MIXTURE OF UNIVERSAL EXPERTS: SCALING VIRTUAL WIDTH VIA DEPTH-WIDTH TRANSFORMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixture-of-Experts (MoE) decouples model capacity from per-token computation, yet their scalability remains limited by the physical dimensions of depth and width. To overcome this, we propose **Mixture of Universal Experts (MOUE)**, a MoE generalization introducing a novel scaling dimension: *Virtual Width*. In general, MoUE aims to reuse a universal layer-agnostic expert pool across layers, converting depth into virtual width under a fixed per-token activation budget. However, two challenges remain: a routing path explosion from recursive expert reuse, and a mismatch between the exposure induced by reuse and the conventional load-balancing objectives. We address these with three core components: a **Staggered Rotational Topology** for structured expert sharing, a **Universal Expert Load Balance** for depth-aware exposure correction, and a **Universal Router** with lightweight trajectory state for coherent multi-step routing. Empirically, MoUE consistently outperforms matched MoE baselines by up to **1.3%** across scaling regimes, enables progressive conversion of existing MoE checkpoints with gains of up to **4.2 percentage points**, and reveals a new scaling dimension for MoE architectures.

1 INTRODUCTION

Large language models, predominantly built on the Transformer architecture (Vaswani et al., 2017), have driven rapid progress in natural language processing. This progress can be shaped by scaling laws, where improvements are closely coupled to increases in model size and training compute (Kaplan et al., 2020; Hoffmann et al., 2022). However, scaling dense Transformers incurs a structural cost: increasing parameters tends to increase both computation (FLOPs) and memory footprint proportionally. The Mixture-of-Experts (MoE) framework alleviates this tension by introducing conditional computation (Fedus et al., 2022; DeepSeek-AI et al., 2024; Feng et al., 2025; Wang et al., 2026), decoupling the total parameters from the per-token activated parameters.

Despite their efficiency, standard MoE architectures remain constrained along two dimensions (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022). In the *depth* dimension, they rely on a fixed layer stacking, which makes it difficult to fully exploit depth for optimizing deep parameters (Shazeer et al., 2017; Lepikhin et al., 2021); moreover, without an explicit recursive structure, they are not naturally suited to fitting complex algorithms that require reusable multi-step computation (Dehghani et al., 2019; Chen et al., 2025). In the *width* dimension, performance is often bounded by the number of experts, yet scaling experts brings substantial system overhead and engineering cost (Fedus et al., 2022; Li et al., 2023). This motivates a fundamental question:

Is there an architecture that can expand model capacity by reusing a model's own depth, while introducing as little additional compute or memory overhead as possible?

The Fourth Dimension (Virtual Width). For a solution, we view *depth* and *width* as two physical scaling dimensions. MOUE introduces a new understanding dimension for MoE, which we call **Virtual Width**: by reusing a shared pool of layer-agnostic Universal Experts (UEs) across layers, MOUE turns additional depth into additional effective width through composable expert paths, while keeping the per-token activation budget fixed. Intuitively, MOUE can be seen as converting a deep, narrow MoE into an extremely wide MoE layer that performs recursive computation over depth.

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.

Importantly, MoUE is a strict superset of standard MoE and is seamlessly compatible: disabling cross-layer reuse recovers the conventional layer-local MoE, whereas enabling reuse decouples physical parameter storage from functional connectivity and unlocks path-composition capacity without requiring expert memory to grow in lockstep.

However, realizing this potential raises two key training challenges. First, cross-layer reuse induces a **combinatorial explosion of routing paths**, making optimization harder and less stable. Second, the standard load-balancing objective does not account for **balanced reuse across depth**, which introduces bias toward repeatedly exposed experts and can suppress the shared pool. We address these challenges with three complementary components: (i) a **Staggered Rotational Topology** that groups consecutive layers into connectivity groups, where layers within a group share the same fixed reachable UE set; (ii) a **Universal Expert Load Balance (UELB)** strategy that explicitly introduces a depth-wise balancing dimension for experts under multi-layer reuse, combined with a warmup schedule to stabilize early exploration; and (iii) an optional **Universal Router** that maintains a lightweight shared-pool state to incorporate preceding tokens into routing decisions.

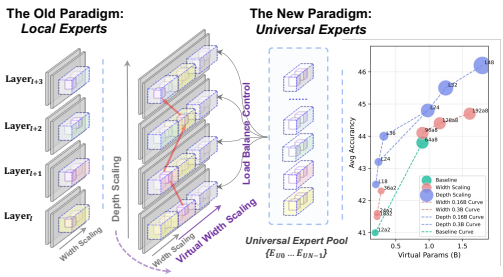


Figure 1: Overview of MoUE. Universal Experts are shared across layers and recursively reused under a fixed activation budget.

For evaluation, we pre-train MoUE models at multiple scales and compare them with matched MoE baselines. In a width-expansion setting, training MoUE from scratch achieves a 1.3% relative performance gain over MoE without increasing parameters or compute. In a depth-expansion setting, MoUE improves performance by 2.5% while keeping FFN parameters unchanged. Across scaling experiments, MoUE establishes a new frontier under both activated-parameter and total-parameter budgets. Moreover, thanks to its compatibility, MoUE can be warm-started from the evaluated MoE checkpoints, with gains depending on the shared-pool size; in continual pre-training, this progressive conversion yields an average gain of 4.2 percentage points, and the advantage persists through large-scale supervised fine-tuning. In short, MoUE reframes MoE scaling with **Virtual Width**, turning depth into capacity via cross-layer expert reuse under a fixed activation budget.

2 Preliminary

The Mixture-of-Experts (MoE) architecture transforms the dense computational paradigm of Transformers by introducing conditional computation (Fedus et al., 2022). Formally, an MoE layer replaces the static feed-forward network (FFN) with a parameterized set of expert functions $\mathcal{E} = \{E_i\}_{i=1}^{N}$. Given an input token representation $x \in \mathbb{R}^d$, a gating network $G_\theta : \mathbb{R}^d \to \Delta^{N-1}$ computes a routing distribution $P(x) = \mathrm{softmax}(W_g x)$. To decouple model capacity from inference cost, a sparse selection mechanism $\mathcal{T}(x) = \mathrm{TopK}(P(x), k)$ is applied, activating only a subset of experts. The layer output is the probability-weighted sum of these activated experts: $y = \sum_{i \in \mathcal{T}(x)} \frac{P_i(x)}{\sum_{j \in \mathcal{T}(x)} P_j(x)} E_i(x)$. While MoE scales parameter count, it remains bound by a fixed depth topology and linear memory growth with respect to expert count, limiting its ability to model complex recursive dependencies or scale width efficiently without prohibitive resource costs.

Standard Load Balancing Optimization. A critical challenge in training MoE models is the *expert collapse* phenomenon. To counteract this, standard protocols optimize an auxiliary load-balancing loss $\mathcal{L}_{\text{balance}}$ alongside the task loss. This is typically formulated as the dot product of the average routing probability vector P and the expert utilization fraction vector f over a batch of N tokens: $\mathcal{L}_{\text{balance}} = N \sum_{i=1}^{N} f_i \cdot P_i$. This objective inherently assumes a uniform prior over expert utility.

3 Motivation: Case for Parameter Reuse

Standard practice implicitly assumes that increasing computational depth must come from introducing new, layer-specific parameters at every layer. We challenge this view and posit that deep networks

exhibit substantial **functional redundancy**: certain specialized operators are repeatedly required and invoked across different levels of semantic abstraction, so strict layer-wise parameter partitioning is neither necessary nor efficient.

Motivated by parameter-fusion studies (Chen et al., 2024a;b), we analyze the **CKA similarity** (Kornblith et al., 2019) of expert weights in a trained standard MoE. First, we observe clear **local functional persistence**: within the consecutive window of L2–L4, many expert pairs achieve CKA larger than 0.4, and the overall CKA similarity is commonly above 0.3. We further repeat the same analysis on a larger **Qwen-30B (Activated 3B)** model after SFT, where expert similarities are even higher; detailed results are provided in the appendix.

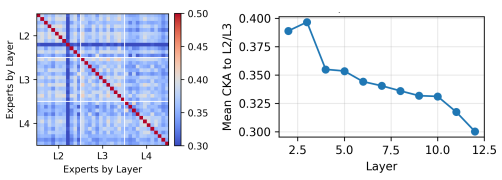


Figure 2: Expert similarity heatmap across shallow and deep layers in a trained MoE.

More importantly, we find pronounced **long-range similarity**. While adjacent layers are indeed much more similar, the very first layers and the very last layers still retain more than 30% similarity. This suggests that certain functional operators are not simply replaced as depth increases, but persist (or reappear) across large depth gaps. These findings suggest that enabling cross-layer expert reuse can mitigate the redundancy caused by repeatedly learning similar expert functions at different depths in standard MoE, thereby allowing more efficient utilization of depth and increased effective capacity through parameter sharing and path composition under a fixed activation budget.

4 The Mixture of Universal Experts (MoUE)

4.1 General Framework

To relax the rigid layer-partitioning assumption in standard MoEs, we propose MoUE, which reinterprets a deep MoE not as a static stack of disjoint expert sets, but as a recursive computation over a unified addressable expert space.

Recursive Formulation. Consider a Transformer of depth L. We decompose experts into layer-local sets $\mathcal{E}_\ell^{\text{local}}$ and a shared pool of Universal Experts (UEs) $\mathcal{E}^{\text{u}}$, with $\mathcal{E}^\star = (\bigcup_\ell \mathcal{E}_\ell^{\text{local}}) \cup \mathcal{E}^{\text{u}}$. MoUE is defined by a connectivity mapping $\mathcal{C} : \mathcal{E}^\star \times \{1, \dots, L\} \to \{0, 1\}$ that specifies which experts are reachable at each layer. The MoE update at layer ℓ becomes a conditional query over this topology:

$$h^{(\ell)} = h^{(\ell-1)} + \sum_{E_i \in \mathcal{E}^\star} \mathcal{C}(E_i, \ell) \cdot \pi_i^{(\ell)}(h^{(\ell-1)}) \cdot E_i(h^{(\ell-1)}). \tag{1}$$

For UEs, $\mathcal{C}(u, \ell)$ can be active for multiple layers, enabling cross-layer reuse: the same expert parameters can be invoked at different semantic stages, decoupling parameter storage from depth-wise computation.

Theoretical Capacity and Optimization Bottleneck: Virtual Width Cross-layer expert reuse unlocks substantial combinatorial capacity, which we term **Virtual Width**. While standard MoE expands compositional space by increasing physical width (more experts per layer, and thus linear memory growth with depth), allowing a shared pool of N_u experts at each layer enables the router to form exponentially many depth-L expert paths: $|\mathcal{T}|_{\text{MoUE}} \approx \left(\binom{N_u}{k}\right)^L$. This exponential growth leverages depth-wise composition without increasing expert storage beyond N_u. However, such Virtual Width introduces optimization challenges: unconstrained reuse creates a vast routing space, weakens gradient signals, and may cause repeated selection of the same experts. Thus, effective utilization of this capacity requires structural constraints and tailored training objectives.

4.2 Structural Optimization: Staggered Connectivity

To make Virtual Width trainable, we need to constrain connectivity. In principle, there are many possible reuse topologies; in our exploration, we found that a simple yet structured design works best

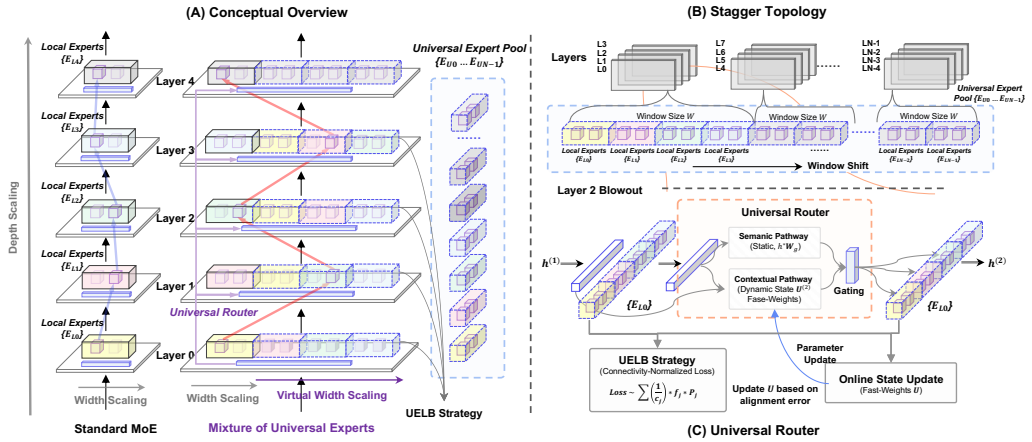


Figure 3: An illustration of MoUE. Each layer routes tokens to a small set of experts, consisting of layer-local experts and a shared pool of Universal Experts (UEs). UEs are connected to multiple layers, enabling cross-layer reuse while keeping the per-token activation budget fixed.

in practice: the **Staggered Rotational Topology**. It acts as a structural regularizer on the connectivity map $\mathcal{C}$, controlling the reachable expert space across depth.

Two-Level Expert Organization within Consecutive Groups. We organize the expert pool at two levels. At the group level, a fixed pool defines the shared experts accessible to a connectivity group comprising G consecutive layers, ensuring that all layers within the group share the same set of reachable UEs. At the layer level, each layer retains its private experts alongside this shared pool. In the primary SRT configuration, pool membership is fixed within each group and does not slide with depth. For a shared expert E_j, the connectivity of group $g(\ell) = \lfloor (\ell - 1)/G \rfloor$ is given by

$$\mathcal{C}(E_j, \ell) = \mathbb{I}\left(E_j \in \mathcal{U}_{g(\ell)}\right), \tag{2}$$

where $\mathcal{U}_{g(\ell)}$ is the group's shared UE pool and $W = |\mathcal{U}_{g(\ell)}|$ is its size. This design preserves layer-private specialization while enabling controlled cross-layer reuse within each group. Group size determines the depth range of reuse, and pool size determines the shared candidates accessible to each layer.

Implementation. We realize $\mathcal{C}$ as per-layer allow-lists over global expert IDs, masking unreachable experts (routing logits set to $-\infty$) before Top-k selection. This reduces the shared-branch candidate set from the full pool of N_u experts to W, transforming global routing into a sequence of localized decisions and improving optimization stability.[1] Sliding-window, reverse-order and sandwich variants are compared in Appendix C. In practice, a shared pool can be placed within an existing expert-parallel (EP) locality so that its cross-layer references reuse resident experts.

4.3 Universal Expert Load Balance (UELB)

While staggered connectivity makes routing feasible, it exposes a fundamental mismatch: standard load balancing offers weak optimization signals for recursive, multi-step expert reuse. Because MoUE relies on exponentially many depth-wise expert paths, the objective must encourage not only intra-layer balance but also inter-layer diversity and path coverage. Traditional load balancing struggles due to (i) **Heterogeneous Exposure**: shared experts are over-penalized for being reachable from many layers; and (ii) **Combinatorics Blindness**: it ignores path redundancy across layers, causing underutilization of Virtual Width.

Based on these observations, we propose **Universal Expert Load Balance (UELB)**. The core innovation is to introduce a *reuse-aware depth-wise* balancing dimension: "balance" is no longer defined purely within a layer, but is calibrated across recursive reuse opportunities over depth. The guiding principle is simple: load balancing should be enforced *relative to topological opportunity*.

[1] Routing complexity is reduced from $\mathcal{O}(L \cdot N_u)$ (all-to-all) to $\mathcal{O}(L \cdot W)$.

That is, the objective should penalize an expert only when it is over-preferred *within the contexts where it is available*, rather than because it is reachable more often by design.

Concretely, let $c_j = \sum_\ell \mathcal{C}(E_j, \ell)$ denote expert j's topological degree (exposure count). UELB rescales the universal branch by $1/c_j$, so the objective reflects *per-exposure* utilization rather than cross-layer aggregated usage:

$$\mathcal{L}_{\text{UELB}} = \alpha_{\text{loc}} \sum_{\ell} \sum_{i \in \mathcal{E}^{\text{loc}}} \bar{f}_i \bar{P}_i + \alpha_{\text{u}} \sum_{j \in \mathcal{E}^{\text{u}}} \frac{1}{c_j} \left(\sum_{\ell=1}^{L} \mathcal{C}(E_j, \ell)\, \bar{f}_j^{(\ell)} \bar{P}_j^{(\ell)} \right). \tag{3}$$

where $\bar{f}$ and $\bar{P}$ are the average routing frequency and expected utilization. The normalization serves two purposes: it removes architecture-induced penalties from heterogeneous exposure (decoupling reachability from preference), and it provides an optimizable depth-wise balance signal that encourages more uniform coverage of the shared expert space across recursive steps, thereby converting combinatorial path complexity into usable capacity.

In practice, we estimate UELB group-wise (layers sharing the same reachability mask) and compute a group-level auxiliary loss with a scale correction to avoid an artificial loss shrinkage due to concatenation. We further apply a lightweight warmup to encourage early exploration of the shared pool and prevent premature path collapse.

4.4 Universal Router

Staggered connectivity makes the routing search space local, and UELB promotes depth-wise balance. To further enhance routing over the shared pool, we introduce the optional **Universal Router**, which augments routing with a lightweight state so decisions can depend on preceding tokens. Each shared pool has a randomly initialized state with a fixed expert-ID ordering, shared by the layers that access that pool.

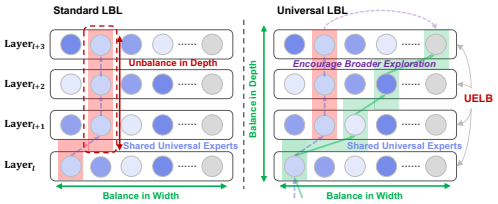


Figure 4: LBL only focuses on balancing expert selection within each layer, while ULBL promotes balanced and diverse expert selection across both depth and width.

Dual-Pathway Routing. We decompose routing logits into a semantic pathway and a contextual pathway: $z^{(\ell)} = h^{(\ell)} W_g + b_g + \beta \cdot \text{Sim}(h^{(\ell)}, U^{(\ell)})$. The semantic term performs standard affine matching. The contextual term reads the shared-pool state $U^{(\ell)}$ updated by earlier layers accessing the same pool. The layer superscript identifies where the state is read; layers accessing the same pool reuse its state, carrying routing information across depth.

Online Fast-Weight Update. After routing at layer ℓ, we update $U^{(\ell)}$ online as a forward-only state without backpropagation. Let $K^{(\ell)}$ stack the current layer's projected token keys and $P_{\text{u}}^{(\ell)}$ contain their detached dispatch weights restricted to shared experts, with zero entries for unselected experts and no separate renormalization of the shared branch. The linear prediction and residual are $\hat{P}^{(\ell)} = K^{(\ell)}(U^{(\ell)})^\top$ and $\Delta^{(\ell)} = \hat{P}^{(\ell)} - P_{\text{u}}^{(\ell)}$. We apply $U^{(\ell)} \leftarrow \text{sg}(U^{(\ell)} - \eta_\ell (\Delta^{(\ell)})^\top K^{(\ell)} / N_{\text{tok}})$, where N_{tok} is the number of token keys in this update and sg stops gradients through it. Token contributions are aggregated within the current layer, and the updated state is used by subsequent layers accessing the same shared pool. Full description in Appendix E.

4.5 Progressive Warm-Start

Training MoUE from scratch is the cleanest setting but can be costly. We therefore propose a **Progressive Transformation Strategy** to convert an existing pre-trained MoE checkpoint into MoUE with a curriculum-style transition.

1. Initializing the Universal Pool. We construct the initial UE pool $\mathcal{E}^{\text{u}}$ by randomly selecting experts from the source MoE and sharing references to their trained FFN weights. Source layers retain their original expert access. This conversion reuses existing FFN weights, retains the source model's total physical parameter budget, and uses ordinary routing without UR or adapters.

2. Curriculum Routing Warmup via Logit Suppression. To introduce cross-layer reuse without disrupting the pre-trained feature space, we keep the original local router and add a parallel universal projection. We then apply a time-dependent negative bias to the universal branch:

$$z_i^{(\ell)} = \begin{cases} h^{(\ell-1)} \cdot W_r^{\text{local}}[\cdot, i] & \text{if } E_i \in \mathcal{E}_\ell^{\text{local}} \\ h^{(\ell-1)} \cdot W_r^{\text{u}}[\cdot, i] - \beta(t) & \text{if } E_i \in \mathcal{E}^{\text{u}} \end{cases} \tag{4}$$

with $\beta(t) \geq 0$ annealed from a large value (suppressing UE usage at the start) to 0, so the model gradually shifts from the original layer-local behavior to topology-aware reuse. Details in App. F.

5 Experiments

5.1 Experimental Setup

Data and Training. We pre-train on samples from OLMo 2 Mix 1124 (3.4T tokens) and follow a fixed data-to-parameter ratio, reserving 2M tokens for validation. All models are trained with Megatron on H800 GPUs using sequence length 4096; we match tokens and activated budgets for iso-parameter and iso-compute comparisons. For warm-start conversion, we initialize from MoE checkpoints and perform continued training under matched settings.

Baselines and Scaling Settings. We evaluate with lm-evaluation-harness (Gao et al., 2024) against two Qwen-3 style MoE backbones spanning different scales and routing granularities: (i) MoE-160M with 12 experts and top-2 routing, and (ii) MoE-700M with 64 experts and top-8 routing. Our main comparisons study two settings. **Width expansion:** MoUE increases capacity only through Virtual Width (larger path space), without increasing activated parameters or total physical parameters. **Depth expansion:** MoUE performs depth scaling by sharing FFN parameters across depth. We report three budgets: activated parameters (Act), total physical parameters (TP), and virtual parameters (VP), where VP counts the effective capacity including Virtual Width. Full details are in Appendix G.

5.2 Main Result

Width Expansion We investigate width expansion, where MoUE increases effective capacity by enlarging the Universal Expert pool without raising activated or total parameters; only virtual parameters grow, reflecting more compositional paths. As shown in Table 1, this yields consistent accuracy gains at fixed compute budgets: for MoE 12A2, Avg. improves from 41.0 to 42.3, and for MoE 64A8, from 43.8 to 44.7, with notable improvements on LogiQA, BoolQ, and LAMBADA. Performance scales further as VP increases, demonstrating that MoUE effectively translates additional virtual capacity into measurable improvements.

Depth Expansion We next examine depth expansion, where MoUE scales the number of computation steps by reusing experts across layers, thus increasing depth with minimal growth in FFN parameters. This approach is highly parameter-efficient: for the 12A2 backbone, MoUE L36 achieves Avg. 44.0 (+3.0) with TP only slightly above baseline; for 64A8, MoUE L48 reaches Avg. 46.2 (+2.4) at TP=0.9B. Gains increase with model depth and scale, and notably, MoUE L36 surpasses MoE 64A8 L16 in accuracy while using about half as activated and total parameters, demonstrating strong scalability under constrained budgets.

Training Dynamics and Stability. Figure 5 summarizes the training-time behavior. Under width expansion, MoUE achieves better validation loss at the same FLOPs, indicating improved compute efficiency without increasing Act or TP. Under depth expansion, MoUE overtakes the baseline early and the gap continues to widen as training proceeds, suggesting a higher performance ceiling with longer training. Across settings, the Max/Mean ratio remains well-controlled, consistent with stable routing dynamics induced by staggered connectivity and UELB.

Scaling Analysis across TP, VP, and Act. Figure 6 summarizes scaling trends across three budgets: total parameters (TP), virtual parameters (VP), and activated parameters (Act). MoUE consistently achieves higher average accuracy than MoE at matched TP, VP, or Act. For TP, width scaling improves accuracy without increasing physical storage, while depth scaling yields even larger gains with only modest TP growth, demonstrating parameter efficiency. For VP, increasing virtual parameters leads to clear accuracy improvements, with depth expansion showing a steeper accuracy gain than width expansion, indicating more effective use of virtual capacity. For Act, MoUE outperforms standard

Table 1: Main results of MoUE width/depth expansion on two Qwen-3 style MoE baselines (MoE-0.16B and MoE-0.7B). We expand the **virtual width** ($\times 1.5$, $\times 2$, $\times 3$) and **depth** ($\times 1.5$, $\times 2$, $\times 3$) via cross-layer expert reuse.

Model	Act.	TP.	VP.	Commonsense & Reading Comprehension						Continued		LM	Avg.
				SciQ	PIQA	WG	ARC-E	ARC-C	Hella.	LogiQA	BoolQ	Lam.	
MoE 12A2 L12	0.1B	0.16B	0.16B	52.0	62.9	49.4	41.3	19.9	30.1	19.7	57.3	36.4	41.0
Width scaling													
MoUE 18A2	0.1B	0.16B	0.19B	**52.6**	62.6	50.2	41.1	**21.6**	30.1	21.4	57.6	36.2	41.5
MoUE 24A2	0.1B	0.16B	0.22B	52.5	**63.3**	50.9	**41.8**	**21.6**	29.3	19.7	59.6	36.1	41.6
MoUE 36A2	0.1B	0.16B	0.28B	**52.6**	**63.3**	**51.2**	41.4	20.4	**31.0**	**23.2**	**60.1**	**37.1**	**42.3**
Depth scaling													
MoUE L18	0.12B	0.16B	0.2B	52.8	62.4	52.2	**40.0**	22.2	30.4	24.6	60.9	37.4	42.5
MoUE L24	0.13B	0.18B	0.24B	55.1	62.8	52.8	39.0	23.2	30.7	25.8	60.3	38.8	43.2
MoUE L36	0.16B	0.19B	0.32B	**56.0**	**63.7**	**53.7**	39.9	**24.1**	**31.5**	**26.7**	**61.2**	**39.6**	**44.0**
MoE 64A8 L16	0.3B	0.7B	0.7B	56.2	65.5	51.5	43.7	22.1	32.9	23.0	57.6	41.8	43.8
Width scaling													
MoUE 96A8	0.3B	0.7B	0.9B	56.7	**65.5**	51.6	43.2	22.4	**33.1**	23.8	57.7	42.6	44.1
MoUE 128A8	0.3B	0.7B	1.16B	57.2	65.1	51.7	**44.7**	22.7	32.4	24.5	57.8	**43.4**	44.4
MoUE 192A8	0.3B	0.7B	1.61B	**57.8**	64.7	**52.8**	44.4	**23.1**	32.8	**25.3**	**57.9**	43.3	**44.7**
Depth scaling													
MoUE L24	0.38B	0.75B	0.98B	57.2	65.3	52.9	42.7	23.8	33.2	26.4	59.3	42.8	44.8
MoUE L32	0.46B	0.8B	1.25B	57.2	**65.6**	53.6	43.2	24.8	33.5	27.8	60.2	43.5	45.5
MoUE L48	0.61B	0.9B	1.8B	**58.3**	65.2	**54.3**	**44.0**	**25.8**	**33.9**	**28.8**	**61.0**	**44.2**	**46.2**

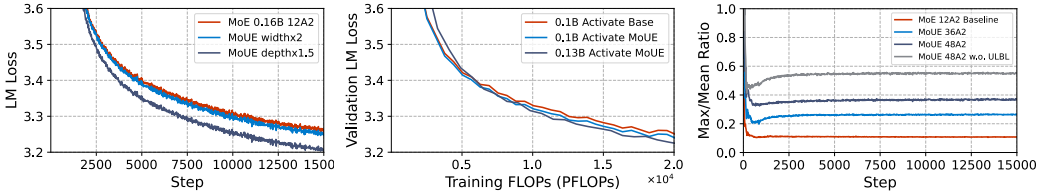


Figure 5: **Left:** LM training loss versus training steps. **Middle:** Validation loss versus training FLOPs. **Right:** Max/Mean ratio versus training steps (routing skew / load-balance indicator).

MoE at the same activated budget, and the advantage grows as Act increases, showing that MoUE leverages combinatorial computation to better utilize activated parameters.

Progressive Warm-Start Training MoUE from scratch is ideal but costly, so we investigate converting existing MoE checkpoints, including JetMoE (Shen et al., 2024) and OLMoE (Muennighoff et al., 2024), into MoUE under matched training and evaluation settings. These experiments use OpenCompass generation and answer extraction, with the same protocol for each baseline and converted model. As shown in Table 2, the best listed SFT configurations improve the nine-task averages by 1.1 percentage points for JetMoE-8E (53.7 to 54.8) and 1.9 points for OLMoE-64E (46.5 to 48.4); OLMoE-64E-CPT improves by 4.2 points (30.2 to 34.4). The shared-pool sweep improves up to twelve shared experts per layer and decreases at the larger tested setting, illustrating the role of pool size in progressive warm-start conversion. Complete results are provided in Appendix F.

5.3 Ablation Studies

Setup. We evaluate ablations under matched training tokens and a fixed activated-parameter budget on the MoUE 36A2 setting, and report validation perplexity. This isolates whether each component is necessary to make Virtual Width *usable* rather than merely *expressible*.

Components Ablation. Ablation results in Table 3 evaluate the contributions of the key components in MoUE. Removing the **staggered rotational topology** leads to the largest degradation (+2.54 PPL), showing that Virtual Width requires structured connectivity for effective optimization; omitting **universal experts** also hurts performance, confirming the importance of cross-layer reuse. Among topology variants, our staggered rotational design consistently outperforms alternatives. For load balancing, replacing the proposed **UELB** with standard methods substantially increases perplexity (+0.97), and dropping connectivity normalization further degrades results, due to biased expert exposure and reduced path diversity. Probabilistic warmup helps early training. Finally, removing

Table 2: Comprehensive evaluation of open-source MoE baselines and corresponding MoUE settings. Each baseline is an open-source MoE model; below each are three MoUE variants with their specific metrics.

Model	Act.	VP.	ARC-c	ARC-e	GSM8K	HellaS.	HumanE.	NQ	Tri.QA	Wino.	MMLU	AVG
JetMoE-8E, adapted by supervised fine-tuning.												
Vanilla MoE	2.2B	8.0B	64.4	79.0	44.7	78.7	42.7	20.4	49.5	56.6	47.7	53.7
MoUE 10A2	2.2B	9.7B	**66.8**	78.8	42.2	78.4	48.2	20.9	50.7	54.1	46.3	54.1
MoUE 12A2	2.2B	11.3B	65.1	79.4	**45.1**	**80.3**	42.7	**21.4**	50.5	56.8	**48.2**	54.4
MoUE 16A2	2.2B	14.6B	65.8	**79.9**	45.0	79.5	**45.7**	20.5	**51.2**	**57.9**	47.9	**54.8**
OLMoE-64E, adapted by supervised fine-tuning.												
Vanilla MoE	1.3B	6.9B	57.0	76.7	**36.3**	**62.1**	24.2	18.1	45.3	52.3	47.1	46.5
MoUE 68A8	1.3B	7.3B	61.0	79.2	34.0	59.4	27.4	18.1	44.1	54.4	47.3	47.2
MoUE 72A8	1.3B	7.7B	61.4	79.5	35.5	58.7	**29.3**	18.0	44.5	**56.5**	46.0	47.7
MoUE 76A8	1.3B	8.1B	**62.7**	**79.9**	34.0	**62.1**	27.4	**19.1**	**47.6**	55.3	**47.7**	**48.4**
OLMoE-64E, adapted by continued pre-training.												
Vanilla MoE	1.3B	6.9B	37.3	40.2	31.0	26.3	13.4	9.6	18.4	49.1	46.3	30.2
MoUE 68A8	1.3B	7.3B	43.7	45.9	**33.5**	26.2	14.0	9.3	22.0	49.9	47.5	32.4
MoUE 72A8	1.3B	7.7B	42.0	48.9	30.0	**29.4**	13.4	**10.7**	**30.2**	48.5	**50.4**	33.7
MoUE 76A8	1.3B	8.1B	**50.9**	**58.2**	31.5	26.4	**14.6**	8.0	22.9	**50.4**	47.1	**34.4**

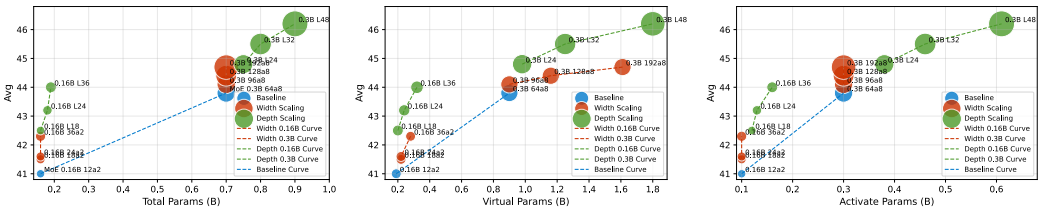


Figure 6: **Left:** Average accuracy versus total parameters, with point size representing activated parameters. **Middle:** Average accuracy versus virtual parameters, with point size representing activated parameters. **Right:** Average accuracy versus activated parameters.

the **contextual pathway** or the **fast-weight update** increases PPL from 25.40 to 25.83 or 25.59, respectively. The stateless configuration still improves on the MoE baseline (26.25), showing that stateful routing further enhances shared expert access. Together, these findings support structured connectivity and connectivity-aware load balancing, with UR providing an additional improvement.

Warm-Start Warmup Ablation. The warm-start conversion relies on a logit-suppression curriculum $\beta(t)$ (Section F): at the beginning, universal experts are strongly suppressed so the converted model behaves like the source MoE; $\beta(t)$ is then annealed to 0 to gradually introduce reuse. If we ablate this warmup and turn on universal routing immediately, the pre-trained representations are disrupted and routing becomes unstable, which degrades convergence and downstream quality.

5.4 Analysis

Universal Expert Utilization. We analyze the activation frequency of Universal Experts across layers to assess whether MoUE achieves cross-layer reuse. As shown in Figure 7, UELB enables Universal Experts to be activated broadly in both shallow and deep layers, with usage patterns comparable to layer-private experts. In deeper layers, private experts show specialization but are still selected by multiple layers, demonstrating effective parameter sharing.

UELB Improves Load Balance Dynamics. To assess training stability, we monitor routing skew via the Max/Mean ratio. Figure 5 (right) shows that without UELB, routing quickly collapses, while UELB maintains low skew throughout. By normalizing for expert exposure and enforcing depth-wise balance, UELB ensures that Virtual Width translates to capacity rather than repeated paths.

Topology Impact and Functional Circuits. Table 4 shows that the staggered rotational topology yields the best performance and forms stable expert circuits, while other topologies introduce

Table 3: Eval perplexity for ablations on the MoUE framework under matched training tokens and a fixed activated-parameter budget. "w/o" denotes removing the corresponding component from the full MoUE model.

Method	Perplexity ↓
MoUE 36A2 (full model)	25.40
Connectivity / Expert Topology	
w/o Staggered Rotational Topology	27.94 (+2.54)
w/o Universal Experts	26.05 (+0.65)
Routing Mechanism	
w/o Contextual Pathway	25.83 (+0.43)
w/o Fast-Weight Update	25.59 (+0.19)
Load Balancing Strategy	
w/o UELB (standard load balance)	26.37 (+0.97)
w/o Connectivity Normalization	25.75 (+0.35)
w/o Probabilistic Warmup	25.66 (+0.26)
MoE 12A2 (iso-activate / full parameter)	26.25 (+0.85)

Table 4: Ablation study on connectivity topologies. We compare our **Staggered Rotational** design against alternative connectivity schemes under matched training tokens and a fixed activated-parameter budget.

Topology Scheme	Perplexity ↓	Circuit Pattern
Staggered Rotational (Ours)	**25.40**	**Re-entrant Loops**
Forward Sliding Window	25.87 (+0.47)	Sequential Drift
Reverse-Order Connection	25.93 (+0.53)	Disordered Flow
Sandwich Architecture	25.61 (+0.21)	Rigid Stratification
Full All-to-All (Unconstrained)	27.94 (+2.54)	Random / Unstable

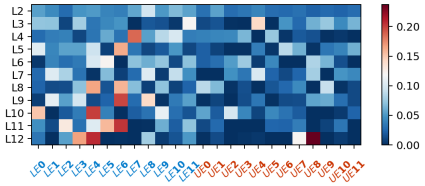


Figure 7: Expert utilization heatmap. LE: Local Expert; UE: Universal Expert.

instability or disordered expert flow, highlighting the importance of structured connectivity for compositional capacity.

Effect of Universal Expert Pool Size. Figure 9 shows that increasing the number of Universal Experts (UEs) per layer in the warm-start setting improves performance up to an optimal point (12 UEs), after which further increases degrade results. This demonstrates that moderate sharing effectively exploits cross-layer redundancy, but excessive sharing can harm layer-specific specialization.

Domain Specialization and Layer-wise Utilization of Universal Experts. As shown in Figure 10, universal experts develop distinct domain preferences (e.g., some specialize in Math, others in Code), indicating spontaneous functional specialization and validating the Universal Router's ability to dynamically match tokens to relevant experts.In Figure 11, that UEs are used sparsely in early layers but much more in deeper layers, suggesting the model learns to rely on LEs for low-level features and on UEs for high-level reasoning.

6 Related Work

MoE models and Shared computation. (Shazeer et al., 2017) employ learnable gating to activate a subset of experts per token, enabling efficient scaling (Lepikhin et al., 2021; Fedus et al., 2022; Jiang et al., 2024; Gu et al., 2025). Routing approaches span load balancing (Fedus et al., 2022; Lepikhin et al., 2021), hash-based (Roller et al., 2021), expert-choice (Zhou et al., 2022), BASE layers (Lewis et al., 2021), and soft mixtures (Puigcerver et al., 2024), but do not address the recursive, multi-layer exposure of universal experts in MoUE. Universal Transformers (Dehghani et al., 2019) and layer-sharing approaches (Lan et al., 2020; Ng & Wang, 2024) reuse transformations across depth. Recent work explores implicit reasoning (Deng et al., 2023; Hao et al., 2024) and test-time scaling (Geiping et al., 2025). MoEUT (Csordás et al., 2024) combines sparse feedforward and attention experts with shared layer groups. Further discussion is in Appendix A.

7 Conclusion

We introduce Mixture of Universal Experts (MoUE), which expands capacity through cross-layer expert reuse and Virtual Width. Structured connectivity, connectivity-normalized load balancing, and stateful routing enable this expansion under fixed activated-parameter or compute budgets. Experiments show consistent gains over MoE baselines in width and depth expansion settings. Additional method and experimental details are provided in the appendix.

Ethics Statement

This work studies a foundational architecture for improving the parameter and compute efficiency of Mixture-of-Experts language models. Its potential positive impact is to make stronger models more accessible under fixed activated-parameter or compute budgets, thereby reducing the marginal cost of training and adapting large models. However, more efficient language model scaling can also lower the cost of building capable generative systems, which may indirectly amplify known risks of language models, including misinformation generation, unsafe automation, biased outputs, or malicious use. Our work does not introduce a user-facing system, collect private user data, or release a high-risk pretrained model, but any future release of checkpoints or large-scale systems based on this architecture should follow appropriate model release policies, dataset licensing requirements, and safety evaluation practices.

Reproducibility Statement

The model formulation, staggered connectivity, load-balancing objective, and router are described in Sections 4.1–4.4 and Appendices B–E. The progressive warm-start procedure is detailed in Appendix F and Algorithm 1. Section 5.1, Appendix G, and Table 11 provide the reported data and training settings, model configurations, evaluation tasks and shot counts, and hyperparameters.

References

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7432–7439. AAAI Press, 2020. doi: 10.1609/aaai.v34i05.6239. URL https://aaai.org/ojs/index.php/AAAI/article/view/6239.

Yilong Chen, Junyuan Shang, Zhenyu Zhang, Shiyao Cui, Tingwen Liu, Shuohuan Wang, Yu Sun, and Hua Wu. LEMON: Reviving stronger and smaller LMs from larger LMs with linear parameter fusion. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8005–8019, Bangkok, Thailand, 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.434. URL https://aclanthology.org/2024.acl-long.434.

Yilong Chen, Linhao Zhang, Junyuan Shang, Zhenyu Zhang, Tingwen Liu, Shuohuan Wang, and Yu Sun. DHA: learning decoupled-head attention from transformer checkpoints via adaptive heads fusion. In *Advances in Neural Information Processing Systems*, volume 37, pp. 45879–45913, 2024b. doi: 10.52202/079017-1459. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/518a75257f37b32f711082dff33c2ffc-Abstract-Conference.html.

Yilong Chen, Junyuan Shang, Zhenyu Zhang, Yanxi Xie, Jiawei Sheng, Tingwen Liu, Shuohuan Wang, Yu Sun, Hua Wu, and Haifeng Wang. Inner thinking transformer: Leveraging dynamic depth scaling to foster adaptive internal thinking, 2025. URL https://arxiv.org/abs/2502.13842.

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 Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *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. *ArXiv preprint*, abs/1803.05457, 2018. URL https://arxiv.org/abs/1803.05457.

Róbert Csordás, Kazuki Irie, Jürgen Schmidhuber, Christopher Potts, and Christopher D. Manning. MoEUT: Mixture-of-Experts Universal Transformers, 2024. URL https://arxiv.org/abs/2405.16039.

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, 2024. URL https://arxiv.org/abs/2401.06066.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jin Chen, Jingyang Yuan, Junjie Qiu, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruizhe Pan, Runxin Xu, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Size Zheng, T. Wang, Tian Pei, Tian Yuan, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Liu, Xin Xie, Xingkai Yu, Xinnan Song, Xinyi Zhou, Xinyu Yang, Xuan Lu, Xuecheng Su, Y. Wu, Y. K. Li, Y. X. Wei, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Zheng, Yichao Zhang, Yiliang Xiong, Yilong Zhao, Ying He, Ying Tang, Yishi Piao, Yixin Dong, Yixuan Tan, Yiyuan Liu, Yongji Wang, Yongqiang Guo, Yuchen Zhu, Yuduan Wang, Yuheng Zou, Yukun Zha, Yunxian Ma, Yuting Yan, Yuxiang You, Yuxuan Liu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhewen Hao, Zhihong Shao, Zhiniu Wen, Zhipeng Xu, Zhongyu Zhang, Zhuoshu Li, Zihan Wang, Zihui Gu, Zilin Li, and Ziwei Xie. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model, 2024. URL https://arxiv.org/abs/2405.04434.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://arxiv.org/abs/1807.03819.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation, 2023. URL https://arxiv.org/abs/2311.01460.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1):5232–5270, 2022.

Yuchen Feng, Bowen Shen, Naibin Gu, Jiaxuan Zhao, Peng Fu, Zheng Lin, and Weiping Wang. Dive into moe: Diversity-enhanced reconstruction of large language models from dense into mixture-of-experts. *ArXiv preprint*, abs/2506.09351, 2025. URL https://arxiv.org/abs/2506.09351.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach, 2025. URL https://arxiv.org/abs/2502.05171.

Naibin Gu, Zhenyu Zhang, Yuchen Feng, Yilong Chen, Peng Fu, Zheng Lin, Shuohuan Wang, Yu Sun, Hua Wu, Weiping Wang, and Haifeng Wang. Elastic moe: Unlocking the inference-time scalability of mixture-of-experts, 2025. URL https://arxiv.org/abs/2509.21892.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space, 2024. URL https://arxiv.org/abs/2412.06769.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *ArXiv preprint*, abs/2203.15556, 2022. URL https://arxiv.org/abs/2203.15556.

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, 2024. URL https://arxiv.org/abs/2401.04088.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *ArXiv preprint*, abs/2001.08361, 2020. URL https://arxiv.org/abs/2001.08361.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://arxiv.org/abs/1909.11942.

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. In *9th International Conference on Learning Representations (ICLR 2021), Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://arxiv.org/abs/2006.16668.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. BASE layers: Simplifying training of large, sparse models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6265–6274. PMLR, 2021. URL http://proceedings.mlr.press/v139/lewis21a.html.

Jiamin Li, Yimin Jiang, Yibo Zhu, Cong Wang, and Hong Xu. Accelerating distributed MoE training and inference with lina. In *2023 USENIX Annual Technical Conference (USENIX ATC 23)*, pp. 945–959, Boston, MA, July 2023. USENIX Association. ISBN 978-1-939133-35-9. URL https://www.usenix.org/conference/atc23/presentation/li-jiamin.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In Christian Bessiere (ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pp. 3622–3628. ijcai.org, 2020. doi: 10.24963/ijcai.2020/501. URL https://doi.org/10.24963/ijcai.2020/501.

Jan Ludziejewski, Jakub Krajewski, Kamil Adamczewski, Maciej Pióro, Michał Krutul, Szymon Antoniak, Kamil Ciebiera, Krystian Król, Tomasz Odrzygóźdź, Piotr Sankowski, Marek Cygan, and Sebastian Jaszczur. Scaling laws for fine-grained mixture of experts. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 33270–33288. PMLR, 2024. URL https://proceedings.mlr.press/v235/ludziejewski24a.html.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. Olmoe: Open mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2409.02060.

Kei-Sing Ng and Qingchen Wang. Loop neural networks for parameter sharing, 2024. URL https://arxiv.org/abs/2409.14199.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, Berlin, Germany, 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1144. URL https://aclanthology.org/P16-1144.

Joan Puigcerver, Carlos Riquelme Ruiz, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=jxpsAj7ltE.

Stephen Roller, Sainbayar Sukhbaatar, Arthur Szlam, and Jason Weston. Hash layers for large sparse models. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 17555–17566, 2021.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 8732–8740. AAAI Press, 2020. doi: 10.1609/aaai.v34i05.6399. URL https://aaai.org/ojs/index.php/AAAI/article/view/6399.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations (ICLR 2017), Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Yikang Shen, Zhen Guo, Tianle Cai, and Zengyi Qin. Jetmoe: Reaching llama2 performance with 0.1m dollars, 2024. URL https://arxiv.org/abs/2404.07413.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://arxiv.org/abs/1706.03762.

Haifeng Wang, Hua Wu, Tian Wu, Yu Sun, Jing Liu, Dianhai Yu, Yanjun Ma, Jingzhou He, Zhongjun He, Dou Hong, Qiwen Liu, Shuohuan Wang, Junyuan Shang, Zhenyu Zhang, Yuchen Ding, Jinle Zeng, Jiabin Yang, Liang Shen, Ruibiao Chen, Weichong Yin, Siyu Ding, Dai Dai, Shikun Feng, Siqi Bao, Bolei He, Yan Chen, Zhenyu Jiao, Ruiqing Zhang, Zeyu Chen, Qingqing Dang, Kaipeng Deng, Jiajun Jiang, Enlei Gong, Guoxia Wang, Yanlin Sha, Yi Liu, Yehan Zheng, Weijian Xu, Jiaxiang Liu, Zengfeng Zeng, Yingqi Qu, Zhongli Li, Zhengkun Zhang, Xiyang Wang, Zixiang Xu, Xinchao Xu, Zhengjie Huang, Dong Wang, Bingjin Chen, Yue Chang, Xing Yuan, Shiwei Huang, Qiao Zhao, Xinzhe Ding, Shuangshuang Qiao, Baoshan Yang, Bihong Tang, Bin Li, Bingquan Wang, Binhan Tang, Binxiong Zheng, Bo Cui, Bo Ke, Bo Zhang, Bowen Zhang, Boyan Zhang, Boyang Liu, Caiji Zhang, Can Li, Chang Xu, Chao Pang, Chao Zhang, Chaoyi Yuan, Chen Chen, Cheng Cui, Chenlin Yin, Chun Gan, Chunguang Chai, Chuyu Fang, Cuiyun Han, Dan Zhang, Danlei Feng, Danxiang Zhu, Dong Sun, Dongbo Li, Dongdong Li, Dongdong Liu, Dongxue Liu, Fan Ding, Fan Hu, Fan Li, Fan Mo, Feisheng Wu, Fengwei Liu, Gangqiang Hu, Gaofeng Lu, Gaopeng Yong, Gexiao Tian, Guan Wang, Guangchen Ni, Guangshuo Wu, Guanzhong Wang, Guihua Liu, Guishun Li, Haibin Li, Haijian Liang, Haipeng Ming, Haisu Wang, Haiyang Lu, Haiye Lin, Han Zhou, Hangting Lou, Hanwen Du, Hanzhi Zhang, Hao Chen, Hao Du, Hao Liu, Hao Zhou, Haochen Jiang, Haodong Tian, Haoshuang Wang, Haozhe Geng, Heju Yin, Hong Chen, Hongchen Xue, Hongen Liu, Honggeng Zhang, Hongji Xu, Hongwei Chen, Hongyang Zhang, Hongyuan Zhang, Hua Lu, Huan Chen, Huan Wang, Huang He, Hui Liu, Hui Zhong, Huibin Ruan, Jiafeng Lu, Jiage Liang, Jiahao Hu, Jiahao Hu, Jiajie Yang, Jialin Li, Jian Chen, Jian Wu, Jianfeng Yang, Jianguang Jiang, Jianhua Wang, Jianye Chen, Jiaodi Liu, Jiarui Zhou, Jiawei Lv, Jiaxin Zhou, Jiaxuan Liu, Jie Han, Jie Sun, Jiefan Fang, Jihan Liu, Jihua Liu, Jing Hu, Jing Qian, Jing Yan, Jingdong Du, Jingdong Wang, Jingjing Wu, Jingyong Li, Jinheng Wang, Jinjin Li, Jinliang Lu, Jinlin Yu, Jinnan Liu, Jixiang Feng, Jiyi Huang, Jiyuan Zhang, Jun Liang, Jun Xia, Jun Yu, Junda Chen, Junhao Feng, Junhong Xiang, Junliang Li, Kai Liu, Kailun Chen, Kairan Su, Kang Hu, Kangkang Zhou, Ke Chen, Ke Wei, Kui Huang, Kun Wu, Kunbin Chen, Lei Han, Lei Sun, Lei Wen, Linghui Meng, Linhao Yu, Liping Ouyang, Liwen Zhang, Longbin Ji, Longzhi Wang, Meng Sun, Meng Tian, Mengfei Li, Mengqi Zeng, Mengyu Zhang, Ming Hong, Mingcheng Zhou, Mingming Huang, Mingxin Chen, Mingzhu Cai, Naibin Gu, Nemin Qiu, Nian Wang, Peng Qiu, Peng Zhao, Pengyu Zou, Qi Wang, Qi Xin, Qian Wang, Qiang Zhu, Qianhui Luo, Qianwei Yang, Qianyue He, Qifei Wu, Qinrui Li, Qiwen Bao, Quan Zhang, Quanxiang Liu, Qunyi Xie, Rongrui Zhan, Rufeng Dai, Rui Peng, Ruian Liu, Ruihao Xu, Ruijie Wang, Ruixi Zhang, Ruixuan Liu, Runsheng Shi, Ruting Wang, Senbo Kang, Shan Lu, Shaofei Yu, Shaotian Gong, Shenwei Hu, Shifeng Zheng, Shihao Guo, Shilong Fan, Shiqin Liu, Shiwei Gu, Shixi Zhang, Shuai Yao, Shuang Zhang, Shuangqiao Liu, Shuhao Liang, Shuwei He, Shuwen Yang, Sijun He, Siming Dai, Siming Wu, Siyi Long, Songhe Deng, Suhui Dong, Suyin Liang, Teng Hu, Tianchan Xu, Tianliang Lv, Tianmeng Yang, Tianyi Wei, Tiezhu Gao, Ting Sun, Ting Zhang, Tingdan Luo, Wei He, Wei Luan, Wei Yin, Wei Zhang, Wei Zhou, Weibao Gong, Weibin Li, Weicheng Huang, Weichong Dang, Weiguo Zhu, Weilong Zhang, Weiqi Tan, Wen Huang, Wenbin Chang, Wenjing Du, Wenlong Miao, Wenpei Luo, Wenquan Wu, Xi Shi, Xi Zhao, Xiang Gao, Xiangguo Zhang, Xiangrui Yu, Xiangsen Wang, Xiangzhe Wang, Xianlong Luo, Xianying Ma, Xiao Tan, Xiaocong Lin, Xiaofei Wang, Xiaofeng Peng, Xiaofeng Wu, Xiaojian Xu, Xiaolan Yuan, Xiaopeng Cui, Xiaotian Han, Xiaoxiong Liu, Xiaoxu Fei, Xiaoxuan Wu, Xiaoyu Wang, Xiaoyu Zhang, Xin Sun, Xin Wang, Xinhui Huang, Xinming Zhu, Xintong Yu, Xinyi Xu, Xinyu Wang, Xiuxian Li, XuanShi Zhu, Xue Xu, Xueying Lv, Xuhong Li, Xulong Wei, Xuyi Chen, Yabing Shi, Yafeng Wang, Yamei Li, Yan Liu, Yanfu Cheng, Yang Gao, Yang Liang, Yang Wang, Yang Wang, Yang Yang, Yanlong Liu, Yannian Fu, Yanpeng Wang, Yanzheng Lin, Yao Chen, Yaozong Shen, Yaqian Han, Yehua Yang, Yekun Chai, Yesong Wang, Yi Song, Yichen Zhang, Yifei Wang, Yifeng Guo, Yifeng Kou, Yilong Chen, Yilong Guo, Yiming Wang, Ying Chen, Ying Wang, Yingsheng Wu, Yingzhan Lin, Yinqi Yang, Yiran Xing, Yishu Lei, Yixiang Tu, Yiyan Chen, Yong Zhang, Yonghua Li, Yongqiang Ma,

Yongxing Dai, Yongyue Zhang, Yu Ran, Yu Sun, Yu-Wen Michael Zhang, Yuang Liu, Yuanle Liu, Yuanyuan Zhou, Yubo Zhang, Yuchen Han, Yucheng Wang, Yude Gao, Yuedong Luo, Yuehu Dong, Yufeng Hu, Yuhui Cao, Yuhui Yun, Yukun Chen, Yukun Gao, Yukun Li, Yumeng Zhang, Yun Fan, Yun Ma, Yunfei Zhang, Yunshen Xie, Yuping Xu, Yuqin Zhang, Yuqing Liu, Yurui Li, Yuwen Wang, Yuxiang Lu, Zefeng Cai, Zelin Zhao, Zelun Zhang, Zenan Lin, Zezhao Dong, Zhaowu Pan, Zhaoyu Liu, Zhe Dong, Zhe Zhang, Zhen Zhang, Zhengfan Wu, Zhengrui Wei, Zhengsheng Ning, Zhenxing Li, Zhenyu Li, Zhenyu Qian, Zhenyun Li, Zhi Li, Zhichao Chen, Zhicheng Dong, Zhida Feng, Zhifan Feng, Zhihao Deng, Zhijin Yu, Zhiyang Chen, Zhonghui Zheng, Zhuangzhuang Guo, Zhujun Zhang, Zhuo Sun, Zichang Liu, Zihan Lin, Zihao Huang, Zihe Zhu, Ziheng Zhao, Ziping Chen, Zixuan Zhu, Ziyang Xu, Ziyi Liang, and Ziyuan Gao. Ernie 5.0 technical report, 2026. URL https://arxiv.org/abs/2602.04705.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In Leon Derczynski, Wei Xu, Alan Ritter, and Tim Baldwin (eds.), *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106, Copenhagen, Denmark, 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4413. URL https://aclanthology.org/W17-4413.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *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.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Y. Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. Mixture-of-experts with expert choice routing. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Table 5: Architectural comparison of MoEUT (Csordás et al., 2024) and MoUE.

Design choice	MoEUT	MoUE
Shared computation	Repeated groups of layers	Routed FFN experts accessible across layers
Attention	SwitchHead MoE attention within repeated groups	Layer-local attention retained
Normalization	Peri-layernorm designed for recurrent residual scaling	Layer-local RMSNorm retained
FFN routing	Sigmoid Top-k in σ-MoE	Top-k over topology-masked expert candidates
Balancing	Entropy regularization of expert selection	UELB accounts for cross-layer expert exposure
Reuse pattern	Recurrently stacked groups of distinct layers	Configurable expert connectivity, including SRT
Extensions studied	Sparse shared-layer Transformer	Checkpoint conversion; optional contextual UR

A Related Architectures

Sparse MoE models use conditional expert selection to scale stored model parameters while evaluating a subset for each token (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022). Expert granularity and routing design provide additional choices in this parameter organization (Dai et al., 2024; Ludziejewski et al., 2024).

MoEUT (Csordás et al., 2024) combines FFN and attention experts within shared layer groups, together with grouping and normalization for recurrent computation. MoUE exposes routed FFN experts across layers while retaining layer-local attention and normalization. Table 5 summarizes the architectural components.

Shared Unit and Recurrence. MoEUT first constructs a group of distinct layers, each with its own parameters, and then recurrently stacks this group. Corresponding layer positions share parameters across repetitions: a two-layer group follows an A–B–A–B pattern. In MoUE, a connectivity group instead specifies which consecutive layers can access the same routed FFN experts. The attention, normalization and residual pathways remain layer-local, while the accessible expert set is specified by the sharing topology. These two uses of grouping distinguish repeated layer blocks from cross-layer access to expert operators.

Attention Experts and Normalization. MoEUT uses σ-MoE feedforward blocks and SwitchHead attention. In each attention head, SwitchHead selects value and output projections from expert sets, while keeping one query and one key projection. MoEUT also introduces peri-layernorm for recurrent residual propagation: normalization is placed on the query/key and expert-selection pathways and before the final classifier, while the attention-value and feedforward main data paths remain unnormalized. A complete MoEUT implementation therefore includes its attention experts, FFN experts, repeated layer groups and normalization scheme.

Relation to the MoUE Controls. MoUE studies expert reuse within a standard MoE backbone, extending the set of FFN experts accessible to each layer and optimizing their use through connectivity and load balancing. The standard-LBL shared-expert control keeps the MoUE backbone, SRT and router fixed and uses layer-local load balancing. Its PPL of 26.37, compared with 25.40 for the full configuration, reports the effect of replacing the balancing objective within this architecture. The architectural comparison with MoEUT concerns the shared computation unit, attention design, normalization and recurrence pattern listed in Table 5.

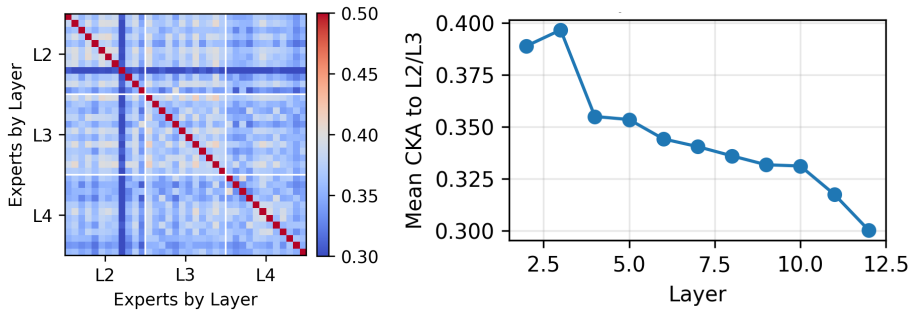


Figure 8: Expert-weight CKA in the analyzed MoE. Left: cross-layer similarity. Right: mean CKA by layer.

B Expert Similarity and Parameter Organization

B.1 Expert-Weight Similarity

Expert-weight CKA describes similarity between parameter representations. The analyzed model exhibits relatively high similarities across nearby layers and some distant layer pairs, motivating cross-layer expert reuse. For centered matrices X and Y with corresponding rows, linear CKA is

$$\mathrm{CKA}(X,Y) = \frac{\|Y^\top X\|_F^2}{\|X^\top X\|_F \, \|Y^\top Y\|_F}. \tag{5}$$

Within the L2–L4 window, many expert pairs have CKA above 0.4; the displayed long-range comparisons also include values above 0.3.

B.2 Physical, Virtual and Activated Parameters

Let $\mathcal{P}$ be the set of independently stored expert parameter objects, $\mathcal{A}_\ell$ the expert identities accessible at layer ℓ, and $S_\ell(x)$ the experts selected for token x. If expert e contains p_e parameters, the resource counts are

$$\begin{aligned} \mathrm{TP} &= P_{\mathrm{other}} + \sum_{e\in\mathcal{P}} p_e, \\ \mathrm{VP} &= P_{\mathrm{other}} + \sum_{\ell=1}^{L} \sum_{e\in\mathcal{A}_\ell} p_e, \\ \mathrm{Act}(x) &= P_{\mathrm{other,act}}(x) + \sum_{\ell=1}^{L} \sum_{e\in S_\ell(x)} p_e. \end{aligned} \tag{6}$$

P_{other} contains the non-expert parameters under the model's counting convention. A shared expert contributes once to TP and once per accessible layer to VP. Each selected invocation contributes to Act, including repeated invocations of the same weights. For equal-size experts, fixed depth L and Top-k, routed Act is Lkp. A bias-free gated FFN with three matrices has $p = 3df$, where d and f are its input and intermediate dimensions.

The nominal 0.16B-total / 0.1B-active model includes approximately 77M embedding parameters, 17.7M attention parameters and 10.7M activated MoE parameters in its full-model active count. Table 6 gives the routed-FFN counts for both backbones. Table 7 gives the checkpoint-adaptation budgets.

B.3 Expert Selection Sequences

At depth L, an access graph with per-layer candidate sets $\mathcal{A}_\ell$ and Top-k admits

$$\prod_{\ell=1}^{L} \binom{|\mathcal{A}_\ell|}{k} \tag{7}$$

Table 6: Routed-FFN parameter accounting for the listed backbone dimensions. A bias-free gated FFN with three weight matrices has $p = 3df$ parameters. Counts are in millions. Width expansion holds the stored expert pool, depth, and Top-k fixed.

Quantity	MoE-160M family	MoE-700M family
Depth L / baseline width / Top-k	$12/12/2$	$16/64/8$
Hidden size d / expert size f	$768/192$	$1536/96$
Parameters per expert p (M)	0.4424	0.4424
Stored routed-FFN TP (M)	63.7010	452.9848
Routed Act per token Lkp (M)	10.6168	56.6231
Routed VP at baseline width (M)	63.7010	452.9848
Routed VP at $1.5\times$ width (M)	95.5515	679.4772
Routed VP at $2\times$ width (M)	127.4020	905.9697
Routed VP at $3\times$ width (M)	191.1030	1,358.9545

Table 7: Checkpoint-adaptation budgets at the reporting precision of Table 10. Expert access expands through shared FFN weights; TP matches the source model and adapters are disabled.

Component	Experimental budget / counting rule
JetMoE SFT: TP / Act	8.0B / 2.2B for baseline and all variants
OLMoE CPT and SFT: TP / Act	6.9B / 1.3B for baseline and all variants
Shared expert FFN weights	Existing modules are referenced; no additional copies
Adapters	None
Expert evaluations per token	Lk at fixed depth and Top-k
Virtual expert parameters	Count each expert once per layer that can access it
Routing state	Forward state, separate from stored model parameters; no UR in CPT/SFT

sequences of selected expert subsets. This expression describes the choices made available by the connectivity graph.

C Topology Configurations

A connectivity group contains consecutive layers with a shared expert pool. Let

$$g(\ell) = \left\lfloor \frac{\ell - 1}{G} \right\rfloor, \qquad \mathcal{A}_\ell = \mathcal{E}_\ell^{\text{local}} \cup \mathcal{U}_{g(\ell)}. \tag{8}$$

The layers of a group access the same fixed shared candidates and retain their private expert sets. The primary configuration uses $G = 5$. Stable expert IDs and per-layer allow-lists specify the access graph, and unreachable experts are masked before Top-k selection. The final group uses its actual number of layers.

Table 8 gives the connectivity results. The all-to-all setting exposes the full shared pool. Forward-window, reverse-order and sandwich connections specify alternative access patterns.

D Grouped Routing Statistics

Let $\mathcal{G}_g = \{\ell : g(\ell) = g\}$ be the layers in group g. For shared experts, token assignments and masked routing probabilities are accumulated across the group's token–layer events $\mathcal{I}_g$. With $T_g = |\mathcal{I}_g|$,

$$f_{g,e} = \frac{1}{kT_g} \sum_{r \in \mathcal{I}_g} \mathbb{I}[e \in S_r], \qquad \bar{P}_{g,e} = \frac{1}{T_g} \sum_{r \in \mathcal{I}_g} P_{r,e}. \tag{9}$$

The discrete assignment count is fixed when differentiating the auxiliary product. $P_{r,e}$ is the routing probability before selected-weight renormalization.

Group aggregation and inverse-exposure weighting have different effects. When each layer contributes an equal number of routing events, pooling before multiplication introduces products of statistics

Table 8: Validation perplexity for the fixed-group SRT configuration and four connectivity alternatives. The comparisons use matched training tokens and activated-parameter budgets; full all-to-all access exposes the complete expert pool.

Topology Scheme	Perplexity ↓
SRT: fixed group sharing (Ours)	**25.40**
Forward Sliding Window	25.87 (+0.47)
Reverse-Order Connection	25.93 (+0.53)
Sandwich Architecture	25.61 (+0.21)
Full All-to-All (Unconstrained)	27.94 (+2.54)

Table 9: Component ablations on MoUE 36A2 at matched training-token and activated-parameter budgets. Entries give validation perplexity and the increase from the full model. The contextual-pathway control disables the state and update; the frozen-update control retains random state with $\eta = 0$.

Method	Perplexity ↓
MoUE 36A2 (full model)	25.40
Connectivity / Expert Topology	
w/o Staggered Rotational Topology	27.94 (+2.54)
w/o Universal Experts	26.05 (+0.65)
Routing Mechanism	
w/o Contextual Pathway	25.83 (+0.43)
w/o Fast-Weight Update	25.59 (+0.19)
Load Balancing Strategy	
w/o UELB (standard load balance)	26.37 (+0.97)
w/o Connectivity Normalization	25.75 (+0.35)
w/o Probabilistic Warmup	25.66 (+0.26)
MoE 12A2 (iso-activate / full parameter)	26.25 (+0.85)

from different layers:

$$f_{g,e}\bar{P}_{g,e} = \frac{1}{m_g^2}\sum_{\ell\in\mathcal{G}_g}\sum_{j\in\mathcal{G}_g} f_{\ell,e}\bar{P}_{j,e}, \qquad m_g = |\mathcal{G}_g|. \tag{10}$$

For shared experts with common exposure c_g within a group, the inverse-exposure factor scales the group's shared product by α_{u}/c_g. Group aggregation determines the cross-layer statistical coupling, while the exposure factor sets the corresponding weight.

Exploration Schedule. During early from-scratch training, a bias increases the probability of selecting universal experts and is annealed away. Multiplying their pre-softmax mass by $\rho(t)$ is equivalent to adding $b(t) = \log \rho(t)$ to their logits. The initial universal bias is 0.75, and the schedule covers the first 5% of training tokens.

E Router Controls and Component Results

Each shared expert pool has a randomly initialized routing state with a stable expert-ID ordering. Layers accessing the same pool share the state. The semantic pathway computes affine routing scores, and the contextual pathway uses the projected input key and the shared state.

The frozen-update control sets the delta step size to zero and retains the contextual term produced by the random initial state. The stateless control removes the contextual term. The contextual score remains input-dependent through the key projection. The frozen-update and stateless configurations have PPL 25.59 and 25.83, respectively; the full configuration reaches 25.40.

F CHECKPOINT ADAPTATION

Shared Expert Access. The universal pool is constructed by randomly selecting source experts and sharing references to their trained FFN weights. Source layers retain their local expert access. A negative routing bias initially suppresses the additional universal branch and is gradually removed during continued training. Algorithm 1 gives the procedure.

Algorithm 1 Progressive Warm-Start

1: **Input:** pretrained MoE; connectivity groups and shared-pool sizes; bias schedule $\beta(t)$
2: Randomly select source experts and expose references to their trained FFN weights
3: Preserve the local router and construct the universal routing projection
4: **for** each training step t **do**
5: Form local routing logits and universal logits with suppression $-\beta(t)$
6: Apply connectivity masks and select Top-k experts
7: Evaluate selected experts and optimize the language-model and auxiliary objectives
8: **end for**
9: **Output:** adapted MoUE checkpoint

Data and Evaluation. OLMoE CPT uses `dolmino-mix-1124`. Supervised fine-tuning uses `tulu-v3.1-mix-preview-4096-OLMoE`. Within each adaptation setting, the baseline and converted models use the same training budget and evaluation protocol. The experiments use ordinary routing, with no UR or adapters. TP remains 8.0B for the JetMoE family and 6.9B for the OLMoE family at the reported precision.

Table 10 lists all twelve configurations and their nine-task results. Evaluation uses OpenCompass generation and answer extraction. AVG is the mean over the nine tasks shown in the table.

Table 10: Complete OpenCompass generative evaluation for JetMoE SFT and OLMoE SFT/CPT. TP and Act are in billions, and TP matches the source model within each family. The experiments use shared expert weights without adapters. AVG is the aggregate over the nine listed tasks.

Model	Act.	TP.	VP.	ARC-c	ARC-e	GSM8K	HellaS.	HumanE.	NQ	Tri.QA	Wino.	MMLU	AVG
JetMoE-8E, adapted by supervised fine-tuning.													
Vanilla MoE	2.2B	8.0B	8.0B	64.4	79.0	44.7	78.7	42.7	20.4	49.5	56.6	47.7	53.7
MoUE 10A2	2.2B	8.0B	9.7B	66.8	78.8	42.2	78.4	48.2	20.9	50.7	54.1	46.3	54.1
MoUE 12A2	2.2B	8.0B	11.3B	65.1	79.4	45.1	80.3	42.7	21.4	50.5	56.8	48.2	54.4
MoUE 16A2	2.2B	8.0B	14.6B	65.8	79.9	45.0	79.5	45.7	20.5	51.2	57.9	47.9	54.8
OLMoE-64E, adapted by supervised fine-tuning.													
Vanilla MoE	1.3B	6.9B	6.9B	57.0	76.7	36.3	62.1	24.2	18.1	45.3	52.3	47.1	46.5
MoUE 68A8	1.3B	6.9B	7.3B	61.0	79.2	34.0	59.4	27.4	18.1	44.1	54.4	47.3	47.2
MoUE 72A8	1.3B	6.9B	7.7B	61.4	79.5	35.5	58.7	29.3	18.0	44.5	56.5	46.0	47.7
MoUE 76A8	1.3B	6.9B	8.1B	62.7	79.9	34.0	62.1	27.4	19.1	47.6	55.3	47.7	48.4
OLMoE-64E, adapted by continued pre-training.													
Vanilla MoE	1.3B	6.9B	6.9B	37.3	40.2	31.0	26.3	13.4	9.6	18.4	49.1	46.3	30.2
MoUE 68A8	1.3B	6.9B	7.3B	43.7	45.9	33.5	26.2	14.0	9.3	22.0	49.9	47.5	32.4
MoUE 72A8	1.3B	6.9B	7.7B	42.0	48.9	30.0	29.4	13.4	10.7	30.2	48.5	50.4	33.7
MoUE 76A8	1.3B	6.9B	8.1B	50.9	58.2	31.5	26.4	14.6	8.0	22.9	50.4	47.1	34.4

F.1 POOL SIZE AND EXPERT USE

The pool-size sweep improves up to twelve shared experts per layer and decreases at the larger tested setting. Figure 10 shows expert-selection frequencies across domains, and Figure 11 shows the universal-expert routing proportion by layer after SFT.

G TRAINING SETTINGS AND COMPLETE SCALING RESULTS

The from-scratch experiments use Qwen-3 style backbones. MoE-160M has 12 layers, hidden size 768, 12 attention heads, four query groups, 12 experts per layer, Top-2, and expert intermediate size

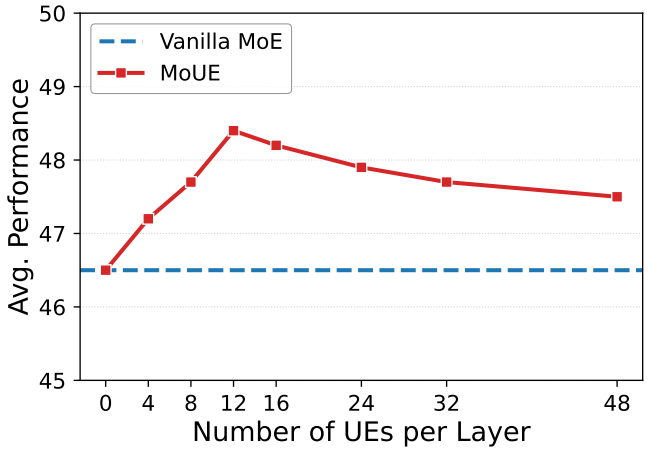


Figure 9: Performance versus the number of shared experts in checkpoint adaptation.

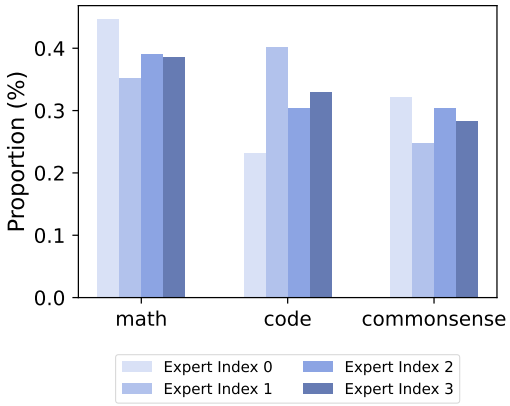


Figure 10: Universal-expert selection across domains, with four shared experts per layer.

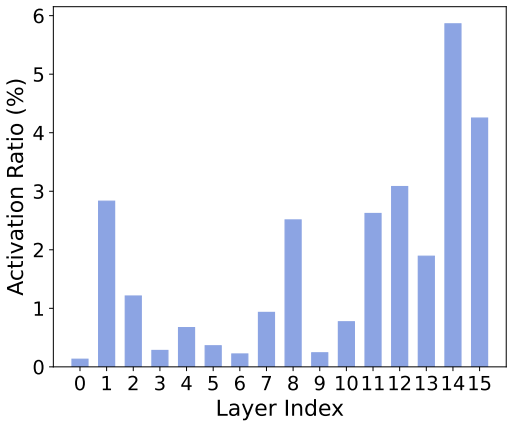


Figure 11: Layer-wise universal-expert routing proportion after SFT.

192. MoE-700M has 16 layers, hidden size 1536, 16 attention heads, four query groups, 64 experts per layer, Top-8, and expert intermediate size 96.

The data-to-parameter ratio is $\mathrm{DN} = D/\mathrm{TP} = 200$, where D is the training-token count and TP is the total physical parameter count. Width expansion changes accessible experts at the reported fixed TP and Act. The depth series uses additional computational layers with shared FFNs. Table 11 summarizes the training settings, and Table 12 gives all fourteen configurations.

Evaluation Tasks. From-scratch evaluation uses `lm-evaluation-harness` (Gao et al., 2024): 0-shot SciQ (Welbl et al., 2017), PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2020), ARC-E (Clark et al., 2018), LogiQA (Liu et al., 2020), and LAMBADA (Paperno et al., 2016); 10-shot HellaSwag (Zellers et al., 2019); 25-shot ARC-C (Clark et al., 2018); and 32-shot BoolQ (Clark et al., 2019).

H Evaluation Statistics and Protocols

The from-scratch uncertainty values are standard errors (SE) from four evaluations of fixed pretrained models. The OpenCompass SFT values are standard deviations (SD) across independent generative evaluations. All dispersion values are expressed in percentage points.

Table 11: Training and routing hyperparameters used in our experiments. Values are reported for the Qwen-3 style MoE backbones and are kept identical between MoE and MoUE unless explicitly noted.

Hyperparameter	Value	Notes
Sequence length	4096	shared
Global batch size	256	shared
Optimizer	AdamW	$(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.1
LR schedule	cosine	shared
Peak LR	5×10^{-4}	shared
Warmup fraction	5% tokens	shared
MoE aux loss coef.	10^{-3}	MoE baseline
DN $= D/\mathrm{TP}$	200	total physical parameters
Connectivity group size G	5	consecutive layers; fixed shared pool
Sharing window	fixed	no sliding-window stride
MoE-160M: layers	12	Qwen-3 style
MoE-160M: hidden size	768	Qwen-3 style
MoE-160M: heads / groups	$12/4$	attention / query groups
MoE-160M: experts / top-k	$12/2$	per layer
MoE-160M: MoE FFN hidden	192	per expert
MoE-700M: layers	16	Qwen-3 style
MoE-700M: hidden size	1536	Qwen-3 style
MoE-700M: heads / groups	$16/4$	attention / query groups
MoE-700M: experts / top-k	$64/8$	per layer
MoE-700M: MoE FFN hidden	96	per expert
UELB α_{loc}	1.0	local-branch weight
UELB α_{u}	0.5	shared-branch weight
UELB warmup b_0	0.75	initial universal logit bias
UELB warmup r	5%	token %
Universal Router β	0.1	contextual pathway weight
Fast-weight lr η	0.1	online update learning rate
Frozen-update control η	0	random state retained; updates disabled
UR state initialization	random	state shared according to expert pool
From-scratch evaluation repeats	4	standard error (SE) reported
Adapters in CPT/SFT	none	source TP budget retained

Table 12: Complete results for width and depth expansion on the MoE-160M and MoE-700M backbones. Virtual-width and depth multipliers are 1.5, 2, and 3. Act, TP, and VP are reported in billions; task scores and their average are percentages.

Model	Act.	TP.	VP.	Commonsense & Reading Comprehension						Continued		LM	Avg.
				SciQ	PIQA	WG	ARC-E	ARC-C	Hella.	LogiQA	BoolQ	Lam.	
MoE 12A2 L12	0.1B	0.16B	0.16B	52.0	62.9	49.4	41.3	19.9	30.1	19.7	57.3	36.4	41.0
Width scaling													
MoUE 18A2	0.1B	0.16B	0.19B	52.6	62.6	50.2	41.1	21.6	30.1	21.4	57.6	36.2	41.5
MoUE 24A2	0.1B	0.16B	0.22B	52.5	63.3	50.9	41.8	21.6	29.3	19.7	59.6	36.1	41.6
MoUE 36A2	0.1B	0.16B	0.28B	52.6	63.3	51.2	41.4	20.4	31.0	23.2	60.1	37.1	42.3
Depth scaling													
MoUE L18	0.12B	0.16B	0.2B	52.8	62.4	52.2	40.0	22.2	30.4	24.6	60.9	37.4	42.5
MoUE L24	0.13B	0.18B	0.24B	55.1	62.8	52.8	39.0	23.2	30.7	25.8	60.3	38.8	43.2
MoUE L36	0.16B	0.19B	0.32B	56.0	63.7	53.7	39.9	24.1	31.5	26.7	61.2	39.6	44.0
MoE 64A8 L16	0.3B	0.7B	0.7B	56.2	65.5	51.5	43.7	22.1	32.9	23.0	57.6	41.8	43.8
Width scaling													
MoUE 96A8	0.3B	0.7B	0.9B	56.7	65.5	51.6	43.2	22.4	33.1	23.8	57.7	42.6	44.1
MoUE 128A8	0.3B	0.7B	1.16B	57.2	65.1	51.7	44.7	22.7	32.4	24.5	57.8	43.4	44.4
MoUE 192A8	0.3B	0.7B	1.61B	57.8	64.7	52.8	44.4	23.1	32.8	25.3	57.9	43.3	44.7
Depth scaling													
MoUE L24	0.38B	0.75B	0.98B	57.2	65.3	52.9	42.7	23.8	33.2	26.4	59.3	42.8	44.8
MoUE L32	0.46B	0.8B	1.25B	57.2	65.6	53.6	43.2	24.8	33.5	27.8	60.2	43.5	45.5
MoUE L48	0.61B	0.9B	1.8B	58.3	65.2	54.3	44.0	25.8	33.9	28.8	61.0	44.2	46.2

Table 15 gives evaluations of two public checkpoints under discriminative `lm-eval` scoring and generative OpenCompass evaluation.

Table 13: Evaluation uncertainty in percentage points. The from-scratch block reports standard errors (SE) from four evaluations of fixed pretrained models. The OpenCompass SFT block reports standard deviations (SD) across independent generative evaluations.

Task	Baseline	SE	MoUE	SE
From scratch: MoE 12A2 L12 / MoUE 36A2				
SciQ	52.0	0.652	52.6	0.669
PIQA	62.9	0.973	63.3	0.996
WinoGrande	49.4	0.127	51.2	0.130
ARC-E	41.3	0.828	41.4	0.847
ARC-C	19.9	0.145	20.4	0.145
HellaSwag	30.1	0.461	31.0	0.474
LogiQA	19.7	0.165	23.2	0.169
BoolQ	57.3	0.863	60.1	0.857
LAMBADA	36.4	0.641	37.1	0.658
OpenCompass SFT: OLMoE / MoUE 76A8				
Task	Baseline	SD	MoUE	SD
ARC-C	57.0	0.416	62.7	0.436
HellaSwag	62.1	1.061	62.1	0.945
WinoGrande	52.3	0.361	55.3	0.566
MMLU	47.1	0.409	47.7	0.312
GSM8K	36.3	0.600	34.0	0.775

Table 14: Loss trajectories for a MoE baseline and two MoUE 24A2 training runs with seeds 36 and 42. Columns give the training-token checkpoints.

Model / training tokens	5B	10B	20B	30B
MoE baseline	3.526	3.413	3.324	3.271
MoUE 24A2, seed 36	3.519	3.408	3.306	3.257
MoUE 24A2, seed 42	3.518	3.407	3.305	3.256

Table 15: Public-checkpoint scores under discriminative lm-eval and generative OpenCompass evaluation. Scores are percentages.

Protocol	MMLU	ARC-E	HellaSwag	WinoGrande	ARC-C
OLMoE-1B-7B-0924					
lm-eval	50.42	78.16	58.02	68.35	46.50
OpenCompass	19.48	31.22	25.25	47.99	26.10
DeepSeek-V2-Lite					
lm-eval	55.65	88.53	58.60	70.88	46.76
OpenCompass	46.81	49.03	25.00	51.38	38.31

Five-Task Aggregates. For ARC-C, HellaSwag, WinoGrande, MMLU and GSM8K, OLMoE CPT averages 38.00 for the baseline and 41.26 for MoUE. The corresponding SFT averages are 50.96 and 52.36.

I LARGE-SCALE TRAINING RESULTS

Across 406 common steps near 419.430B nominal token positions, average training loss decreases from 1.650948 to 1.642122, a reduction of 0.535%. Average iteration time changes from 2.694676 to 2.952326 seconds, an increase of 9.56%. The MoUE checkpoint contains 1.5 times the baseline's distinct routed-expert content.

Table 16: Large-scale operational results on 256 GPUs with Top-8 routing. Training loss and step time are averaged over 406 common steps near 419.430B nominal training-token positions. Serialized tensor entries are counted from checkpoint tensor shapes. Routed content is normalized to the MoE baseline.

Model	Serialized entries (B)	Routed content	Training loss	Step time (s)
MoE	35.859	1.0×	1.650948	2.694676
MoUE	57.565	1.5×	1.642122	2.952326

J LLM USAGE

AI tools assisted with manuscript organization, language editing, LaTeX preparation, and analysis of experimental data.