Mixture-of-Experts Can Surpass Dense LLMs Under Strictly Equal Resource

Houyi Li, Ka Man Lo, Shijie Xuyang, Ziqi Wang, Wenzhen Zheng, Haocheng Zhang, Zhao Li, Shuigeng Zhou, Xiangyu Zhang, Daxin Jiang

ICLR 2026 · Oral

huggingface.co/kamanphoebe/moe_surpass_dense

1. Motivation

- Prior MoE↔Dense comparisons leave N (total parameters) free:
- **Perspective A** (fix D): MoE matches Dense but with $m\times$ larger N
  e.g. DeepSeekMoE 16B [1] ↔ 7B Dense; Mixtral 8×7B [2] ↔ LLaMA-2-70B
- **Perspective B** (fix C and N_a): sparser MoE requires much larger N (Kimi-2 Paper [3])

Shared blind spot — neither perspective controls N.
But N = capacity = size of one DP group = HBM footprint at inference.
When prior work says "MoE matches Dense", it really says: *"MoE matches Dense because MoE got a parameter subsidy."*

The Core Research Question

Can Mixture-of-Experts surpass dense LLMs under equal total parameter, compute, and data constraints?

2. The Equal Resource Framework

Per-Token Forward Computation (M):

$$M_{dense} \approx 2N \cdot \kappa_{Dense} \quad \text{vs} \quad M_{MoE} \approx 2r_a N \cdot \kappa_{MoE}$$

If $C_{dense} = C_{MoE}$ and $N_{dense} = N_{MoE}$, MoE's training token requirement D scales inversely with the activation rate r_a ($D_{MoE}/D_{dense} \propto 1/r_a$).

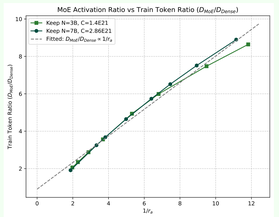


Figure 1: With Fixed N, C, r_a and D present a strict trade-off.

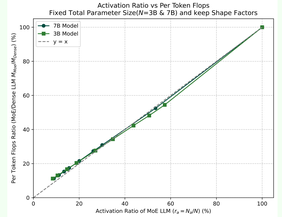


Figure 2: We empirically verified MoE per-token FLOPs correlate linearly with r_a.

3. Three-Step Methodology

- **Step 1 — Greedy Architecture Search.** Layer arrangement → Top-K allocation → shape (ζ, β); lock optimal backbone ⇒ fix κ_{MoE}.
- **Step 2 — Activation-rate analysis.** Sweep r_a at fixed N and C; find optimal $r_a^\star$.
- **Step 3 — Data-reuse strategy.** Equalize unique tokens with Dense; loose (2-ep) vs. strict (68B-cap) reuse.

Controlled-variable discipline. Dense baselines are *also* tuned (aspect ratio & FFN expansion from Li et al. 2025, Fig. 9). Hyperparameters via **StepLaw**, applied **uniformly** to both Dense and MoE — no "heavily-tuned MoE vs. neglected Dense."

4. Step1: Architecture Search (P1: Conclusions)

Component	Optimal Setting
Layer Arrangement	1dense+SE (One initial dense layer + MoE with Shared Expert)
Gate Normalization	Normalized Top-K gating works for $K > 1$
Top-K Routing	Intermediate K (avoid $K = 1$ or overly large K)
Shape Ratios	$\zeta \approx 88$ (Hidden/Layers) and $\mu \approx 22$ (MoE FFN ratio)

[1] Dai, D. et al. (2024). DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models. *arXiv preprint arXiv:2401.06066*.
[2] Jiang, A. Q. et al. (2024). Mixtral of Experts. *arXiv preprint arXiv:2401.04088*.
[3] Team Kimi (2025). Kimi K2 Technical Report. *arXiv preprint arXiv:2507.20534*.

5. Step2: Scaling Analysis at 2B

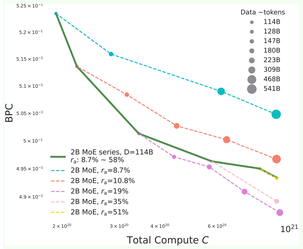


Figure 3: (a) Fixed r_a, add $D \to$ log-log linear. Fixed D, raise $r_a \to$ NOT log-log linear. **First finding — loss is not log-log-linear in compute in general.**

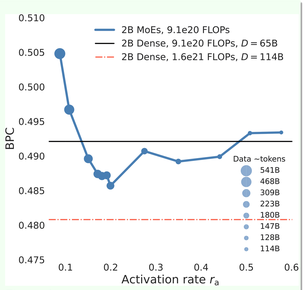


Figure 4: (b) Fixed N, C, sweep $r_a \Rightarrow$ optimal region around 20%. **Second finding — is not "More sparse is better".**

6. Step3A: AR Analysis and Data-Reuse at 3B

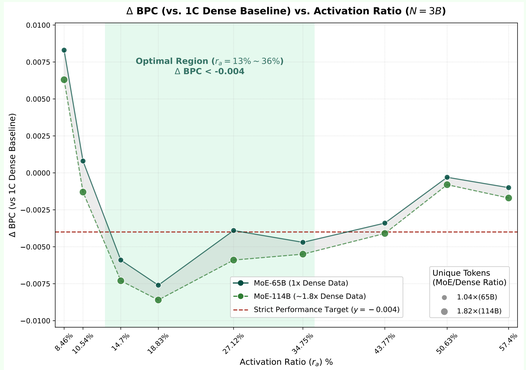


Figure 5: **Finding: optimal region also around 20%, Data Reuse works.** MoE is comparable with Dense LLM at same N, same C, and even same unique token.

4. Step1: Architecture Search (P2: Experiments Results)

N	N_a	M	H	D_h	L	E	K	D_e	D_{se}	Scheme	$\mathcal{L}$	Conclusion
2.02B	346M	8.77e8	22	64	16	35	2	800	1600	full+SE	1.6813	interleave>full
2.02B	346M	8.77e8	22	64	16	68	2	800	1600	interleave+SE	1.6766	
2.02B	346M	8.77e8	22	64	16	70	4	800	0	interleave	1.6697	
2.15B	366M	6.63e9	11	128	16	85	5	352	1760	ibmse+SE	1.6700	ibmse+SE is best
2.15B	366M	6.63e9	11	128	16	85	5	800	1760	ibmse+SE	1.6537	
2.15B	367M	6.63e9	11	128	16	70	4	800	0	interleave	1.6737	
2.15B	367M	6.63e9	22	64	16	70	4	800	0	interleave	1.6920	
2.15B	366M	9.31e8	22	64	17	37	4	800	0	ibmse	1.6752	D_{se} ratio impacts little
2.15B	368M	9.31e8	22	64	17	35	2	800	1600	ibmse+SE	1.6732	
2.15B	368M	9.31e8	22	64	17	35	2	800	1600	ibmse+SE	1.6726	

N	N_a	r_a (%)	M	E	K	D_e	D_{se}	$\mathcal{L}$
2.15B	591M	27.47	8.00e9	8	1	3528	3528	2.0470
2.15B	591M	27.40	8.00e9	88	11	320	3520	**2.0338**
2.15B	949M	44.00	1.01e10	8	2	3176	6352	**1.9996**
2.15B	948M	44.05	1.01e10	88	22	288	6336	2.0266
2.15B	1.24B	57.57	1.19e10	8	3	2888	8664	**2.0156**
2.11B	1.22B	57.68	1.18e10	88	33	256	8448	2.0235

N	N_a	r_a (%)	M	E	K	D_e	D_{se}	Norm	$\mathcal{L}$	$\mathcal{L}_{balance}$
2.15B	368M	17.08	9.31e8	35	2	800	1600	Y	1.6726	**1.355**
2.15B	368M	17.08	9.31e8	35	2	800	1600	N	1.6712	1.452
2.15B	368M	17.08	9.31e8	37	4	800	0	Y	1.6752	**1.409**
2.15B	368M	17.08	9.31e8	37	4	800	0	N	1.6750	1.440

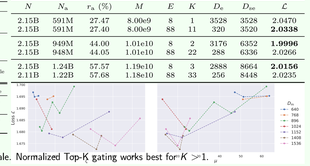


Table 1: Architecture search results and ablation studies at the 2B scale. Normalized Top-K gating works best for $K > 1$.

7. Step3B: Scaling to 7B — AR Analysis & Data Reuse

Scale-invariance: Optimal region also around 20% (13%–30%)

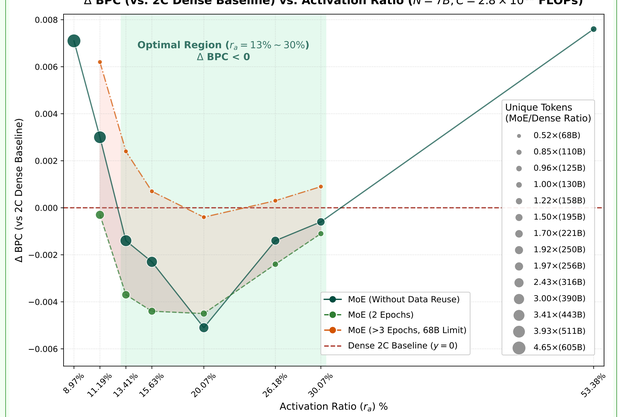


Figure 6: 7B Scale: ΔBPC vs 2C Dense Baseline under various Data Reuse configurations.

⋆ The Sweet Spot — MoE-68B-Exp4 at $r_a = 20\%$

1/2 compute of Dense-2C, 1/2 unique tokens, same N, BPC matches Dense-2C1, at 21.5% per-token FLOPs ⇒ ∼ 5× inference speedup at equal memory.

Model / Strategy	r_a (%)	Unique (D)	Epochs	FLOPs/Tok	BPC	Δ vs 2C
Dense 1C	100%	68 B	1.00	100% (4.21e10)	0.4736	+0.0142
Dense 2C (Red line)	100%	130 B	1.00	100% (4.21e10)	0.4594	**0.0000**
MoE-Unique (Blue line)	11.19%	511 B	1.00	13.3% (5.59e9)	0.4624	+0.0030
MoE-Unique (Blue line)	13.41%	443 B	1.00	15.3% (6.46e9)	0.4580	-0.0014
MoE-Unique (Blue line)	15.63%	390 B	1.00	17.4% (7.33e9)	0.4571	-0.0023
MoE-Unique (Blue line)	20.07%	316 B	1.00	**21.5%** (9.07e9)	**0.4543**	**0.0051**
MoE-Unique (Blue line)	26.18%	250 B	1.00	27.2% (1.15e10)	0.4580	-0.0014
MoE-Loose (Green line)	11.19%	256 B	2.00	13.3% (5.59e9)	0.4591	-0.0003
MoE-Loose (Green line)	13.41%	221 B	2.00	15.3% (6.46e9)	0.4557	-0.0037
MoE-Loose (Green line)	15.63%	195 B	2.00	17.4% (7.33e9)	0.4550	-0.0044
MoE-Loose (Green line)	20.07%	158 B	2.00	**21.5%** (9.07e9)	**0.4549**	**-0.0045**
MoE-Loose (Green line)	26.18%	125 B	2.00	27.2% (1.15e10)	0.4570	-0.0024
MoE-68B (Orange line)	11.19%	68 B	7.52	13.3% (5.59e9)	0.4656	+0.0062
MoE-68B (Orange line)	13.41%	68 B	6.51	15.3% (6.46e9)	0.4618	+0.0024
MoE-68B (Orange line)	15.63%	68 B	5.74	17.4% (7.33e9)	0.4601	+0.0007
MoE-68B (Orange)	**20.07%**	**68 B**	**4.65**	**21.5%** (9.07e9)	**0.4590**	**-0.0004**
MoE-68B (Orange line)	26.18%	68 B	3.67	27.2% (1.15e10)	0.4597	+0.0003

Table 2: Comprehensive 7B Models Resource & Performance Comparison ($C_{MoE} = 0.5C_{Dense-2C}$)

Moderate Data Reuse Boosts MoE

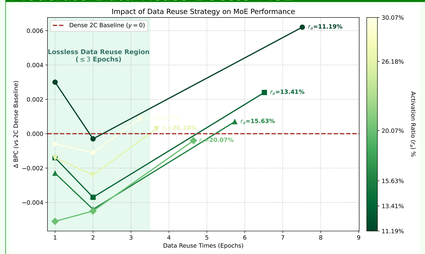


Figure 7: Impact of Data Reuse Strategy on MoE Performance.

Findings:

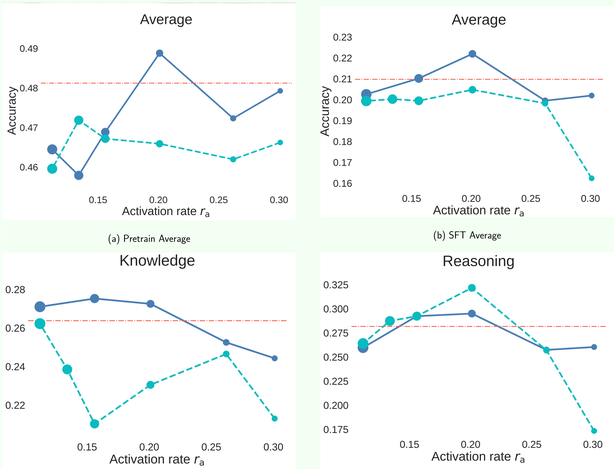
- 1ep→2ep **boosts performance** for $r_a \in \{13.4, 15.6, 20.1\}$%.
- ≤ 3 epochs **preserves MoE lead** over Dense-2C on a wide band.
- > 3 epochs **degrades**, sharply at sparsest configs.

MODELS TRAINED ∼250 LLMs — From-scratch (incl. 50 at 7B scale)

DATA PROCESSED 50T Token — Tokens cumulatively processed

SOTA RECIPE MoE + Reuse — For robust fully-trained LLMs

8. Step3C: Downstream Task Evaluation (7B)



(a) Pretrain Average (b) SFT Average (c) SFT Knowledge (d) SFT Reasoning

Figure 8: Downstream benchmark performance of 7B aligned models. Solid blue: MoE unique data; Dashed cyan: MoE strict data reuse ($\hat{D} = 68B$); Red dashed: Dense baseline ($D = 130B$).

- **Unique Tokens > Dense:** At $r_a \in [16\%, 20\%]$, the Average Benchmark score is **significantly higher** than the Dense LLM.
- **Unique Tokens = $\frac{1}{2}$ Dense:** At the optimal r_a, the Average Benchmark score **matches** the Dense-2C model.
- **Trade-off: Knowledge** recall degrades with fewer unique tokens; **Reasoning** is largely **preserved** and even surpasses Dense at optimal r_a with strict reuse.

9. Take Aways

- **MoE Surpasses Dense With Equal Total Model Size:** MoE can strictly outperform Dense LLMs even when Total Parameters (N), Compute (C), and Unique Data are equalized.
- **Scale-Invariant Optimal Sparsity Region:** "Sparser is better" is a misconception. An optimal AR ($r_a^\star \approx 20\%$) intrinsically exists and consistently emerges across 2B, 3B, and 7B scales.
- **Data Reuse Enables Strict Equal Comparison:** ≤3 epochs of reuse offsets MoE's token demand, preserving (and sometimes boosting) performance; equal-$N/C/D$ MoE still wins.
- **For SOTA Pretrained LLM Model:**
  - **Sparse MoE + Heavy Data Scaling + Moderate Reuse (≤3 ep)**
  - **Too Sparse MoE** leads to performance degradation.

Contact:
Houyi Li
LLM Researcher