Expert Calibration Lens for Pruning Mixture of Experts

Luis Frentzen Salim[1,2], Chia-Chun Wu[1], Tran Van Nhiem[3], Lun-Wei Ku[1], Yung-Hui Li[3],
[1]Institute of Information Science, Academia Sinica,
[2]National Taiwan University of Science and Technology,
[3]AI Research Center, Hon Hai Research Institute,
Correspondence: luisfrentzen@gmail.com

Abstract

Expert pruning is a practical deployment technique for Mixture-of-Experts (MoE) models. It reduces resource usage and mitigates expert redundancy, but its success depends strongly on the calibration set used for pruning. In domain-general settings, it is unclear which properties of the calibration data drive good pruning outcomes, and the effects of calibration perturbations are often unintuitive. We observe, for example, that calibration sets in different languages can lead to very similar pruning results despite appearing dissimilar on the surface. To address this, we propose *Expert Calibration Lens*, a lightweight analysis tool that compares expert activation patterns across datasets to predict the impact of calibration perturbations without repeatedly running expensive pruning procedures. We use activations that are quick to compute and evaluate the resulting analysis for downstream task performance.

1 Introduction

Mixture of Experts (MoE) architecture is a standard approach for scaling Large Language Model (LLM) capacity without a proportional increase in inference cost (Jiang et al., 2024; Riquelme et al., 2021; Dai et al., 2024). By activating only a subset of parameters per token, MoE models achieve strong performance while keeping compute budgets manageable during training and inference. However, deploying MoE language models remains costly due to their memory footprint and operational complexity of serving sparse architectures at scale. Post-training expert pruning is an appealing alternative for reducing the effective model size and overlapping experts (Chen et al., 2022; Li et al., 2025; Xie et al., 2024). A crucial component, yet often overlooked, of this process is the calibration set, a set of sequences used to determine which experts are important and therefore retained.

In practice, calibration outcomes are unintuitive: increasing the number of samples does not always improve pruning outcomes, and calibration sets in different languages can lead to similar results. These perturbation effects make pruning decisions brittle and expensive to diagnose because validating them typically requires running multiple ablations.

We propose *Expert Calibration Lens*, a system that profiles calibration data using collected activations and presents both within-dataset and cross-dataset analyses. Given a dataset A and a perturbed or alternative dataset B, the system provides univariate summaries for A and B and bivariate comparisons between them, derived from their activation statistics. This allows users to carefully estimate the effect of calibration perturbations, such as format, size, or domain, without rerunning full pruning procedures. Our system would be useful for Machine Learning practitioners, foundation model researchers, and engineers deploying MoE systems who require fast pruning iteration and calibration optimization.

Expert Calibration Lens includes a web-based interface and a model wrapper for the GPT-OSS MoE architecture. We plan to extend the wrapper to be model-agnostic with future development efforts. A public demo page is accessible at `http://4.151.237.144:8886/`[1]. *Expert Calibration Lens* is part of the **FoxBrain** project, a family of LLMs developed by the AI research center at the **Foxconn Research Institute (Hon Hai Technology Group)**. The core system is currently maintained under a proprietary license with open-source consideration for future development.

2 System Description

Our system comprises three core components: an activation collection pipeline, a rigorous analytical backend, and an interactive web interface.

[1]A short system demonstration video is accessible at `https://tinyurl.com/expertcalibdemo`

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 736–742
July 2-7, 2026 ©2026 Association for Computational Linguistics

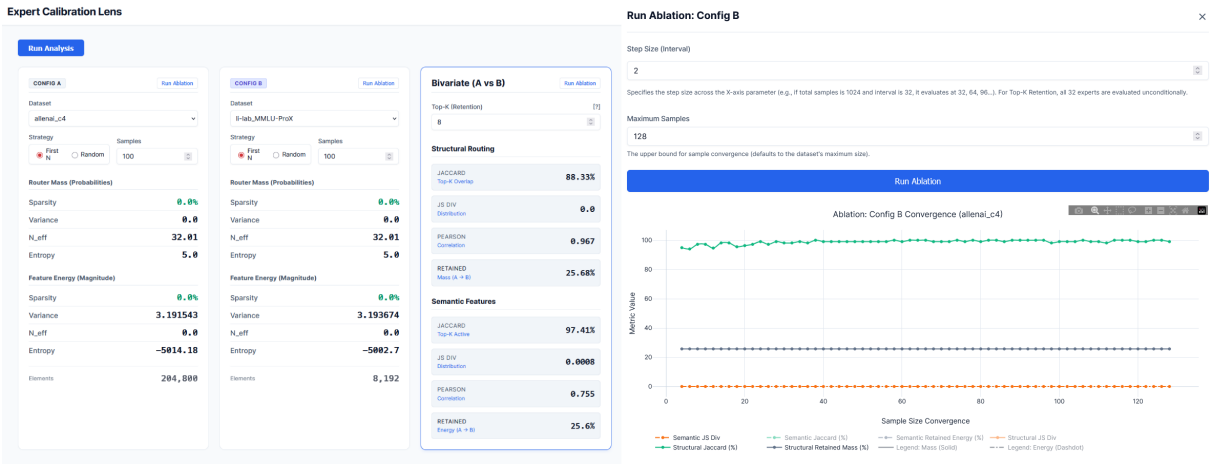

Figure 1: Main interface of Expert Calibration Lens. The interface hosts two univariate and bivariate metric cards (Left), and a generated ablation window and graph view (Right).

2.1 Activation Collection Pipeline

Our collection pipeline operates as a non-intrusive attachment to capture routing semantics without altering the computational graph.

Model Interception and Logged Signals To observe calibration behavior without repeatedly pruning the model, we attach a wrapper around each MoE MLP module. The wrapper intercepts the forward pass *at the router output* and records router distributions and optional expert contribution proxies, while leaving the computational graph and model outputs unchanged.

Dense router distribution. For sample i, MoE layer $l \in \{1, \ldots, L\}$, and token position t, let $\mathbf{g}_{i,l,t} \in \mathbb{R}^E$ denote the router logits over E experts. We compute the dense routing distribution vector $\mathbf{p}_{i,l,t} \in \Delta^{E-1}$ as:

$$\mathbf{p}_{i,l,t} = \mathrm{softmax}(\mathbf{g}_{i,l,t}). \tag{1}$$

This distribution is defined *prior* to any top-K truncation used by sparse routing. We log the individual probability $p_{i,l,t,e}$ for each expert $e \in \{1, \ldots, E\}$, which enables analysis of near-threshold experts and routing uncertainty.

Padding-aware token accounting. If an attention mask $a_{i,t} \in \{0, 1\}$ is available, we define the number of non-padding tokens

$$T_i = \sum_{t=1}^{S} a_{i,t}, \tag{2}$$

and we record only padding-aware statistics (padding positions are treated as zeros in stored tensors). This ensures dataset comparisons are token-weighted rather than biased by padding length.

Optional expert contribution proxy (feature energy). In addition to router distributions, the wrapper can optionally record a lightweight proxy for per-expert contribution after sparse routing is applied. Let $\mathcal{K}_{i,l,t}$ denote the set of top-K experts selected for token t in layer l, and let $w_{i,l,t,e}$ be the corresponding renormalized routing weight for $e \in \mathcal{K}_{i,l,t}$. Let $\mathbf{y}_{i,l,t,e} \in \mathbb{R}^H$ denote the expert output vector.

We define a nonnegative contribution proxy $\mathcal{E}_{i,l,t,e}$ as:

$$\mathcal{E}_{i,l,t,e} = \mathbf{1}[e \in \mathcal{K}_{i,l,t}]\, w_{i,l,t,e}^2 \left\|\mathbf{y}_{i,l,t,e}\right\|_2^2, \tag{3}$$

which accumulates over tokens and examples, as described below. This signal is intended as a fast heuristic for expert "importance" and does not exactly reproduce any particular pruning objective.

On-disk shard format (per layer). During calibration inference, each MoE layer writes a sequence of shard files `layerXXX_shardYYY.pt`. Each shard contains a list of batch records. A batch record stores:

- `example_ids` of shape $[B]$,
- `ntokens` of shape $[B]$ (equal to T_i for each sample),
- `routing_probs` of shape $[B, S, E]$ (stored in bfloat16), where padded positions are zeroed,
- optionally, per-sample accumulated `energy` summaries (e.g., $[B, E]$) if enabled.

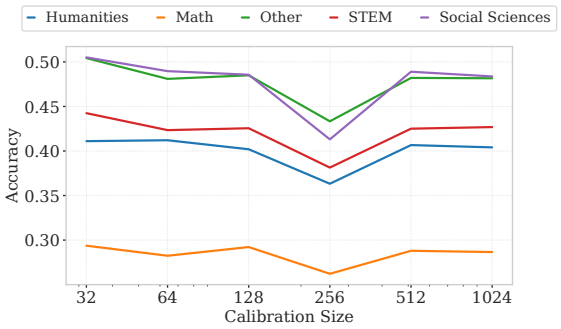

Figure 2: Evaluation performance across different calibration set sizes shows that scaling up calibration does not necessarily yield better post-prune performance. We fixed the number of retained experts to be 24.

This design avoids storing text and keeps per-token logging limited to router outputs.

Derived statistics index. Raw `routing_probs` tensors are too large to scan repeatedly for interactive analysis. Therefore, a one-time indexing step compiles the shard files into a compact dataset-level bundle `derived_stats.pt`. For each example i and layer l, we store the unnormalized per-expert probability sums

$$\text{prob\_sum}_{i,l,e} = \sum_{t=1}^{S} a_{i,t}\, p_{i,l,t,e}, \tag{4}$$

along with the token denominator $\text{token\_count}_i = T_i$. If energy is enabled, we analogously store

$$\text{energy\_sum}_{i,l,e} = \sum_{t=1}^{S} a_{i,t}\, \text{energy}_{i,l,t,e}. \tag{5}$$

From these derived statistics, the backend can compute token-weighted subset signatures (Section 2), enabling fast analysis over calibration perturbations without pruning multiple times. From these derived statistics, we compute the signatures defined below.

Token-weighted dataset signature. Given a dataset (or subset) S, we define the layerwise expert-utilization signature

$$P_l(S)[e] = \frac{\sum_{i \in S} \sum_{t=1}^{T_i} p_{i,l,t,e}}{\sum_{i \in S} T_i}. \tag{6}$$

By construction, $\sum_{e=1}^{E} P_l(S)[e] = 1$ (up to numerical error).

Figure 3: Structural Jaccard similarity of retained experts across varying calibration sequence counts. The analysis evaluates up to 128 sequences at intervals of 2. Activations converge after approximately 32–64 sequences, indicating that the resulting expert pruning mask stabilizes rapidly without requiring large amounts of calibration data.

Feature energy (expert contribution proxy). Building on our token-level energy proxy $\mathcal{E}_{i,l,t,e}$, we aggregate the token-level proxy into a dataset-level energy signature for each expert e. Let S denote the dataset and T_i the number of non-padding tokens for sample $i \in S$. We define the per-dataset energy as:

$$E_l(S)[e] = \frac{1}{\sum_{i \in S} T_i} \sum_{i \in S} \sum_{t=1}^{T_i} \mathcal{E}_{i,l,t,e}. \tag{7}$$

Note that $E_l(S)$ is a proxy for expert contribution, it does not reproduce the pruning objective exactly, but it is fast to collect and empirically predictive (Section 3).

2.2 Analytical Backend

The backend processes the precomputed index to execute queries against the routing behavior. It evaluates how a calibration dataset stimulates the MoE architecture with respect to univariate metrics and cross-dataset (bivariate) metrics.

2.2.1 Univariate Metrics

To quantify how a dataset utilizes the model's capacity, we compute the following layer-wise heuristics from the token-weighted signature $P_l(S)$ in Equation 6.

Sparsity (%): Identifies long-tail, underutilized experts. It computes the percentage of experts in

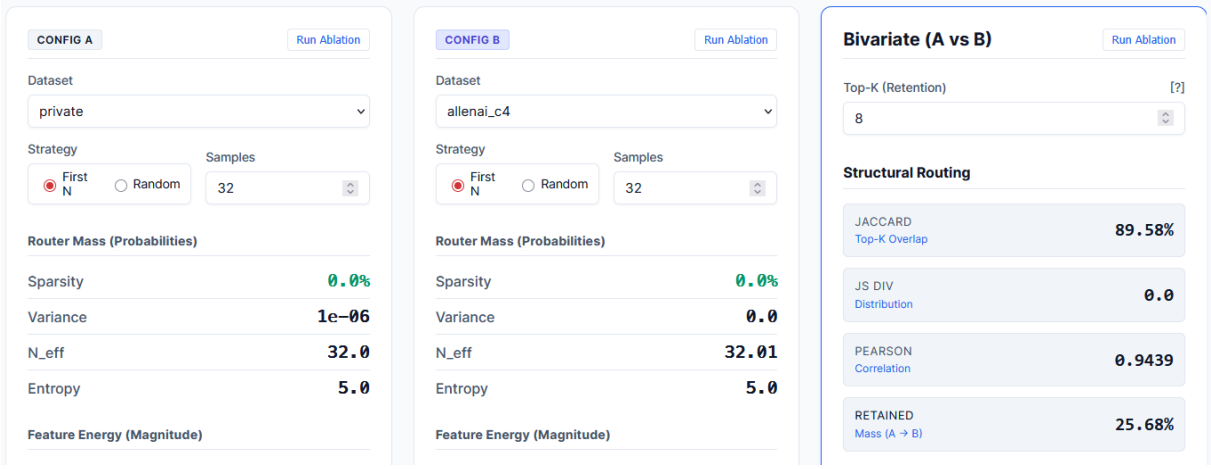

Figure 4: The proprietary FoxBrain pretraining dataset (Traditional Chinese) and the C4 dataset (English) exhibit high Jaccard similarity in their routing behavior. This structural alignment translates directly to downstream evaluations, yielding highly overlapping retained experts and comparable performance scores.

layer l that fall below a critical mass threshold (e.g., $P_l(S)[e] < 0.01$). High sparsity indicates the dataset relies on a highly constrained subset of the model's total capacity.

Variance (σ^2): Measures the dispersion of probability allocation across the E experts. Higher variance implies more disproportionate routing concentration.

Effective Experts (N_{eff}): Translates variance into an intuitive metric representing the actual number of mathematically active experts via the inverse participation ratio:

$$N_{\text{eff}} = \frac{1}{\sum_{e=1}^{E} P_l(S)[e]^2 + \epsilon} \tag{8}$$

For an architecture with $E = 32$, an N_{eff} of 3.5 implies that the layer mathematically behaves as if it had only 3.5 uniform experts activated by the current dataset.

Shannon Entropy ($\mathcal{H}$): Quantifies the "uncertainty" of the router's distribution:

$$\mathcal{H} = -\sum_{e=1}^{E} P_l(S)[e] \log_2(P_l(S)[e] + \epsilon) \tag{9}$$

Low entropy indicates highly opinionated, sharp routing paths tailored to the domain, whereas high entropy indicates a more dispersed, generalized distribution of capacity.

2.2.2 Bivariate Metrics

The core predictive capability of the *Expert Calibration Lens* is to evaluate how two datasets ($\mathcal{A}$ and $\mathcal{B}$) align dynamically. By computing structural divergence, we can predict whether tuning parameters on $\mathcal{A}$ will safely generalize to $\mathcal{B}$ during downstream expert pruning.

Proxy pruning mask. For a retention budget r (retaining r experts per layer), we define the mask:

$$\mathcal{M}_l(S; r) = \text{Top}_r(P_l(S)) \subseteq \{1, \ldots, E\}. \tag{10}$$

The retained probability mass for S equals the top-r share of $P_l(S)$, which is strictly less than 1 unless $r = E$.

Our collection of bivariate metrics and their details is as follows:

Jensen-Shannon (JS) Divergence: Measures the symmetric distributional distance between the normalized routing signatures. Let $p \equiv P_l(\mathcal{A})$ and $q \equiv P_l(\mathcal{B})$. With $M = \frac{1}{2}(p + q)$, we define JS Divergence as:

$$D_{\text{JS}}(p \parallel q) = \frac{1}{2}\big[D_{\text{KL}}(p \parallel M) + D_{\text{KL}}(q \parallel M)\big]. \tag{11}$$

A lower JS divergence implies that the two sets yield identical macroscopic routing distributions.

Jaccard Overlap (J): Operating at a given pruning retention budget r (e.g., pruning down to top-8 experts), this calculates the intersection over union of the top experts activated:

$$J_l(\mathcal{A}, \mathcal{B}; r) = \frac{|\mathcal{M}_l(\mathcal{A}; r) \cap \mathcal{M}_l(\mathcal{B}; r)|}{|\mathcal{M}_l(\mathcal{A}; r) \cup \mathcal{M}_l(\mathcal{B}; r)|}. \tag{12}$$

High Jaccard overlap implies that pruning the model with dataset $\mathcal{A}$ would retain the same physical experts as if dataset $\mathcal{B}$ had been used.

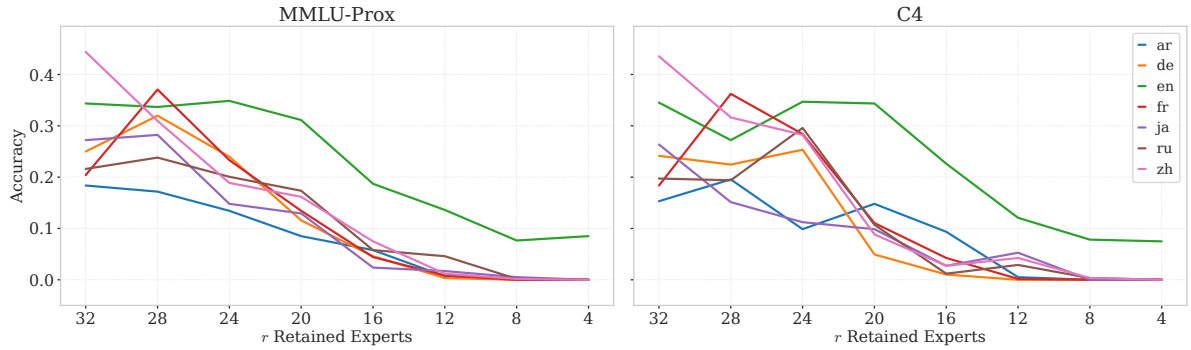

Figure 5: Evaluation performance across different languages in the MMLU ProX dataset of the pruned *gpt-oss-20b* model. We selected 7 typologically diverse languages as proxies for measuring cross-lingual calibration.

Pearson Correlation (ρ): Evaluates the linear correlation of the raw, unnormalized *Feature Energy* magnitudes. Unlike probability mass, correlation assesses whether the absolute semantic activation load scales proportionally across datasets, even when the routing probabilities align perfectly.

Retained Mass ($\mathcal{M}_{\text{ret}}$): An asymmetric metric ($\mathcal{A} \to \mathcal{B}$) and a strong heuristic for safe pruning generalization. It measures how much of dataset $\mathcal{B}$'s probability mass survives if the model is pruned using the top-r experts selected by dataset $\mathcal{A}$:

$$\mathcal{M}_{\text{ret},l}(\mathcal{A} \to \mathcal{B}; r) = \sum_{e \in \mathcal{M}_l(\mathcal{A};r)} P_l(\mathcal{B})[e]. \quad (13)$$

Pruning Implication: $\mathcal{M}_{\text{ret}}$ serves as a strict proxy for zero-shot generalization. If $\mathcal{M}_{\text{ret}}$ is high (e.g., $> 95\%$), dataset $\mathcal{A}$ behaves as a globally safe "surrogate" calibration set for dataset $\mathcal{B}$'s domain. If $\mathcal{M}_{\text{ret}}$ is low, pruning via $\mathcal{A}$ will systematically delete experts vital to processing $\mathcal{B}$, rapidly degrading task performance.

2.3 Interactive Interface and Ablation Engine

Fast Alignment: As control parameters update, the backend instantaneously queries `derived_stats.pt` to render the univariate properties alongside the bivariate cross-domain predictors, bypassing the latency of touching the actual PyTorch model weights.

Ablation Visualization: To determine the precise point where calibration sets stabilize, the interface includes an interactive Plotly.js ablation engine. It dynamically slices a given dataset into progressively finer intervals. It plots the conditional structural divergence across these steps.

Activation Collection: The interface gives the option to collect new interfaces from local or Huggingface datasets.

3 Case Study

We validate the *Expert Calibration Lens* framework by comparing its proxy metrics against empirical outcomes from a budgeted set of pruning runs. For each calibration perturbation, we compute our univariate and bivariate metrics. We then evaluate (i) the overlap of the resulting pruned expert sets and (ii) the downstream task performance of the resulting deployable models. For the demo, we provide three precomputed activation sets for the *gpt-oss-20b* model: the C4 dataset (Dodge et al., 2021), MMLU ProX (Xuan et al., 2025), and the proprietary Traditional Chinese pretraining dataset curated by the FoxBrain team. To illustrate the system's efficacy, we evaluate downstream task performance on MMLU (Hendrycks et al., 2021) and MathQA (Amini et al., 2019). All evaluations are conducted via the LM Evaluation Harness framework (Gao et al., 2023) and averaged across three iterations. Below, we detail two key observations that illustrate these relationships and demonstrate the predictive reliability of our metrics.

3.1 Calibration Set Size

Intuitively, a limited calibration dataset might fail to capture the broader pretraining distribution, theoretically leading to suboptimal pruning. However, as demonstrated in Figure 2, the impact of calibration set size on post-pruning performance rapidly plateaus, scaling beyond a small number of sequences yields no consistent gains. This data efficiency is practically advantageous. It suggests that a concise, domain-general calibration set is

sufficient to eliminate expert redundancy without overfitting to specific downstream tasks.

Crucially, the *Expert Calibration Lens* enables us to diagnose data saturation *a priori* via activation analysis, thereby bypassing computationally expensive iterative pruning. Figure 3 visualizes the step-wise similarity of top-K routing decisions as the sequence count increases. We observe that structural routing similarity converges to 100% after approximately 64 sequences. This indicates that the pruning mask stabilizes early, and incorporating additional calibration data will not alter the resulting expert selection. Consequently, we decided to limit our calibration set to 32–64 sequences for this specific case study, however, the exact convergence threshold may vary between datasets.

3.2 Cross-lingual Calibration

Figure 5 illustrates the efficacy of cross-lingual calibration, where the language used to calibrate the pruning mask differs from the evaluation language. Within the optimal sparsity regime, we find that cross-lingual calibration performs comparably to in-language. Notably, our preliminary results indicate that calibration using our proprietary Traditional Chinese data generalizes effectively to English evaluation on MMLU, yielding performance matching that of English calibration. This empirical observation motivates further exploration of cross-lingual calibration across additional languages with varying scripts and typologies.

Figure 4 corroborates these findings through a bivariate analysis of the routing activations. We compare activations derived from the English C4 dataset against our proprietary Traditional Chinese dataset spanning 28 domains. Despite the typological difference, the activation patterns reveal highly similar routing distributions. Preliminary downstream evaluations (Table 1) further confirm the performance parity between models calibrated on the C4 and proprietary datasets. Ultimately, these downstream results successfully validate the high similarity scores predicted *a priori* by the *Expert Calibration Lens*. Prior research on cross-lingual ability transfer suggests that models may learn largely language-agnostic internal representations, which could potentially explain this phenomenon (Salim et al., 2026; Bandarkar et al., 2025). However, these findings also suggest that the effects of calibration perturbation can be highly unintuitive.

Model	Calibration Set	MathQA	MMLU
baseline	N/A	0.365	0.563
pruned	MATH	**0.408**	0.585
	C4	0.3889	**0.617**
	Private	0.405	0.611

Table 1: MathQA and MMLU performance after pruning with different calibration sets. The MATH calibration is sampled from the training or validation sets of a few math evaluation datasets, which we do not explore extensively beyond preliminary work.

4 Related Systems

The *Expert Calibration Lens* complements existing MoE training and inference toolkits, such as DeepSpeed-MoE (Rajbhandari et al., 2022). While these foundational systems focus on execution, they are not designed to answer the nuanced, data-dependent calibration questions that arise during expert pruning. More broadly, our system draws inspiration from interactive model-inspection tools that visualize internal signals for model transparency. For instance, libraries such as TransformerLens (Nanda and Bloom, 2022) and the Language Interpretability Tool (LIT) (Tenney et al., 2020) facilitate the collection of deep activations, while the LM Transparency Tool (Tufanov et al., 2024) provides a visual interface for interpretability-driven decision-making. These frameworks collect activations but do not provide the pre-aggregated, macro routing metrics, specifically expert pruning calibration data. *Expert Calibration Lens* allows practitioners to anticipate pruning sensitivity and validate calibration sets *without* repeatedly executing expensive pruning ablations. Consequently, it can be utilized alongside existing pruning algorithms and MoE runtimes to streamline the deployment pipeline.

5 Limitations

For demonstration purposes, we have disabled live collection of new activation data and restricted the interactive demo to three precomputed activation sets. This constraint was implemented to ensure server stability and protect the system's storage capacity from unauthorized or excessive requests. Furthermore, the current implementation of our system is specifically tailored to the GPT-OSS model architecture. Future development efforts will focus on expanding compatibility to encompass a broader range of MoE architectures.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Lucas Bandarkar, Chenyuan Yang, Mohsen Fayyaz, Junlin Hu, and Nanyun Peng. 2025. Multilingual routing in mixture-of-experts. In *The Fourteenth International Conference on Learning Representations*.

Tianyu Chen, Shaohan Huang, Yuan Xie, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. 2022. Task-specific expert pruning for sparse mixture-of-experts. *Preprint*, arXiv:2206.00277.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *Preprint*, arXiv:2401.06066.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. *Preprint*, arXiv:2104.08758.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2023. A framework for few-shot language model evaluation.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. In *The Ninth International Conference on Learning Representations (ICLR)*. ArXiv:2009.03300.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Zichong Li, Chen Liang, Zixuan Zhang, Ilgee Hong, Young Jin Kim, Weizhu Chen, and Tuo Zhao. 2025. Slimmoe: Structured compression of large moe models via expert slimming and distillation. *Preprint*, arXiv:2506.18349.

Neel Nanda and Joseph Bloom. 2022. Transformerlens. `https://github.com/TransformerLensOrg/TransformerLens`.

Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. 2022. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale. *Preprint*, arXiv:2201.05596.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. 2021. Scaling vision with sparse mixture of experts. *Preprint*, arXiv:2106.05974.

Luis Frentzen Salim, Lun-Wei Ku, and Hsing-Kuo Kenneth Pao. 2026. Positional cognitive specialization: Where do llms learn to comprehend and speak your language? *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(39):32875–32883.

Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and Ann Yuan. 2020. The language interpretability tool: Extensible, interactive visualizations and analysis for nlp models. *Preprint*, arXiv:2008.05122.

Igor Tufanov, Karen Hambardzumyan, Javier Ferrando, and Elena Voita. 2024. LM transparency tool: Interactive tool for analyzing transformer language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 51–60, Bangkok, Thailand. Association for Computational Linguistics.

Yanyue Xie, Zhi Zhang, Ding Zhou, Cong Xie, Ziang Song, Xin Liu, Yanzhi Wang, Xue Lin, and An Xu. 2024. Moe-pruner: Pruning mixture-of-experts large language model using the hints from its router. *Preprint*, arXiv:2410.12013.

Weihao Xuan, Rui Yang, Heli Qi, Qingcheng Zeng, Yunze Xiao, Aosong Feng, Dairui Liu, Yun Xing, Junjue Wang, Fan Gao, Jinghui Lu, Yuang Jiang, Huitao Li, Xin Li, Kunyu Yu, Ruihai Dong, Shangding Gu, Yuekang Li, Xiaofei Xie, and 13 others. 2025. Mmlu-prox: A multilingual benchmark for advanced large language model evaluation. *Preprint*, arXiv:2503.10497.