Beyond Memory Limits: Scaling Mixture-of-Experts Models

Scaling Laws and Model Training Framework for Compute-Efficient Massive Models

Wesley Medford* John McBride†

Abstract

The rapid progress of large language models (LLMs) is hitting a hardware wall: memory bandwidth and device capacity now limit scale more than raw compute. Sparse mixture-of-experts (MoE) Transformers mitigate this by activating only a small fraction of parameters per token, yet state-of-the-art models still top out at a few hundred experts and remain memory-bound.

We make a two-part contribution. First, we propose a set of fine-grained MoE scaling laws that analytically predict super-linear gains in throughput and energy efficiency as the number of experts increases and their individual size shrinks—extending the intuition of Mixture-of-a-Million-Experts [] to a general theoretical framework. These laws assume balanced routing and the ability to stream weights from slower memory tiers.

Second, we introduce LLM720, an open-source training framework built to test these laws at scales beyond current practice. LLM720 integrates: (1) product-key routing that indexes at least 10^6 experts with sub-linear lookup

*wryanmedford@gmail.com
†jpmmcbride@gmail.com

cost compared to traditional top-k gating; (2) a fused Triton kernel that exploits the reversed size ratio between token batches and expert weights; and (3) hierarchical weight streaming that leverages Unified Virtual Addressing and cache-aware prefetching to operate outside VRAM limits.

We also define an ablation suite spanning constant-compute, constant-active-parameter, and sparsity-sweep regimes, thereby providing the first systematic roadmap for million-expert MoE evaluation. By unifying theory and system design, this work lays an empirical foundation for scaling MoE models well beyond contemporary memory constraints and invites the community to replicate, refine, and extend our findings.

1 Introduction

The last five years of large-language-model (LLM) research have appeared to follow a simple recipe: scale parameters, scale data, and harvest monotonic gains in accuracy. Early empirical laws quantified this trend by showing that test loss falls predictably as a power law in model size and training compute. However, continuing that trajectory now collides with a hardware wall—high-bandwidth memory (HBM) capacity and memory-access energy dominate cost long before GPUs reach their theoretical FLOP limits. Training or serving a dense-parameter model beyond a few hundred billion weights demands either exotic multi-node pipelines or aggressive precision trade-offs that sacrifice throughput.

Mixture-of-experts (MoE) Transformers address this bottleneck by activating only a small subset of weights per token. Conditional computation was first proven practical at 600B parameters in GShard [] and later at the trillion-parameter scale in Switch Transformer [], both of which used 64–128 experts per MoE layer. Subsequent work such as DeepSeek-V3 [] increased the pool to 256 experts and adopted FP8 training to curtail bandwidth pressure. Yet even these state-of-the-art systems remain memory-bound: every expert must reside in on-device memory, and routing imbalance can leave available bandwidth under-utilized. Recent work on Mixture-of-a-Million-Experts [] pushed the idea of fine-grained experts—that is, reducing each expert by orders of magnitude so that the expert pool can grow to

10^6 entries without exploding active FLOPs. Although PEER provided a promising prototype, the broader community still lacks (i) a general theoretical framework that explains when and why throughput or energy efficiency should improve as expert granularity increases and (ii) an open test-bed for validating such fine-grained scaling laws under realistic memory budgets.

This paper tackles both issues. In Section 3 we derive a set of fine-grained MoE scaling laws that extend prior intuition to predict super-linear gains in both throughput and energy efficiency if (i) routing remains balanced and (ii) expert weights can be streamed from slower memory tiers. In addition, we present LLM720—a purpose-built training framework that couples (i) product-key routing (to index at least 10^6 experts with sub-linear lookup cost based on the algorithm proposed in []), (ii) a fused Triton kernel that exploits the reversed size ratio between token batches and expert weights, and (iii) hierarchical weight streaming (based on Unified Virtual Addressing) to operate safely beyond VRAM limits. By unifying theory with an open implementation, we aim to shift LLM scaling from being memory-constrained to capacity-driven. Our work directly complements recent calls for scaling laws that incorporate inference memory and bandwidth costs [] and offers the community a replicable platform for memory-aware MoE research.

2 Background and Related Works

2.1 Foundational Mixture of Experts Works

Research on conditional computation predates Transformers. Jacobs and Jordan [] introduced the original Mixture of Experts framework for small feed-forward networks. The idea was scaled to sequence modelling in sparsely-gated MoE for RNN LMs [], which demonstrated that activating only 1–2 experts per token could increase parameter counts $20\times$ while keeping FLOPs nearly constant.

Modern Transformer variants soon followed. GShard applied top-2 gating to a 600B-parameter multilingual translation model, using expert-parallel all-to-all to

distribute tokens across accelerators []. Switch Transformer simplified routing to top-1, enabling a 1.6T-parameter model trained with 64 experts per layer []. In vision, V-MoE inserted sparse experts into vision transformers and matched dense models that were four times larger while saving inference compute []. These studies established MoE as a practical path to scaling—but only with tens or hundreds of experts resident in GPU memory.

2.2 Routing and Load-Balancing Innovations

Large expert pools amplify two classical MoE challenges: hot experts that monopolize tokens and cold experts that never learn. Several strategies have been introduced to address these issues:

- Auxiliary load-balance losses (popularized by Shazeer []) penalize high variance in token counts per expert.
- BASE Layers recast routing as a linear-assignment problem (solving a Hungarian matching every batch) to guarantee perfect expert balance without auxiliary terms [].
- Expert-Choice Routing inverts the paradigm—experts select their top-k tokens—thereby achieving near-uniform utilization and faster convergence [].
- Parameter-Efficient Expert Routing (introduced by He []) splits each token's hidden state into two subvectors, performs product-key maximum-inner-product search to retrieve candidate experts, and applies a batch-normalization–style logit adjustment that continuously centers each expert's gating score. This normalization (acting as an online bias) pushes down hot experts and boosts cold ones, yielding near-uniform token counts without auxiliary losses or assignment solving.

Our work adopts the PEER strategy: we combine product-key lookup with batch-norm normalization to keep million-expert routing balanced, foregoing BASE-style assignments. Although we do not explore merging Expert-Choice with PEER, the two approaches are orthogonal and could be combined in future work.

2.3 System-Level Optimizations for Sparse Experts

Moving millions of tiny experts through the memory hierarchy stresses both bandwidth and kernel efficiency.

- MegaBlocks reformulated MoE matrix multiplications as block-sparse operations, gaining up to 40% training speedup on A100 GPUs [**?**].
- Scatter-MoE and Triton-based fused kernels further eliminate padding by fusing gather-weights and GEMM into a single kernel [].
- For inference, fMoE [] and Fiddler [**?**] stream infrequently used experts from host memory or interleave expert fetch with computation to exploit locality.
- Weight-offload at training scale has been demonstrated on wafer-scale hardware, although it remains to be validated on commodity GPU clusters.

LLM720 builds on these advances. A Triton-fused kernel (inspired by MegaBlocks/Scatter-MoE) in conjunction with Unified Virtual Addressing (UVA) retains "hot" experts in VRAM while streaming "cold" ones from host memory. Because each expert is tiny, the high a priori cache-hit rate when millions of candidates are resident means the chance of an entirely missed cache is negligible. This contrasts with ProMoE [], which employs speculative prefetching; LLM720 instead overlaps expert retrieval with computation, thereby hiding latency.

2.4 Fine-Grained Scaling Laws and Memory-Aware Analysis

Empirical scaling laws for dense models (e.g., Kaplan's compute-optimal curves and Hoffmann's data-parameter trade-offs) do not account for memory cost. Recent work has begun to close this gap for sparse models:

- Fine-Grained MoE Scaling Laws show that shrinking expert size while increasing their count improves perplexity per FLOP up to some 1M experts [].

- Joint Dense–Sparse Scaling incorporates HBM budgets and predicts that MoEs become memory-optimal when memory cost dominates compute [].
- Sardana et al. [] call for bandwidth-aware scaling metrics, arguing that inference latency is an increasingly critical bottleneck.
- Mixture-of-a-Million-Experts (PEER) validated some of these predictions using a product-key router and 1M micro-experts, though without providing a public framework for replication [].

We extend this work by (i) formalizing scaling equations that incorporate cache-miss probability and streaming time and (ii) providing LLM720 as an open framework to test these predictions under controlled ablations. Our approach draws inspiration from methods such as Flash Attention, wherein reducing memory bandwidth usage is prioritized over computational complexity.

2.5 Completely Open-Sourced Model Development

Few projects have demonstrated end-to-end model development. LLM360, developed by Lieu et al. (2023), showcased a robust training pipeline using entirely open data sources and methods. LLM720 is our attempt to extend that idea by unifying state-of-the-art methodologies for model development within a single framework.

2.6 Summary

Prior research has established MoE's viability, introduced sophisticated routing schemes, and delivered key kernel and memory optimizations. Yet no public platform has systematically evaluated million-expert regimes or connected empirical results with a unifying theory. LLM720 aims to close that gap, enabling reproducible experiments that probe when sparse capacity truly outperforms dense scaling under modern memory constraints.

References

References

et al., D.-A. (2024). Deepseek-v3 technical report. *ArXiv*, *abs/2412.19437*. Retrieved from https://arxiv.org/abs/2412.19437

et al., K. (2024). Fine-grained mixture-of-experts scaling laws. *ArXiv*, *abs/2402.07871*. Retrieved from https://arxiv.org/abs/2402.07871

et al., L. (2025). Memory-efficient mixture-of-experts training: Theoretical insights. *ArXiv*, *abs/2502.05172*. Retrieved from https://arxiv.org/abs/2502.05172

et al., S. (2024). Promoe: Efficient mixture of experts with prefetching. *ArXiv*, *abs/2410.22134*. Retrieved from https://arxiv.org/abs/2410.22134

Fedus, W., et al. (2021). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *ArXiv*, *abs/2101.03961*. Retrieved from https://arxiv.org/abs/2101.03961

He, X. O. (2024). Mixture of a million experts. *ArXiv*, *abs/2407.04153*. Retrieved from https://arxiv.org/abs/2407.04153

Jacobs, R. A., & Jordan, M. I. (1991). Adaptive mixture of local experts. *Neural Computation*, *3*(1), 79–87. doi: 10.1162/neco.1991.3.1.79

Lepikhin, D., et al. (2020). Gshard: Scaling giant models with conditional computation and automatic sharding. *ArXiv*, *abs/2006.16668*. Retrieved from https://arxiv.org/abs/2006.16668

Lewis, M., et al. (2021). Base layers: Simplifying training and inference of large-scale models. *ArXiv*, *abs/2103.16716*. Retrieved from https://arxiv.org/abs/2103.16716

Riquelme, C., et al. (2021). Scaling vision with sparse mixture of experts. *ArXiv*, *abs/2106.05974*. Retrieved from https://arxiv.org/abs/2106.05974

Sardana, N., et al. (2024). Beyond chinchilla-optimal: Accounting for inference in language model scaling laws. *ArXiv*, *abs/2401.00448*. Retrieved from https://arxiv.org/abs/2401.00448

Shazeer, N., et al. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *ArXiv*, *abs/1701.06538*. Retrieved from https://arxiv.org/abs/1701.06538

Yu, H., et al. (2025). fmoe: Fine-grained expert offloading for large mixture-of-experts serving. *ArXiv*, *abs/2502.05370*. Retrieved from https://arxiv.org/abs/2502.05370

Zhou, Y., et al. (2022). Expert choice routing for scalable mixture-of-experts models. *ArXiv*, *abs/2202.09368*. Retrieved from https://arxiv.org/abs/2202.09368