EXPERT-AWARE ROUTER: COUPLING ROUTERS AND EXPERTS IN MIXTURE-OF-EXPERTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse Mixture-of-Experts (MoE) models scale Transformer capacity by activating only a small subset of experts for each token. However, standard MoE routers are parameterized independently from the experts they select, so routing scores do not directly reflect evolving expert representations. Existing expert-aware methods strengthen this coupling using expert-side activations, requiring activating many or all experts before routing, introducing substantial training overhead. We propose **Expert-Aware Router (EAR)**, a lightweight routing mechanism that derives each expert's routing representation directly from its gate projection matrix through a shared learnable probe vector. The resulting expert-derived routing matrix is combined with the standard learned router, allowing updates to expert parameters to directly affect routing decisions while retaining flexibility in token assignment. This coupling enables the router and experts to co-adapt during training while preserving the standard linear routing formulation. Across MoE models from 1B to 16B parameters and training budgets up to 500B tokens, EAR consistently improves downstream performance, increasing average accuracy by up to **0.76 points** over standard MoE. For the 16B model trained for 500B tokens, it reduces the maximum expert-load violation by **71.7%**, increases router–expert top-k overlap from **19.45% to 21.72%**, and substantially reduces inactive neurons. EAR also improves token efficiency, achieving a **1.12**$\times$ gain.

1 INTRODUCTION

Sparse Mixture-of-Experts (MoE) models scale Transformer capacity by activating only a subset of experts for each token (Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2022; Jiang et al., 2024a; Zhao et al., 2023). By leveraging sparse expert activation, it enables training models with substantially larger numbers of parameters under the same computational budget. Modern MoE architectures typically adopt highly sparse and fine-grained expert configurations, requiring the router to select appropriate experts while maintaining balanced token allocation across a large expert pool (Dai et al., 2024; Yang et al., 2025; Jiang et al., 2024a; Olmo et al., 2025; Abdin et al., 2024; DeepSeek-AI, 2026; GLM-5-Team et al., 2026).

Although recent work identifies geometric coupling between routers and experts induced by their shared routed-token history (Ahrac et al., 2026), standard MoE still learns the router separately from the experts. Routing scores therefore do not directly depend on expert parameters, and router–expert coupling can only emerge indirectly through training. As a result, expert selection is not directly conditioned on the expert's internal response to the input. An expert may therefore be selected even when its intermediate activations are weak, leading to ineffective use of expert capacity. We quantify this mismatch by comparing router-induced rankings with expert output norms in open-source MoE models. As shown in Figure 1 (c), substantial disagreement remains under sparse, fine-grained expert configurations (Yang et al., 2025; Dai et al., 2024; Google DeepMind, 2026; Abdin et al., 2024).

Recent studies have explored expert-aware routing to reduce this separation (Nguyen et al., 2025). AoE (Lv et al., 2025b) and RoutingFree (Liu et al., 2026) derive routing signals from activations of all experts, but require activating parts of all experts before selection, introducing substantial training overhead. The ERC loss avoids dense expert activations by regularizing router–expert alignment through an auxiliary objective, but the coupling remains indirect (Lv et al., 2025a). These approaches therefore face a trade-off between explicit router–expert coupling and efficient sparse routing.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

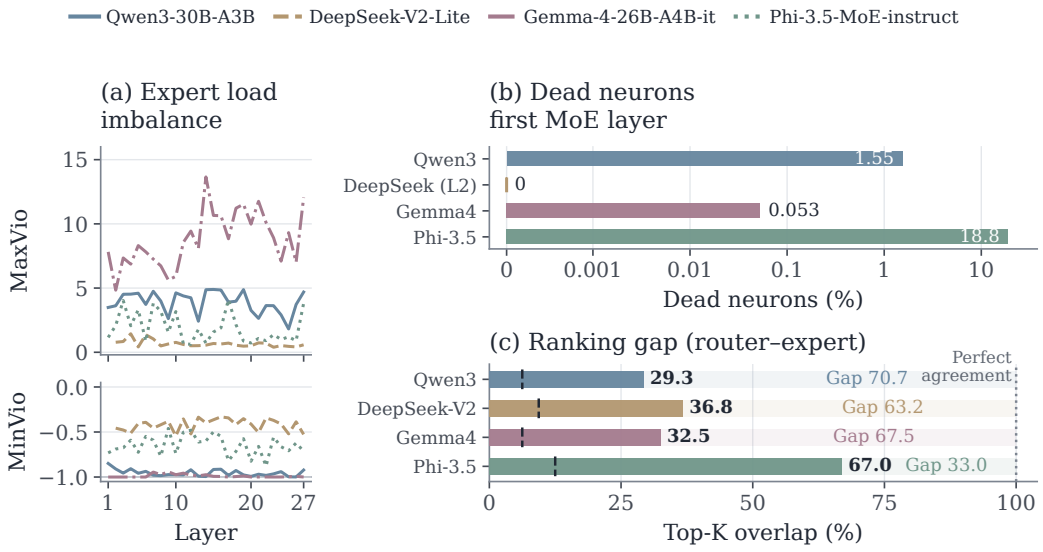


Figure 1: Expert distributions and rank alignment between routers and experts among 4 open-source MoE models

To address this tension, we propose the Expert-Aware Router (EAR), a lightweight routing mechanism that derives an expert-aware routing component directly from expert parameters. For each expert, EAR maps its SwiGLU gate projection into a routing vector using a shared learnable probe and combines the resulting expert-derived routing matrix with the standard learned router. Consequently, updates to an expert's gate projection directly modify its routing scores, allowing expert specialization and expert selection to co-adapt during training. Meanwhile, EAR avoids activating parts of all experts before selection, as required by AoE and RoutingFree, keeping training overhead low. At inference, the expert-derived component can be merged with the independent router into a single linear routing matrix, preserving the standard MoE routing form without additional computation. We evaluate EAR on MoE models ranging from 1B to 16B parameters and training budgets from 100B to 500B tokens. EAR consistently **improves average downstream performance** over standard MoE, while achieving lower validation loss and more balanced expert allocation with 500B training tokens on the 16B model. It also increases the agreement between router rankings and expert responses and improves training efficiency. Further analyses show that expert-derived routing and the independent router play complementary roles, balancing expert awareness with routing flexibility.

2 Preliminaries

Mixture-of-Experts. A standard Transformer-based MoE layer replaces the dense feed-forward network (FFN) with N expert FFNs $\{E_j\}_{j=1}^{N}$, each activated only for a selected subset of tokens. Given an input hidden state $\mathbf{x} \in \mathbb{R}^d$ with hidden dimension d, the router assigns a score to each expert and selects the top-K experts (Lepikhin et al., 2020; Fedus et al., 2022). The layer output is

$$\mathbf{y} = \sum_{j \in \mathcal{T}(\mathbf{x})} s_j \cdot E_j(\mathbf{x}), \tag{1}$$

where $\mathcal{T}(\mathbf{x})$ denotes the selected expert set and s_j is the corresponding normalized gating weight. We summarize the main notations used in Appendix A.

SwiGLU Expert. Prior work often interprets FFNs as key–value memories (Geva et al., 2021). Modern MoE models (Jiang et al., 2024b; Yang et al., 2025; Dai et al., 2024) commonly replace the

standard FFN in each expert E_j with a GLU-style gated FFN (Dauphin et al., 2017; Shazeer, 2020):

$$\mathbf{k}^j = \mathrm{Swish}\left(\mathrm{W}^{\mathrm{gate},j}\mathbf{x}\right) \odot \left(\mathrm{W}^{\mathrm{up},j}\mathbf{x}\right) \in \mathbb{R}^{d_{\mathrm{ff}}}, \quad E_j(\mathbf{x}) = \sum_{i=1}^{d_{\mathrm{ff}}} \mathbf{k}_i^j \cdot W_i^{\mathrm{down},j}, \tag{2}$$

where $\mathrm{W}^{\mathrm{gate},j}, \mathrm{W}^{\mathrm{up},j} \in \mathbb{R}^{d_{\mathrm{ff}} \times d}$, $\mathrm{W}^{\mathrm{down},j} \in \mathbb{R}^{d \times d_{\mathrm{ff}}}$ and $\mathrm{Swish}(x) = x \cdot \mathrm{sigmoid}(x)$. The same key–value memory interpretation naturally extends to SwiGLU: the relative magnitude of each element in $\mathbf{k}_j$ reflects the activation strength of the corresponding memory component.

Top-K Router. In the standard MoE router, expert scores are computed by an independent router matrix with activation functions (Yang et al., 2025):

$$\mathbf{u} = \mathrm{W}^{\text{router}}, \mathbf{x} \quad \mathcal{T}(\mathbf{x}) = \mathrm{TopK}(\mathbf{u}), \quad \mathbf{t}_j = \begin{cases} \mathbf{u}_j, & j \in \mathcal{T}(\mathbf{x}), \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where $\mathrm{W}^{\text{router}} \in \mathbb{R}^{N \times d}$. Whether expert j is selected depends solely on its router score $\mathbf{u}_j = \langle \mathrm{W}_j^{\text{router}}, \mathbf{x} \rangle$. The relationship between routers and experts (Equation 1) admits a similar key–value style interpretation to that of SwiGLU (Equation 2). This structural correspondence has led prior work to observe a high degree of similarity between the gate projection $\mathrm{W}^{\mathrm{gate},j}$ and the corresponding router vector $\mathrm{W}_j^{\mathrm{router}}$ (Lo et al., 2025).

3 Motivation

We analyze standard MoE routing from three perspectives: expert allocation, intra-expert activation, and router–expert alignment. These observations motivate coupling routing decisions more directly with expert parameters.

Imbalanced Expert Allocation. We evaluate four MoE models on a 164K-token corpus spanning mathematics, code, and multilingual text. As shown in Figure 1 (a), token allocation is highly imbalanced across experts. We quantify this imbalance using the maximum and minimum load violations (Wang et al., 2024), where values closer to zero indicate more balanced allocation. [1] Across the four models, MaxVio remains substantially above zero in many layers, while MinVio frequently approaches -1, revealing simultaneous expert overload and severe under-utilization.

Incomplete Expert Activation. Even among experts that receive tokens, not all intermediate neurons are active. As shown in Equation 2, intermediate activations $\mathbf{k}^j$ determine individual neurons' contributions to the expert output. We therefore measure the fraction of *dead neurons*, whose activations remain negligible across their expert's routed inputs.[2] Figure 1 (b) reports the fraction of dead neurons in the first MoE layer. Alhough these experts are selected by the router, some exhibit a non-negligible fraction of inactive intermediate neurons. Since the router is parameterized independently from the experts, selection does not necessarily imply a strong internal response from the selected expert. This motivates examining whether router decisions are aligned with expert responses more directly.

Router–Expert Mismatch. We therefore compare router-induced expert rankings with rankings based on expert output norms. Norm-based expert selection is motivated by dense-to-sparse approximation, where larger expert output norms indicate greater contribution under simplifying assumptions, and has been adopted by several routing methods (Su, 2025; Lv et al., 2025b). We activate all experts on diagnostic inputs and compare the top-K sets induced by router logits and expert output norms. As shown in Figure 1 (c), the overlap substantially exceeds the random expectation of K/N, but remains only 29.3%–67.0% across the four models. This observation is consistent with recent evidence that router scores can be imperfect proxies for expert utility (Li, 2026). This gap motivates constructing routing representations directly from expert parameters.

[1] Let f_i denote the fraction of tokens routed to expert i among N experts. We define $\mathrm{MaxVio} = \max_i(Nf_i - 1)$ and $\mathrm{MinVio} = \min_i(Nf_i - 1)$. Under perfectly balanced routing, both are 0.

[2] For expert j, let $\mathcal{D}_j$ denote its routed inputs. We classify neuron i as dead if $\max_{\mathbf{x} \in \mathcal{D}_j} |\mathbf{k}_i^j(\mathbf{x})| \leq 10^{-4}$. Experts receiving no tokens are excluded.

Gate-Based Coupling. These observations suggest that routing should better reflect expert-internal behavior rather than relying solely on an independent router. This raises a natural question: which expert parameters can provide a suitable representation for routing? We focus on the gate projection, which directly controls intermediate activations and has also been observed to exhibit empirical similarity to learned router vectors (Lo et al., 2025). We next provide a simple analysis explaining why the gate projection is particularly suited for constructing a linear routing signal. Consider approximating a dense sum of expert outputs using only K experts:

$$\mathbf{y}_{\text{dense}} = \sum_{j=1}^{N} E_j(\mathbf{x}), \qquad \mathbf{y}_{\text{sparse}} = \sum_{j \in \mathcal{T}} E_j(\mathbf{x}). \tag{4}$$

Under mutually orthogonal expert outputs, selecting the K largest output norms minimizes the approximation error (Su, 2025). In this simplified setting, routing should therefore preserve the ranking induced by $\|E_j(\mathbf{x})\|_2$. To examine how expert parameters relate to this ranking, consider a single-neuron SwiGLU expert:

$$E_j(\mathbf{x}) = \text{Swish}(a_j)\, b_j\, \mathbf{w}^{\text{down},j}, \qquad a_j = (\mathbf{w}^{\text{gate},j})^\top \mathbf{x}, \quad b_j = (\mathbf{w}^{\text{up},j})^\top \mathbf{x}. \tag{5}$$

Its output norm is

$$\|E_j(\mathbf{x})\|_2 = \sigma(a_j)|a_j|\,|b_j|\,\|\mathbf{w}^{\text{down},j}\|_2. \tag{6}$$

Assuming equal down-projection norms across experts, the ranking depends on $\sigma(a_j)|a_j|\,|b_j|$.

For the up projection, the factor $|b_j|$ is not monotonic in the signed pre-activation b_j. A linear score based directly on b_j therefore need not preserve its magnitude ranking. The gate factor $\sigma(a_j)|a_j|$, by contrast, increases monotonically with a_j for $a_j \geq 0$, while the sigmoid attenuates negative pre-activations. This makes the gate pre-activation a plausible linear signal for the gate-dependent component of response strength. The down projection contributes a token-independent scale, although differences in this scale can still affect rankings across experts.

This analysis motivates using gate projections to construct routing representations. It does not imply exact recovery of expert output rankings, which also depend on the other projections and interactions among neurons, but provides a rationale for the parameter coupling introduced next. We empirically compare gate, up, and down projections in Appendix 6.3, where the gate projection performs best overall.

4 METHODOLOGY

Figure 2 provides an overview of EAR. We describe its expert-aware router construction, router–expert coupling, and implementation under expert parallelism.

4.1 EXPERT-AWARE ROUTER

We propose the Expert-Aware Router (EAR), which augments the standard MoE router with an expert-derived routing component. For expert j, let $W^{\text{gate},j} \in \mathbb{R}^{d_{\text{ff}} \times d}$ denote its SwiGLU gate projection. EAR derives a routing vector directly from this expert parameter:

$$\mathbf{r}_j^{\text{EAR}} = \left(W^{\text{gate},j}\right)^\top \boldsymbol{\alpha}, \qquad \boldsymbol{\alpha} \in \mathbb{R}^{d_{\text{ff}}}, \tag{7}$$

where the learnable probe $\boldsymbol{\alpha}$ is shared across all experts. Stacking these vectors gives the expert-aware router matrix

$$R_{\text{EAR}} = \left[\mathbf{r}_1^{\text{EAR}} \quad \cdots \quad \mathbf{r}_N^{\text{EAR}}\right]^\top \in \mathbb{R}^{N \times d}. \tag{8}$$

EAR combines this matrix with the independently learned router $W_r \in \mathbb{R}^{N \times d}$:

$$W_{\text{hyb}} = W_r + R_{\text{EAR}}, \qquad \mathbf{u} = W_{\text{hyb}}\mathbf{x}. \tag{9}$$

Equivalently, the routing logit for expert j is

$$u_j(\mathbf{x}) = (\mathbf{w}_j^{\text{router}})^\top \mathbf{x} + \boldsymbol{\alpha}^\top W^{\text{gate},j}\mathbf{x}. \tag{10}$$

Because the same $\boldsymbol{\alpha}$ is shared across experts, differences among the expert-aware routing vectors arise from the expert parameters themselves. The probe learns how intermediate dimensions of the gate projection should be aggregated into a routing representation. This construction explicitly injects expert information into routing while retaining the flexibility of an independently learned router.

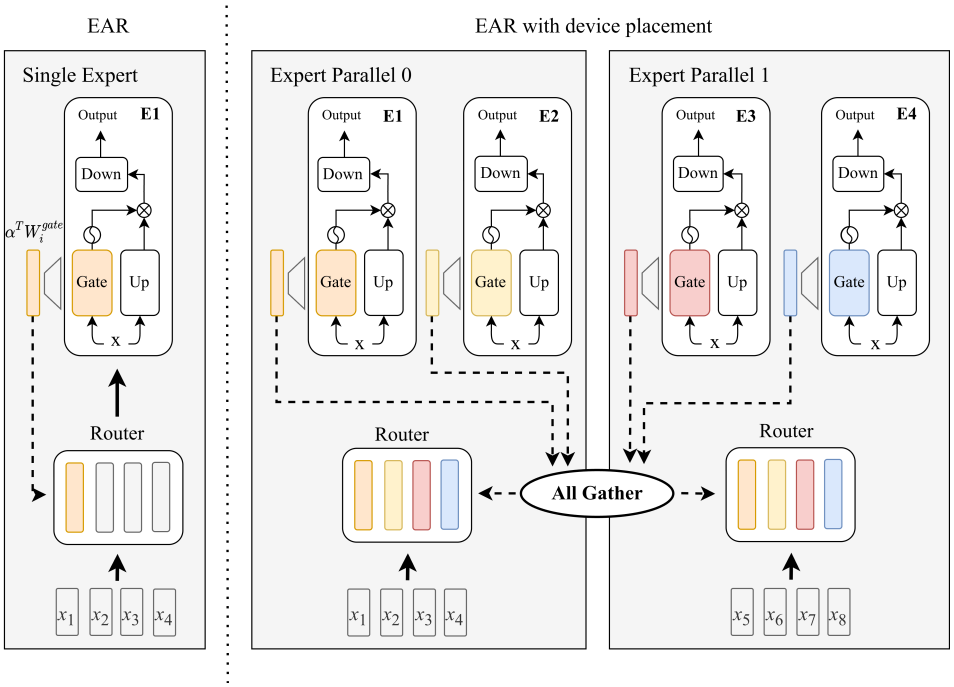


Figure 2: EAR derives routing vectors from each expert's gate projection. Under expert parallelism, each shard constructs routing vectors for its local experts, and an All-Gather operation assembles the global router matrix.

4.2 Backpropagation Through Router–Expert Coupling

We analyze how updates to an expert gate projection propagate to routing decisions. Let $u_j(\mathbf{x})$ denote the routing logit for expert j given input $\mathbf{x}$. In a standard MoE, u_j is independent of expert parameters, whereas in EAR it contains an expert-aware component derived from $W^{\text{gate},j}$. Therefore, an update $\Delta W^{\text{gate},j}$ induces

$$\Delta u_j = \begin{cases} 0, & \text{MoE}, \\ \boldsymbol{\alpha}^\top \Delta W^{\text{gate},j} \mathbf{x}, & \text{EAR}. \end{cases} \tag{11}$$

Thus, expert updates in EAR can directly modify routing scores and potentially the top-K expert set, enabling co-adaptation between expert specialization and expert selection while preserving the standard learned router.

4.3 Implementation Details

We implement EAR with expert parallelism in Megatron-LM (Shoeybi et al., 2020) (Figure 2). Each device computes the expert-aware routing vectors of its local experts using Equation 7, followed by a differentiable all-gather over the expert-parallel group to assemble the full expert-aware router matrix. After training, R_{EAR} can be materialized and merged with the standard router as $W_{\text{hyb}} = W_r + R_{\text{EAR}}$, introducing **no additional inference-time computation**. For stable initialization, we initialize $\boldsymbol{\alpha} = \frac{1}{d_{\text{ff}}}\mathbf{1}$, which keeps the initial expert-aware routing logits well scaled.

5 Experiments

5.1 Experimental Setup

We resample a 500B- and a 100B-token dataset from the OLMo3 (Olmo et al., 2026) 6T-token mixture. We primarily adopt two MoE architectures under three token-per-parameter (TPP) budgets: (1) a 1B-A0.1B model with 160 experts and top-8 routing, trained on 100B tokens (100 TPP); and (2)

Table 1: Comparison of router–expert coupling methods.

Method	Router Logits $u_{t,j}$	Aux. loss	Computation	Comm.
MoE	$(\mathbf{w}_j^{\text{router}})^\top \mathbf{x}_t$	$\mathcal{L}_{\text{bal}}$	$O(TNd)$	/
AoE	$\|\tilde{\mathbf{h}}_j(\mathbf{x}_t)\|_2$	$\mathcal{L}_{\text{bal}}$	$O(TNdr + TNr)$	$O(TN)$
RoutingFree	$\|W_j^{\text{gate}_{\text{down}}} \mathbf{x}_t\|_2$	$\mathcal{L}_{\text{rf bal}}$	$O(TNdd_{\text{ff}_{\text{low}}} + TNd_{\text{ff}_{\text{low}}})$	$O(TN)$
ERC loss	$(\mathbf{w}_j^{\text{router}})^\top \mathbf{x}_t$	$\mathcal{L}_{\text{bal}} + \mathcal{L}_{\text{ERC}}$	$O(TNd + N^2 dd_{\text{ff}})$	$O(1)$
EAR	$(\mathbf{w}_j^{\text{router}})^\top \mathbf{x}_t + \boldsymbol{\alpha}^\top W_j^{\text{gate}} \mathbf{x}_t$	$\mathcal{L}_{\text{bal}}$	$O(Ndd_{\text{ff}} + TNd)$	$O(Nd)$

a 16B-A1.6B model with 256 experts and top-8 routing, trained on 100B and 500B tokens (6.25 and 31.25 TPP, respectively). Compared with prior work on router–expert coupling, these sparse, fine-grained configurations more closely reflect the design of modern MoE models (DeepSeek-AI, 2026; DeepSeek-AI et al., 2025; Team et al., 2025; GLM-5-Team et al., 2026; Tian et al., 2026). Moreover, our maximum TPP budgets of 31.25 and 100 are substantially larger than those of mainstream models (e.g., 21 TPP for DeepSeek-V4-Pro and 22 TPP for Kimi-K2), enabling a better examination of MoE dynamics under long-term training. We tune the baseline over 8 configurations of model depth, batch size, and learning rate, selecting 16 layers, batch size 256, and learning rate 8.6×10^{-4}; see Appendix E.

5.2 Baselines

We compare EAR with three representative routing designs, summarised in Table 1. (1) **Vanilla MoE** uses an independent linear router with the standard auxiliary load-balancing loss. (2) **AoE** (Lv et al., 2025b) removes the router and ranks experts by activation norms, directly coupling routing with expert responses but requiring all experts to produce routing signals before selection. (3) **ERC** (Lv et al., 2025a) retains the standard router and introduces an auxiliary router–expert coupling objective based on $M[i,j] = |\tilde{R}[i] W_j^{\text{gate}}|_2$, encouraging each router embedding to activate its matched expert most strongly. (4) **EAR** directly constructs part of the routing representation from expert parameters while retaining an independent router. For ablation, we additionally evaluate **EAR(w/o Indep.)**, which removes the independent router and relies entirely on the expert-derived routing component. We also consider (5) **RoutingFreeMoE** (Liu et al., 2026), which determines routing from expert-side responses; however, its early-stage training requires evaluating all experts, making it prohibitively expensive under our highly sparse settings, so we exclude it from training comparisons.

5.3 Performance Across Model Scales

Table 2: Downstream evaluation across model scales under a fixed 100B-token training budget.

Benchmark	*1B-A0.1B, 100B tokens (100 TPP)*					*16B-A1.6B, 100B tokens (6.25 TPP)*			
	MoE	AoE	ERC	EAR (w/o Indep.)	EAR	MoE	ERC	EAR (w/o Indep.)	EAR
LAMBADA	41.68	26.85	40.87	<u>41.98</u>	**42.97**	<u>63.19</u>	**63.48**	62.04	62.92
HellaSwag	41.86	30.92	**42.04**	<u>41.88</u>	41.59	**46.90**	46.63	45.33	<u>46.84</u>
WinoGrande	51.22	51.38	<u>51.85</u>	50.59	**52.64**	<u>56.99</u>	55.64	54.70	**59.04**
ARC-E	<u>48.57</u>	37.75	48.53	**50.08**	48.48	<u>68.31</u>	**70.37**	65.61	67.00
ARC-C	26.62	23.72	26.37	<u>27.65</u>	**28.50**	<u>40.02</u>	38.57	39.25	**40.19**
PIQA	**69.70**	58.37	69.48	<u>69.53</u>	69.21	<u>75.19</u>	73.45	73.56	**75.63**
Avg.$_6$ ↑	46.61	38.17	46.52	<u>46.95</u>	**47.23**	<u>58.43</u>	58.02	56.75	**58.60**

Scaling Across Model Sizes We first evaluate EAR across model scales under a fixed 100B-token training budget. As shown in Table 2, EAR achieves the best Avg.$_6$ at both scales, improving MoE from 46.61 to 47.23 on the 1B model and from 58.43 to 58.60 on the 16B model.

Extended Training at 16B Scale We further evaluate EAR under long training by scaling the 16B-A1.6B model to 500B training tokens (31.25 TPP). As shown in Table 3, EAR maintains the best

Table 3: Evaluation of 16B-A1.6B models after 500B training tokens (31.25 TPP). Effective speedup measures the relative compute cost to reach the final validation loss relative to vanilla MoE.

Benchmark	MoE	EAR (w/o Indep.)	EAR
LAMBADA	57.05	59.85	**60.94**
HellaSwag	57.31	56.75	**57.40**
WinoGrande	**57.77**	56.43	57.38
ARC-E	**64.18**	61.24	61.87
ARC-C	36.95	36.69	**37.29**
PIQA	73.72	**75.30**	74.92
MMLU	36.91	36.88	**37.55**
GSM8K	**14.10**	12.28	13.80
MATH	6.78	6.36	**6.86**
C-Eval	27.94	25.31	**29.75**
CMMLU	27.76	25.20	**29.90**
BoolQ	56.24	49.82	**60.21**
OpenBookQA	**38.60**	37.60	37.60
OpenBookQA-Fact	**50.00**	48.80	**50.00**
HumanEval Pass@1	16.46	**18.29**	17.68
Avg._{15} ↑	41.45	40.45	**42.21**
MaxVio ↓	3.670	7.427	**1.038**
MinVio ↑	-1	-1	**-0.850**
MeanMaxVio ↓	0.768	1.420	**0.543**
MeanMinVio ↑	**-0.418**	-0.785	-0.426
DeadNeurons ↓	0.318%	0.096%	**0.043%**
Top-k Overlap	19.45%	29.43%	21.72%
Validation loss ↓	2.311	2.350	**2.308**
Effective speedup	1.0x	0.59x	**1.12x**

Avg._{15}, improving MoE from 41.45 to 42.21, while also achieving lower validation loss, indicating that its downstream gains are accompanied by improved language-modeling quality.

More importantly, EAR also improves the utilization-related metrics identified in Section 3. In particular, MaxVio decreases from 3.670 to 1.038, while dead neurons are reduced from 0.318% to 0.043%. Other load-balance metrics show consistent improvements, indicating more balanced and effective expert utilization. It improves routing–output alignment over the standard MoE baseline. Removing the independent router further strengthens this alignment but degrades downstream performance and load balance, indicating that alignment alone is insufficient and that the independent and expert-derived routing components are complementary.

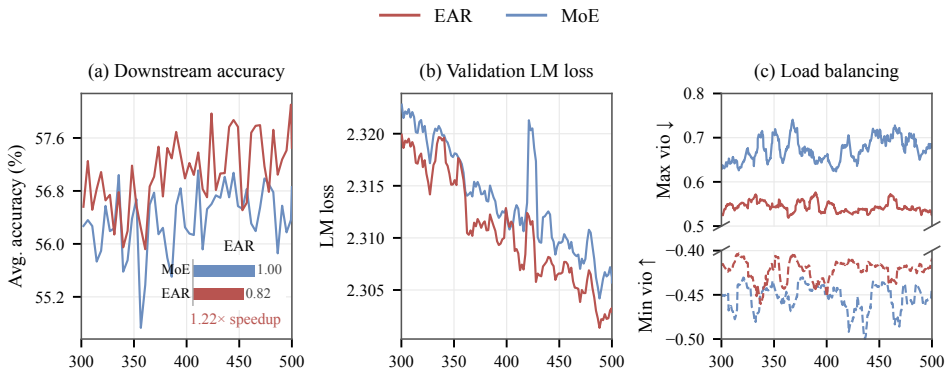


Figure 3: Training dynamics of the 16B model over 500B tokens. EAR achieves stronger downstream performance and lower validation LM loss while maintaining more balanced expert loads.

Figure 3 shows that these gains persist throughout training. Better load balance reduces bottlenecks, giving EAR a **1.22**× iteration-time speedup. EAR also improves training efficiency, achieving a **1.12**× speedup.

6 Analysis

6.1 Role of the Independent Router

We next examine the role of the independent router by comparing EAR with EAR(w/o Indep.). As shown in Table 2 and Figure 3, removing the independent router consistently degrades downstream performance and worse validation loss at both 1B and 16B scales, despite further increasing routing–output top-k overlap. This suggests that stronger router–expert alignment alone does not guarantee better routing quality.

Relying solely on expert-derived routing may be overly restrictive: the independent router provides flexibility in token assignment, while the expert-derived component introduces expert-aware inductive bias. Their combination therefore balances routing flexibility and expert awareness. To further examine these dynamics, we conduct additional 1B-scale runs in Appendix D, show that expert-derived routing is associated with fewer inactive experts early in training and larger internal activation magnitudes, while the independent router helps moderate routing-entropy fluctuations.

6.2 Gradient Flow Ablation

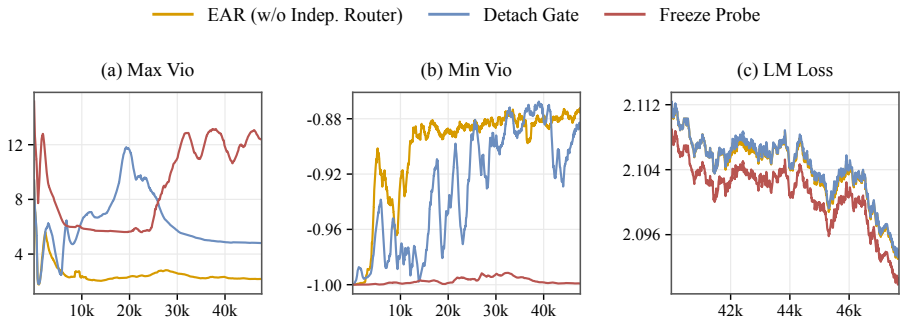


Figure 4: Gradient-flow ablation of EAR without the independent router.

We examine whether EAR benefits from jointly optimizing the shared probe and expert gate projections through the routing objective. To isolate this interaction, all variants remove the independent router and use

$$s_j = \boldsymbol{\alpha}^\top W_j^{\text{gate}} \mathbf{x}. \tag{12}$$

The full variant allows routing gradients to update both $\boldsymbol{\alpha}$ and W_j^{gate}. **Detach Gate** blocks routing gradients to W_j^{gate}, while **Freeze Probe** keeps $\boldsymbol{\alpha}$ fixed. In all cases, W_j^{gate} remains trainable through the expert forward path. Figure 4 shows that both gradient paths are important. **Detach Gate** leads to less stable load balancing and higher language-modeling loss, indicating that routing gradients to expert parameters contribute to effective coupling. **Freeze probe** causes substantially more severe expert imbalance, although it attains lower language-modeling loss. In contrast, jointly optimizing both components yields the most stable expert allocation

We hypothesize that the two gradient paths play complementary roles. Updating $\boldsymbol{\alpha}$ allows the routing criterion to adapt to evolving expert representations, while routing gradients to W_j^{gate} directly couple expert learning with expert selection. Blocking either path weakens this co-adaptation: detaching the gate makes routing less responsive to expert updates, whereas freezing the probe can amplify routing imbalance. Joint optimization therefore yields more stable router–expert coupling.

Table 4: Ablation of the projection used for expert-aware routing on the 1B-scale model. All variants remove the independent router to isolate the effect of the source projection.

Method	LAMB.	HellaS.	Wino.	ARC-E	ARC-C	PIQA	Avg._6
EAR(w/o Indep.)	**41.98**	**41.88**	50.59	**50.08**	**27.65**	**69.53**	**46.95**
EAR (w/o Indep. , Up)	<u>39.86</u>	39.52	**51.46**	46.34	25.34	67.79	45.05
EAR (w/o Indep. , Down)	39.36	<u>40.53</u>	<u>51.22</u>	<u>46.59</u>	<u>27.56</u>	<u>69.15</u>	<u>45.74</u>

6.3 Expert Projection Ablation

To isolate the effect of the expert-side projection from that of the independent router, we conduct all ablations using EAR without the independent router. We ablate the source projection used to generate routing vectors:

$$\begin{aligned} u_j^{(q)} &= \boldsymbol{\alpha}^\top A_j^{(q)} \mathbf{x}, \\ A_j^{(q)} &\in \left\{ W_j^{\text{gate}}, W_j^{\text{up}}, (W_j^{\text{down}})^\top \right\}. \end{aligned} \tag{13}$$

This directly tests the theoretical claim in Section 3: gate projections should better preserve expert-importance ordering than up or down projections. As shown in Table 4, using the gate projection consistently outperforms the up- and down-projection variants.

7 Related Work

Sparse Mixture-of-Experts. Sparse MoE layers enable large parameter counts with moderate per-token computation (Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2022). Recent surveys and system reports show that MoE has become a practical scaling strategy for modern LLMs (Zhao et al., 2023; Lo et al., 2025; Wei et al., 2024). Recent large language models further demonstrate the practical value of sparse expert architectures (Jiang et al., 2024a; Dai et al., 2024; DeepSeek-AI et al., 2025; Yang et al., 2025). EAR is orthogonal to these architectures: it changes how experts are scored, while leaving the sparse expert computation pattern intact.

Router Design and Load Balancing. Most MoE systems rely on learned top-K routers plus auxiliary load-balancing objectives (Fedus et al., 2022; Zoph et al., 2022). Other studies explore expert-choice routing (Zhou et al., 2022), auxiliary-loss-free or adaptive load balancing (Wang et al., 2024; Wen et al., 2025), token-adaptive expert counts (Zeng et al., 2024), router architecture variants (Harvey et al., 2025), and dynamically generated routers (Do et al., 2023). These methods improve routing behavior, but usually keep the router parameterization separate from expert internals.

Coupling Routers and Experts. Recent work argues that router–expert separation can lead to sub-optimal expert selection. AoE lets experts evaluate themselves through internal activation norms (Lv et al., 2025b), competition-based routing selects experts according to neural response (Pham et al., 2024; Nguyen et al., 2025), and ERC loss encourages router embeddings to match expert capabilities (Lv et al., 2025a). In contrast, our method constructs router weights directly from expert gate projections, thereby enabling explicit parameter coupling during training while retaining the standard linear routing formulation at inference.

8 Conclusion

We presented EAR, an expert-aware routing mechanism that derives routing representations from expert gate projections and combines them with a standard learned router. EAR explicitly couples routing with expert parameters while preserving the efficient linear routing form of standard MoE. Across model scales and training budgets, it improves downstream performance, router–expert alignment, load balancing, and training efficiency. Our analyses further show that expert-derived routing and the independent router are complementary, balancing expert awareness with flexibility in token assignment.

Ethics Statement

We follow the required rules for artifact use and reporting. Our use of existing artifacts, including models, tokenizers, datasets, and evaluation benchmarks, is consistent with their intended research use. For artifacts created in this work, including code and experimental outputs, the intended use is research on MoE routing, and this use is compatible with the access conditions of the original artifacts. This work does not introduce a new dataset, user-facing application, or human-subject study.

AI Use Statement

AI tools were used for manuscript polishing. All AI-assisted code generation was manually reviewed by the authors. All AI-assisted content and code were manually reviewed and verified by the authors.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

Sagi Ahrac, Noya Hochwald, and Mor Geva. Routers learn the geometry of their experts: Geometric coupling in sparse mixture-of-experts, 2026. URL https://arxiv.org/abs/2605.12476.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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)*, pp. 2924–2936, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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. DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models, January 2024. URL http://arxiv.org/abs/2401.06066.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 933–941. PMLR, 2017. URL https://proceedings.mlr.press/v70/dauphin17a.html.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence, April 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Flash/blob/main/DeepSeek_V4.pdf.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. DeepSeek-V3 Technical Report, February 2025. URL http://arxiv.org/abs/2412.19437.

Giang Do, Khiem Le, Quang Pham, TrungTin Nguyen, Thanh-Nam Doan, Bint T. Nguyen, Chenghao Liu, Savitha Ramasamy, Xiaoli Li, and Steven Hoi. HyperRouter: Towards Efficient Training and

Inference of Sparse Mixture of Experts, December 2023. URL http://arxiv.org/abs/2312.07035.

William Fedus, Barret Zoph, and Noam Shazeer. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity, June 2022. URL http://arxiv.org/abs/2101.03961.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. URL https://aclanthology.org/2021.emnlp-main.446/.

GLM-5-Team, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, Huilong Chen, Jiajie Zhang, Jian Jiao, Jiaqi Guo, Jingsen Wang, Jingzhao Du, Jinzhu Wu, Kedong Wang, Lei Li, Lin Fan, Lucen Zhong, Mingdao Liu, Mingming Zhao, Pengfan Du, Qian Dong, Rui Lu, Shuang-Li, Shulin Cao, Song Liu, Ting Jiang, Xiaodong Chen, Xiaohan Zhang, Xuancheng Huang, Xuezhen Dong, Yabo Xu, Yao Wei, Yifan An, Yilin Niu, Yitong Zhu, Yuanhao Wen, Yukuo Cen, Yushi Bai, Zhongpei Qiao, Zihan Wang, Zikang Wang, Zilin Zhu, Ziqiang Liu, Zixuan Li, Bojie Wang, Bosi Wen, Can Huang, Changpeng Cai, Chao Yu, Chen Li, Chengwei Hu, Chenhui Zhang, Dan Zhang, Daoyan Lin, Dayong Yang, Di Wang, Ding Ai, Erle Zhu, Fangzhou Yi, Feiyu Chen, Guohong Wen, Hailong Sun, Haisha Zhao, Haiyi Hu, Hanchen Zhang, Hanrui Liu, Hanyu Zhang, Hao Peng, Hao Tai, Haobo Zhang, He Liu, Hongwei Wang, Hongxi Yan, Hongyu Ge, Huan Liu, Huanpeng Chu, Jia'ni Zhao, Jiachen Wang, Jiajing Zhao, Jiamin Ren, Jiapeng Wang, Jiaxin Zhang, Jiayi Gui, Jiayue Zhao, Jijie Li, Jing An, Jing Li, Jingwei Yuan, Jinhua Du, Jinxin Liu, Junkai Zhi, Junwen Duan, Kaiyue Zhou, Kangjian Wei, Ke Wang, Keyun Luo, Laiqiang Zhang, Leigang Sha, Liang Xu, Lindong Wu, Lintao Ding, Lu Chen, Minghao Li, Nianyi Lin, Pan Ta, Qiang Zou, Rongjun Song, Ruiqi Yang, Shangqing Tu, Shangtong Yang, Shaoxiang Wu, Shengyan Zhang, Shijie Li, Shuang Li, Shuyi Fan, Wei Qin, Wei Tian, Weining Zhang, Wenbo Yu, Wenjie Liang, Xiang Kuang, Xiangmeng Cheng, Xiangyang Li, Xiaoquan Yan, Xiaowei Hu, Xiaoying Ling, Xing Fan, Xingye Xia, Xinyuan Zhang, Xinze Zhang, Xirui Pan, Xu Zou, Xunkai Zhang, Yadi Liu, Yandong Wu, Yanfu Li, Yidong Wang, Yifan Zhu, Yijun Tan, Yilin Zhou, Yiming Pan, Ying Zhang, Yinpei Su, Yipeng Geng, Yong Yan, Yonglin Tan, Yuean Bi, Yuhan Shen, Yuhao Yang, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yurong Wu, Yutao Zhang, Yuxi Duan, Yuxuan Zhang, Zezhen Liu, Zhengtao Jiang, Zhenhe Yan, Zheyu Zhang, Zhixiang Wei, Zhuo Chen, Zhuoer Feng, Zijun Yao, Ziwei Chai, Ziyuan Wang, Zuzhou Zhang, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. GLM-5: from Vibe Coding to Agentic Engineering, February 2026. URL http://arxiv.org/abs/2602.15763.

Google DeepMind. Gemma 4: Open models by Google DeepMind (gemma-4-26b-a4b-it). https://huggingface.co/google/gemma-4-26B-A4B-it, 2026. Hugging Face model repository.

Daniel Fidel Harvey, George Weale, and Berk Yilmaz. Optimizing MoE Routers: Design, Implementation, and Evaluation in Transformer Models, June 2025. URL http://arxiv.org/abs/2506.16419.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021b.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems*, 2023.

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, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of Experts, January 2024a. URL http://arxiv.org/abs/2401.04088.

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, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024b. URL https://arxiv.org/abs/2401.04088.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding, June 2020. URL http://arxiv.org/abs/2006.16668.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. Cmmlu: Measuring massive multitask language understanding in chinese, 2023.

Zongfei Li. Beyond routing: Decoupling expert dispatch and aggregation in sparse mixture-of-experts, 2026. URL https://arxiv.org/abs/2608.08853.

Yilun Liu, Jinru Han, Sikuan Yan, Volker Tresp, and Yunpu Ma. Routing-free mixture-of-experts, 2026. URL https://arxiv.org/abs/2604.00801.

Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. A closer look into mixture-of-experts in large language models, 2025. URL https://arxiv.org/abs/2406.18219.

Ang Lv, Jin Ma, Yiyuan Ma, and Siyuan Qiao. Coupling Experts and Routers in Mixture-of-Experts via an Auxiliary Loss, December 2025a. URL http://arxiv.org/abs/2512.23447.

Ang Lv, Ruobing Xie, Yining Qian, Songhao Wu, Xingwu Sun, Zhanhui Kang, Di Wang, and Rui Yan. Autonomy-of-Experts Models, May 2025b. URL http://arxiv.org/abs/2501.13074.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

Nam V. Nguyen, Huy Nguyen, Quang Pham, Van Nguyen, Savitha Ramasamy, and Nhat Ho. CompeteSMoE – Statistically Guaranteed Mixture of Experts Training via Competition, May 2025. URL http://arxiv.org/abs/2505.13380.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

Team Olmo, :, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa

Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2026. URL https://arxiv.org/abs/2512.13961.

Denis Paperno, German Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1525–1534, Berlin, Germany, 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1144. URL https://aclanthology.org/P16-1144/.

Quang Pham, Giang Do, Huy Nguyen, TrungTin Nguyen, Chenghao Liu, Mina Sartipi, Binh T. Nguyen, Savitha Ramasamy, Xiaoli Li, Steven Hoi, and Nhat Ho. CompeteSMoE – Effective Training of Sparse Mixture of Experts via Competition, February 2024. URL http://arxiv.org/abs/2402.02526.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. doi: 10.1609/aaai.v34i05.6399. URL https://ojs.aaai.org/index.php/AAAI/article/view/6399.

Noam Shazeer. Glu variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017. URL https://arxiv.org/abs/1701.06538.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism, 2020. URL https://arxiv.org/abs/1909.08053.

Jianlin Su. Moe's travelogue: 1. starting from the geometric meaning, Feb 2025. URL https://spaces.ac.cn/archives/10699.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, Yichen Feng, Kelin Fu, Bofei Gao, Hongcheng Gao, Peizhong Gao, Tong Gao, Xinran Gu, Longyu Guan, Haiqing Guo, Jianhang Guo, Hao Hu, Xiaoru Hao, Tianhong He, Weiran He, Wenyang He, Chao Hong, Yangyang Hu, Zhenxing Hu, Weixiao Huang, Zhiqi Huang, Zihao Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yongsheng Kang, Guokun Lai, Cheng Li, Fang Li, Haoyang Li, Ming Li, Wentao Li, Yanhao Li, Yiwei Li, Zhaowei Li, Zheming Li, Hongzhan Lin, Xiaohan Lin, Zongyu Lin, Chengyin Liu, Chenyu Liu, Hongzhang Liu, Jingyuan Liu, Junqi Liu, Liang Liu, Shaowei Liu, T. Y. Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yibo Liu, Yiping Liu, Yue Liu, Zhengying Liu, Enzhe Lu, Lijun Lu, Shengling Ma, Xinyu Ma, Yingwei Ma, Shaoguang Mao, Jie Mei, Xin Men, Yibo Miao, Siyuan Pan, Yebo Peng, Ruoyu Qin, Bowen Qu, Zeyu Shang, Lidong Shi, Shengyuan Shi, Feifan Song, Jianlin Su, Zhengyuan Su, Xinjie Sun, Flood Sung, Heyi Tang, Jiawen Tao, Qifeng Teng, Chensi Wang, Dinglu Wang, Feng Wang, Haiming Wang, Jianzhou Wang, Jiaxing Wang, Jinhong Wang, Shengjie Wang, Shuyi Wang, Yao Wang, Yejie Wang, Yiqin Wang, Yuxin Wang, Yuzhi Wang, Zhaoji Wang, Zhengtao Wang, Zhexu Wang, Chu Wei, Qianqian Wei, Wenhao Wu, Xingzhe Wu, Yuxin Wu, Chenjun Xiao, Xiaotong Xie, Weimin Xiong, Boyu Xu, Jing Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ziyao Xu, Junjie Yan, Yuzi Yan, Xiaofei Yang, Ying Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Xingcheng Yao, Wenjie Ye, Zhuorui Ye, Bohong Yin, Longhui Yu, Enming Yuan, Hongbang Yuan, Mengjie Yuan, Haobing Zhan, Dehao Zhang, Hao Zhang, Wanlu Zhang, Xiaobin Zhang, Yangkun Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Haotian Zhao, Yikai Zhao, Huabin Zheng, Shaojie Zheng, Jianren Zhou, Xinyu Zhou, Zaida Zhou, Zhen

Zhu, Weiyu Zhuang, and Xinxing Zu. Kimi K2: Open Agentic Intelligence, July 2025. URL http://arxiv.org/abs/2507.20534.

Changxin Tian, Kunlong Chen, Jia Liu, Ziqi Liu, Zhiqiang Zhang, and Jun Zhou. Hyperparameter scaling laws across moe sparsity, 2026. URL https://arxiv.org/abs/2609.08690.

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts, 2024. URL https://arxiv.org/abs/2408.15664.

Tianwen Wei, Bo Zhu, Liang Zhao, Cheng Cheng, Biye Li, Weiwei Lü, Peng Cheng, Jianhao Zhang, Xiaoyu Zhang, Liang Zeng, Xiaokun Wang, Yutuan Ma, Rui Hu, Shuicheng Yan, Han Fang, and Yahui Zhou. Skywork-MoE: A Deep Dive into Training Techniques for Mixture-of-Experts Language Models, June 2024. URL http://arxiv.org/abs/2406.06563.

Jialin Wen, Xiaojun Li, Junping Yao, Xinyan Kong, and Peng Cheng. Adaptive-expert-weight-based load balance scheme for dynamic routing of MoE. *Frontiers in Neurorobotics*, 19, October 2025. ISSN 1662-5218. doi: 10.3389/fnbot.2025.1590994. URL https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2025.1590994/full.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report, May 2025. URL http://arxiv.org/abs/2505.09388.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Zihao Zeng, Yibo Miao, Hongcheng Gao, Hao Zhang, and Zhijie Deng. AdaMoE: Token-Adaptive Routing with Null Experts for Mixture-of-Experts Language Models, October 2024. URL http://arxiv.org/abs/2406.13233.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2023. URL https://arxiv.org/abs/2303.18223.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. Mixture-of-Experts with Expert Choice Routing, October 2022. URL http://arxiv.org/abs/2202.09368.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. ST-MoE: Designing Stable and Transferable Sparse Expert Models, April 2022. URL http://arxiv.org/abs/2202.08906.

A NOTATION

Unless otherwise specified, subscripts usually denote tensor indices (e.g., vector elements or matrix rows), while superscripts are used as labels or identifiers (e.g., expert index j or projection type).

Notation	Description
d	Transformer hidden dimension
d_{ff}	FFN intermediate dimension
N	Number of experts
$\mathbf{x} \in \mathbb{R}^d$	Input hidden state
E_j	Expert j
$\mathcal{T}(\mathbf{x})$	Top-K selected expert set
s_j	Gating weight of expert j
$\mathbf{u}$	Router score vector
$\mathbf{u}_j$	Router score of expert j
$\mathrm{W}^{\text{router}}$	Router projection matrix
$\mathrm{W}^{\text{gate},j}$	Gate projection matrix of expert j
$\mathrm{W}^{\text{up},j}$	Up projection matrix of expert j
$\mathrm{W}^{\text{down},j}$	Down projection matrix of expert j
$\mathbf{k}^j$	Intermediate SwiGLU activation of expert j
$\mathbf{k}_i^j$	i-th intermediate activation in expert j
$W_i^{\text{down},j}$	i-th row of $\mathrm{W}^{\text{down},j}$
$\odot$	Element-wise multiplication
$\langle \cdot, \cdot \rangle$	Inner product
$\lvert \cdot \rvert$	Absolute value of a scalar
$\lVert \cdot \rVert$	Norm of a vector or matrix

Table 5: Main notations used in this paper.

We track the load balancing metrics within a global batch during training using `max_vio` and `min_vio`:

$$\begin{aligned} \max_{\text{v}}\text{io} &= \max_i \left(N f_i - 1\right), \\ \min_{\text{v}}\text{io} &= \min_i \left(N f_i - 1\right). \end{aligned} \tag{14}$$

Notably, $\min_\text{vio} = -1$ indicates that there exists at least one dead expert that receives no routed tokens within the batch.

B RANK ALIGNMENT DETAILS

This appendix describes the router–expert rank-alignment diagnostic. The main figures report top-K overlap, which measures whether the router selects the same expert set as the output-norm reference ranking. We also define correlation and NDCG variants for completeness.

Evaluation inputs. We construct a small diagnostic token set containing heterogeneous text patterns, including English sentences, Chinese sentences, mathematical expressions, and code snippets. This set is not used for training. For each model, we run a forward pass in evaluation mode and record the hidden state $\mathbf{x}_{\ell,t}$ entering every MoE layer ℓ for every diagnostic token t.

During this diagnostic forward pass, we additionally compute *full expert activations*: for each token and each MoE layer, all experts are evaluated under `no_grad` only for measurement. This does not change the actual sparse forward computation or the reported model outputs. For a layer with N experts, we record two vectors:

$$\mathbf{u}_{\ell,t} = \text{Router}_\ell(\mathbf{x}_{\ell,t}) \in \mathbb{R}^N, \tag{15}$$

and

$$\mathbf{n}_{\ell,t} = \left[\lVert E_{\ell,j}(\mathbf{x}_{\ell,t})\rVert_2\right]_{j=1}^N. \tag{16}$$

Here, $\mathbf{u}_{\ell,t}$ denotes router logits and $\mathbf{n}_{\ell,t}$ denotes expert output norms. For standard MoE, $\mathbf{u}_{\ell,t} = W_{r,\ell}\mathbf{x}_{\ell,t}$. For EAR(w/o Indep.), $\mathbf{u}_{\ell,t,j} = \boldsymbol{\alpha}_\ell^\top W_{\ell,j}^{\text{gate}} \mathbf{x}_{\ell,t}$. For EAR, $\mathbf{u}_{\ell,t} = (W_{r,\ell} + R_{\text{EAR},\ell})\mathbf{x}_{\ell,t}$.

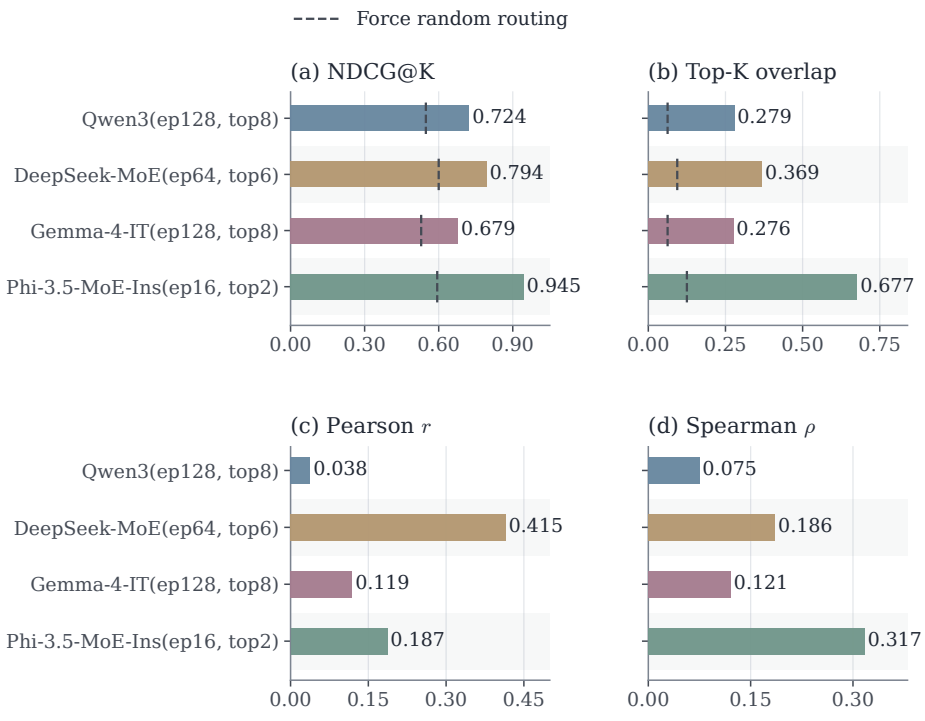


Figure 5: Full router–expert alignment diagnostics for open-source MoE models.

Pearson correlation. Pearson correlation measures linear association between router logits and expert output norms.

$$\text{Pearson}_{\ell,t} = \frac{\tilde{\mathbf{u}}_{\ell,t}^{\top}\tilde{\mathbf{n}}_{\ell,t}}{\|\tilde{\mathbf{u}}_{\ell,t}\|_2\|\tilde{\mathbf{n}}_{\ell,t}\|_2 + \varepsilon}, \tag{17}$$

where $\tilde{\mathbf{u}}_{\ell,t} = \mathbf{u}_{\ell,t} - \text{mean}(\mathbf{u}_{\ell,t})\mathbf{1}$

and $\tilde{\mathbf{n}}_{\ell,t} = \mathbf{n}_{\ell,t} - \text{mean}(\mathbf{n}_{\ell,t})\mathbf{1}$.

Spearman correlation. Spearman correlation measures rank-level agreement. Let $\text{rank}(\mathbf{u}_{\ell,t})$ and $\text{rank}(\mathbf{n}_{\ell,t})$ be the expert ranks induced by router logits and expert output norms, respectively. We compute

$$\text{Spearman}_{\ell,t} = \text{Pearson}\left(\text{rank}(\mathbf{u}_{\ell,t}), \text{rank}(\mathbf{n}_{\ell,t})\right). \tag{18}$$

This metric is more relevant than Pearson for routing, because top-K expert selection depends on ordering rather than score magnitude.

Top-K overlap. Top-K overlap measures whether the router selects the same expert set as the norm-induced oracle. Let

$$\mathcal{T}_{\ell,t}^{s} = \text{TopK}(\mathbf{u}_{\ell,t}), \qquad \mathcal{T}_{\ell,t}^{n} = \text{TopK}(\mathbf{n}_{\ell,t}). \tag{19}$$

The overlap score is

$$\text{Overlap@}\,K_{\ell,t} = \frac{|\mathcal{T}_{\ell,t}^{s} \cap \mathcal{T}_{\ell,t}^{n}|}{K}. \tag{20}$$

A random router has expected overlap K/N, which is used as the dashed random baseline in Figure 1.

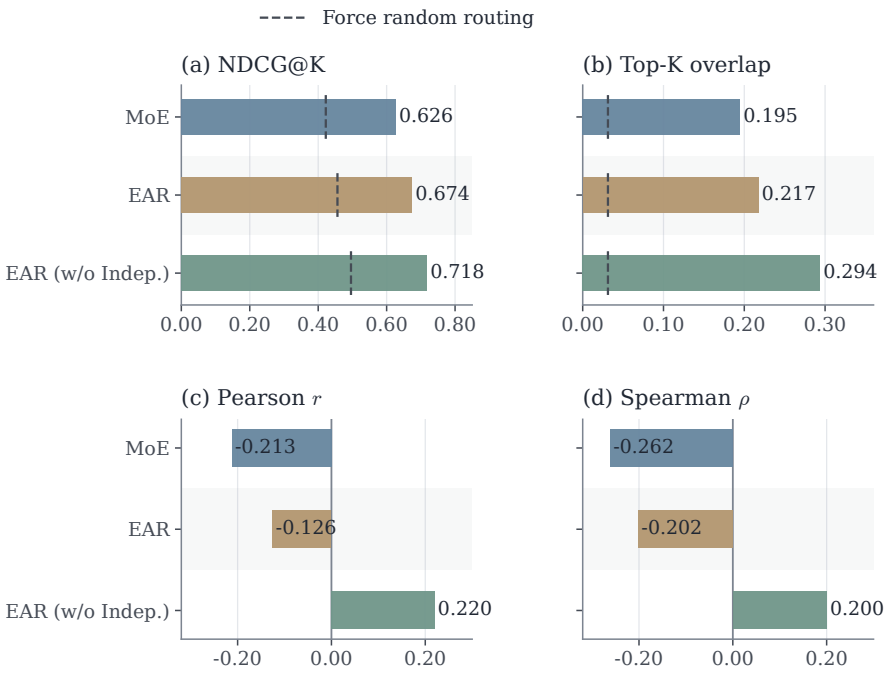


Figure 6: Full router–expert alignment diagnostics for the controlled 16B-A1.6B models trained on 500B tokens.

NDCG@K. NDCG@K evaluates whether experts with large output norms are placed near the top of the router ranking. Let $\pi^s_{\ell,t}$ be the expert order obtained by sorting $\mathbf{u}_{\ell,t}$ in descending order, and let $\pi^n_{\ell,t}$ be the order obtained by sorting $\mathbf{n}_{\ell,t}$ in descending order. We define

$$\mathrm{DCG@}\,K_{\ell,t} = \sum_{r=1}^{K} \frac{n_{\ell,t,\pi^s_{\ell,t}(r)}}{\log_2(r+1)}, \tag{21}$$

and

$$\mathrm{IDCG@}\,K_{\ell,t} = \sum_{r=1}^{K} \frac{n_{\ell,t,\pi^n_{\ell,t}(r)}}{\log_2(r+1)}. \tag{22}$$

The normalized score is

$$\mathrm{NDCG@}\,K_{\ell,t} = \frac{\mathrm{DCG@}\,K_{\ell,t}}{\mathrm{IDCG@}\,K_{\ell,t} + \varepsilon}. \tag{23}$$

Unlike top-K overlap, NDCG@K gives larger credit when high-norm experts appear earlier in the router ranking.

Aggregation. All metrics are first computed for each token at each MoE layer. We then average over tokens within each layer:

$$m_\ell = \frac{1}{|\mathcal{D}|} \sum_{t \in \mathcal{D}} m_{\ell,t}, \tag{24}$$

where m denotes one of Pearson, Spearman, Overlap@K, or NDCG@K. The final model-level metric is the mean over MoE layers:

$$m = \frac{1}{L} \sum_{\ell=1}^{L} m_\ell. \tag{25}$$

Thus, the reported number is a layer-balanced average rather than a raw average over all token–layer pairs. This avoids over-weighting models or layers that produce more valid diagnostic tokens after tokenization.

C Orthogonal Loss Ablation

This appendix reports an additional ablation on orthogonal regularization. The motivation is that load balancing encourages experts to receive similar numbers of tokens, but it does not directly require different experts to learn distinct routing directions. Orthogonal regularization instead penalizes similarity among expert routing vectors.

Orthogonal loss. For each MoE layer ℓ, let $R_\ell \in \mathbb{R}^{N\times d}$ denote the routing matrix whose j-th row corresponds to expert j. The definition of R_ℓ depends on the routing method:

$$R_\ell = \begin{cases} W_{r,\ell}, & \text{MoE}, \\ R_\ell^{\text{EAR}}, & \text{EAR(w/o Indep.)}, \\ W_{r,\ell} + R_\ell^{\text{EAR}}, & \text{EAR}, \end{cases} \tag{26}$$

where the expert-aware matrix is

$$R_\ell^{\text{EAR}} = \begin{bmatrix} ((W_{\ell,1}^{\text{gate}})^\top \boldsymbol{\alpha}_\ell)^\top \\ \vdots \\ ((W_{\ell,N}^{\text{gate}})^\top \boldsymbol{\alpha}_\ell)^\top \end{bmatrix}. \tag{27}$$

We first normalize each row:

$$\hat{\mathbf{r}}_i = \frac{\mathbf{r}_i}{\|\mathbf{r}_i\|_2 + \varepsilon}, \qquad \hat{\mathbf{R}} = [\hat{\mathbf{r}}_1; \ldots; \hat{\mathbf{r}}_N]. \tag{28}$$

We then compute the Gram matrix

$$\mathbf{G} = \hat{\mathbf{R}}\hat{\mathbf{R}}^\top, \tag{29}$$

and penalize its deviation from the identity matrix:

The orthogonal loss is

$$\mathcal{L}_{\text{orth}} = \frac{1}{L}\sum_{\ell=1}^{L} \frac{\left\| \tilde{R}_\ell \tilde{R}_\ell^\top - I_N \right\|_{\text{F}}^2}{N^2}. \tag{30}$$

Since each row of $\tilde{R}_\ell$ is normalized, the diagonal of $\tilde{R}_\ell \tilde{R}_\ell^\top$ is close to one. Therefore, this loss mainly penalizes off-diagonal cosine similarity between different expert routing directions.

The final training objective is

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda_{\text{aux}}\mathcal{L}_{\text{aux}} + \lambda_{\text{orth}}\mathcal{L}_{\text{orth}}. \tag{31}$$

We apply the same $\lambda_{\text{orth}} = 0.005$ across MoE, EAR(w/o Indep.), and EAR for a controlled comparison.

Why it differs across methods. For standard MoE, $\mathcal{L}_{\text{orth}}$ only regularizes the free router matrix W_r. It can diversify router embeddings, but it does not directly constrain expert parameters. For EAR, the loss is applied to the combined matrix, encouraging the independent router component and the expert-derived component to form diverse final routing directions.

Results Table 6 reports the final-step metrics for the 1B-scale runs. We compare each routing method with and without orthogonal regularization using the same training setup.

Orthogonal regularization has method-dependent effects. For standard MoE, it leaves the language-modeling loss nearly unchanged and slightly improves under-utilization, but increases the maximum overload. For EAR(w/o Indep.), orthogonal regularization reduces $\max_{\text{v}}\text{io}$ from 2.078 to 1.503, suggesting that encouraging diversity among expert-derived routing directions can alleviate severe overload. However, this improvement comes with worse language-modeling loss and slightly stronger under-utilization. In contrast, EAR shows a more favorable overall effect: adding the orthogonal loss improves LM loss from 2.1627 to 2.1609 and improves $\min_{\text{v}}\text{io}$ from -0.617 to -0.504, while only moderately increasing $\max_{\text{v}}\text{io}$. These results suggest that the independent router provides additional flexibility for satisfying routing-level diversity constraints, allowing orthogonal regularization to diversify the final routing directions without placing the full constraint directly on the expert-derived component.

Table 6: Orthogonal-loss ablation at the final logged training step (step 47682). We report language-modeling loss and load-balancing violations. For min_vio, values closer to zero indicate fewer severely under-used experts.

Method	LM loss ↓	max_vio ↓	min_vio ↑
MoE	2.1628	0.733	-0.726
MoE + Orth.	2.1636	0.881	-0.715
EAR(w/o Indep.)	2.1662	2.078	-0.854
EAR(w/o Indep.) + Orth.	2.1775	1.503	-0.875
EAR	2.1627	0.848	-0.617
EAR + Orth.	**2.1609**	1.013	**-0.504**

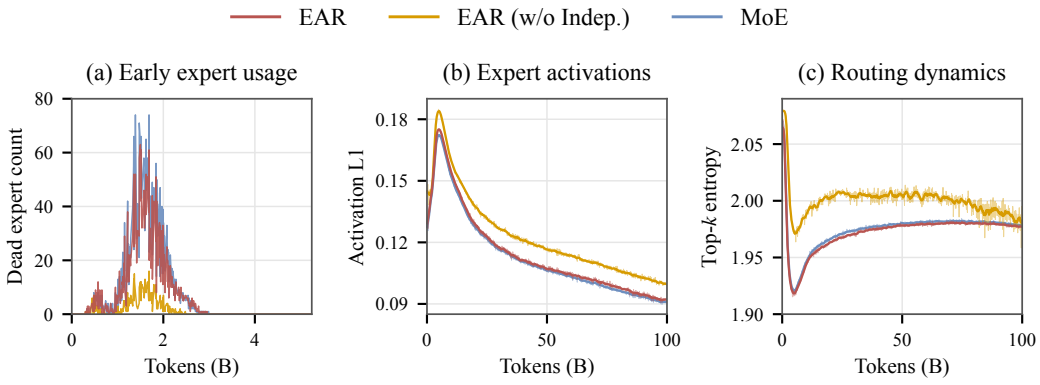


Figure 7: Expert utilization and routing dynamics in 1B diagnostic runs. (a) early inactive-expert counts, (b) internal activation L1, and (c) top-k routing entropy. Panels (b,c) show raw curves with 500-step moving averages. The utilization/activation and entropy diagnostics use separate training configurations.

D Expert Utilization and Routing Dynamics

Figure 7 shows that both EAR and EAR(w/o Indep.) exhibit fewer inactive experts early in training and larger internal activation magnitudes than vanilla MoE, supporting the usefulness of expert-derived routing. The reduction in early inactive experts is particularly pronounced for EAR(w/o Indep.). Meanwhile, EAR exhibits slightly lower routing entropy than MoE and substantially smaller entropy fluctuations than EAR(w/o Indep.), suggesting that the independent router helps stabilize routing dynamics.

E Baseline Hyperparameter Search

Table 7: Grid search for the vanilla MoE baseline. We vary model depth, global batch size, and learning rate, and select the configuration with the best overall validation and downstream performance.

Layers	Global Batch	Learning Rate	Train Loss	Val Loss	Avg. Acc.
16	256	9.3×10^{-4}	2.13319	2.71638	46.02
16	**256**	$\mathbf{8.6 \times 10^{-4}}$	**2.04940**	**2.71223**	**47.65**
10	256	9.3×10^{-4}	2.12858	2.71783	46.14
16	192	8.6×10^{-4}	2.17308	2.72390	46.69
16	192	9.3×10^{-4}	2.05319	2.73371	46.26
10	192	9.3×10^{-4}	2.13218	2.74612	46.46
10	256	8.6×10^{-4}	2.12498	2.72189	46.14
10	192	8.6×10^{-4}	2.20907	2.74474	46.73

Table 8: Pretraining configurations for the three controlled runs.

Configuration item	Small-100B	Large-100B	Large-500B
Model architecture			
Token budget	100B	100B	500B
Layers	16	20	20
Hidden size d	1024	2048	2048
MoE FFN size	128	512	512
Shared FFN size	–	512	512
Experts	160	256	256
Active experts K	8	8	8
MoE layers	16/16	19/20	19/20
Attention heads	8	16	16
KV groups	1	4	4
Training hyperparameters			
Sequence length	8192	8192	8192
Global batch size	256	1024	1024
Optimizer	Muon	Muon	Muon
LR schedule	constant	cosine	constant
Peak LR	8.6×10^{-4}	3×10^{-3}	3×10^{-3}
Minimum LR	8.6×10^{-4}	3×10^{-4}	3×10^{-3}
Token Per Parameter	100	6.25	31.25

We conduct a grid search over model depth, global batch size, and learning rate to obtain a strong vanilla MoE baseline. In total, we evaluate eight configurations. Table 7 reports the final training loss, validation loss, and average downstream accuracy. The 16-layer model with global batch size 256 and learning rate 8.6×10^{-4} achieves the best validation loss and average accuracy, and is therefore adopted as our baseline configuration.

F Model and Training Configurations

This appendix summarizes the pretraining configurations used in our controlled experiments. We report the three configurations used for the 100B-token small-scale run, the 500B-token large-scale run, and the 100B-token large-scale run. The effective token budget is computed as

$$T = I_{\text{train}} \times B_{\text{global}} \times S, \tag{32}$$

where I_{train} is the number of training iterations, B_{global} is the global batch size, and S is the sequence length.

Evaluation Details We report zero-shot downstream accuracy on LAMBADA (Paperno et al., 2016), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2020), ARC-Easy and ARC-Challenge (Clark et al., 2018), PIQA (Bisk et al., 2020), MMLU (Hendrycks et al., 2021a), GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), C-Eval (Huang et al., 2023), CMMLU (Li et al., 2023), BoolQ (Clark et al., 2019), and OpenBookQA (Mihaylov et al., 2018) We additionally report OpenBookQA-Fact and HumanEval (Chen et al., 2021), using Pass@1 for HumanEval. Avg.$_{15}$ is the unweighted average over these 15 downstream scores.

We further evaluate expert utilization using MaxVio, MinVio, their layer-wise averages MeanMaxVio and MeanMinVio, and the fraction of dead neurons defined in Section 3. Router–expert alignment is measured by Top-k Overlap as described in Section 3. Finally, we report the relative change in WikiText perplexity ($\Delta\%$) and the resulting effective training speedup relative to the standard MoE baseline.