NEURAL INFORMATION PROCESSING SYSTEMS

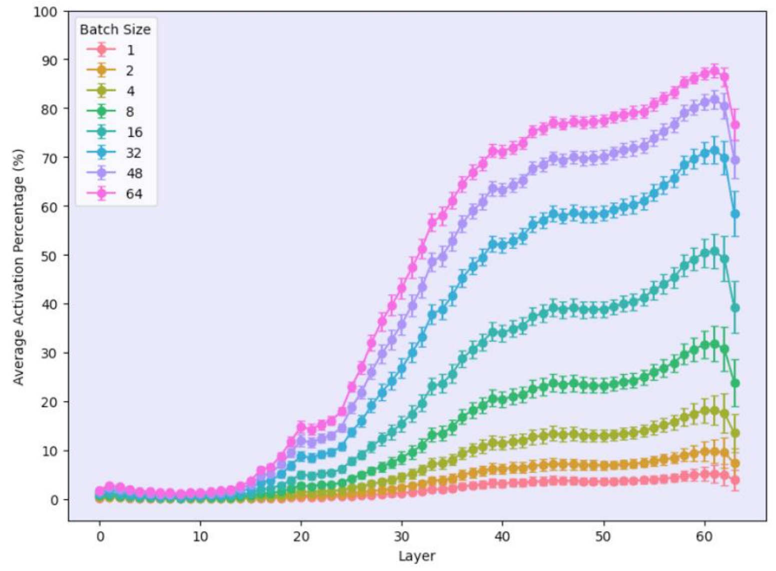
Polar Sparsity: High Throughput LLM Inferencing with Scalable Contextual Sparsity

Susav Shrestha, Brad Settlemyer, Nikoli Dryden, Narasimha Reddy

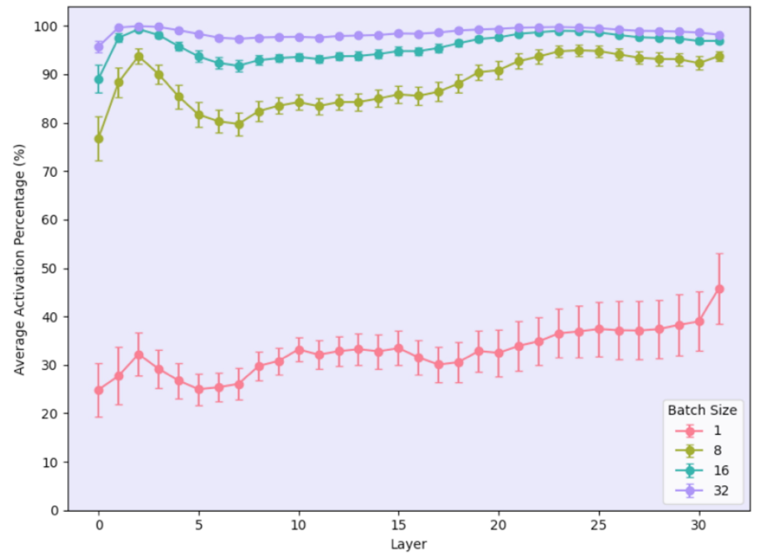
TEXAS A&M UNIVERSITY NVIDIA Lawrence Livermore National Laboratory

Background and Motivation

- **Contextual activation sparsity**:
  - A small, input-dependent subset of neurons/heads are required to produce nearly identical outputs to the full dense model for a given input context.
  - Effective for single-query inference.
- **Breakdown under batching**:
  - As batch size increases, union activation across tokens causes the effective sparsity in MLPs to collapse, reducing their acceleration benefit.



(a) OPT 66b batch neuron activations



(b) ReLU Llama-2-7b batch neuron activations

NEURAL INFORMATION PROCESSING SYSTEMS

Polar Sparsity

- As batch size increases, **the decode latency is bottlenecked by attention computation**.
- In large-batch inference, the cost of accessing model parameters is largely amortized, since the entire batch utilizes the same model weights.
- Each batch has a unique KV cache, making attention bottlenecked by IO.

$$KV\ memory = batchsize * 2 * seqlen * kvheads * headdim * precision * layers$$

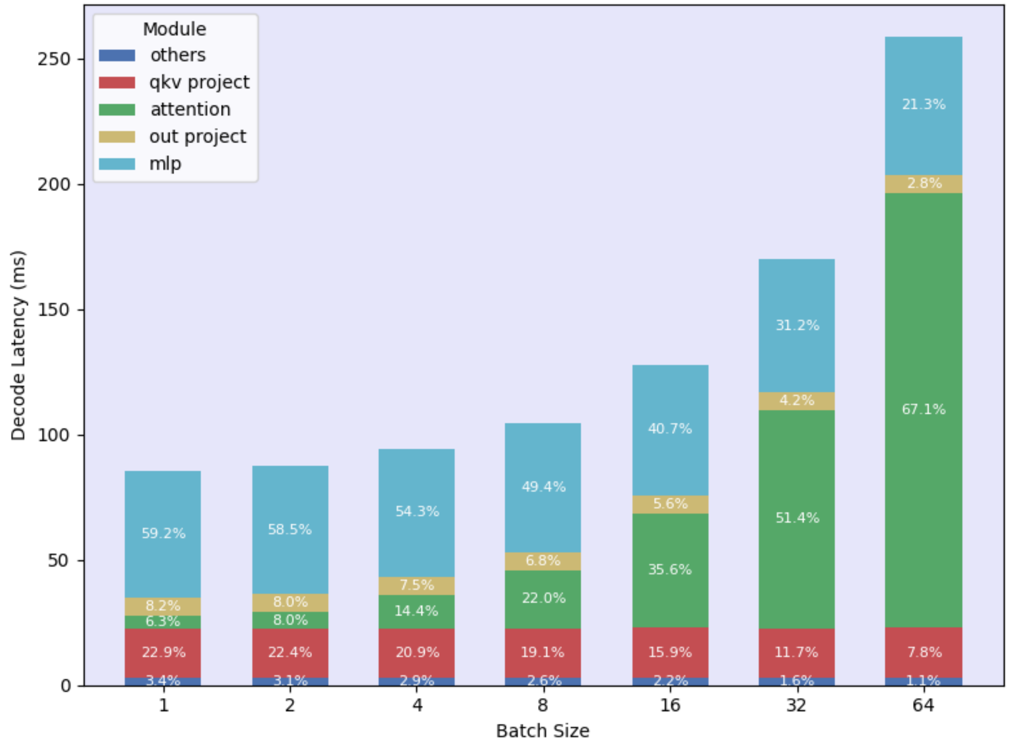


Figure 1 (a). As batch size increases, attention layers dominate end-to-end latency. OPT 66b model, seq len 1920.

Polar Sparsity

- Polar Sparsity Insight:
  - As batch size and sequence length increase, the dominant source of sparsity shifts from MLP layers to Attention layers.
- Two efficiency regimes emerge:
  - **MLP sparsity** → excels at small-batch, low-latency inference (single-query settings).
  - **Attention head sparsity** → sustains efficiency in large-batch, high-throughput decoding.
- Key takeaway:
  - Optimizing for attention head sparsity is critical for accelerating LLM inference at scale.

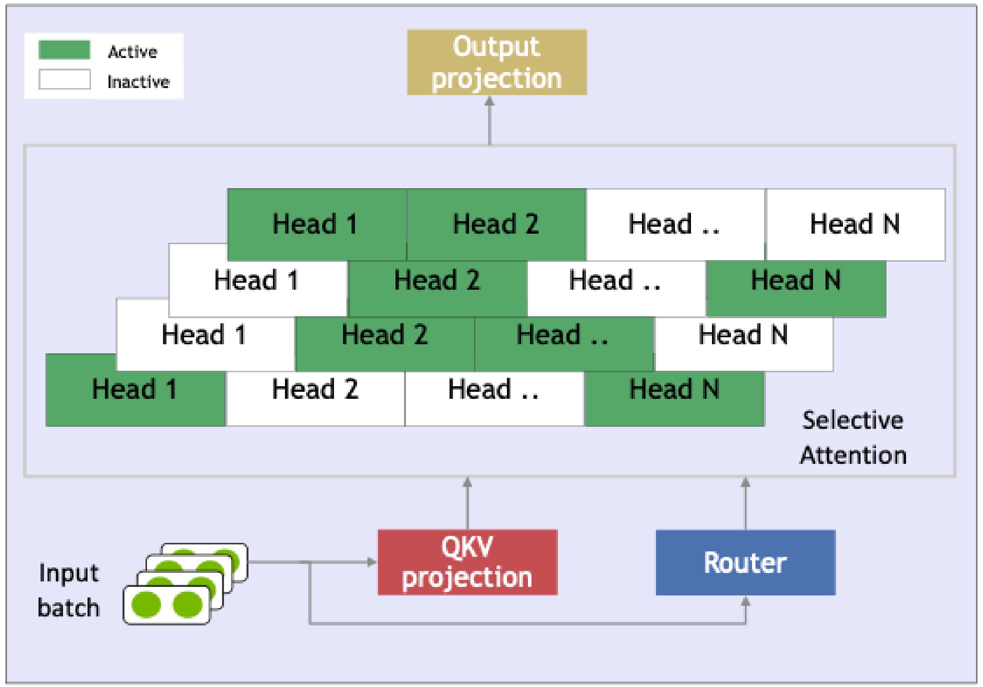


Figure 1 (b). Selective Head Attention only activates the most critical heads for each request and accelerates high throughput inference.

Selective Head Attention

- Selective Head Attention:
  - Dense QKV, Output projection and Selective Attention Head computation.
  - Only the KV cache of the activated heads are accessed for computation.
- Head routing mechanism:
  - Lightweight routers predict the top-k attention heads with highest output norms for each query.
  - Routers are single-layer feed-forward classifiers, trained per layer to approximate head importance.
- Per-request adaptivity:
  - Head routing is performed independently for each input, allowing different batches to activate completely distinct head subsets.

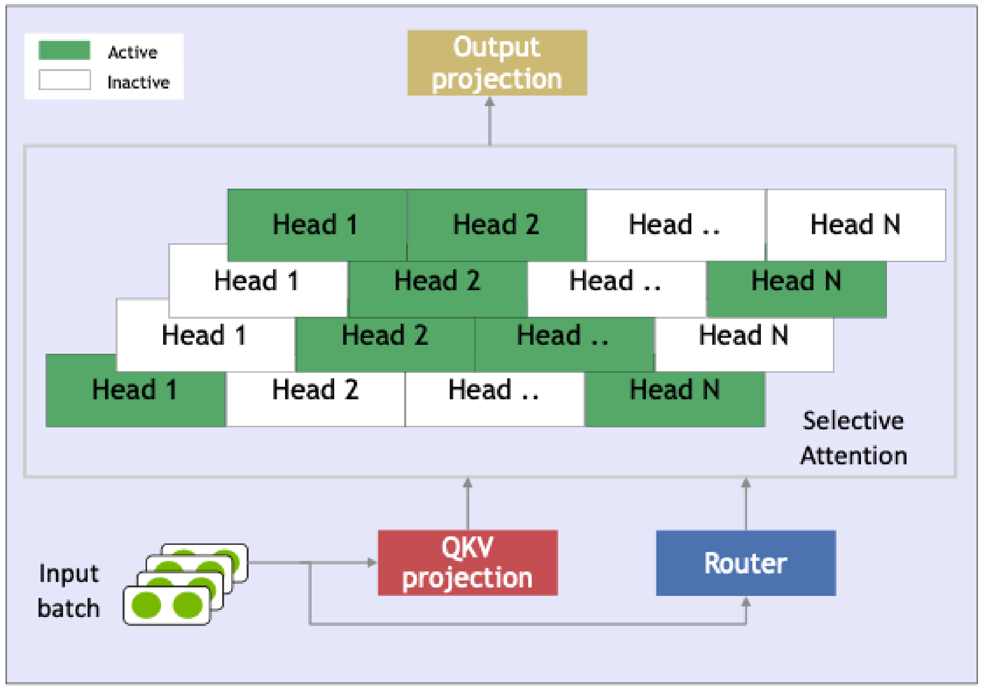


Figure 1 (b). Selective Head Attention only activates the most critical heads for each request and accelerates high throughput inference.

NEURAL INFORMATION PROCESSING SYSTEMS

Selective Head Attention Kernel

Algorithm 1 Selective Head FlashAttention (Decode)

```
Require: $\mathbf{Q} \in \mathbb{R}^{B\times H\times 1\times d}$, $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{B\times H\times N_{kv}\times d}$, batch_head_index $\in \mathbb{Z}^{B\times \text{top\_k}}$, $M_{SRAM}$, $s = 1/\sqrt{d}$
    Output: $\mathbf{O} \in \mathbb{R}^{B\times H\times 1\times d}$ (written to HBM)
 1: Determine target batch index $b$ and top-k index $k$ assigned to this unit from the grid dimensions.
 2: head_idx $\leftarrow$ batch_head_index$[b, k]$                ▷ Get the actual head index to compute
 3: $B_c = \lfloor M_{SRAM}/(4d) \rfloor$; $(\mathbf{O}_{acc}, l_{acc}, m_{acc}) \leftarrow (\vec{0}, 0, -\infty)$; $T_c = \lceil N_{kv}/B_c \rceil$
 4: Load $\mathbf{q} \in \mathbb{R}^{1\times d}$ from $\mathbf{Q}[b, \text{head\_idx}, 0, :]$      ▷ Get the activated query vector for the batch
 5: for $j = 1$ to $T_c$ do
 6:     $k_{start} = (j-1)B_c$, $k_{end} = \min(jB_c, N_{kv})$; Load $\mathbf{K}_j, \mathbf{V}_j$ from $\mathbf{K}, \mathbf{V}[b, \text{head\_idx}, k_{start} : k_{end}, :]$
 7:     $\mathbf{S}_j = s(\mathbf{q} @ \mathbf{K}_j^T)$; $\tilde{m}_j = \max(\mathbf{S}_j)$; $\tilde{\mathbf{P}}_j = \exp(\mathbf{S}_j - \tilde{m}_j)$; $\tilde{l}_j = \sum \tilde{\mathbf{P}}_j$
 8:     $m_{new} = \max(m_{acc}, \tilde{m}_j)$; $\alpha = e^{m_{acc} - m_{new}}$; $\beta = e^{\tilde{m}_j - m_{new}}$; $l_{new} = \alpha l_{acc} + \beta \tilde{l}_j$
 9:     $\mathbf{O}_{acc} \leftarrow (\alpha l_{acc} \mathbf{O}_{acc} + \beta(\tilde{\mathbf{P}}_j @ \mathbf{V}_j))/l_{new}$; $l_{acc} \leftarrow l_{new}$; $m_{acc} \leftarrow m_{new}$
10: end for
11: Write $\mathbf{O}_{acc}$ to $\mathbf{O}[b, \text{head\_idx}, 0, :]$
```

Selective Head Attention kernel achieves linear speedup with the induced head sparsity.

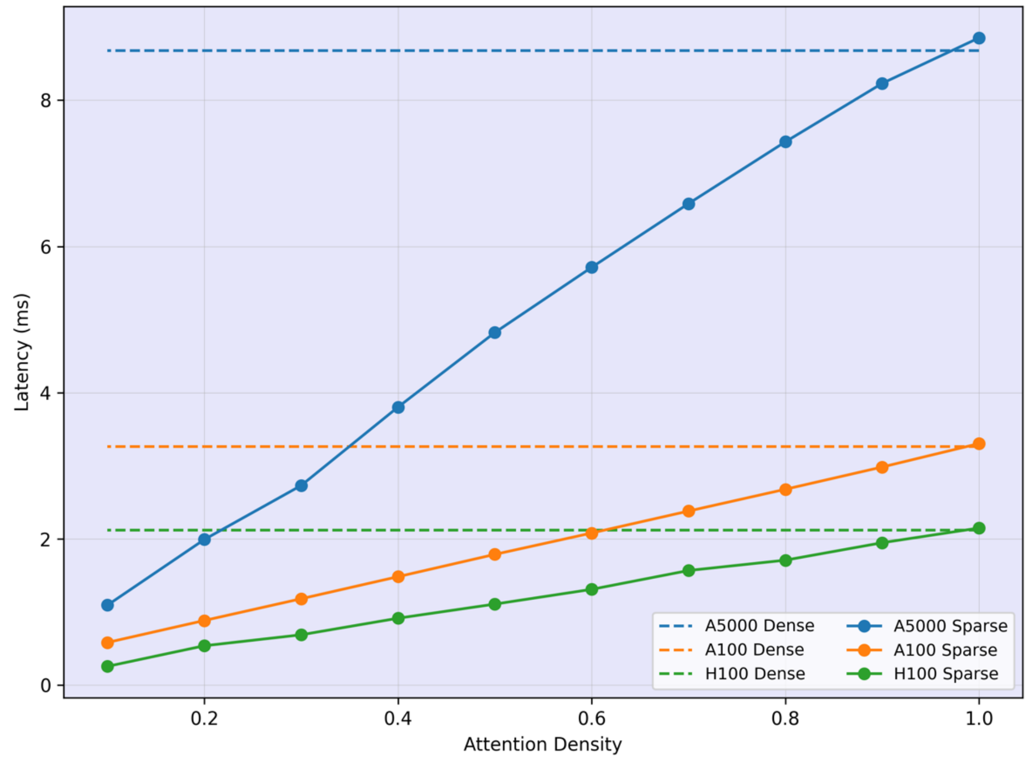


Figure 11. Selective Head Attention Kernel Performance across GPU Families. Details: MHA, 72 heads, Head dim 128, Batch size 64, Seq len 1920

NEURAL INFORMATION PROCESSING SYSTEMS

Evaluation

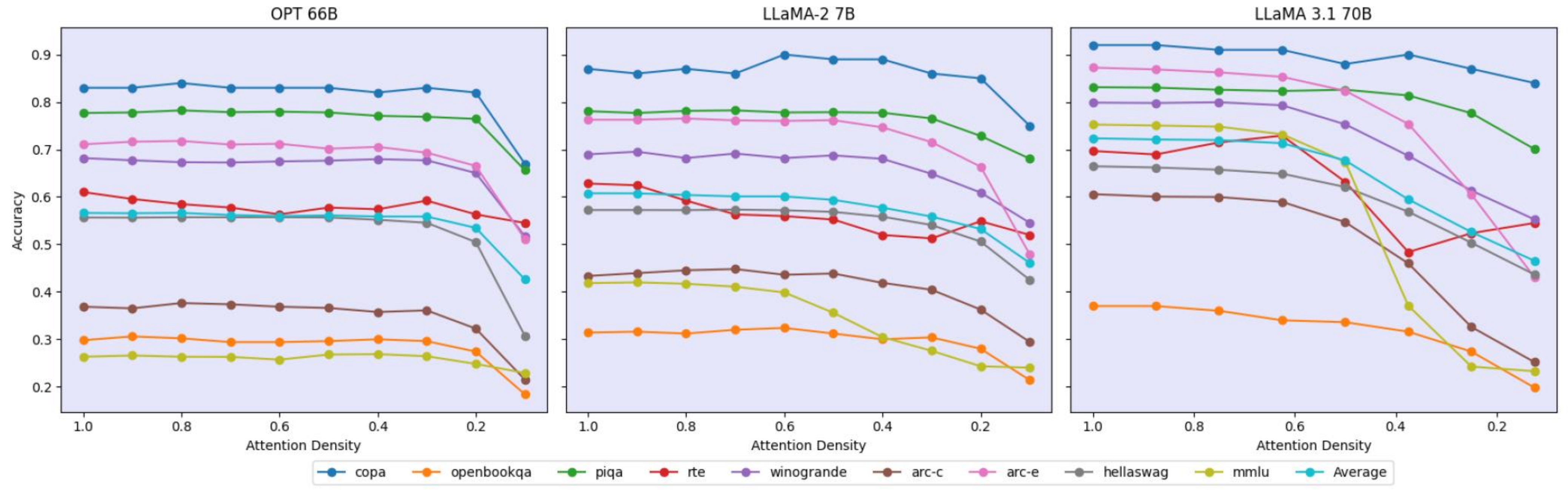
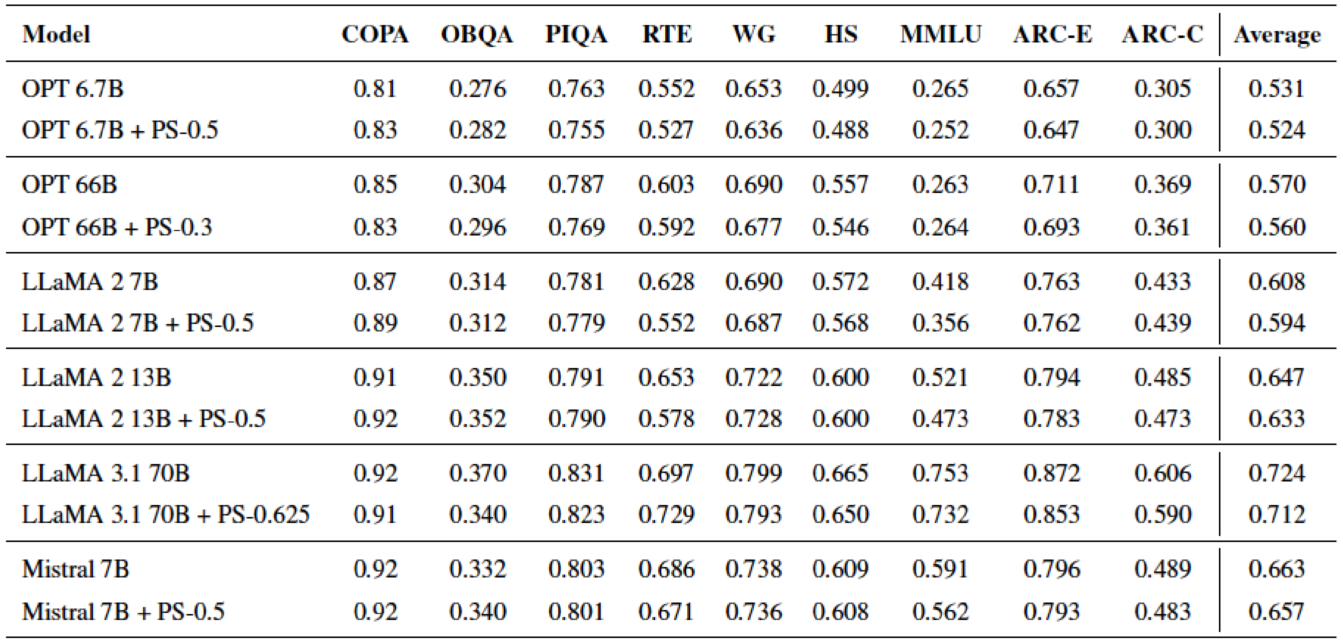


Figure 5: Accuracy vs Attention Density. **Left**: OPT 66b model with dynamic sparse MLP + Select Head Attention. **Middle**: LLaMA 2-7b model with Select Head Attention **Right**: LLaMA 3.1 70b model with Select Group Attention. Dense attention in layer 0 used for all models.

NEURAL INFORMATION PROCESSING SYSTEMS

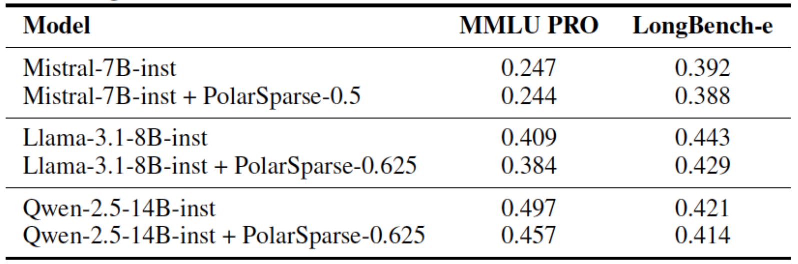
Evaluation

Table 1: LLM zero-shot evaluation at critical thresholds. Polar Sparsity (PS) is competitive with the dense baseline with average accuracy within 1%.



Model	COPA	OBQA	PIQA	RTE	WG	HS	MMLU	ARC-E	ARC-C	Average
OPT 6.7B	0.81	0.276	0.763	0.552	0.653	0.499	0.265	0.657	0.305	0.531
OPT 6.7B + PS-0.5	0.83	0.282	0.755	0.527	0.636	0.488	0.252	0.647	0.300	0.524
OPT 66B	0.85	0.304	0.787	0.603	0.690	0.557	0.263	0.711	0.369	0.570
OPT 66B + PS-0.3	0.83	0.296	0.769	0.592	0.677	0.546	0.264	0.693	0.361	0.560
LLaMA 2 7B	0.87	0.314	0.781	0.628	0.690	0.572	0.418	0.763	0.433	0.608
LLaMA 2 7B + PS-0.5	0.89	0.312	0.779	0.552	0.687	0.568	0.356	0.762	0.439	0.594
LLaMA 2 13B	0.91	0.350	0.791	0.653	0.722	0.600	0.521	0.794	0.485	0.647
LLaMA 2 13B + PS-0.5	0.92	0.352	0.790	0.578	0.728	0.600	0.473	0.783	0.473	0.633
LLaMA 3.1 70B	0.92	0.370	0.831	0.697	0.799	0.665	0.753	0.872	0.606	0.724
LLaMA 3.1 70B + PS-0.625	0.91	0.340	0.823	0.729	0.793	0.650	0.732	0.853	0.590	0.712
Mistral 7B	0.92	0.332	0.803	0.686	0.738	0.609	0.591	0.796	0.489	0.663
Mistral 7B + PS-0.5	0.92	0.340	0.801	0.671	0.736	0.608	0.562	0.793	0.483	0.657

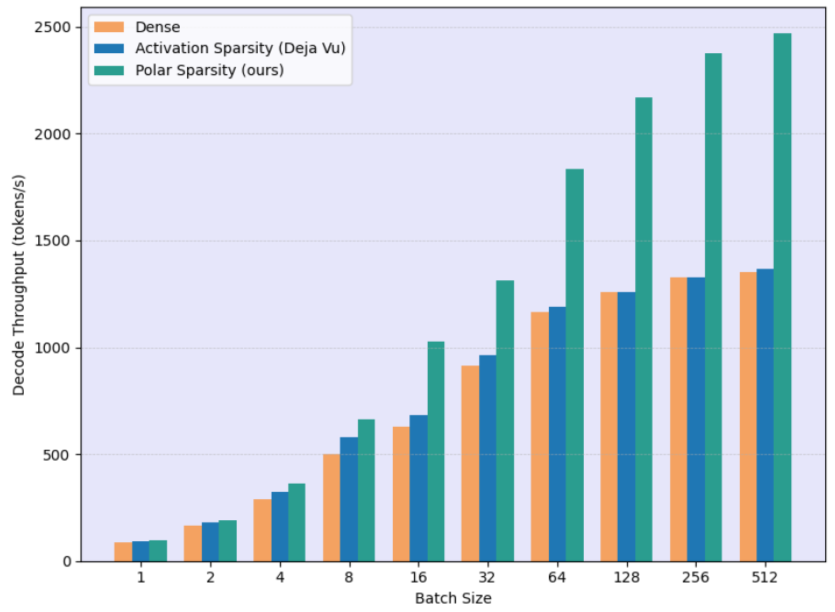
Table 2: Evaluation of instruction-tuned LLMs at critical sparsity thresholds. Polar Sparsity maintains strong performance on generative tasks.



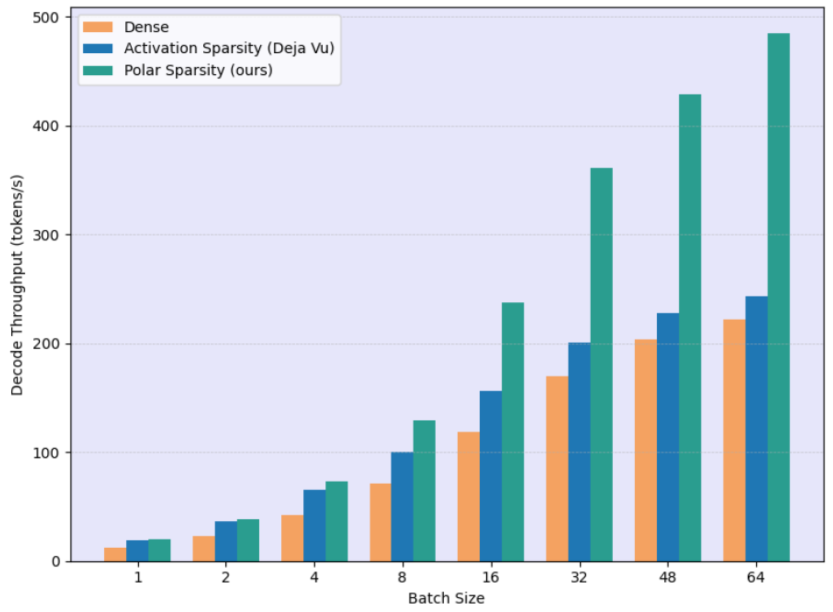
Model	MMLU PRO	LongBench-e
Mistral-7B-inst	0.247	0.392
Mistral-7B-inst + PolarSparse-0.5	0.244	0.388
Llama-3.1-8B-inst	0.409	0.443
Llama-3.1-8B-inst + PolarSparse-0.625	0.384	0.429
Qwen-2.5-14B-inst	0.497	0.421
Qwen-2.5-14B-inst + PolarSparse-0.625	0.457	0.414

NEURAL INFORMATION PROCESSING SYSTEMS

Inference Throughput



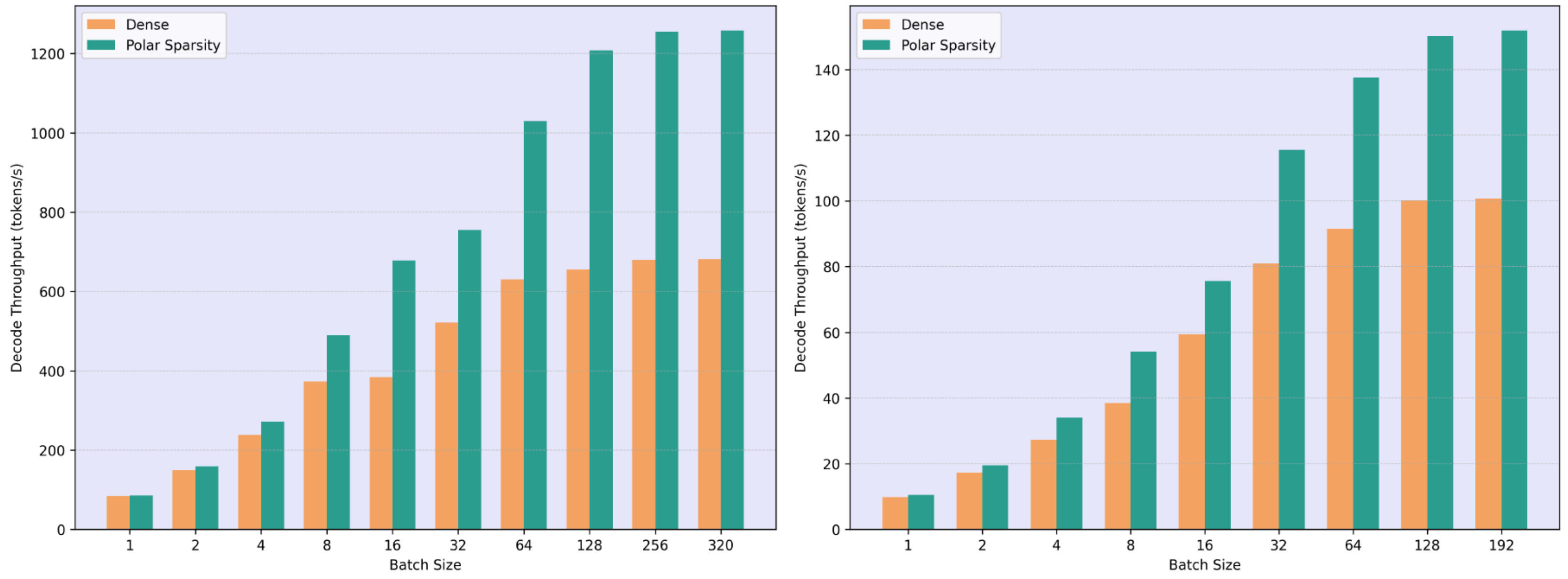
(a) OPT 6.7b



(b) OPT 66b

Figure 6: OPT models sparse decoding throughput, seq len 1920 using pipeline parallelism. (a) OPT 6.7b, critical threshold 50%. (b) OPT 66b, critical threshold 30%. Polar Sparsity delivers up to 2.2× higher throughput than dense and up to 2× than standard activation sparsity at scale.

Inference Throughput



(a) LLaMA-2-7b (b) LLaMA-3.1-70b

Figure 7: LLaMA models sparse decode throughput results using pipeline parallelism. (a) LLaMA-2-7b seq len 3968, critical threshold 50%.(b) LLaMA-3.1-70b seq len 8192, critical threshold 62.5%. Polar Sparsity delivers up to $1.85\times$ higher throughput than dense baseline at scale.

Conclusion

- **Scalable Contextual Sparsity:**
  - Demonstrated that contextual sparsity *extends to the large-batch regime*, not just single-query inference.
- **Polar Sparsity:**
  - As batch size and context length grow, sparsity importance *shifts from MLP neurons to Attention heads*.
  - MLP sparsity collapses due to union activation.
  - Head-level sparsity remains stable and *batch-invariant*.
- **Selective Head Attention (SHA):**
  - Designed *sparsity-aware GPU kernels* that compute only for activated heads and neurons, minimizing redundant memory I/O.
- **Performance Gains:**
  - Up to **2.2× throughput improvement** across different models with **<1% accuracy loss**.
  - Enables *efficient, high-throughput, and cost-effective* deployment of LLMs on large-scale inference systems.