CUR-MoE: Portable Mixture-of-Experts with High-Ratio Compression

Ritesh Bhirud
rbhirud@umass.edu
University of Massachusetts Amherst
Amherst, Massachusetts, USA

Rabab Alomairy (Advisor)
rabab.alomairy@mit.edu
Massachusetts Institute of Technology
Cambridge, Massachusetts, USA

1 INTRODUCTION

Mixture-of-Experts (MoE) architectures enable trillion-parameter models with sub-linear computational growth, representing a significant advancement for scalable HPC-driven AI workloads. However, practical deployment faces three major barriers: (1) prohibitive memory scaling for MoE models in HPC settings [1], (2) limited interpretability of existing compression approaches that use abstract factorization [3], and (3) vendor-specific implementations that hinder efficient utilization of heterogeneous multi-GPU clusters.

To address these challenges, we present the first Julia-based Mixture-of-Experts framework, introducing novel CUR decomposition-based compression [3] combined with a hardware-agnostic design for cross-vendor portability. While SVD-based compression achieves effective model compression [2], our CUR-MoE approach provides comparable compression performance with enhanced interpretability through preserved column/row structure. CUR-MoE maintains functionality at 70% compression ratios (35.29 perplexity), demonstrating the viability of interpretable compression alternatives. Built on Julia's LLVM compilation pipeline, the framework eliminates Python's GIL overhead and CUDA-only dependencies, enabling seamless execution across NVIDIA, AMD, Intel, and Apple GPUs. Empirical results demonstrate consistent speedups of 6.3× and 5.3× on NVIDIA and AMD GPUs respectively, while supporting interpretable and vendor-independent distributed training.

2 METHODOLOGY

2.1 CUR Decomposition for Expert Compression

While existing SVD-based compression provides effective model compression [2], it uses abstract factorization that limits interpretability for production analysis [3]. We implement CUR decomposition for MoE experts, selecting matrix columns and rows based on statistical leverage scores rather than energy concentration [3].

CUR decomposes expert matrices $W \in \mathbb{R}^{m \times n}$ as $W \approx CUR$, where C contains selected columns from W, R contains selected rows from W, and U is a small intersection matrix. Leverage scores from top-k singular vectors guide column/row selection, preserving essential weight patterns while maintaining interpretability through actual data elements rather than mathematical abstractions [3].

2.2 Hardware-Agnostic Implementation

Julia's LLVM compilation enables comprehensive cross-vendor portability through unified abstractions. The portable MoE layer provides automatic GPU detection, dynamic workload distribution, and mixed-precision support (FP16/FP32/FP64) for heterogeneous HPC environments containing mixed GPU configurations.

Cross-vendor validation on 512→2048→512 architectures demonstrates consistent acceleration: NVIDIA RTX 5060 Ti achieves 1.312ms (6.3× speedup), AMD RX 7600 reaches 1.572ms (5.3× speedup), versus 8.276ms CPU baseline. This consistency eliminates performance uncertainty in heterogeneous HPC deployments.

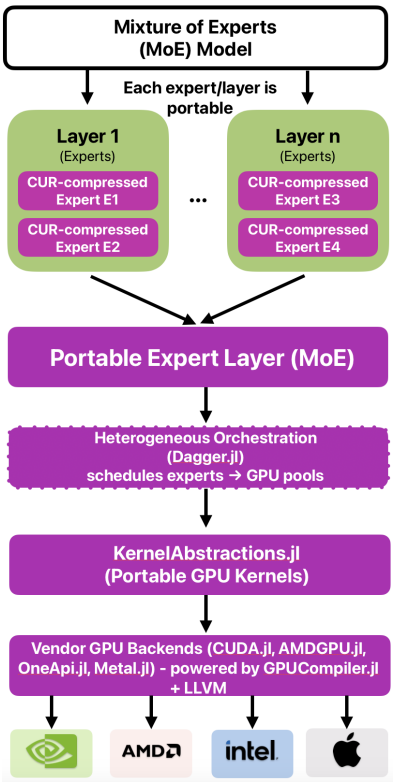

Figure 1: Portable MoE architecture with CUR-compressed experts across diverse GPU backends.

2.3 Comprehensive Gating Mechanisms

We implement multiple routing strategies with quantitative analysis for optimal HPC workload selection. ExpertChoice achieves optimal load distribution (0.815 balance coefficient, 0.954 utilization, 167.9 tokens/second). TopK provides robust performance (0.713 balance, 0.849 utilization, 138.4 tokens/second). Switch enables simplified architectures (0.672 balance, 0.749 utilization, 170.5 tokens/second).

Gating Mechanism	Load Balance (higher = better)	Expert Utilization (higher = better)	Inference Speed (tokens/sec)	Text Diversity (0-1 scale)	Routing Stability (higher = better)
Random	0.955	1.008	134.6	0.809	0.870
TopK	0.713	0.849	135.4	0.845	0.924
Switch	0.672	0.749	170.5	0.810	0.873
StochasticTopK	0.739	0.853	153.1	0.829	0.842
ExpertChoice	0.815	0.954	167.9	0.770	0.899
SoftMoE	0.779	0.911	141.8	0.785	0.904
Hash	0.598	0.664	155.6	0.830	0.988
SharedExpert	0.810	0.907	143.2	0.791	0.908

Table 1: Performance comparison of MoE gating mechanisms across key metrics for HPC deployment optimization. Green highlighting indicates top 3 performers in each category.

3 RESULTS AND ANALYSIS

3.1 Compression Performance Analysis

WikiText-2 benchmark evaluation (standard dataset with 100M+ Wikipedia tokens) demonstrates comparable compression performance between CUR-MoE and existing SVD-based approaches [2]. At moderate compression (20-40%), both methods perform equivalently with perplexity increasing from 3.92 to 8.66, confirming effective compression capabilities.

At 70% compression, CUR-MoE maintains functionality (35.29 perplexity, 9× degradation from baseline), demonstrating robustness in extreme compression regimes. SVD-MoE approaches were not tested at this compression ratio in existing literature.

The key distinction lies in interpretability: CUR maintains expert interpretability through preserved column/row structures essential for production debugging [3]. Unlike SVD's abstract factorization that creates mathematical abstractions [3], CUR preserves actual data elements enabling expert behavior analysis crucial for HPC deployment optimization.

3.2 HPC Scalability and Performance

The MoE architecture exhibits natural parallelization through independent expert computations, enabling near-linear scalability across heterogeneous clusters. Mixed-architecture support optimizes utilization through automatic load balancing based on hardware capabilities and memory hierarchies.

Routing stability analysis quantifies trade-offs across gating mechanisms: Random (0.870 stability, 0.809 diversity), TopK (0.924 stability, 0.845 diversity), ExpertChoice (0.899 stability, 0.770 diversity), enabling systematic mechanism selection for specific HPC workload characteristics.

3.3 Expert Interpretability

CUR uniquely exposes expert specialization through interpretable column/row selections [3], enabling enhanced monitoring and debugging in production HPC environments. This addresses interpretability limitations in large-scale MoE deployments by providing debugging capabilities for managing large-scale MoE deployments.

4 CONTRIBUTIONS AND HPC IMPACT

This work delivers significant advancements for portable MoE deployments:

Memory & Performance: High-ratio compression enables larger model deployment within existing constraints while achieving consistent 5-6× GPU acceleration across diverse hardware ecosystems.

Vendor Independence: Cross-platform implementation eliminates hardware lock-in, enabling optimal heterogeneous cluster utilization while reducing procurement costs and maximizing existing hardware investments.

Production Capabilities: First Julia-based MoE framework enables portable, interpretable expert analysis across heterogeneous HPC hardware architectures.

Interpretable Compression: CUR decomposition provides alternative compression approach with preserved structural information, enabling expert behavior analysis unavailable in abstract factorization methods [3].

5 CONCLUSION

We present a novel MoE implementation combining CUR decomposition [3] with hardware-agnostic portability through Julia's LLVM pipeline. Results demonstrate the viability of interpretable compression alternatives: CUR-MoE maintains functionality at 70% compression ratios while providing comparable performance to existing approaches [2]. The framework achieves consistent 5-6× GPU acceleration across diverse hardware ecosystems.

This establishes foundational capabilities for sparse model deployment in heterogeneous HPC environments, offering an interpretable alternative to existing compression methods while enabling cross-vendor portability. Future directions include MPI/Slurm integration for scalable deployment, establishing the foundation for portable, scalable and interpretable MoE deployment in next-generation HPC infrastructures.

REFERENCES

[1] D. Dai, C. Deng, K. Zhao, R. X. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu, Z. Xie, Y. K. Li, P. Huang, F. Luo, C. Zheng, W. Liu, and G. Liang. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

[2] W. Li, L. Lujun, H. Hao, G. Gu, Y. Liang, H. Huang, M. G. Lee, S. Sun, W. Xue, and Y. Guo. MoE-SVD: Structured mixture-of-experts (LLM) compression via singular value decomposition. In *Proceedings of the Forty-second International Conference on Machine Learning (ICML '24)*.

[3] M. W. Mahoney and P. Drineas. CUR matrix decomposition for improved data analysis. *Proc. Natl. Acad. Sci. U.S.A.*, vol. 106, no. 3, pp. 697–702, 2009.