MoE Routing Bench

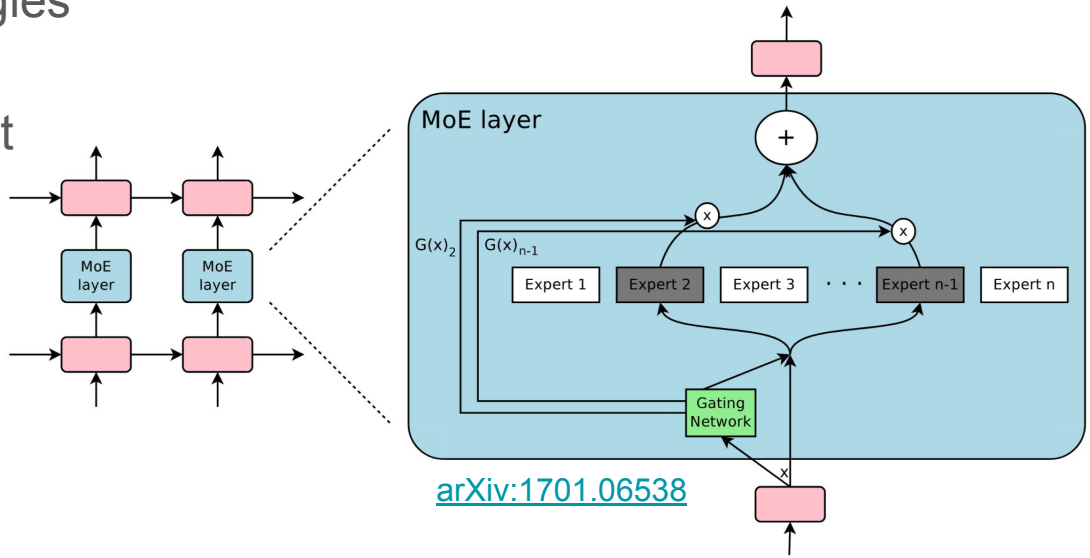
Routing Trade-offs & PERFT Insights

Yuning Xia (yx87), Daniel Zhang (dfz1), Shaoyang Zhang (sz121), Richard Xu (rgx1)

What is Mixture of Experts (MoE)?

Mixture of experts offers a way to maintain model capacity compared to typical dense neural networks with much lower computational cost.

- Test several routing strategies to analyze trade offs
- Analyze parameter efficient fine-tuning (PEFT) strategies
- Created a benchmarking framework on GitHub



arXiv:1701.06538

The Routing Bottleneck: Why MoE Performance is a Balancing Act

Mixture-of-Experts scales model capacity efficiently, but its effectiveness hinges on the routing algorithm. Poor routing creates a cascade of problems:

Load Imbalance
Some experts are overworked while others are idle, leading to inefficient hardware use and potential routing collapse.

Token Dropping
When expert capacity is exceeded, tokens are discarded, resulting in direct information loss and degraded model quality.

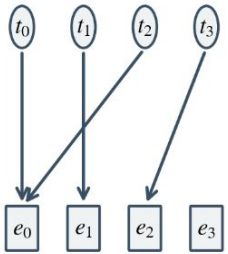
Training Instability
The discrete, non-differentiable nature of hard routing decisions complicates optimization and can hinder convergence.

Suboptimal Specialization
Without effective routing, experts may fail to develop distinct, meaningful functions, negating the primary benefit of the MoE architecture.

MoE Routing Strategies Studied

Top-1

$\mathcal{I} = \arg\max(\mathbf{g})$

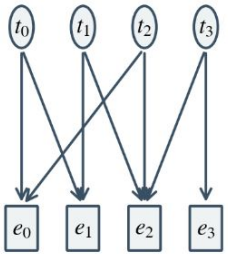


T→1 expert

$k_{eff} = 1$
fastest
high drop

Top-k Hard

$\mathcal{I} = \text{TopK}(\mathbf{g})$

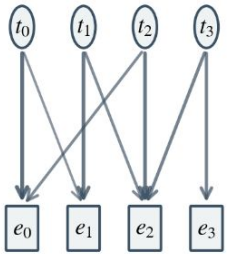


T→k experts, uniform

$k_{eff} = k$
no gates
moderate

Soft Top-k

$w = \text{softmax}(\mathbf{g}_{\mathcal{I}})$

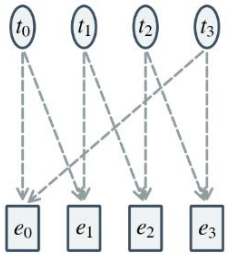


T→k experts, learned

$k_{eff} = k$
learned w
best PPL

Hash

$\mathcal{I} = h(\text{pos}) \bmod E$

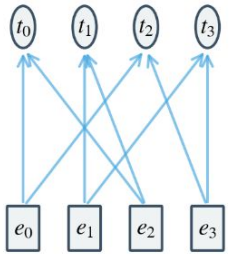


deterministic, uniform

$k_{eff} = k$
$w = 1/k$
CV=0

Expert Choice

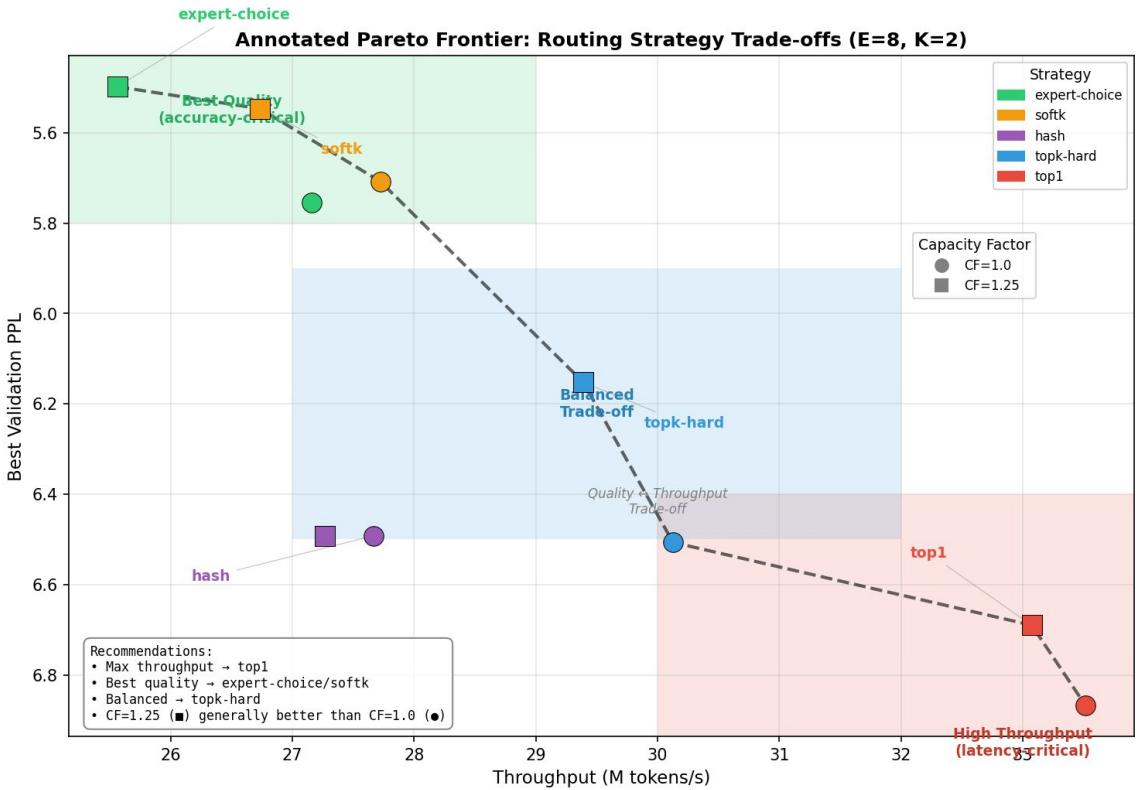
$\mathcal{I}_e = \text{TopK}_e(\mathbf{g}^T)$



E→T selection

$k_{eff} = k$
E→T first
balanced

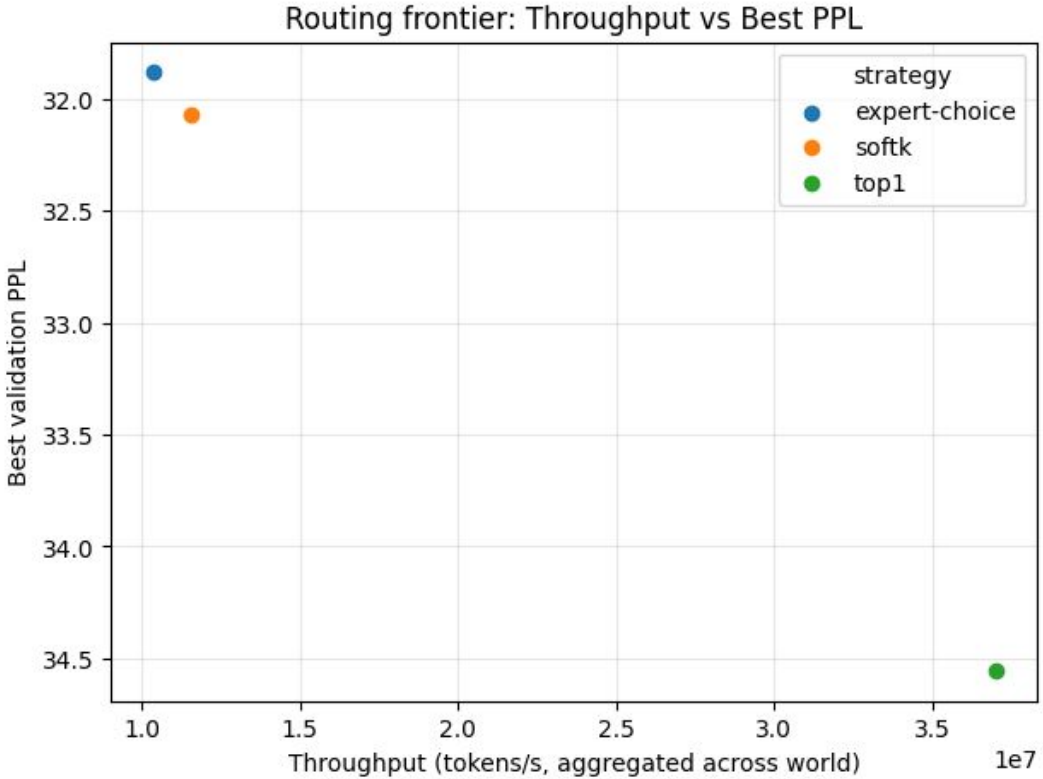
Pareto Frontier for Perplexity vs. Throughput



Benchmark notes:

- Model: TinyMoE
- Dataset: TinyStories
- Hardware: 4×L40S (BF16), DDP; 500–2000 steps
- Capacity factor controls allocated memory; chosen values were tuned

Large Scale Analysis Demonstrates Scalability

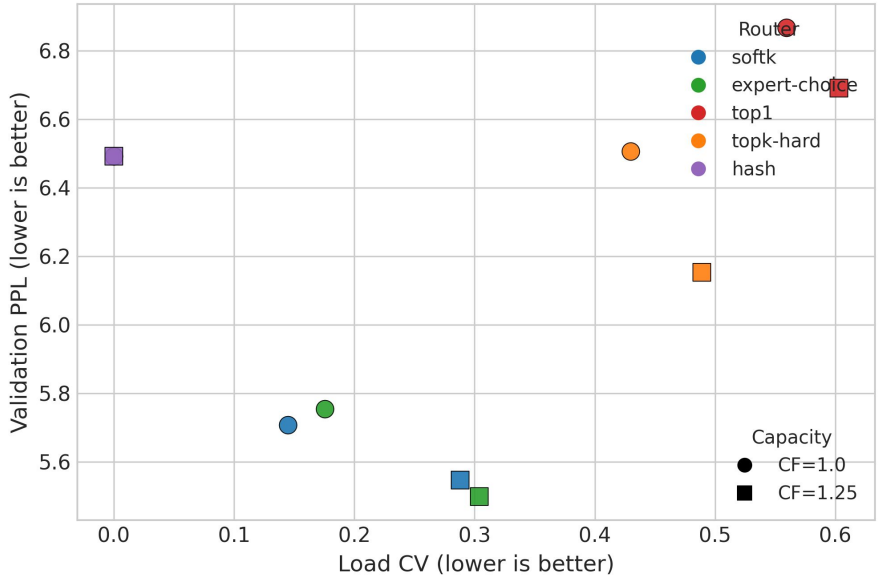


- Uses 32 experts instead of 8
- Run on sub-word tokens instead of character tokens
- Tradeoffs remain the same, even at larger scale

Specialization and Load Balancing Tension

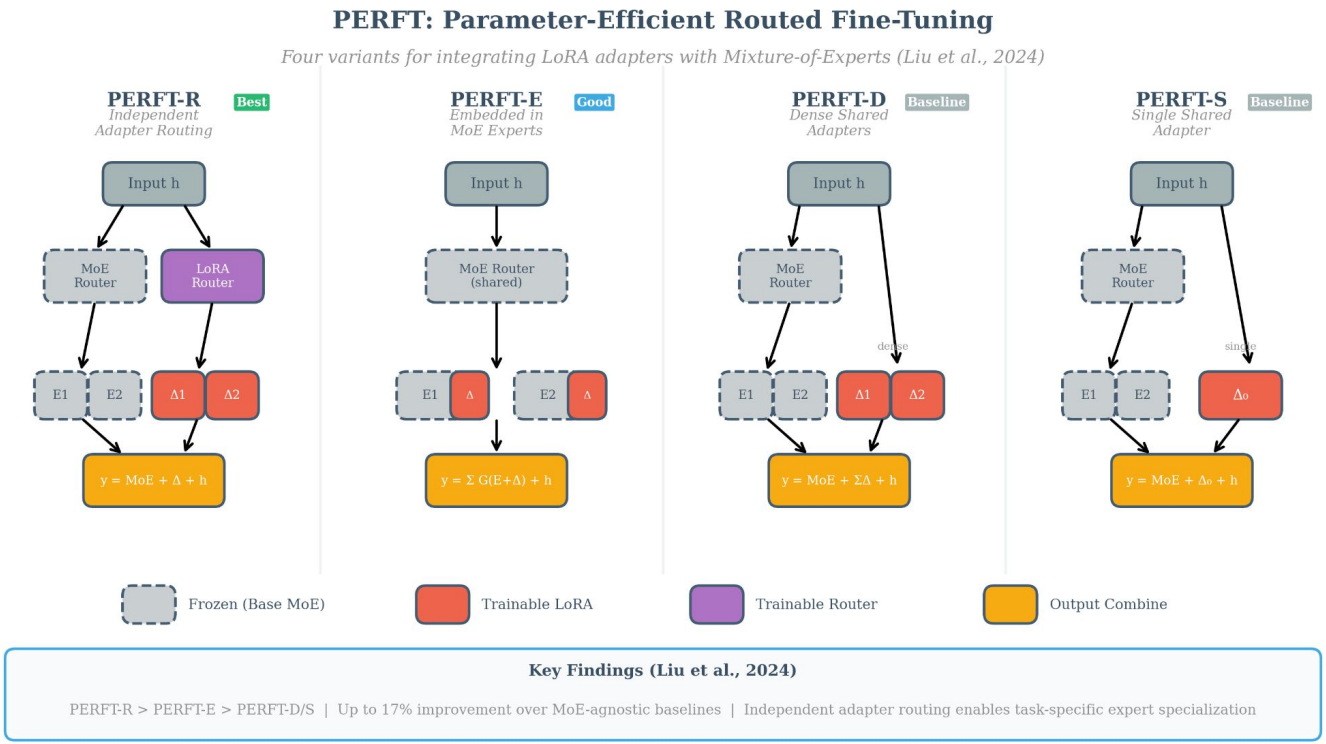
Want experts to be specialized but trained equally

- Aggressive specialization leads to expert collapse
- Aggressive load balancing leads to loss of specialization

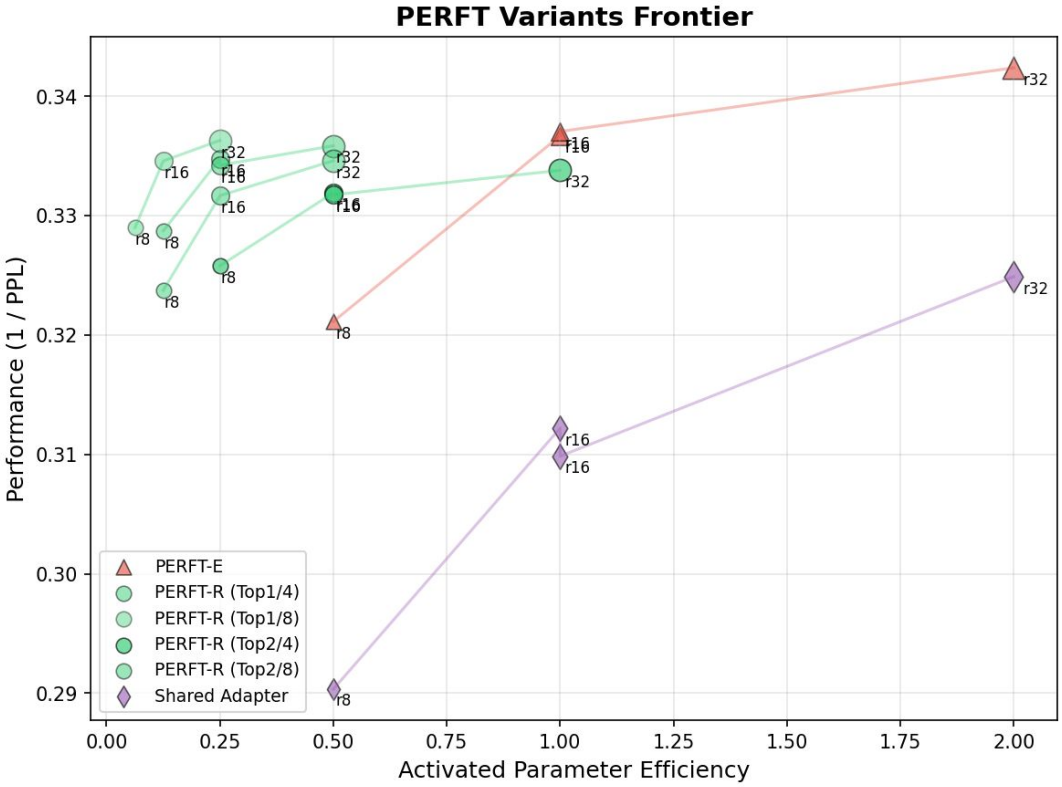


Strategy	Gate Entropy	Load CV
hash	0.69 (uniform)	0.00
expert_choice	0.33-0.42	0.54-0.56
softk	0.44-0.52	0.62-0.64
topk-hard	0.00	0.71
top1	0.00	0.86

PEFT Strategies for MoE Routing (PERFT)



PERFT Frontier Shows Tradeoff in Adapter Routing



This benchmark can be extended to explore the next frontier of dynamic and sequential routing

Auxiliary-Loss-Free Balancing

Integrating modern techniques like DeepSeek-V3's dynamic bias adjustment to remove the quality-balance tension from auxiliary losses.

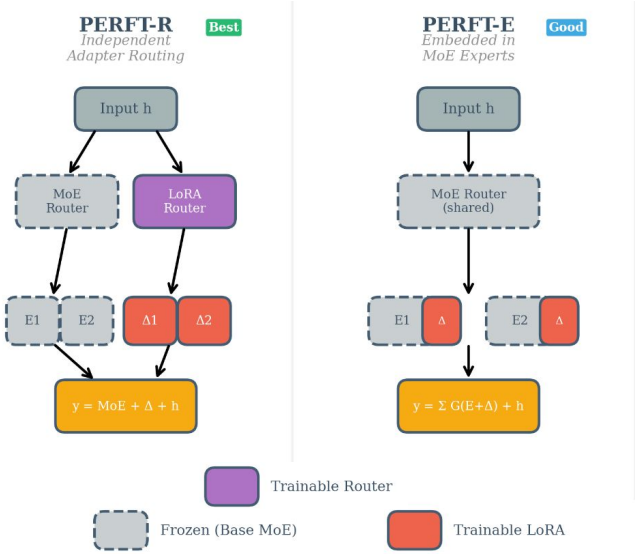
Sequential vs. Parallel Experts

Comparing our parallel routing to architectures like Chain-of-Experts (CoE) to quantify the trade-offs for multi-step reasoning tasks.

Serving Integration

Directly plugging the benchmark into serving stacks like vLLM to measure end-to-end latency and validate performance in a production environment.

Discussion of broader context of MoE research




```
Static ◄──────────────────────────────────────────► Dynamic

hash          top1          topk-hard      softk          expert_choice
  │             │              │              │               │
  ▼             ▼              ▼              ▼               ▼
Fixed         Learned        Learned        Learned +       Learned +
assignment    discrete       discrete       soft weights    inverted direction
              (deterministic)               (probabilistic)
```

- Newest/most performant methods lend themselves to dynamic routing
- The most performant PERFT schemes also benefit from the balance offered by dynamic routing

Thank You

Questions?

Explore the benchmark and full report:
github.com/yuninxia/moe-routing-bench