EdgeReasoning

Characterizing Reasoning LLM Deployment on Edge GPUs

Benjamin Kubwimana, Jenny Huang

bkubwimana@nvidia.com

IISWC 2025

Outline

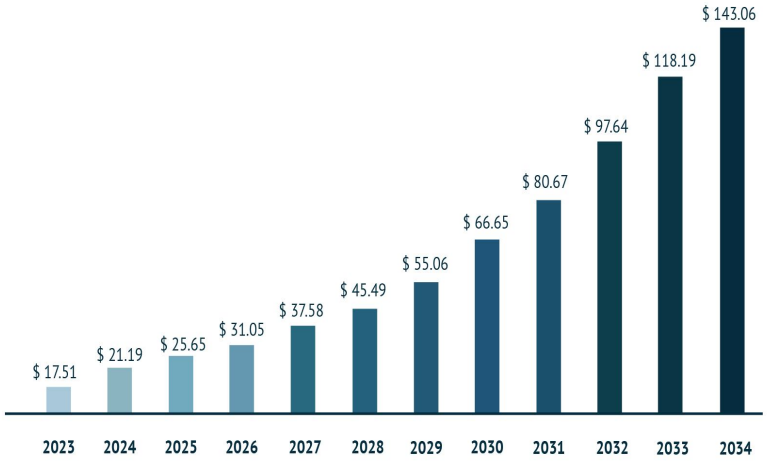
1. Background & Motivation
2. Challenges & Research Gap
3. Methodology
   a. Analytical
   b. Experimental
4. Results
5. Conclusion

Background & Motivation

Edge AI is on the Rise

- Edge AI is becoming more prevalent
- SLM with reasoning capabilities

Annual growth rate: 21.04% → 2025 to 2034



Edge AI Market Size 2023 to 2034 (USD Billion)

Inference:

- Reasoning models → higher accuracy
- Will be dominant workload
- More on-device GenAI (eg. in smartphones)

Sources: Edge Ai Foundation, IDC: International Data Corporation

Background & Motivation

Benefits

Edge AI is becoming critical in emerging autonomous systems

AMRs

Humanoids

Data Privacy

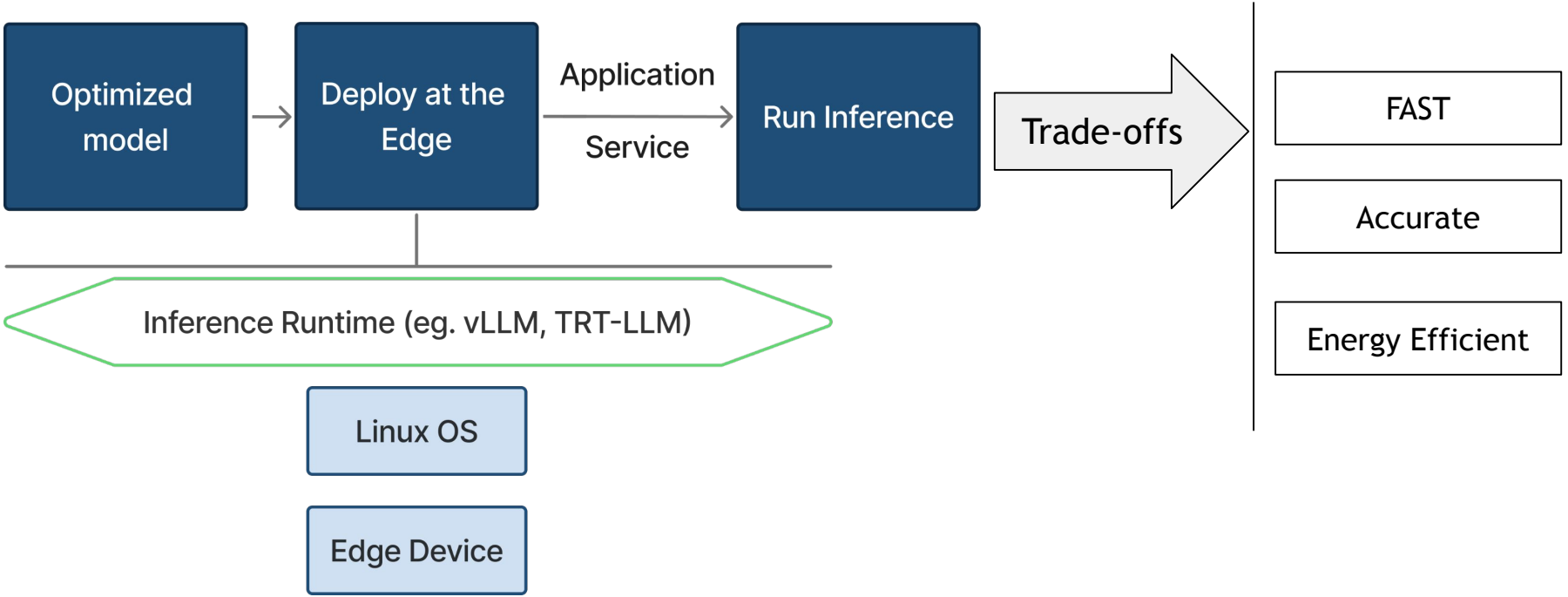
Flexible Deployment

Low Cost

Connectivity Resilience

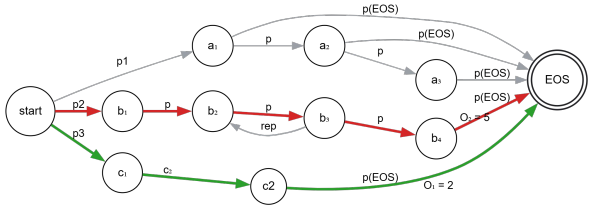
Challenges & Research Gap

Model Pipeline



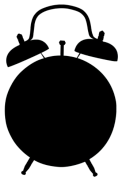
Challenges & Research Gap

Challenges

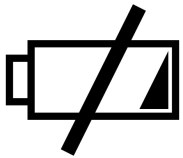


Reasoning has non-deterministic output length

- OpenAI's o1 costs 6x > non-reasoning GPT-4o



Strict latency constraints



Limited device capacity – limits model size

Research Problem

Which combinations of:

1. LLM architecture/size,
2. token-length control,
3. test-time scaling

Are Pareto Optimal for

1. Accuracy
2. Latency
3. Energy efficiency

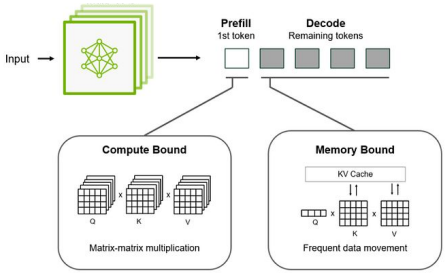
on edge GPUs with tight memory-compute & power limits?

Methodology

Characterization & Analytical Modeling:

Prefill & Decode

- Latency
- Energy and power consumption



Token Control (sequential & Parallel scaling):

- Soft limit (prompt based)
- Hard limit
- Non-reasoning system prompt
- Parallel scaling w/ majority vote

Quantization:

- Activation-aware weight quantization W4A16

Methodology

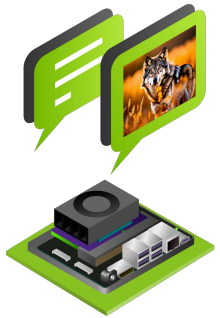
Experimental Setup

Models

Direct	Distilled and reasoning	Quantized
Qwen-2.5 1.5B -Instruct	DeepSeek-R1-Distill-Qwen-1.5B	DSR1-1.5B-llmc-awq-w4
LLama-3.1 8B -Instruct	DeepSeek-R1-Distill-Llama-8B	DSR1-8B-llmc-awq-w4
Qwen-2.5 14B -Instruct	DeepSeek-R1-Distill-Qwen-14B	DSR1-14B-llmc-awq-w4

Methodology

Experimental Setup



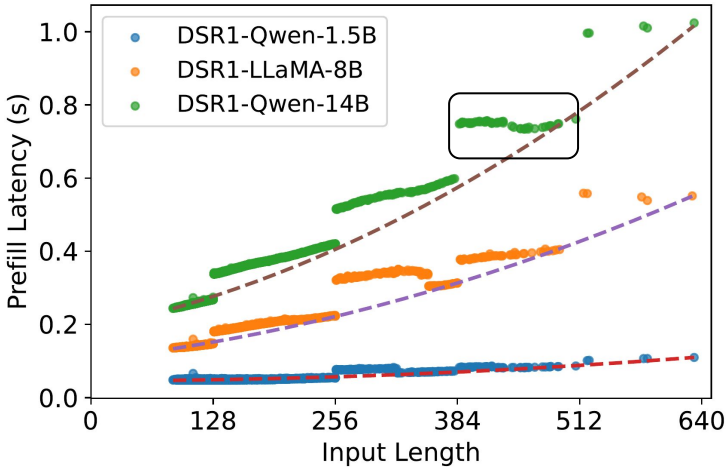
Benchmarks:

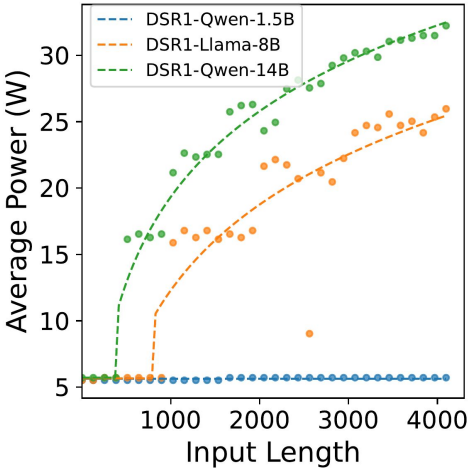
- MMLU-Redux
- AIME2024
- Google Deepmind’s Natural Planner

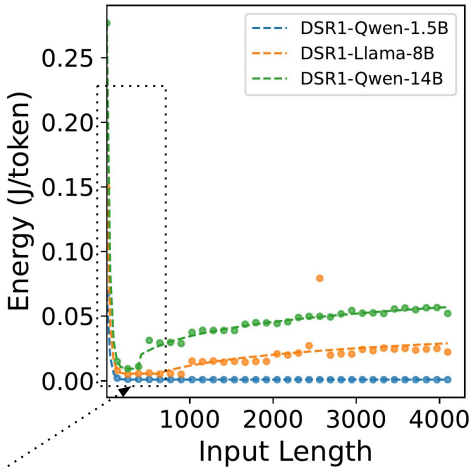
HW: AGX Orin 64GB, 64 Tensor Cores, up to 275 Sparse INT8 TOPS

Results

Characterization: Prefill





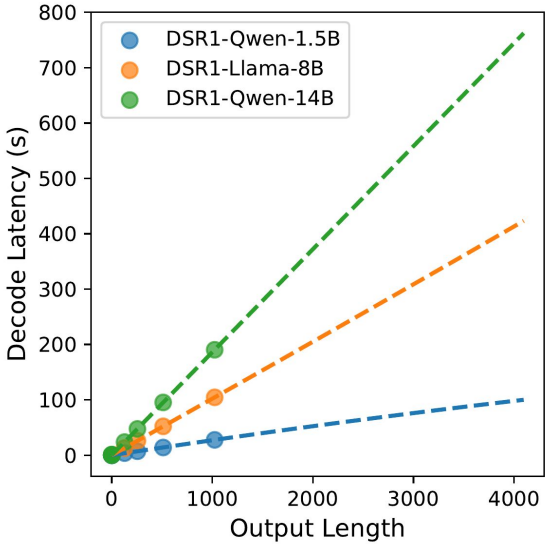


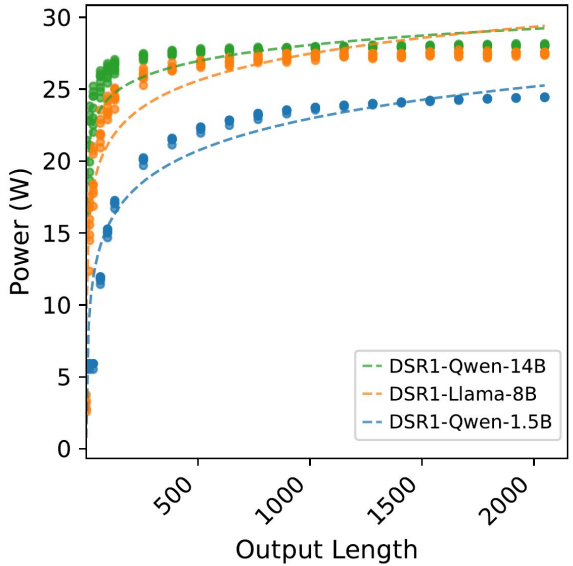
- We observe tile quantization effects at compute bound lengths
- Prefill phase can be modeled to map (I) →latency/energy on Jetson Orin

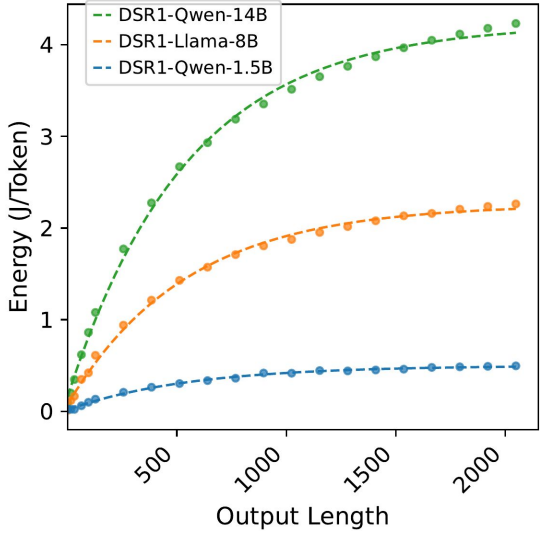
- Smaller Models have constant prefill power
- Setup energy cost dominates total energy at small prefill lengths

Results

Characterization: Decode







Input length: 512

- Decode latency grows linearly w.r.t the output length: from autoregression
- Decode: > 99.5% of total inference time

- Power increases logarithmically w.r.t sequence length.
- Growing computational and memory demands in the attention layer as the context window expands

Results

Analytical Models: fit functions

$$L_{\text{prefill}}(I) = aI_{\text{pad}}^2 + bI_{\text{pad}} + c$$

$$L_{\text{decode}}(I, O) = nO + m\left(IO + \frac{O(O-1)}{2}\right)$$

$$P_{\text{prefill}}(I) = \begin{cases} P_0, & I \leq v \\ P_0 + w\ln(I), & I > v \end{cases}$$

$$P_{\text{decode}}(O) = \begin{cases} P_0, & O \leq t \\ P_0 + y\ln(O), & O > t \end{cases}$$

Variable	Description	Units
I, O	Input/Output tokens	tokens
I_{pad}	Padded input tokens	tokens
a, n	Quadratic coefficients	s/token^2
b, m	Linear coefficients	s/token
c	Setup cost	s
P_0	Base power	W
v, t	Transition thresholds	tokens
w, y	Log coefficients	W

Results

Analytical Models: Main takeaways

- *Edge inference latency of LLMs can be accurately modeled using polynomial functions.*
- *Average power and total energy consumption increase logarithmically with sequence length on NVIDIA Jetson AGX Orin platform.*

Results

Accuracy, Latency and Energy

1. Sequential Scaling

2. Test Time Parallel Scaling

3. Quantization

Results

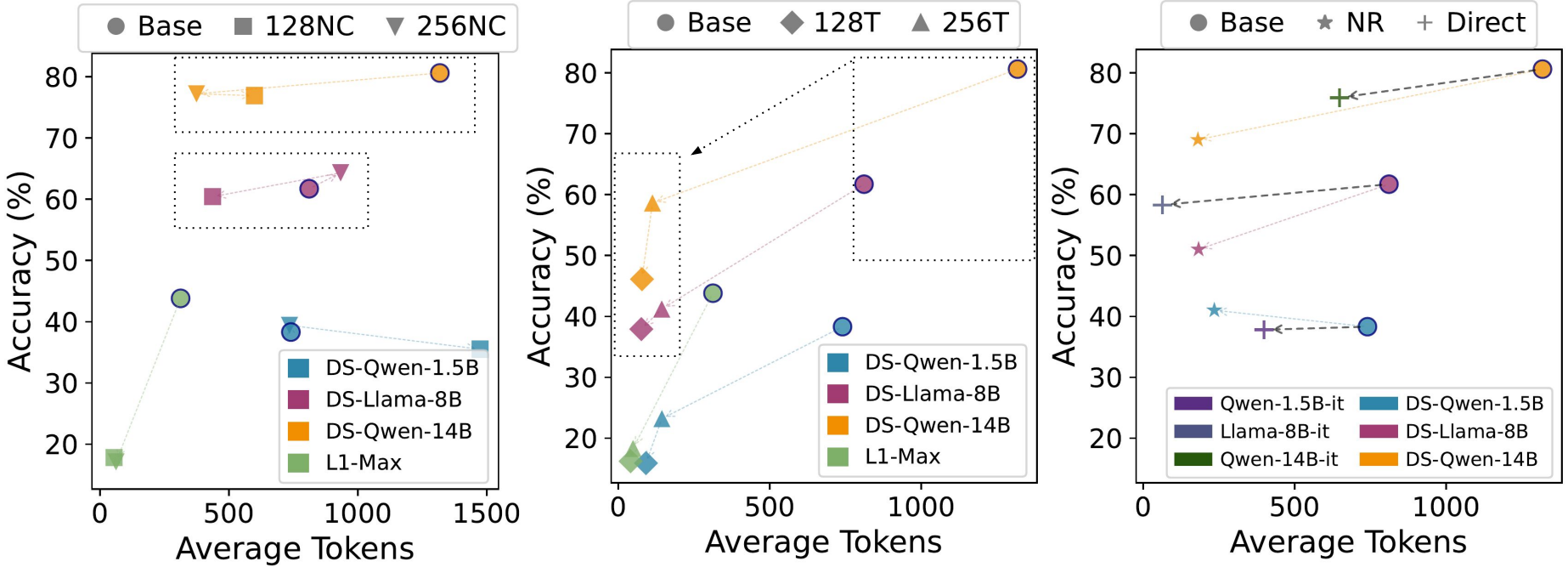
Accuracy, Latency and Energy

1. Sequential Scaling

- Baselines (reasoning & direct)
- Soft and hard limit
- Non-reasoning
- Fine-tuned L1Max

Results

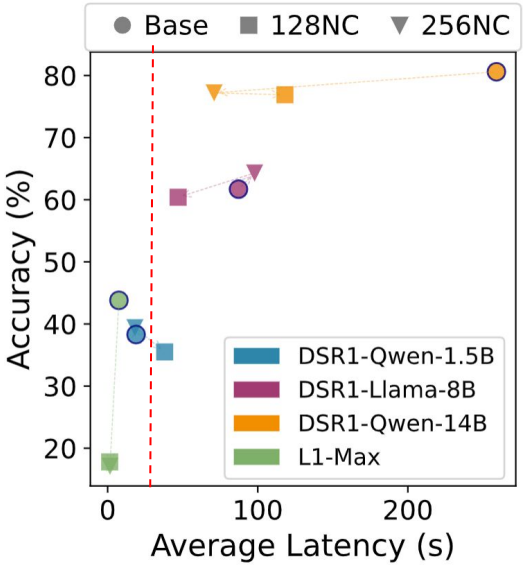
Sequential Scaling: Accuracy Vs Token Length



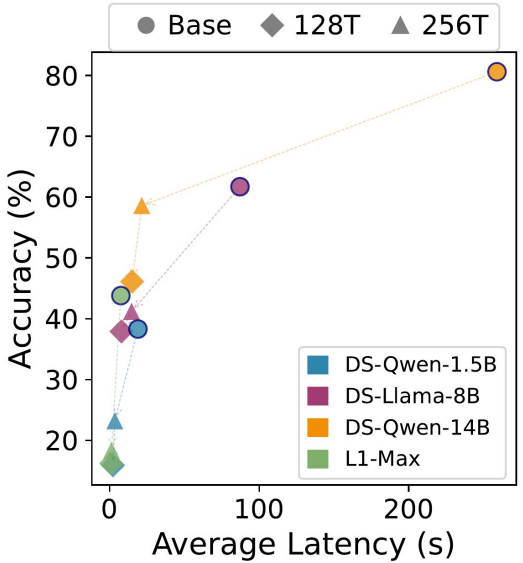
(a) Soft limit (b) Hard limit (c) No Reasoning

Results

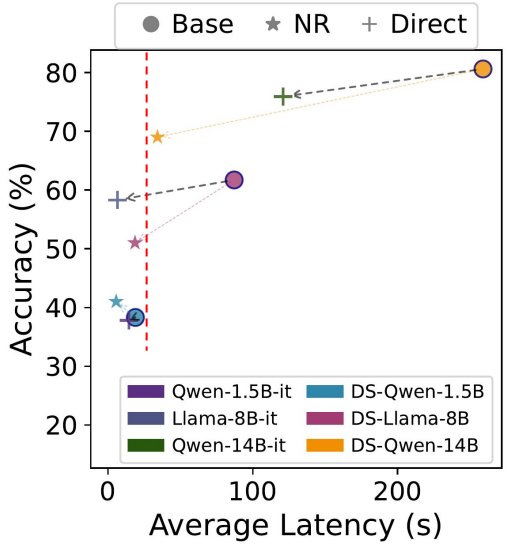
Sequential Scaling: Accuracy Vs Latency



(a) Soft limit



(b) Hard limit



(c) No Reasoning

- Sub-5s latency: Exclusively served by 1.5B models.
- 15-30s latency: Non-reasoning 8B models are preferred.
- 30s latency: DSR1-Qwen-14B emerges as optimal.

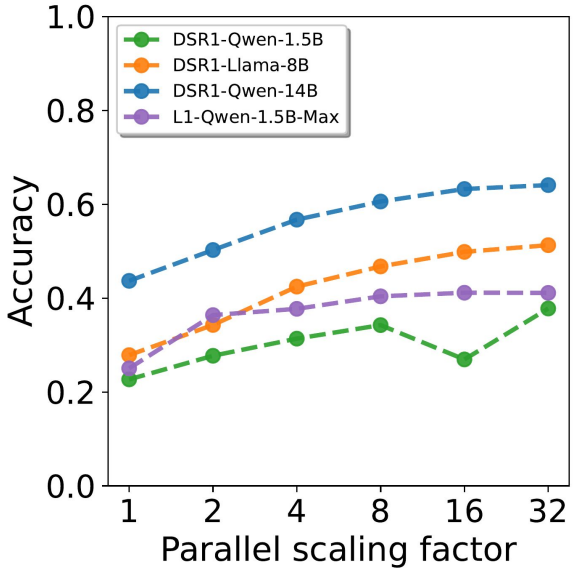
Results

Sequential Scaling: Main takeaways

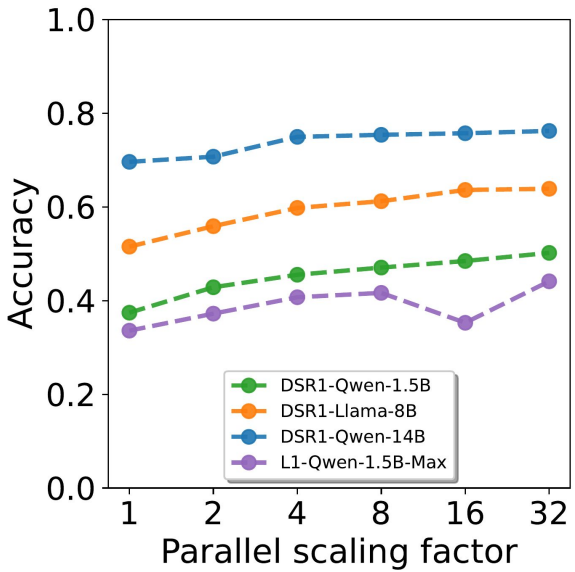
- *Sequential scaling holds even when reasoning token control is applied.*
- *Non-reasoning models offer a competitive latency-accuracy trade-off compared to reasoning models on a low token and latency budget.*

Results

Test Time Scaling: Latency & Energy



Output length 128



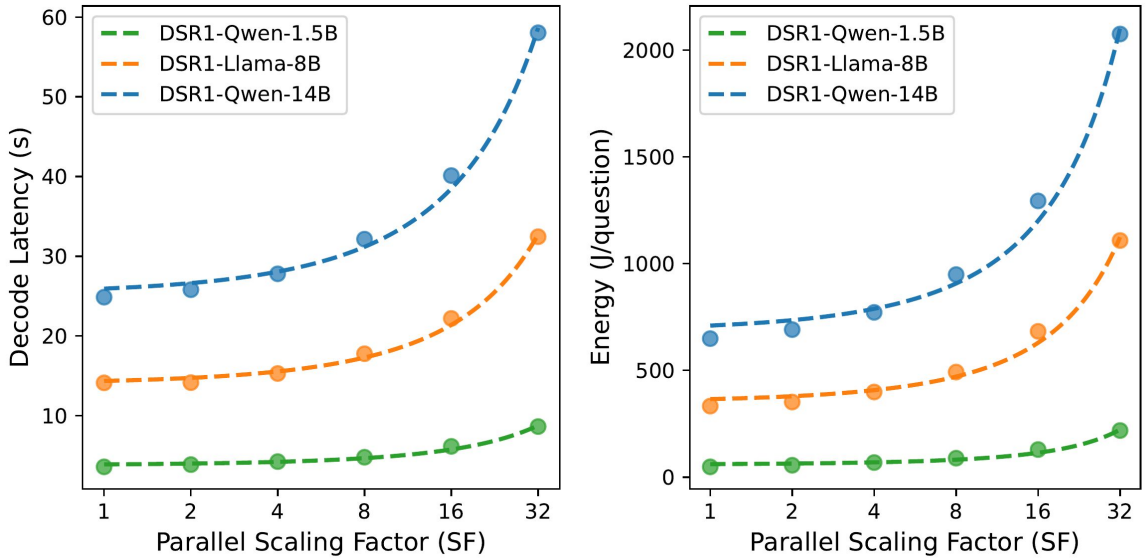
Output length 512

- Sequential + Parallel scaling improves accuracy up to 20%

Results

Test Time Scaling: Latency & Energy

Output length 128



- Latency and energy cost start increasing beyond SF 8

Results

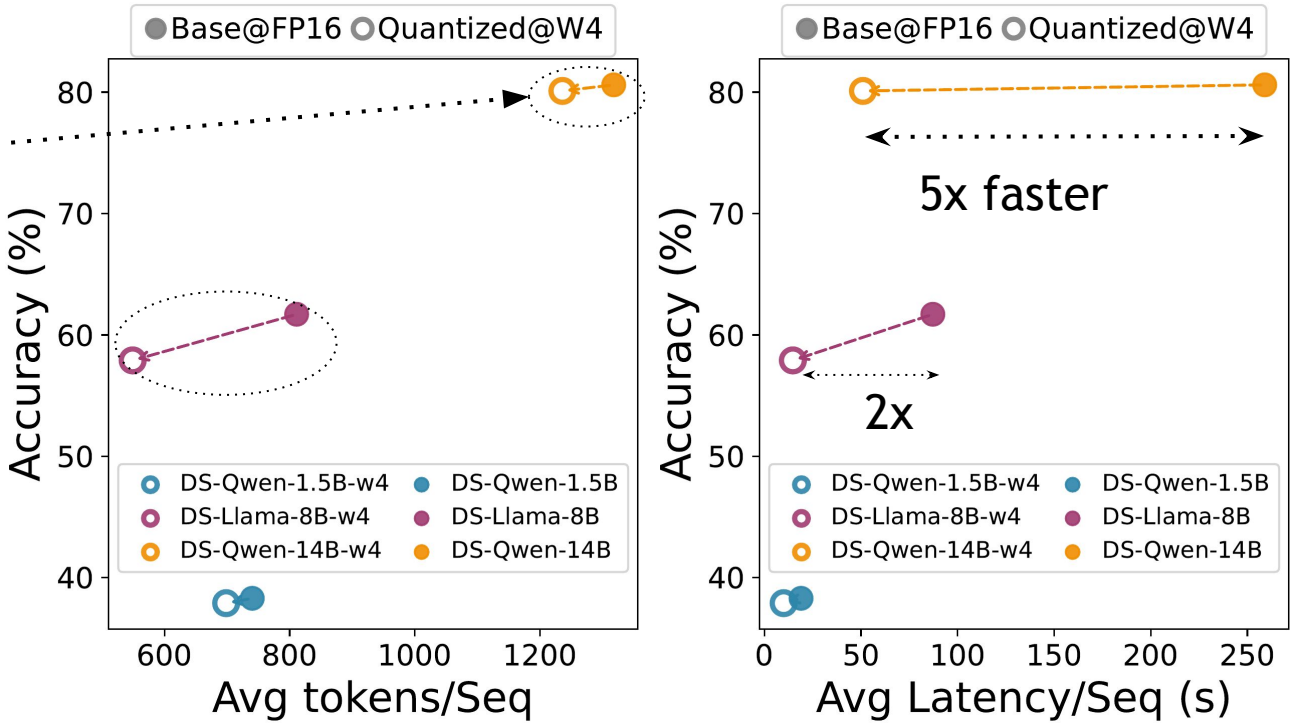
Impact of Test-Time Scaling: Main takeaways

- *Parallel scaling utilizes hardware resources effectively and improves the overall GPU utilization*
- *Parallel scaling improves accuracy with minimal latency and energy overhead at small scaling factors **(≤ 8)**.*

Results

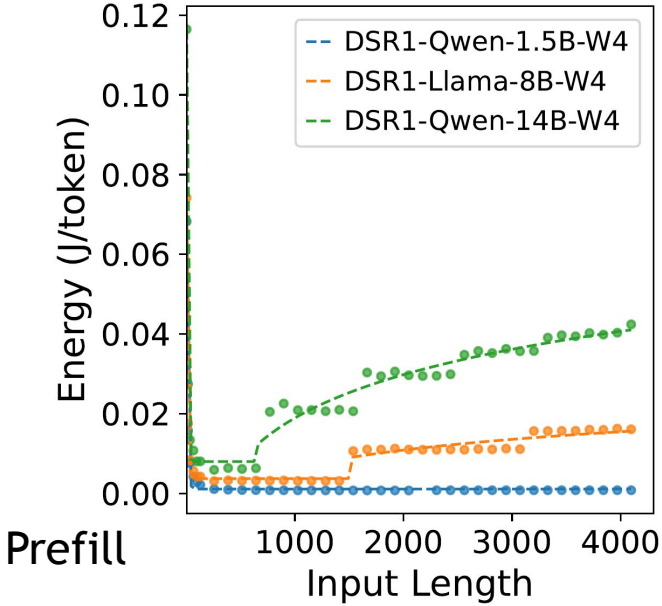
Impact of Quantization: Accuracy Vs tokens length & Latency

DSR1-Qwen-14B:
-0.62%

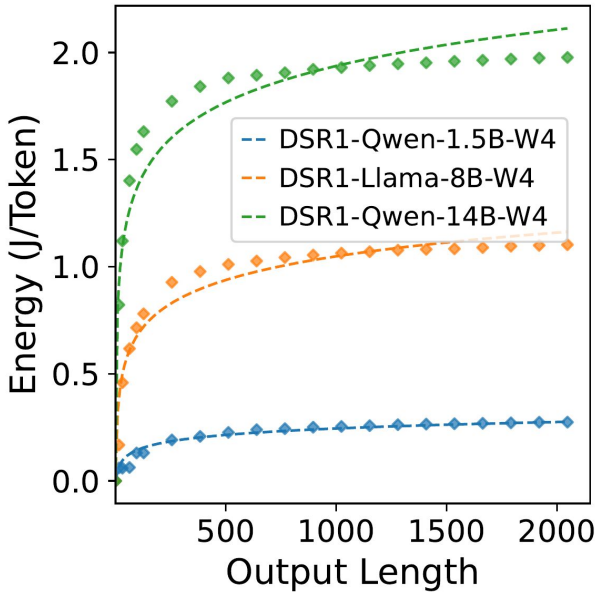


Results

Impact of Quantization: Prefill & Decode Energy/token



Prefill



Decode

Energy/token is 2x lower for quantized models (8B & 14B)

Results

Impact of Quantization: Main takeaways

AWQ based W4 quantization reduces:

1. *latency by roughly **2–5×** and*
2. *energy per token*
3. *small accuracy changes*

Benefits increase with model size.

Conclusions

➔ We systematically quantify the impact of:
- Model scale,
- Input/output sequence lengths,
- Inference time scaling techniques

On →
- Accuracy
- Latency,
- Power,
- Energy efficiency

➔ We use analytical models to map these parameters to
- performance metrics for rapid evaluation of optimal deployment strategies W/O exhaustive hardware testing

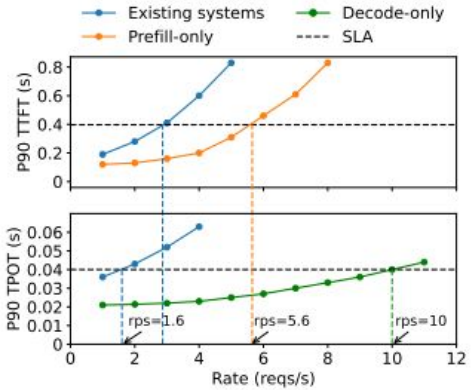
➔ We provide configuration guidelines for maximizing accuracy under diverse latency constraints

Github: https://github.com/edge-inference/edgereasoning
Questions? bkubwimana@nvidia.com

Artifacts
- Available
- Reviewed
- Reproducible


NVIDIA
Questions ?

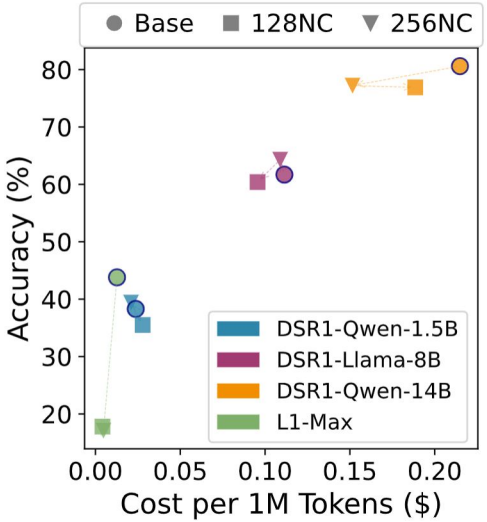
Disaggregated Prefill/Decode



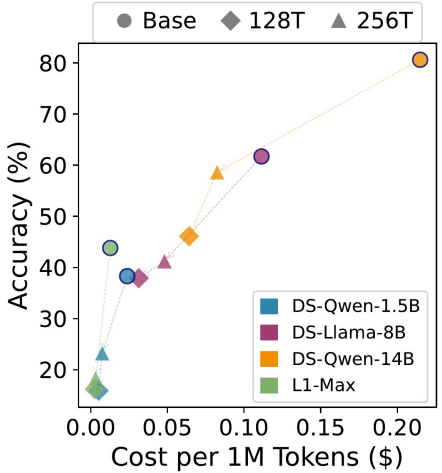
- Observed higher throughput at faster TTFT and TPOT
- The reduced inference time has low energy implications

DistServe: Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving , Yinmin Zhong et al.

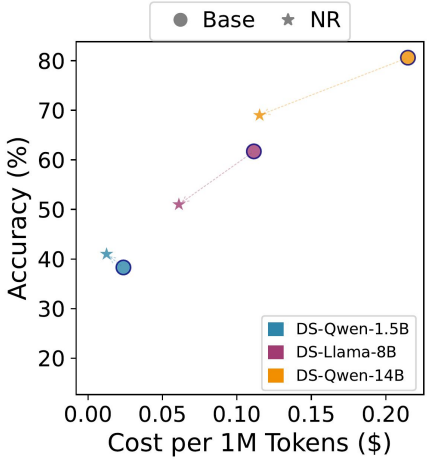
Budgeted Decode: Accuracy Vs. Cost



(a) Soft limit



(b) Hard limit



(c) No Reasoning

Edge costs **$0.027–$0.302 / 1M tokens** vs cloud reasoning up to **$60 / 1M output**

Inference Engines

Choosing the right inference engine matters: 10-15% improvement

Input Length	Output Length	HF Latency (s)	vLLM Latency (s)	vLLM Speedup (vs HF)	TRT-LLM Latency (s)	TRT-LLM Speedup (vs vLLM)
16	128	14.23	12.73	1.12x	12.79	1.00x
64	128	14.29	12.75	1.12x	12.46	1.05x
128	128	14.41	12.78	1.13x	12.88	0.99x

Conclusions

A guided way to choose optimal deployment recipe

1. Run a benchmark of target application (e.g agentic planning, video generation etc)
2. Use **EdgeReasoning** to
    i. collect metrics
    ii. make analytical models for a device and application
3. Estimate latency and energy consumption for different input/output length pairs
4. Create a pareto front