# Cross-Layer Parallel Routing for MoE Inference Optimization

**Project Proposal**

Prashant Shihora(ps5226), Aryan Tomar(at6304)

## 1. Goal/Objective

Conduct comprehensive research on cross-layer routing predictability in Mixture-of-Experts (MoE) models to validate the feasibility of speculative routing optimizations. Specifically, we will quantify inter-layer expert affinity patterns across different task domains in Mixtral-8x7B [5], measuring single-hop and multi-hop prediction accuracy to determine whether routing decisions exhibit sufficient correlation to enable future runtime optimizations.

## 2. Challenges

MoE models like Mixtral-8x7B activate only 2 out of 8 experts per token, achieving massive parameter counts with reasonable inference costs [5]. However, despite using far fewer FLOPs than dense models, MoE inference is surprisingly slow. Recent profiling reveals that in large-scale distributed MoE deployments, **routing and dispatch overhead consumes 47-59% of the total per-layer time** [2]. The bottleneck is not computation, but how we schedule it.

At every MoE layer today, three stages run sequentially: (1) **Routing**: For each token, run a gating network to decide which 2 experts to activate; (2) **Dispatch**: Reorganize token data and scatter it to selected expert buffers; (3) **Compute**: The experts process their assigned tokens. While Layer N's experts compute, Layer N+1's routing sits idle. We lose substantial time per layer waiting for bookkeeping that could potentially run in parallel with computation—but only if routing decisions are predictable across layers.

**The core research question:** Do routing decisions exhibit sufficient inter-layer correlation to enable speculative parallel execution? How does this predictability vary across task domains and layer depths?

## 3. Approach/Techniques

Our approach focuses on **statistical validation of cross-layer routing patterns** through comprehensive profiling and analysis:

### Phase 1: Profiling and Data Collection

- Instrument vLLM's MoE layer to log routing decisions at every layer for every token
- Run Mixtral-8x7B inference on two distinct task domains (coding and mathematical reasoning)
- Collect routing decision logs including: layer index, token ID, selected expert indices, and timing breakdowns
- Target: 10,000+ routing decisions across 32 transformer layers [5]

### Phase 2: Multi-Hop Predictability Analysis

- **Single-hop prediction**: Compute P(expert at layer N+1 — expert at layer N) for all layer pairs
- **Multi-hop path analysis**: Compute P(expert at layer N+k — experts at layers 0...N) for k=1 to 31
- **Path frequency analysis**: Identify frequently occurring expert sequences across the full 32-layer depth
- **Cross-domain comparison**: Quantify whether routing patterns differ between coding and mathematical reasoning tasks
- Generate correlation matrices, heatmaps, and statistical distributions

This analysis will validate whether the inter-layer expert affinity patterns observed in prior work [1] hold with sufficient strength to support speculative routing optimizations.

## 4. Implementation Details

**Hardware:** NVIDIA A100 (80GB) or H100 on NYU Greene cluster or cloud GPU infrastructure. Single GPU setup (no distributed infrastructure required).

**Software:** vLLM for MoE inference with Mixtral-8x7B; PyTorch Profiler for latency analysis; Python with NumPy, Pandas for statistical analysis; Matplotlib/Seaborn for visualizations; Instrument `vllm/model_executor/layers` to add logging hooks.

**Dataset: HumanEval** (164 Python coding problems) - coding/syntax domain, standard OpenAI benchmark used in Mixtral evaluations; **GSM8K** (grade school math reasoning, subset of 500 from standard benchmark) - mathematical reasoning domain. These represent distinct semantic domains, allowing us to test whether routing patterns are domain-dependent. Both are industry-standard benchmarks used in top-tier MoE research.

## 5. Demo Planned

**Deliverable Format:** Interactive Jupyter notebook with embedded visualizations and analysis.

**Demo Components:**

1. **Live Profiling Dashboard**: Real-time display of routing decisions as Mixtral-8x7B processes sample prompts; per-layer timing breakdown; validation of routing overhead claims.
2. **Statistical Visualizations**: Inter-layer affinity heatmaps (32×32 matrices); prediction accuracy curves (1-hop to 31-hop); path frequency histograms; domain comparison plots (HumanEval vs. GSM8K).
3. **Statistical Summary Tables**: Conditional probability matrices; single-step and multi-step prediction accuracy percentages; cross-layer correlation coefficients.

**Reproducibility:** All profiling logs, analysis scripts, and the complete notebook will be provided.

*Note: Following consultation with Professor Sura, we focused this project on comprehensive statistical analysis of routing predictability. She confirmed this analysis constitutes sufficient course project scope, with actual system implementation deferred as future work.*

## 6. References

1. Huang, Y., Liu, Z., et al. "Exploiting Inter-Layer Expert Affinity for Accelerating Mixture-of-Experts Model Inference." *IEEE IPDPS 2024*. arXiv:2401.08383.
2. Zhang, J., Zhu, Y., et al. "Communication Efficient Parallel MoE Inference with Speculative Token and Expert Pre-scheduling." arXiv:2503.04398, March 2025.
3. Zhou, Y., Zhao, T., et al. "SP-MoE: Speculative Decoding and Prefetching for Accelerating MoE-based Model Inference." arXiv:2510.10302, October 2025.
4. Chen, X., Guo, J., et al. "Layerwise Recurrent Router for Mixture-of-Experts." arXiv:2408.06793, August 2024.
5. Jiang, A., Sablayrolles, A., et al. "Mixtral of Experts." arXiv:2401.04088, January 2024.

**How Our Project Builds on These Works: ExFlow [1]** analyzes inter-layer expert affinity for static expert placement. We extend their observation to validate whether correlations support runtime speculative execution and multi-hop predictions they did not explore. **Speculative MoE [2]** optimizes within-layer communication in distributed training. We target cross-layer pipelining—overlapping Layer N's computation with Layer N+1's routing preparation. **SP-MoE [3]** prefetches experts from CPU to GPU for I/O bottlenecks. We target routing computation overhead in on-GPU systems where all experts fit in memory.

# Cross-Layer Parallel Routing for MoE Inference Optimization

**Project Proposal**

Prashant Shihora(ps5226), Aryan Tomar(at6304)

## 1. Goal/Objective

Conduct comprehensive research on cross-layer routing predictability in Mixture-of-Experts (MoE) models to validate the feasibility of speculative routing optimizations. Specifically, we will quantify inter-layer expert affinity patterns across different task domains in Mixtral-8x7B [5], measuring single-hop and multi-hop prediction accuracy to determine whether routing decisions exhibit sufficient correlation to enable future runtime optimizations.

## 2. Challenges

MoE models like Mixtral-8x7B activate only 2 out of 8 experts per token, achieving massive parameter counts with reasonable inference costs [5]. However, despite using far fewer FLOPs than dense models, MoE inference is surprisingly slow. Recent profiling reveals that in large-scale distributed MoE deployments, **routing and dispatch overhead consumes 47-59% of the total per-layer time** [2]. The bottleneck is not computation, but how we schedule it.

At every MoE layer today, three stages run sequentially: (1) **Routing**: For each token, run a gating network to decide which 2 experts to activate; (2) **Dispatch**: Reorganize token data and scatter it to selected expert buffers; (3) **Compute**: The experts process their assigned tokens. While Layer N's experts compute, Layer N+1's routing sits idle. We lose substantial time per layer waiting for bookkeeping that could potentially run in parallel with computation—but only if routing decisions are predictable across layers.

**The core research question:** Do routing decisions exhibit sufficient inter-layer correlation to enable speculative parallel execution? How does this predictability vary across task domains and layer depths?

## 3. Approach/Techniques

Our approach focuses on **statistical validation of cross-layer routing patterns** through comprehensive profiling and analysis:

**Phase 1: Profiling and Data Collection**

- Instrument vLLM's MoE layer to log routing decisions at every layer for every token
- Run Mixtral-8x7B inference on two distinct task domains (coding and mathematical reasoning)
- Collect routing decision logs including: layer index, token ID, selected expert indices, and timing breakdowns
- Target: 10,000+ routing decisions across 32 transformer layers [5]

**Phase 2: Multi-Hop Predictability Analysis**

- **Single-hop prediction**: Compute P(expert at layer N+1 — expert at layer N) for all layer pairs
- **Multi-hop path analysis**: Compute P(expert at layer N+k — experts at layers 0...N) for k=1 to 31
- **Path frequency analysis**: Identify frequently occurring expert sequences across the full 32-layer depth
- **Cross-domain comparison**: Quantify whether routing patterns differ between coding and mathematical reasoning tasks
- Generate correlation matrices, heatmaps, and statistical distributions

This analysis will validate whether the inter-layer expert affinity patterns observed in prior work [1] hold with sufficient strength to support speculative routing optimizations.

## 4. Implementation Details

**Hardware:** NVIDIA A100 (80GB) or H100 on NYU Greene cluster or cloud GPU infrastructure. Single GPU setup (no distributed infrastructure required).

**Software:** vLLM for MoE inference with Mixtral-8x7B; PyTorch Profiler for latency analysis; Python with NumPy, Pandas for statistical analysis; Matplotlib/Seaborn for visualizations; Instrument `vllm/model_executor/layers` to add logging hooks.

**Dataset: HumanEval** (164 Python coding problems) - coding/syntax domain, standard OpenAI benchmark used in Mixtral evaluations; **GSM8K** (grade school math reasoning, subset of 500 from standard benchmark) - mathematical reasoning domain. These represent distinct semantic domains, allowing us to test whether routing patterns are domain-dependent. Both are industry-standard benchmarks used in top-tier MoE research.

## 5. Demo Planned

**Deliverable Format:** Interactive Jupyter notebook with embedded visualizations and analysis.

**Demo Components:**

1. **Live Profiling Dashboard**: Real-time display of routing decisions as Mixtral-8x7B processes sample prompts; per-layer timing breakdown; validation of routing overhead claims.
2. **Statistical Visualizations**: Inter-layer affinity heatmaps (32×32 matrices); prediction accuracy curves (1-hop to 31-hop); path frequency histograms; domain comparison plots (HumanEval vs. GSM8K).
3. **Statistical Summary Tables**: Conditional probability matrices; single-step and multi-step prediction accuracy percentages; cross-layer correlation coefficients.

**Reproducibility:** All profiling logs, analysis scripts, and the complete notebook will be provided.

*Note: Following consultation with Professor Sura, we focused this project on comprehensive statistical analysis of routing predictability. She confirmed this analysis constitutes sufficient course project scope, with actual system implementation deferred as future work.*

## 6. References

1. Huang, Y., Liu, Z., et al. “Exploiting Inter-Layer Expert Affinity for Accelerating Mixture-of-Experts Model Inference.” *IEEE IPDPS 2024*. arXiv:2401.08383.
2. Zhang, J., Zhu, Y., et al. “Communication Efficient Parallel MoE Inference with Speculative Token and Expert Pre-scheduling.” arXiv:2503.04398, March 2025.
3. Zhou, Y., Zhao, T., et al. “SP-MoE: Speculative Decoding and Prefetching for Accelerating MoE-based Model Inference.” arXiv:2510.10302, October 2025.
4. Chen, X., Guo, J., et al. “Layerwise Recurrent Router for Mixture-of-Experts.” arXiv:2408.06793, August 2024.
5. Jiang, A., Sablayrolles, A., et al. “Mixtral of Experts.” arXiv:2401.04088, January 2024.

**How Our Project Builds on These Works: ExFlow [1]** analyzes inter-layer expert affinity for static expert placement. We extend their observation to validate whether correlations support runtime speculative execution and multi-hop predictions they did not explore. **Speculative MoE [2]** optimizes within-layer communication in distributed training. We target cross-layer pipelining—overlapping Layer N’s computation with Layer N+1’s routing preparation. **SP-MoE [3]** prefetches experts from CPU to GPU for I/O bottlenecks. We target routing computation overhead in on-GPU systems where all experts fit in memory.