NUS National University of Singapore

xML Lab

NEURAL INFORMATION PROCESSING SYSTEMS

NeurIPS 2026

dMoE: dLLMs with Learnable Block Experts

Sicheng Feng, Zigeng Chen, Gongfan Fang, Xinyin Ma, Xinchao Wang*

National University of Singapore · fengsicheng@u.nus.edu, xinchao@nus.edu.sg · *Corresponding author

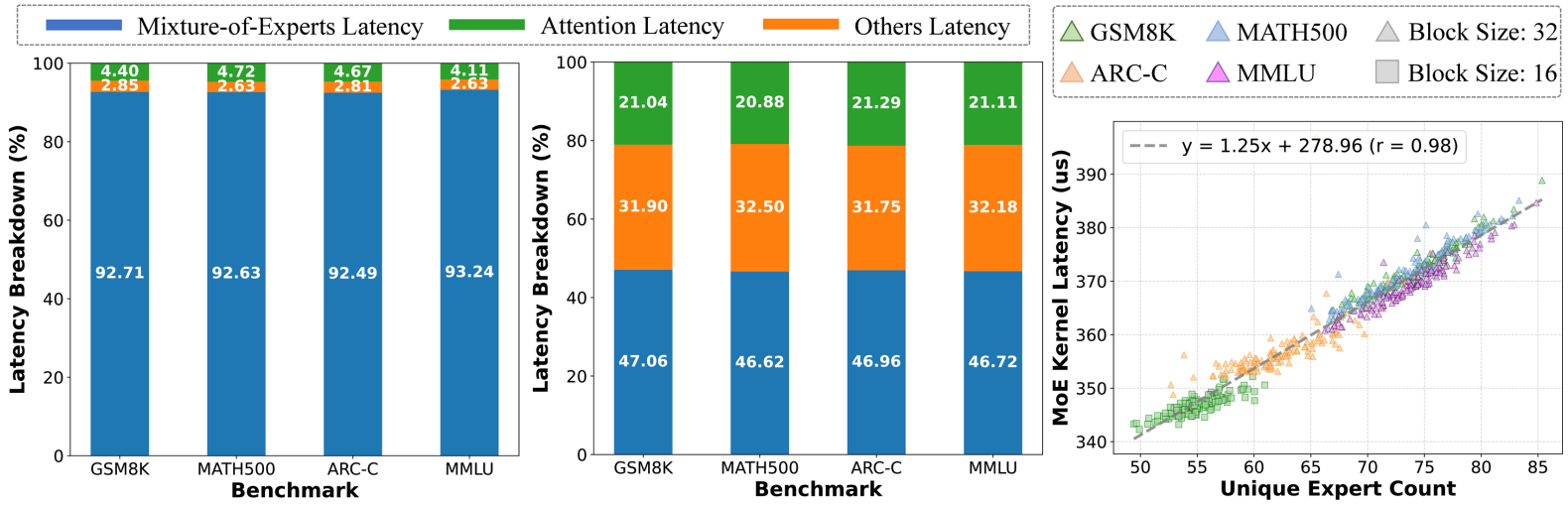
TL;DR **Route experts per block, not per token**

4.19×	99.55%	2.04×
fewer unique experts	of the original accuracy	peak end-to-end speedup

github.com/fscdc/dMoE

MoE dLLMs are memory-bound

dLLMs denoise a whole block of tokens in each forward pass, but every MoE layer still routes each token on its own, so the experts picked by all tokens in the block must be loaded from memory.



(a) Latency Breakdown (Transformers) (b) Latency Breakdown (SGLang) (c) MoE Kernel Latency (SGLang)

Empirical study on LLaDA2.0-mini with block diffusion decoding: (a, b) latency breakdown under Transformers and SGLang; (c) MoE kernel latency vs. unique expert count.

≈70

unique experts (of 256) per layer for one block

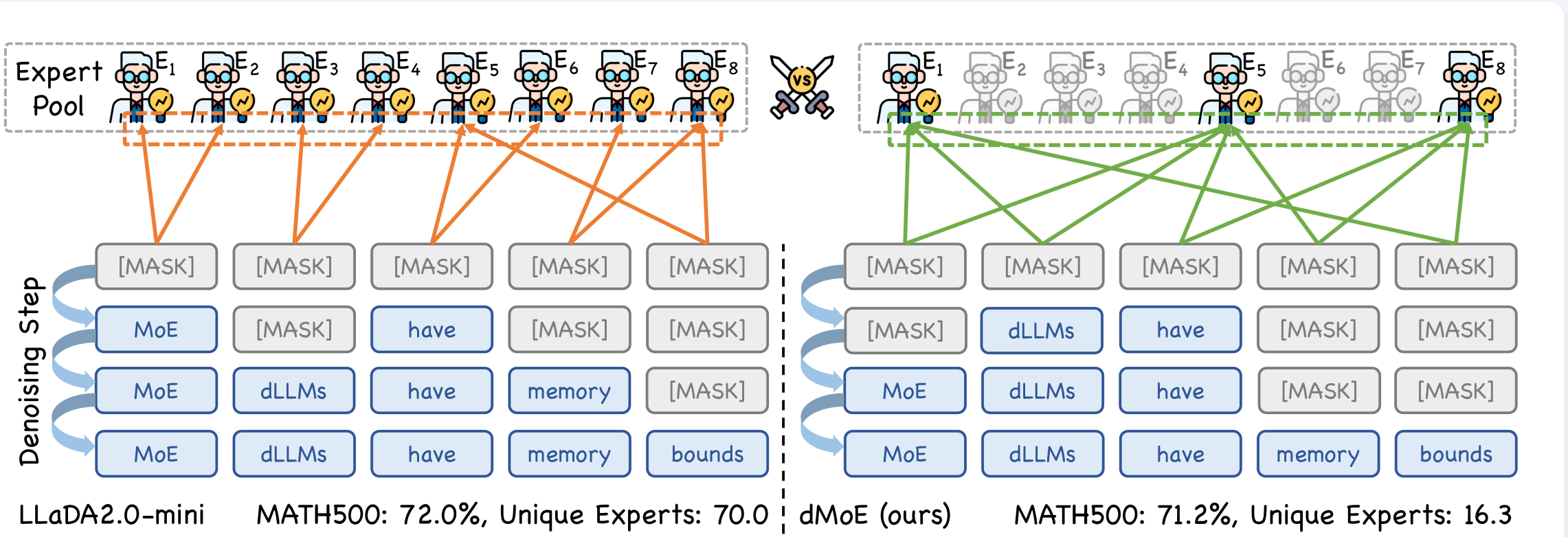
≈93%

of end-to-end latency is MoE (Transformers backend)

r = 0.98

MoE kernel latency vs. unique expert count (SGLang)

Route per block, not per token



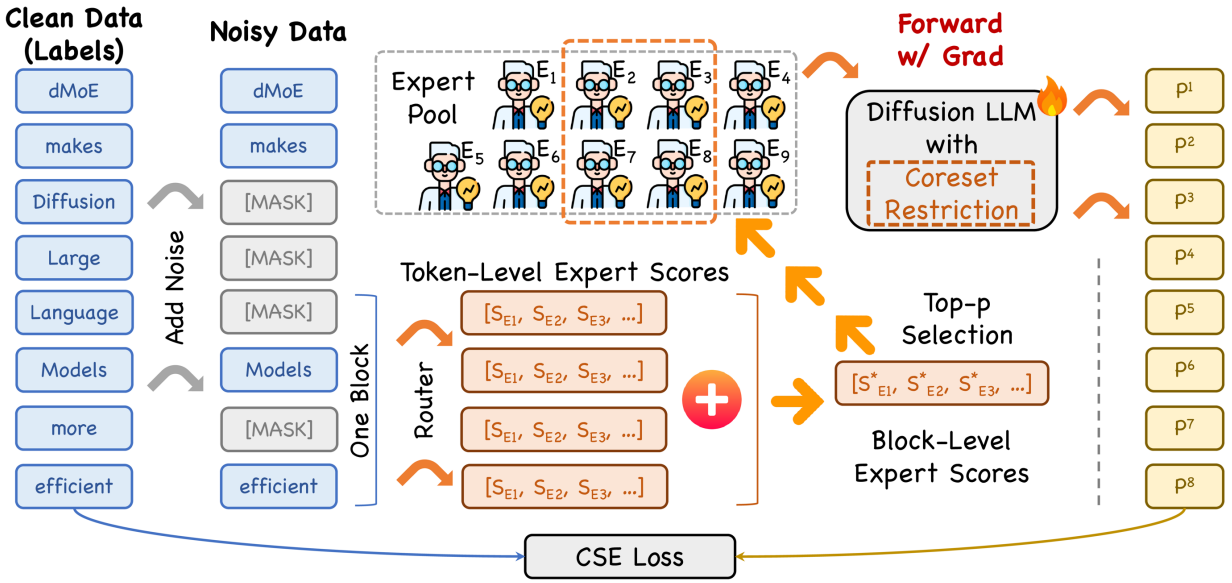
Per-token routing (LLaDA2.0-mini)

Each token picks its own top-k experts, so one block touches ≈70 experts per layer.

Per-block routing (dMoE)

All tokens in the block route within one shared coreset (≈16 experts) built from their router scores.

dMoE: learnable block-level routing



Learnable via self-distillation

- Labels: ≈700K responses generated by LLaDA2.0-mini itself.
- Training and inference share the same dMoE forward pass.

1 Token-level scores

The router scores every token as usual.

$$\mathbf{s}_i = \mathrm{Router}(t_i)$$

2 Block-level aggregation

Sum the tokens' top-k scores per expert.

$$S_{block} = \sum_{i \in \mathcal{B}} \mathrm{TopKMask}(\mathbf{s}_i,\, k)$$

3 Adaptive coreset (top-p)

Keep the fewest experts covering mass p.

$$\mathcal{C} = \text{top-p}\left(\tilde{S}_{block},\, p\right)$$

4 Restricted token routing

Each token picks k experts from the coreset.

$$R_i = \mathrm{TopK}\left(\mathbf{s}_i^{\mathcal{C}},\, k\right)$$

Fewer experts, accuracy preserved

4.19×

fewer unique experts per layer
(69.5 → 16.6)

99.55%

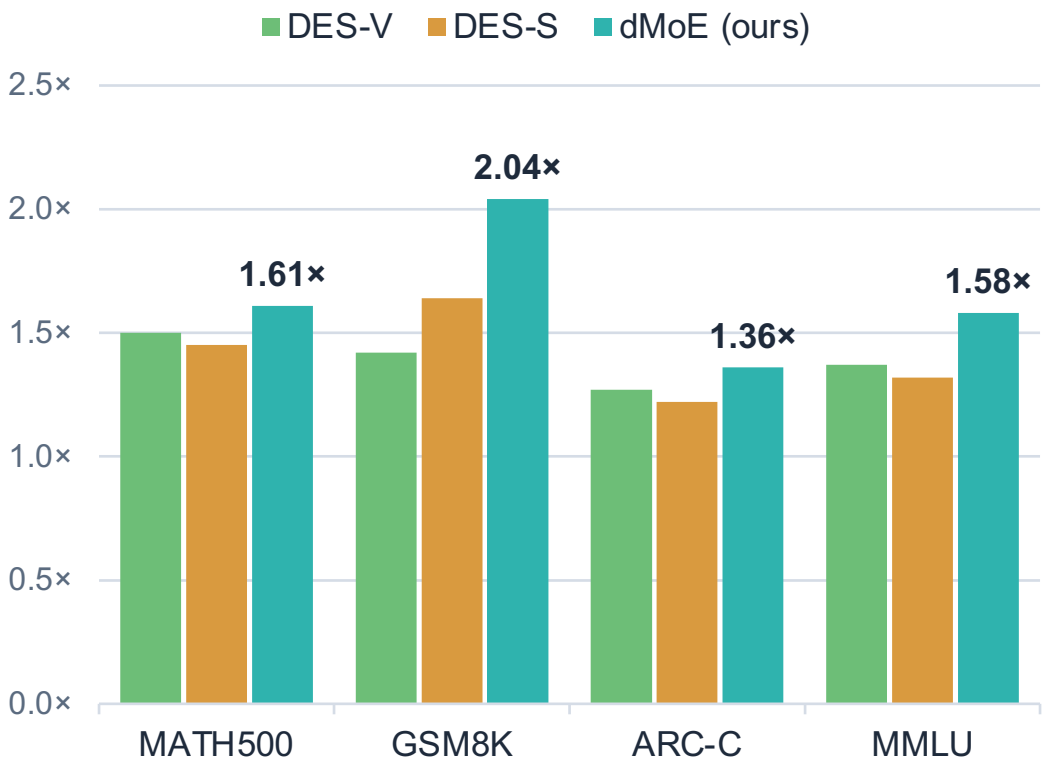
of the original accuracy
(average over four benchmarks)

Method	MATH500 Acc. / EC	GSM8K Acc. / EC	ARC-C Acc. / EC	MMLU Acc. / EC
LLaDA2.0-mini	72.0 / 70.0	90.8 / 71.5	84.4 / 65.9	88.6 / 70.6
+ Fine-Tuning	72.2 / 79.1	91.0 / 79.7	87.0 / 70.7	89.2 / 78.9
Top-4	52.6 / 43.5	72.4 / 44.3	74.0 / 42.2	61.5 / 43.8
DES-S	70.8 / 41.0	90.7 / 41.9	83.3 / 39.9	87.1 / 41.4
DES-V	71.8 / 36.9	90.5 / 37.0	82.3 / 36.8	85.6 / 37.0
DES-S*	53.8 / 12.9	74.0 / 13.2	62.9 / 13.0	70.7 / 13.0
DES-V*	53.0 / 13.0	73.5 / 12.9	49.1 / 13.0	72.6 / 13.0
TEAM	71.2 / 43.6	90.1 / 44.8	85.0 / 38.4	87.4 / 43.8
dMoE (p = 0.6)	**71.2 / 16.3**	**88.6 / 16.6**	**87.1 / 16.9**	**87.4 / 16.5**
dMoE (p = 0.5)	**69.8 / 13.7**	**87.1 / 13.8**	**84.7 / 13.9**	**86.7 / 13.6**

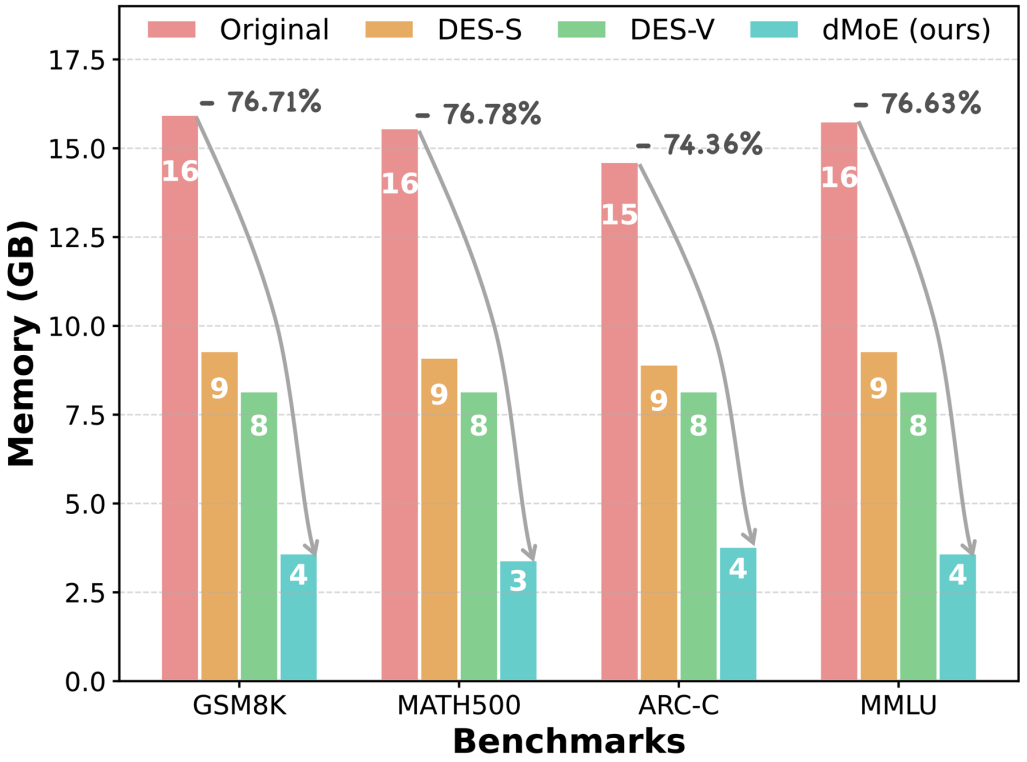
Acc.: accuracy (%). EC: unique activated experts per layer (out of 256).
**DES threshold tuned for higher compression. KV cache enabled for all methods.*

Real speedup and memory savings

2.04× peak end-to-end speedup at matched accuracy (1.36–2.04×)



74–77% less MoE memory than the original model



Takeaways

1. **Problem:** per-token routing makes MoE dLLMs memory-bound; one block activates ≈70 of 256 experts per layer.

2. **dMoE:** one adaptive top-p expert coreset per block and layer, learned by self-distillation; each token still uses k = 8 experts.

3. **Results:** 4.19× fewer unique experts with 99.55% of the original accuracy, up to 2.04× faster and 74–77% less MoE memory.

Code & contact

GitHub

WeChat

github.com/fscdc/dMoE
fengsicheng@u.nus.edu
xinchao@nus.edu.sg

Thank you!