RETHINKING ROUTING EXPLORATION IN MODERN MIXTURE-OF-EXPERTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixture-of-experts (MoE) models have become an important approach for scaling model parameters. Modern MoE architectures increasingly combine two design choices: fine-grained expert segmentation and shared experts. The former increases expressiveness by enabling more expert combinations at a fixed activation ratio, and the latter introduces experts that are active for every token. We identify an *exploration dilemma* between these designs: shared experts can make routing less exploratory by reducing the variability of token representations. This conflicts with the greater need for exploration induced by fine-grained expert segmentation due to the combinatorially enlarged routing space. Consequently, the additional expressiveness offered by fine-grained segmentation may not be fully realized in practice. MirrorMoE is proposed to reconcile this exploration dilemma by shrinking the exploration space through token-adaptive suppression of unnecessary experts. MirrorMoE consistently improves performance across model scales.

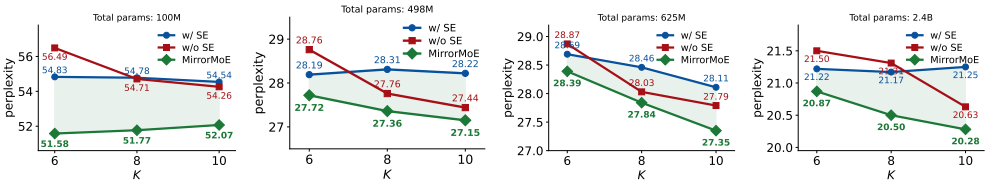
1 INTRODUCTION

Mixture-of-experts (MoE) models have been instrumental in scaling large language models (LLMs) to trillions of parameters and beyond (Lepikhin et al., 2020; Fedus et al., 2022). MoEs adopt a sparse conditional-computation paradigm, where each token activates only a subset of experts, or subnetworks. During pretraining, recent work has shown that MoEs can outperform their dense counterparts under matched total parameter counts, compute, and data budgets (Li et al., 2026). At inference, sparse activation allows computational cost to grow sublinearly with total parameters. Thanks to these advantages, MoEs have become a core architecture underlying model families such as DeepSeek (Dai et al., 2024; Liu et al., 2024), Kimi (Team-Kimi, 2025; 2026), and Qwen (Team-Qwen, 2024; Yang et al., 2025), among many others (Jiang et al., 2024; Abdin et al., 2024).

At the core of MoEs is the MoE layer, which consists of a router and E experts, each implemented as a feedforward network (FFN). For each input token, the router selects K experts ($K < E$) for sparse activation (Lepikhin et al., 2020; Fedus et al., 2022). Building on this basic design, recent MoEs have increasingly adopted two notable architectural advances. The first is *fine-grained expert segmentation* (Team-Kimi, 2026; Liu et al., 2024). Recent MoEs tend to use larger numbers of both total and activated experts, i.e., larger E and K. For example, at the same activation ratio K/E, $(E, K) = (64, 8)$ is often preferred over $(E, K) = (8, 1)$. Since there are $\binom{E}{K}$ expert combinations, the latter yields richer routing space for more expressive token representations (Elango et al., 2026). The second advance is *shared experts*. Activated for every token, shared experts capture common knowledge that would otherwise be learned redundantly by routed experts. This allows routed ones to devote more capacity to specialization. The combination of the above architectural choices have become standard in modern MoEs (Dai et al., 2024; Liu et al., 2024; Team-Qwen, 2024).

This work reexamines the interaction between these two trends from the angle of *routing exploration*. The goal of exploration is to search for alternative expert combinations during pretraining to identify (near-)optimal routes for each token. On the one hand, realizing the expressive potential of fine-grained expert segmentation requires exploring a *combinatorially* larger routing space within each MoE layer. For example, the number of expert subsets increases from 8 when $(E, K) = (8, 1)$ to $\binom{64}{8} \approx 4.4 \times 10^9$ with $(E, K) = (64, 8)$, i.e., several orders of magnitude increase. On the other hand, our analysis reveals an asymmetry during pretraining: shared experts dominate the learned token representations, leaving routed experts with less influence on the layer output. Consequently,

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.



(a) 100M (b) 498M (c) 625M (d) 2.4B

Figure 1: Comparison of MoEs with and without shared experts (SE) and the proposed MirrorMoE. MirrorMoE consistently i) achieves the best performance, ii) realizes the expressiveness gains from larger K, and iii) enables shared-expert MoEs to outperform their non-shared counterparts.

even activating different routed experts only leads to limited variation in the learned features, making alternative routes harder to distinguish and thereby undermining the goal of exploration.

This exploration dilemma has tangible consequences for shared-expert MoEs. To see this, we train matched MoE models with and without shared experts at 4 scales, ranging from 100M to 2.4B total parameters. At each scale, we fix the total number of experts E and vary the number of (total) activated experts K over $\{6, 8, 10\}$. A larger K simultaneously increases model expressiveness and demands more strategic exploration due to enlarged routing space. As shown in Fig.1, shared experts no longer provide benefits over their non-shared counterparts as K increases. More strikingly, the performance of shared-expert MoEs can even degrade as K increases in Figs.1b and 1d, suggesting that the inefficient exploration hinders realization of the additional expressiveness from larger K.

MirrorMoE is proposed to reconcile the conflicting architecture designs above. It adaptively suppresses routed experts on a token basis, reducing the exploration space to better match the model's exploration capability. Technically, MirrorMoE is derived by mirroring the routing logits via sign negation, hence its name. As shown in Fig.1, MirrorMoE improves the performance of shared-expert MoEs and attains expressiveness gains from larger K. In summary, our contributions are as follows:

- ❖ We identify an **exploration dilemma** in modern MoEs: the enlarged routing space induced by fine-grained expert segmentation demands greater exploration, while shared experts make routing less explorative. As a result, the benefits of shared experts diminish as the routing space grows.
- ❖ We propose **MirrorMoE**, which adaptively shrinks the exploration space to reconcile competing demands above. MirrorMoE is conceptually simple, and its adaptivity is theoretically characterized.
- ❖ We demonstrate that MirrorMoE addresses the exploration dilemma across MoEs of different scales. On the 2.4B model, MirrorMoE improves pretraining and downstream performance by 0.67 and 1.64 points, respectively. Moreover, it enables shared-expert MoEs to consistently outperform their non-shared counterparts and restores the performance gains from increasing K; see Fig.1.

1.1 Related Work

MoEs. Modern sparse MoEs originated with the sparsely gated MoE layer of Shazeer et al. (2017) and were extended to large-scale Transformers in e.g., Lepikhin et al. (2020); Fedus et al. (2022). Later work focused on improving scalability, stability, and expert specialization. GLaM (Du et al., 2022) demonstrated that sparse language models could scale beyond one trillion parameters, while ST-MoE (Zoph et al., 2022) improved training stability. MoEs have since become widely adopted. Beyond those mentioned earlier, other representative examples include Gemma4, OLMoE and gpt-oss (Team-Gemma, 2026; Muennighoff et al., 2025; Agarwal et al., 2025). Across this development trajectory, two trends have clearly emerged: fine-grained expert segmentation and shared experts.

Tension between expressiveness and pretraining. A growing line of work shows that a model's expressive capacity may not be readily realized without carefully designed training procedures (Lion et al., 2026; Zhang et al., 2026; Jha & Reagen, 2026; Wei et al., 2025a). Within the context of MoEs, related evidence is provided by Jin et al. (2024), which augments learnable experts with fixed zero, copy, and constant experts. Although standard FFN experts can, in principle, learn these operations, explicitly incorporating them still improves empirical performance. The benefit of directly defining these operations, rather than learning them through training, suggests that the available expressiveness may not be fully realized in practice. Our work further shows that the additional expressiveness provided by fine-grained expert segmentation is not naturally attained in shared-expert MoEs.

Additional related work on MoE routing and systems is deferred to App.A due to limited space.

2 The Exploration Dilemma in Shared-Expert MoEs

This section first reviews MoEs and then examines the effect of shared experts on exploration.

2.1 MoE recap

MoEs replace the FFN in each Transformer layer with a collection of FFNs, each referred to as an *expert*. Among E experts at layer l, there are two types: S shared experts $\{\mathcal{SE}_{l,s}\}_s$, activated for every token (Liu et al., 2024), and $R{:=}E{-}S$ routed experts $\{\mathcal{RE}_{l,e}\}_e$, activated in a token-driven manner. We use K to denote the total number of activated experts, including the shared ones.

For $\mathbf{h}_l^n \in \mathbb{R}^d$, the representation of the n-th token entering the l-th MoE layer, the routed experts to be activated are selected by a linear projection $\mathbf{G}_l \in \mathbb{R}^{R\times d}$, known as the *router*. The router produces a probability distribution over the routed experts, and those with the highest probabilities are activated. The activated routed experts are collected in $\mathcal{A}_l(\mathbf{h}_l^n)$:

$$p_{l,e}(\mathbf{h}_l^n) = \frac{\exp(\mathbf{G}_l[e,:]\mathbf{h}_l^n)}{\sum_{j=1}^{R}\exp(\mathbf{G}_l[j,:]\mathbf{h}_l^n)}, \quad \mathcal{A}_l(\mathbf{h}_l^n) = \mathsf{Top}_{K-S}\big(\{p_{l,e}(\mathbf{h}_l^n)\}, e\in[R]\big). \tag{1}$$

Here, $\mathsf{Top}_{K-S}(\cdot)$ deterministically selects $K-S$ experts with the largest probabilities and $[R] := \{1,\dots,R\}$ denotes all the routed experts. The learned token representation of this layer combines the outputs of both shared and routed experts, where shared experts capture common features and routed experts learn specialized ones, i.e.,

$$\mathbf{h}_{l+1}^n = \sum_{s=1}^{S}\mathcal{SE}_{l,s}(\mathbf{h}_l^n) + \sum_{e\in\mathcal{A}_l(\mathbf{h}_l^n)} \frac{p_{l,e}(\mathbf{h}_l^n)}{\sum_{j\in\mathcal{A}_l(\mathbf{h}_l^n)} p_{l,j}(\mathbf{h}_l^n)} \cdot \mathcal{RE}_{l,e}(\mathbf{h}_l^n). \tag{2}$$

Vanilla MoEs without shared experts are recovered when $S=0$. For convenience, we omit the dependence of $p_{l,e}$ and $\mathcal{A}_l$ on $\mathbf{h}_l^n$, as well as the token index n whenever clear from context.

Beyond shared experts, recent MoEs have also shifted toward fine-grained expert segmentation (Liu et al., 2024; Yang et al., 2025; Elango et al., 2026). For the same activation ratio, this trend favors configurations such as $R=64, K-S=8$ over $R=8, K-S=1$, enabling a combinatorially richer set of expert compositions and, hence, more expressive token representations. However, realizing such expressiveness through pretraining is far from straightforward, where the challenge also stems from the *combinatorial* expansion of the routing space. Identifying optimal expert combinations therefore requires more exploration. This motivates us to revisit exploration in MoE training.

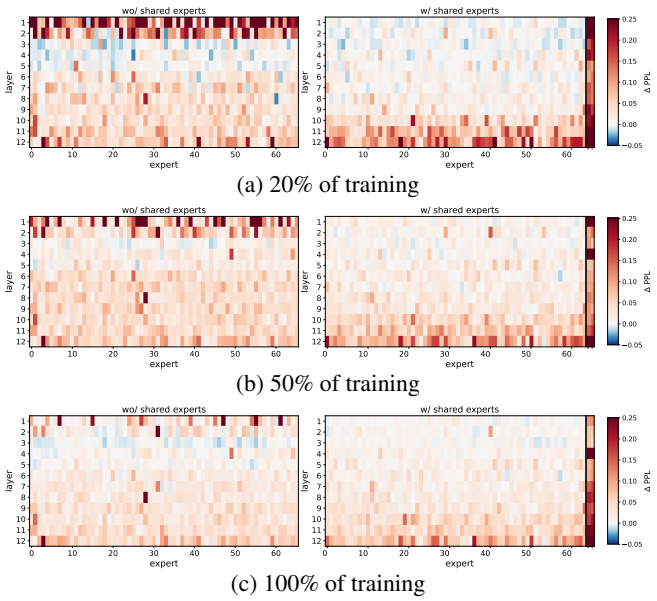
2.2 Shared experts weaken routing exploration

This subsection reexamines the roles of shared and routed experts, reveals an asymmetry in their learning dynamics, and shows how shared-expert dominance can hinder exploration.

To isolate the impact of shared experts, we compare MoEs with and without them using matching datasets, tokenizers, and configurations. The models in this subsection have 121M activated and 498M total parameters, with $E=66$ and $K=8$. We set $S{=}2$ for shared-expert MoEs, while MoEs without shared experts amount to $S{=}0$. Details are deferred to App.B.1. We select checkpoints at 20%, 50%, and 100% of pretraining to represent the early, middle, and final stages, respectively.

Asymmetric training dynamics in shared-expert MoEs. We first ablate individual experts to assess their importance. Specifically, we zero out each expert's output and measure the resulting perplexity change, $\Delta\mathsf{PPL} = \mathsf{PPL}_{\text{after}} - \mathsf{PPL}_{\text{base}}$. In Fig.2, the two rightmost columns in the right-hand plots correspond to shared experts. Across all three checkpoints, ablating a shared expert leads to a substantially larger perplexity increase than ablating a routed expert. This shows that shared experts play a disproportionately important role in model performance.

Another means to quantify the asymmetric roles of experts is through the so-termed *routing criticality*, which replaces the learned router with uniform random routing and measures the resulting performance change. We apply this intervention one layer at a time and report the relative perplexity increase, $\frac{\mathsf{PPL}_{\text{after}} - \mathsf{PPL}_{\text{base}}}{\mathsf{PPL}_{\text{base}}}$. Larger values indicate greater dependence on the learned router. As shown in Fig.3a, uniform routing in the early and middle layers of the shared-expert MoE leads to a much smaller perplexity increase than in standard MoEs. In other words, *shared experts in these layers capture such a large fraction of the learned representation that the layers are robust to random routing.*



(a) 20% of training

(b) 50% of training

(c) 100% of training

Figure 2: Perplexity increase after ablating individual experts. In the right subplot, the two rightmost columns, separated by a solid line, correspond to shared experts.

Next, we examine how this asymmetry in expert roles shapes exploration. Exploration aims to discover alternative routes that produce diverse token representations. We therefore quantify exploration by measuring how much the learned representation changes under different routing decisions. Even before this analysis, the asymmetry between shared and routed experts suggests that exploration may be ineffective: if a highly exploratory routing policy, such as uniform routing, barely changes the output of a shared-expert layer, other routing is also unlikely to produce substantially different representations. We next confirm this intuition using two complementary variance metrics, one induced by experts and the other by routers.

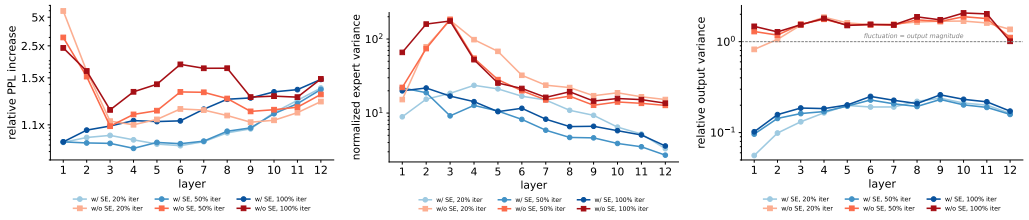
Shared experts suppress expert-induced variance. We first quantify how much an MoE layer's output can vary when different experts are activated. To this end, we introduce the notion of *expert variance*. Given an input token $\mathbf{h}_l$ and its learned output $\mathbf{h}_{l+1}$ in (2), we define the expert variance as

$$\mathsf{ExpertVar}_l = \frac{1}{R}\sum_{e=1}^{R} \frac{\|\tilde{\mathbf{h}}_{l+1,e} - \mathbf{h}_{l+1}\|^2}{\|\mathbf{h}_{l+1}\|^2} \quad \text{where} \quad \tilde{\mathbf{h}}_{l+1,e} = \sum_{s=1}^{S} \mathcal{SE}_{l,s}(\mathbf{h}_l) + \mathcal{RE}_{l,e}(\mathbf{h}_l).$$

In other words, $\tilde{\mathbf{h}}_{l+1,e}$ is the representation obtained if only a single expert e is activated. A larger $\mathsf{ExpertVar}_l$ indicates that changing routed experts can produce substantially different representations compared to $\mathbf{h}_{l+1}$, indicating greater potential for exploration. Moreover, $\mathsf{ExpertVar}_l$ bounds the relative change induced by *any* routing policies from above. Let $\hat{\mathbf{h}}_{l+1}$ denote the representation obtained from any set of activated experts and routing probabilities. As shown in App.C.1, we can prove that $\max_{\hat{\mathbf{h}}_{l+1}} \|\hat{\mathbf{h}}_{l+1} - \mathbf{h}_{l+1}\|^2 / \|\mathbf{h}_{l+1}\|^2 \leq R \cdot \mathsf{ExpertVar}_l$. We evaluate $\mathsf{ExpertVar}_l$, averaged over tokens, for every MoE layer in Fig.3b. Expert variance is smaller across all layers when shared experts are present. This is because shared experts dominate the output, while routed experts can only make relatively small variations. Consequently, shared-expert MoEs are less explorative even with different activated experts.

Shared experts reduce router-induced variance. Next, we show that when routing decisions are sampled from the router's learned distribution, shared-expert MoEs also produce less variable output features. To this end, we replace deterministic top-$(K-S)$ routing in (1) with stochastic expert selection according to the learned routing probabilities. For each token, we independently sample $M = 256$ expert sets and denote the resulting representation of the m-th sample by $\hat{\mathbf{h}}_{l+1}(m)$, where $m \in [M]$. We then define the router-induced variance as

$$\mathsf{RouterVar}_l = \frac{1}{M}\sum_{m} \frac{\|\hat{\mathbf{h}}_{l+1}(m) - \bar{\mathbf{h}}_{l+1}\|^2}{\|\bar{\mathbf{h}}_{l+1}\|^2} \quad \text{where} \quad \bar{\mathbf{h}}_{l+1} = \frac{1}{M}\sum_{m} \hat{\mathbf{h}}_{l+1}(m).$$



(a) Routing criticality (b) Expert variance (c) Router variance

Figure 3: Shared experts make MoE routing less exploratory.

A larger $\mathsf{RouterVar}_l$ indicates that sampling according to the routing distribution is more likely to produce diverse features, and thus more explorative. Fig.3c reports $\mathsf{RouterVar}_l$ averaged over tokens. Because shared experts dominate the layer output, they reduce $\mathsf{RouterVar}_l$ by approximately an order of magnitude relative to a standard MoE. The stochastic routing above is equivalent to adding Gumbel noise with $\beta = 1$ to the routing logits (Nie et al., 2021). We further evaluate perturbed routing distributions with $\beta \in \{0.5, 2\}$. Results in App.B.2 exhibit the same trend as Fig.3c. Thus, even under routing distributions perturbed from the learned one, shared-expert MoEs are less likely to produce diverse representations.

In summary, our diagnosis suggests that exploration faces inherent limitations in shared-expert MoEs.

Takeaway. Shared experts can dominate representation learning, limiting representational diversity and thereby weakening routing exploration.

2.3 Consequences of the Exploration Dilemma

The preceding discussions reveals a fundamental *exploration dilemma* in shared-expert MoEs. On the one hand, the combinatorial growth of the routing space induced by fine-grained expert segmentation demands greater exploration to identify (near-)optimal expert combinations. On the other hand, shared experts reduce feature variation, thereby weakening exploration. This mismatch gives rise to several practical inefficiencies. We reveal them by evaluating models ranging from 100M to 2.4B total parameters. We set $R \in \{64, 128\}$ and $S = 2$ for shared-expert MoEs, and match the total number of experts for MoEs without shared experts. We vary the number of activated experts as $K \in \{6, 8, 10\}$. In this scenario, increasing K improves expressiveness but also enlarges the exploration space.

Inefficiency 1. Benefits of shared experts are eroded. The first consequence is that shared-expert MoEs do not consistently improve upon standard MoEs. As shown in Fig.1, when K is small and hence demanding less exploration, shared-expert MoEs outperform their non-shared counterparts. As K increases, however, the benefits of shared experts diminish or even disappear.

Inefficiency 2. Shared experts can diminish the expressive benefits of increasing K. Across the four model scales in Fig.1, increasing K from 6 to 10 yields perplexity gains of $\{2.23, 1.32, 1.08, 0.87\}$ for MoEs without shared experts. With shared experts, however, these gains drop to $\{0.29, -0.03, 0.58, -0.03\}$, i.e., negative at two scales. These negative numbers indicate that additional expressiveness from activating more experts is not fully realized with shared experts.

Inefficiency 3. Shared experts slow routing specialization. As shown in Fig.3a, replacing the learned router with a uniformly random one leads to a steadily increasing relative perplexity gap across training stages for MoEs without shared experts. This is expected as routing becomes increasingly specialized. In contrast, shared-expert MoEs exhibit little increase in this gap between the 20% and 50% checkpoints, suggesting slower routing specialization.

3 MirrorMoE for Pretraining Shared-Expert MoEs

To reconcile this exploration dilemma and mitigate the inefficiencies discussed earlier, we propose MirrorMoE, a simple yet effective pretraining approach tailored to MoEs with shared experts.

3.1 MIRRORMOE SHRINKS THE EXPLORATION SPACE

One solution to the exploration dilemma is to encourage exploration. However, this can introduce tradeoffs in MoE training, such as reduced stability and increased optimization variance, as further illustrated in App.B.3. We thus take the opposite approach: *shrink the routing space to better match the model's exploration capability.*

An intuitive way to view this is through the number of effective expert combinations, which scales with $\binom{R}{K_{\text{eff}}}$. We formally define K_{eff} in Sec.3.2; for now, it can be understood as the number of functionally active experts induced by the routing method. The routing space is reduced when $K_{\text{eff}} < K - S$. However, as shown by the ablation in Sec.4.3, naively reducing activated experts is suboptimal. MirrorMoE reduces K_{eff} by leveraging experts with low routing probabilities. It selects these low-probability experts in a token-adaptive manner but assigns them negligible aggregation weights. They therefore contribute little to the learned token representation and are functionally suppressed, reducing the number of experts that need to be explored. We detail this approach below.

Mirror routing. Recall that $\mathbf{G}_l$ denotes the router weights at the l-th layer, and consider a token representation $\mathbf{h}_l \in \mathbb{R}^d$. Mirror routing activates experts according to a new routing probability

$$p_{l,e} = (1-\rho)\underbrace{\frac{\exp(\mathbf{G}_l[e,:]\mathbf{h}_l)}{\sum_{j=1}^{R}\exp(\mathbf{G}_l[j,:]\mathbf{h}_l)}}_{:=\xi_{l,e}} + \rho\underbrace{\frac{\exp(-\mathbf{G}_l[e,:]\mathbf{h}_l)}{\sum_{j=1}^{R}\exp(-\mathbf{G}_l[j,:]\mathbf{h}_l)}}_{:=\zeta_{l,e}} \tag{3}$$

where any $\rho \in [0,1]$ yields a valid probability distribution. The first term $\xi_{l,e}$ is the standard routing distribution and favors experts with larger logits. The second term $\zeta_{l,e}$ is obtained by negating the sign of $\mathbf{G}_l$, and therefore favors experts with smaller logits. This sign reversal motivates the name mirror routing. The convex combination of $\xi_{l,e}$ and $\zeta_{l,e}$ shifts routing probability towards both ends of the logit range. Experts with larger $p_{l,e}$ are activated

$$\mathcal{A}_l(\mathbf{h}_l) = \mathsf{Top}_{K-S}\big(\{p_{l,e}\}, e \in [R]\big). \tag{4}$$

Mirror aggregation. This is the central mechanism for shrinking the exploration space, where we aggregate the selected experts using the probabilities $\xi_{l,e}$ instead of $p_{l,e}$

$$\mathbf{h}_{l+1} = \sum_{s=1}^{S}\mathcal{SE}_{l,s}(\mathbf{h}_l) + \sum_{e\in\mathcal{A}_l}\frac{\xi_{l,e}}{\sum_{j\in\mathcal{A}_l}\xi_{l,j}}\mathcal{RE}_{l,e}(\mathbf{h}_l). \tag{5}$$

Consequently, even when low-logit experts are activated, they contribute minimally to the representation $\mathbf{h}_{l+1}$ and are therefore functionally suppressed, reducing K_{eff}. This reduction is *adaptive*: it is determined by $\xi_{l,e}$ and can vary across tokens and layers, as visualized shortly in Sec.3.2. Viewed jointly, MirrorMoE router $\mathbf{G}_l$ serves two complementary roles. As in standard routing, it governs exploitation by selecting experts with large logits. More importantly, it leverages small logits, which standard routing discards, to determine how many experts to suppress.

Lastly, we make two practical modifications to the pretraining procedure to further improve performance. First, because permanently suppressing experts could limit the expressiveness of the final model, we gradually anneal ρ to 0 over the course of training in (3). A cosine schedule is found effective in our experiments; see Sec.4.3. This yields a temporally strategic exploration scheme: the routing space gradually expands only after experts have developed more stable specialization.

Second, a modest additional improvement can be achieved by balancing the probabilities $\xi_{l,e}(\mathbf{h}_l^n)$ rather than the mirror-routing probabilities $p_{l,e}(\mathbf{h}_l^n)$. Specifically, modify the load-balancing loss to

$$\mathcal{L}_{\text{bal}} = R\sum_{l=1}^{L}\sum_{e=1}^{R} f_{l,e}\bar{q}_{l,e}, \quad \text{with} \quad f_{l,e} = \frac{c_{l,e}}{N(K-S)}, \quad \bar{q}_{l,e} = \frac{1}{N}\sum_{n=1}^{N}\xi_{l,e}(\mathbf{h}_l^n) \tag{6}$$

where N is the number of tokens and $c_{l,e}$ is the number of tokens routed to expert e at lth layer. The intuition is that experts with small $\xi_{l,e}(\mathbf{h}_l^n)$ contribute little to the resulting token representations, even if activated. The load-balancing objective therefore focuses on the experts that are functionally active. As shown in App.B.5, this modification improves performance. The complete MirrorMoE pretraining procedure is summarized in Alg.1.

Algorithm 1 MirrorMoE training iteration t

1: **Forward:** $\rho_t \leftarrow \mathrm{CosineAnneal}(\rho_0, t)$
Select experts for each token using (3) and (4), and aggregate their outputs using (5)
Compute $\mathcal{L} \leftarrow \mathcal{L}_{\text{task}} + \lambda_{\text{bal}}\mathcal{L}_{\text{bal}}$, where $\mathcal{L}_{\text{bal}}$ is defined in (6)
2: **Backward:** Backpropagate $\mathcal{L}$ and update the model parameters

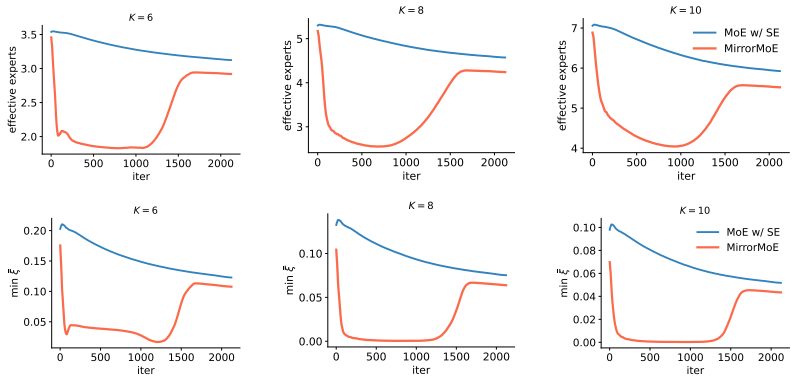


Figure 4: Preview of how MirrorMoE in Alg.1 actively suppresses experts on a 100M MoE model. The rebound in K_{eff} results from annealing ρ.

3.2 Visualization of routing space reduction in MirrorMoE

Next, we empirically show that MirrorMoE shrinks the exploration space early in training and ultimately learns a more confident routing policy, indicating stronger expert specialization. The notion of *effective experts* is introduced to quantify this.

Definition 3.1 ((Token-wise) effective experts)**.** For a given token $\mathbf{h}_l^n$, recall the mirror aggregation in (5). The effective experts for this token is defined as

$$K_{\text{eff}}(\mathbf{h}_l^n) = \exp\Big(-\sum_{e\in\mathcal{A}_l(\mathbf{h}_l^n)} \bar{\xi}_{l,e}(\mathbf{h}_l^n)\log\bar{\xi}_{l,e}(\mathbf{h}_l^n)\Big), \quad \text{where } \bar{\xi}_{l,e}(\mathbf{h}_l^n) = \frac{\xi_{l,e}(\mathbf{h}_l^n)}{\sum_{j\in\mathcal{A}_l(\mathbf{h}_l^n)}\xi_{l,j}(\mathbf{h}_l^n)}.$$

We omit the dependence on $\mathbf{h}_l^n$ for convenience. This definition measures how many of the activated experts contribute substantially to the aggregated representation. In other words, $K_{\text{eff}}(\mathbf{h}_l^n)$ is a smooth approximation to the number of functionally active experts, excluding shared ones. If $\bar{\xi}_{l,e} = 1/(K-S)$ for every $e \in \mathcal{A}_l$, then $K_{\text{eff}} = K - S$, indicating that all routed experts contribute significantly to the aggregation. Conversely, if $\bar{\xi}_{l,e} = 1$ for one expert and zero for others, then $K_{\text{eff}} = 1$, indicating that the final token representation is determined by a single expert. Intuitively, if roughly K_{eff} experts contribute, the number of expert combinations to explore scales as $\binom{R}{K_{\text{eff}}}$, and smaller K_{eff} corresponds to fewer "active" experts and a smaller effective exploration space.

We plot K_{eff}, averaged over all tokens and layers, against iterations in Fig.4. We train an MoE with 100M parameters and vary $K \in \{6, 8, 10\}$. Alongside K_{eff}, we report the minimum normalized aggregation weight among the activated experts, $\min_{e\in\mathcal{A}_l}\bar{\xi}_{l,e}$, also averaged over tokens and layers. It can be seen that MirrorMoE has smaller K_{eff} than the baseline during the first half of training. Meanwhile, $\min_{e\in\mathcal{A}_l}\bar{\xi}_{l,e}$ approaches 0, showing that some activated experts contribute negligibly to the aggregated representation. Together, these results confirm that MirrorMoE reduces exploration space by functionally suppressing activated experts. We further demonstrate that the adaptivity of the suppression in App.B.4 by plotting the layer-wise version of Fig.4. Lastly, as K_{eff} is the exponential entropy of the aggregation probabilities, its smaller value at the end of training in Fig.4 shows that MirrorMoE learns a more concentrated routing policy, consistent with stronger expert specialization.

3.3 Discussions on adaptivity of MirrorMoE

Lastly, we theoretically characterize the *adaptivity* of MirrorMoE by deriving *sufficient* conditions for both expert suppression and preservation at mirror routing. We begin with the former. When an

Table 1: Pretraining performance of MirrorMoE across scales. D/S in the depth column denotes dense/sparse Transformer layers.

Model	Experts	Depth	**PPL** (↓)		**iteration time (ms)** (↓)	
Total-activated params.	$K/R/S$		Vanilla	**MirrorMoE**	Vanilla	**MirrorMoE**
MoE-100M-A38M	8/64/2	1D + 8S	54.78	**51.77**	7871.3 ± 65.9	**7844.4** ± 85.6
MoE-498M-A121M	8/64/2	1D + 12S	28.31	**27.36**	13550.0 ± 63.0	**13426.7** ± 83.3
MoE-625M-A100M	8/128/2	1D + 8S	28.46	**27.84**	9527.5 ± 45.4	**9462.5** ± 61.2
MoE-2.4B-A270M	8/128/2	1D + 15S	21.17	**20.50**	25711.1 ± 176.2	**25635.4** ± 109.4

expert has a sufficiently small $\xi_{l,e}$ and ρ is chosen appropriately, MirrorMoE activates this expert but functionally removes it as negligible aggregation weight is assigned. For convenience, we omit the layer index throughout this subsection.

Proposition 3.2 (Sufficient condition for expert suppression)**.** *For a given token, let $\{\xi_e\}$ and $\{p_e\}$ be probabilities obtained as (3). Let $\mathcal{A}^p := \mathcal{A}$, i.e., (4), denote the set of experts activated by MirrorMoE, and define $\mathcal{A}^\xi := \mathsf{Top}_{K-S}(\{\xi_e\}_{e=1}^R)$ as the set that would be activated under the standard routing probabilities $\{\xi_e\}$. Let $e_{\min} := \arg\min_{e\in[R]} \xi_e$ with $\xi_{e_{\min}} = \epsilon$, and let $j := \arg\min_{e\in\mathcal{A}^\xi} \xi_e$ with $\xi_j = c/R$. As $\epsilon \to 0$, $\rho > c/(1+c)$ is sufficient for $e_{\min} \in \mathcal{A}^p \setminus \mathcal{A}^\xi$; that is, MirrorMoE activates this expert while functionally suppressing its contribution during aggregation.*

This explains the reduced K_{eff} as visualized in Fig.4. Next, we show the token adaptivity of MirrorMoE by identifying a regime in which no expert is suppressed. Specifically, when the router is uncertain about its choices, MirrorMoE behaves similarly to standard routing.

Proposition 3.3 (Sufficient condition for expert preservation)**.** *Let $\xi_{\max} := \max_{e\in[R]} \xi_e$, $\xi_{\min} := \min_{e\in[R]} \xi_e$, and $\kappa := \sqrt{(1-\rho)/\rho}$. If $\xi_{\max}/\xi_{\min} \le \kappa$, $\mathcal{A}^p = \mathcal{A}^\xi$. Moreover, with $\bar{\xi}_e$ defined in Def.3.1, it satisfies $\bar{\xi}_e \ge 1/((K-S)\kappa), \forall e \in \mathcal{A}^p$, i.e., no selected expert is functionally suppressed.*

Another interpretation for MirrorMoE. The above discussions provide an alternative interpretation of MirrorMoE beyond exploration. In MoEs with large E and K, not every token may require the full expressiveness from all activated experts. If a token can be learned well using fewer experts, the capacity of remaining experts can be devoted to other tokens. MirrorMoE can therefore be viewed as an approach for tokens to adaptively determine their number of activated experts, where this adaptivity arises from the expert suppression and preservation shown in Props. 3.2 and 3.3, respectively.

4 Numerical Results

This section evaluates the performance of MirrorMoE. Our implementation builds on (Kang et al., 2025) and uses the Megatron-LM framework (Shoeybi et al., 2019) with the DCLM dataset (Li et al., 2024). All experiments are conducted on NVIDIA H100 GPUs.

4.1 Pretraining performance of MirrorMoE

We evaluate four shared-expert MoEs ranging from 100M to 2.4B total parameters, with $R \in \{64, 128\}$ experts. For each token, 6 routed experts and $S = 2$ shared experts are activated, yielding $K = 8$. Activated parameters account for 16% to 38% of the total parameters across models. The model configurations are summarized in Tab.1 and detailed in App.B.1.

Perplexity. We first compare the test perplexity of pretrained MoE models in Tab.1. MirrorMoE consistently achieves lower perplexity across all model scales. Notably, MoE-498M-A121M and MoE-625M-A100M use the same hidden dimensions and differ primarily in numbers of layers and experts. Since the number of end-to-end routing paths scales as $\binom{E-S}{K-S}^L$, where L denotes the number of sparse layers, MoE-498M-A121M has a larger routing space under our configurations. MirrorMoE also yields a larger performance gain on this model, suggesting that shrinking the routing space is particularly beneficial when the exploration dilemma is more pronounced.

Runtime. MirrorMoE slightly reduces the per-iteration training time despite introducing an additional softmax operation in (3) at each layer. It suggests improved load balance. Note that MoE-100M-A38M is trained on two GPUs, whereas all other models are trained on four.

Table 2: Downstream performance of MoE models trained with MirrorMoE.

Model	Method	ARC-E	OBQA	HellaSwag	PIQA	BoolQ	SIQA	Avg (↑)
MoE-100M-A38M	Vanilla	32.11	26.40	26.62	57.56	42.66	34.29	36.61
	MirrorMoE	34.22	26.00	25.76	58.00	42.48	34.49	**36.82**
MoE-498M-A121M	Vanilla	39.44	26.80	29.87	61.43	58.65	36.69	42.15
	MirrorMoE	41.04	27.80	30.33	64.31	59.20	36.95	**43.27**
MoE-625M-A100M	Vanilla	41.12	28.20	29.97	61.10	61.47	37.36	43.20
	MirrorMoE	41.37	29.80	30.23	62.08	61.16	36.59	**43.53**
MoE-2.4B-A270M	Vanilla	45.37	30.00	36.63	66.59	54.50	38.33	45.23
	MirrorMoE	46.63	32.40	37.40	67.95	58.53	38.28	**46.87**

Downstream tasks. We further demonstrate that the gains of MirrorMoE translate into improved downstream performance on six commonsense reasoning tasks in Tab.2. For the 2.4B model, MirrorMoE improves the average performance by 1.64.

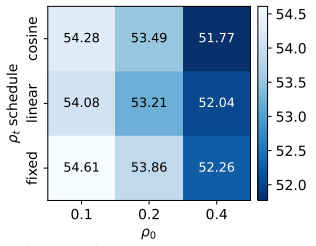


Figure 5: Hyperparameters.

4.2 Comparison under enlarged routing space

Building on the same model configuration, we vary the number of activated experts as $K \in \{6, 8, 10\}$. Increasing K enables richer expert combinations, thereby enlarging both the exploration space and model expressiveness. As shown in Fig.1, the standard shared-expert MoE is outperformed by its non-shared counterpart at large K. In contrast, MirrorMoE mitigates the exploration dilemma, enabling the shared-expert model to consistently outperform the non-shared MoE across all tested cases. More importantly, the perplexity of MirrorMoE consistently decreases as K increases, suggesting that the additional expressiveness enabled by larger K is realized.

Table 3: Comparison with non-adaptive schedules.

Method	Test PPL (↓)
MoE	54.78
Linear scheduled $\widetilde{K}_t$	54.91
Cosine scheduled $\widetilde{K}_t$	54.19
MirrorMoE	**51.77**

4.3 Additional experiments

We use a MoE-100M-A38M with $K = 8$ (2 shared and 6 routed experts) for ablation study.

Hyperparameters. Fig.5 compares constant, linear, and cosine schedules for ρ_t in (3), together with initial values $\rho_0 \in \{0.1, 0.2, 0.4\}$. A cosine schedule with $\rho_0 = 0.4$ achieves the best performance and is therefore used in all preceding experiments.

Benefit of adaptivity. We demonstrate the importance of token/layer adaptivity in MirrorMoE by comparing it with two simple approaches that also have smaller exploration space. Both begin with fewer activated experts and progressively increase their number. Specifically, we set $K = \mathsf{round}(\widetilde{K}_t)$, where $\widetilde{K}_t$ increases from 4 to 8 following either a linear or cosine schedule. Here, $\widetilde{K}_t$ includes two shared experts, and the initial value $\widetilde{K}_0 = 4$ is informed by the number of effective experts in Fig.4. These baselines use the same K for all tokens and layers and therefore lack adaptivity compared to MirrorMoE. As shown in Tab.3, neither matches the performance of MirrorMoE.

MirrorMoE is tailored to shared-expert MoEs. We additionally apply MirrorMoE to MoEs *without* shared experts to further validate our motivation based on the exploration dilemma induced by shared experts. As shown in Tab.7 in App.B.6, MirrorMoE performs comparably to vanilla training. This is consistent with our analysis: without shared experts, the exploration dilemma is less pronounced, leaving less need to shrink the exploration space.

5 Conclusion

This work identifies an exploration dilemma in MoEs that combine fine-grained expert segmentation with shared experts. This tension can prevent the additional expressiveness enabled by fine-grained segmentation from being fully realized while also diminishing the benefits of shared experts. To address this, we propose MirrorMoE, which adaptively shrinks the exploration space to better match the model's exploration capability. MirrorMoE consistently improves performance across scales.

AI USE STATEMENT

In this work, we used generative AI tools for providing feedback on research methodology or experiments, and implementing methods. We have not used generative AI tools for proposing or refining hypotheses, assisting in the writing of proofs, interpreting results, supporting qualitative and thematic data analysis, developing theoretical models or conceptual frameworks, formulating mathematical claims, or providing critical ingredients for proving mathematical claims. The rest of the required disclosure tasks, including generating synthetic datasets, assisting with translation, and cleaning datasets are not applicable to this work. Additionally, we used generative AI tools for creating or modifying scientific figures, editing a research paper to improve readability, summarizing existing literature. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work focuses on improving the training of MoE models and does not involve human subjects or personally identifiable information. As with other large language models, caution is warranted for applications involving content generation. Users should carefully review generated outputs and consider appropriate safety mechanisms to improve reliability and trustworthiness.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, the algorithms, detailed experimental settings, and hyperparameters are provided in Sec.4 and App.B.1. The complete theoretical proofs are included in App.C. We plan to release the code repository upon acceptance.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv e-prints*, pp. arXiv–2404, 2024.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Aidan Clark, Diego de Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. Unified scaling laws for routed language models. In *International conference on machine learning*, pp. 4057–4086. PMLR, 2022.

Róbert Csordás, Christopher D Manning, and Christopher Potts. Do language models use their depth efficiently? *arXiv preprint arXiv:2505.13898*, 2025.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1280–1297, 2024.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International conference on machine learning*, pp. 5547–5569. PMLR, 2022.

Venmugil Elango, Nidhi Bhatia, Roger Waleffe, Rasoul Shafipour, Tomer Asida, Abhinav Khattar, Nave Assaf, Maximilian Golub, Joey Guman, Tiyasa Mitra, et al. Latentmoe: Toward optimal accuracy per flop and parameter in mixture of experts. *arXiv preprint arXiv:2601.18089*, 2026.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. Harder task needs more experts: Dynamic routing in moe models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12883–12895, 2024.

Changho Hwang, Wei Cui, Yifan Xiong, Ziyue Yang, Ze Liu, Han Hu, Zilong Wang, Rafael Salas, Jithin Jose, Prabhat Ram, et al. Tutel: Adaptive mixture-of-experts at scale. *Proceedings of Machine Learning and Systems*, 5:269–287, 2023.

Nandan Kumar Jha and Brandon Reagen. Same architecture, different capacity: Optimizer-induced spectral scaling laws. *arXiv preprint arXiv:2605.21803*, 2026.

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, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Peng Jin, Bo Zhu, Li Yuan, and Shuicheng Yan. Moe++: Accelerating mixture-of-experts methods with zero-computation experts. *arXiv preprint arXiv:2410.07348*, 2024.

Gyunyeop Kim Sangwoo Kang. Exploration-driven reinforcement learning for expert routing improvement in mixture-of-experts language models. 2025.

Hao Kang, Zichun Yu, and Chenyan Xiong. Flame-moe: A transparent end-to-end research platform for mixture-of-experts language models. *arXiv preprint arXiv:2505.20225*, 2025.

Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. pp. 3499–3508. PMLR, 2019.

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. *arXiv preprint arXiv:2006.16668*, 2020.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. Base layers: Simplifying training of large, sparse models. In *International Conference on Machine Learning*, pp. 6265–6274. PMLR, 2021.

Houyi Li, Ka Man Lo, Shijie Xuyang, Ziqi Wang, Wenzhen Zheng, Haocheng Zhang, Zhao Li, Shuigeng Zhou, Xiangyu Zhang, and Daxin Jiang. Mixture-of-experts can surpass dense llms under strictly equal resource. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.

Jingcong Liang, Siyuan Wang, Miren Tian, Yitong Li, Duyu Tang, et al. Not all models suit expert offloading: On local routing consistency of mixture-of-expert models. In *International Conference on Learning Representations*, volume 2026, pp. 64993–65021, 2026.

Kai Lion, Liang Zhang, Bingcong Li, and Niao He. Polar: Polar-decomposed low-rank adapter representation. *Advances in Neural Information Processing Systems*, 38:149313–149358, 2026.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Xinyi Liu, Yujie Wang, Fangcheng Fu, Xupeng Miao, Shenhan Zhu, Xiaonan Nie, and Bin Cui. NetMoE: Accelerating moe training through dynamic sample placement. In *The Thirteenth International Conference on Learning Representations*, 2025.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. Olmoe: Open mixture-of-experts language models. In *International Conference on Learning Representations*, volume 2025, pp. 62061–62121, 2025.

Mohammed Muqeeth, Haokun Liu, and Colin Raffel. Soft merging of experts with adaptive routing. *arXiv preprint arXiv:2306.03745*, 2023.

Xiaonan Nie, Xupeng Miao, Shijie Cao, Lingxiao Ma, Qibin Liu, Jilong Xue, Youshan Miao, Yi Liu, Zhi Yang, and Bin Cui. Evomoe: An evolutional mixture-of-experts training framework via dense-to-sparse gate. *arXiv preprint arXiv:2112.14397*, 2021.

Stephen Roller, Sainbayar Sukhbaatar, Jason Weston, et al. Hash layers for large sparse models. *advances in neural information processing systems*, 34:17555–17566, 2021.

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. *arXiv preprint arXiv:1701.06538*, 2017.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Team-Gemma. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Team-Kimi. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Team-Kimi. Kimi k3: Open frontier intelligence. 2026. URL https://www.kimi.com/blog/kimi-k3.

Team-Qwen. Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters". *URL https://qwenlm.github.io/blog/qwen-moe*, 2024.

Amirhossein Vahidi, Hesam Asadollahzadeh, Navid Akhavan Attar, Marie Moullet, Kevin Ly, Xingyi Yang, and Mohammad Lotfollahi. Dirmoe: Dirichlet-routed mixture of experts. *arXiv preprint arXiv:2602.09001*, 2026.

Ziteng Wang, Jun Zhu, and Jianfei Chen. Remoe: Fully differentiable mixture-of-experts with relu routing. In *International Conference on Learning Representations*, volume 2025, pp. 59486–59507, 2025.

Yudong Wei, Liang Zhang, Bingcong Li, and Niao He. On the benefits of weight normalization for overparameterized matrix sensing. *arXiv preprint arXiv:2510.01175*, 2025a.

Yujie Wei, Shiwei Zhang, Hangjie Yuan, Yujin Han, Zhekai Chen, Jiayu Wang, Difan Zou, Xihui Liu, Yingya Zhang, Yu Liu, et al. Routing matters in moe: Scaling diffusion transformers with explicit routing guidance. *arXiv preprint arXiv:2510.24711*, 2025b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Tongtian Yue, Longteng Guo, Jie Cheng, Xuange Gao, Hua Huang, and Jing Liu. Ada-K routing: Boosting the efficiency of moe-based llms. In *The Thirteenth International Conference on Learning Representations*, 2024.

Zihao Zeng, Yibo Miao, Hongcheng Gao, Hao Zhang, and Zhijie Deng. Adamoe: Token-adaptive routing with null experts for mixture-of-experts language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 6223–6235, 2024.

Shulai Zhang, Ningxin Zheng, Haibin Lin, Ziheng Jiang, Wenlei Bao, Chengquan Jiang, Qi Hou, Weihao Cui, Size Zheng, Li-Wen Chang, et al. Comet: Fine-grained computation-communication overlapping for mixture-of-experts. *Proceedings of Machine Learning and Systems*, 7, 2025.

Yilang Zhang, Bingcong Li, Niao He, and Georgios B Giannakis. Ancre: Adaptive neural connection reassignment for efficient depth scaling. *arXiv preprint arXiv:2602.09009*, 2026.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

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. *arXiv preprint arXiv:2202.08906*, 2022.

Appendix

A Additional related work

Expressiveness does not necessarily translate directly into performance. Beyond those we have discussed earlier in Sec.1.1, additional works similarly suggest that the expressive capacity of neural networks may not be fully utilized in practice. For example, (Csordás et al., 2025) shows that many deeper layers in Llama 3.1 70B behave close to identity mappings. Since identity mappings are available "for free", this motivates redesign of residual connections for scaling deeper (Zhang et al., 2026).

MoE systems. A complementary line of work focuses on efficient MoE systems, including faster and more principled expert parallelism, fine-grained all-to-all communication, and topology-aware communication strategies; see e.g., (Hwang et al., 2023; Liu et al., 2025; Zhang et al., 2025). These methods aim to improve hardware utilization, communication efficiency, and end-to-end throughput. MirrorMoE is largely orthogonal to these system-level optimizations and can be integrated with them.

Routing in MoEs. A broad line of work studies alternative routing mechanisms for MoEs. Hash Layers (Roller et al., 2021) use hash-based assignments, while BASE Layers (Lewis et al., 2021) and expert-choice routing (Zhou et al., 2022) promote balanced expert utilization through carefully designed token assignments, at the cost of not being strictly autoregressive. Clark et al. (2022) study scaling laws for routed models. Other approaches replace hard top-K selection with differentiable routing mechanisms (Muqeeth et al., 2023; Wang et al., 2025; Vahidi et al., 2026). Several works post-train or fine-tune routers to enable token-adaptive computation (Yue et al., 2024; Zeng et al., 2024), while Kang (2025) uses reinforcement learning to improve routing decisions. During pretraining, noisy routing has also been explored to encourage broader expert exploration: Shazeer et al. (2017) introduce Gaussian noise, whereas Nie et al. (2021) use Gumbel noise. Routing has also been explicitly studied for multimodal models (Wei et al., 2025b). In contrast to approaches that encourage broader exploration, MirrorMoE adaptively suppresses experts to shrink the exploration space. The work of Liang et al. (2026) studies shared experts from an inference perspective, showing that they can hinder expert offloading, partly because shared experts reduce the importance of routed experts. However, their focus is on routing behavior at inference time, whereas we study how shared experts affect routing throughout pretraining.

B Experimental details

B.1 Training details

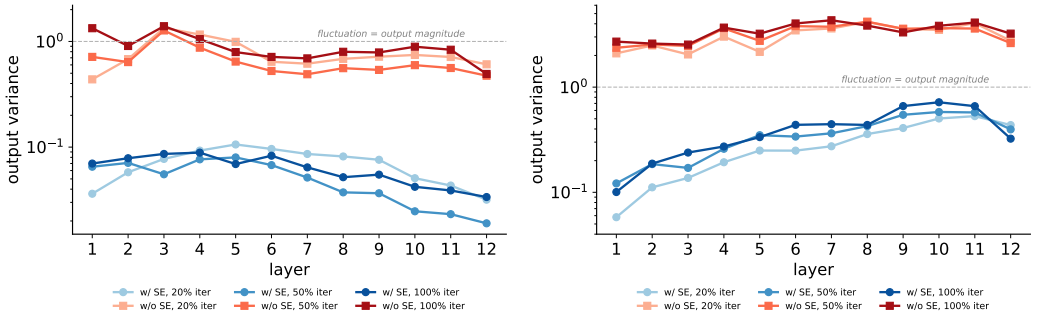
The model configurations for the shared-expert MoE baseline and MirrorMoE are summarized in Tab.4. Following Kang et al. (2025), we adopt a Llama-style MoE architecture. We also follow DeepSeekMoE (Dai et al., 2024; Liu et al., 2024) by using a dense FFN in the first Transformer layer, so that the first sparse layer receives sufficiently developed token representations for routing. MoE-100M-A38M is trained on two H100 GPUs, while all other models are trained on four. DCLM dataset (Li et al., 2024) is adopted. We use AdamW for optimization. The learning rate is selected from $\{1, 3, 5\} \times 10^{-4}$ based on performance after 500 training steps on a held-out validation set. We adopt a warmup-stable-decay (WSD) learning-rate schedule. The global batch size is 1024, with gradient accumulation used to reduce memory consumption.

The model analyzed in Sec.2.2 is MoE-498M-A121M.

For MoEs without shared experts, we convert the shared experts into routed experts to preserve the same total parameter count. Due to implementation constraints in Megatron-LM, this requires a different expert layout across GPUs, i.e., a different expert-parallelism configuration, and consequently a different number of GPUs. Specifically, Megatron-LM requires the number of routed experts to be divisible by the expert-parallelism size. Therefore, configurations with 66 or 130 routed experts use three or five GPUs, respectively. The global batch size consequently differs across configurations, while the total number of training tokens is kept approximately the same to ensure a fair comparison.

Table 4: Model configurations used in our experiments.

Parameter	MoE-100M-A38M	MoE-498M-A121M	MoE-625M-A100M	MoE-2.4B-A270M
Dense / MoE layers	1 / 8	1 / 12	1 / 8	1 / 15
Layer layout	1D + 8S	1D + 12S	1D+8S	1D + 15S
Hidden size (d_{model})	256	512	512	768
Dense FFN hidden	1368	2736	2736	3648
MoE FFN hidden (per expert)	176	352	352	512
Shared expert FFN ($2\times$)	352	704	704	1024
# Routed experts	64	64	128	128
Top-k (activated)	6	6	6	6
Sequence length	2048	2048	2048	2048
Load balancing	0.1	0.1	0.1	0.1
# Training tokens	4.4B	7.6B	7.6B	12.8B



(a) Gumbel noise with $\beta = 0.5$

(b) Gumbel noise with $\beta = 2$

Figure 6: Router variance under different levels of Gumbel perturbation.

B.2 Router Variance under Different Levels of Gumbel Noise

When evaluating $\mathsf{RouterVar}_l$, we replace deterministic top-K (for non-shared MoEs) or top-$(K-S)$ (for shared-expert MoEs) routing with stochastic sampling from the learned routing distribution. Such sampling is equivalent to perturbing the routing logits with unit-scale Gumbel noise ($\beta = 1$) and then applying top-K selection (Kool et al., 2019; Nie et al., 2021). We further extend this evaluation by defining $\mathsf{RouterVar}_l^{\beta}$, where the routing logits are perturbed by Gumbel noise parameterized by other βs. We consider $\beta \in \{0.5, 2\}$ and report the resulting $\mathsf{RouterVar}_l^{\beta}$ for MoE-498M-A121M model in Fig.6. The same trend persists across these perturbation levels, further showing that shared experts consistently reduce routing-induced variance.

B.3 Increasing Exploration during Pretraining Can Be Tricky

An alternative way to address the exploration dilemma is to encourage greater exploration during pretraining. However, this can be challenging for MoEs because stochastic routing may introduce unstable router dynamics and increased training variance. To examine this tradeoff, we consider two alternative routing strategies under $K = 8$. First, we replace deterministic top-$(K-S)$ routing with stochastic sampling of $K-S$ experts, denoted as Sto-$(K-S)$ in Tab.5. Second, we deterministically select the top-2 experts and stochastically sample the remaining $K-S-2$ experts, denoted as Top-2 + Sto-$(K-S-2)$.

As shown in Tab.5, on a MoE-100M-A38M model, deterministic top-$(K-S)$ routing performs best, followed by Top-2+Sto-$(K-S-2)$, while fully stochastic routing performs worst. These results suggest that the additional instability introduced by exploration can outweigh its potential benefits during pretraining. For this reason, MirrorMoE takes the opposite approach by shrinking the exploration space.

Table 5: Intrinsic difficulty of increasing exploration

	Top-$(K-S)$	Sto-$(K-S)$	Top-2 with Sto-$(K-S-2)$
PPL	54.78	55.74	55.38

Table 6: Comparison of different load balancing loss for MirrorMoE

	vanilla balancing loss	our load balancing loss (6)
PPL	52.94	51.77
runtime	7727.1 ± 91.8	7844.4 ± 85.6

B.4 Layer-wise effective experts

Extending Fig.4, Fig.7 reports the layer-wise number of effective experts, averaged over tokens when training an MoE-100M-A38M model. For example, when $K = 8$ in Fig.7b, Layers 1 and 2 maintain roughly three effective experts during the first half of training, whereas Layer 3 suppresses more aggressively and approaches a single effective expert. In contrast, Layers 7 and 8 are more conservative, retaining around four effective experts. This confirms that MirrorMoE shrinks the exploration space in a layer-adaptive manner.

B.5 On the Load-Balancing Loss

We use an MoE-100M-A38M to compare the standard load-balancing loss with our modified loss in (6). The two objectives differ in whether they balance the mirror routing probabilities $\{p_{l,e}\}_e$ or the original routing probabilities $\{\xi_{l,e}\}_e$. As shown in Tab.6, balancing $\{\xi_{l,e}\}_e$ reduces perplexity, at the cost of a slight increase in per-iteration training time. Since our primary objective is model performance, we adopt the modified load-balancing loss throughout our experiments.

B.6 MirrorMoE performance degrades for MoEs without shared experts

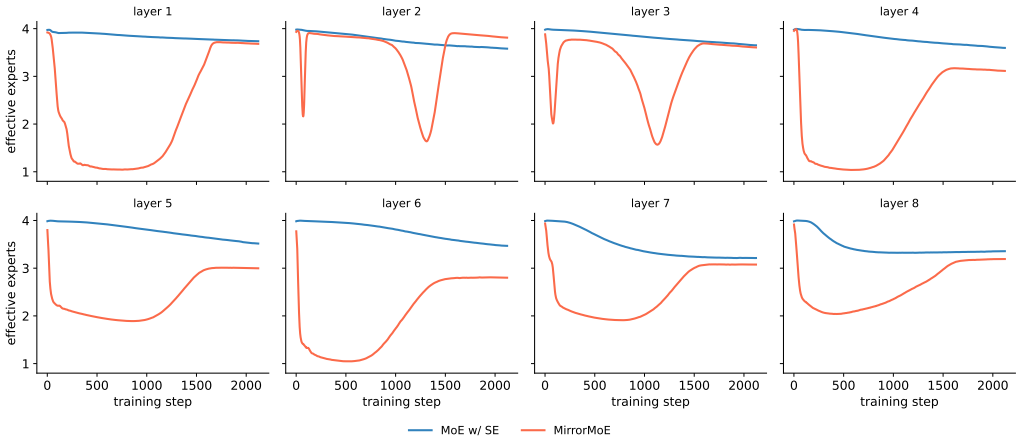
We further evaluate MirrorMoE on MoEs without shared experts. For ease of parallelization, we use $E \in \{64, 128\}$ and $K = 8$, with slightly adjusted model depths. These results are therefore not directly comparable to those in Tab.1. As shown in Tab.7, without shared experts, MirrorMoE with $\rho_0 = 0.4$ and a cosine schedule performs comparably to standard training, with no significant improvement. This result is consistent with our motivation: when shared experts are absent, the exploration dilemma is less pronounced, leaving less need to shrink the exploration space.

B.7 Comparison with top-p routing

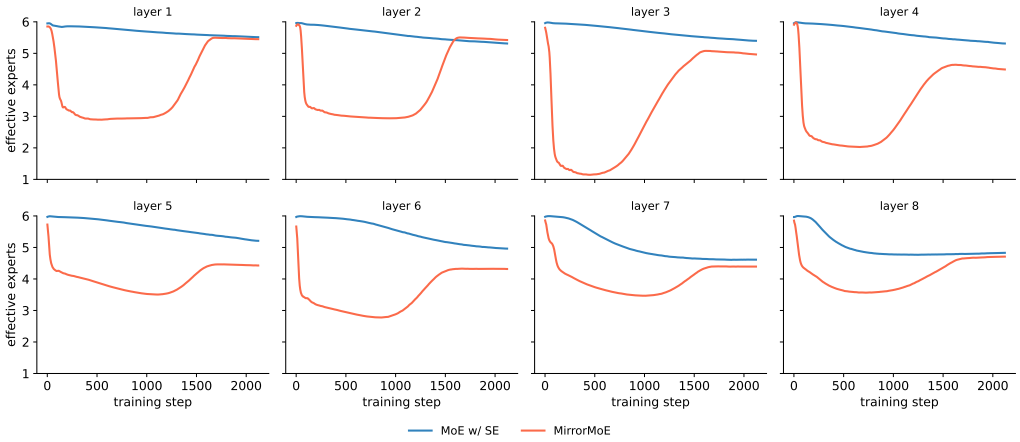
We additionally compare MirrorMoE with another adaptive routing approach, namely top-p routing (Huang et al., 2024). Top-p routing activates the smallest set of experts whose cumulative probability exceeds p, where $p \in (0, 1)$ is a threshold parameter. Intuitively, it also adapts the number of activated experts over training. Early in training, when routing probabilities are relatively uniform, more experts are required to exceed the threshold p. As routing becomes more concentrated, fewer experts are sufficient.

However, this adaptivity makes a fair comparison challenging. In our experiments, we observe that the average number of activated experts per token decreases over the course of training. As a result, top-p routing uses different amounts of activated computation at different training stages, making direct comparison with MirrorMoE and the baseline difficult, as both use a fixed number of activated experts. Tab.8 reports the number of activated experts (including shared ones) averaged over layers and tokens for MoE-100M-A38M with $K = 8$ and $S = 2$.

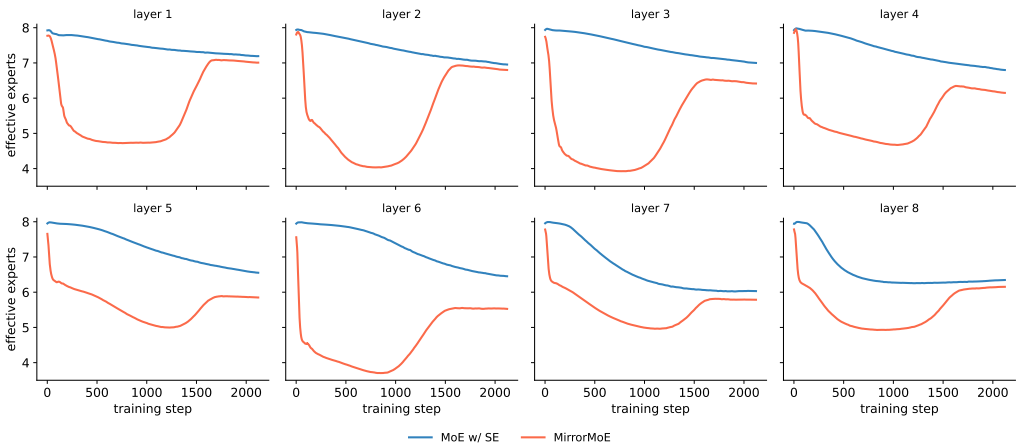
In terms of training performance, we use entropy regularization with a coefficient of 10^{-4} for top-p routing. As shown in Tab.8, even with $p = 0.6$, which activates more experts on average, top-p routing still underperforms MirrorMoE.



(a) $K = 6$



(b) $K = 8$



(c) $K = 10$

Figure 7: Effective experts vs. iterations plot by layers.

Table 7: MirrorMoE has less gains when pretraining MoEs without shared experts.

Model	Experts	Depth	**PPL** (↓)	
Total-activated params	$K/R/S$		Vanilla	**MirrorMoE**
MoE-NS-97M-A36M	8/64/0	8S	52.01	51.23
MoE-NS-480M-A115M	8/64/0	12S	28.24	28.20
MoE-NS-614M-A95M	8/128/0	8S	28.42	28.49
MoE-NS-2.5B-A270M	8/128/0	16S	20.85	20.83

Table 8: Comparison with top-p routing. Activated experts are averaged over layers and tokens for top-p routing.

Method	PPL (↓)	# activated experts at first iteration	# activated experts at last iteration
Vanilla	54.78	8	8
MirrorMoE	**51.77**	8	8
Top-p, $p = 0.4$	55.64	18.65	6.73
Top-p, $p = 0.6$	54.09	29.46	10.79

C Missing proofs

C.1 $\mathsf{ExpertVar}_l$ upper-bounds routing-induced changes

To see that $\mathsf{ExpertVar}_l$ bounds the relative change induced by *any* routing policies from above, let $\tilde{\mathbf{h}}_{l+1}(\boldsymbol{\alpha}) = \sum_{e=1}^{R} \alpha_e \tilde{\mathbf{h}}_{l+1,e}$ where $\boldsymbol{\alpha} \in \mathbb{R}^R$ is any routing probability with $\boldsymbol{\alpha} \geq \mathbf{0}$, and $\mathbf{1}^\top \boldsymbol{\alpha} = 1$. One can easily see through convexity of $\|\cdot\|^2$ that

$$\begin{aligned}\frac{\|\tilde{\mathbf{h}}_{l+1}(\boldsymbol{\alpha}) - \mathbf{h}_{l+1}\|^2}{\|\mathbf{h}_{l+1}\|^2} &\leq \sum_{e=1}^{R} \alpha_e \frac{\|\tilde{\mathbf{h}}_{l+1,e} - \mathbf{h}_{l+1}\|^2}{\|\mathbf{h}_{l+1}\|^2} \\ &\leq \sum_{e=1}^{R} \frac{\|\tilde{\mathbf{h}}_{l+1,e} - \mathbf{h}_{l+1}\|^2}{\|\mathbf{h}_{l+1}\|^2} = R \cdot \mathsf{ExpertVar}_l.\end{aligned}$$

C.2 Proofs of Prop. 3.2

Proof. Recall that in MirrorMoE, one can write $p_e = (1-\rho)\xi_e + \rho\zeta_e$. Using the detailed expression of ξ_e and ζ_e in (3), one can rewrite $\zeta_e = \frac{1/\xi_e}{\sum_{r=1}^{R} 1/\xi_r}$. Given that $e_{\min} := \arg\min_{e\in[R]} \xi_e$ with $\xi_{e_{\min}} = \epsilon$, we have that

$$\sum_{e=1}^{R} \frac{1}{\xi_e} \leq \frac{R}{\xi_{e_{\min}}}.$$

With $j := \arg\min_{e\in\mathcal{A}^\xi} \xi_e$, it follows that

$$\begin{aligned}\zeta_{e_{\min}} - \zeta_j = \frac{1/\xi_{e_{\min}} - 1/\xi_j}{\sum_{r=1}^{R} 1/\xi_r} &\geq \frac{1}{R}\left(1 - \frac{\xi_{e_{\min}}}{\xi_j}\right) \\ &= \frac{1}{R}\left(1 - \frac{\epsilon}{\xi_j}\right).\end{aligned}$$

Using the inequality above, we have that

$$\begin{aligned}p_{e_{\min}} - p_j &= -(1-\rho)\left(\xi_j - \xi_{e_{\min}}\right) + \rho\left(\zeta_{e_{\min}} - \zeta_j\right) \\ &\geq -(1-\rho)\xi_j + \frac{\rho}{R}\left(1 - \frac{\epsilon}{\xi_j}\right).\end{aligned}$$

The right-hand side is positive whenever

$$\rho > \frac{R\xi_j}{R\xi_j + 1 - \epsilon/\xi_j} = \frac{c}{c + 1 - \epsilon R/c}$$

where the last equation comes from $\xi_j = c/R$. As $\epsilon \to 0$, we get the condition for ρ in the proposition.

So far we have proved $p_{e_{\min}} \geq p_j$. To complete the proof, we will show that $e_{\min} \in \mathcal{A}^p$, i.e., $e_{\min}$ is indeed activated in MirrorMoE. Because $j := \arg\min_{e \in \mathcal{A}^\xi} \xi_e$, every non-activated expert $e \notin \mathcal{A}^\xi$ satisfies $\xi_{e_{\min}} \leq \xi_e \leq \xi_j$. Hence, there exists some $\lambda_e \in [0, 1]$ such that

$$\xi_e = \lambda_e \xi_{e_{\min}} + (1 - \lambda_e)\xi_j.$$

Since $f(x) = 1/x$ is convex on $\mathbb{R}_{++}$, we have that

$$\frac{1}{\xi_e} \leq \frac{\lambda_e}{\xi_{e_{\min}}} + \frac{1 - \lambda_e}{\xi_j}.$$

Denote $C = \left(\sum_{e=1}^R \frac{1}{\xi_e}\right)^{-1}$. It then follows that

$$\begin{aligned}
p_e &= (1 - \rho)\xi_e + \rho\frac{C}{\xi_e} \\
&\leq \lambda_e \left((1 - \rho)\xi_{e_{\min}} + \frac{\rho C}{\xi_{e_{\min}}}\right) + (1 - \lambda_e)\left((1 - \rho)\xi_j + \frac{\rho C}{\xi_j}\right) \\
&= \lambda_e p_{e_{\min}} + (1 - \lambda_e)p_j \\
&\leq p_{e_{\min}}, \forall e \notin \mathcal{A}^\xi
\end{aligned}$$

where the final inequality follows from $p_{e_{\min}} > p_j$. Combining $p_{e_{\min}} \geq p_j$ and $p_{e_{\min}} \geq p_e, \forall e \notin \mathcal{A}^\xi$, it can be seen that $e_{\min}$ has to be activated in MirrorMoE, i.e., $e_{\min} \in \mathcal{A}^p$.

Finally, since $e_{\min} \in \mathcal{A}^p$, according to the aggregation in (5), $\xi_{e_{\min}} = \epsilon \to 0$ ensures that $e_{\min}$ is functionally suppressed. The proof is thus completed. □

C.3 Proof of Prop. 3.3

Proof. Let $C = \left(\sum_{r=1}^R \frac{1}{\xi_r}\right)^{-1}$. Recall from (3), $p_e = (1 - \rho)\xi_e + \rho\zeta_e$ and $\zeta_e = \frac{C}{\xi_e}$.

We first show that $C \leq \frac{1}{R^2}$ via Cauchy-Schwarz inequality. To see this, we have

$$\left(\sum_{r=1}^R \xi_r\right)\left(\sum_{r=1}^R \frac{1}{\xi_r}\right) \geq R^2.$$

It is easy to see $C \leq \frac{1}{R^2}$ once noticing $\sum_r \xi_r = 1$.

Now, consider any two experts i and j satisfying $\xi_i > \xi_j$. We have

$$\begin{aligned}
p_i - p_j &= (1 - \rho)(\xi_i - \xi_j) + \rho C\left(\frac{1}{\xi_i} - \frac{1}{\xi_j}\right) \\
&= (\xi_i - \xi_j)\left[(1 - \rho) - \rho\frac{C}{\xi_i\xi_j}\right].
\end{aligned}$$

Since $\xi_i\xi_j \geq \xi_{\min}^2$ and $\xi_{\max} \geq 1/R$, it can be seen that

$$\frac{C}{\xi_i\xi_j} \leq \frac{1}{R^2\xi_i\xi_j} \leq \frac{1}{R^2\xi_{\min}^2} \leq \left(\frac{\xi_{\max}}{\xi_{\min}}\right)^2 < \kappa^2 = \frac{1 - \rho}{\rho}.$$

Combining the above two equations, it is straightforward to see $\xi_i > \xi_j \implies p_i > p_j$ for all i and j. Thus, it can be seen that the ordering of experts are same if ranked with $\{\xi_e\}$ and $\{p_e\}$. This indicates the top-$(K - S)$ aggregation will choose the same set of experts, i.e., $\mathcal{A}^\xi = \mathcal{A}^p$.

It remains to show that none of the selected experts has a vanishing score. For any selected expert $e \in \mathcal{A}^p$,

$$\bar{\xi}_e = \frac{\xi_e}{\sum_{r \in \mathcal{A}^p} \xi_r} \geq \frac{\xi_{\min}}{(K-S)\xi_{\max}} > \frac{1}{(K-S)\kappa}.$$

Thus, every selected expert has a non-vanishing aggregation weight, which rules out functional path pruning. □