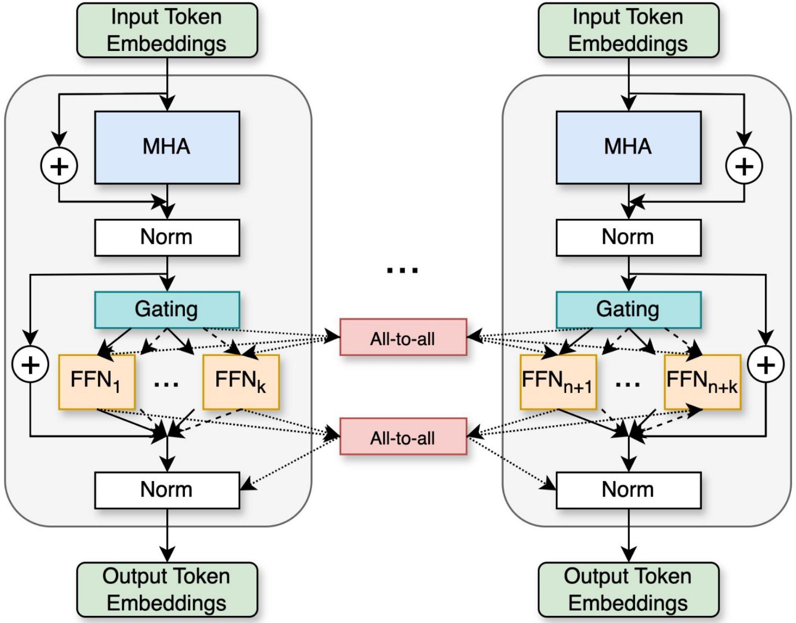
Toward Efficient Inference for Mixture of Experts

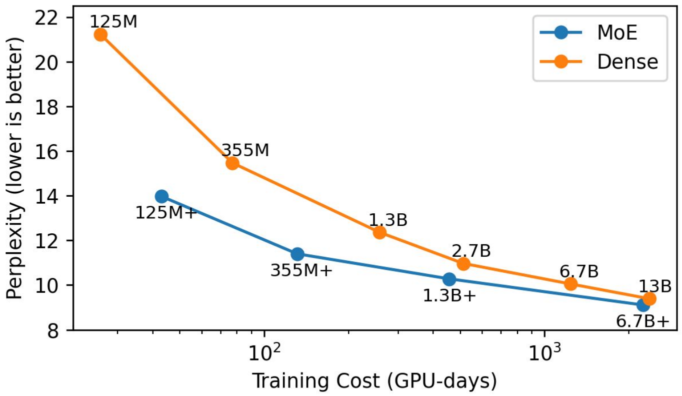
Haiyang Huang, Newsha Ardalani, Anna Sun, Liu Ke
Hsien-Hsin S. Lee, Shruti Bhosale, Carole-Jean Wu, Benjamin Lee

NeurIPS, Dec 2024

MoE Performance in Training Scenario

Mixture of Experts (MoE) models with expert parallelism

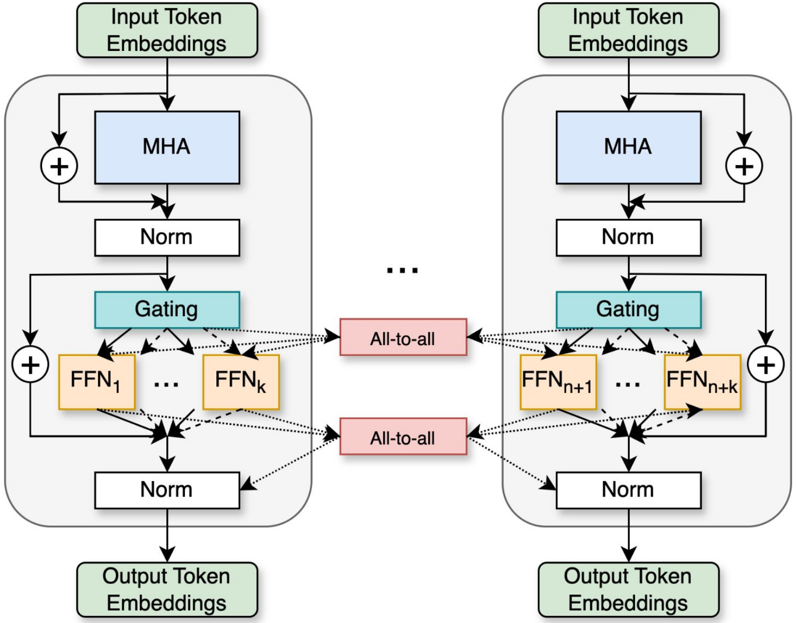


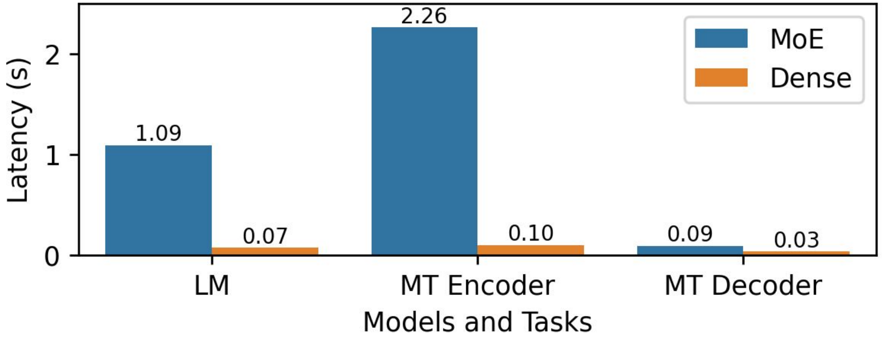


Lower **training** cost by up to 5x compared to dense Transformer

MoE Performance in Inference Scenario

Mixture of Experts (MoE) models with expert parallelism



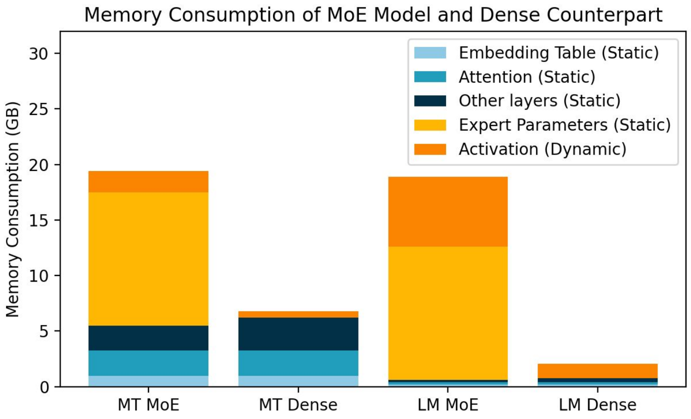


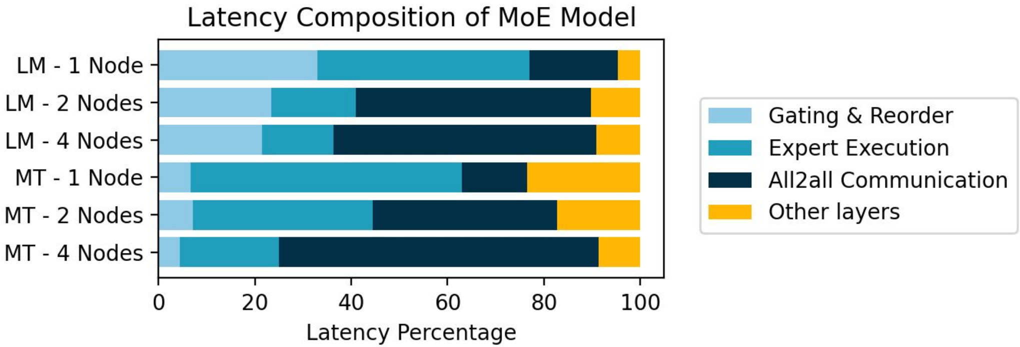
Lower **training** cost by up to 5x compared to dense Transformer.

Higher **inference** cost, 15x slower for language models and >3x slower for machine translation.

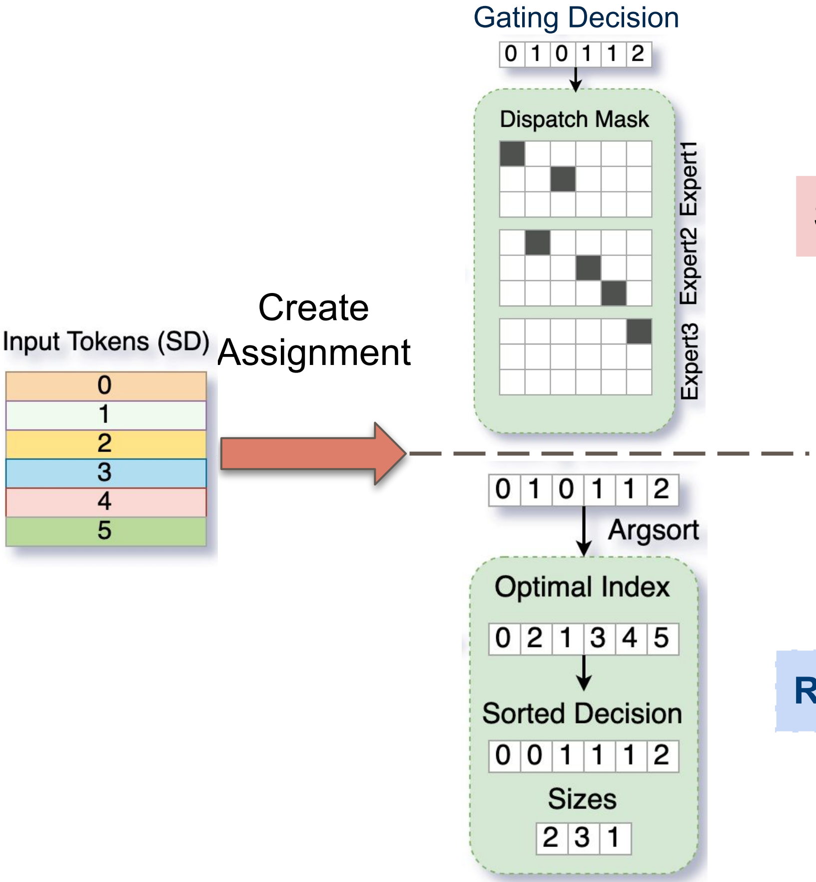
MoE Inference Latency and Memory Characterizations

- **slower** than equivalent* dense counterparts
- consumes **more memory** than equivalent* dense counterparts





Dynamic Gating: Less Memory

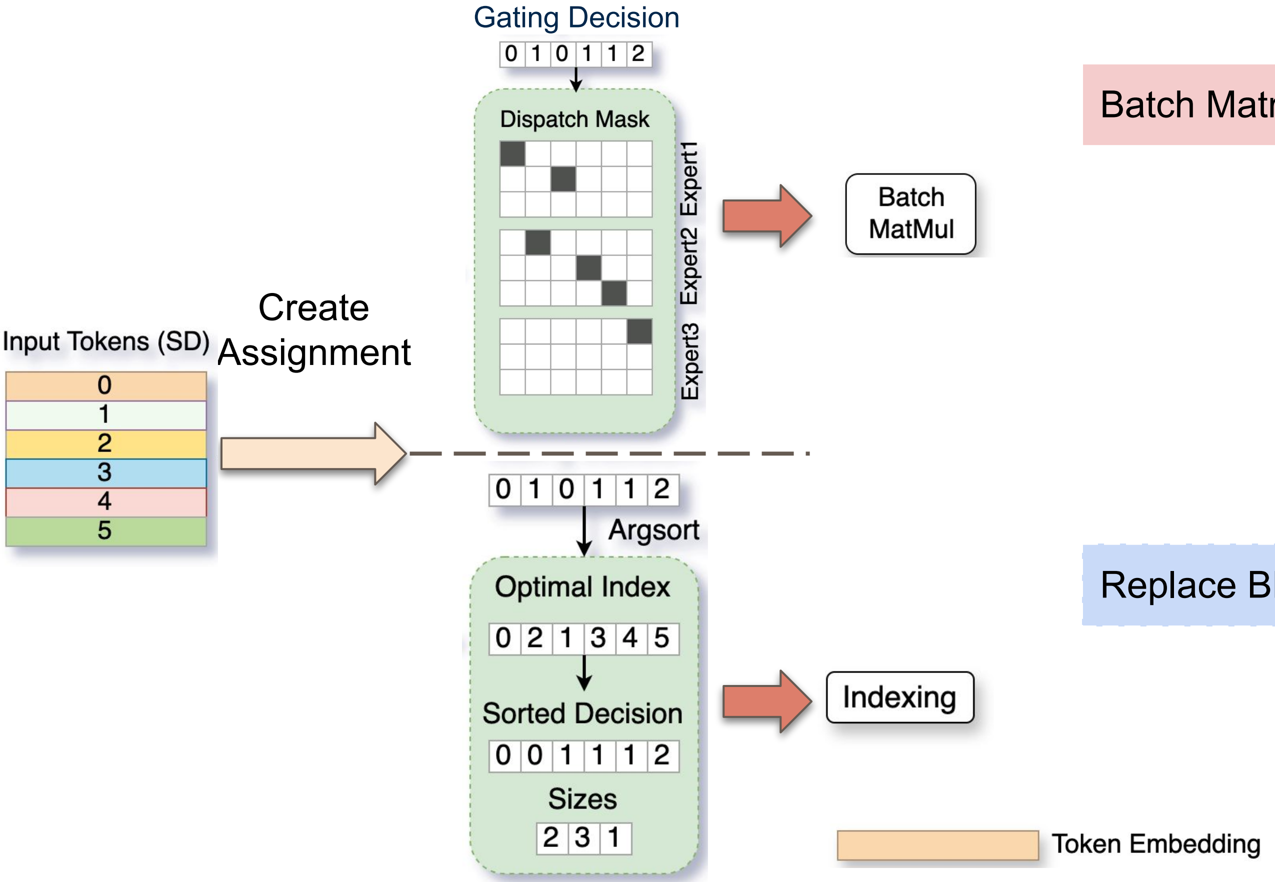


Sparse dispatch mask is **memory consuming**

Reduce memory allocation by removing the sparse mask

Token Embedding   1 in sparse mask   0 in sparse mask

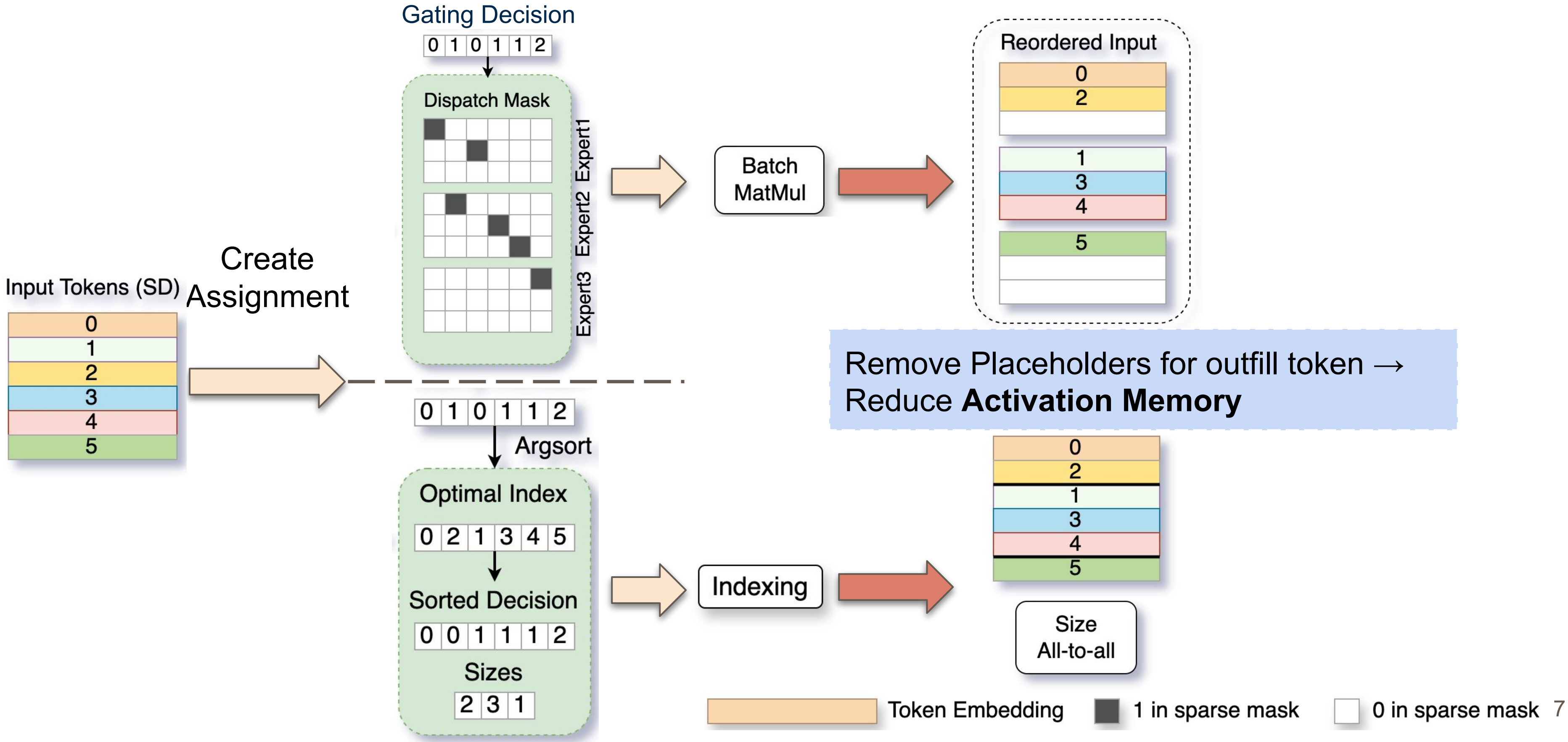
Dynamic Gating: Less Latency



Batch Matrix Multiplication is **slow**

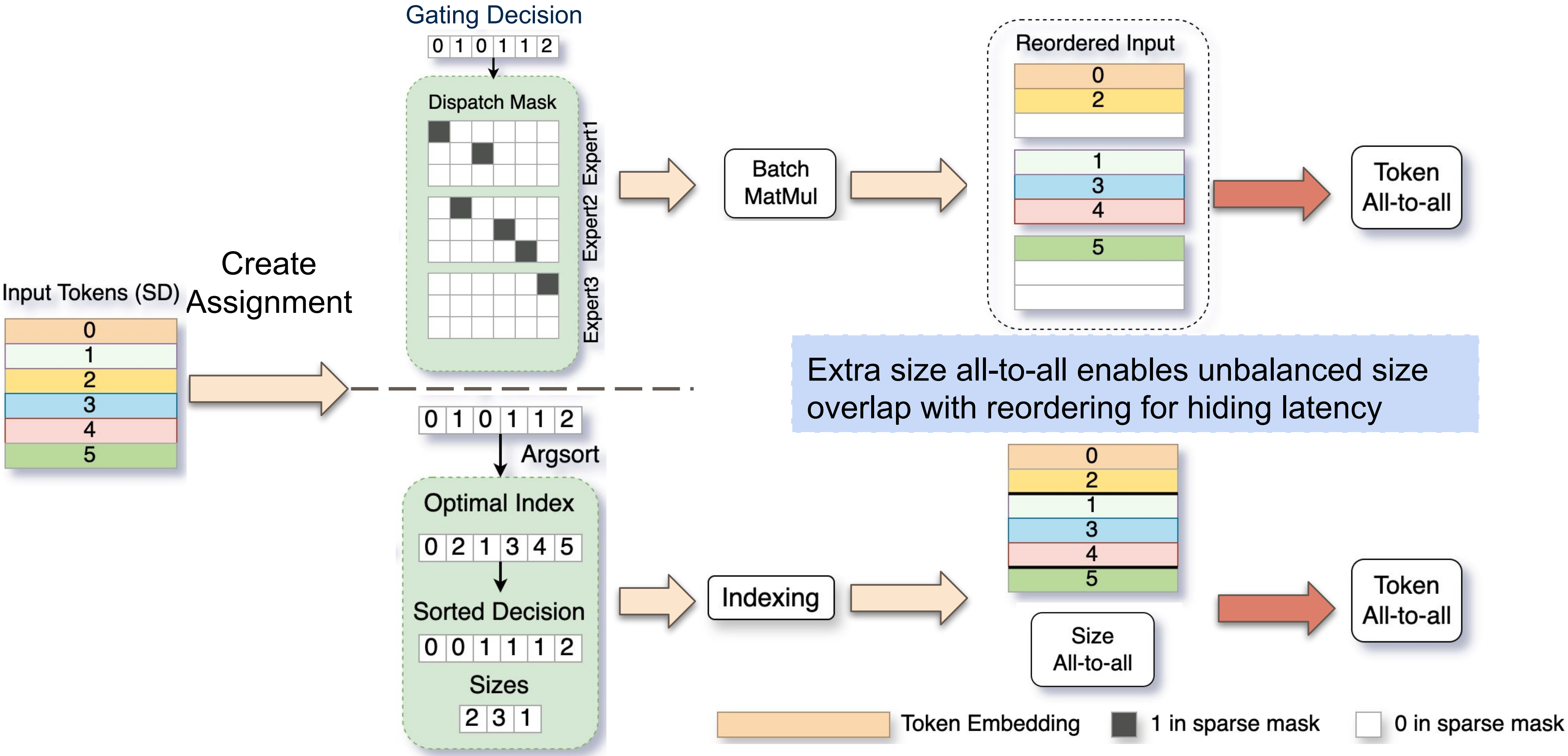
Replace BMM with fast **indexing** operation

Dynamic Gating: No More Placeholders



Remove Placeholders for outfill token →
Reduce **Activation Memory**

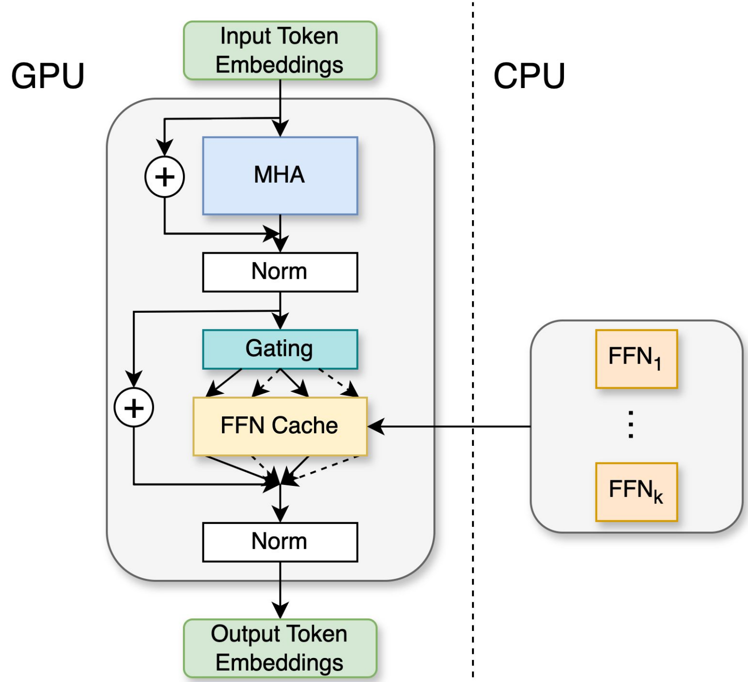
Dynamic Gating: Dynamic All-to-all



Extra size all-to-all enables unbalanced size overlap with reordering for hiding latency

Expert Buffering: Reduce Static Memory Usage

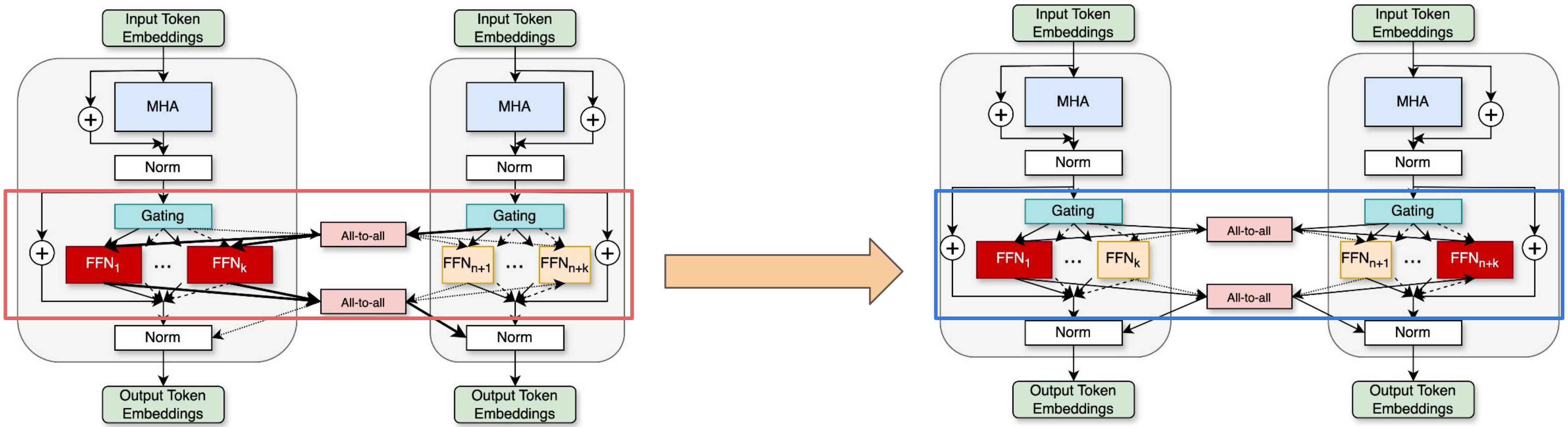
Only **a subset of experts** is activated in each batch



- Store bulk of the expert parameters in main memory
  → **Reduce Parameter Memory**
- Maintain *LIFO* cache on GPU that stores activated experts
  → **Mitigate GPU-CPU Latency**

Load Balancing: Improve Service Robustness

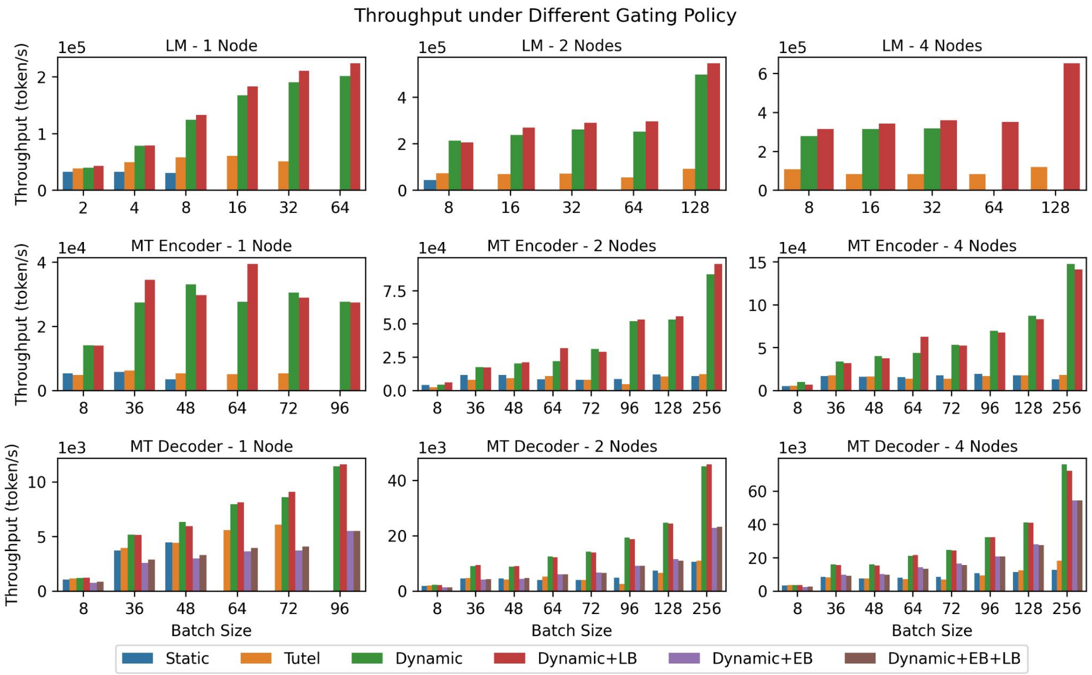
Imbalanced load on each GPU create **memory spikes** and **bottlenecks**



- Estimate expert load from historical activation
- Assign expert based on balancing load

Results

Improved **throughput** and maximum **batch size**



Check our paper and code for more information!

Paper

Paper (arXiv version)

Code