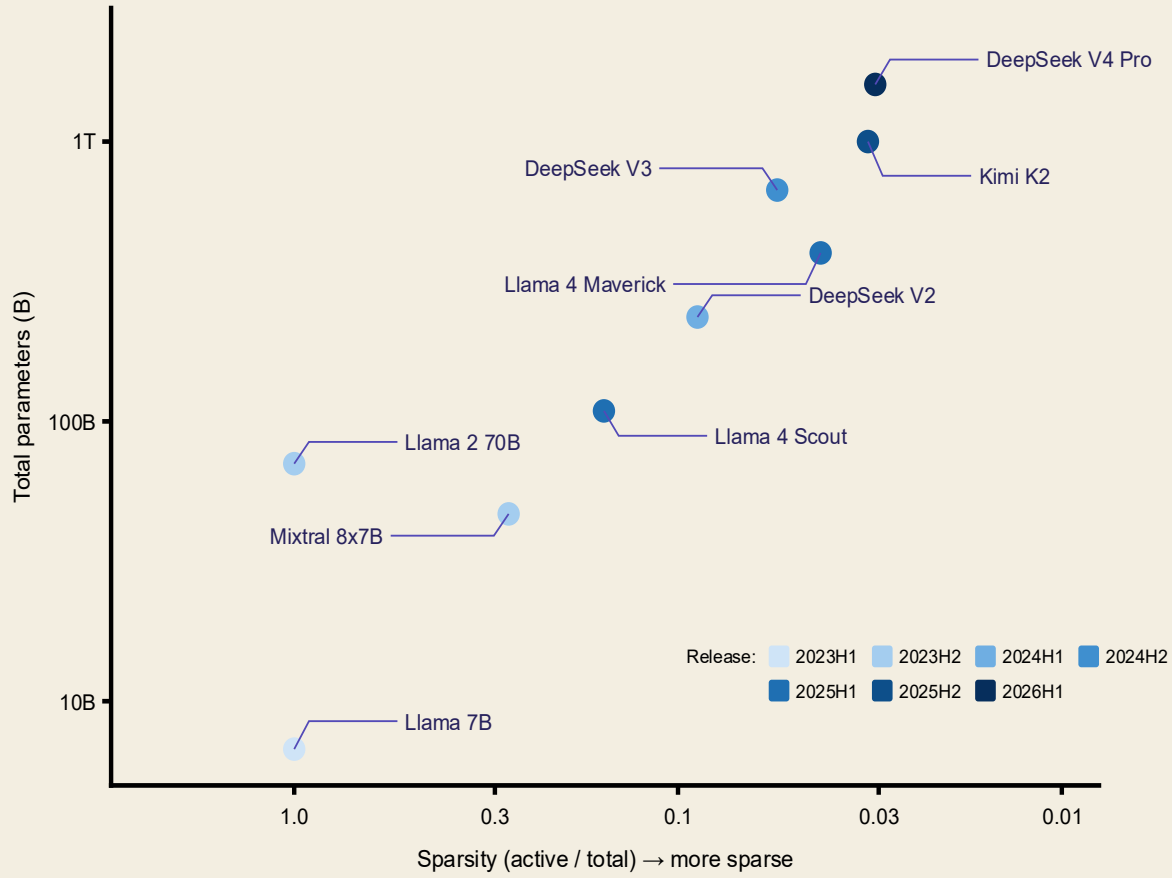
FarSkip-Collective: Unhobbling Blocking Communication in Mixture of Experts Models

Yonatan Dukler, Guihong Li, Deval Shah, Jiang Liu, Vikram Appia, Emad Barsoum

AMD
together we advance_

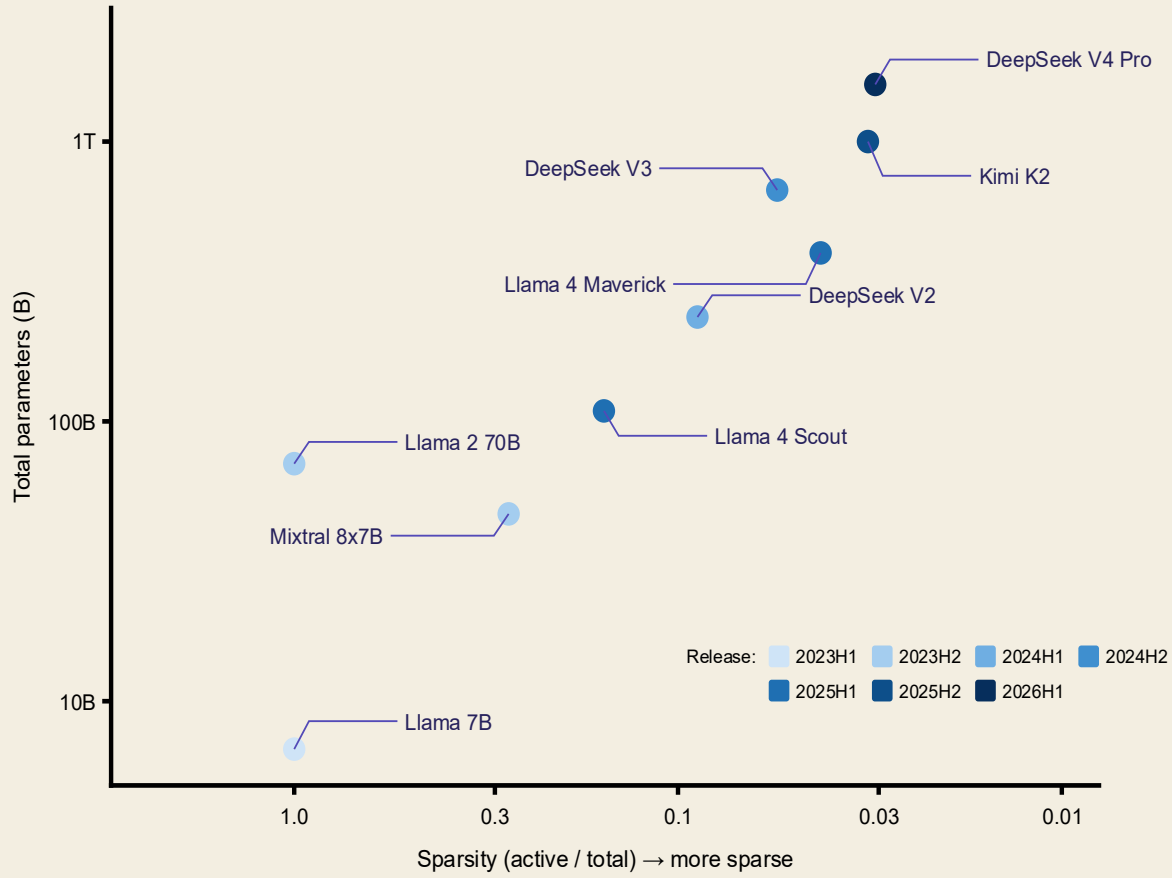
Scaling with MoEs

- MoEs use conditional computation
  - Reduce FLOPs per token



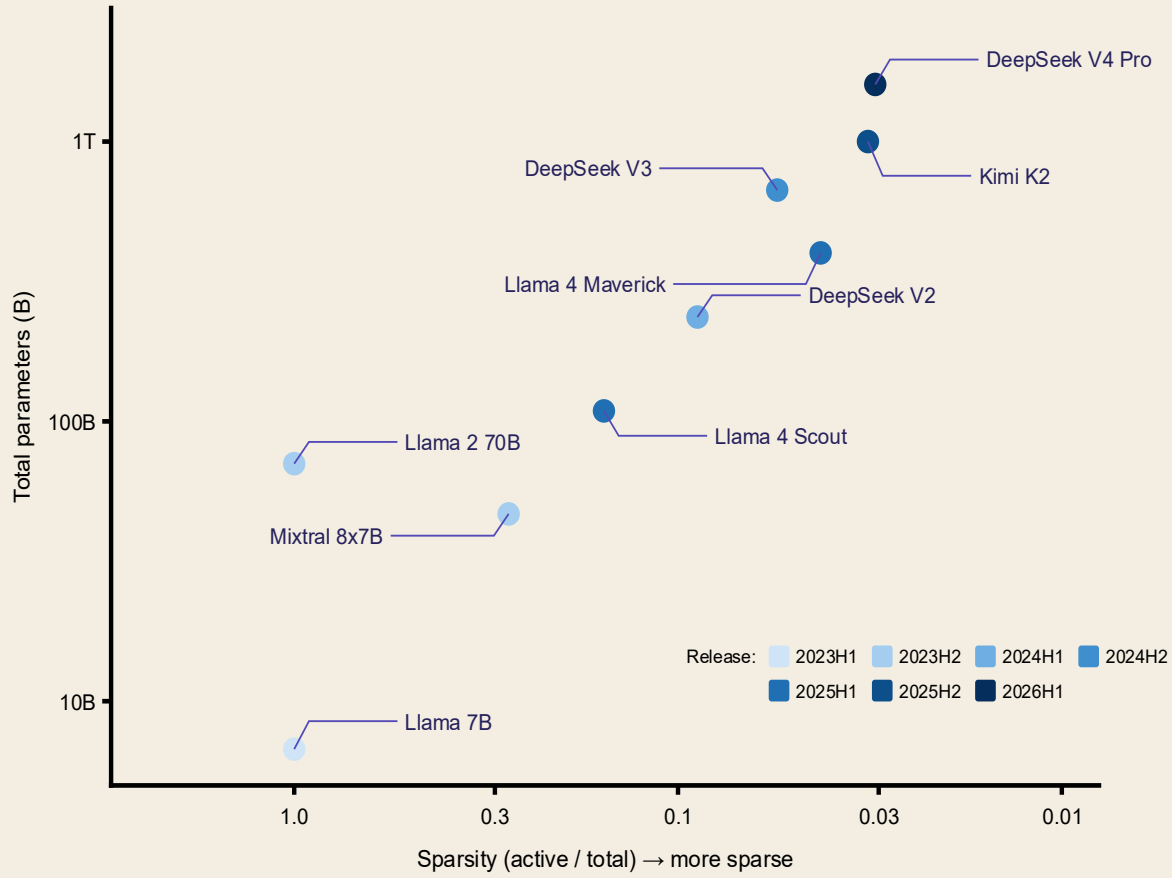
Scaling with MoEs

- MoEs use conditional computation
  - Reduce FLOPs per token
- Can be scaled further
  - Larger models and batch sizes



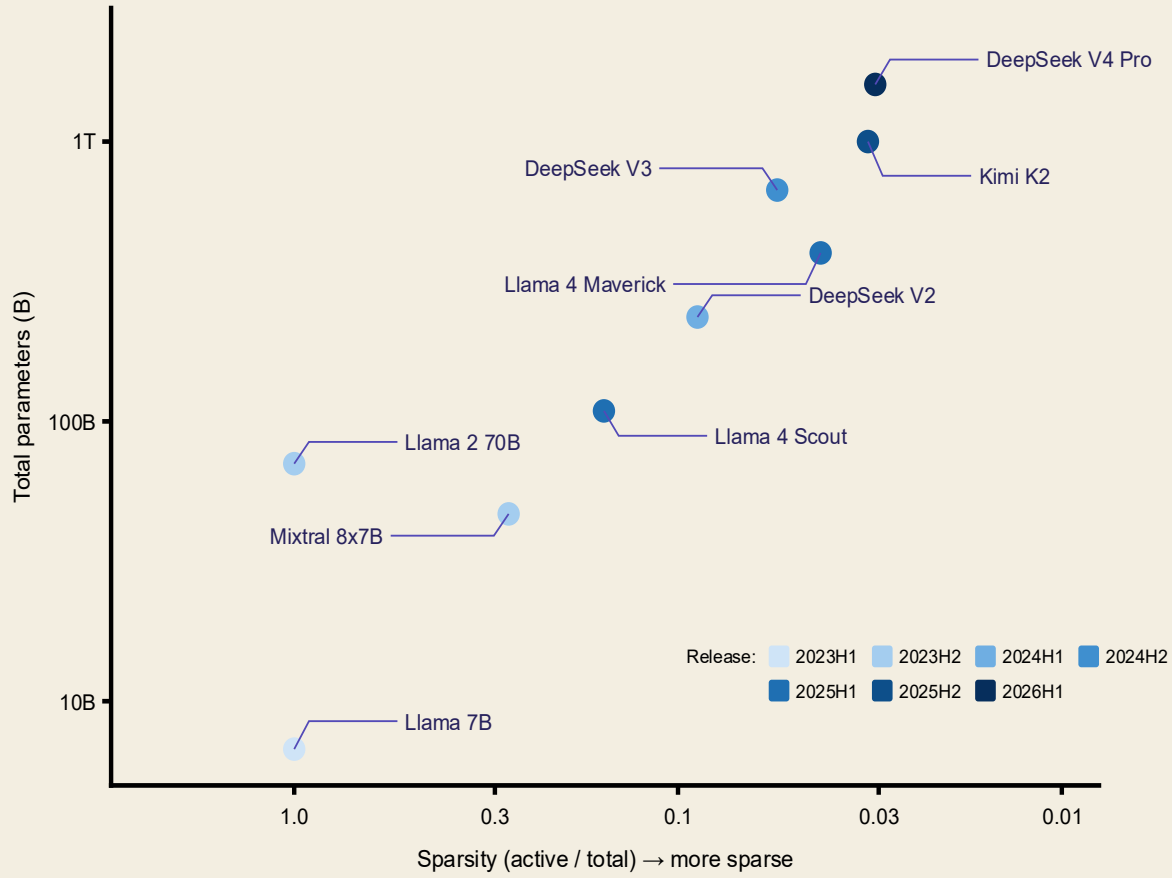
Scaling with MoEs

- MoEs use conditional computation
  - Reduce FLOPs per token
- Can be scaled further
  - Larger models and batch sizes
- More parallelism
  - Less compute more communication



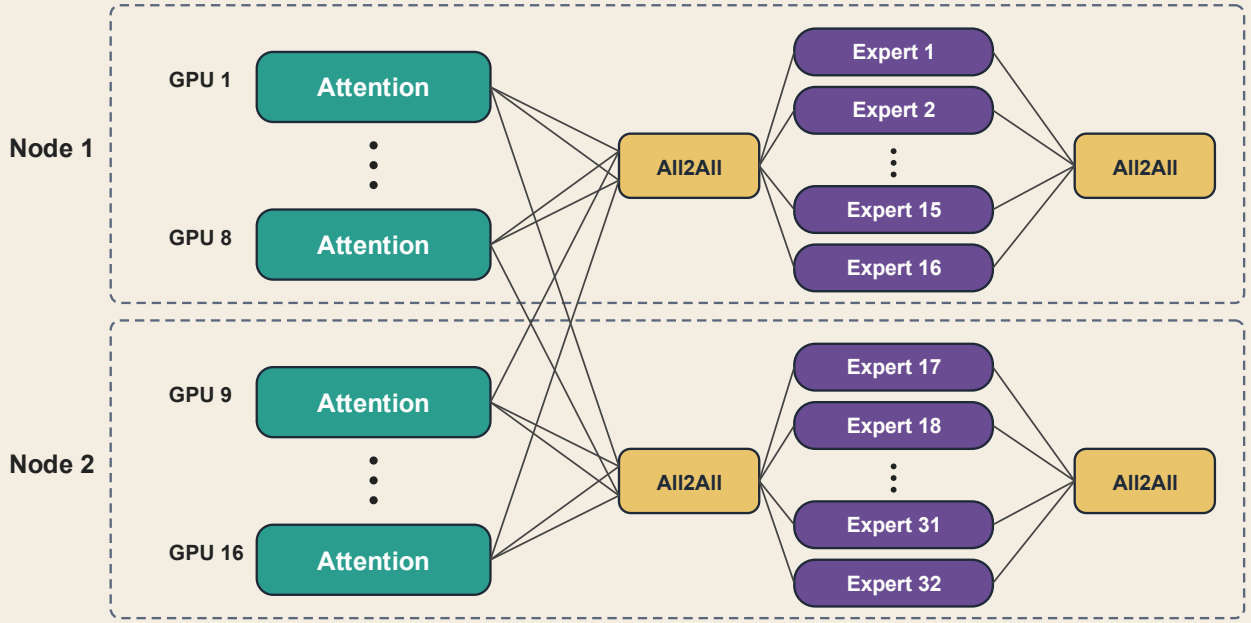
Scaling with MoEs

- MoEs use conditional computation
  - Reduce FLOPs per token
- Can be scaled further
  - Larger models and batch sizes
- More parallelism
  - Less compute more communication
- How do we keep scaling?
  - Go even larger, even sparser



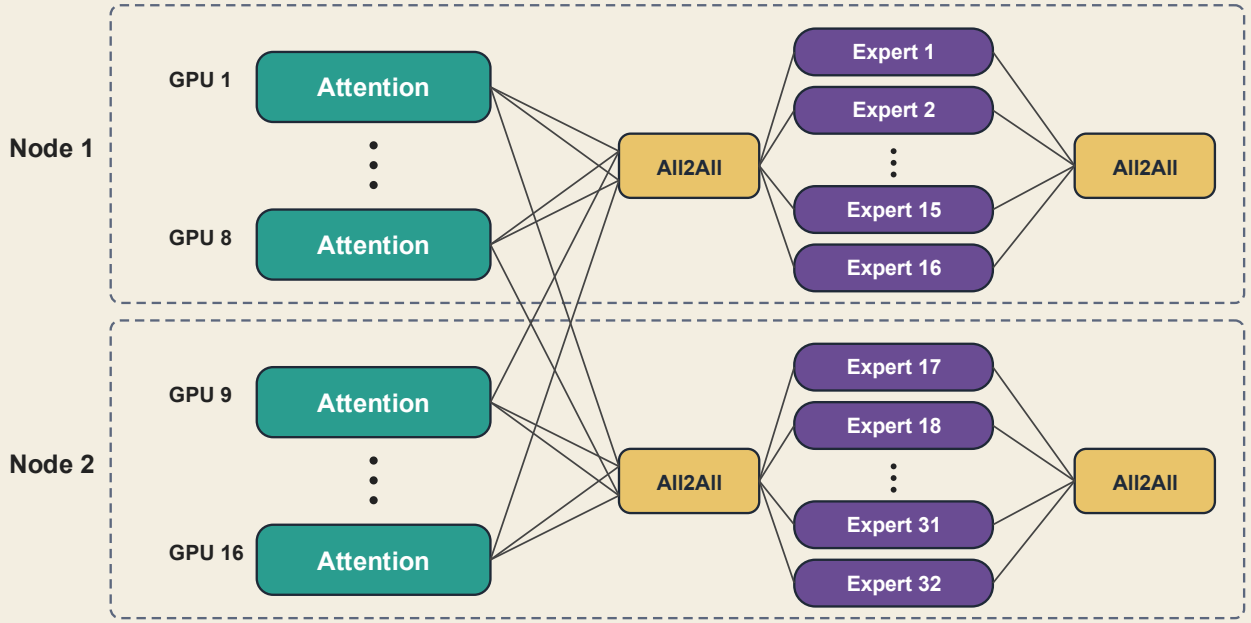
MoE Communication

- Key Communication: Expert Parallelism



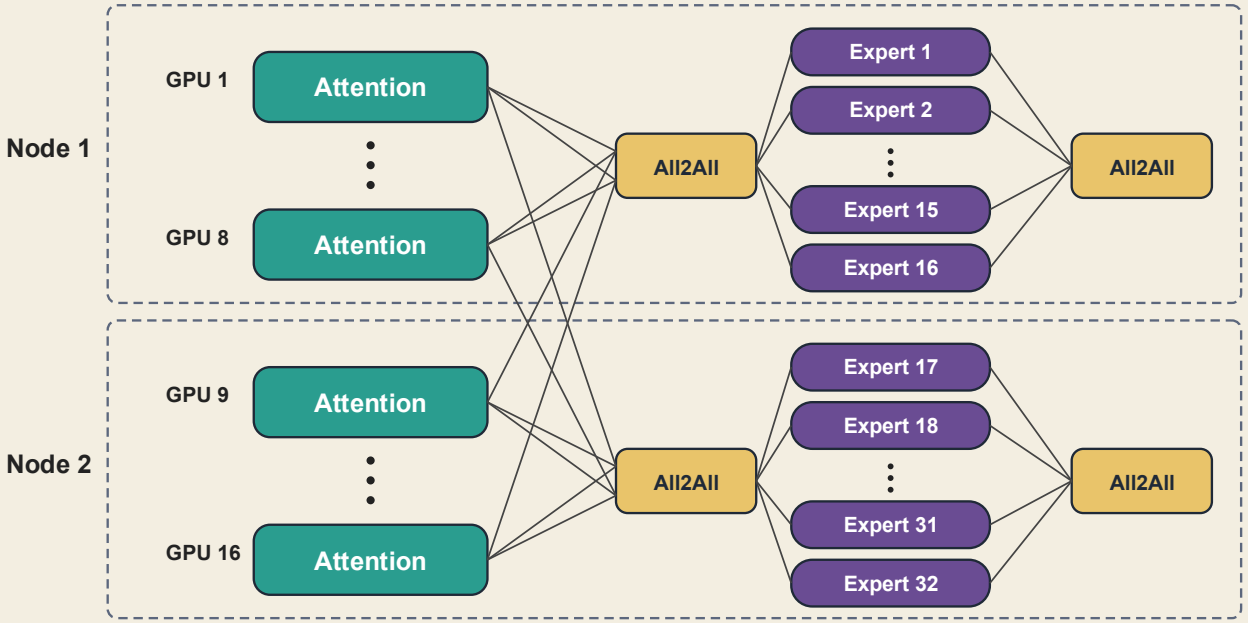
MoE Communication

- Key Communication: Expert Parallelism
  - **Dispatch** post-attention compute router scores and dispatch tokens from DP ranks to experts (All2All)



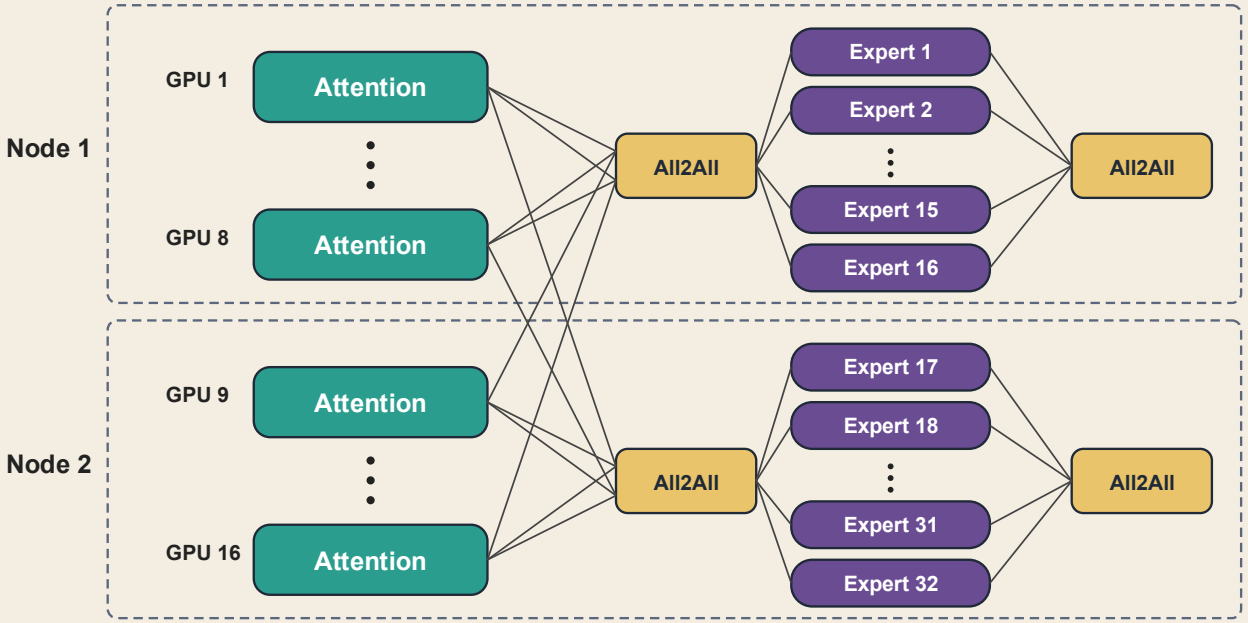
MoE Communication

- Key Communication: Expert Parallelism
  - **Dispatch** post-attention compute router scores and dispatch tokens from DP ranks to experts (All2All)
  - **MoE Computation** calculate activations of selected experts



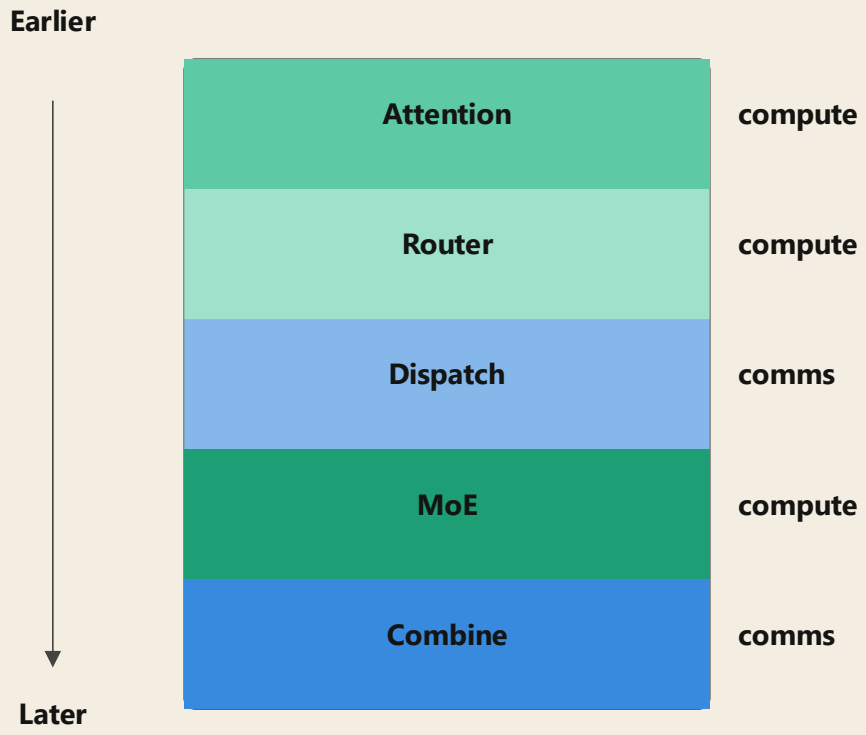
MoE Communication

- Key Communication: Expert Parallelism
  - **Dispatch** post-attention compute router scores and dispatch tokens from DP ranks to experts (All2All)
  - **MoE Computation** calculate activations of selected experts
  - **Combine** from experts back to DP ranks (All2All)



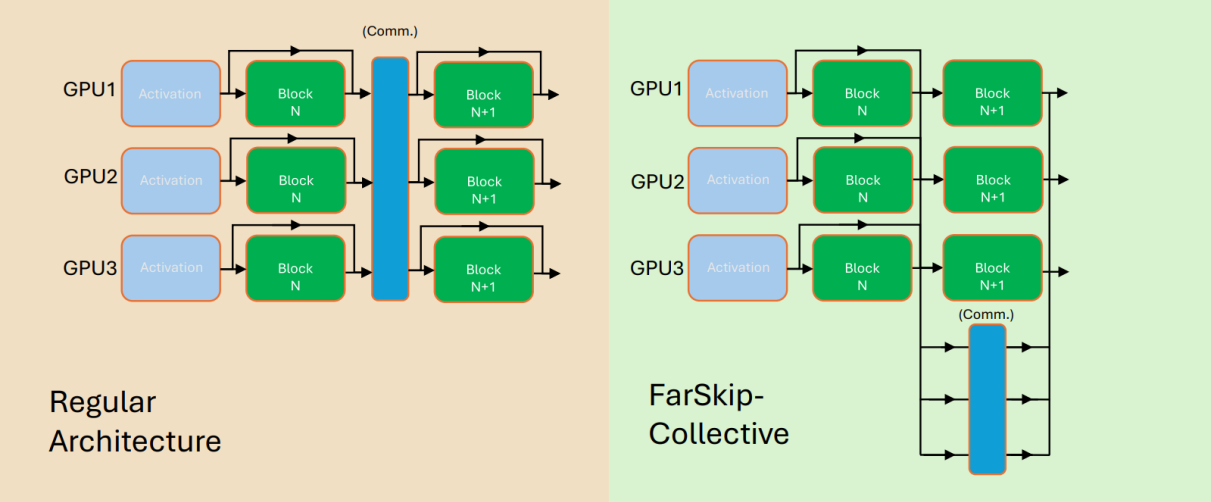
MoE blocking communication

- Attention output
- Router output
- Dispatch (communication)
- MoE output
- Combine (communication)



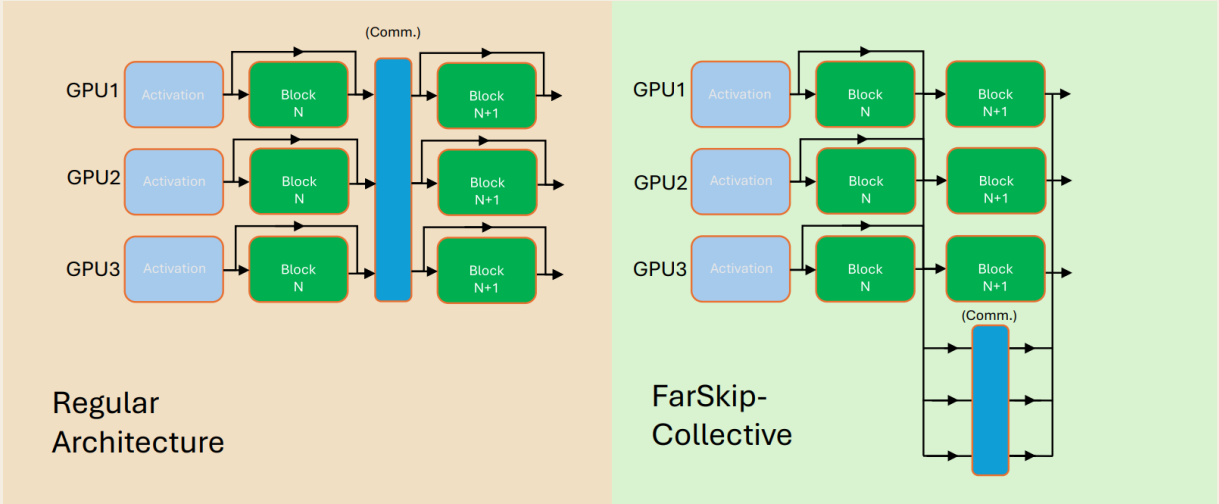
Unhobbling blocking communication

- Modify the computation graph to allow computation to run in parallel with communication



Unhobbling blocking communication

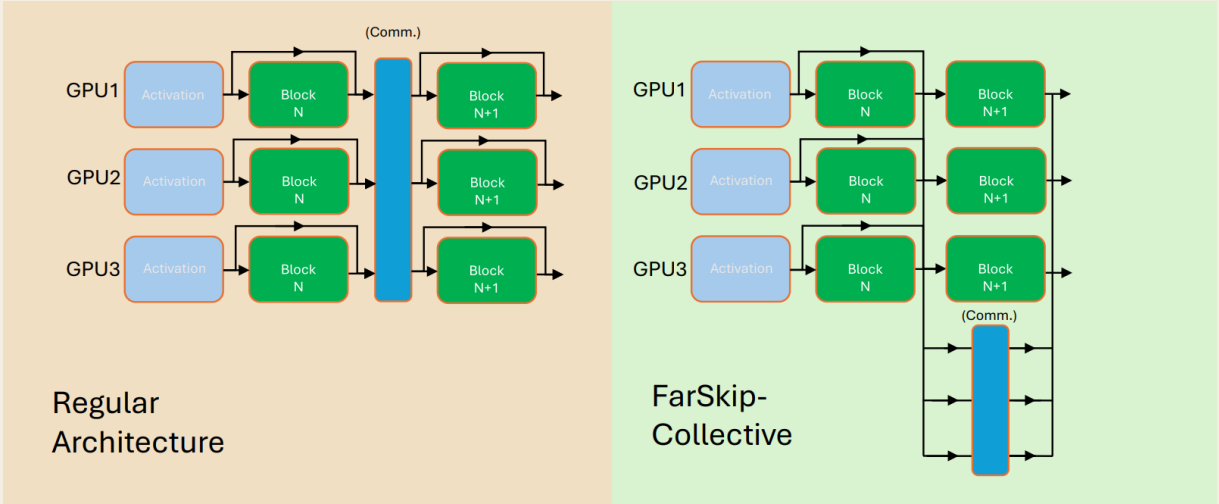
- Modify the computation graph to allow computation to run in parallel with communication
- Input to block is partial or outdated



Unhobbling blocking communication

- Modify the computation graph to allow computation to run in parallel with communication
- Input to block is partial or outdated

- $R_{n+1} = R_n + F(R_n)$ (standard residual)



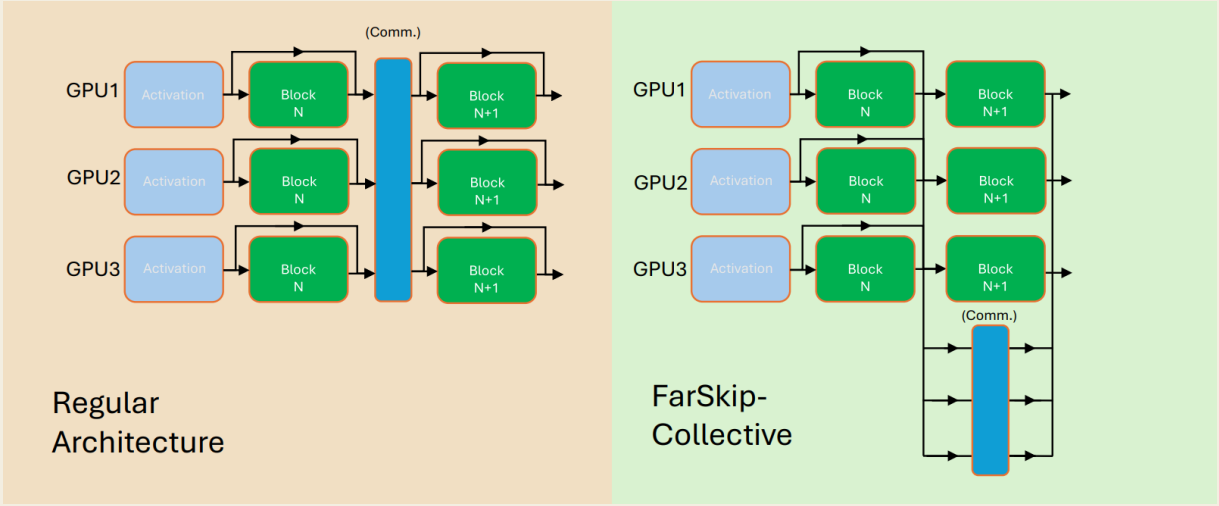
Unhobbling blocking communication

- Modify the computation graph to allow computation to run in parallel with communication
- Input to block is partial or outdated

- $R_{n+1} = R_n + F(R_n)$ (standard residual)

- FarSkip-Collective disentangles

$R_{n+1} = R_n + F(R_n^*)$ (disentangled; $R_n \neq R_n^*$)



Unhobbling blocking communication

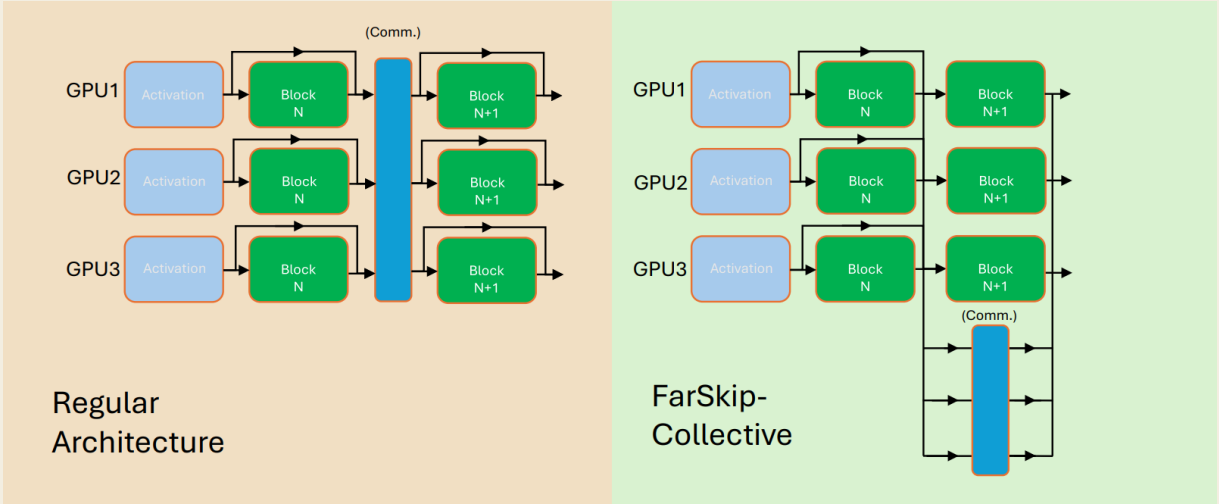
- Modify the computation graph to allow computation to run in parallel with communication
- Input to block is partial or outdated

- $R_{n+1} = R_n + F(R_n)$ (standard residual)

- FarSkip-Collective disentangles

$R_{n+1} = R_n + F(R_n^*)$ (disentangled; $R_n \neq R_n^*$)

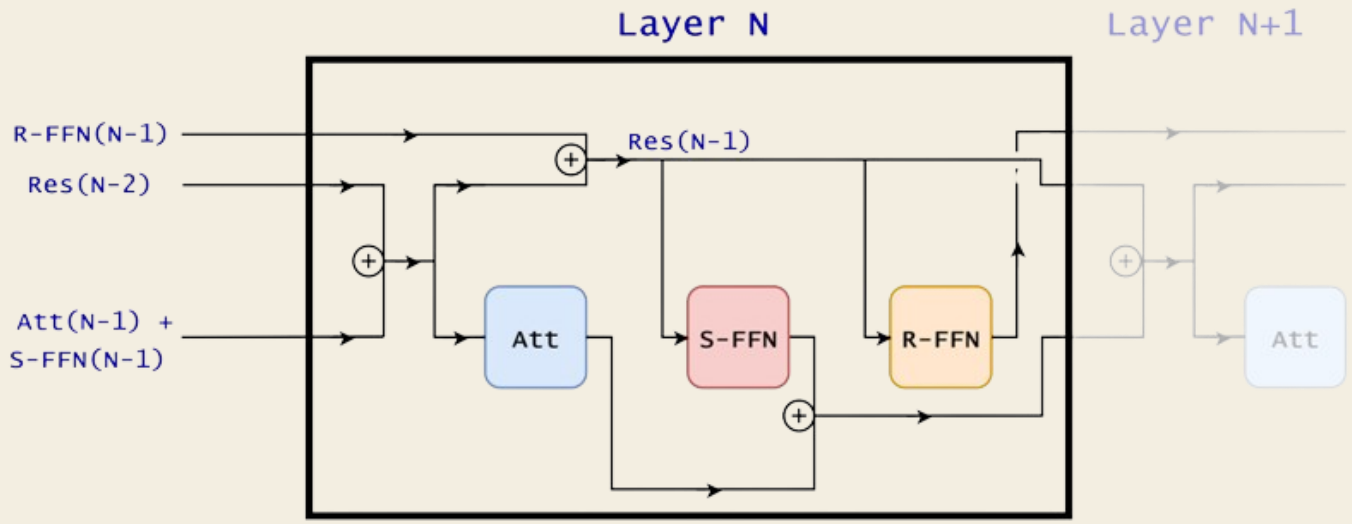
- R_n^* set as outdated and partial activations to remove blocking of F



FarSkip-Collective Architecture

Attention input: partial activation (attention & shared experts)

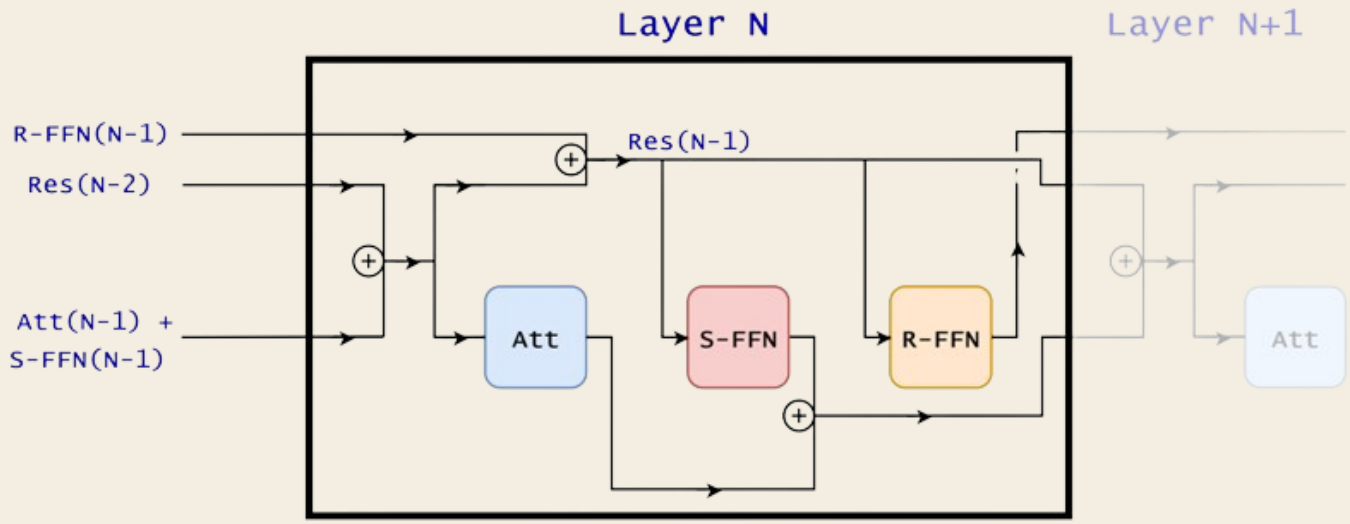
MoE input: outdated activation



FarSkip-Collective Architecture

Attention input: partial activation (attention & shared experts)

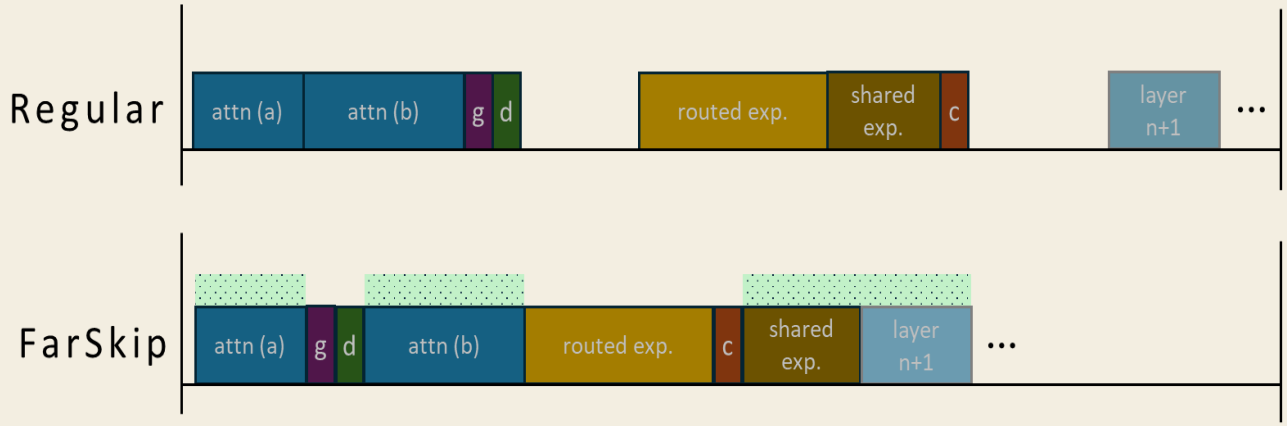
MoE input: outdated activation



Most minimal intervention to allow for overlapping (A) *Dispatch* (B) *Combine* and (C) *Attention comm*.

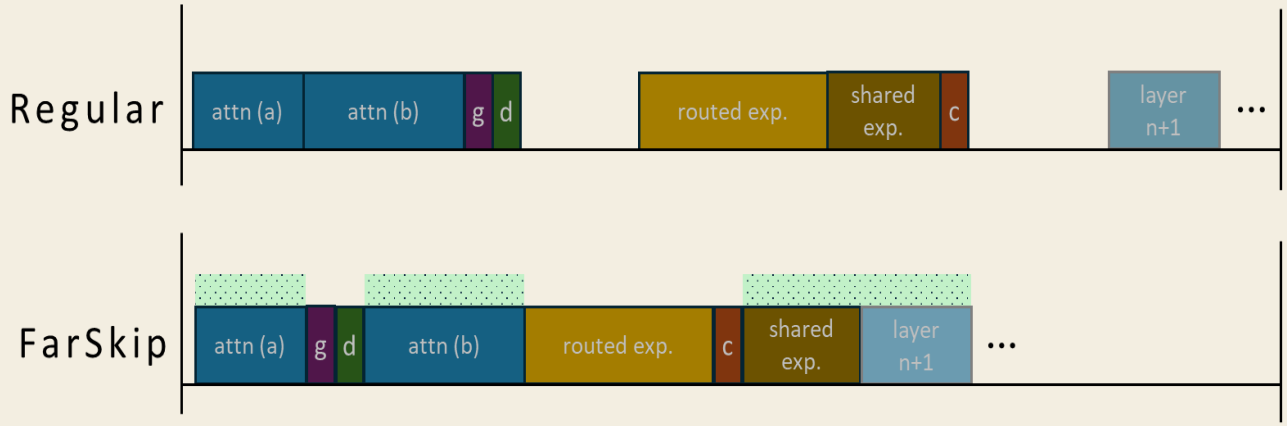
FarSkip-Collective MoE: Cost vs. Benefit

Benefit: Communication overlap speedup



FarSkip-Collective MoE: Cost vs. Benefit

Benefit: Communication overlap speedup



Cost: Potentially damaging for MoE expressivity

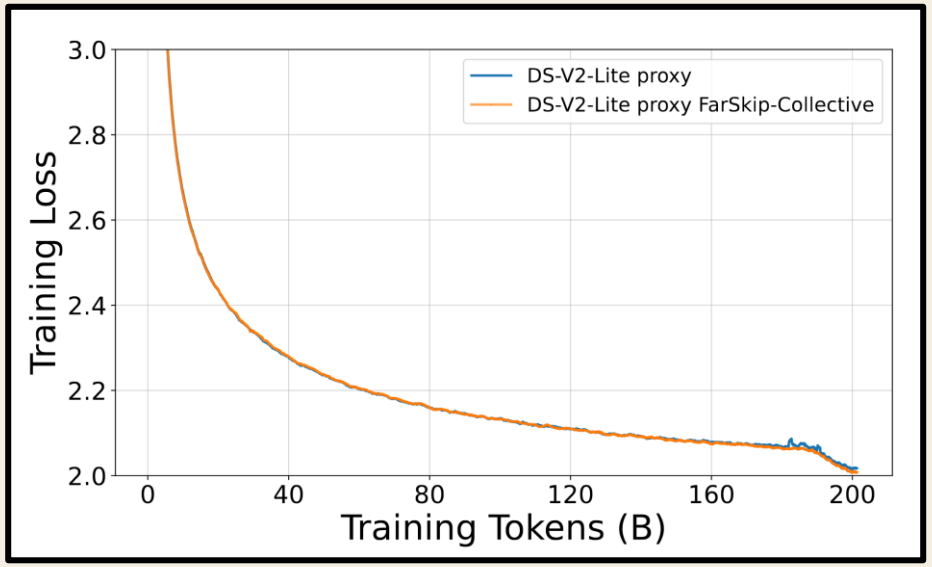
FarSkip-Collective models work at scale

FarSkip-Collective models work at scale

- Pretraining

FarSkip-Collective models work at scale

- Pretraining
- DeepSeek-V2 Lite (16B-64E); 200B training tokens



Benchmark	DS-V2-Lite-arch Reg.	DS-V2-Lite-arch Far.
PIQA	78.2	79.2
ARC-E	70.3	70.4
ARC-C	43.9	44.5
HS	69.2	69.3
WG	62.4	62.6
MMLU	43.3	41.7
OpenBook	41.0	40.0
GSM-8K	30.9	31.0
HEval+	26.8	23.8
MBPP+	48.7	49.2
Avg	51.5	51.2

FarSkip-Collective models work at scale

- Distillation

FarSkip-Collective models work at scale

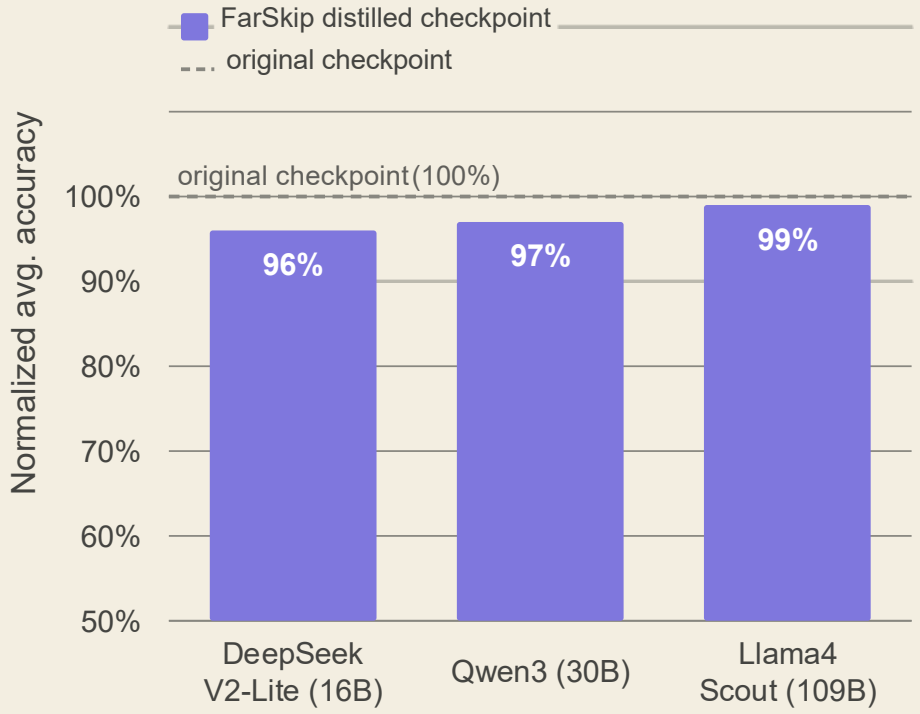
- Distillation
- Only modifies connectivity of the model

FarSkip-Collective models work at scale

- Distillation
- Only modifies connectivity of the model
- Directly distill MoE Checkpoints into FarSkip-Collective checkpoints

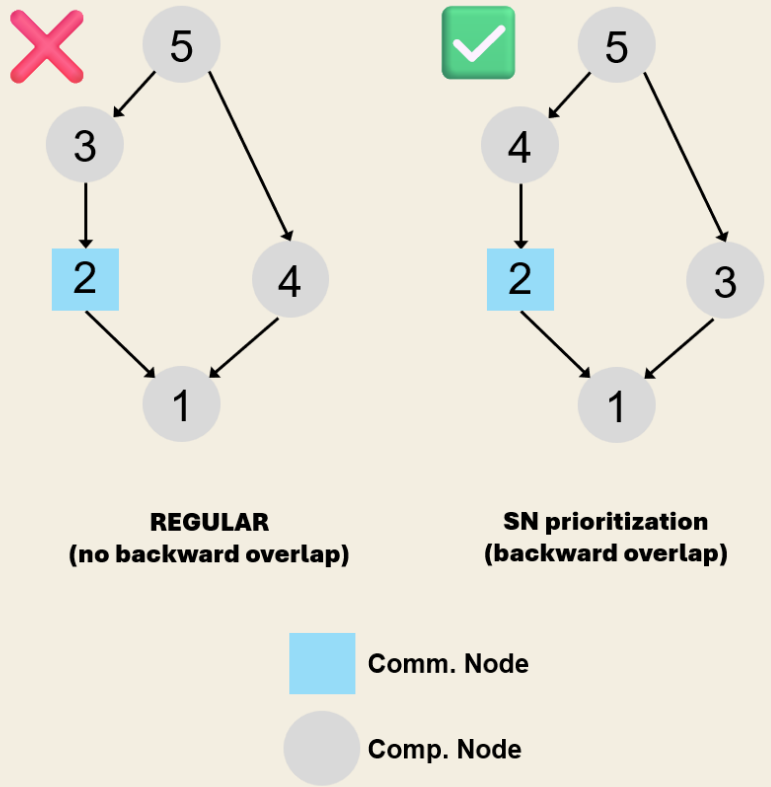
FarSkip-Collective models work at scale

- Distillation
- Only modifies connectivity of the model
- Directly distill MoE Checkpoints into FarSkip-Collective checkpoints
- FCSD: fully convert into FarSkip-Collective connectivity with self-distillation < 10B tokens



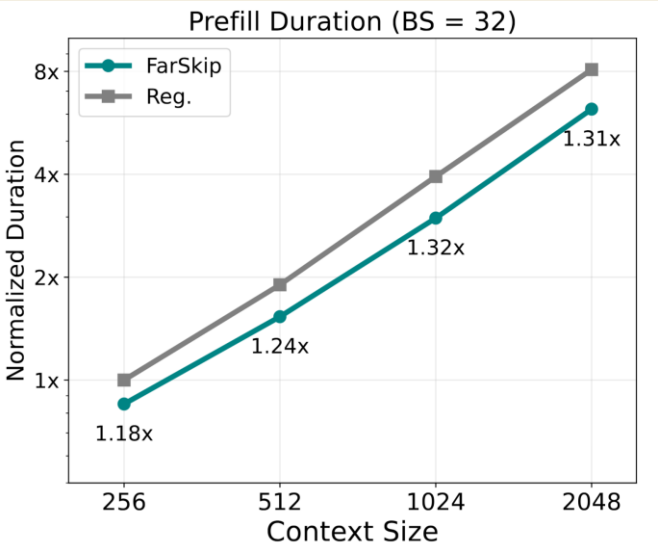
FarSkip-Collective Overlapped Implementation

- Implement at PyTorch layer for generality
- Training
  - Implement with Primus / MegatronLM
  - Communication-overlap in Forward and Backward
- Inference
  - Implement with vLLM and SGLang
  - Merged communication to reduce communication bandwidth

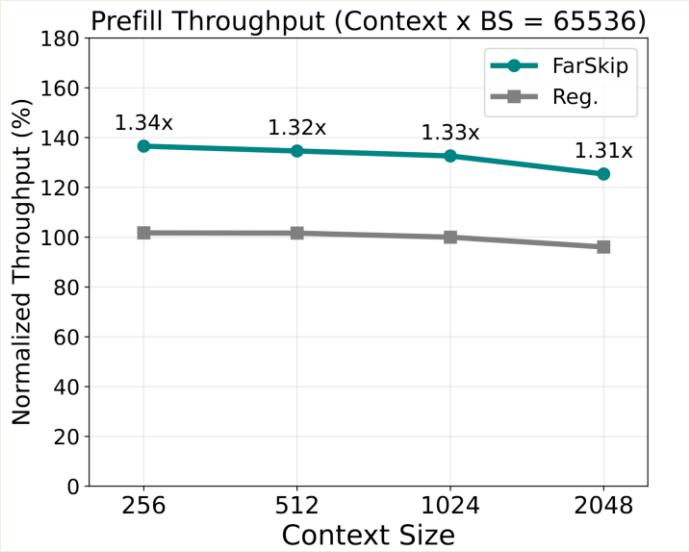


Benchmarking FarSkip-Collective

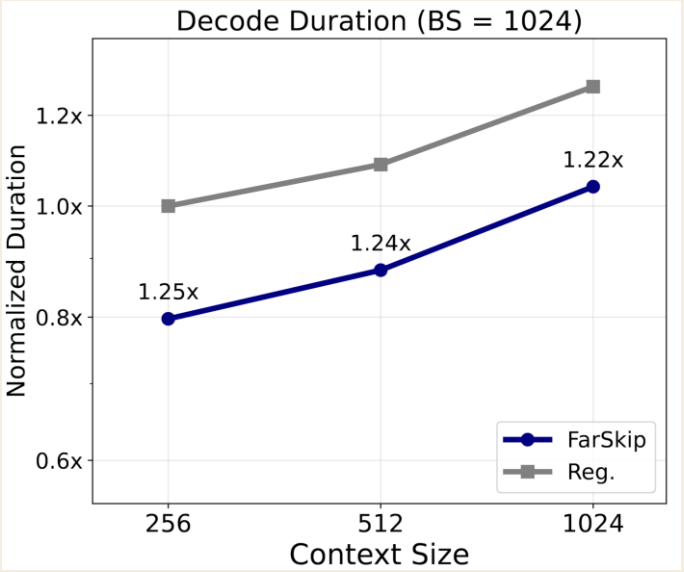
DeepSeek-V3 Inference



EP=8 / TP=8



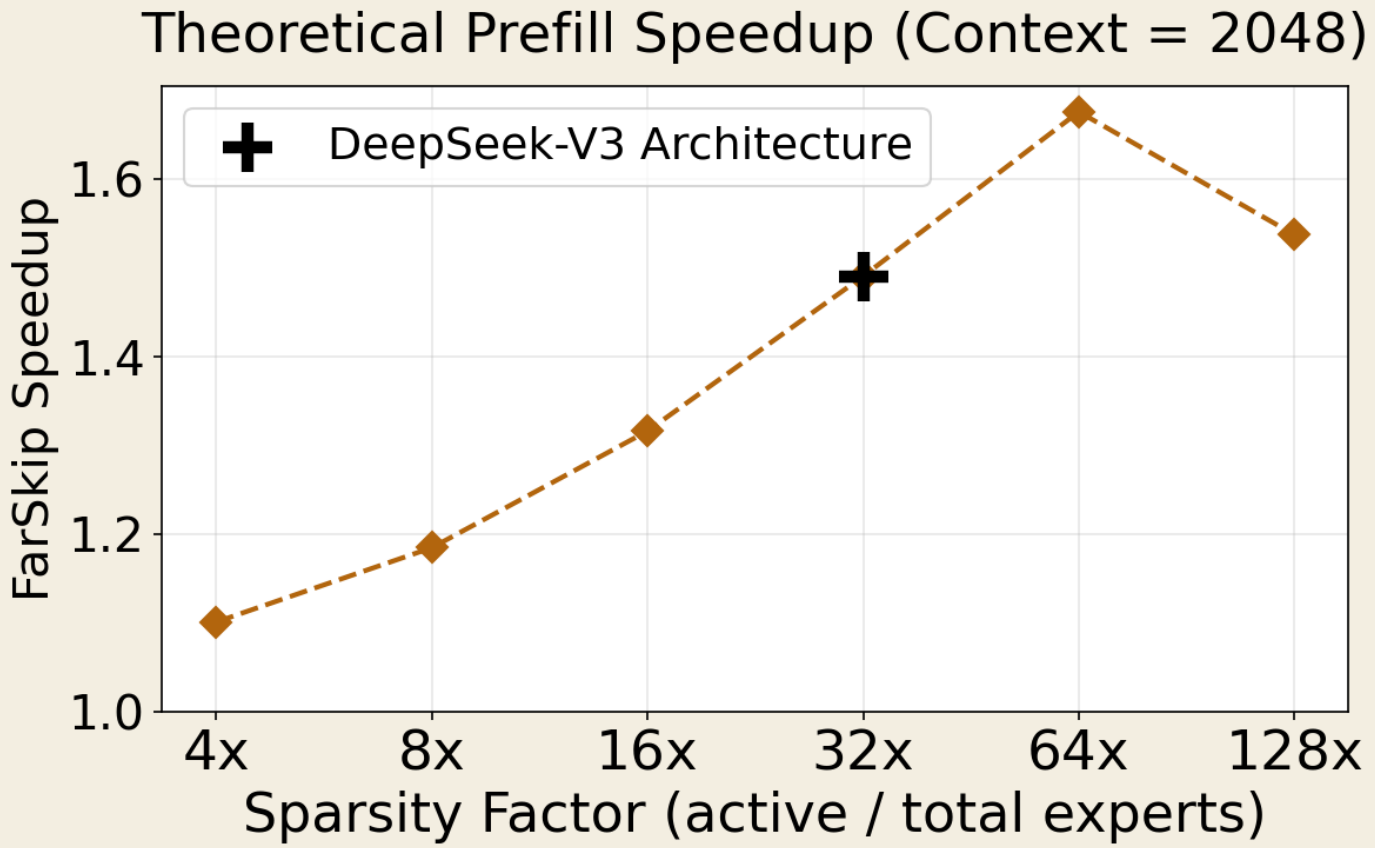
EP=8 / TP=8



WideEP (EP=16 / TP=16)

Towards Sparser MoEs

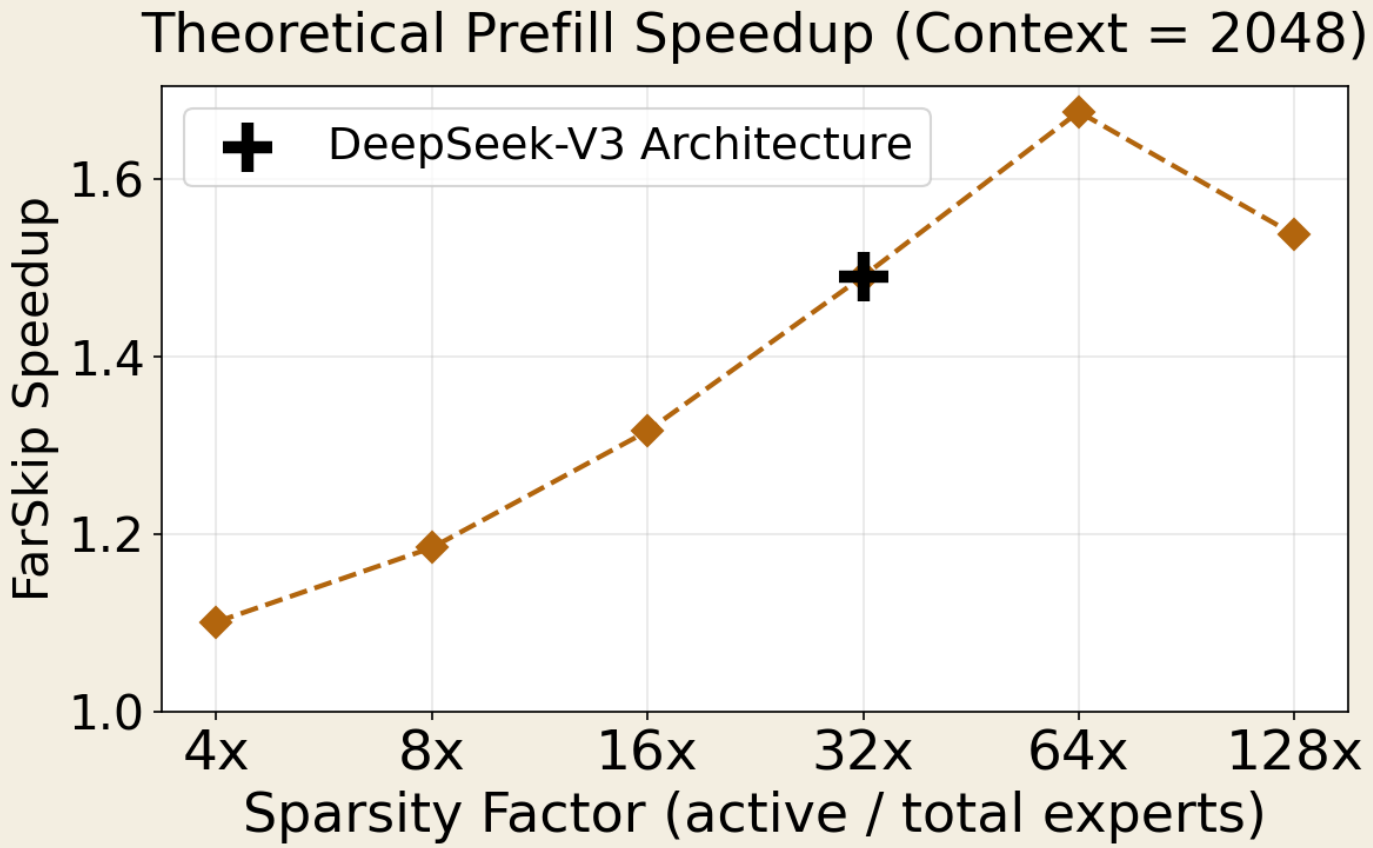
- Analytical Modeling of FarSkip-Collective speed-up



Towards Sparser MoEs

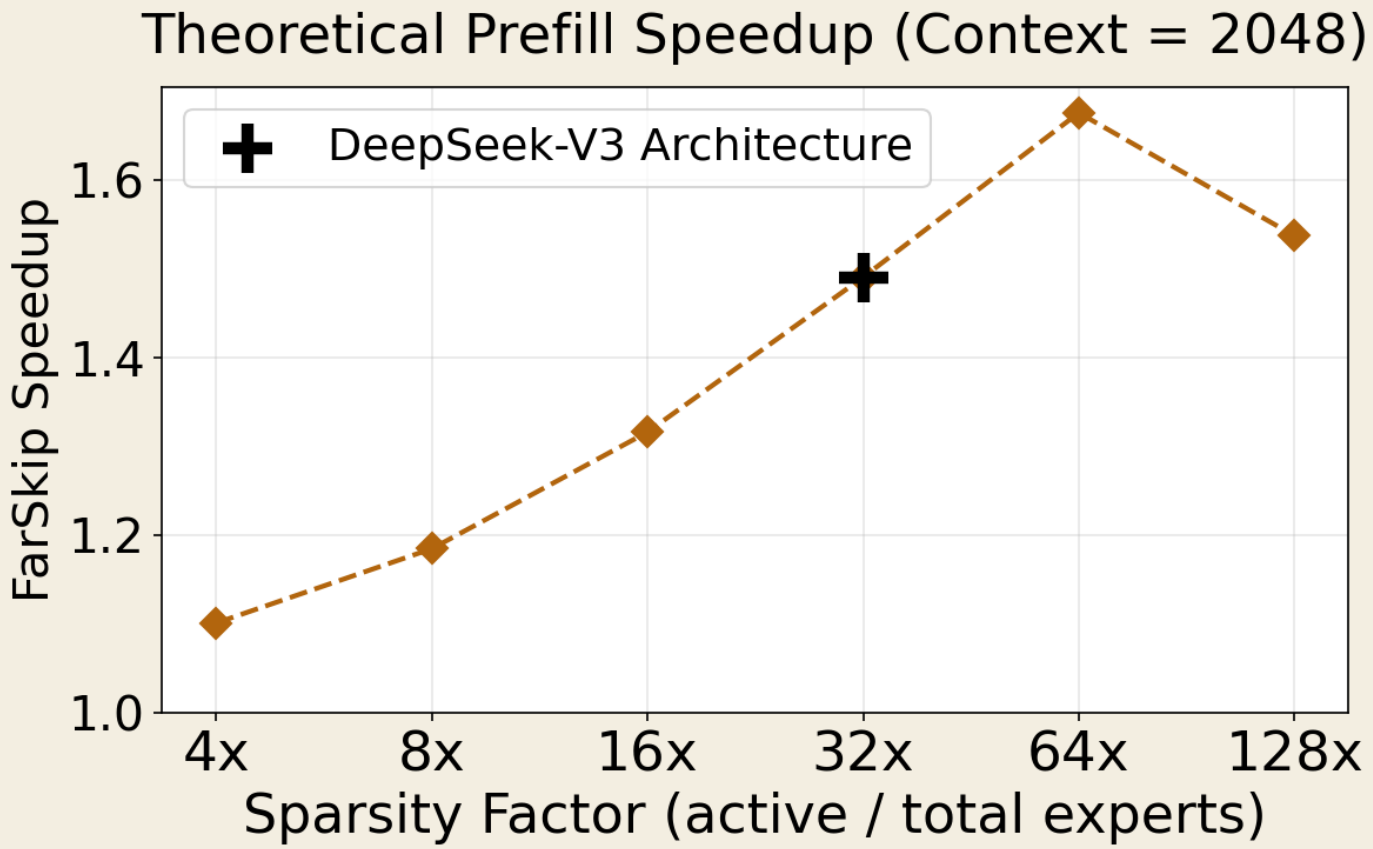
- Analytical Modeling of FarSkip-Collective speed-up

- Prefill assumptions
  - Compute-bound
  - Comm-bandwidth-bound



Towards Sparser MoEs

- Analytical Modeling of FarSkip-Collective speed-up
- Prefill assumptions
  - Compute-bound
  - Comm-bandwidth-bound
- Unlocking sparser models



Thank You