

Deconstructing Pre-training: Knowledge Attribution Analysis in MoE and Dense Models

Bo Wang¹, Junzhuo Li^{1,2}, Hong Chen¹, Yuanlin Chu¹, Yuxuan Fan¹, Xuming Hu^{1,2*}

¹The Hong Kong University of Science and Technology (Guangzhou)

²The Hong Kong University of Science and Technology

{bo.wang, jz.li, hchen763, ychu763, yfan546}@connect.hkust-gz.edu.cn

xuminghu@hkust-gz.edu.cn

Abstract

Mixture-of-Experts (MoE) architectures decouple model capacity from per-token computation, enabling scaling beyond the computational limits imposed by dense scaling laws. Yet how MoE architectures shape knowledge acquisition during pre-training—and how this process differs from dense architectures—remains unknown. To address this issue, we introduce Gated-LPI (Log-Probability Increase), a neuron-level attribution metric that decomposes log-probability increase across neurons. We present a time-resolved comparison of knowledge acquisition dynamics in MoE and dense architectures, tracking checkpoints over 1.2M ($\approx 5.0T$ tokens) and 600K ($\approx 2.5T$ tokens) training steps, respectively. Our experiments uncover three patterns: (1) **Low-entropy backbone**. The top approximately 1% of MoE neurons capture over 45% of positive updates, forming a high-utility core, which is absent in the dense baseline. (2) **Early consolidation**. The MoE model locks into a stable importance profile within $< 100K$ steps, whereas the dense model remains volatile throughout training. (3) **Functional robustness**. Masking the ten most important MoE attention heads reduces relational HIT@10 by $< 10\%$, compared with $>50\%$ for the dense model, showing that sparsity fosters distributed—rather than brittle—knowledge storage. These patterns collectively demonstrate that sparsity fosters an intrinsically stable and distributed computational backbone from early in training, helping bridge the gap between sparse architectures and training-time interpretability.

Introduction

Large language models (LLMs) continue to scale in parameter count, but dense architectures make such scaling computationally inefficient, given the near-linear growth of per-token FLOPs with model size (Kaplan et al. 2020; Hoffmann et al. 2022). To mitigate this, Mixture-of-Experts (MoE) architectures leverage conditional computation, expanding total capacity while activating only a sparse subset of experts per token, thus decoupling parameter growth from per-token compute (Shazeer et al. 2017; Lepikhin et al. 2020; Abnar et al. 2025). Building on this idea, recent large-scale LLMs increasingly adopt MoE architectures (Liu et al. 2024; Comanici et al. 2025; Cai et al. 2025; Mu and Lin 2025),

achieving strong downstream performance while scaling to trillion-parameter regimes under manageable per-token compute (Yin et al. 2025; Team et al. 2025; Yang et al. 2025; Li et al. 2025a).

Despite the impressive performance and efficiency of MoE models, our understanding of their internal working mechanisms remains limited. Recent interpretability research has developed a suite of neuron attribution techniques—such as Integrated Gradients (IG) (Sundararajan, Taly, and Yan 2017; Dai et al. 2021), Layer-wise Relevance Propagation (LRP) (Chefer, Gur, and Wolf 2021; Ali et al. 2022), Log-Probability Increase (LPI) (Yu and Ananiadou 2023, 2024), and Neuron-Grad (NEG) (Zhao, Jiang, and Yoshinaga 2025). However, these studies are constrained by two key limitations. First, they have focused on traditional dense Transformers. Second, existing analyses are almost post-hoc, dissecting models only after they have been fully trained. Recent studies have begun to trace representation during learning—for instance, linear probes that chart the rise of trustworthiness dimensions across 360 checkpoints, and circuit-tracing work that follows knowledge sub-graphs through finetuning (Qian et al. 2024; Ou et al. 2025). However, these efforts again focus on dense models, leaving the question: how architectural differences influence the dynamic knowledge acquisition process.

In this paper, we present a neuron-level, time-resolved comparison of MoE and dense Transformers throughout pre-training to fill this gap. Building on LPI, we introduce a gated attribution measure that decomposes log-probability increase across experts (feed-forward network), and attention heads. We apply this method to OLMoE-1B-7B (MoE) and OLMo-7B (dense), two open-source Transformers, with 1.2M training steps ($\approx 5.0T$ tokens) and 600K steps ($\approx 2.5T$ tokens), respectively. By tracking pre-training checkpoints, we quantify and compare the evolution of importance for individual neurons in both the feed-forward network (FFN) and attention (ATTN) layers, and analyze how their importance evolves from three complementary perspectives: (i) neuron-level stability, (ii) layer-level consolidation dynamics, and (iii) functional robustness. We uncover three findings:

- **Low-entropy MoE backbone**. Top approximately 1% of MoE neurons receive $> 45\%$ of positive updates, and this set is preserved—and reinforced—throughout training,

*Corresponding author

contrasting with the churn observed in the dense model.

- **Early consolidation in MoE.** Less than 100K steps suffice for MoE FFN and ATTN layers to lock into a stable importance profile, whereas the dense model exhibits volatility.
- **Functional robustness.** Ablating the ten most important MoE attention heads reduces relational HIT@10 by < 10%, and the top 1% of MoE FFN neurons by $\approx 35\%$, whereas the dense baseline suffers $\approx 50\%$ and $\approx 96\%$ drops, respectively—indicating that sparsity fosters distributed rather than brittle knowledge storage.

By coupling scaling-friendly architectures with pre-training interpretability, our study bridges two previously disjoint threads and reveals that MoE models rapidly consolidate a stable and distributed backbone, from neurons to layers, that supports robust relational reasoning, in contrast to the dense model’s volatile and brittle organization. We hope this work will catalyze deeper dialogue between the MoE and interpretability communities and pave the way for training diagnostics that scale alongside model capacity.

Method

MoE Architecture

Mixture-of-Experts (MoE) models extend the standard Transformer architecture by introducing sparsely activated experts within certain layers. Each MoE layer contains E independent feed-forward networks, known as experts, which share the same input dimension and structure but maintain separate parameters.

During inference, a gating network selects the top- k experts for each input token based on their routing. The final output of an MoE layer is computed as a weighted sum of the selected expert outputs:

$$\text{MoE}(x) = \sum_{\mathcal{E} \in \text{Top-}k} \text{gate}_{\mathcal{E}}(x) \cdot \text{FFN}_{\mathcal{E}}(x), \quad (1)$$

where x is the residual stream input, $\text{gate}_{\mathcal{E}}(x)$ is the gating score for expert \mathcal{E} , and each expert FFN follows a two-layer structure with a nonlinearity:

$$\text{FFN}_{\mathcal{E}}(x) = W_{\mathcal{E}}^{(2)} \cdot \sigma(W_{\mathcal{E}}^{(1)} x). \quad (2)$$

While dense Transformers process all tokens through the same parameters, MoE introduces conditional computation—only a small subset of experts is activated per token. This sparsity enables parameter-efficient scaling but also alters how information and knowledge are distributed across the network.

Knowledge attribution seeks to identify which internal components of a language model contribute to specific predictions or knowledge representations. The sparsity of MoE architectures presents challenges for such analysis, as activations are no longer uniform across layers or inputs. In this work, we study how knowledge is stored and routed within MoE models, introducing a neuron-level attribution method to trace the predictive contributions of individual components over time.

Neuron Attribution in MoE Models

We adopt a neuron attribution method originally proposed for dense models (Yu and Ananiadou 2023), and extend it to MoE models (Li et al. 2025d). This method allows us to attribute the model’s prediction for a given token to individual neurons based on their effect on the output log-probability.

Expert Neuron ($v^{\mathcal{E}}$) Definition. In each selected expert, we define a **neuron** as a column vector v in the output projection matrix $W_{\mathcal{E}}^{(2)}$ (also referred to as a subvalue). Its activation is controlled by an input-dependent scalar coefficient $m = \sigma(w^{(1)} \cdot x)$, where $w^{(1)}$ is a row in $W_{\mathcal{E}}^{(1)}$ (the corresponding subkey) and x is the residual stream input.

In MoE models with Top- k routing, which is the standard approach where each token activates multiple experts (Jiang et al. 2024; Dai et al. 2024; Qwen 2024), the final output added to the residual stream is the weighted sum over all selected experts:

$$\text{output}_v = \sum_{\mathcal{E} \in \text{Top-}k} \text{gate}_{\mathcal{E}}(x) \cdot m_{\mathcal{E}} \cdot v_{\mathcal{E}}, \quad (3)$$

where each $(m_{\mathcal{E}}, v_{\mathcal{E}})$ pair corresponds to a value neuron in expert e selected for the input token. This formulation naturally extends neuron attribution to MoE models by incorporating the gated contributions of multiple expert neurons.

In other words, each FFN neuron corresponds to one column vector (subvalue) in the transformer’s output projection matrix rather than a single activation, and its strength is determined by the inner product between the input and its subkey, followed by a nonlinear transformation (Yu and Ananiadou 2023).

Attention Neuron (v^A) Definition. For attention layers, we extend the notion of neuron attribution to **attention neurons**. Each multi-head attention layer uses per-head value and output projection matrices $W_j^V, W_j^O \in \mathbb{R}^{d \times d}$ for heads $j = 1, \dots, H$, applied to the value vectors V . The attention output at position i can be written as

$$\text{Attn}(x)_i = \sum_{j=1}^H \sum_{p=1}^T \alpha_{i,j,p} \cdot W_j^O(W_j^V h_p), \quad (4)$$

where $\alpha_{i,j,p}$ is the attention weight computed via softmax over the dot product between the query at position i and the key at position p in head j .

We define the k -th column of W_j^O as the k -th attention subvalue in head j , analogous to FFN subvalues. Its corresponding attention subkey is the k -th row of W_j^V . We refer to each (subvalue, subkey) pair as an attention neuron v^A .

We attribute importance to each attention neuron by measuring the change in the log-probability of the target token when its contribution is added to the residual stream.

Neuron Importance Scoring. The importance of a neuron v is defined by the increase in the log-probability of the target token w when its output is added to the residual stream:

$$I(v) = \log p(w | x + \text{output}_v) - \log p(w | x). \quad (5)$$

This formulation allows us to rank neurons by their predictive influence, and identify high-attribution neurons that encode semantic or syntactic information.

Experiment Settings

Models

We compare two open-source Transformer models released by the same lab, ensuring consistent implementation and comparable training pipelines: **OLMo-7B** (dense) and **OLMoE-1B-7B** (MoE). Their key architectural specifications are summarized in Table 1.

OLMo-7B (Groeneveld et al. 2024) serves as our dense baseline. It is a 7B-parameter decoder-only Transformer trained for 600K steps on 2.5T tokens from the Dolma corpus (Soldaini et al. 2024), which primarily consists of diverse web text and general-domain data.

As a sparse counterpart, OLMoE-1B-7B (Muennighoff et al. 2024b) implements a Mixture-of-Experts architecture with approximately 1.3B active parameters per token. Each of its 16 layers is an MoE block containing 64 independent experts, with a Top-8 routing mechanism. The model was trained for 1.2M steps on 5.0T tokens, following the same per-step token budget (4.2M tokens) as OLMo-7B. OLMoE-1B-7B uses the OLMoE-MIX corpus (Muennighoff et al. 2024a)—a blend of DCLM (Li et al. 2024) and Dolma (Soldaini et al. 2024) with greater weight on code/math—whereas OLMo-7B is trained on Dolma ($\geq 2T$ tokens). We acknowledge this composition difference as a potential confound and control for it in our analyses where applicable.

MoE models are typically trained for more steps to ensure sufficient expert utilization (Li et al. 2025b) and to align with compute-optimal scaling principles (Du et al. 2024; Ludziejewski et al. 2025). This extended training stabilizes the routing dynamics and allows experts to converge toward specialized functional roles.

Datasets

To investigate knowledge attribution at the neuron level, we leverage a relational facts dataset. This dataset is a carefully selected subset from the comprehensive collection introduced by Hernandez et al. (Hernandez et al. 2023), which was originally designed to probe for linear relational structures in large language models. Our subset is structured to provide a balanced evaluation across different knowledge domains.

It comprises a total of 906 subject-object examples, spanning 12 distinct relations from four principal categories: *Linguistic*, *Commonsense*, *Factual*, and *Bias*. Each data point consists of a subject-object pair (e.g., *Paris, France*), a corresponding cloze-style prompt template (e.g., “The capital of { } is the city of”), and the ground-truth object token. The specific relations included are:

- **Linguistic:** adjective_antonym, word_first_letter, word_last_letter
- **Commonsense:** object_superclass, fruit_inside_color, work_location
- **Factual:** country_language, country_capital_city
- **Bias:** name_religion, occupation_age, occupation_gender, name_birthplace

This diverse selection of relations allows for a multi-faceted analysis, enabling us to examine how knowledge of varying types—from lexical patterns to world facts and learned social associations—is encoded and processed during pre-training in both MoE and dense architectures.

Metrics

We assess model performance using **HIT@10**, which evaluates the accuracy of relational knowledge prediction. Given a subject entity and a cloze-style prompt template (e.g., “The capital of { } is the city of”), the model is asked to predict the correct object entity (e.g., “Moscow”). The model generates a ranked list of candidate tokens, and HIT@10 measures the proportion of cases where the ground-truth object appears among the top-10 predictions.

$$\text{HIT@10} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}[y_i \in \text{TopK}(\hat{y}_i, 10)], \quad (6)$$

where N is the number of test examples, y_i is the ground-truth token for the i -th example, and \hat{y}_i is the vector of predicted scores (e.g., logits) over the vocabulary. The indicator function $\mathbb{I}[\cdot]$ returns 1 if the condition is true and 0 otherwise.

To better evaluate the stability of neurons and layers throughout pre-training, we define four quantitative metrics:

Top-1% Set Stability (J_{stab}). This metric quantifies the temporal consistency of the most important neurons across training checkpoints. At each step t , we identify the top-1% most important neurons (based on absolute importance score $|I|$) and compute their Jaccard overlap with the next checkpoint:

$$J_{\text{stab}} = \frac{1}{S-1} \sum_{t=1}^{S-1} \frac{|\mathcal{T}_t \cap \mathcal{T}_{t+1}|}{|\mathcal{T}_t \cup \mathcal{T}_{t+1}|}, \quad (7)$$

where \mathcal{T}_t denotes the index set of the top-1% neurons at step t , and S is the total number of checkpoints. Higher J_{stab} values indicate stronger persistence of top neurons across time, reflecting more stable activation patterns.

Positive-Gain Concentration (R_t). This metric measures how concentrated the positive importance updates are among the most reinforced neurons at each training step. Let $\Delta^+|I_{i,t}| = \max(0, |I_{i,t}| - |I_{i,t-1}|)$ denote the positive gain in importance for neuron i at step t . Then the concentration ratio is defined as:

$$R_t = \frac{\sum_{i \in \mathcal{T}_t} \Delta^+|I_{i,t}|}{\sum_{j \in \mathcal{N}} \Delta^+|I_{j,t}|}, \quad (8)$$

where \mathcal{N} represents all neurons in the layer, and \mathcal{T}_t corresponds to the top-1% neurons ranked by $\Delta^+|I_{i,t}|$. A higher R_t indicates that the learning signal is concentrated on a small subset of neurons, implying sparse but dominant functional adaptation.

Layer-Distribution Consistency (ρ_{avg}) assesses the similarity of layer-wise importance score distributions across different training steps. A higher value indicates more stable and consistent usage patterns across the model’s depth.

Model Name	Layers	FFN / Expert Dimension	Number of Experts	Routing Strategy	Attention Heads	Head Dimension
OLMo-7B	32	11008	–	–	32	128
OLMoE-1B-7B	16	1024	64	Top-8	16	128

Table 1: Model Configuration.

$$\rho_{\text{avg}} = \frac{2}{S(S-1)} \sum_{1 \leq i < j \leq S} \text{Corr}(I_i, I_j), \quad (9)$$

where S is the total number of checkpoints, and $I_i = [I_{i,1}, \dots, I_{i,L}]$ denotes the importance scores across L layers at step i . The Pearson correlation coefficient is defined as:

$$\text{Corr}(I_i, I_j) = \frac{\sum_{l=1}^L (I_{i,l} - \bar{I}_i)(I_{j,l} - \bar{I}_j)}{\sigma_i \sigma_j}, \quad (10)$$

where $\bar{I}_i = \frac{1}{L} \sum_{l=1}^L I_{i,l}$ is the average importance scores across all layers at step i , and $\sigma_i^2 = \frac{1}{L} \sum_{l=1}^L (I_{i,l} - \bar{I}_i)^2$ is the corresponding variance.

Cross-step Coefficient of Variation (σ_{rel}) measures the relative fluctuation amplitude of each layer’s importance scores across their time series, where lower values indicate higher stability.

$$\sigma_{\text{rel}} = \frac{1}{L} \sum_{l=1}^L \frac{\sigma_l}{|\mu_l|}, \quad (11)$$

where $\mu_l = \frac{1}{S} \sum_{s=1}^S I_{s,l}$ is the average importance scores across all steps for layer l , and $\sigma_l^2 = \frac{1}{S} \sum_{s=1}^S (I_{s,l} - \mu_l)^2$ is the corresponding variance.

Neuron-Level Dynamics

In this section, we probe the micro-level evolution of model knowledge by tracking FFN neurons in the MoE model ($v^{\mathcal{E}}$) and in the dense model ($v^{\mathcal{F}}$), as well as attention neurons ($v^{\mathcal{A}}$) with two complementary metrics introduced above (J_{stab} and R_t).

MoE’s Reinforced Core in FFN Neurons

Figure 1 illustrates the trajectories of $\mathcal{J}_{\text{stab},t}$ across checkpoints, while Table 2 reports the averaged stability and concentration values. These results allow us to compare how dense and MoE models differ in the consistency and focus of their most important neurons over time.

As shown in Figure 1, OLMoE exhibits markedly higher Top-1% set stability and stronger concentration of positive updates. Specifically, OLMoE’s Jaccard curve follows a three-phase *explore–refine–consolidate* trajectory: it rises to 27% at 70K steps, dips to 17.1% at 200K, and then climbs monotonically to 44.2% by 1.2M steps. This dip reflects selective refinement rather than noise, since about 13% of neurons survive across 100K–300K steps, compared to only 0.4% under random. Table 2 further confirms this trend,

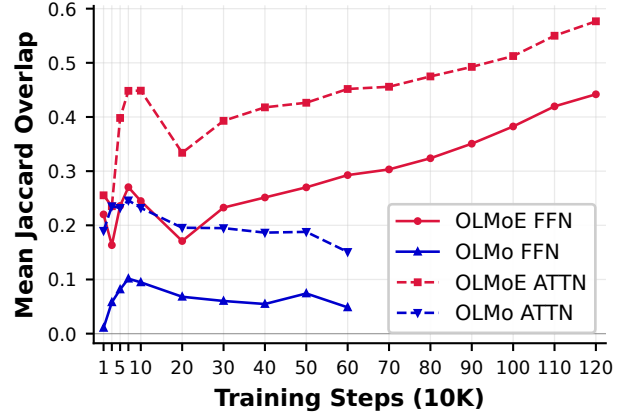


Figure 1: Top-1% FFN and ATTN neurons Jaccard overlap between consecutive checkpoints.

Model	Mean J_{stab} (%)		Mean R_t (%)
	10K–70K	100K–600K	
OLMo	6.3	6.1	26.4
OLMoE	22.2	25.3	48.9

Table 2: Neuron-level stability metrics.

showing that during consolidation the positive-gain concentration remains high ($R_t \approx 45\text{--}52\%$), meaning that nearly half of all positive updates continually reinforce the same top-1% neurons.

In contrast, OLMo never develops a durable core. Its Jaccard overlap peaks briefly at 10.2% around 70K steps, but then declines to 4.9% by 600K. Likewise, R_t is substantially lower (26.4% on average), showing that positive updates are frequently shifted to newly promoted neurons rather than consolidating a stable set.

MoE’s Sustained Stability in ATTN Neurons

To complement our findings on FFN neurons ($v^{\mathcal{E}}$ and $v^{\mathcal{F}}$), we apply the same two-metric analysis to ATTN neurons $v^{\mathcal{A}}$, aiming to assess whether neurons in attention heads exhibit similar stability during pre-training.

There is a significant difference in the stability of the core neuron sets. As shown in Figure 1, the Jaccard overlap of the two models follows distinctly different trajectories. OLMoE again follows the *explore–refine–consolidate* trajectory. Specifically, its Jaccard overlap rises rapidly to 44.9% at 100K steps after an initial exploration, dips to 33.4% at 200K steps. We further calculated the overlap between the top-1% neurons at 100K and 300K steps to be 26.73%. This

value, being lower than the consecutive overlap from 100K–200K, confirms the existence of the “refine” phase, yet it also shows that roughly a quarter of the core neurons were preserved through this adjustment and formed the foundation for the subsequent stable set. To verify this value, we also computed the overlap for a randomly selected 1% of neurons between the same steps, which averaged only 0.005. However, this adjustment paves the way for a more optimal convergence: thereafter, the overlap enters a sustained and robust upward trajectory, ultimately reaching 0.577 at 1200K steps.

Importantly, the consolidation dynamics, measured by the proportion of importance score from the top-1% neurons (R_t) relative to the total positive importance gain across the layer, perfectly complements the observed Jaccard trajectory. Specifically, this importance score ratio steadily climbs from an initial 21.5%, stabilizing in a plateau of 33%–34% after 200K steps. These trends are also clearly reflected in the metrics summarized in Table 2, which shows that OLMoE exhibits substantially higher neuron-level stability and average importance ratio than OLMo. Together, these results demonstrate that OLMoE is actively identifying and consistently reinforcing a stable core subset of v^A throughout training.

By comparison, OLMo fails to consolidate a stable set of attention neurons. Its Jaccard overlap peaks at 24.6% around 70K steps but then steadily declines to 15.1% by 600K. This downward trajectory reflects continual replacement and high churn, with important neurons frequently discarded rather than stabilized.

Both FFN and ATTN layers in OLMoE converge toward reinforced cores of neurons that remain consistently important throughout training, whereas OLMo exhibits only transient overlaps without durable consolidation.

Layer-Level Stability During Pre-training

In this section, we now ask whether this consolidation propagates upward to the layer scale. We assess this by tracking the stability of layer-wise importance profiles using two complementary metrics (ρ_{avg} and σ_{rel}).

Early Consolidation of FFN Layers in MoE

As shown in Figure 2, the average FFN importance for OLMoE rapidly increases and plateaus early in training, reaching 0.298 by 100K steps and peaking at 0.328 by 500K steps with minimal fluctuation. By contrast, OLMo’s FFN curve is unstable, dipping to 0.064 at 30K steps before climbing irregularly to 0.270 by 600K steps.

This difference in stability is further quantified in Table 3. OLMoE achieves a near-perfect ρ_{avg} of 0.97 between 10K and 600K steps, indicating that its layer-wise importance profile locks in early. OLMoE’s σ_{rel} is an order of magnitude lower than OLMo’s (**0.37 vs. 5.01**), showing that its layer-wise importance profile locks in early and drifts minimally across steps. Conversely, OLMo’s ρ_{avg} is only 0.54, reflecting weaker convergence and greater variability.

Figure 3 further illustrates this contrast: OLMoE’s FFN importance distribution is nearly identical between 50K and

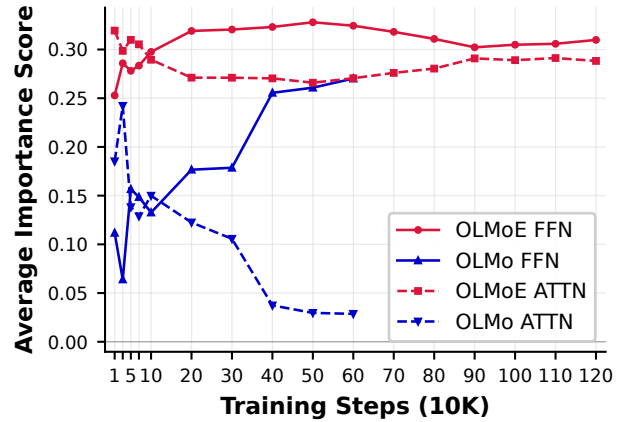


Figure 2: Mean FFN and ATTN importance scores across all layers over training steps. OLMoE shows smoother and earlier stabilization compared to OLMo.

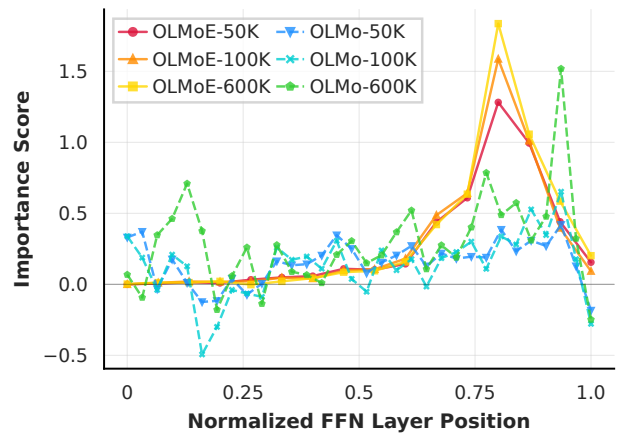


Figure 3: Snapshots of FFN layer importance distribution at different training steps.

600K steps, whereas OLMo continues to exhibit noticeable shifts across layers.

Together, these findings suggest that the MoE architecture promotes an efficient and early consolidation of FFN layer utility, a distinction from the protracted optimization process in its dense counterpart.

Sustained Stability of ATTN Layers in MoE

We next investigate whether this stabilization is unique to FFN layers or a global property of the MoE architecture by analyzing the ATTN layers. The dynamics of ATTN layers present a different but equally revealing pattern. The two models present a complementary pattern between their FFN and ATTN layer importance. As illustrated in Figure 2, OLMoE’s average ATTN importance remains high and stable throughout training, starting at 0.319 and settling into a narrow band around 0.290. In sharp contrast, OLMo’s ATTN importance undergoes a monotonic decline. This suggests

Model	FFN Metrics		ATTN Metrics	
	ρ_{avg}	σ_{rel}	ρ_{avg}	σ_{rel}
OLMo (10K–600K)	0.54	5.01	0.49	8.64
OLMoE (10K–600K)	0.97	0.37	0.97	0.49
OLMoE (10K–1.2M)	0.97	0.47	0.95	0.76

Table 3: Comparison of FFN and ATTN stability metrics. ρ_{avg} : Layer-Distribution Consistency; σ_{rel} : Cross-step Coefficient of Variation.

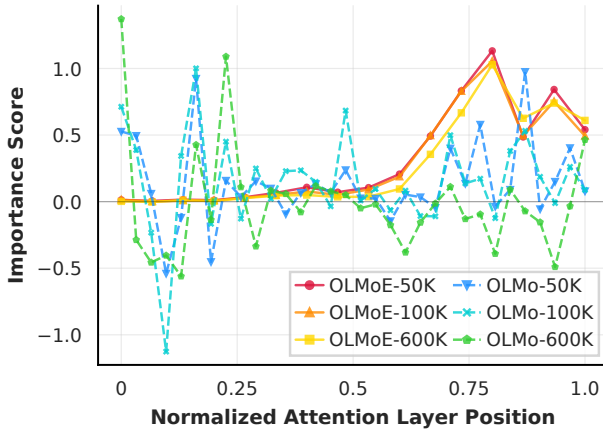


Figure 4: Snapshots of ATTN layer importance distribution at different training steps.

that as the dense model’s FFN layers consolidate their utility, its attention heads become increasingly redundant.

The stability metrics in Table 3 confirm this divergence. OLMoE achieves a high ρ_{avg} (0.97) and a low σ_{rel} of 0.49, demonstrating a stable attention layer profile that is visually supported by the stable importance distributions in Figure 4. OLMo, with a ρ_{avg} (0.49) and a σ_{rel} of 8.64, shows the opposite: its attention layer hierarchy is continuously reconfigured with high volatility throughout training, which we have already discussed from a micro perspective—the neuron level.

In summary, the layer-level stability observed in this section can be regarded as the macro-level projection of the phenomena described in the previous section, namely the sustained reinforcement of the top-1% neurons. This suggests that, during training, MoE propagates stability from a micro-level core toward an early stabilization of the layer-wise importance profile, thereby providing the structural foundation for the inference-time robustness analyzed in the following section.

Functional Robustness: Causal Intervention

Our preceding analysis showed that the MoE develops a stable computational structure from neuron level to layer level, whereas the dense remains volatile across training. This raises a functional question: does early structural stabilization translate into robustness at inference time, or merely into a brittle concentration of knowledge? If this

Model	Top-1 head	Top-10 heads	Top-1% FFN neurons
OLMoE	0.06%	9.44%	35.47%
OLMo	16.46%	50.43%	96.19%

Table 4: Performance Drop (%) from Masking.

stability reflects distributed knowledge storage rather than brittle concentration, then ablating a small, attribution-selected set of important components should incur only minor drops—especially for MoE.

To test this, we performed causal interventions by masking the most important neurons and attention heads to measure the performance drop on the HIT@10 metric, which evaluates the accuracy of relational knowledge prediction.

Ablating the Top-1% FFN neurons reveals a MoE–dense contrast. OLMoE exhibits a moderate performance reduction of 35.5% as shown in Table 4 and Figure 5, consistent with a distributed knowledge backbone. OLMo nearly collapses, with a 96.1% mean drop, and the performance approaches zero under the same intervention.

In the Top-10 heads study, OLMoE demonstrates remarkable resilience similar to the neuron ablation, causing only a minor 9.44% average performance drop. As shown in Figure 5, the performance drop is not only substantially smaller than in OLMo, but this drop progressively shrinks as training advances. This stands in stark contrast to OLMo, where the performance drop is large, volatile and averages a catastrophic 50.43%, showing no clear trend towards greater resilience. This is further clarified by the Top-1 head ablation. Ablating the single most important head in OLMo causes a performance drop (16.46%), whereas the same intervention has a negligible effect on OLMoE. This indicates that OLMoE learns resilience by distributing knowledge across components, rather than concentrating it in a few critical ones.

These findings provide evidence that a model’s architecture is a decisive factor in shaping its function. We show that OLMoE’s learning process, characterized by the stable reinforcement of a broad neuron set, is closely associated with its robust, distributed knowledge. This stable internal structure, formed early and reinforced consistently, makes the model resilient to perturbations. OLMo’s continual churn is the very mechanism that underlies its fragility: by concentrating knowledge in an ever-changing set of transient, brittle components, the dense architecture creates a vulnerable system. These results support the view that stability fosters robustness, while volatility leads to fragility.

Related Work

Mixture of Experts Mixture-of-experts (MoE) models were originally proposed as gated ensembles of specialized predictors, where a learned router softly partitions the input space and combines expert outputs (Jacobs et al. 1991; Jordan and Jacobs 1994). Modern MoE architectures expand model capacity to address computational cost via conditional computation in large neural networks (Bengio, Léonard, and Courville 2013; Cho and Bengio 2014;

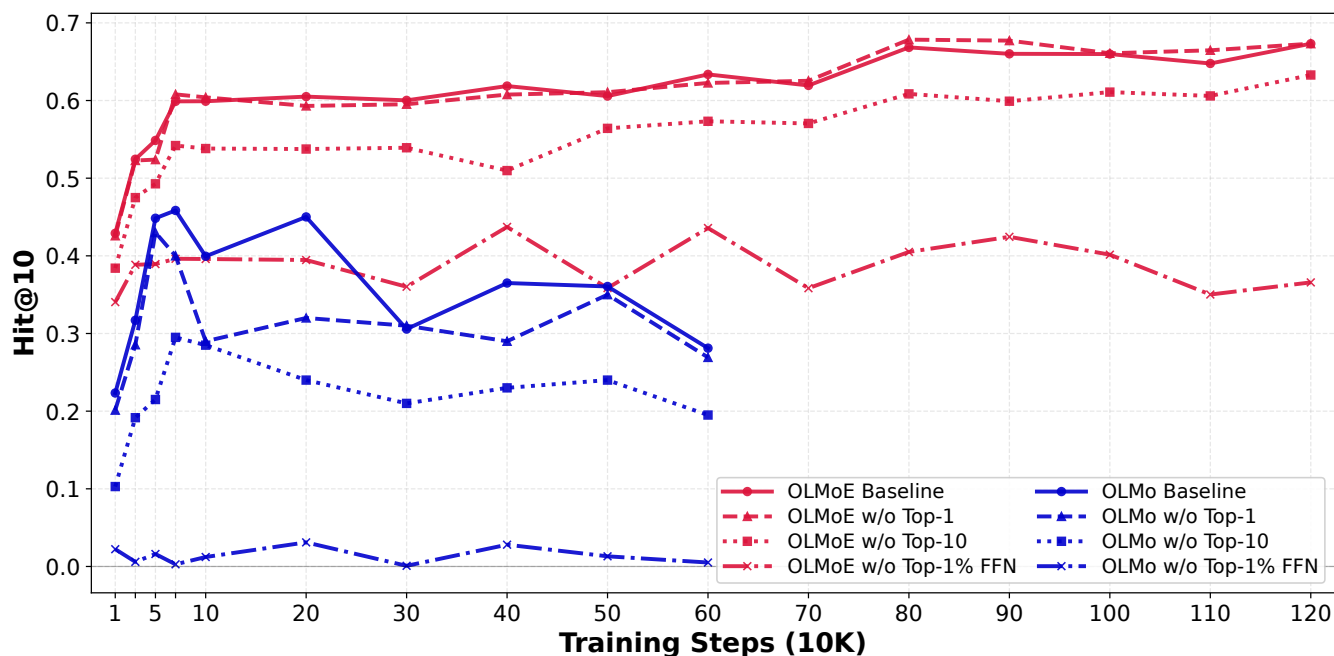


Figure 5: Performance of the model under ablation interventions (Top-1 head, Top-10 heads and Top-1% FFN neurons) measured by the HIT@10 metric.

Shazeer et al. 2017; Lepikhin et al. 2020; Fedus, Zoph, and Shazeer 2022). Lepikhin et al. (2020) introduced large-scale sparse activation via softmax Top-2 routing and an auxiliary load-balancing loss. Subsequent refinements, such as the router z -loss (Zoph et al. 2022), further stabilized optimization. Recent MoE language models typically adopt Top- k ($k > 1$) routing to improve expert utilization—e.g., Mixtral (Jiang et al. 2024), OLMoE (Muennighoff et al. 2024b), and Qwen-MoE (Qwen 2024)—and enhance specialization through shared or routed experts (Shazeer et al. 2017; Dai et al. 2024; Li et al. 2025c).

Mechanism Interpretability. Mechanism interpretability can be viewed along two complementary axes: (A) post-hoc analyses that recover internal mechanisms of trained models, and (B) training-time analyses that trace how such mechanisms emerge. For (A), neuron-level attribution methods quantify importance: Integrated Gradients (IG) (Sundararajan, Taly, and Yan 2017) and its neuron-level extensions (Shrikumar, Su, and Kundaje 2018) support applications such as “knowledge neurons” and pruning for compression (Dai et al. 2021; Yvinec et al. 2022; Ali et al. 2025); Layer-wise Relevance Propagation (LRP) is adapted to transformers with principled propagation across attention/residual paths and improved conservation/robustness (Chefer, Gur, and Wolf 2021; Ali et al. 2022); Log-Probability Increase (LPI) scores neuron importance via log-likelihood deltas under neuron insertion/removal (Yu and Ananiadou 2023), enabling analyses in multi-hop reasoning and bias and extending to MoE expert specialization in post-hoc settings (Yu, Belinkov, and Ananiadou 2025; Li et al. 2025d). Complementary to these, the circuits line de-

composes transformer into verifiable subgraphs of attention heads and MLP components (Conmy et al. 2023; Bhaskar et al. 2024). For (B), recent studies track when mechanisms form during pre-training: linear probes over hundreds of checkpoints show “trust” dimensions becoming separable early, while circuits-based analyses reveal evolving internal subgraphs that support knowledge acquisition (Qian et al. 2024; Ou et al. 2025). Most approaches focus on the trained dense model. This raises a question: how do architectural differences influence knowledge acquisition during pre-training?

Conclusion

In this paper, we conducted a comparison of knowledge acquisition dynamics in MoE and dense models across their pre-training, enabled by a gated attribution method that tracks neuron importance beyond traditional post-hoc analysis. Our findings reveal that MoE architectures form a low-entropy backbone of consistently reinforced neurons, which leads to an early consolidation of their importance profiles and, in turn, underpins their functional robustness. This resilience, stemming from more distributed knowledge storage, contrasts with the greater brittleness and knowledge concentration in the dense model. These phenomena collectively demonstrate that architectural sparsity is not merely a computational shortcut but also acts as a useful inductive bias that fosters stable and robust learning, offering insights for the design of stability-aware expert pruning and explore-then-stabilize routing strategies in future.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No.62506318); Guangdong Provincial Department of Education Project (Grant No.2024KQNCX028); CAAI-Ant Group Research Fund; Scientific Research Projects for the Higher-educational Institutions (Grant No.2024312096), Education Bureau of Guangzhou Municipality; Guangzhou-HKUST(GZ) Joint Funding Program (Grant No.2025A03J3957), Education Bureau of Guangzhou Municipality.

References

- Abnar, S.; Shah, H.; Busbridge, D.; Ali, A. M. E.; Susskind, J.; and Thilak, V. 2025. Parameters vs flops: Scaling laws for optimal sparsity for mixture-of-experts language models. *arXiv preprint arXiv:2501.12370*.
- Ali, A.; Katz, S.; Wolf, L.; and Titov, I. 2025. Detecting and Pruning Prominent but Detrimental Neurons in Large Language Models. *arXiv preprint arXiv:2507.09185*.
- Ali, A.; Schnake, T.; Eberle, O.; Montavon, G.; Müller, K.-R.; and Wolf, L. 2022. XAI for transformers: Better explanations through conservative propagation. In *International conference on machine learning*, 435–451. PMLR.
- Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.
- Bhaskar, A.; Wettig, A.; Friedman, D.; and Chen, D. 2024. Finding transformer circuits with edge pruning. *Advances in Neural Information Processing Systems*, 37: 18506–18534.
- Cai, W.; Jiang, J.; Wang, F.; Tang, J.; Kim, S.; and Huang, J. 2025. A survey on mixture of experts in large language models. *IEEE Transactions on Knowledge and Data Engineering*.
- Chefer, H.; Gur, S.; and Wolf, L. 2021. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 782–791.
- Cho, K.; and Bengio, Y. 2014. Exponentially increasing the capacity-to-computation ratio for conditional computation in deep learning. *arXiv preprint arXiv:1406.7362*.
- Comanici, G.; Bieber, E.; Schaekermann, M.; Pasupat, I.; Sachdeva, N.; Dhillon, I.; Blistein, M.; Ram, O.; Zhang, D.; Rosen, E.; et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.
- Conmy, A.; Mavor-Parker, A.; Lynch, A.; Heimersheim, S.; and Garriga-Alonso, A. 2023. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36: 16318–16352.
- Dai, D.; Deng, C.; Zhao, C.; Xu, R.; Gao, H.; Chen, D.; Li, J.; Zeng, W.; Yu, X.; Wu, Y.; et al. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*.
- Dai, D.; Dong, L.; Hao, Y.; Sui, Z.; Chang, B.; and Wei, F. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.
- Du, X.; Gunter, T.; Kong, X.; Lee, M.; Wang, Z.; Zhang, A.; Du, N.; and Pang, R. 2024. Revisiting moe and dense speed-accuracy comparisons for llm training. *arXiv preprint arXiv:2405.15052*.
- Fedus, W.; Zoph, B.; and Shazeer, N. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120): 1–39.
- Groeneveld, D.; Beltagy, I.; Walsh, P.; Bhagia, A.; Kinney, R.; Tafjord, O.; Jha, A. H.; Ivison, H.; Magnusson, I.; Wang, Y.; et al. 2024. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*.
- Hernandez, E.; Sharma, A. S.; Haklay, T.; Meng, K.; Wattenberg, M.; Andreas, J.; Belinkov, Y.; and Bau, D. 2023. Linearity of relation decoding in transformer language models. *arXiv preprint arXiv:2308.09124*.
- Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; Casas, D. d. L.; Hendricks, L. A.; Welbl, J.; Clark, A.; et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.
- Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J.; and Hinton, G. E. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1): 79–87.
- Jiang, A. Q.; Sablayrolles, A.; Roux, A.; Mensch, A.; Savary, B.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Hanna, E. B.; Bressand, F.; et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.
- Jordan, M. I.; and Jacobs, R. A. 1994. Hierarchical mixtures of experts and the EM algorithm. *Neural computation*, 6(2): 181–214.
- Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.
- Lepikhin, D.; Lee, H.; Xu, Y.; Chen, D.; Firat, O.; Huang, Y.; Krikun, M.; Shazeer, N.; and Chen, Z. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*.
- Li, A.; Liu, B.; Hu, B.; Li, B.; Zeng, B.; Ye, B.; Tang, C.; Tian, C.; Huang, C.; Zhang, C.; et al. 2025a. Every Activation Boosted: Scaling General Reasoner to 1 Trillion Open Language Foundation. *arXiv preprint arXiv:2510.22115*.
- Li, H.; Lo, K. M.; Wang, Z.; Wang, Z.; Zheng, W.; Zhou, S.; Zhang, X.; and Jiang, D. 2025b. Can Mixture-of-Experts Surpass Dense LLMs Under Strictly Equal Resources? *arXiv preprint arXiv:2506.12119*.
- Li, J.; Fang, A.; Smyrnis, G.; Ivgi, M.; Jordan, M.; Gadre, S. Y.; Bansal, H.; Guha, E.; Keh, S. S.; Arora, K.; et al. 2024. Datacomp-1m: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37: 14200–14282.
- Li, J.; Wang, B.; Zhou, X.; and Hu, X. 2025c. Dynamic Expert Specialization: Towards Catastrophic Forgetting-Free Multi-Domain MoE Adaptation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 18489–18504.

- Li, J.; Wang, B.; Zhou, X.; Jiang, P.; Liu, J.; and Hu, X. 2025d. Decoding Knowledge Attribution in Mixture-of-Experts: A Framework of Basic-Refinement Collaboration and Efficiency Analysis. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 22431–22446. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.
- Liu, A.; Feng, B.; Wang, B.; Wang, B.; Liu, B.; Zhao, C.; Deng, C.; Ruan, C.; Dai, D.; Guo, D.; et al. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*.
- Ludziejewski, J.; Pióro, M.; Krajewski, J.; Stefaniak, M.; Krutul, M.; Małańicki, J.; Cygan, M.; Sankowski, P.; Adamczewski, K.; Miłoś, P.; et al. 2025. Joint MoE Scaling Laws: Mixture of Experts Can Be Memory Efficient. *arXiv preprint arXiv:2502.05172*.
- Mu, S.; and Lin, S. 2025. A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications. *arXiv preprint arXiv:2503.07137*.
- Muennighoff, N.; Soldaini, L.; Groeneveld, D.; Lo, K.; Morrison, J.; Min, S.; Shi, W.; Walsh, P.; Tafjord, O.; Lambert, N.; Gu, Y.; Arora, S.; Bhagia, A.; Schwenk, D.; Wadden, D.; Wettig, A.; Hui, B.; Dettmers, T.; Kiela, D.; Farhadi, A.; Smith, N. A.; Koh, P. W.; Singh, A.; and Hajishirzi, H. 2024a. OLMoE: Open Mixture-of-Experts Language Models. *arXiv:2409.02060*.
- Muennighoff, N.; Soldaini, L.; Groeneveld, D.; Lo, K.; Morrison, J.; Min, S.; Shi, W.; Walsh, P.; Tafjord, O.; Lambert, N.; et al. 2024b. OlmoE: Open mixture-of-experts language models. *arXiv preprint arXiv:2409.02060*.
- Ou, Y.; Yao, Y.; Zhang, N.; Jin, H.; Sun, J.; Deng, S.; Li, Z.; and Chen, H. 2025. How Do LLMs Acquire New Knowledge? A Knowledge Circuits Perspective on Continual Pre-Training. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Findings of the Association for Computational Linguistics: ACL 2025*, 19889–19913. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-256-5.
- Qian, C.; Zhang, J.; Yao, W.; Liu, D.; Yin, Z.; Qiao, Y.; Liu, Y.; and Shao, J. 2024. Towards tracing trustworthiness dynamics: Revisiting pre-training period of large language models. *arXiv preprint arXiv:2402.19465*.
- Qwen, T. 2024. Qwen1.5-MoE: Matching 7B Model Performance with 1/3 Activated Parameters”.
- Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; and Dean, J. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.
- Shrikumar, A.; Su, J.; and Kundaje, A. 2018. Computationally efficient measures of internal neuron importance. *arXiv preprint arXiv:1807.09946*.
- Soldaini, L.; Kinney, R.; Bhagia, A.; Schwenk, D.; Atkinson, D.; Authur, R.; Bogin, B.; Chandu, K.; Dumas, J.; Elazar, Y.; Hofmann, V.; Jha, A.; Kumar, S.; Lucy, L.; Lyu, X.; Lambert, N.; Magnusson, I.; Morrison, J.; Muennighoff, N.; Naik, A.; Nam, C.; Peters, M.; Ravichander, A.; Richardson, K.; Shen, Z.; Strubell, E.; Subramani, N.; Tafjord, O.; Walsh, E.; Zettlemoyer, L.; Smith, N.; Hajishirzi, H.; Beltagy, I.; Groeneveld, D.; Dodge, J.; and Lo, K. 2024. Dolma: an Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 15725–15788. Bangkok, Thailand: Association for Computational Linguistics.
- Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, 3319–3328. PMLR.
- Team, K.; Bai, Y.; Bao, Y.; Chen, G.; Chen, J.; Chen, N.; Chen, R.; Chen, Y.; Chen, Y.; Chen, Y.; et al. 2025. Kimi K2: Open Agentic Intelligence. *arXiv preprint arXiv:2507.20534*.
- Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Yin, Y.; Huang, W.; Song, K.; Tang, Y.; Wu, X.; Guo, W.; Guo, P.; Wang, Y.; Meng, X.; Wang, Y.; et al. 2025. Pangu ultra: Pushing the limits of dense large language models on ascend npus. *arXiv preprint arXiv:2504.07866*.
- Yu, Z.; and Ananiadou, S. 2023. Neuron-level knowledge attribution in large language models. *arXiv preprint arXiv:2312.12141*.
- Yu, Z.; and Ananiadou, S. 2024. Interpreting arithmetic mechanism in large language models through comparative neuron analysis. *arXiv preprint arXiv:2409.14144*.
- Yu, Z.; Belinkov, Y.; and Ananiadou, S. 2025. Back attention: Understanding and enhancing multi-hop reasoning in large language models. *arXiv preprint arXiv:2502.10835*.
- Yvinec, E.; Dapogny, A.; Cord, M.; and Bailly, K. 2022. Singe: Sparsity via integrated gradients estimation of neuron relevance. *Advances in Neural Information Processing Systems*, 35: 35392–35403.
- Zhao, X.; Jiang, Z.; and Yoshinaga, N. 2025. Neuron Empirical Gradient: Discovering and Quantifying Neurons’ Global Linear Controllability. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 21446–21477. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.
- Zoph, B.; Bello, I.; Kumar, S.; Du, N.; Huang, Y.; Dean, J.; Shazeer, N.; and Fedus, W. 2022. ST-MoE: Designing Stable and Transferable Sparse Expert Models. *arXiv:2202.08906*.