# MoE-CAP: Benchmarking Cost, Accuracy and Performance of Sparse Mixture-of-Experts Systems

Yinsicheng Jiang[*1], Yao Fu[*1], Yeqi Huang[*1], Ping Nie[3], Zhan Lu[1], Leyang Xue[1], Congjie He[1], Man-kit Sit[1], Jilong Xue[2], Li Dong[2], Ziming Miao[2], Dayou Du[1], Tairan Xu[1], Kai Zou[4], Edoardo Ponti[1 5], Luo Mai[1]

University of Edinburgh[1], Microsoft Research[2], Peking University[3], NetMind.AI[4], NVIDIA[5]

# Challenges for Deploying MoE Systems

**What is MoE system?** **MoE system is software that incorporates specific optimizations for MoE inference:**
**- Expert Parallelism, Quantization, Expert offloading (from GPU VRAM to CPU DRAM), …**

**1. Sparsity largely impacts bandwidth requirements for inference**

e.g. DeepSeek-R1 – 671B total params – 37B activated params per token

Target 10 tokens/s

RTX 4090
RTX 5090

DeepSeek-R1

Batch size = 1 (1 token per inference step) → 37B Activated Params (5.5% activated) → 1040 GB/s → GDDR6X, GDDR7

13.19X

DGX-H100
GB300

Batch size = 1000 (1000 tokens per inference step) → 671B Activated Params (Fully activated) → 13719 GB/s → HBM

**2. Too many deployment scenarios**

- single-user vs. multi-user
- small vs. large batch sizes

**Seeking benchmarks to answer this question**

**Given a model and a deployment scenario, what is the most suitable hardware and MoE system?**

# Existing Benchmarks Limitations

**Inadequate insights to guide hardware selection and further optimization**

- Open-LLM-Leaderboard
  - Accuracy, $CO_2$ Cost
  - No system performance reported

- LLM-Perf, Artificial Analysis, MLPerf
  - Prefill latency, Decoding throughput, Energy usage, Accuracy
  - Lack insights for hardware utilization and MoE system further improvement

- Databricks benchmark, LLM-Viewer
  - MBU (Model Bandwidth Utilization), MFU (Model FLOPS Utilization), Roofline model
  - Inaccurate for MoE systems.

# Our Contributions

## We are the first to introduce CAP benchmarking for MoE systems

**How to classify current MoE systems?**

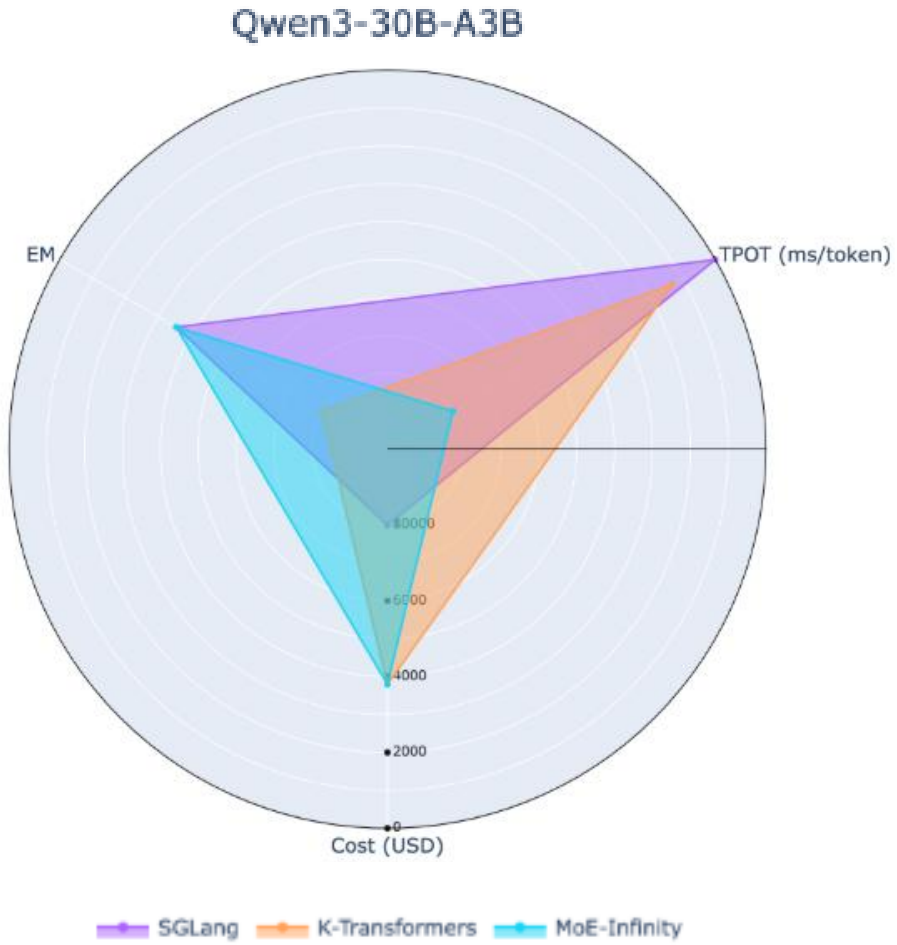
- PA systems (Expert Parallelism)
- PC systems (Quantization)
- CA systems (Offloading)

**How to measure accurate hardware utilization for MoE systems?**

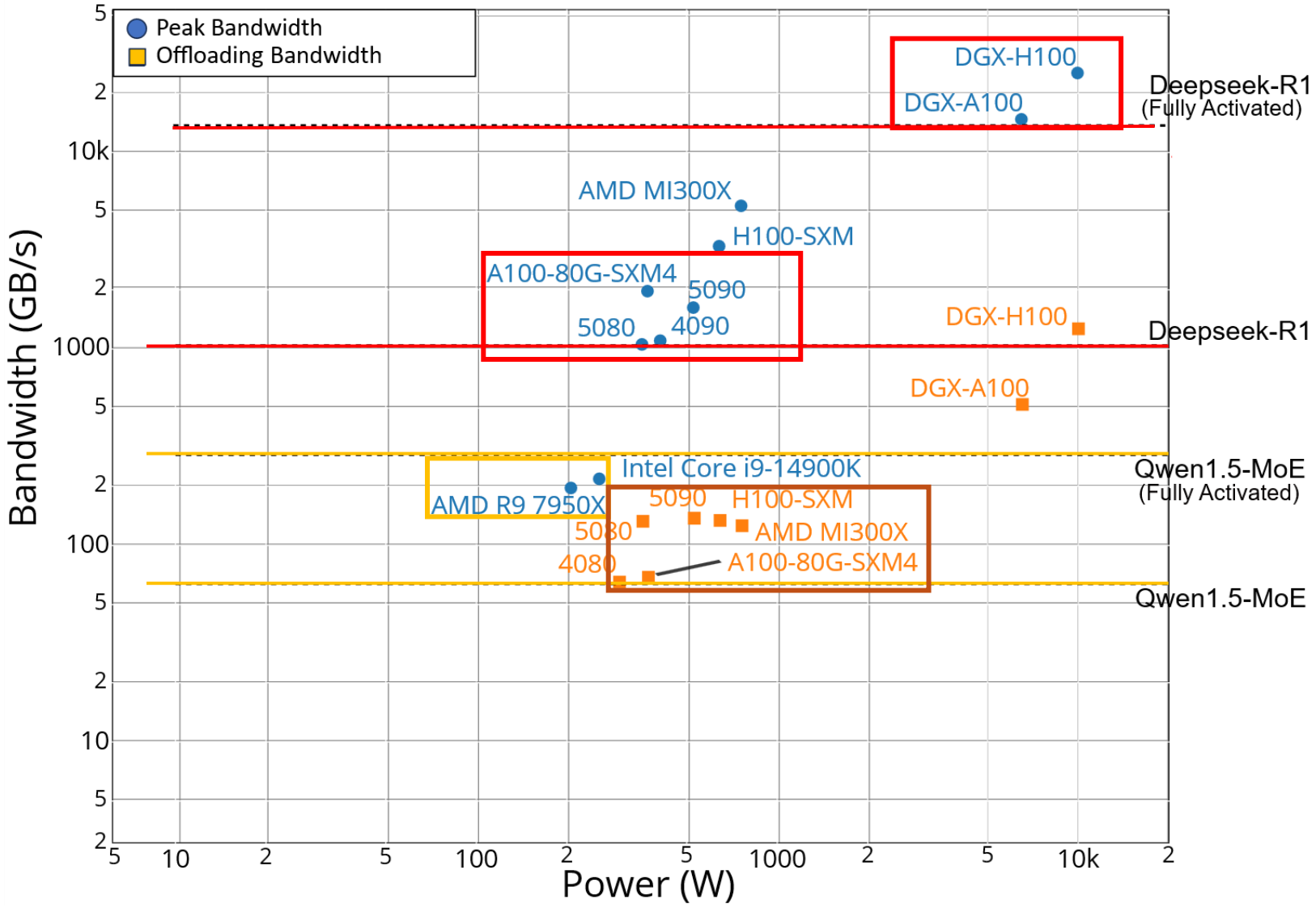
- S-MFU (Sparse Model FLOPS Utilization)
- S-MBU (Sparse Model Bandwidth Utilization)

**How to visualize the benchmark?**

- CAP Radar diagram



# Benchmark and Takeaways




**MoE systems enable a broader range of devices to perform inference.**
**- Personal machines can serve large MoE models at small batch size**

**Hybrid computing will become more prevalent.**
**- CPU matrix computation capabilities are increasing and DRAM can be easily scaled.**

**MoE systems should be co-designed to align with specific applications and deployment scenarios.**

**New benchmarking and design principles are needed for emerging sparse AI systems.**
**- MoE is not everything for sparse AI systems. What about sparse KV Cache?**

# Epilogue

MoE-CAP can

- Offer accurate hardware utilization metrics
- Guide further improvements in the system efficiency
- Help choose the best hardware based on the use case for cost savings
- Analyze the trade-off of Cost, Accuracy and Performance among different MoE systems


More details of MoE-CAP, please refer to our paper and code:

https://arxiv.org/abs/2412.07067

https://github.com/sparse-generative-ai/MoE-CAP

Let's work together to enhance MoE efficiency!