Not All Models Suit Expert Offloading

On Local Routing Consistency of Mixture-of-Expert Models

Jingcong Liang[1] Siyuan Wang[2] Miren Tian[3] Yitong Li[3] Duyu Tang[3]
Zhongyu Wei[1,4]

[1]Fudan University [2]University of Southern California
[3]Huawei Technologies Ltd. [4]Shanghai Innovation Institute

ICLR 2026

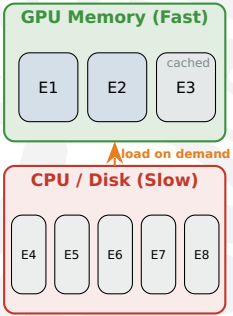
Background: MoE Models and Expert Offloading

Mixture-of-Experts (MoE) LLMs

- Replace dense FFN layers with multiple **expert** modules
- Only a **subset** of experts activated per token
- Efficient scaling — but **all experts reside in memory**

Expert Offloading

- Cache a subset of experts in **GPU memory**
- Store the rest in **CPU memory / disk**
- Problem: Frequent cache misses ⇒ **slow inference**



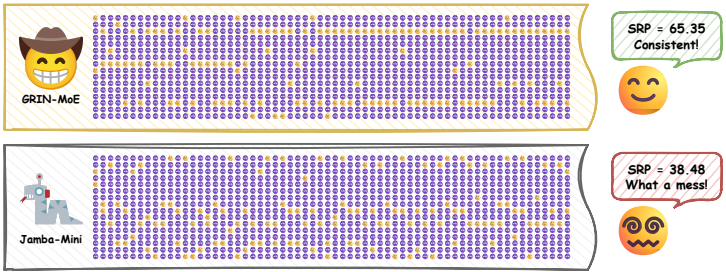
Key Question

Do consecutive tokens activate **similar experts**? Can we exploit this?

Local Routing Consistency: The Core Concept

Definition: Local routing consistency (LRC)

The property that consecutive tokens activate **similar sets of experts**, enabling effective expert caching.

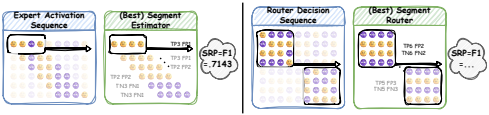


*But how do we **measure** LRC? And which models have it?*

Two Proposed Metrics

Segment Routing Best Performance (SRP)

- A **segment router** picks fixed experts for all tokens in a length-m segment
- SRP = best **F1 score** between original/segment router decisions
- **Parameter free** (only depends on m)
- Enables **individual expert** analysis



Segment Cache Best Hit Rate (SCH)

- An oracle cache with **limited size** = $\rho \times k$ (where k = active experts)
- Evicts experts **least activated** in next m tokens
- SCH = **hit rate** of this cache
- Closer to **real offloading systems**

SRP and SCH are **highly correlated** ($r > 0.97$ at $\rho = 1.5$), mutually reinforcing each other.

Large-Scale Analysis: 20 MoE Models → 4 Groups

Setup: 20 MoE LLMs (3B–57B), 11 text domains, 22,528 samples × 512 tokens

Models	**SRP** ($m=16$)
G1: LLaMA-MoE-v2, Yuan2.0, Qwen3, Phi-3.5, PowerMoE, OLMoE	> 0.50
G2: GRIN-MoE, Mixtral-8x7B, LLaMA-MoE-v1, MiniCPM, JetMoE	~ 0.48
G3: XVERSE, Jamba-Mini, DeepSeek-V2-Lite, DeepSeekMoE, Qwen2	~ 0.37
G4: NLLB-MoE, Qwen1.5, OpenMoE, SwitchTF	< 0.31

SRP are similar at $m=4$, but **diverge significantly at longer segments** ($m \geq 16$).

Not all MoE models benefit equally from expert offloading!

What Affects Local Routing Consistency?

Verified with **toy models** (1.43B OLMoE-like, each varying one factor):

Load Balance Trade-off

- **Local** load balance trade off LRC
- But **global** load balance can coexist with high LRC
- Domain-specialized experts achieve both

Shared Experts Hurt

- **All** G1–G2 models have **no** shared experts
- Shared experts bypass the MoE routing
- Reduce expert **combination space** ⇒ less flexibility

Expert Combination Space

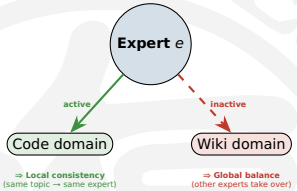
- Fewer experts or fewer activated ⇒ fewer **combinations**
- Restricts router **flexibility** for local adjustments
- More combinations ⇒ higher LRC

Domain Specialization Drives Routing Consistency

Key Insight

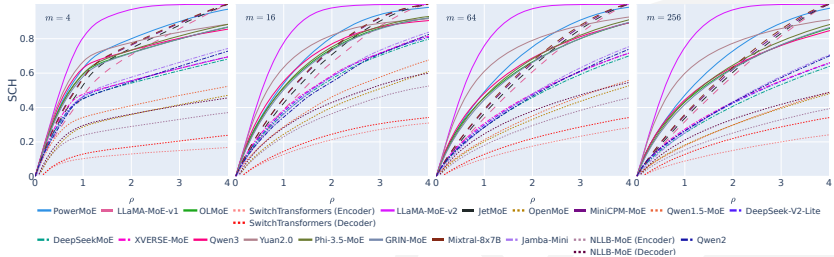
Domain-specialized experts contribute significantly more to local routing consistency than **vocabulary-specialized** ones. (Domain Spec. is more correlated to SRP.)

Why Domain Spec.?



Best of both worlds: high local routing consistency + good global load balance (Qwen3, OLMoE, GRIN-MoE)

Insights from SCH: Cache Size ≈ 2× Active Experts



SCH under different cache ratios ρ

SCH correlates with real caches

	$m=4$	$m=16$	$m=64$	$m=256$
LRU	81.2	90.4	93.1	97.5
LFU	77.4	88.7	92.8	99.2

Pearson's r (%) between SCH & real cache hit rates

Recommendation

Cache size = **2× active experts** balances effectiveness and efficiency for most models.

Conclusion and Takeaways

Contributions

1. **Two metrics** for local routing consistency:
   - **SRP** — parameter-free, fine-grained
   - **SCH** — cache-aware, practical
2. **Empirical analysis** of 20 MoE LLMs + toy models:
   - Local load balance vs. LRC trade-off
   - Shared experts harm LRC
   - Domain specialization helps LRC
3. **Practical guideline:**
   Cache $\approx 2\times$ active experts is optimal

Design Implications

For MoE model designers:

- Avoid shared experts if targeting edge deployment
- Encourage domain specialization
- Allow sufficient expert combination space

For offloading system designers:

- Not all MoE models benefit equally
- Check LRC before deploying
- $\rho = 2$ is a robust default cache size

Paper

Code

Thank you!