



MoE-Inference-Bench

Performance Evaluation of Mixture of Expert Large Language and Vision Models

*Krishna Teja Chitty-Venkata^{*1}, Sylvia Howland²,
Golara Azar², Daria Soboleva², Natalia Vassilieva²,
Siddhisanket Raskar^{3\$}, Murali Emani^{1\$}, Venkatram Vishwanath¹*

¹Argonne National Laboratory, ²Cerebras, ³Pacific Northwest National Laboratory

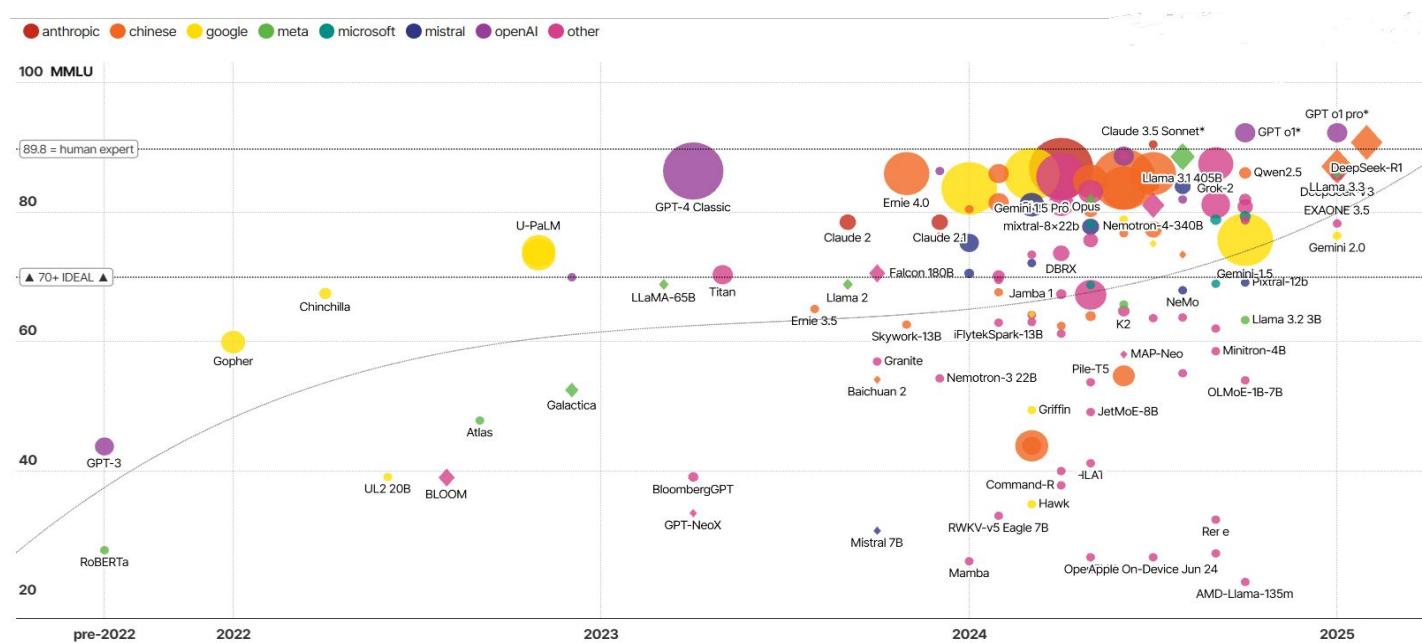
^{\$}Speaker

*Now at Red Hat AI

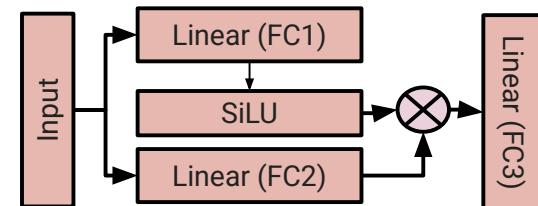
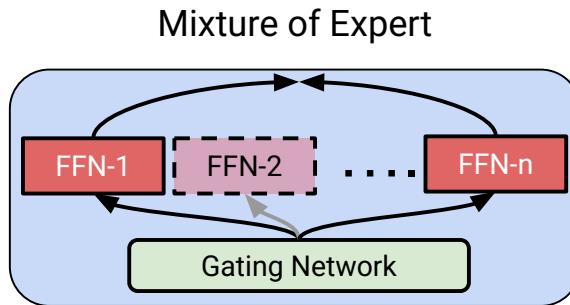
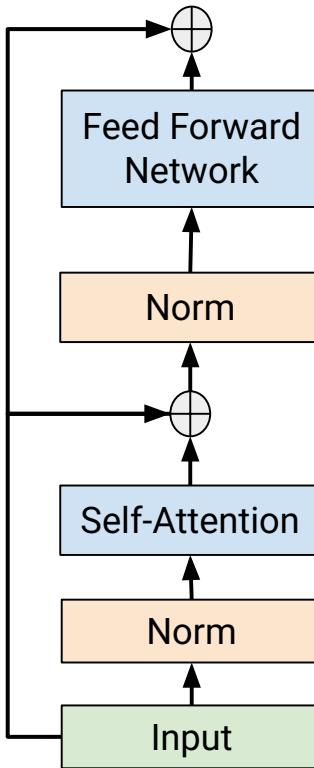
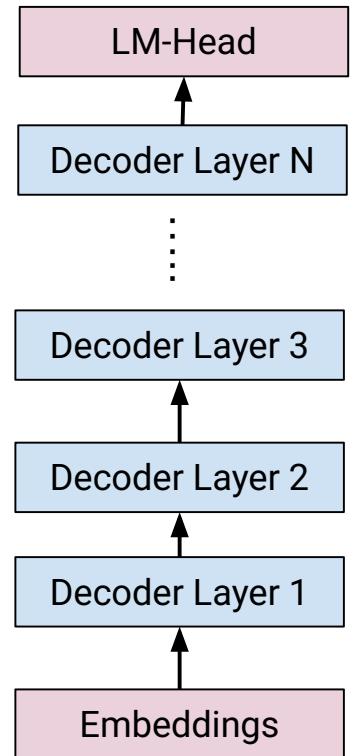
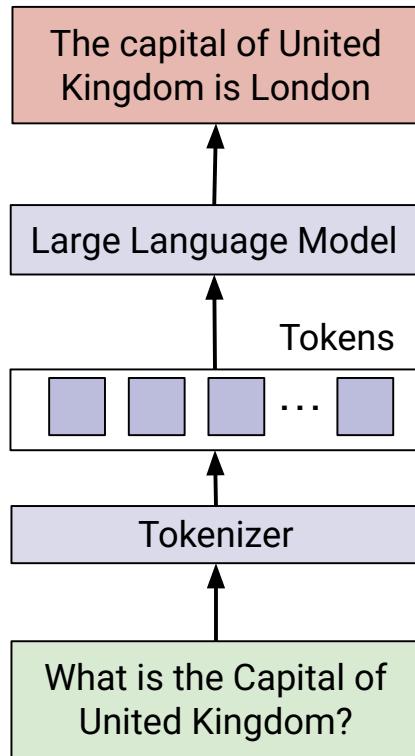


Large Language and Vision Models (LLMs and VLMs)

- LLM is an AI algorithm that can summarize, translate, predict, and generate human-like text to convey ideas
 - They are trained on vast amounts of data using sophisticated algorithms to perform a wide range of tasks
 - They rely on a massive number of parameters, which allows them to capture patterns and context

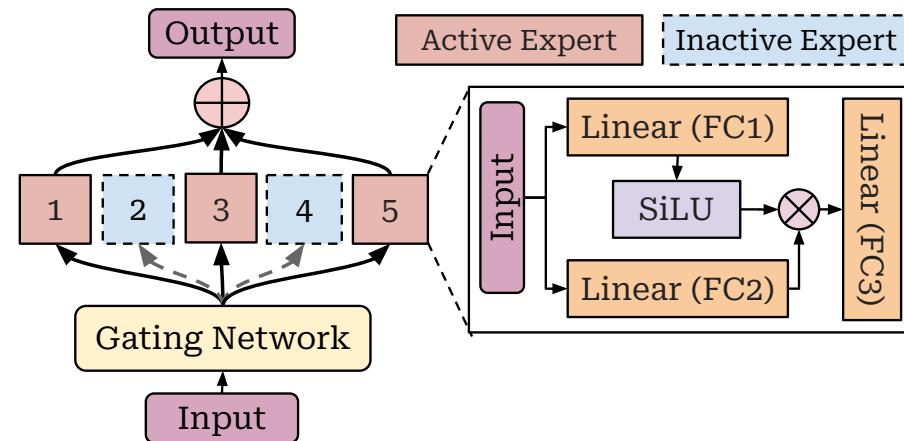


Large Language Models (LLMs)



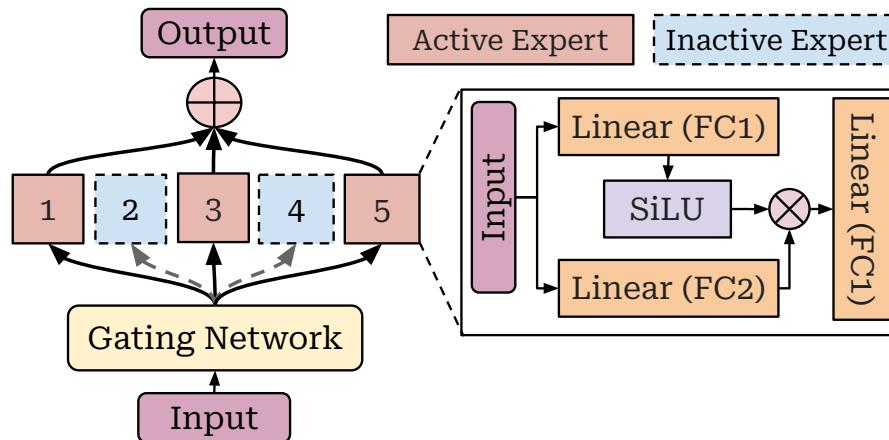
Mixture of Experts (MoEs)

- **Dynamic Expert Selection:** Specialized expert subnetworks are dynamically selected for each input, enhancing efficiency and performance.
- **Scalable Architecture:** Scales to massive sizes by activating only relevant experts, maintaining computational efficiency.
- **Gating Network:** Routes inputs to the most appropriate experts, orchestrating the selection process.



Mixture of Experts (MoEs)

- Mixture of Experts (MoE) is an architecture designed to enhance model efficiency and performance by dynamically selecting a subset of specialized expert subnetworks for each input
- This selective activation enables the model to scale to massive sizes while maintaining computational cost by activating only relevant experts, rather than the entire network
- The gating network orchestrates this process by routing inputs to the most appropriate experts, which perform distinct transformations to provide superior and diverse predictions
- MoEs are particularly effective in handling complex, multi-task learning and large-scale problems, striking an optimal balance between capacity and efficiency

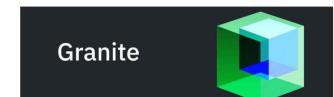
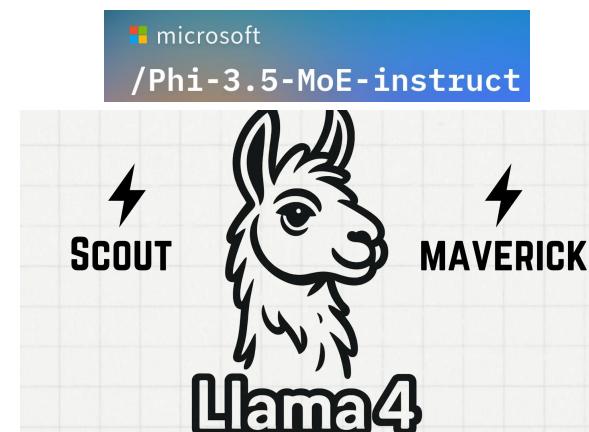


Rise and Rise of Mixture of Experts

- Powered by advances from leading organizations and open-source communities,
- MoE architectures like GPT-OSS, Mixtral, Llama-4, and Deepseek R1 combine specialized expert networks
- Deliver greater performance per training FLOP compared to traditional dense models



Deepseek R1

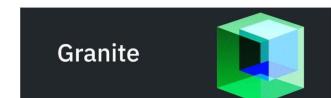
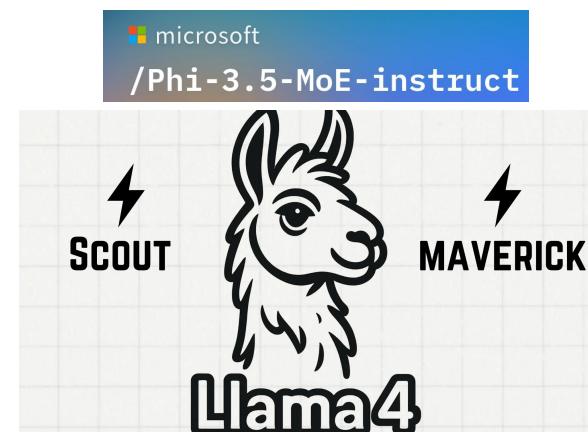


Rise and Rise of Mixture of Experts

- The MoE paradigm is rapidly **transforming** the landscape of Large Language and Vision Models, enabling unprecedented gains in **efficiency and scalability**
- Powered by advances from leading organizations and open-source communities, MoE architectures like GPT-OSS, Mistral, Llama-4, and Deepseek R1 combine specialized expert networks to deliver greater performance per parameter compared to traditional dense models
- This rise of MoE-based models marks a **new era of innovation**—driving smarter, more adaptable, and more resource-efficient AI across research and industry

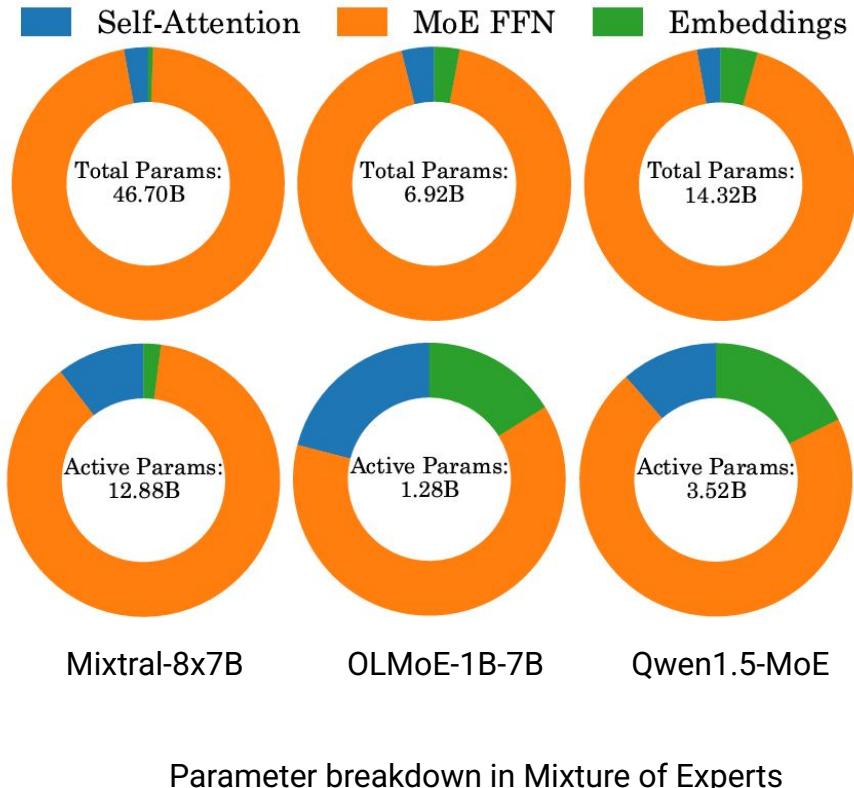


Deepseek R1



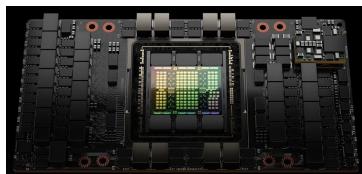
Why MoE Optimization is Important?

- **FFN layers** account for the majority of both total and active parameters
- Critical Impact on Memory and Compute
- Optimizing MoE layers directly impacts throughput, latency, and hardware utilization



Experimental Setup

Hardware Architectures



Nvidia H100 GPU



Cerebras CS-3

Inference Framework



MoE Models

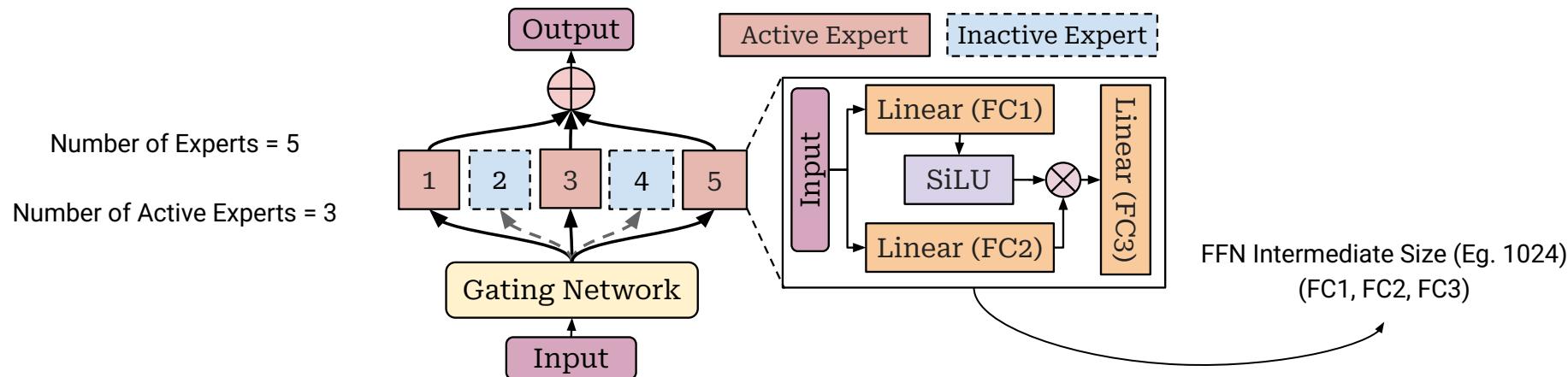
Model	Model Type	Modality	#Layers	#Hidden Size	#FFN Dimension	#Experts	#Active Experts	Model Size	Active Parameters
Mixtral 8x7B	Transformer	Text	32	4096	14336	8	2	47B	12.9B
Qwen 1.5 MoE	Transformer	Text	24	2048	5632	60	4	14.3B	2.7B
Qwen3-30B-A3B	Transformer	Text	48	5120	13824	128	8	30.5B	3.3B
DeepSeek V2 Lite	Transformer	Text	27	2048	1408	64	6	15.7B	2.4B
Phi 3.5 MoE	Transformer	Text	32	4096	6400	16	2	41.9B	6.6B
OLMoE-1B-7B	Transformer	Text	16	2048	8192	64	8	7.2B	1.3B
DeepSeek VL2 Tiny	Transformer	Text + Image	16	1536	8960	8	2	3B	1.0B
DeepSeek VL2 Small	Transformer	Text + Image	24	2048	11008	8	2	16B	2.8B
DeepSeek VL2	Transformer	Text + Image	32	4096	14336	8	2	27B	4.5B

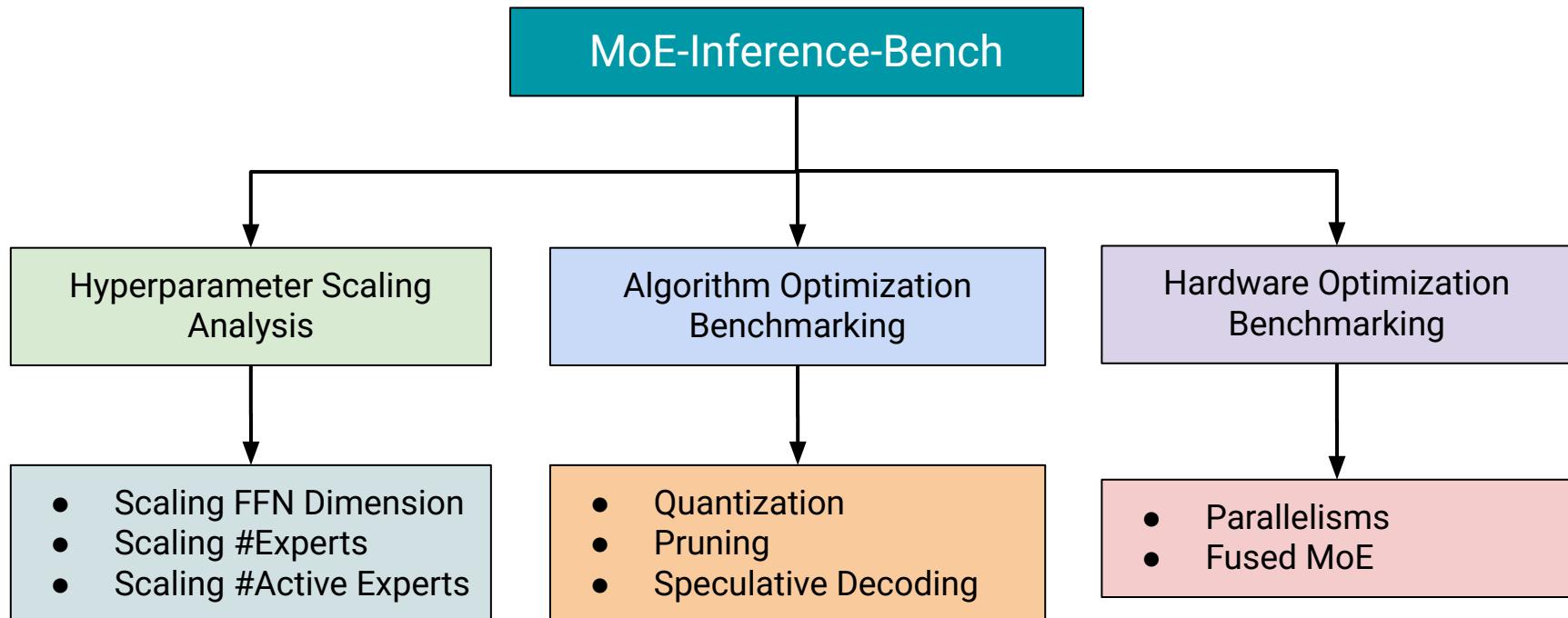
Performance Metric

$$\text{ITL} = \frac{\text{End-to-End Latency} - \text{TTFT}}{\text{Batch Size} \times \text{Output Tokens} - 1}$$

$$\text{Throughput} = \frac{\text{Batch Size} \times (\text{Input Tokens} + \text{Output Tokens})}{\text{End-to-End Inference Latency}}$$

- **Number of Experts:** The total number of experts represents the full set of FFN networks available as experts per layer
- **Number of Active Experts:** The number of active experts determines how many are dynamically chosen for each input
- **FFN Intermediate Size/ FFN Dimension:** Defined as the size of each individual expert

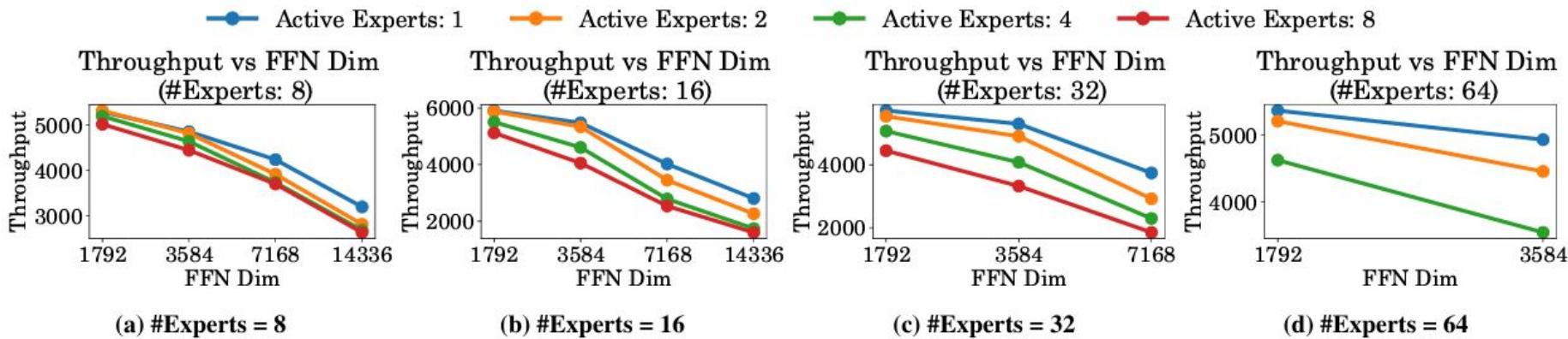




Scaling FFN Dimension

- Inverse Relationship between FFN Dimension and Throughput
- **Impact of Active Experts:** The reduction in throughput is more pronounced in configurations with a higher number of active experts.
- **Hardware-Level Bottlenecks:** At the highest FFN dimensions, the throughput across different active expert configurations begins to converge.

Memory bandwidth saturation can override parallelism gains from larger FFNs
Balancing FFN capacity and throughput is crucial for efficient deployment.

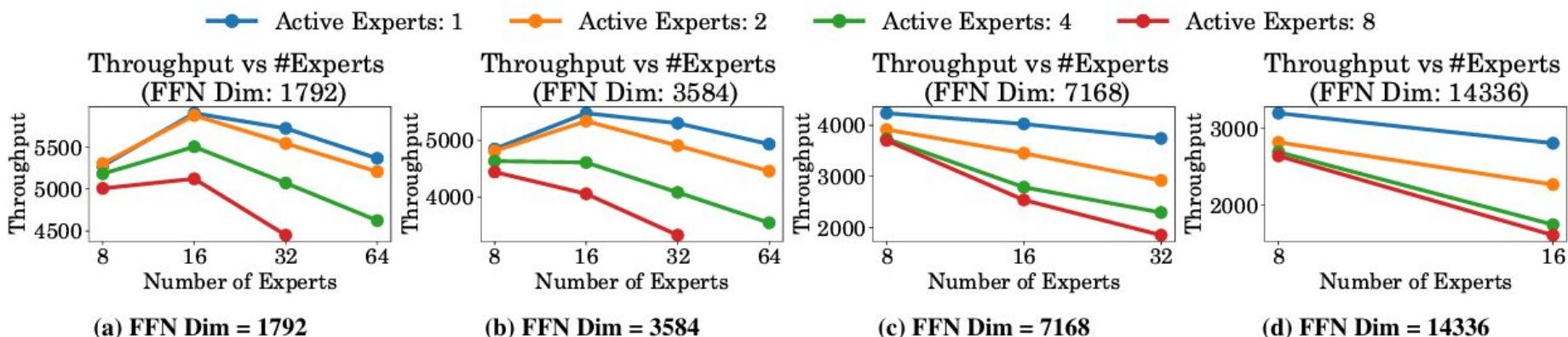


Batch Size 16 and Input/Output Length 2048 on 4 H100 GPUs on Mixtral-8x7B

Scaling Number of Experts

- **Smaller FFN Dimensions:** increasing the number of experts can lead to a slight improvement in throughput
- **Larger FFN Dimensions:** the benefit of adding more experts diminishes due to memory bandwidth limitations.
- **Diminishing Returns:** When more experts are active, the performance gains from adding more total experts flatten out,

- Optimize the total parameter budget rather than simply maximizing the expert count.
- Performance gains from adding more experts can be outweighed by the costs of communication overhead and memory limitations.

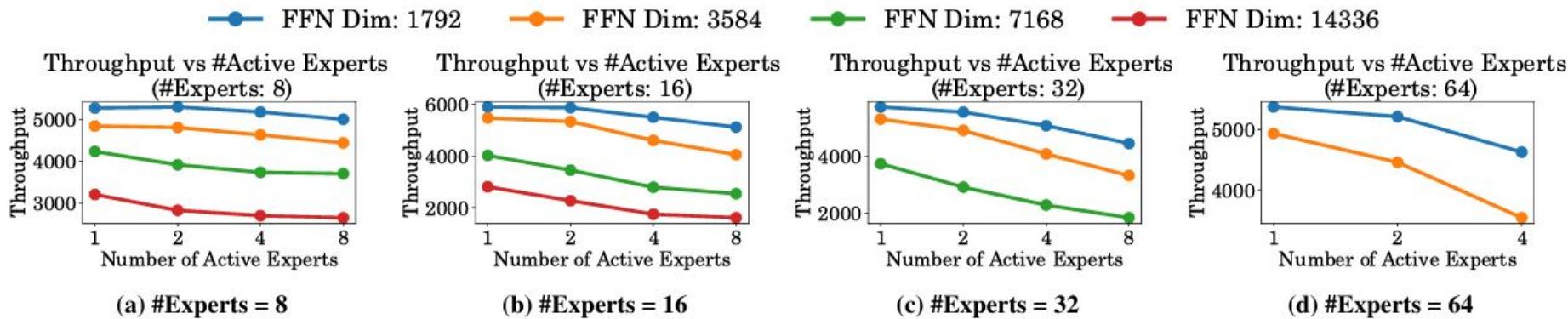


Batch Size 16 and Input/Output Length 2048 on 4 H100 GPUs on Mixtral-8x7B

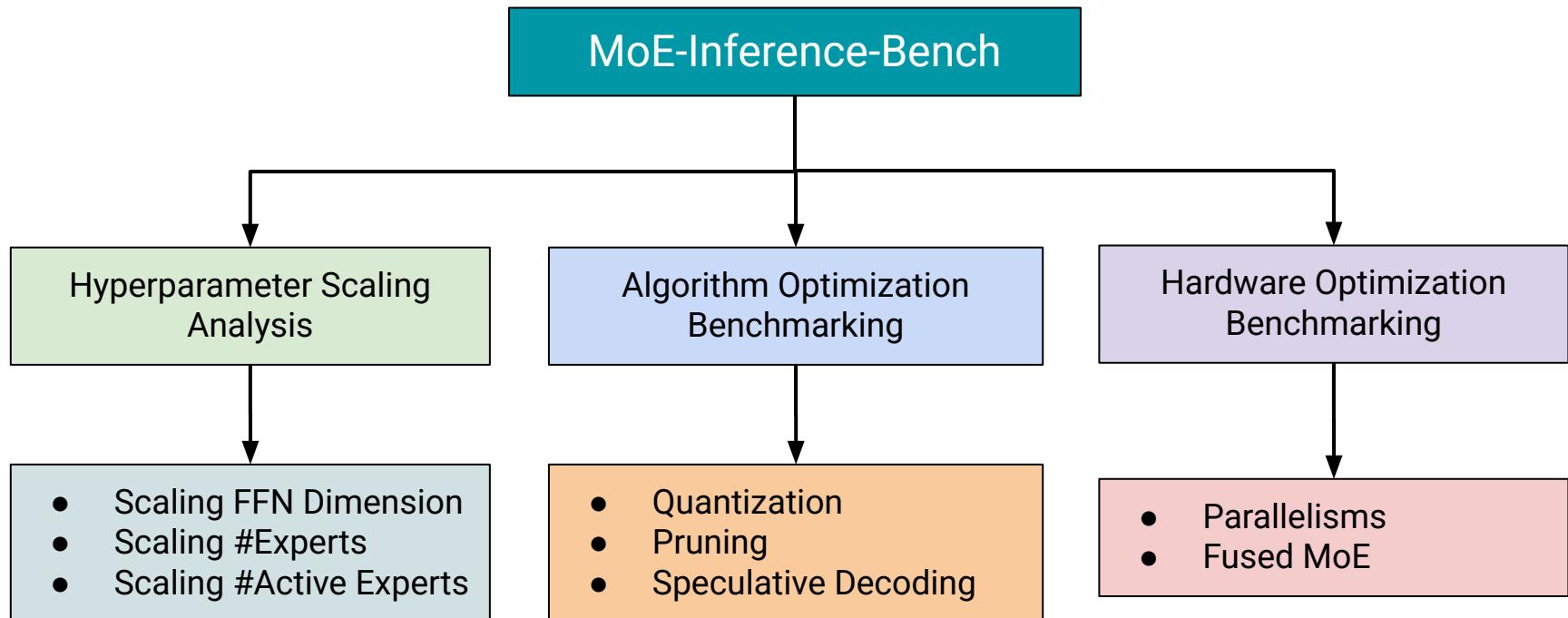
Scaling Number of Active Experts

- Throughput consistently decreases as the number of active experts increases
- The performance gap between 1 and 8 active experts is more pronounced at larger FFN dimensions
- Managing the number of active experts is a primary lever for optimizing inference performance

Optimal performance, especially with large FFN dimensions, using fewer active experts is critical to avoid significant performance degradation and out-of-memory issues.

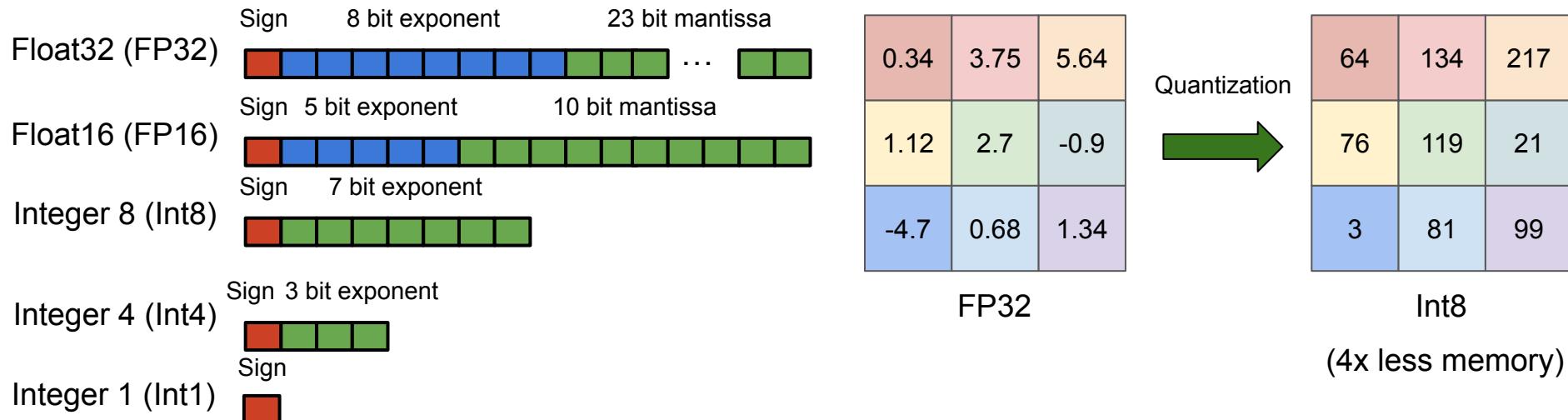


Batch Size 16 and Input/Output Length 2048 on 4 H100 GPUs on Mixtral-8x7B



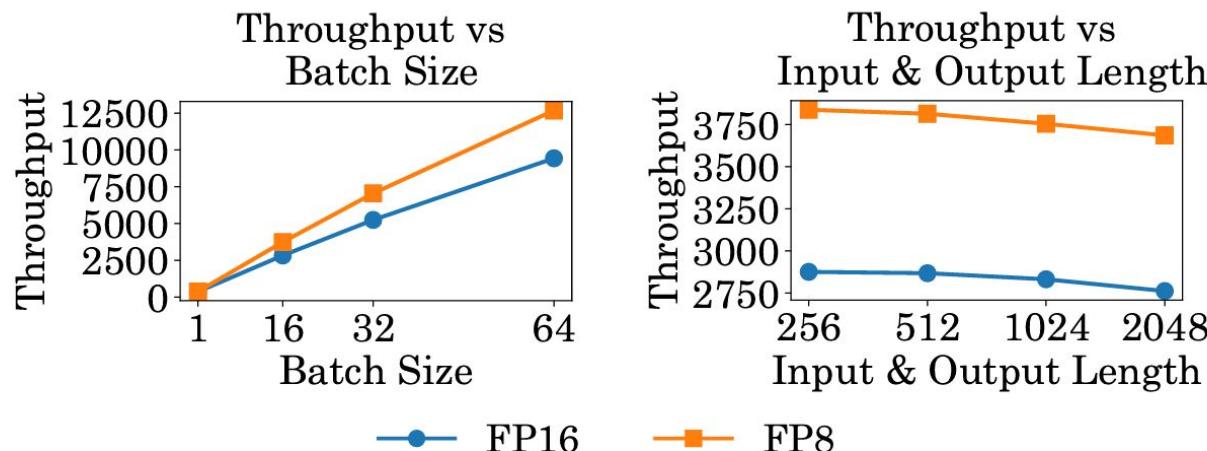
MoE Quantization

- Quantization reduces model size and memory footprint by using fewer bits per parameter
- Improves efficiency and speed, with a trade-off between precision and accuracy
- More the precision/bit-width, more the accuracy and vice versa



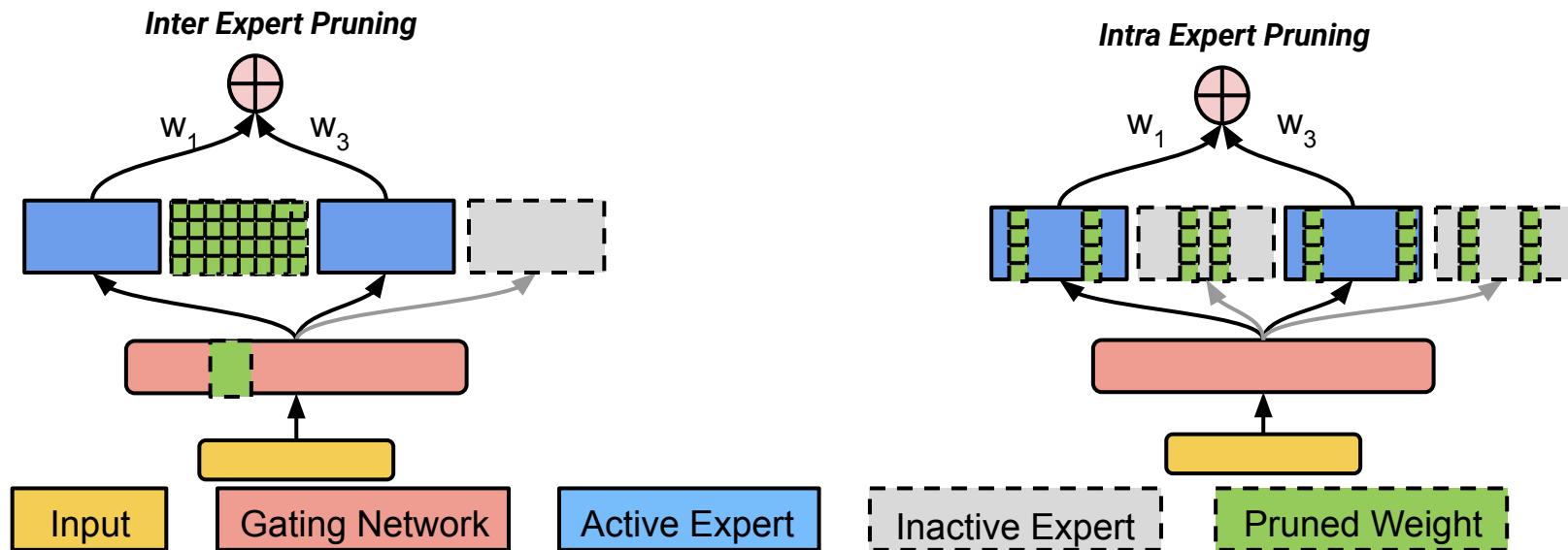
- **FP8 precision consistently outperforms FP16** on the H100 GPU across all batch sizes and sequence lengths, achieving up to 25–30% higher throughput.
- The performance gap widens with larger batch sizes, highlighting FP8's superior scalability and efficiency.

These results demonstrate FP8's strong potential for boosting computational and memory efficiency in large-scale inference workloads.



Comparison of Mixtral-8x7B with FP16 and FP8 on Nvidia H100 GPUs

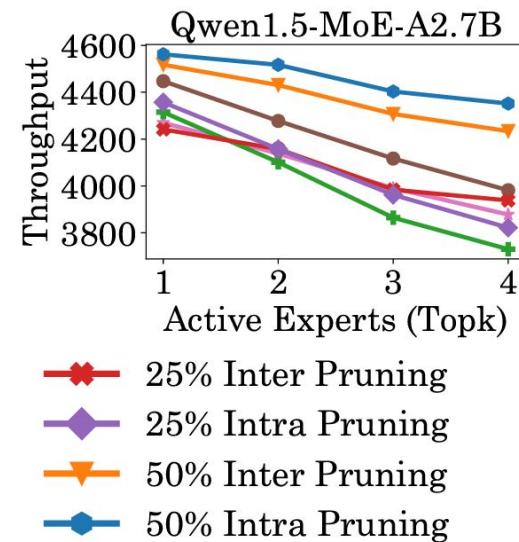
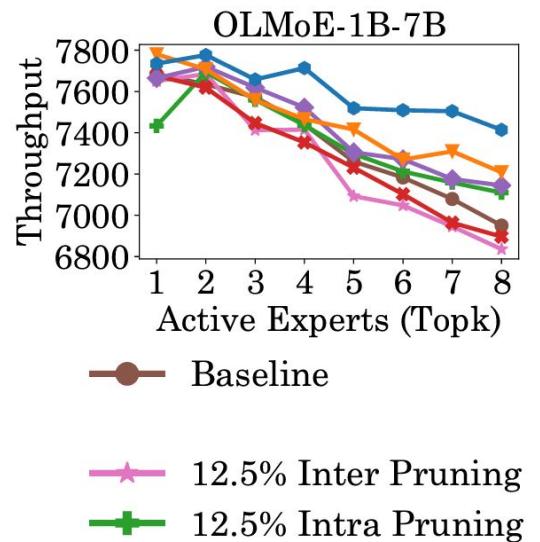
- **Inter-expert pruning** removes an entire expert along with their routing weights, reducing memory while keeping the same number of active experts during inference
- **Intra-expert pruning** reduces the FFN Dimension inside each expert, keeping the number of experts unchanged but lowering the computation per expert.
- In our experiments, we apply pruning ratios of {12.5%, 25%, 50%}



MoE Pruning Benchmarking

- Pruning Impact Varies by Model
- Small pruning percentages (12.5% or 25%) can sometimes lead to a decrease in throughput

Significant performance gains can be achieved with a high pruning ratio (e.g., 50%), while lower levels of pruning may not be beneficial and can even be detrimental to throughput.

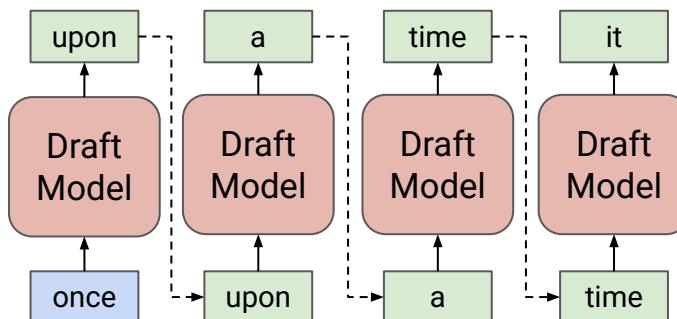


4 H100GPUs
Batch Size 16
Input/output Length of 2048

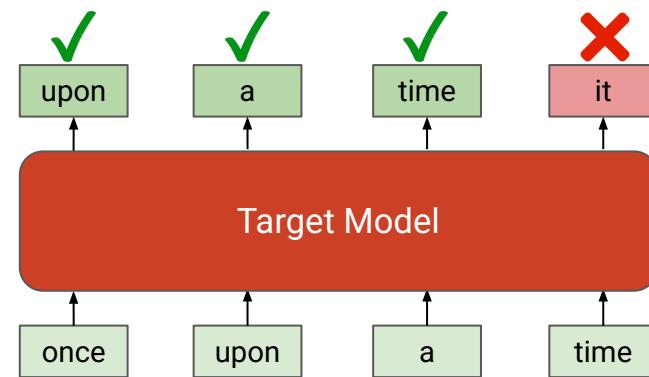
Speculative Decoding

Speculative decoding is an optimization technique that accelerates inference in large language

- a small draft model quickly generates a sequence of several tokens
 - fast but may not be as accurate
- The larger target model then takes these draft tokens and verifies them all at once, in parallel.
- The tokens that the target model agrees with are accepted.
- If the target model rejects a token, it corrects it, and the generation process continues from that point.



(a) Draft Model Token Generation

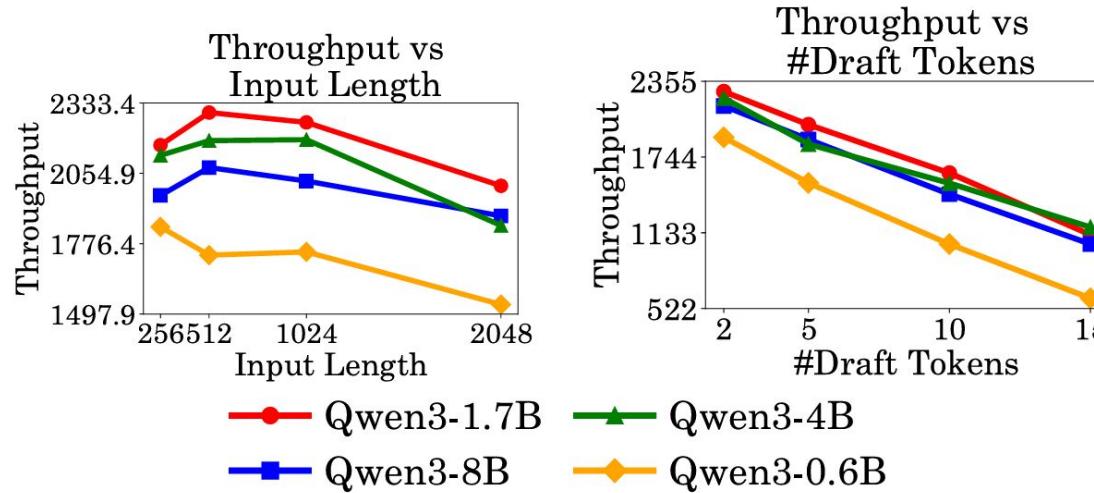


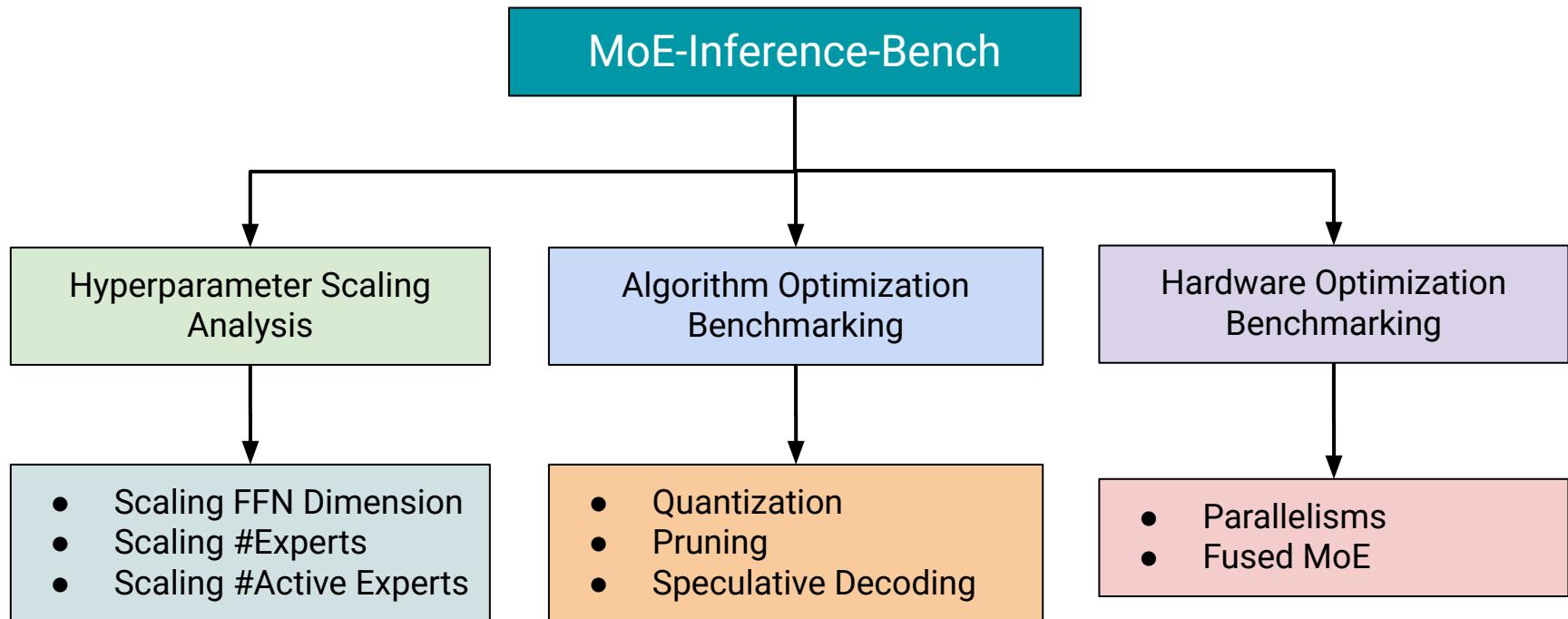
(b) Target Model Parallel Verification

Speculative Decoding Benchmarking

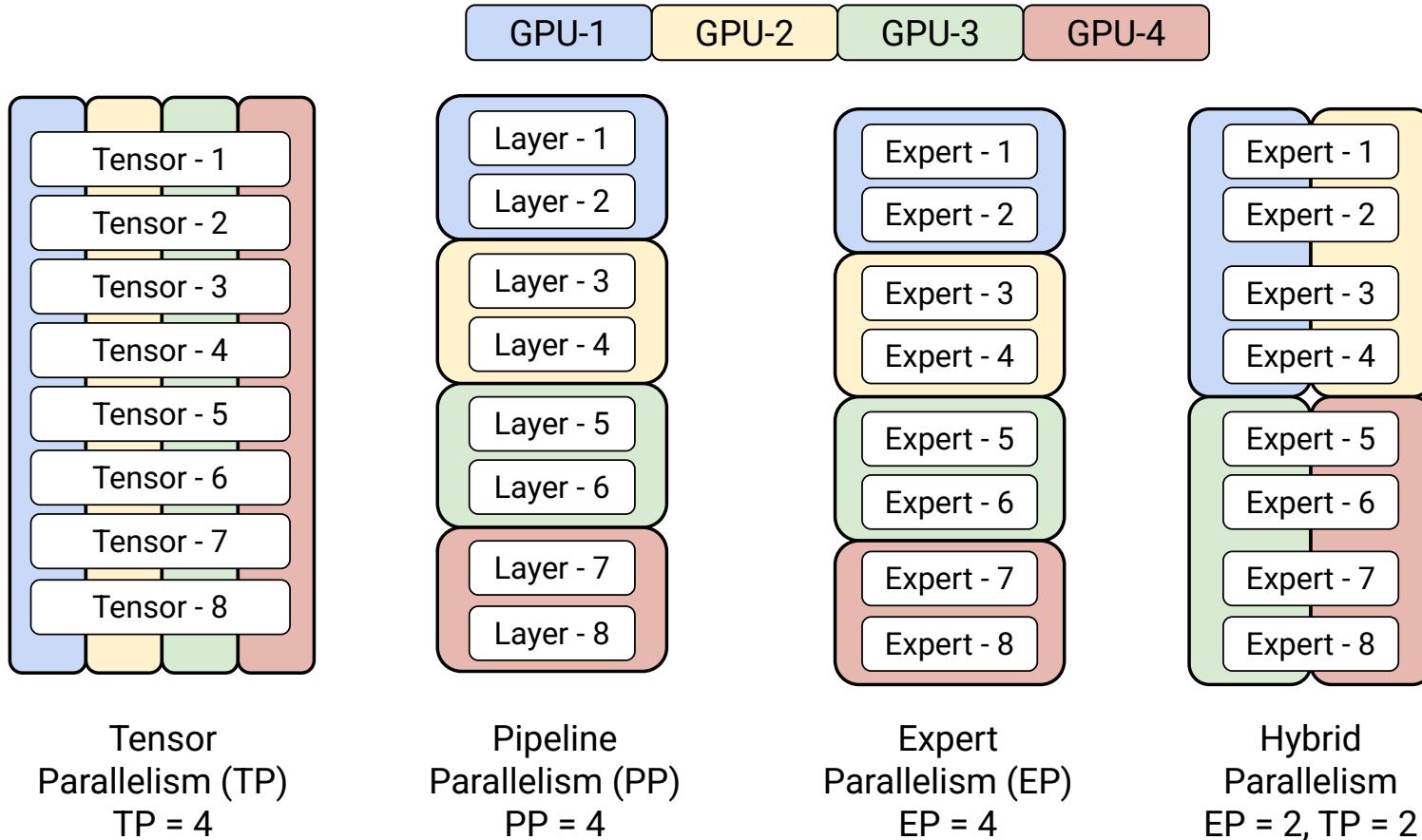
- **Optimal Draft Model Size:** A medium-sized draft model provides the best throughput and outperforms the largest draft model at shorter input lengths.
- **Scalability with Input Length:** Throughput decreases as input length increases for all draft models
- Impact of Draft Tokens: Increasing the number of draft tokens consistently reduces throughput due to higher validation overhead.

**A medium-sized draft model offers the best balance of accuracy and efficiency
both very small and very large draft models can lead to slower performance.**



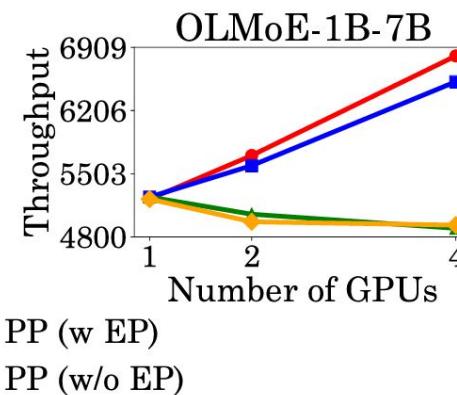
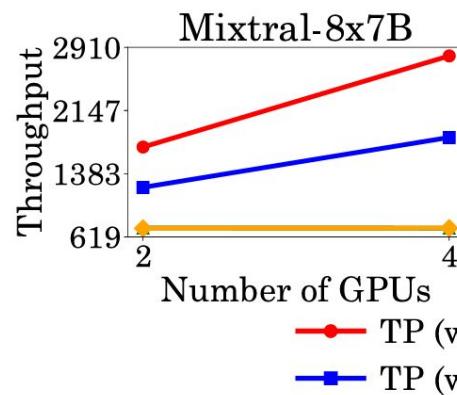


GPU Parallelism

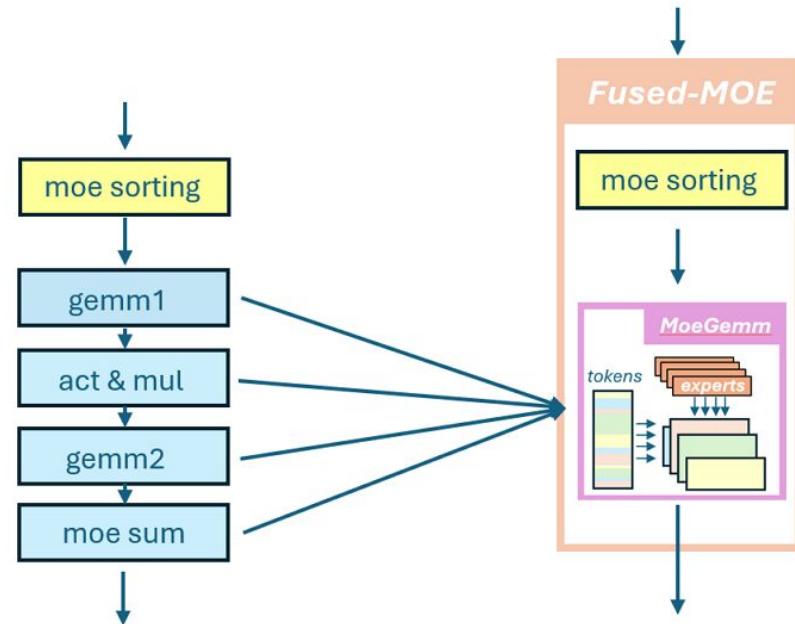


GPU Parallelism Benchmarking

- **TP without EP Wins:** Delivers the highest throughput, achieving over 2x performance gains from 1 to 4 GPUs on H100.
- **TP with EP:** Shows lower scaling efficiency compared to TP alone.
- **PP Performance:** Pipeline Parallelism (PP) with or without EP shows minimal throughput improvement, indicating poor scalability.
- **Why TP Dominates:** Tensor Parallelism over the entire model utilizes all GPU devices effectively, while EP and PP often result in resource underutilization.



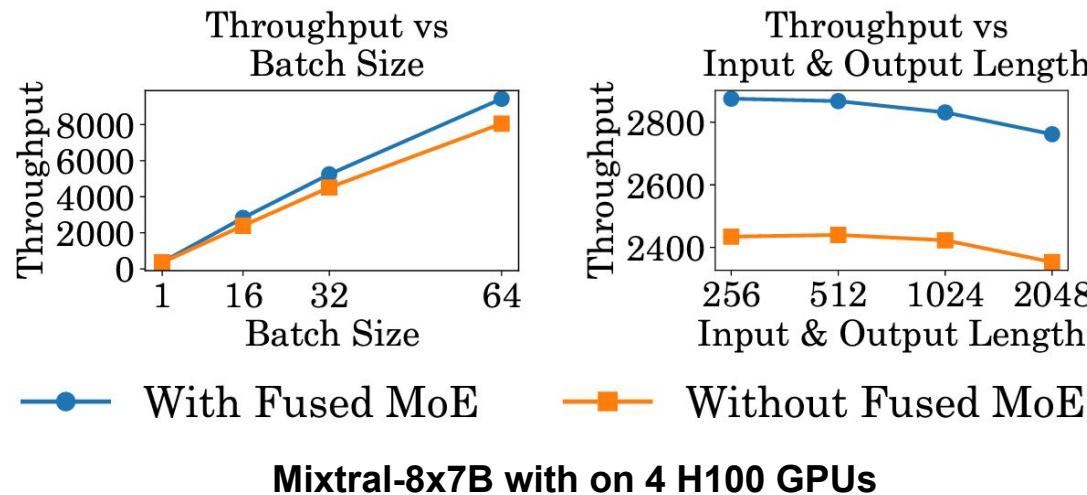
- **Kernel Fusion:** Combines multiple matrix operations into a single, efficient GPU kernel.
- **Performance Benefits:** Improves runtime performance and reduces memory overhead, critical for models with many experts.
- **Advanced Techniques:** Leverages Triton or CUTLASS GEMM with tensor parallelism support for efficient token routing.
- **Key Advantages:** Higher throughput, better resource utilization, and hardware/quantization compatibility for both research and production.



Fused MoE Benchmarking

- Fused MoE consistently **outperforms** the non-fused version
- **Scalability with Batch Size:** advantage of Fused MoE becomes more significant as the batch size increases
- Resilience to **Longer Sequences**

Kernel fusion is an effective optimization to boost throughput and maintain efficiency as computational and memory demands increase.

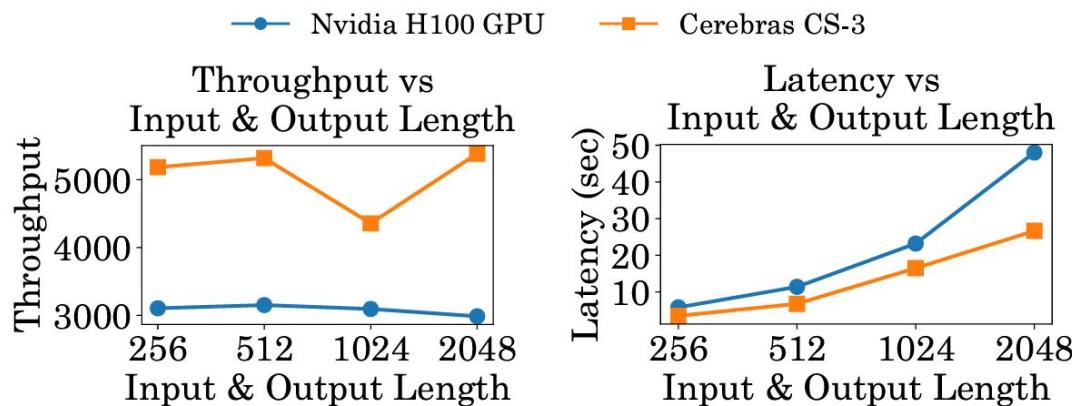


Comparing Accelerators - Nvidia H100 GPU vs Cerebras CS-3

- **Latency Advantage of CS-3:** The Cerebras CS-3 demonstrates significantly lower latency compared to the Nvidia H100 GPU, especially as sequence lengths increase.
- **Scalability with Sequence Length:** The H100 shows a sharp increase in latency beyond 1024 tokens, whereas the CS-3's latency grows more gradually.

CS-3 WSE provides orders of magnitude more memory bandwidth

enables rapid inference pipelining, which is only slightly slowed by infrequent cross-node communication.

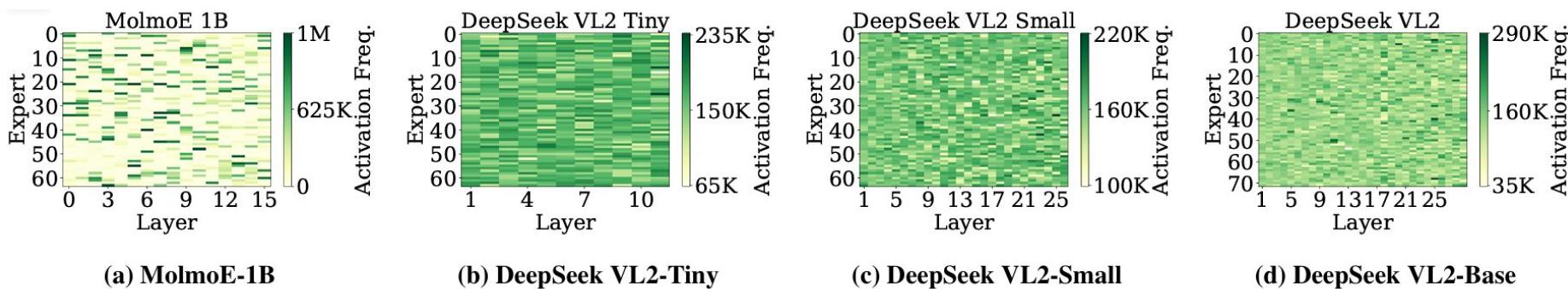


Single H100 GPU and a single CS-3 system Llama-4-Scout-17B-16E

MoE Load Balancing

- **DeepSeek-VL2:** Exhibits relatively uniform activation patterns across experts and layers.
- **MolmoE-1B:** Shows sparse activation patterns with certain experts triggered far more frequently, reaching up to 1M activations compared to DeepSeek-VL2's peak of 290K.
- Auxiliary Loss for Balance: DeepSeek-V2 incorporates auxiliary loss during training to ensure even expert utilization.

Activation frequency alone is not a reliable metric for expert importance in well-balanced models.

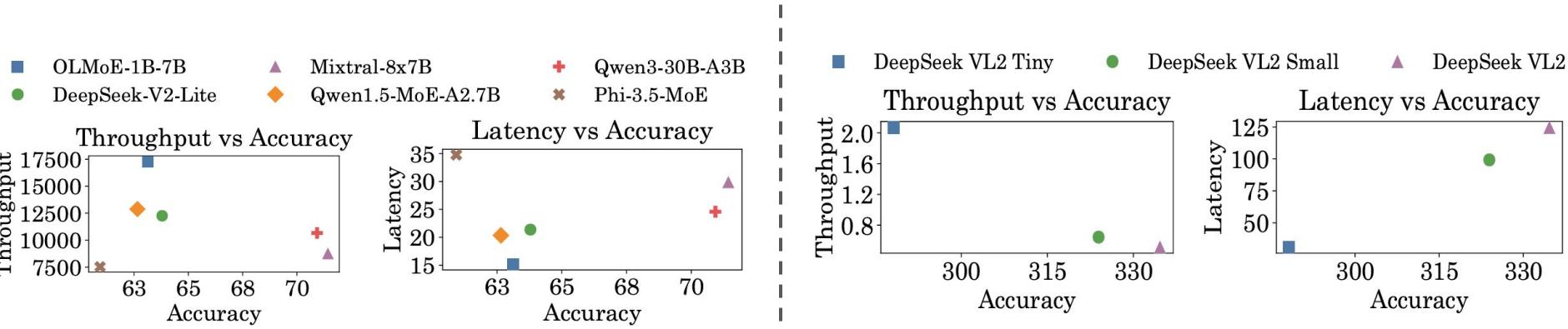


MolmoE and DeepSeek VL2 Models on MME Task

Comparing MoE Models

- **Performance vs. Efficiency:** There's a clear trade-off between accuracy and inference speed.
- Qwen3-30B-A3B and Mixtral-8x7B provide the best accuracy but come with a significant cost in terms of higher latency and lower throughput
- DeepSeek-V2-Lite and Qwen1.5-MoE-A2.7B occupy a middle ground
- Phi-3.5-MoE shows the lowest throughput and highest latency of the group.

There is a consistent performance-efficiency frontier in MoE models. Smaller models excel in throughput and latency at the cost of accuracy, while larger models dominate in accuracy but are less efficient.



Conclusion

The performance of Mixture-of-Experts models is not just about scale; it's about the intricate balance of hyperparameters, parallelism strategies, and optimization techniques.

- MoE Hyperparameters:
 - Number of active Experts and FFN Dimension are Critical
- Quantization and Pruning
 - Using lower precision, such as FP8, can significantly boost throughput without a major impact on quality
 - Aggressive pruning can improve throughput, while low levels of pruning may actually hurt performance.
- Parallelism Strategies Matter:
 - Tensor Parallelism (TP) is more effective than Expert Parallelism (EP) or Pipeline Parallelism (PP)
 - Expert Parallelism (EP) can be limited by load-balancing issues
- Advanced Inference Techniques:
 - Speculative Decoding: The choice of the draft model is key.
 - Fused MoE: This technique consistently improves throughput by reducing memory transfers and overhead