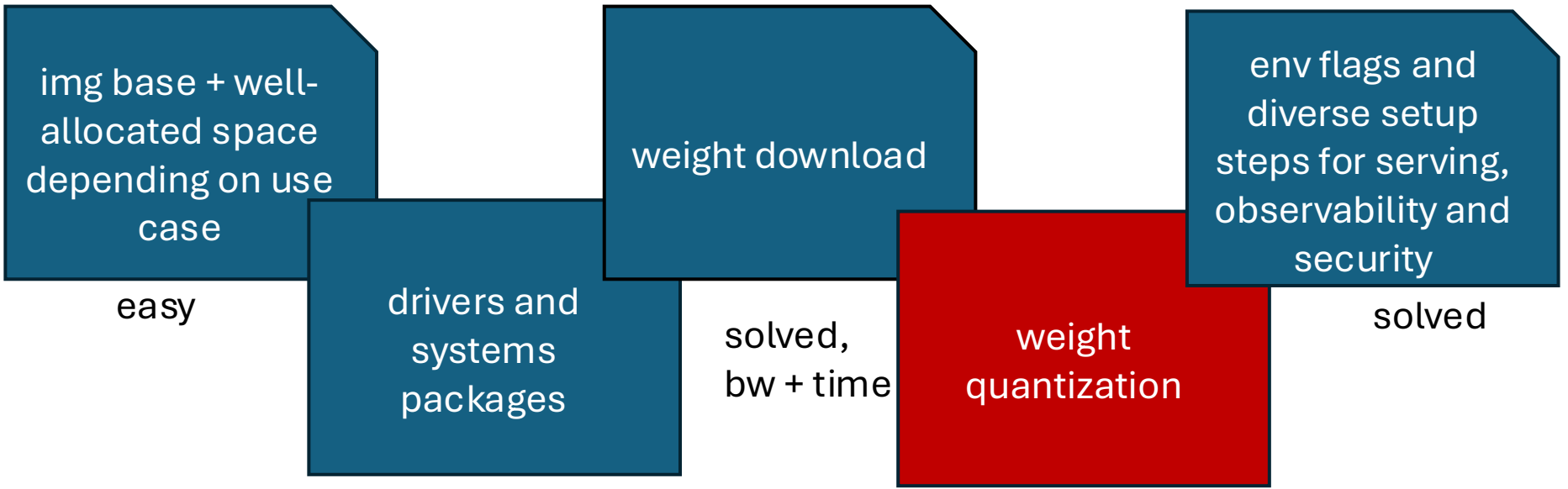
deepseek uses fp8 for training + inference



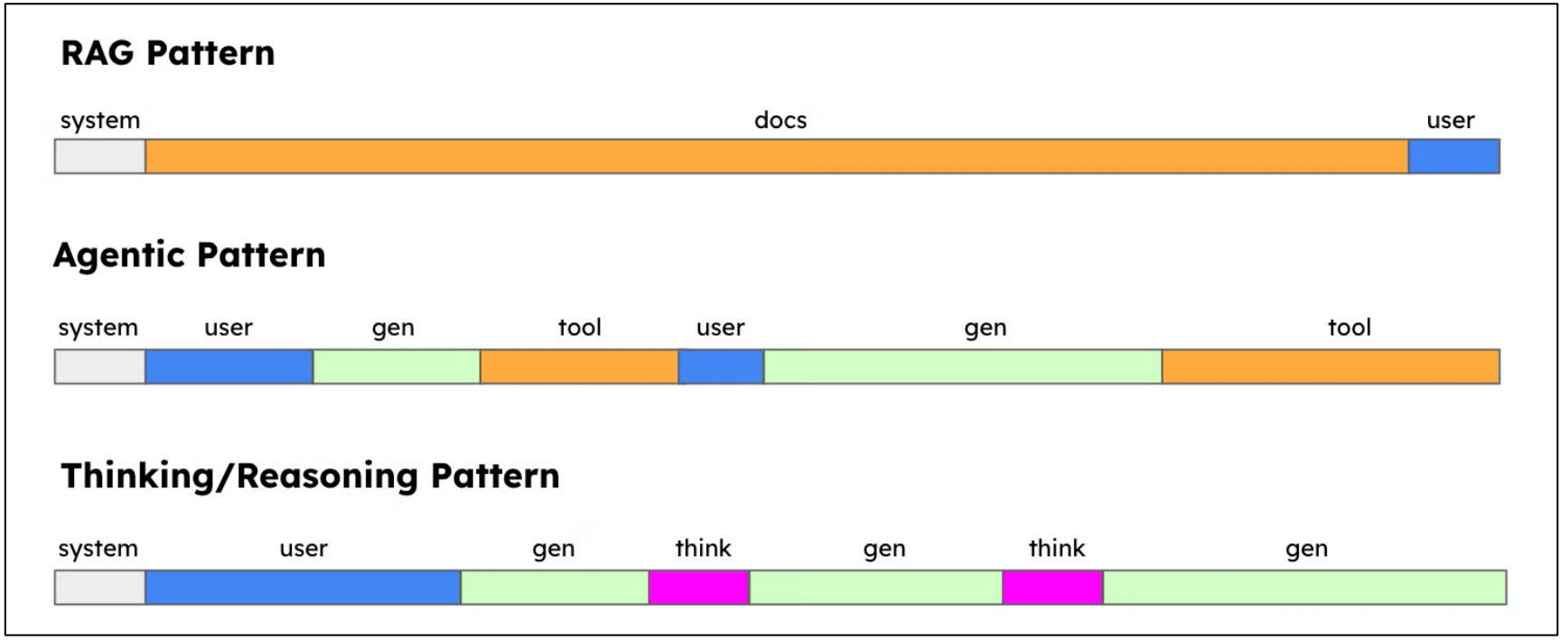
https://chat.com

Recap

What our images should do:

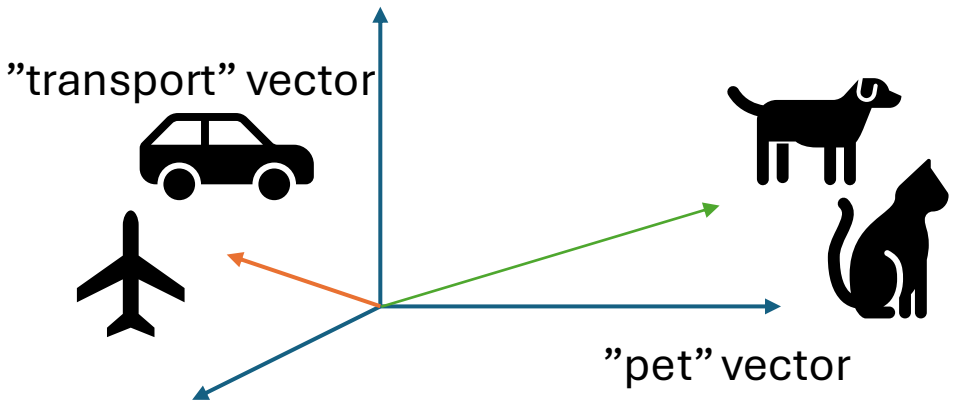


Context length usage



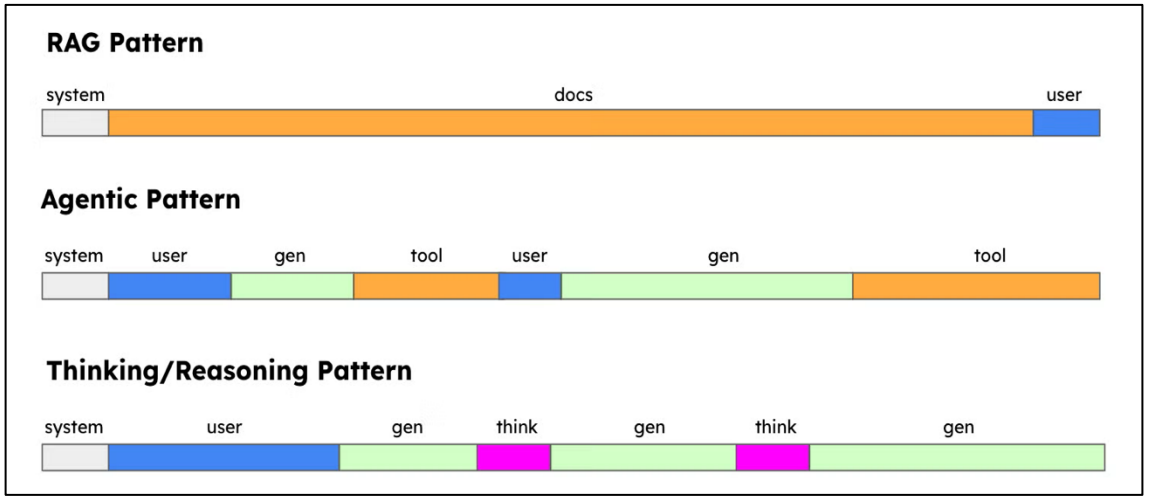
https://pytorch.org/blog/hybrid-models-as-first-class-citizens-in-vllm/

Remember: everything is linear algebra



ingest demo?

RAG - CONTEXT

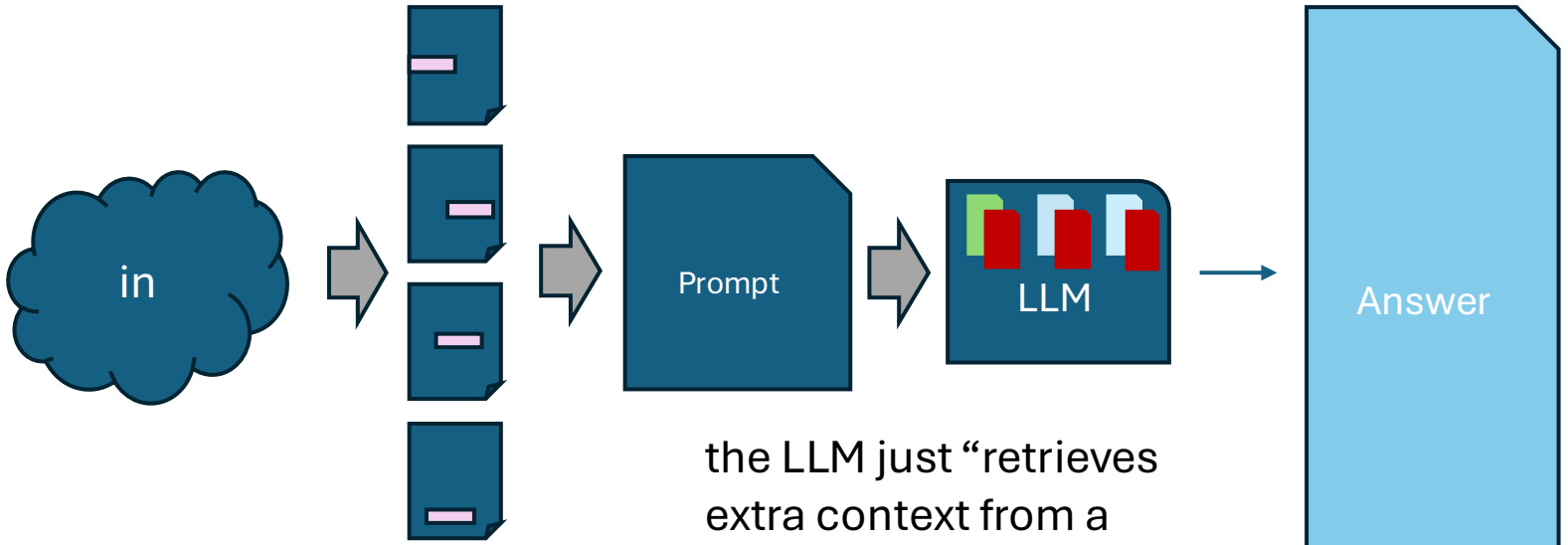


https://pytorch.org/blog/hybrid-models-as-first-class-citizens-in-vllm/

To find “sources”, “references” in databases, compute angle between prompt and avail docs!

(it’s literally called cosine similarity)

RAG (with a demo?)



SVD and low-rank updates

- any matrix A with dimension (m, n) can be decomposed as
- A = U S V.T
- U as (m, n) S as (n, n) V as (m, n)

- and we can reconstruct A without the full matrices (with some quantifiable
- information loss)

- we even have a short demo: take a picture

- Idea is: reconstruct vector spaces that have
- highly redundant data with by relying on “subspace

```python
import cv2
import numpy as np
import matplotlib.pyplot as plt

if __name__ == '__main__':
    cam = cv2.VideoCapture(0)
    _, frame = cam.read()
    cam.release()

    img = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    U, S, Vt = np.linalg.svd(img, full_matrices=True)


    k = 10
    recon = U[:, :k] @ np.diag(S[:k]) @ Vt[:k, :]
    print("matmul is", U[:, :k].shape, np.diag(S[:k]).shape, Vt[:k, :].shape)
    print("=", recon.shape)
    fig, (ax1, ax2, ax3) = plt.subplots(1, 3)
    ax1.imshow(img, cmap="grey"); ax1.set_title('Original')
    ax2.imshow(recon,cmap="grey"); ax2.set_title(f'Rank-{k}')
    ax3.imshow(np.abs(recon-img),); ax3.set_title(f'diff')

    plt.show()
```

LoRA and other finetuning approaches use a similar idea! («PeFT»)

- Training is expensive (unavailable to us for “usual-scale” data)
  - -> mem required for full-scale training ~ 2 x param + 1 x param
  - (1 x fwd, 1 x bwd, 1x state … bad approximation but it’s the OOM)

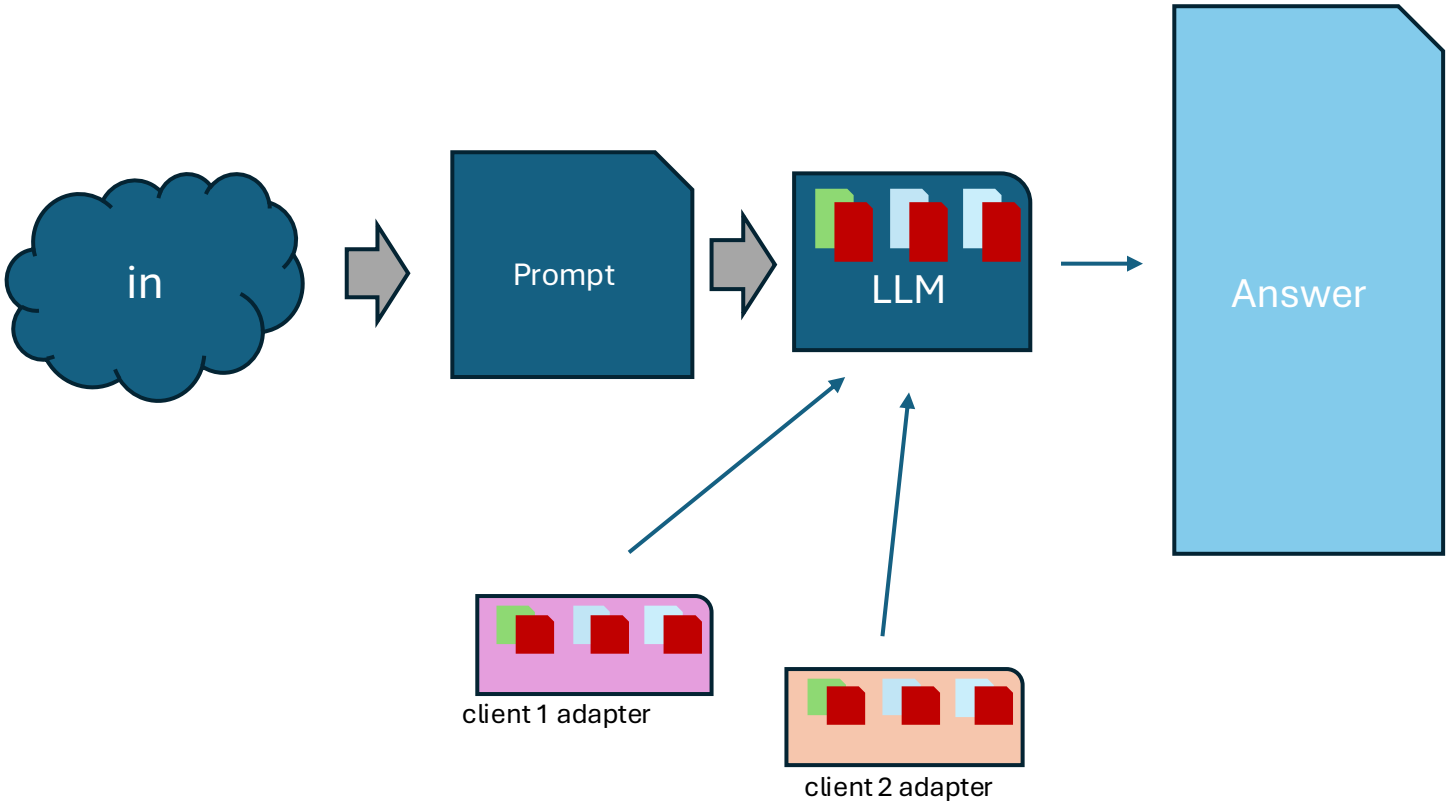
- we would also need to multiply this – DDP/FSDP – and replicate for large data

- Post-training approaches of interest here: LoRA, QLoRA

- AND: We can enhance TP with them (as they’re now small
- operators vLLM lets us plug in on the fly!)

- More difficult: RLHF and full-state posttraining training

Finetuning



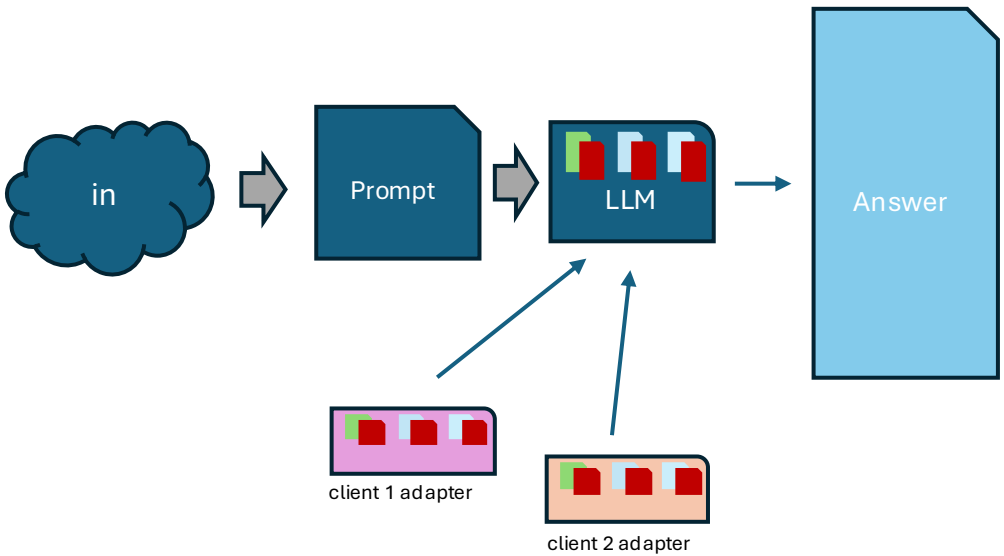
Finetuning – data and how?

```
{
  "conversations": [
    {
      "from": "system",
      "value": "You are a helpful, unbiased AI assistant."
    },
    {
      "from": "gpt",
      "value": "Hi, welcome to this interaction! I can help with your Go programming and Godoc related inquiries."
    },
    {
```

quick demo, easy serving?

finetuning is not trivial:

- data curation
- quantization
- model behavior in finetuning
- precision requirements?



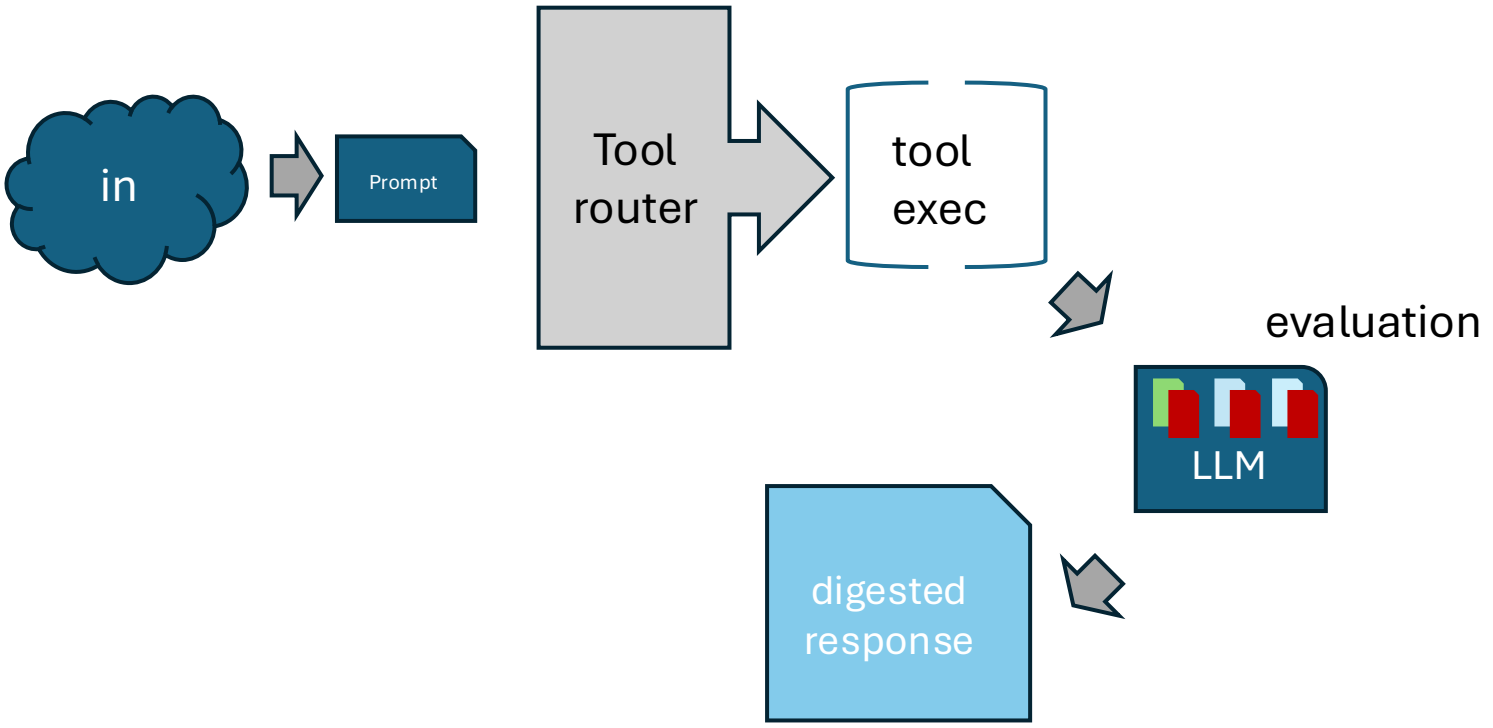
Agents

- “tool use”, “MCP” …?

- Lots of tools and ideas to bring “programmable AI” safely to life.

- Most are just protocol ideas how to structure the model’s interaction with
- some kind of environment.

Agents


in
Prompt
Tool
router
tool
exec
evaluation
LLM
digested
response

More use cases: Generative and other

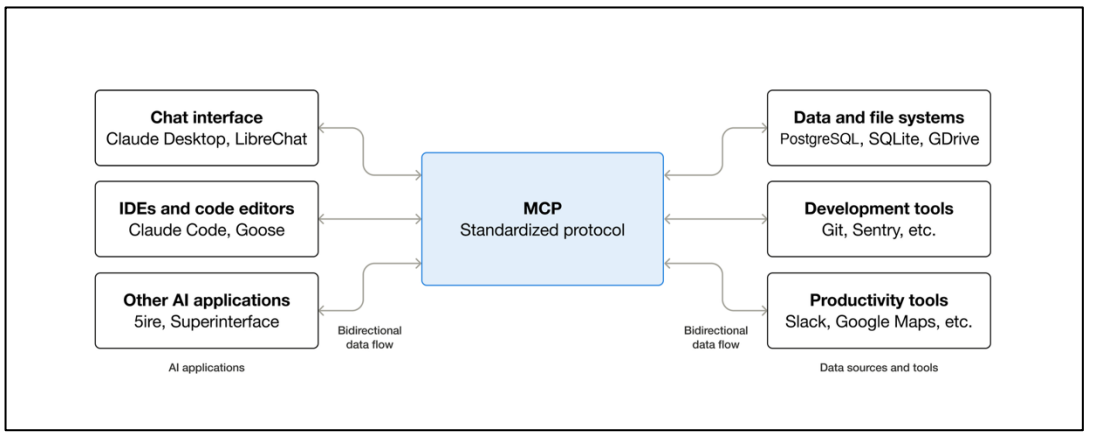
- models: base, instruct, reasoning
- distillation
- compression
- RLHF
- RL environments

Get started

What is the Model Context Protocol (MCP)?

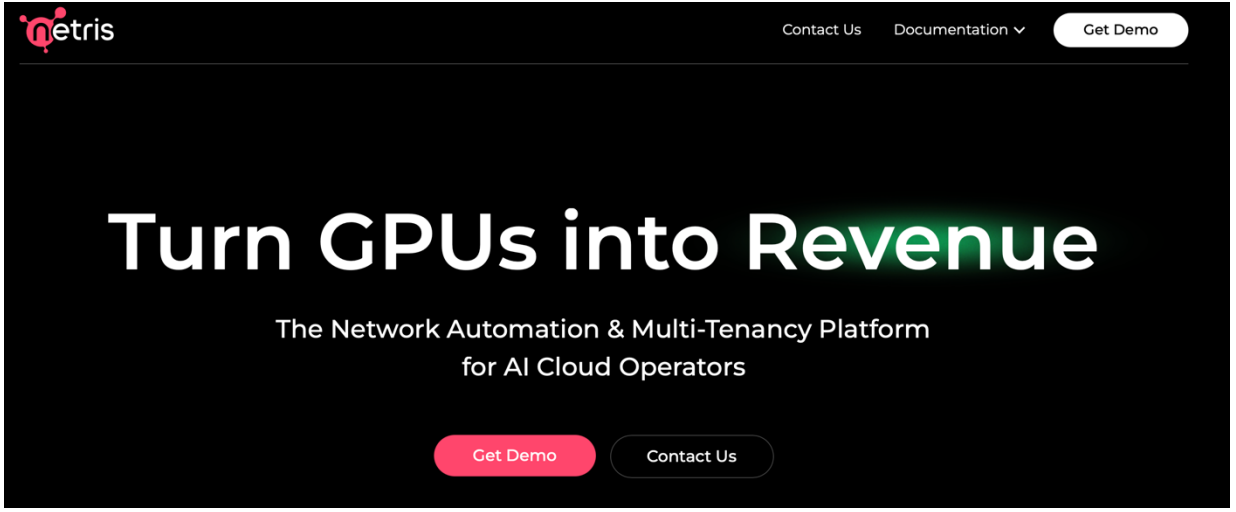
Copy page

MCP (Model Context Protocol) is an open-source standard for connecting AI applications to external systems.



Top and bottom:
https://modelcontextprotocol.io/docs/getting-started/intro

More use cases: Generative and other



https://netris.io

Takeaways

- systems engineering is where it’s at
- we have a mostly solved setup for most LLM-related applications
- quantization can bite us with brittle tooling
- ”agents” are just <smartly> chained prompts within controlled environments

Fragen?

Links

- https://www.stepping-stone.ch/
- https://www.stoney-backup.com/
- https://www.stoney-cloud.com/
- https://www.stoney-mail.com/
- https://www.stoney-meet.com/
- https://www.stoney-office.com/
- https://www.stoney-services.com/
- https://www.stoney-storage.com/
- https://www.stoney-wiki.com/

stepping stone AG
Wasserwerkgasse 7
CH-3011 Bern

Telefon: +41 31 332 53 63
www.stepping-stone.ch
info@stepping-stone.ch