MorphServe: Efficient and Workload-Aware LLM Serving via Runtime Quantized Layer Swapping and KV Cache Resizing

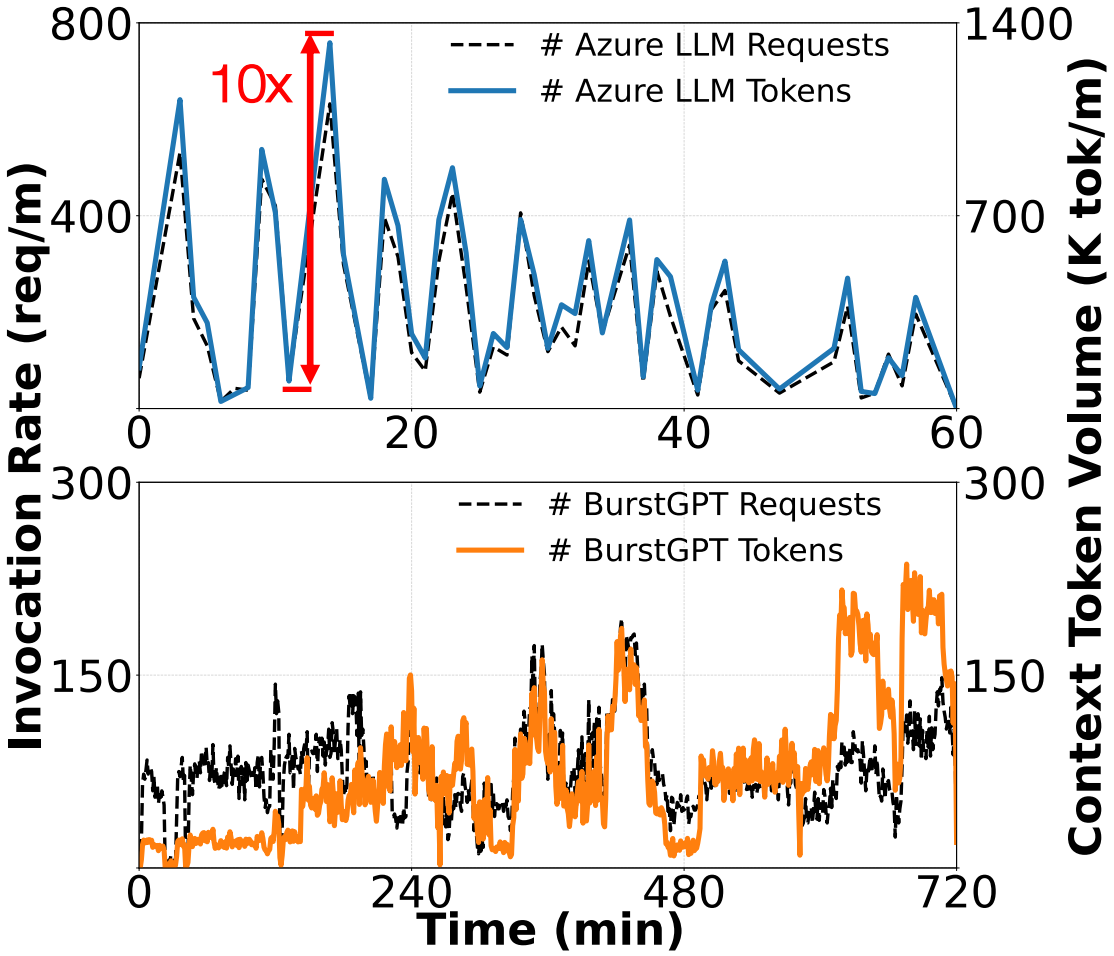
Zhaoyuan Su[1], Zeyu Zhang[1], Tingfeng Lan[1], Zirui Wang[1]
Haiying Shen[1], Juncheng Yang[2], Yue Cheng[1]

[1]

HARVARD UNIVERSITY [2]

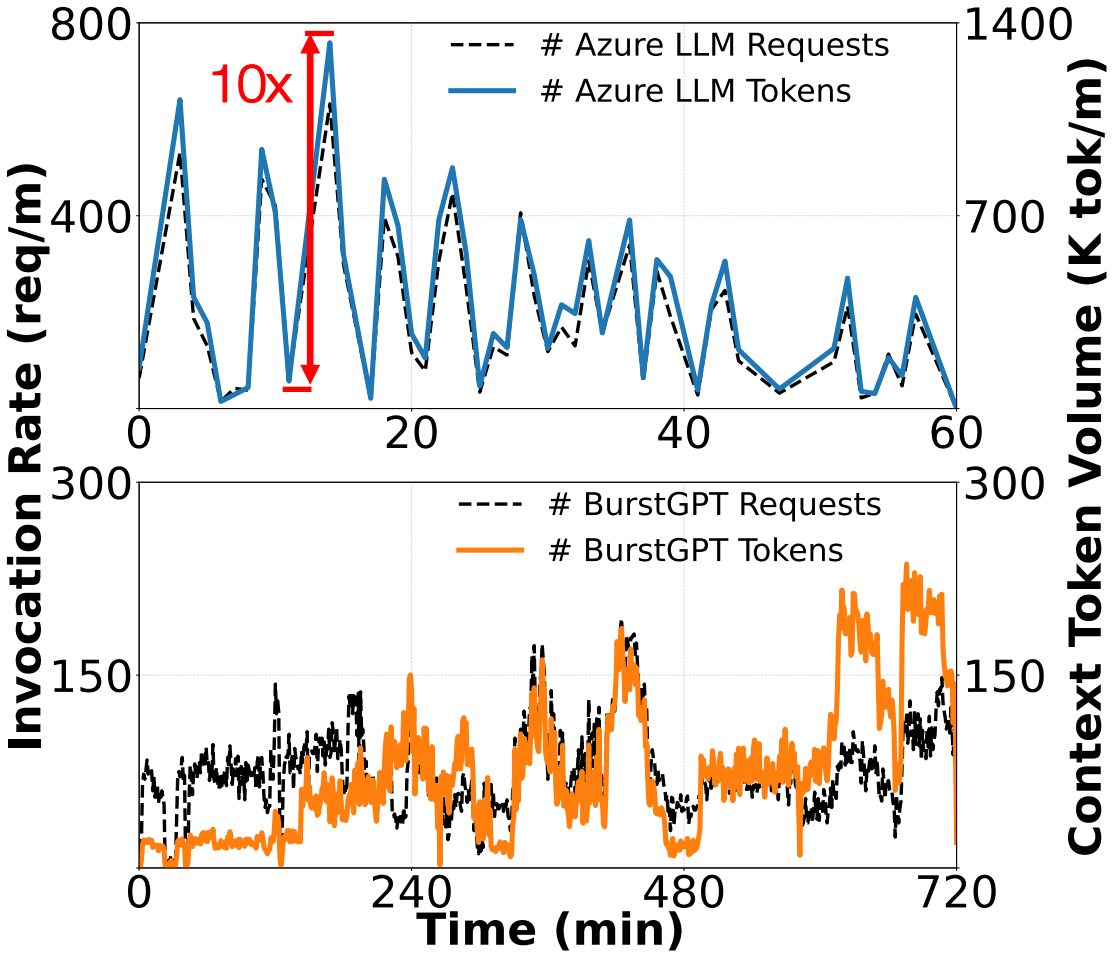
Motivation 1: Real-world LLM workloads are highly dynamic

- Bursty in **requests** and **tokens**, **10x** spikes within minutes



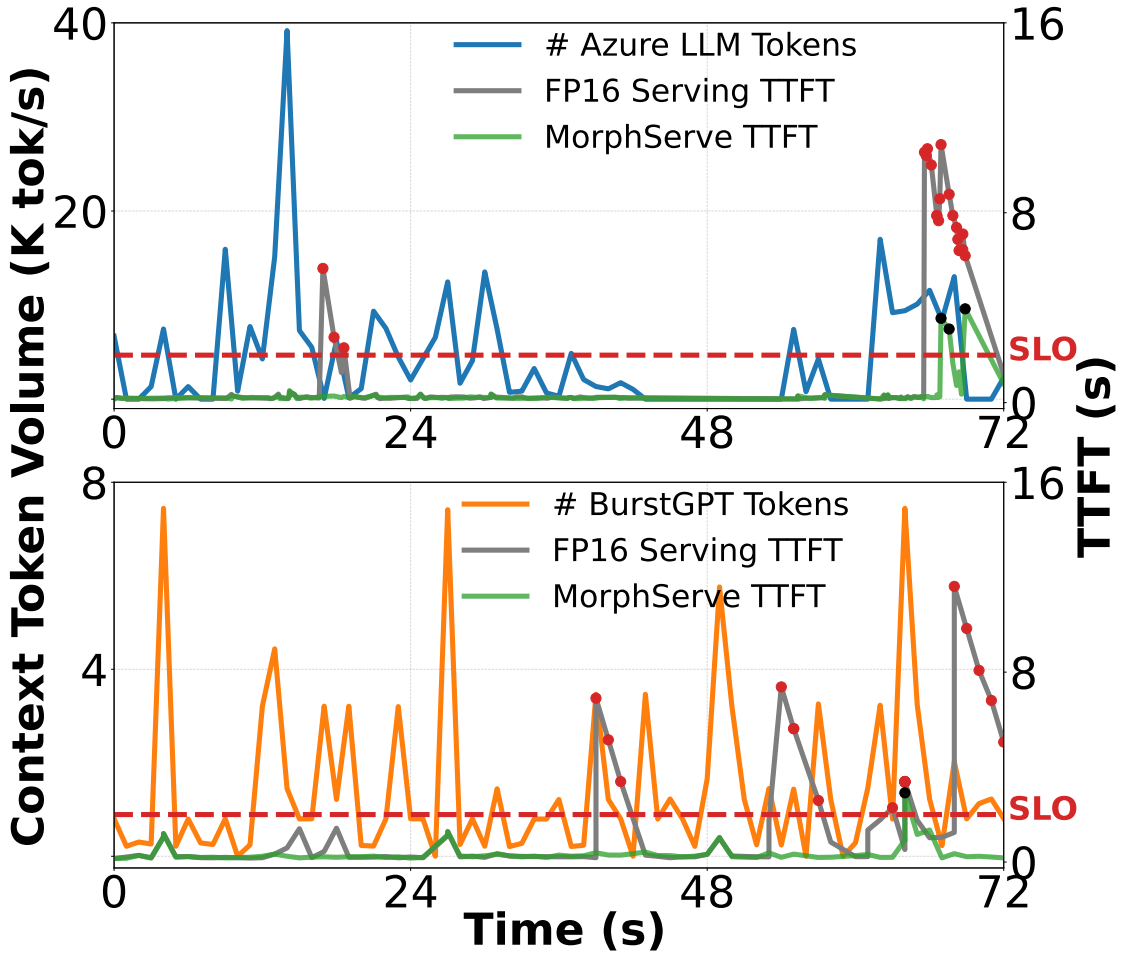
Motivation 1: Real-world LLM workloads are highly dynamic

- Bursty in **requests** and **tokens**, **10x** spikes within minutes
- Existing serving frameworks overlook **runtime workload variability** with a static config



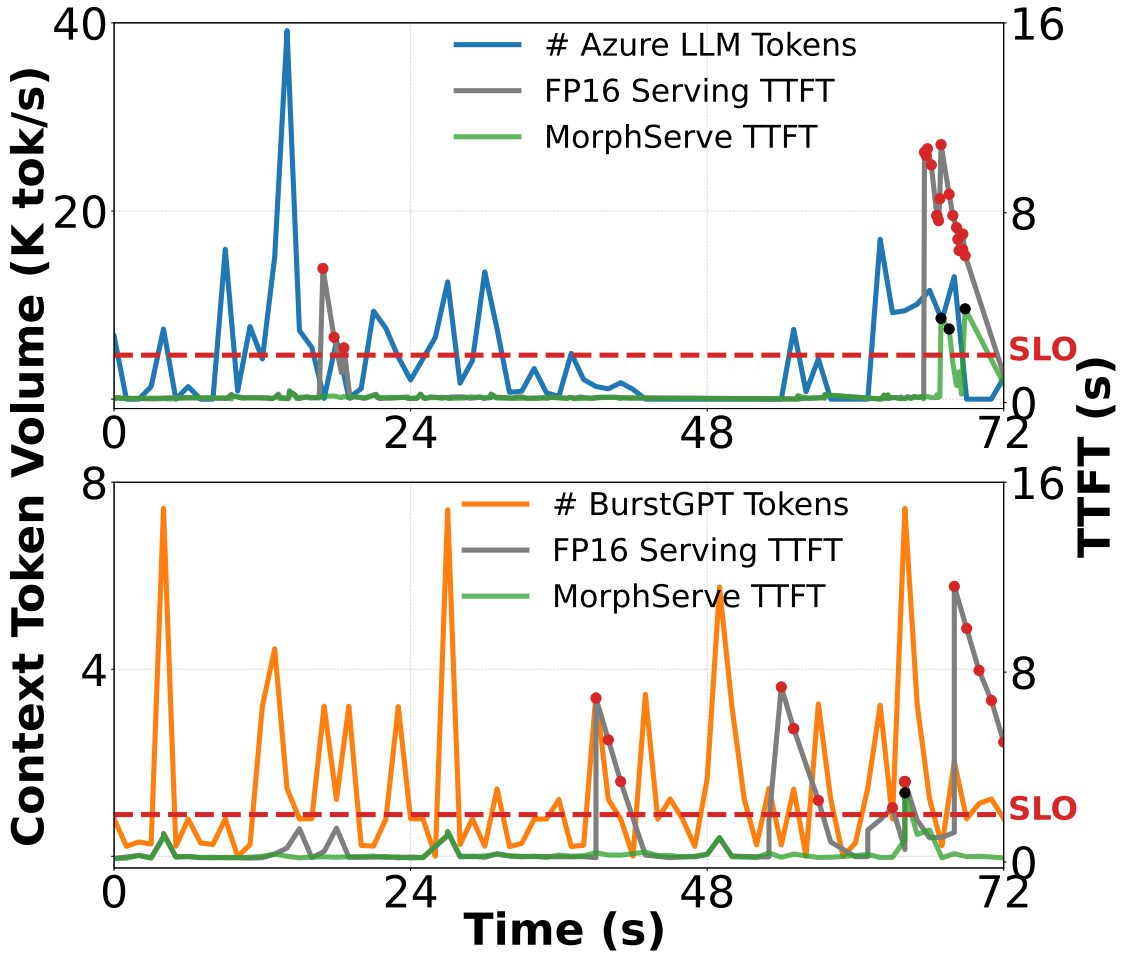
Motivation 2: Request bursts cause SLO violations

- Full-precision serving offers high accuracy



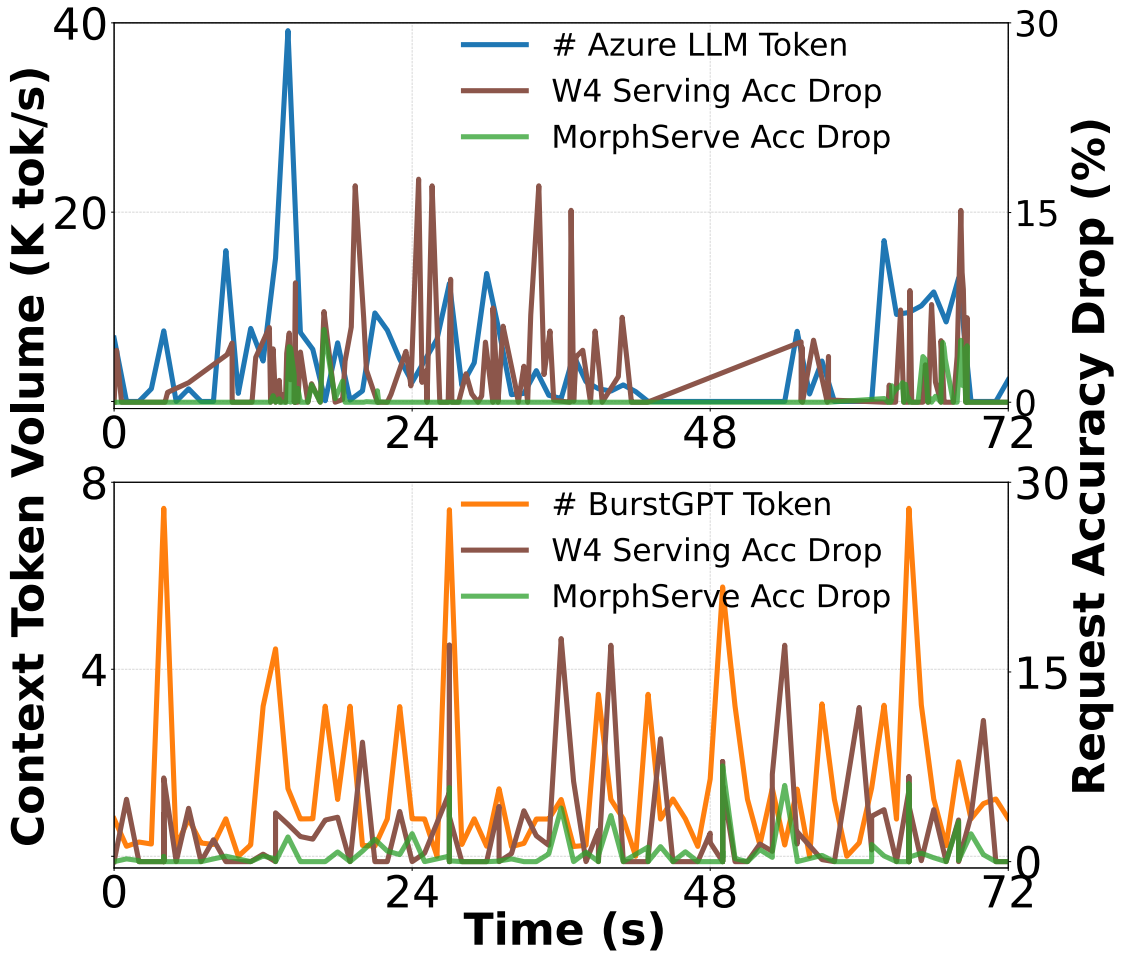
Motivation 2: Request bursts cause SLO violations

- Full-precision serving offers **high accuracy**
- Bursts may **overload** GPU computation / memory, leading to **SLO violations**



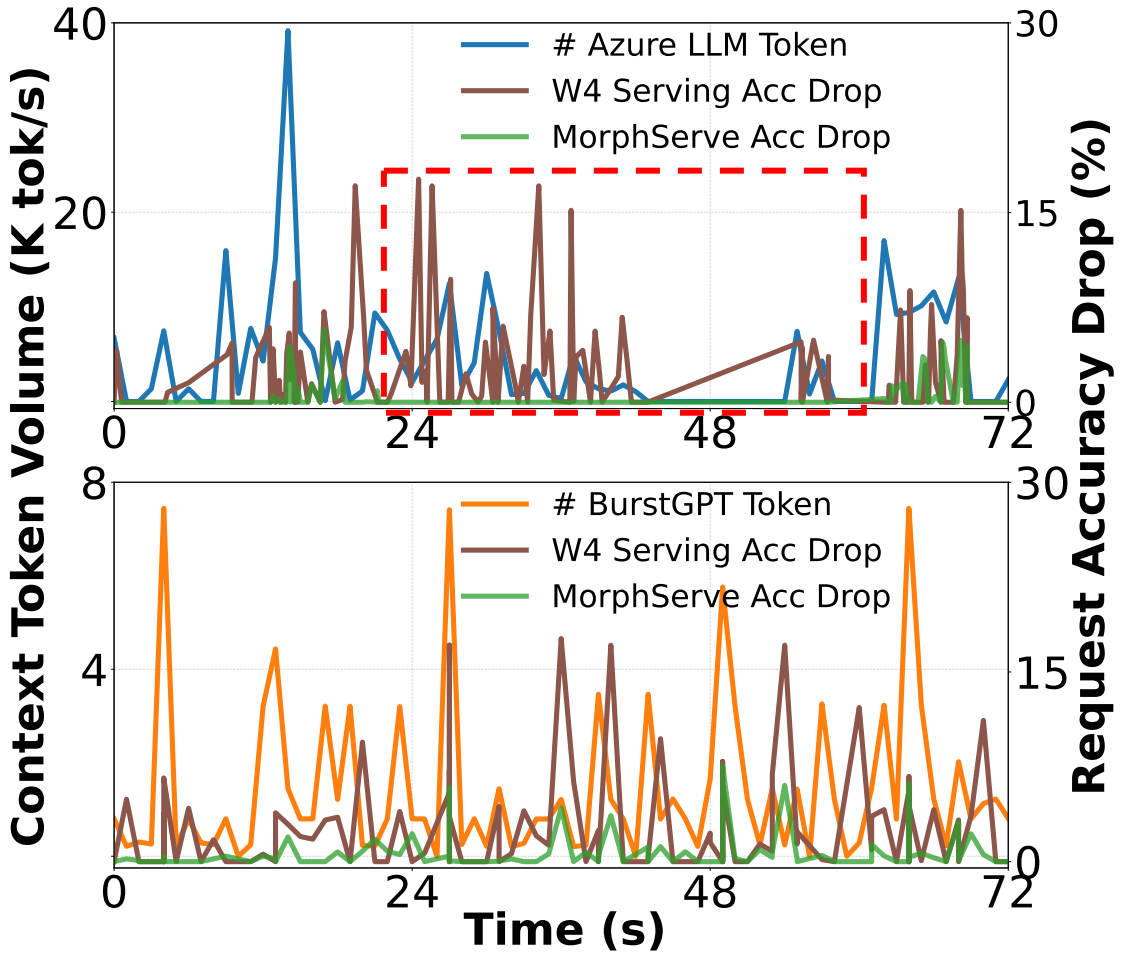
Motivation 3: Static quantization permanently hurts accuracy

- Quantization reduces memory and accelerates computation

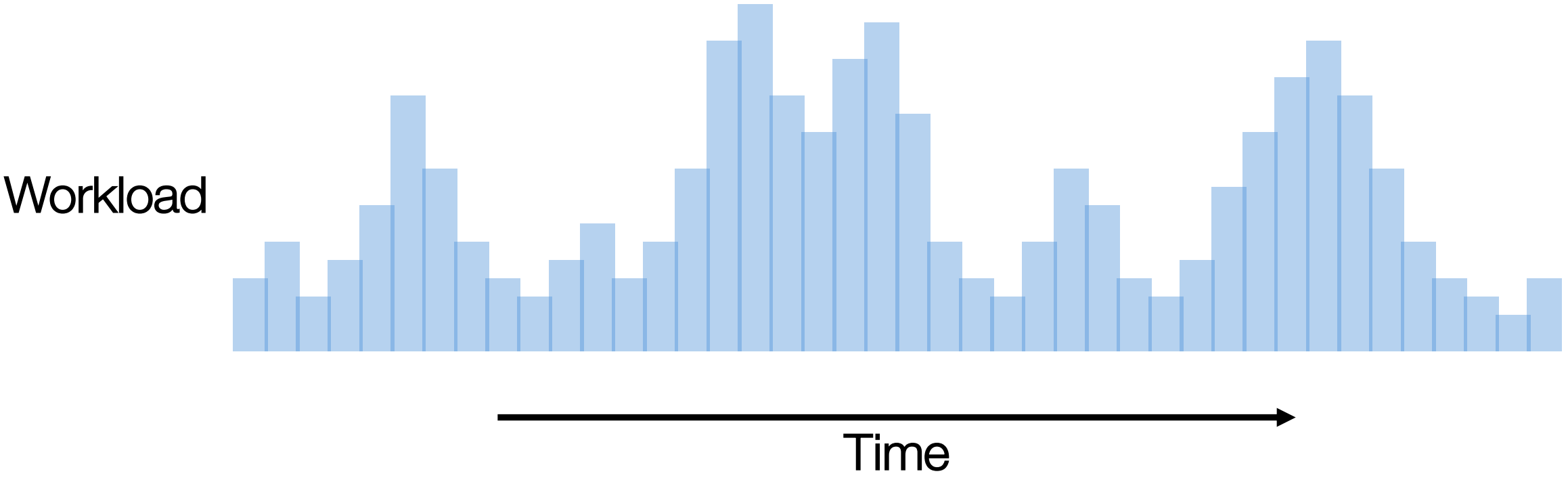


Motivation 3: Static quantization permanently hurts accuracy

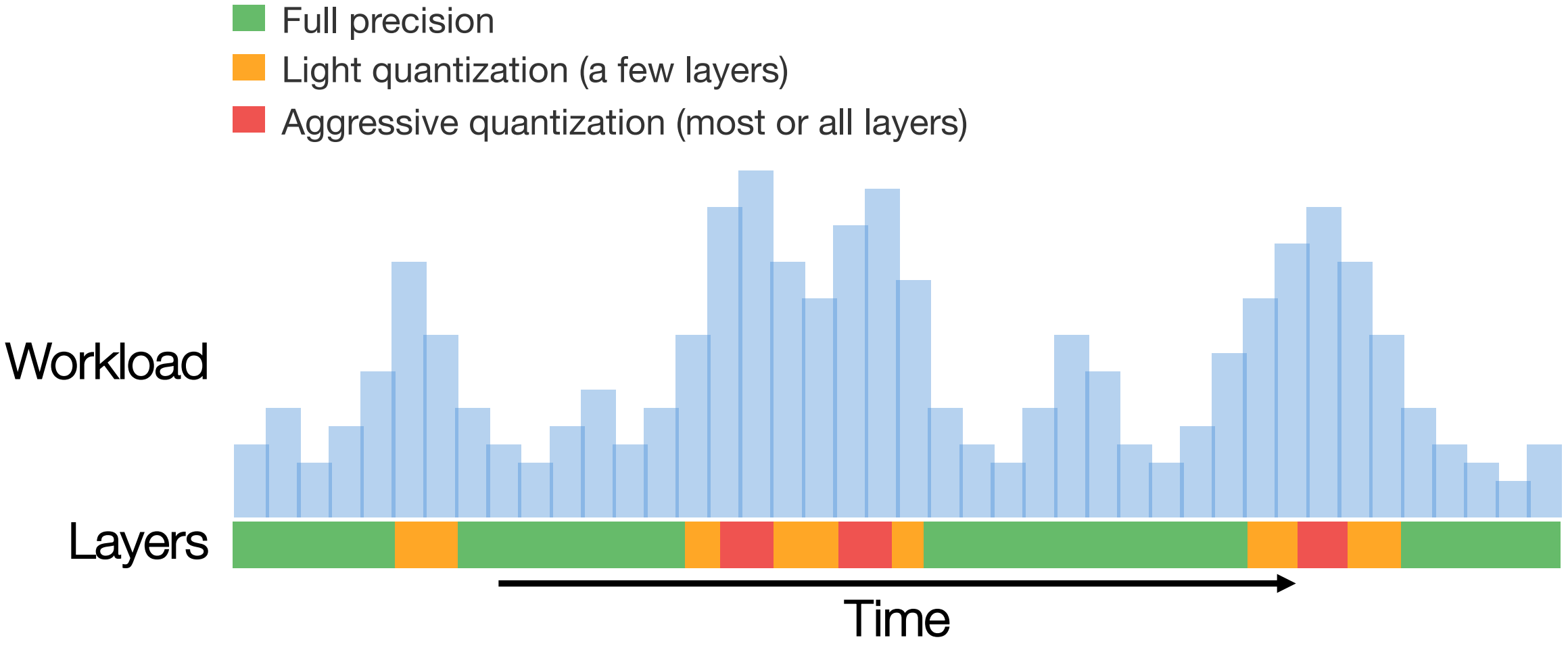
- Quantization reduces memory and accelerates computation
- The quality drop is **constant**, even when full-precision serving is feasible



Ideal system: Adaptive to workloads in real time

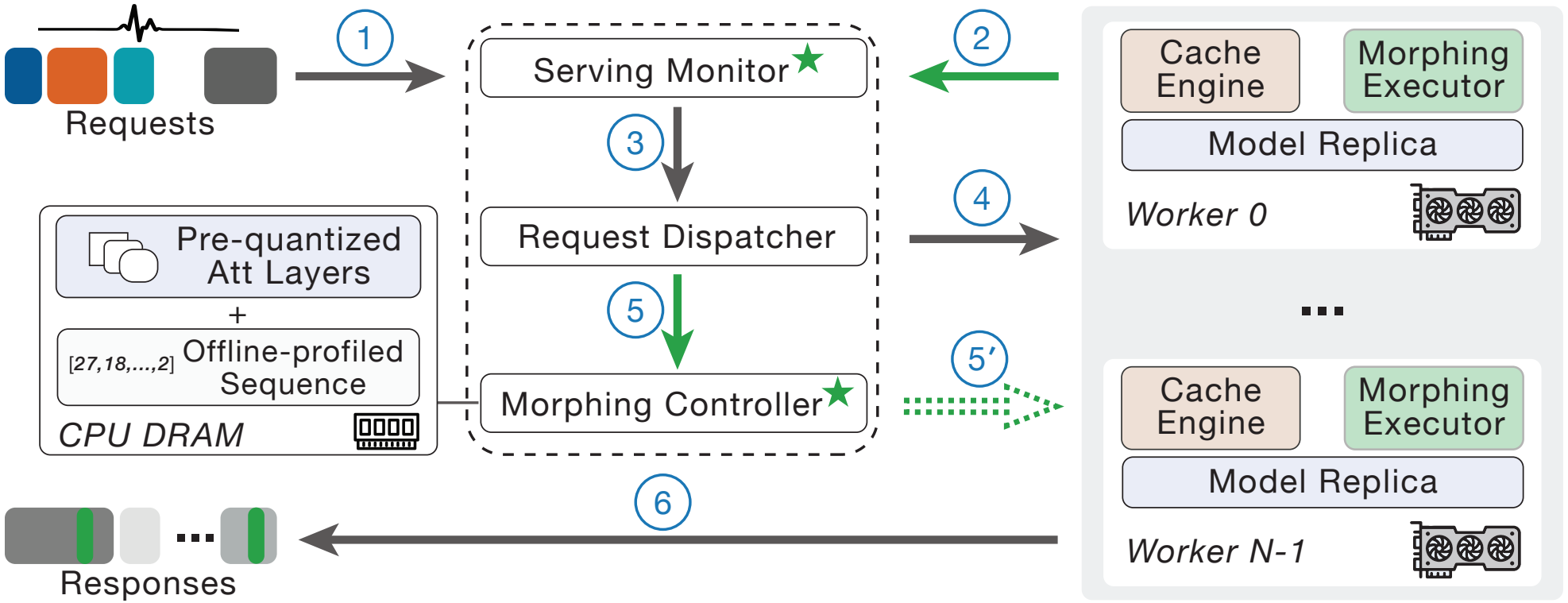


Ideal case: Adaptive to workloads in real time



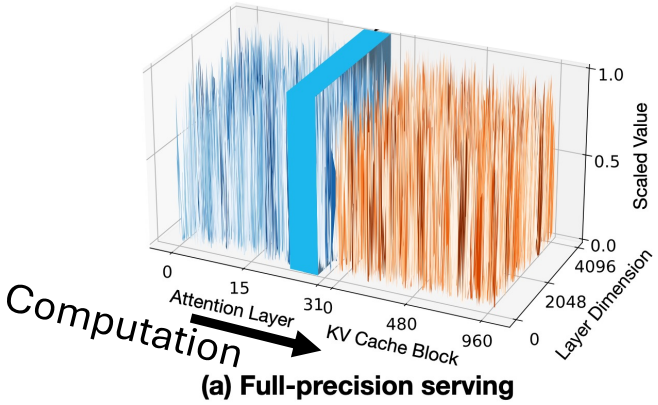
MorphServe: System overview

Token-level adaptation by a closed-loop feedback architecture.



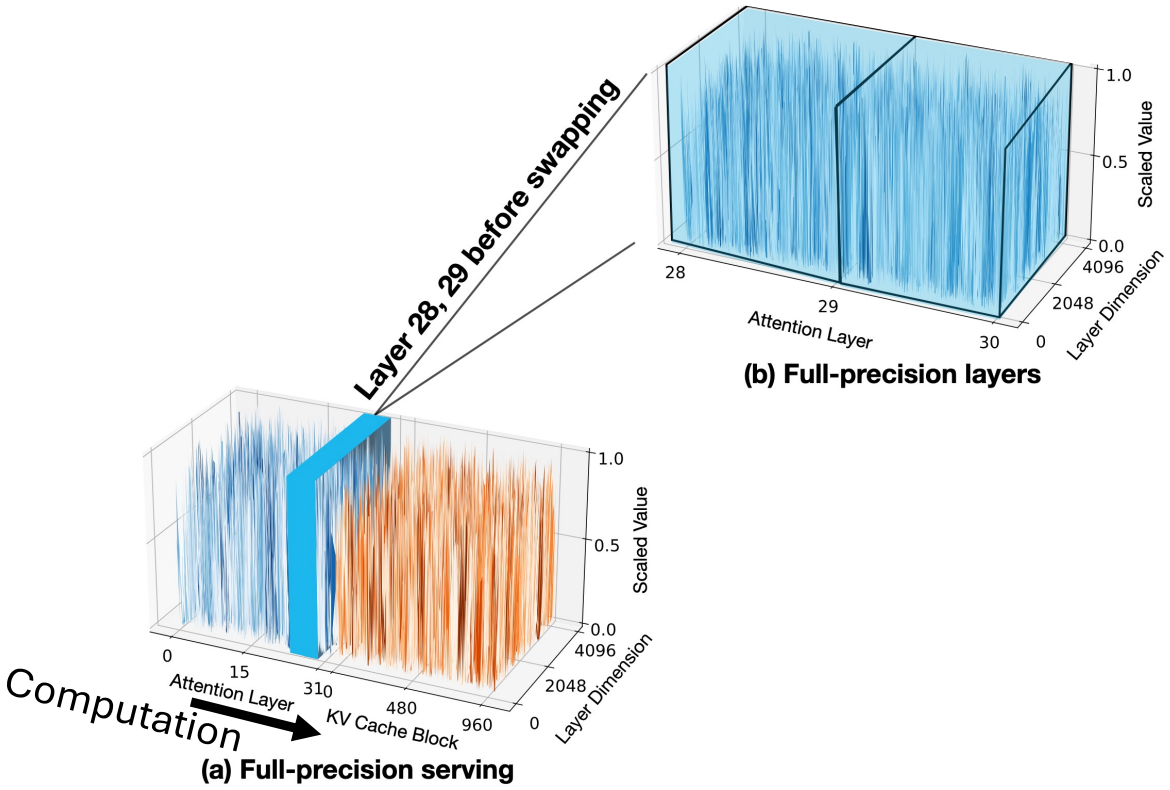
Layer morphing: Initial state

Layers at **full-precision (FP16)** with a **fixed** number of KVC blocks.



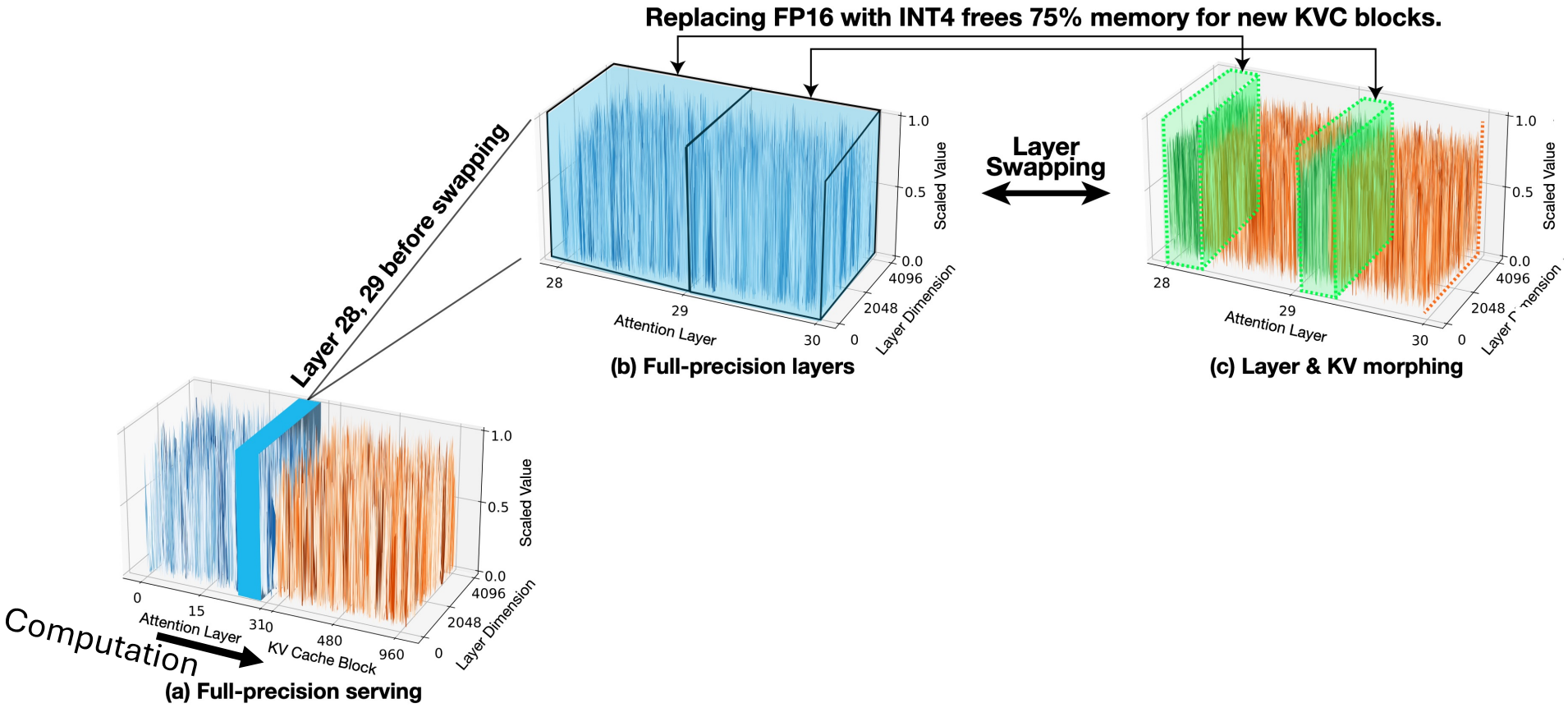
Layer morphing: Initial state

Zooming in two decoder layers, 28 and 29.



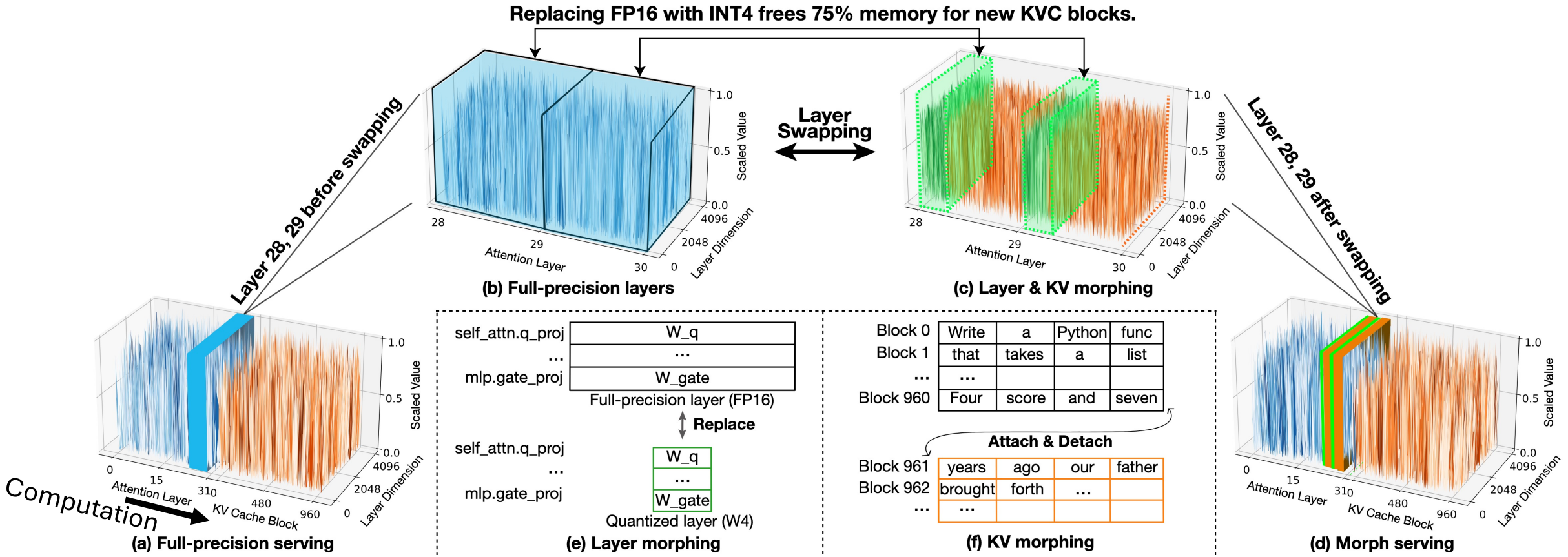
Layer morphing: Quantized layer swapping & KV resizing

Morphing triggered by **memory** or **computation pressure**;
Quantized layers (green) with expended KVC blocks (orange)



Layer morphing: Asynchronous mechanism

A morphing process takes only **~6ms** for Llama 2 7B model on L4, and can be fully **overlapped** with the computation.



Evaluation Setup

Models

- Vicuna 7B, Llama 2 7B, Llama 3 8B, CodeLlama 34B

Workload Traces

- Azure LLM Trace 2023 Code
- BurstGPT Trace

Datasets:

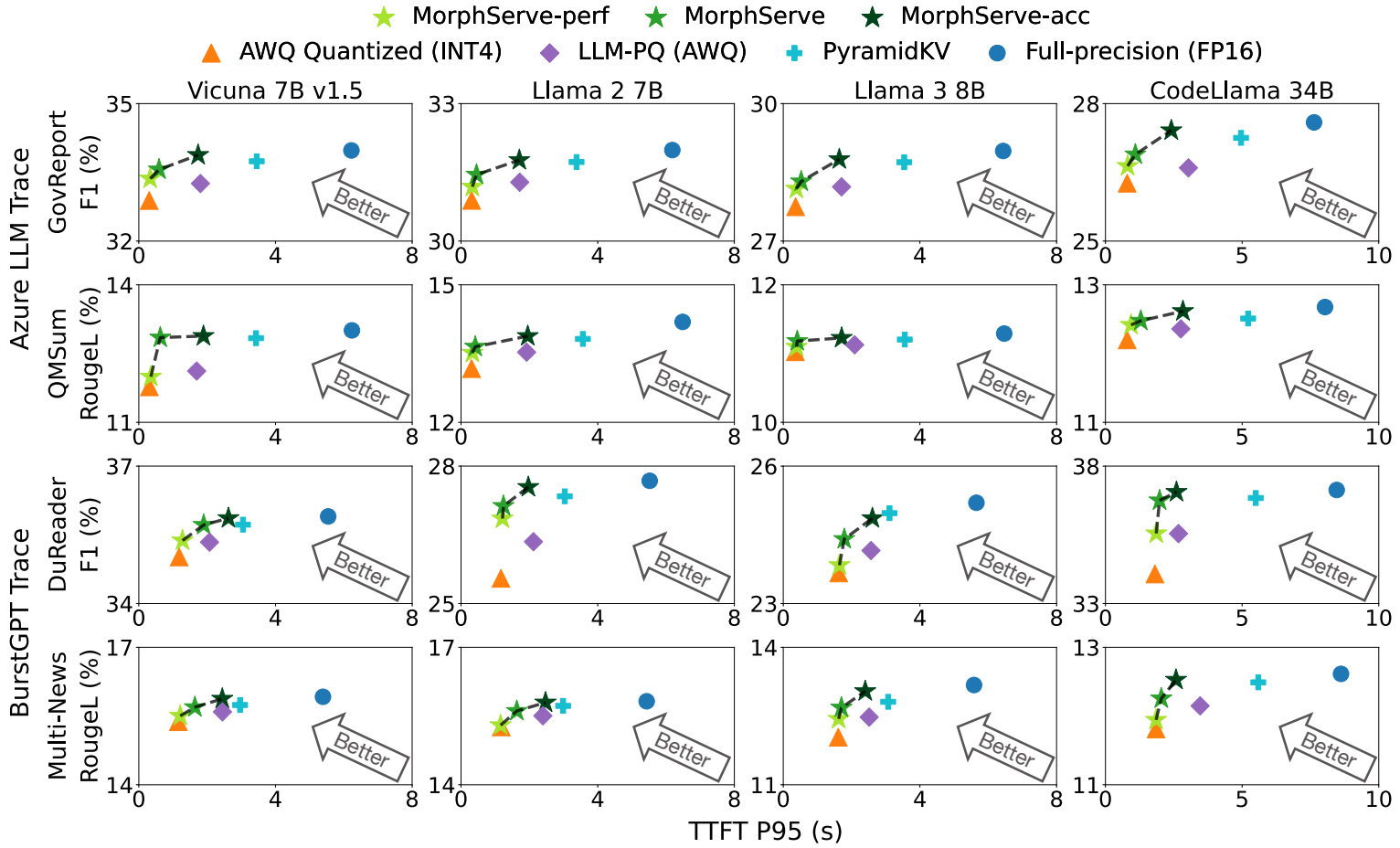
- GovReport, Multi-News, QMSum, DuReader

Hardware:

- L4 (24 GB HBM) for ≤8B models
- A100 (80 GB HBM) for 34B model

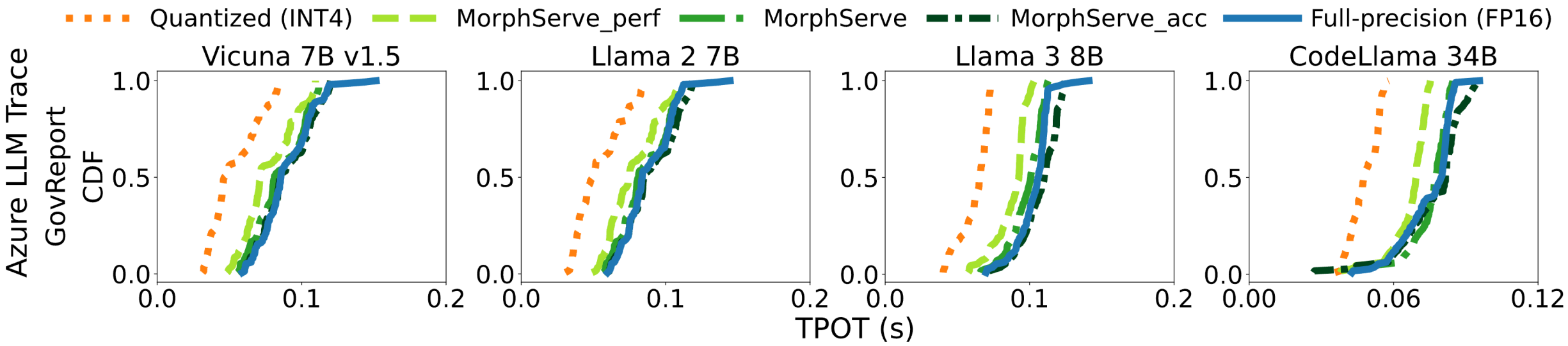
Evaluation: Best TTFT latency–accuracy tradeoff

- **Dominates** all baselines in latency-quality tradeoff
- **15.7×** vs. full-precision and **2.4×** vs. KV cache compression
- **41.3%** less accuracy degradation vs. static quantization



Evaluation: Improving tail TPOT latency

MorphServe incurs negligible runtime overhead while **reducing tail TPOT latency** compared to full-precision serving.



Conclusion

- MorphServe is **not** a new **quantization** or **KVC compression**
- **A runtime serving system** that asynchronously coordinates weight precision and KVC capacity, and is fully compatible with existing quant and KVC compression
- **Full-quality** under **normal load**; **graceful degradation** under **overload**
- Advances the **accuracy–efficiency Pareto frontier**

MorphServe: Efficient and Workload-Aware LLM Serving via Runtime Quantized Layer Swapping and KV Cache Resizing

Zhaoyuan Su[1], Zeyu Zhang[1], Tingfeng Lan[1], Zirui Wang[1]
Haiying Shen[1], Juncheng Yang[2], Yue Cheng[1]

[1]

HARVARD UNIVERSITY [2]