Efficient Long-context LLM Serving + Physical Estimation in VLMs

Pankaj Pansari

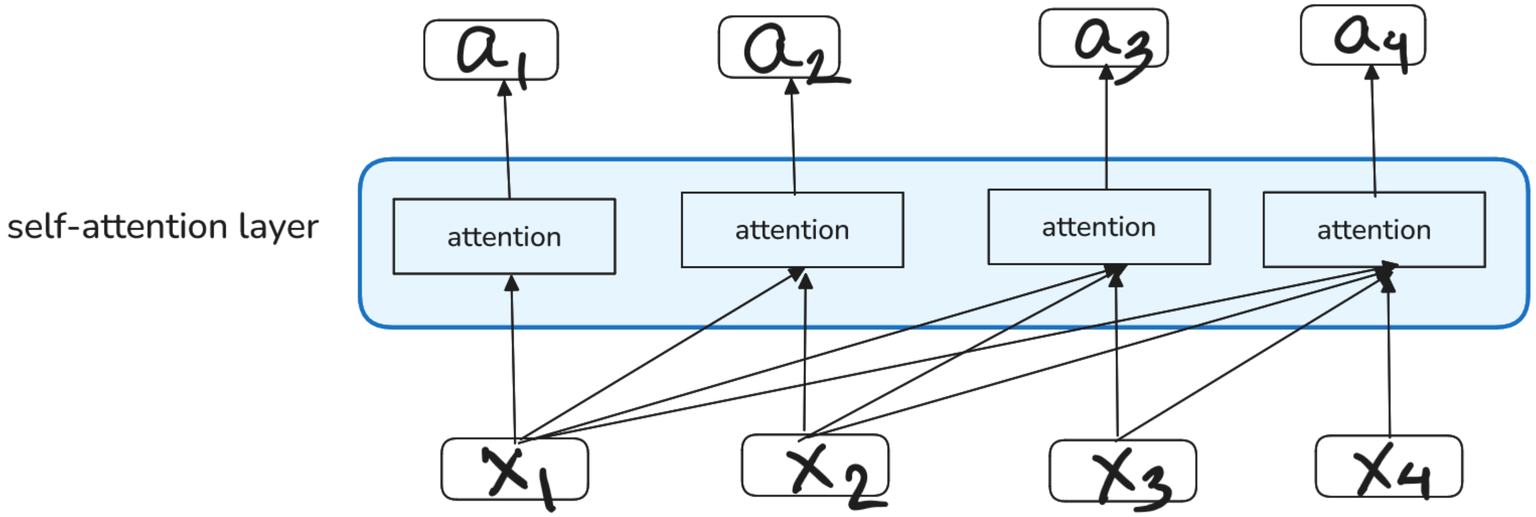
Plaksha University

25 Feb, 2026

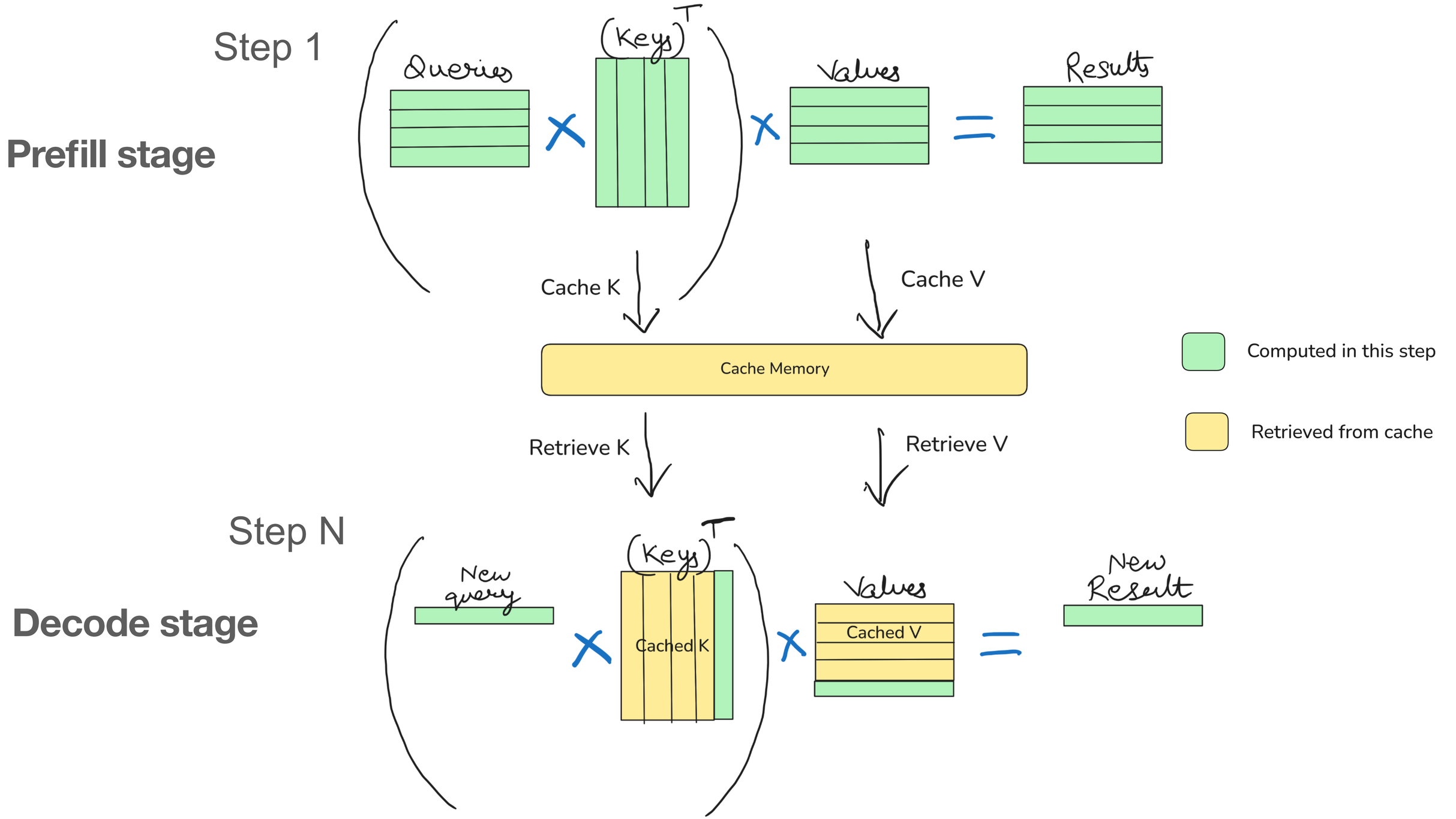
Efficient Long-context KV Cache Management

With Arbaaz Shafiq and Mukundan Guruswamy (Plaksha University)

KV Cache - Recap



- Autoregressive model : a token only attends to previous tokens during dot-product attention
- Once query, key, value vectors of a token get computed, they don’t change. Wasteful to recompute!
- Store key, value of every token in kv-cache in memory

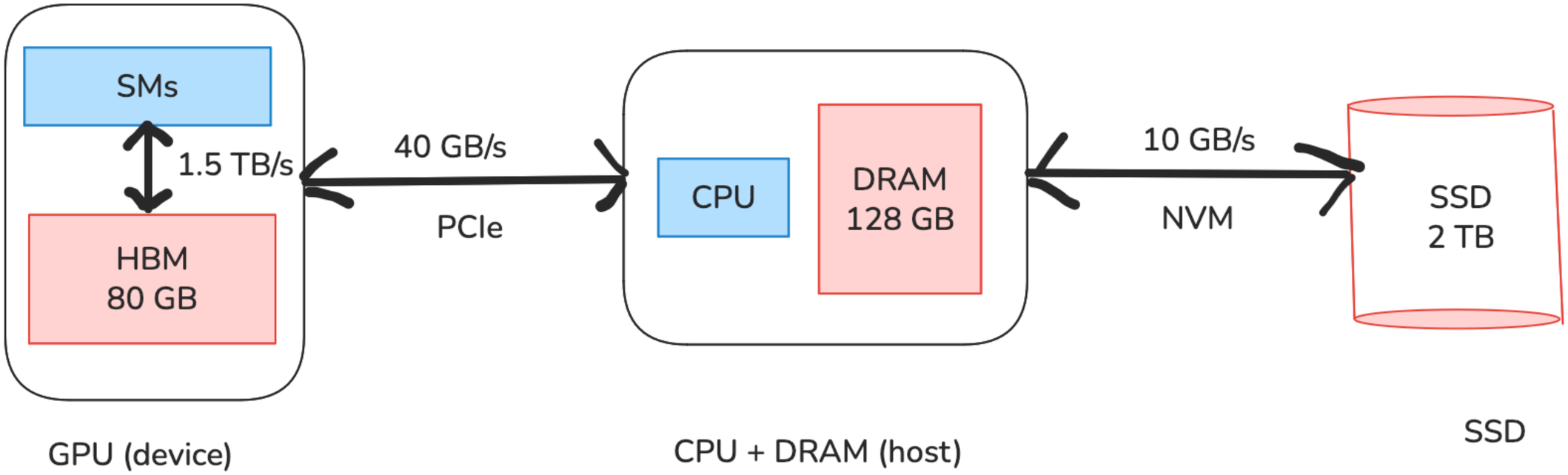

Step 1
Prefill stage
Queries
(Keys)^T
Values
Results
Cache K
Cache V
Cache Memory
Retrieve K
Retrieve V
Computed in this step
Retrieved from cache
Step N
Decode stage
New query
(Keys)^T
Cached K
Values
Cached V
New Result

KV Caches and Offloading

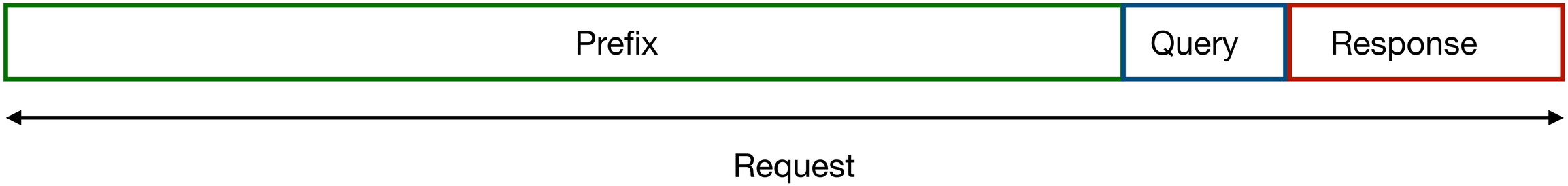
Model	Context Len	KV cache per layer	KV cache all layers (1 request)	Maximum concurrency
Llama-3.1-8B	128k	0.52 GB	16.77 GB	3
Gemma-7B	8k	0.10 GB	2.82 GB	23
Qwen-7B	16k	0.26 GB	8.39 GB	7
Qwen-14B	16k	0.38 GB	13.10 GB	3

KV cache sizes for long context. Assuming 80GB GPU HBM. FP16 for KVs

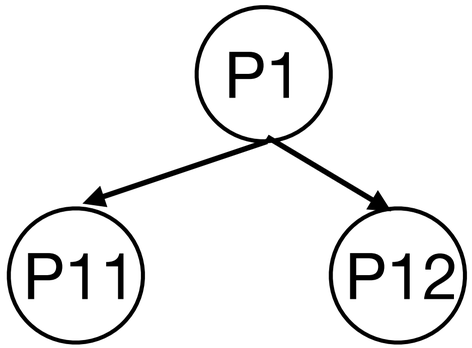
Multi-tier Storage

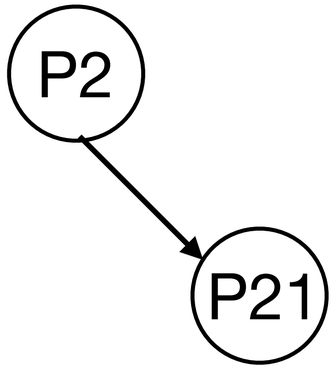


Terminology

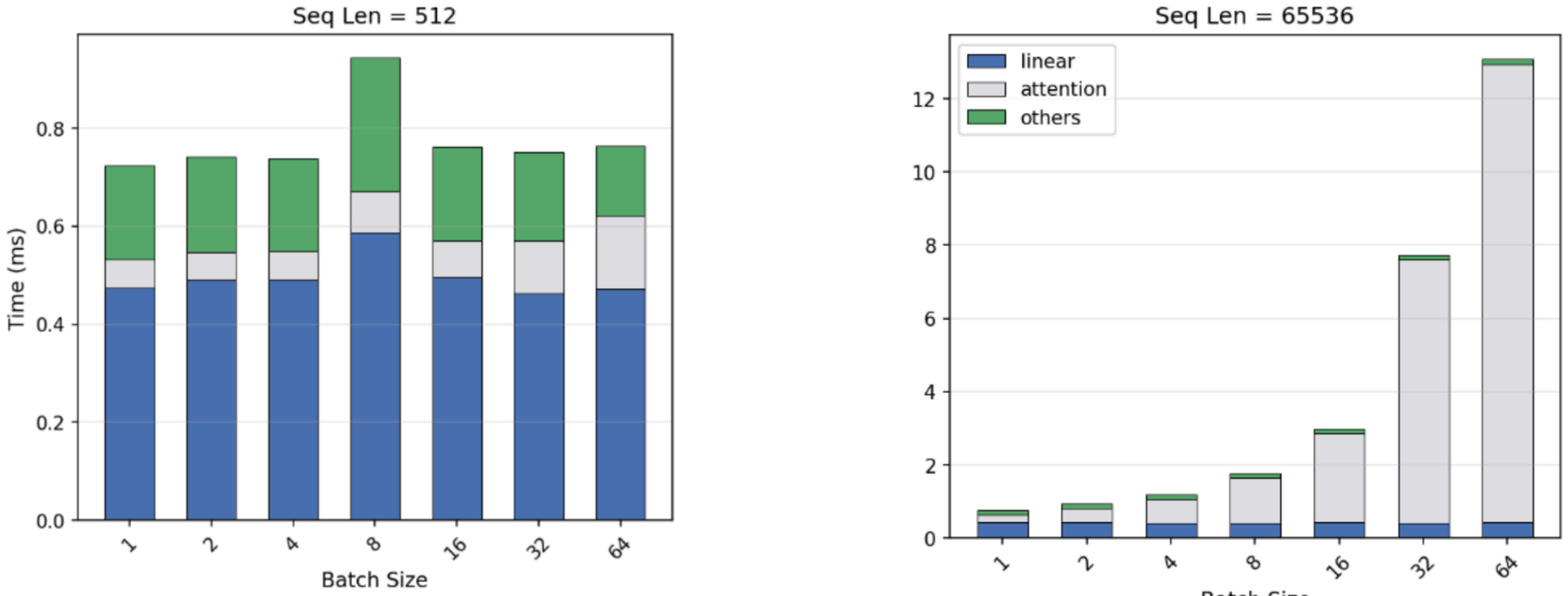


- In general, set of all prefixes form a forest





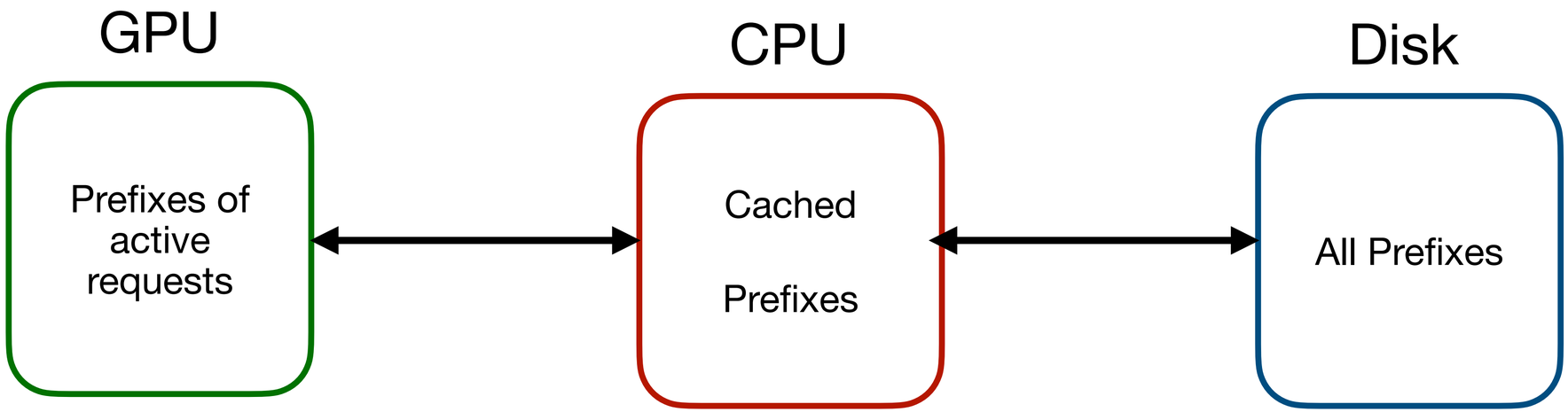
Decoding in Long vs Short Contexts



Decode time breakdown Llama-3.1-8B (1 layer)

- Small context: Linear layers dominate -> batching good
- Long context: Self-attention dominates -> batching not helpful
- **Note**: Attention kernel cannot be batched since each request has own KV cache

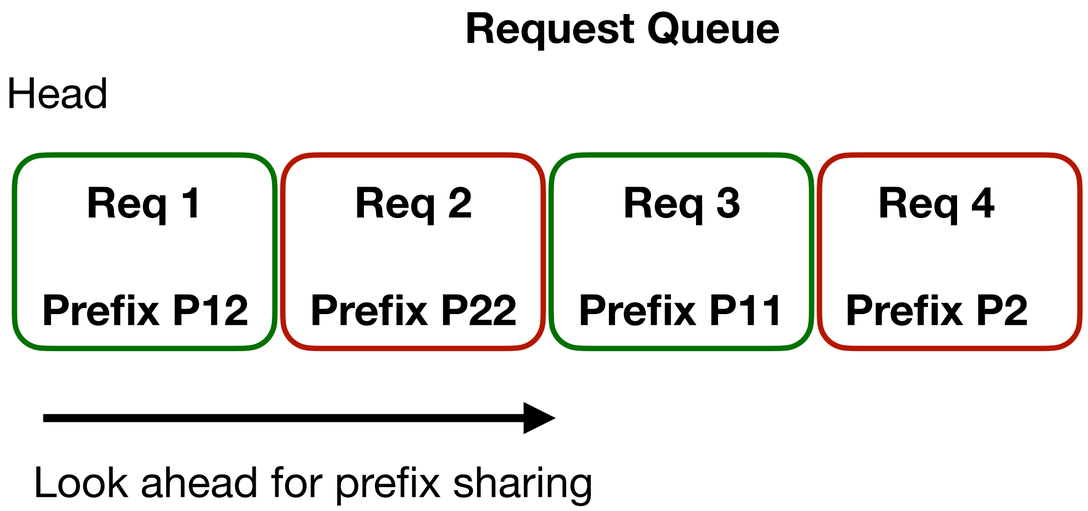
Memory Manager



Memory manager decides which the set of prefixes present on GPU and CPU. Disk stores all prefixes

Scheduler

- Three levels of scheduling:
  - **Request-level**: Which set of requests should execute next on GPU?

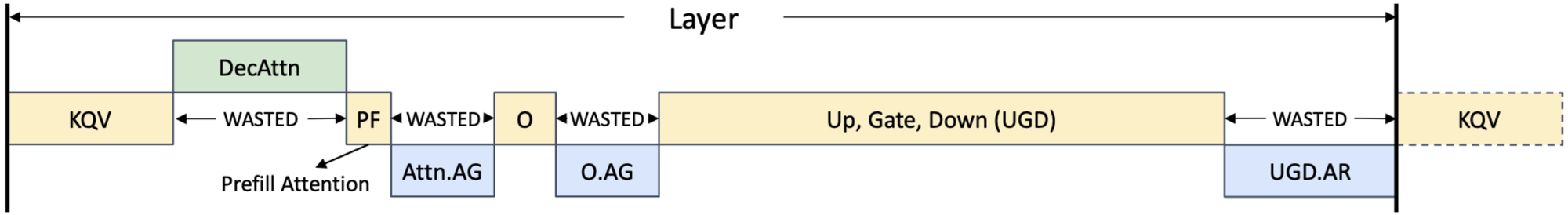


Scheduler

- Three levels of scheduling:
  - Request-level: Which set of requests should execute next on GPU?
  - **Iteration-level**: Which active request(s) on GPU should run in next iteration?
    - Batch schedule decode iterations sharing prefixes [PagedAttention (Kwon et al., 2023)]
    - Round-robin among different prefixes (using deadline slack [Niyama (Goel et al., 2025)]

Scheduler

- Three levels of scheduling:
  - Request-level: Which set of requests should execute next on GPU?
  - Iteration-level: Which active request(s) on GPU should run in next iteration?
  - **Operator-level**: How to schedule operations such that GPU execution cores are not idle?



Execution pipeline of one layer of LLM inference (naive)

Ref - *NanoFlow: Towards Optimal Large Language Model Serving Throughput,* Zhu et al., 2025

Scheduler <> Memory Manager

- Scheduler talks to the memory manager during request scheduling
  - Which prefixes are already present on GPU HBM? Gang-schedule
  - Out-of-order scheduling; needs care to prevent starvation
- Memory manager gets hints from scheduler to prefetch prefixes from disk to SSD to GPU

Benchmarking and Augmenting Visual Estimation in VLMs

With Varchita Lalwani, Pratham Arora (Plaksha University)

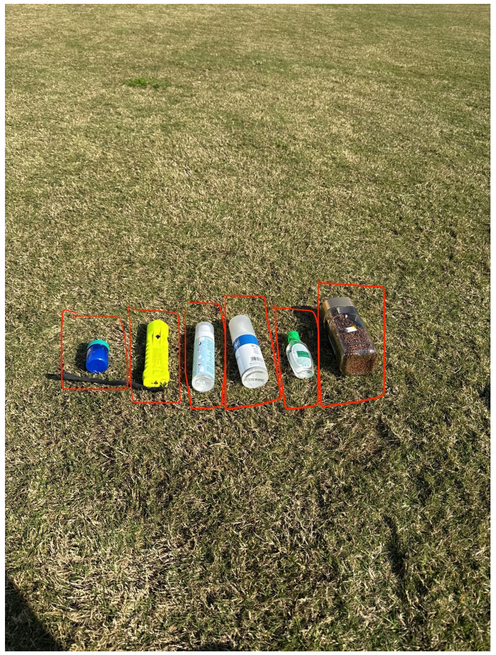
Motivation

- We implicitly make visual estimations all the time -> important to navigate the world
- Existing benchmarks:
  - focusing heavily on distances, sizes, and relative positions - SpatialVLM (Chen et al., 2024), SpatialRGPT (Cheng et al., 2024), Q-SpatialBench (Liao et al., 2024)
  - More diverse set of properties but
    - qualitative [PhysBench (Chow et al., ICLR 2025)]
    - video-based and focused on kinematics [QuantiPhy (Li et al., 2025)]
    - synthetic images with issues in sim-to-real transfer [Physics Context Builders (Balazadeh et al., ICCV 2025)]

Benchmark Dataset

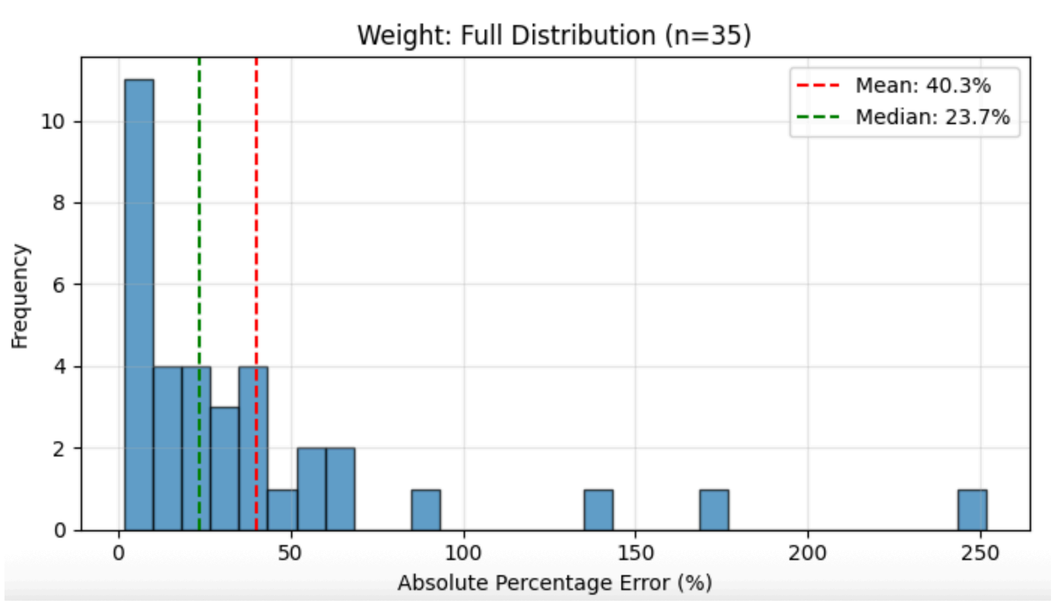
- 5 kinds of estimation tasks of static properties: weight, volume, angle, geometric fit, and stability (WIP: ~40/100 images for each)
- Real-world hand-collected images. Precise measurements for weight, angle, and volume. Stability and geometric fit are binary for now.
- Not adversarial, but decently challenging for humans
- An image rarely shows complete information to faithfully answer such questions; humans use a rich body of cues acquired over a lifetime of interacting with the world

Weight Estimation



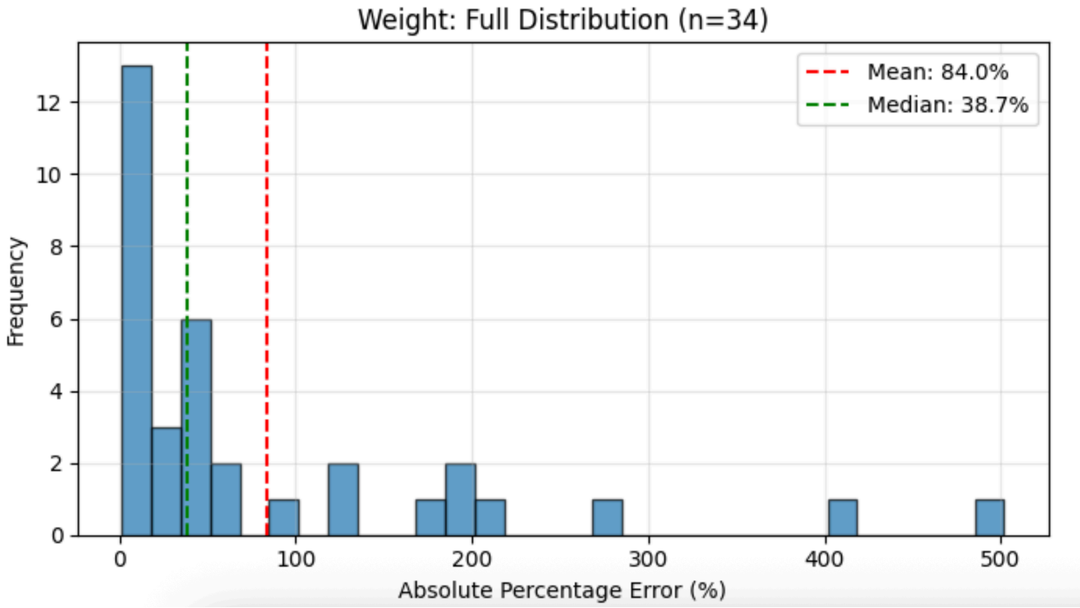
Large range of weights - ~100g - ~7kg

Needs material identification + density reasoning



Gemini 3-Pro

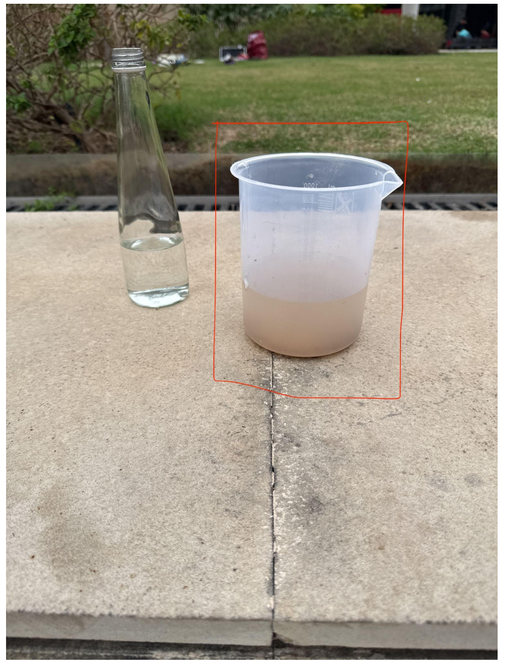
(High reasoning effort)



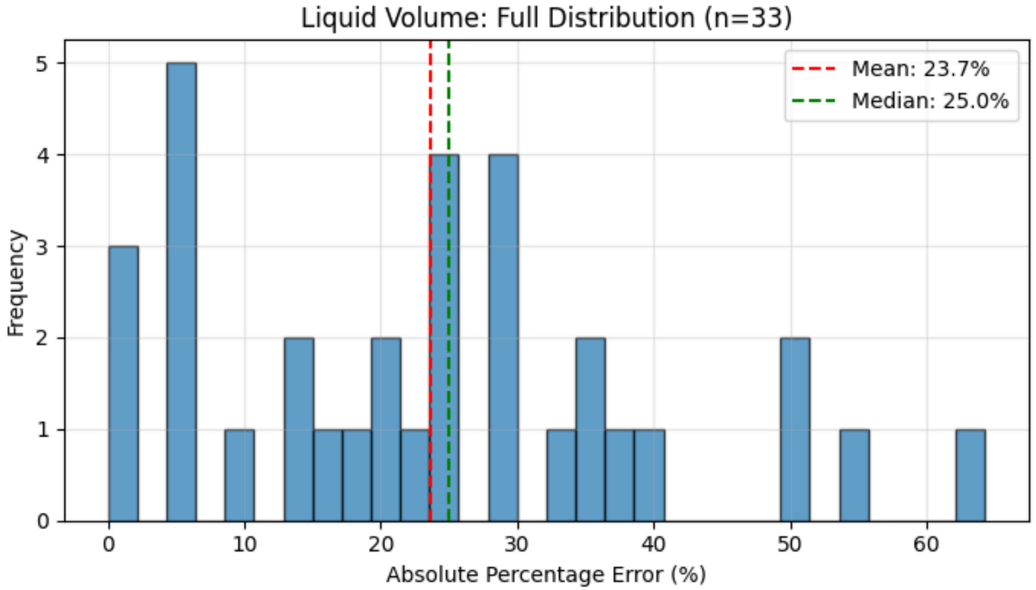
GPT-5.2

(High reasoning effort)

Volume Estimation

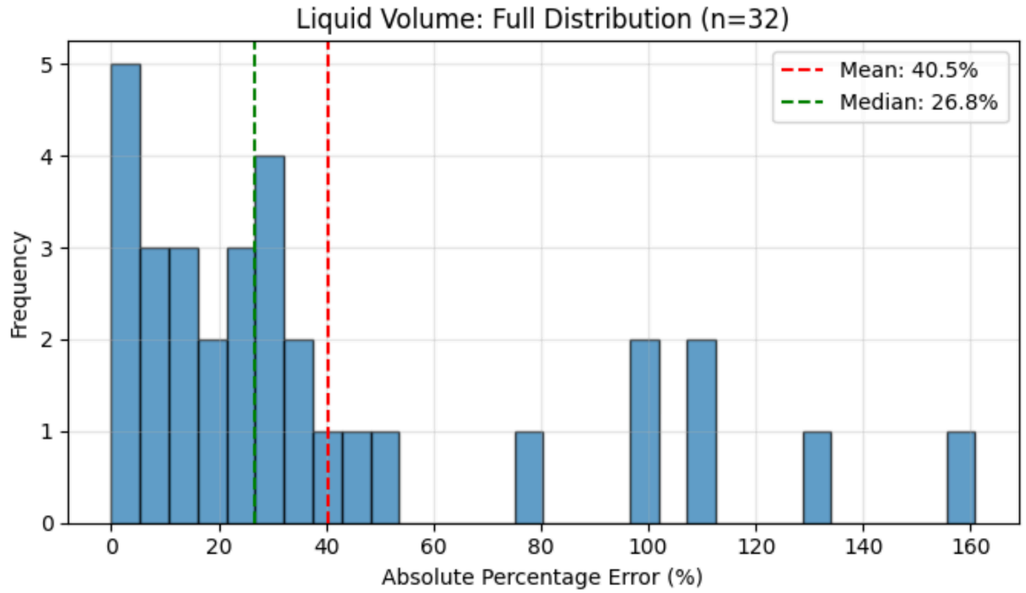


Needs reasoning of internal container geometry + fill-level estimation



Gemini 3-Pro

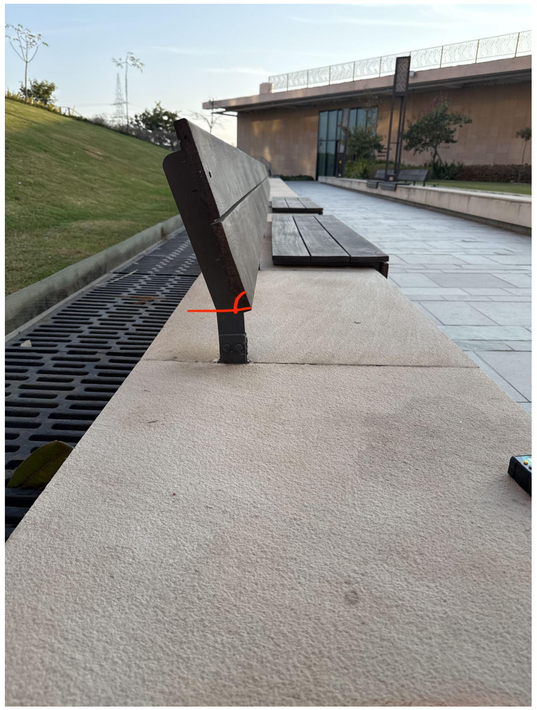
(High reasoning effort)



GPT-5.2

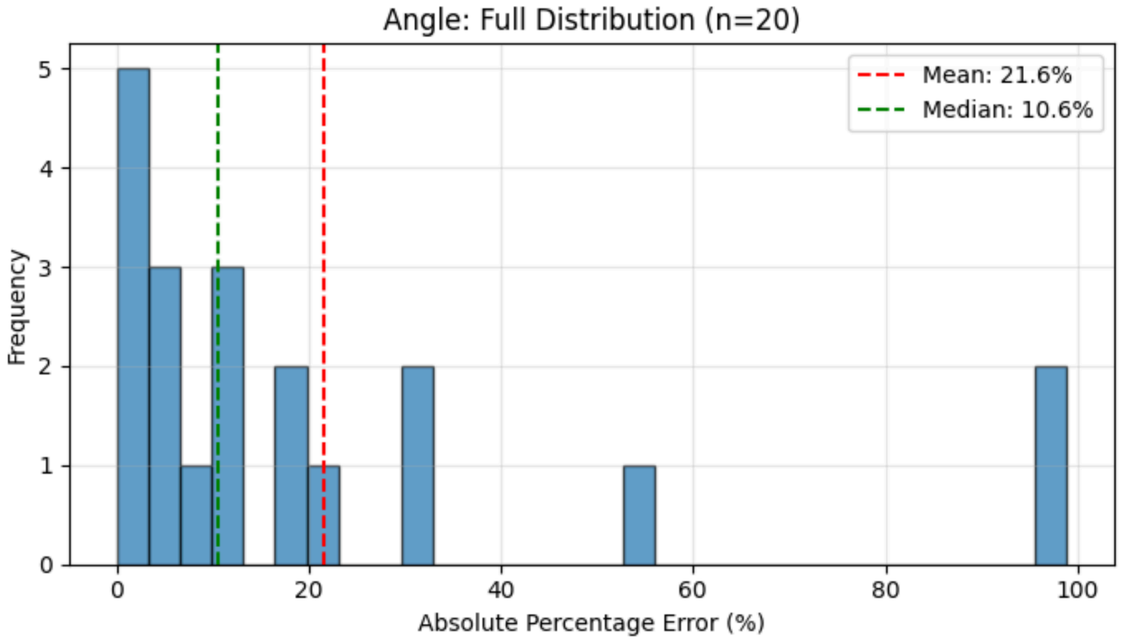
(High reasoning effort)

Angle Estimation



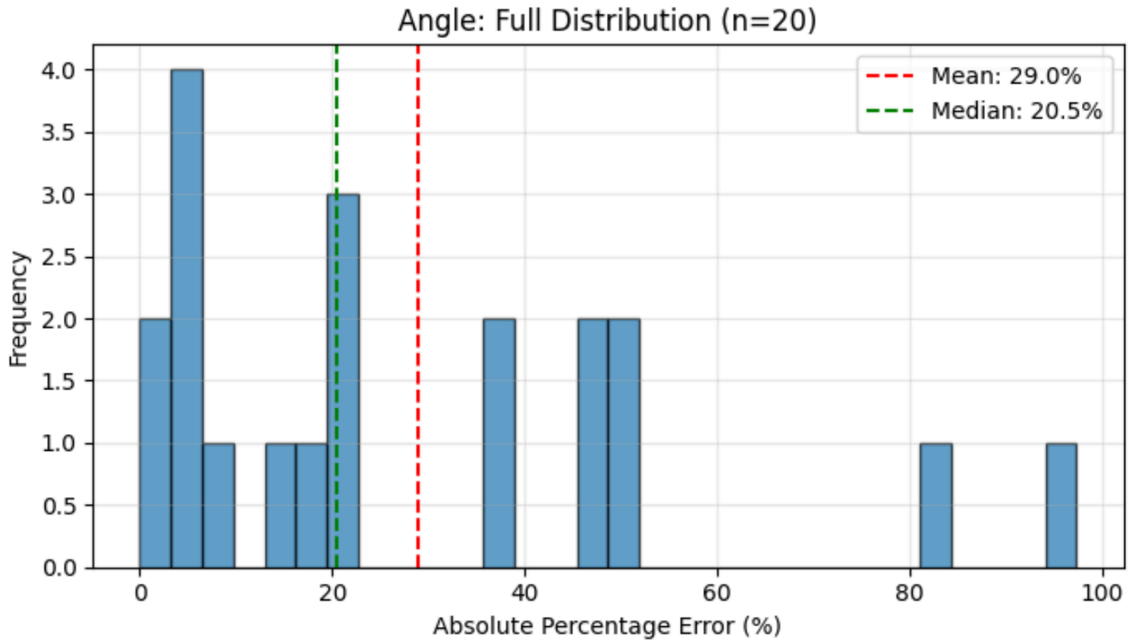
Geometric, local contact geometry

Angle Estimation



Gemini 3-Pro

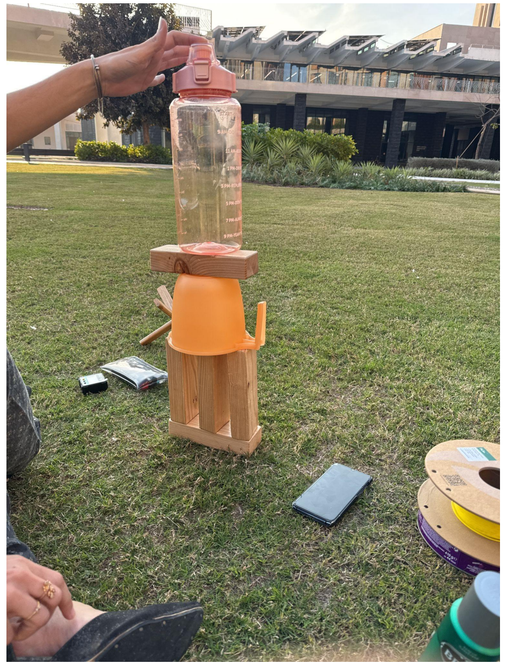
(High reasoning effort)

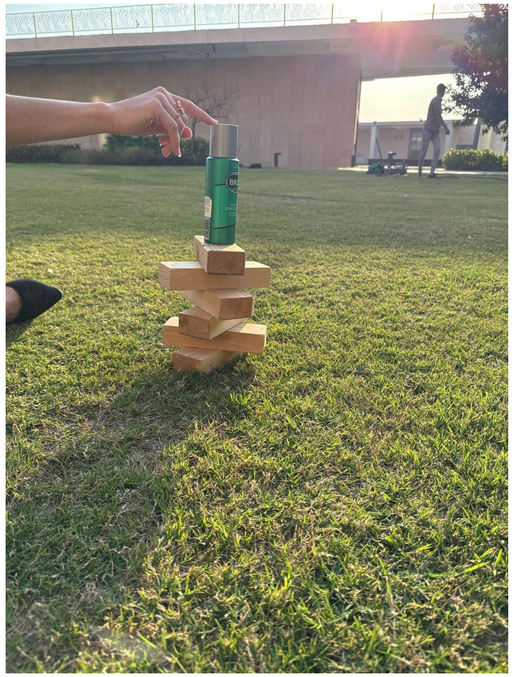


GPT-5.2

(High reasoning effort)

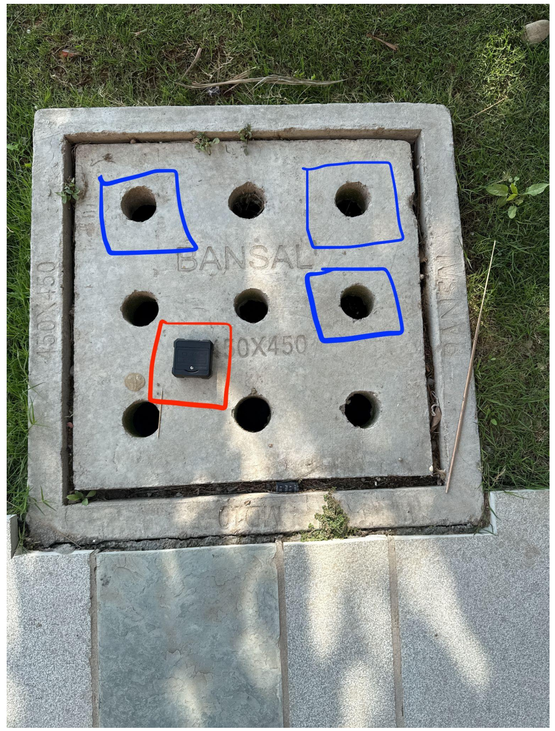
Stability Estimation

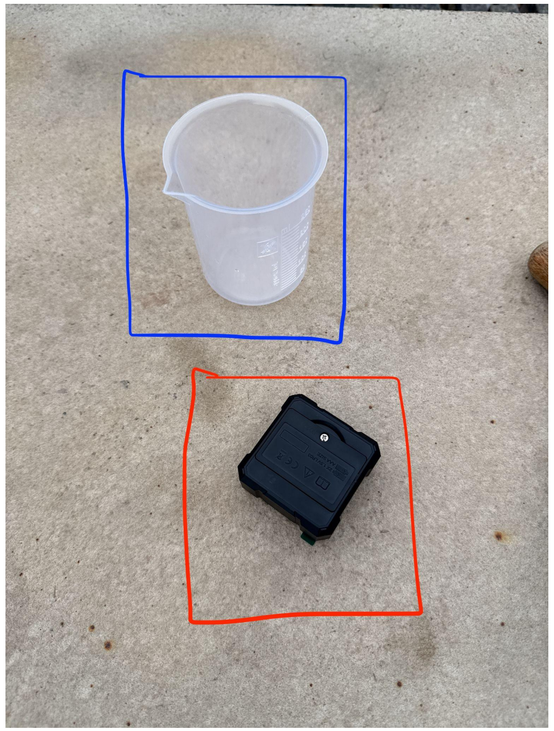



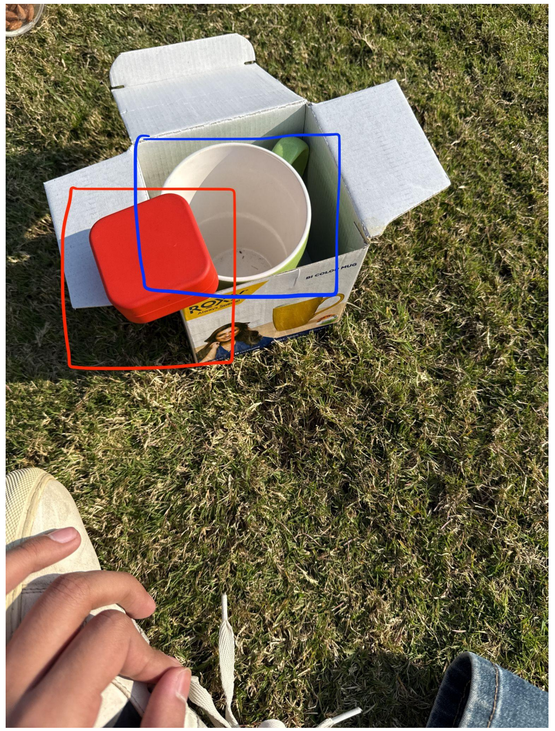


- Challenging to estimate stability of configuration from an image
- Accuracy over ~20 samples (balanced): 0.48 (Gemini 3-Pro), 0.48 (GPT-5.2)

Geometric Fit Estimation







- Close to spatial reasoning, but needs compositional judgment
- Accuracy over ~20 samples (balanced): 0.76 (Gemini 3-Pro), 0.48 (GPT-5.2)

Disentangling reasoning and grounding (Planned)

Base run with just image and questions (no tools)

Augmented run -> prepend structured context

- Structured context = Processed model outputs
  - Bounding boxes + class labels (Grounding-DINO, OWLv2)
  - Metric depth map (DepthAnything, UniDepth) + camera intrinsics
  - Density lookup table (weight); segmentation mask (volume); friction coefficients (stability)

Delta between 2 runs disentangles the 2 abilities

Thank You