SYNERGISTIC INTRA- AND CROSS-LAYER REGULARIZATION LOSSES FOR MOE EXPERT SPECIALIZATION

Rizhen Hu[1,*] **Yuan Cao**[1,*] **Boao Kong**[1,*] **Mou Sun**[2,†] **Kun Yuan**[1,†]

[1]Peking University, Beijing, China [2]Zhejiang Lab, Hangzhou, China

[*]Equal contributions [†]Corresponding authors

ABSTRACT

Sparse Mixture-of-Experts (MoE) models scale Transformers efficiently but suffer from expert overlap—redundant representations across experts and routing ambiguity, resulting in severely underutilized model capacity. We propose two plug-and-play regularization losses that improve specialization and routing consistency without changing router or model architectures. First, an intra-layer specialization loss penalizes cosine similarity between experts' SwiGLU activations on identical tokens to encourage complementary representations. Second, a cross-layer coupling loss maximizes joint Top-k routing probabilities across adjacent layers to promote coherent expert pathways through depth. The two losses are mutually reinforcing: improved specialization reduces overlap and stabilizes pathways, while coupling reduces routing volatility and amplifies specialization. Both losses are orthogonal to the standard load-balancing loss and compatible with both the shared-expert architecture in DeepSeekMoE and vanilla top-k MoE architectures. We implement both losses as a drop-in Megatron-LM module. Extensive experiments across pre-training and downstream benchmarks demonstrate consistent task gains, higher expert specialization, and lower-entropy routing; together, these improvements translate into faster inference via more stable expert pathways.

1 INTRODUCTION

Sparse Mixture-of-Experts (MoE) has emerged as a standard approach for scaling Transformers by expanding parameters while keeping per-token compute roughly constant (Shazeer et al., 2017; Jacobs et al., 1991). In MoE, a learned router activates only a small subset of experts—typically feed-forward networks—for each token (Fedus et al., 2022). From early sparsely gated layers to modern large language models (Du et al., 2022; Fedus et al., 2022; Lepikhin et al., 2020; Zoph et al., 2022; Dai et al., 2024), this design has delivered strong accuracy–efficiency trade-offs. Nevertheless, a fundamental challenge remains: expert specialization progressively deteriorates during training, with tokens routed to different experts exhibiting excessive uniformity and overlap, leading multiple experts to learn redundant knowledge (Dai et al., 2024). This redundancy confronts the router with ambiguous choices among functionally equivalent experts, eroding token-to-expert boundaries and substantially underutilizing model capacity.

Recent work has sought to encourage specialization through architectural modifications. DeepSeekMoE (Dai et al., 2024), HMoE (Wang et al., 2025a), and MoDSE (Sun et al., 2024) redesign expert layouts (e.g., shared or heterogeneous experts) to better match token complexity and balance load, while large-scale routed variants such as Mixtral (Jiang et al., 2024), Mixture of a Million Experts (He, 2024), and ReMoE (Wang et al., 2025b) adjust layer composition, expert granularity, or routing mechanisms to improve accuracy–efficiency trade-offs. These approaches primarily modify architectures and routers and rely on intra-layer dynamics, leaving open whether expert specialization itself can be treated as a first-class training objective.

In contrast to architectural modifications, this paper adopts an orthogonal, loss-centric perspective:

Treating expert specialization as a primary training objective rather than a structural property.

[†]Corresponding emails: kunyuan@pku.edu.cn (K. Yuan), 123sssmmm@gmail.com (M. Sun)

This approach directly shapes expert behavior and complements prior structural innovations. To design these losses, we identify two failure modes of specialization: **(1) Expert Overlap**, where different experts produce nearly identical activations for the same tokens, creating redundancy. **(2) Routing Ambiguity**, where similar inputs are dispatched inconsistently across experts, indicating ill-defined routing rules. When either occurs, experts collapse toward overlapping knowledge while the router faces ambiguous choices among functionally equivalent experts, undermining the core principle of specialization.

To address these failures, we introduce two complementary regularization loss functions that work in concert:

(L1) Intra-Layer specialization loss: This loss penalizes high cosine similarity between different experts' activations for the same token. It discourages functional redundancy and pushes each expert in a layer to develop unique specialization. This targets **Expert Overlap** by discouraging redundant responses from co-activated experts.

(L2) Cross-Layer coupling loss: This loss promotes coherent routing across adjacent layers by maximizing the joint probability of top-ranked expert pairs. By encouraging tokens to follow consistent expert sequences through depth, termed expert paths, it sharpens routing distributions, lowers entropy, and enables system-level optimizations such as path-aware placement and caching. This mitigates **Routing Ambiguity** and strengthens depth-wise specialization.

Together, these loss functions translate our diagnosed failure modes into targeted supervision, producing experts that are both functionally distinct within layers and coherently utilized across them.

Theoretical analysis. We analyze how the two losses shape expert updates and routing. The intra-layer specialization loss drives co-activated experts toward nearly orthogonal activations and gradients, yielding distinct learning trajectories. The cross-layer coupling loss propagates this specialization across depth under a mild continuity assumption on adjacent-layer representations. Together, the two losses induce a feedback loop in which weak specialization margins sharpen routing, decisive routing purifies each expert's training distribution, and these effects remain compatible with standard load-balancing regularizers.

Empirical evaluation. We provide targeted, mechanism-aligned sanity checks in Section 6. Comprehensive LLM-scale training and inference evaluations—including ablations and robustness—are deferred to Appendix D.

Core contributions. Our contributions are listed as follows:

(C1) Loss-centric specialization. We address expert overlap and routing ambiguity with two complementary regularization losses: an intra-layer term penalizing same-token activation similarity and a cross-layer term encouraging coherent expert paths through depth.

(C2) Theory for specialization and coupling. We theoretically show that the intra-layer specialization loss drives co-activated experts toward orthogonality and that cross-layer coupling propagates specialization across depth while remaining compatible with load-balancing objectives.

(C3) Evidence. We include targeted, mechanism-aligned sanity checks in the main text to validate the two regularizers. Comprehensive LLM-scale evaluations—covering both training and inference/efficiency, together with ablations and robustness studies

Related works. We discuss the prior works which are closely related to our proposed approach in Appendix A.

2 Mixture-of-Experts Models: Preliminaries

MoE layer. An MoE layer replaces the dense feed-forward network (FFN) in a Transformer block with E experts and a router. For the i-th token at layer l, the computation proceeds in three steps.

Step 1. Routing. Let $x_i^{(l)} \in \mathbb{R}^h$ denote the input representation. The router computes a logit $q_i^{(l,e)}$ for each expert e using a learnable routing vector $r^{(l,e)} \in \mathbb{R}^h$, then applies softmax to obtain routing

scores

$$q_i^{(l,e)} = \langle x_i^{(l)}, r^{(l,e)} \rangle, \quad s_i^{(l,e)} := \frac{\exp(q_i^{(l,e)})}{\sum_{j=1}^{E} \exp(q_i^{(l,j)})}, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product.

Step 2. Expert processing. The router activates the top-k experts $A_i^{(l)} \subseteq \{1, \ldots, E\}$ for token $x_i^{(l)}$. Each expert is an FFN, typically implemented as SwiGLU, with parameters $(W_{\text{gate}}^{(l,e)}, W_{\text{up}}^{(l,e)}, W_{\text{down}}^{(l,e)})$. The expert computation is:

$$z_i^{(l,e)} = \text{Swish}\left(W_{\text{gate}}^{(l,e)} x_i^{(l)}\right) \odot \left(W_{\text{up}}^{(l,e)} x_i^{(l)}\right), \quad y_i^{(l,e)} = W_{\text{down}}^{(l,e)} z_i^{(l,e)}, \tag{2}$$

where $z_i^{(l,e)}$ denotes the intermediate activation, $y_i^{(l,e)}$ is the expert output, and $\odot$ denotes element-wise multiplication.

Step 3. Combination. The layer output is a weighted sum of activated experts:

$$y_i^{(l)} = \sum_{e \in \mathbb{A}_i^{(l)}} s_i^{(l,e)} \, y_i^{(l,e)}. \tag{3}$$

Properties of well-behaved MoE. Putting these pieces together, we use "well-behaved MoE" as convenient shorthand for layers that approximately satisfy two qualitative properties: (i) experts are functionally disjoint on the token distribution, and (ii) routing is decisive and low-entropy. These properties are not formal requirements but rather capture the behaviors our losses are designed to encourage.

When these properties break down, we observe the two empirical failure modes introduced in Section 1. Expert overlap arises when experts in $A_i^{(l)}$ produce highly similar activations $z_i^{(l,e)}$ on the same token, rendering their contributions nearly mergeable and wasting model capacity. Routing ambiguity occurs when near-identical inputs $x_i^{(l)}$ are dispatched to different experts, blurring token–expert boundaries and preventing experts from developing distinct roles. When either failure mode occurs, experts collapse toward redundant representations while the router faces ambiguous choices among functionally equivalent experts. The intra-layer specialization loss (Section 3) and the cross-layer coupling loss (Section 4) are designed to directly counteract these two pathologies.

3 Intra-layer specialization loss

This section introduces an intra-layer regularization loss that targets *expert functional overlap* within each MoE layer. Standard load-balancing losses enforce even utilization, but do not prevent multiple experts from learning redundant transformations on the *same* tokens, which weakens the intended division of labor in MoE.

Functional view of expert overlap. Following the view of FFN blocks as concept extractors and writers to the residual stream (Geva et al., 2022), we decompose each expert into an intermediate activation $z_i^{(\ell,e)}$ and a down-projection $W_{\text{down}}^{(\ell,e)}$ as in Eq. (2). If two co-activated experts produce nearly identical intermediates $z_i^{(\ell,e)} \approx z_i^{(\ell,\nu)}$ on the same token, their contributions become algebraically mergeable on that token and the experts are functionally redundant.

Loss definition. Motivated by this view, we discourage same-token similarity in the intermediate activations. For token x_i we define the intra-layer specialization loss

$$\mathcal{R}_{\text{sp}}(x_i) = \sum_{\ell=1}^{L} \sum_{e \neq \nu \in \mathbb{A}_i^{(l)}} \left[\cos\big(z_i^{(\ell,e)}, \, z_i^{(\ell,\nu)}\big) \right]^2. \tag{4}$$

where $\mathbb{A}_i^{(l)}$ is the set of experts activated for x_i at layer ℓ. Squaring the cosine emphasizes highly overlapping pairs while keeping the penalty smooth. Crucially, the specialization loss $\mathcal{R}_{\text{sp}}$ only acts on experts that are *co-activated on the same token and layer*, leaving shared features across different tokens or contexts unconstrained.

Effect on expert updates. Beyond interpretability, the intermediate activation $z_i^{(\ell,e)}$ also governs how expert e is updated through its down-projection matrix $W_{\text{down}}^{(\ell,e)}$. The next proposition makes this connection explicit.

Proposition 3.1 (Activation-gradient alignment)**.** *For any two activated experts $e, \nu \in \mathbb{A}_i^{(\ell)}$, the cosine similarity between the gradients of the total loss $\mathcal{L}$ with respect to their down-projection matrices satisfies*

$$\cos\left(\frac{\partial \mathcal{L}}{\partial W_{\text{down}}^{(\ell,e)}}, \frac{\partial \mathcal{L}}{\partial W_{\text{down}}^{(\ell,\nu)}}\right) = \cos\left(z_i^{(\ell,e)}, z_i^{(\ell,\nu)}\right), \tag{5}$$

where $z_i^{(\ell,e)}$ and $z_i^{(\ell,\nu)}$ denote the corresponding intermediate activations. (The proof is provided in Appendix B.1.)

Proposition 3.1 links representation geometry to optimization dynamics: for co-activated experts on the same token, the cosine similarity of their activations $z_i^{(\ell,\cdot)}$ *exactly* equals the cosine similarity of their W_{down} gradients. We summarize the key implication for expert updates as follows:

Penalizing same-token activation similarity makes co-activated experts' W_{down} gradients more orthogonal, driving distinct learning trajectories and strengthening intra-layer functional specialization.

4 Cross-layer coupling loss

This section introduces a cross-layer coupling loss to address routing ambiguity and propagate expert specialization across depth. When near-identical tokens are dispatched to different experts, each expert receives a mixed and largely overlapping data distribution; consequently, their gradients become correlated and updates drive them toward similar functionality. Without stable, consistent assignments, experts cannot develop distinct roles, token–expert boundaries remain blurred, and the intended division of labor in MoE collapses into redundant behavior.

Cross-layer coupling in pre-trained MoE models. Recent work has identified an emergent property in MoE known as *cross-layer coupling* (Cai et al., 2024; Yao et al., 2024): routing decisions in adjacent layers are strongly correlated, such that the expert activated at layer l is highly predictive of the expert activated at layer $l+1$. During training, models spontaneously develop such structured pathways, forming coherent information pipelines through depth.

Cross-layer coupling as a specialization amplifier. Cross-layer coupling clearly promotes routing stability: when tokens consistently traverse fixed expert sequences (e.g., "expert 3 in layer 7 followed by expert 5 in layer 8"), routing ambiguity is eliminated by definition. However, its effect on expert specialization is less obvious. Specifically, we ask: *how does inter-layer structural consistency influence intra-layer expert differentiation?* In other words, if tokens follow stable paths across layers, does this help individual experts within each layer become more specialized? Our analysis reveals that the answer is yes, through a propagation mechanism formalized below.

Proposition 4.1. *Let $\mathbb{A}_i^{(l)}$ denote the set of activated experts for token x_i at layer l. Consider two adjacent layers l and $l+1$ that satisfy the following conditions:*

1. Representation continuity. *For a token x_i, its representations evolve smoothly across layers: $\cos(x_i^{(l)}, x_i^{(l+1)}) \geq 1 - \delta^2$ for small $\delta \in (0,1)$.*

2. Source-layer specialization. *Layer l exhibits expert specialization with nearly orthogonal router weights: for experts $e_1 \in \mathbb{A}_i^{(l)}$ and $e_2 \in \mathbb{A}_j^{(l)}$ processing different tokens $x_i \neq x_j$, we have $|\cos(r^{(l,e_1)}, r^{(l,e_2)})| \leq \varepsilon$ for a small $\varepsilon \in (0,1)$.*

3. Strong cross-layer coupling. *Adjacent layers exhibit stable expert pathways with high routing correlation. For any expert $e \in \mathbb{A}_i^{(l)}$ activated by token x_i, there exists a corresponding expert $\nu \in \mathbb{A}_i^{(l+1)}$ such that both routing decisions are confident: $\cos(x_i^{(l)}, r^{(l,e)}) \geq 1 - \iota^2$ and $\cos(x_i^{(l+1)}, r^{(l+1,\nu)}) \geq 1 - \iota^2$ for small $\iota \in (0,1)$.*

Under these conditions, layer $l+1$ inherits the specialization structure from layer l:

$$\left|\cos\big(r^{(l+1,\nu_1)}, r^{(l+1,\nu_2)}\big)\right| \leq \varepsilon + O(\delta, \iota) \tag{6}$$

for experts $\nu_1 \in \mathbb{A}_i^{(l+1)}$ and $\nu_2 \in \mathbb{A}_j^{(l+1)}$ processing different tokens, where the error term $O(\delta, \iota)$ vanishes as δ and ι decrease to 0 (proof in Appendix B.2).

Proposition 4.1 shows that when layer l has well-specialized experts (Condition 2) and is strongly coupled to layer $l+1$ (Condition 3), the specialization structure is transferred to the adjacent layer with bounded degradation (Eq. (6)). Localized specialization can therefore cascade through depth, yielding globally specialized representations. This motivates the following summary.

*Cross-layer coupling acts as a **specialization amplifier**: it transforms localized expert differentiation into network-wide functional diversity by creating stable pathways that propagate specialization across depth.*

Cross-layer coupling loss. Although cross-layer coupling emerges naturally, it develops slowly and incompletely, especially early in training when routing ambiguity is severe. Rather than waiting for it to appear organically, we introduce a coupling regularizer $\mathcal{R}_{\text{cp}}$ that directly maximizes joint routing probabilities between adjacent layers. For each token x_i, we define the pathway strength between expert e in layer l and expert ν in layer $l+1$ as $P_i^{(l,(e,\nu))} = s_i^{(l,e)} s_i^{(l+1,\nu)}$, the product of their routing scores. The loss focuses on the Top-k strongest cross-layer connections for each activated expert:

$$\mathcal{R}_{\text{cp}}(x_i) = -\sum_{l=1}^{L-1}\sum_{e=1}^{E}\sum_{\nu\in\mathbb{T}_i^{(l,e)}} P_i^{(l,(e,\nu))}, \quad \text{where} \quad P_i^{(l,(e,\nu))} = s_i^{(l,e)} s_i^{(l+1,\nu)}. \tag{7}$$

Here $s_i^{(l,e)}$ is defined in Eq. (1), and $\mathbb{T}_i^{(l,e)}$ contains the k experts in layer $l+1$ with the highest joint probabilities with expert e. Minimizing $\mathcal{R}_{\text{cp}}$ encourages decisive, high-probability pathways between adjacent layers, and by Proposition 4.1 these pathways create the structural conditions for specialization to propagate throughout the network.

By concentrating probability mass on a few strong cross-layer expert pairs, $\mathcal{R}_{\text{cp}}$ reduces routing ambiguity and encourages consistent expert identities across depth. This lowers token-distribution overlap among experts, decreases gradient sharing, and promotes divergent specialization, while remaining complementary to standard load-balancing objectives discussed in Section 5.

5 Unified Training Objective and Theoretical Picture

Joint training objective. With the intra-layer specialization loss $\mathcal{R}_{\text{sp}}$ (Section 3) and the cross-layer coupling loss $\mathcal{R}_{\text{cp}}$ (Section 4), we train MoE models by adding them as plug-in regularizers on top of the standard MoE objective. For each token x_i, the full training objective is

$$\mathcal{L}_{\text{lb,sp,cp}}(x_i) := \underbrace{\mathcal{L}(x_i) + \mathcal{R}_{\text{lb}}(x_i)}_{\text{standard MoE objective}} + \underbrace{\lambda_{\text{sp}}\,\mathcal{R}_{\text{sp}}(x_i) + \lambda_{\text{cp}}\,\mathcal{R}_{\text{cp}}(x_i)}_{\text{proposed regularizers}}, \tag{8}$$

where $\mathcal{L}(x_i)$ is the language modeling loss, $\mathcal{R}_{\text{lb}}(x_i)$ is the standard load-balancing regularizer, and λ_{sp}, λ_{cp} control the strength of specialization and coupling regularization. We use $\mathcal{L}_{(\cdot)}$ to denote the full objective with the indicated regularizers (e.g., $\mathcal{L}_{\text{lb}}$, $\mathcal{L}_{\text{lb,sp}}$, $\mathcal{L}_{\text{lb,cp}}$, and $\mathcal{L}_{\text{lb,sp,cp}}$). In practice, router-stabilization terms such as z-loss can be included on top of Eq. (8); our two losses only reuse intermediate activations and routing scores already produced in the forward pass and therefore require no architectural or routing-code modifications.

Practical overhead. Both $\mathcal{R}_{\text{sp}}$ and $\mathcal{R}_{\text{cp}}$ are lightweight auxiliaries. $\mathcal{R}_{\text{sp}}$ only uses the Top-k activated experts per token and adds an $O(k^2 d)$ similarity computation that can reuse cached activations. $\mathcal{R}_{\text{cp}}$ is computed from scalar routing scores and introduces no additional matrix multiplications. A detailed compute/memory accounting is provided in Appendix F.

Core theoretical picture: why the two losses form a coherent system. Sections 3 and 4 introduced $\mathcal{R}_{\text{sp}}$ and $\mathcal{R}_{\text{cp}}$ as direct supervision signals for expert overlap and routing ambiguity. At a high

level, Proposition 3.1 shows that reducing same-token activation similarity directly decorrelates co-activated experts' update directions through W_{down}, while Proposition 4.1 shows that strong cross-layer coupling can propagate specialization structure across depth with bounded degradation. Moreover, our regularizers remain compatible with standard load balancing (formal constructions in Appendix C). We also empirically verify that adding $\mathcal{R}_{\text{sp}}$ and $\mathcal{R}_{\text{cp}}$ does not destabilize load balancing by analyzing load-balance loss curves throughout training (Appendix D.1.5).

5.1 A CLOSED-LOOP MECHANISM: SPECIALIZATION AND ROUTING SHARPEN EACH OTHER

Key quantities. To make the closed-loop statement precise, we summarize three simple quantities that will be used both in theory and in our empirical diagnostics. For clarity, we describe them in a top-1 routing view (the intuition extends to top-k). Let $g_e(t)$ denote the router softmax probability for expert e given token t, and let $\ell_e(t)$ be the per-token loss incurred by expert e. Define the best expert $e^\star(t) = \arg\min_e \ell_e(t)$ when it is unique, and the token-wise loss margin

$$\Delta(t) \;:=\; \min_{e \neq e^\star(t)} \big(\ell_e(t) - \ell_{e^\star(t)}(t)\big) \;\in [0, \infty). \tag{9}$$

A larger $\Delta(t)$ indicates a clearer "best expert" advantage, which tends to promote routing decisiveness under the standard mixture objective. We quantify routing decisiveness using router entropy

$$H(g(t)) \;:=\; -\sum_{e=1}^{E} g_e(t) \log g_e(t), \tag{10}$$

where lower entropy corresponds to more decisive routing. Finally, to connect with the cross-layer story, denote by $C_\ell(t) \in \{1, \ldots, E\}$ the selected expert at layer ℓ under top-1 routing. We measure adjacent-layer pathway consistency by a permutation-invariant coupling coefficient

$$\kappa_{\ell \to \ell+1} \;:=\; \max_{\pi \in S_E} \; \mathbb{P}_{t \sim D}\big[\pi\big(C_\ell(t)\big) = C_{\ell+1}(t)\big], \tag{11}$$

where S_E is the symmetric group (expert relabeling), and larger $\kappa_{\ell \to \ell+1}$ means more consistent expert identities across depth.

Closed-loop intuition. Our key theoretical contribution is a self-reinforcing mechanism linking specialization and routing: (i) if one expert is even mildly better on a token, the router gradient tends to increase its probability, sharpening routing; (ii) once routing is sharp and stable, each expert effectively trains on a purer subset of tokens, which amplifies its regional advantage and strengthens specialization. This loop explains why the two losses are complementary: $\mathcal{R}_{\text{sp}}$ helps create and maintain meaningful expert advantages by discouraging within-layer overlap, while $\mathcal{R}_{\text{cp}}$ stabilizes token–expert paths across depth so that purity and specialization persist and propagate.

5.1.1 DIRECTION I: WEAK ADVANTAGE INDUCES DECISIVE ROUTING

Weak advantage ⇒ routing sharpness. The first direction formalizes that the standard soft mixture objective already prefers decisive routing: once there is a nontrivial best-expert advantage on most tokens, router updates increase the best expert's logit and reduce entropy. We state an informal version here and defer the full proof to Appendix C.1.

Theorem 5.1 (Weak advantage sharpens routing (informal; formal in Appendix C.1))**.** *Assume that for most tokens the best expert $e^\star(t)$ is unique and enjoys a nontrivial loss margin $\Delta(t)$ with high probability. Under a locally-constant expert-loss approximation with respect to router logits, gradient descent increases the best expert's logit. Moreover, if the mixture loss stays close to the best-expert loss on average, then for a large fraction of tokens the router places nearly all mass on $e^\star(t)$, i.e., $g_{e^\star(t)}(t) \geq 1 - \delta$, which implies low routing entropy.*

Interpretation. As soon as experts are even mildly differentiated, the router is biased toward becoming decisive. Our $\mathcal{R}_{\text{sp}}$ encourages such differentiation by reducing within-layer overlap, making best-expert advantages more persistent and easier for routing to amplify.

5.1.2 DIRECTION II: DECISIVE ROUTING AMPLIFIES SPECIALIZATION

Routing sharpness ⇒ specialization amplification. The reverse direction shows that decisive routing is a data-selection mechanism: each expert is trained on an effective distribution weighted by

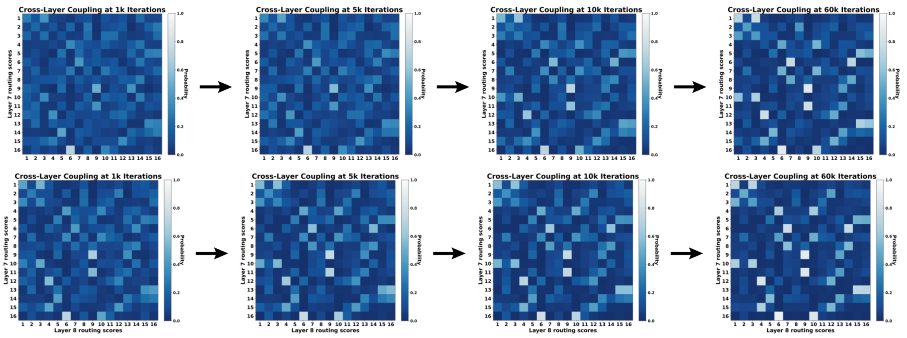


Figure 1: Conditional activation probabilities between experts in layers 7 and 8 for a 0.4B MoE model. *Top:* training with only load-balance regularization. *Bottom:* training with both load-balance and coupling regularization.

routing probabilities, and when routing is sharp and aligned, this distribution becomes high-purity (concentrated on tokens where the expert already has advantage). Any optimization progress then translates into a strictly improved regional advantage. We again present an informal statement and defer the full proof to Appendix C.1.

Theorem 5.2 (Clear routing amplifies regional specialization (informal; formal in Appendix C.1))**.** *If expert e receives a high-purity effective training distribution (most routed tokens lie in its advantage region), then any nontrivial decrease in its effective risk implies a strict decrease of its region-conditional risk on that region, provided the impurity is sufficiently small.*

Interpretation. Decisive routing reduces gradient interference by making each expert train on a cleaner subset of tokens. This widens regional advantages over time and reinforces future routing sharpness, closing the loop with Theorem 5.1.

5.1.3 Compatibility with load balancing

The proposed objectives are orthogonal to standard load balancing: balanced partitions can coexist with specialization, and the coupling objective admits optima under perfect balance. We defer the formal constructions to Appendix C.3.2.

5.2 Cross-layer coupling as a stability signal

Coupling stabilizes and propagates the loop across depth. The closed-loop mechanism above is within-layer and local in depth. Cross-layer coupling strengthens it by stabilizing token–expert identities across layers and enabling routing structure to transfer across depth. Here we introduce a new theoretical modeling of cross-layer coupling that both explains *why* such coupling emerges in pre-trained MoE models and further formalizes *how* coupling acts as a stability signal that propagates the specialization–routing loop across depth.

A token-distribution prior view (intuition). A useful way to understand why coupling emerges is to view routing as approximating a shared *token-distribution prior* induced by the data distribution itself. Tokens are not uniformly distributed in representation space: they form latent semantic/functional groups (e.g., reasoning patterns, syntactic structures, domain-specific knowledge), and different experts are naturally better suited for different groups. This induces an (implicit) latent assignment

$$A : \mathcal{T} \to \{1, \dots, E\},$$

which partitions tokens into E groups up to permutation. We do *not* assume A is known or globally optimal—it is a stylized abstraction capturing the data-induced routing structure that routers at different layers tend to learn.

Why coupling becomes stronger in deeper layers. A key empirical observation is that cross-layer coupling becomes stronger in deeper layers . (See Fig. 3 in Appendix, where deeper-layer pairs exhibit sharper coupling patterns .) Our modeling provides a simple explanation. Let $h_\ell(t)$ denote the token representation at layer ℓ. If the latent assignment $A(t)$ reflects data-induced structure in representation space, then the best achievable *linear* routing error at layer ℓ is

$$\alpha_\ell \;\coloneqq\; \inf_{W_\ell \in \mathbb{R}^{E \times d_\ell}} \mathbb{P}_{t \sim D}\Big[\arg\max_e \langle W_{\ell,e}, h_\ell(t)\rangle \neq A(t)\Big].$$

Deep networks typically learn progressively more linearly separable representations with depth, which empirically and theoretically supports $\alpha_{\ell+1} \leq \alpha_\ell$ in many regimes (e.g., Zhang et al. (2023)).

Since standard MoE routers are approximately linear classifiers on $h_\ell(t)$, later-layer routers can more faithfully align with $A(t)$ (up to permutation), so adjacent layers increasingly agree on expert identities, yielding stronger coupling deeper in the model.

Coupling as a backward-transfer signal. If layer $\ell{+}1$ is (on average) a better approximation to $A(t)$, then its routing decisions can serve as a self-supervised teacher signal for layer ℓ. Using the permutation-invariant coupling coefficient $\kappa_{\ell\to\ell+1}$ in Eq. (11), we formalize a simple backward-transfer guarantee: if layer $\ell{+}1$ routing approximates $A(t)$ and $\kappa_{\ell\to\ell+1}$ is large, then after alignment layer ℓ cannot drift far from that allocation. We state an informal version here and defer the full proof (and an empirical proxy validating the token-partition prior) to Appendix C.2.

Theorem 5.3 (Backward transfer via cross-layer coupling (informal; formal in Appendix C.2))**.** *There exists a permutation $\pi^\star$ such that the aligned routing decision $\pi^\star(C_\ell(t))$ satisfies*

$$\mathbb{P}_{t\sim D}\big[\pi^\star\big(C_\ell(t)\big) \neq A(t)\big] \;\le\; \varepsilon_{\ell+1} + \big(1-\kappa_{\ell\to\ell+1}\big),$$

where $A(t)$ is the latent assignment and $\varepsilon_{\ell+1}$ is the routing error of layer $\ell{+}1$ relative to A.

Interpretation. Cross-layer coupling turns later-layer routing into a stability/teacher signal for earlier layers: when $\kappa_{\ell\to\ell+1}$ is close to 1, aligning layer ℓ to layer $\ell{+}1$ transfers routing structure backward (up to permutation), reducing ambiguity early in training and stabilizing token–expert identities across depth. Our coupling loss $\mathcal{R}_{\text{cp}}$ explicitly increases adjacent-layer agreement, thereby stabilizing pathways and making the specialization–routing loop persist and propagate across depth.

6 Targeted Sanity Checks

We provide two minimal experiments that directly validate the empirical claims behind $\mathcal{R}_{\text{sp}}$ and $\mathcal{R}_{\text{cp}}$. Full-scale evaluations including LLMs pre-training and inference are deferred to Appendix D.

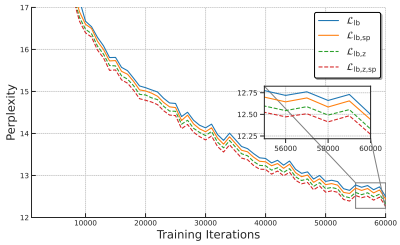


Figure 2: The perplexity for training a 1.1B model with different regularization. Setup is in Table 1.

Effect of $\mathcal{R}_{\text{sp}}$ (intra-layer overlap). To validate that $\mathcal{R}_{\text{sp}}$ captures specialization and improves training, we pre-train a 1.1B MoE model (100M activated parameters) with and without this regularizer while keeping all other settings identical. We denote by $\mathcal{L}_{(\cdot)}$ the full training objective, i.e., the language-modeling loss together with the indicated regularization terms. We evaluate four configurations: $\mathcal{L}_{\text{lb}}$ (load balancing only), $\mathcal{L}_{\text{lb,sp}}$ (load balancing + specialization), $\mathcal{L}_{\text{lb,z}}$ (load balancing + z-loss (Zoph et al., 2022)), and $\mathcal{L}_{\text{lb,z,sp}}$ (all three losses involved). Figure 2 shows that incorporating $\mathcal{R}_{\text{sp}}$ consistently reduces perplexity, with the combined $\mathcal{L}_{\text{lb,z,sp}}$ achieving the best performance.

Effect of $\mathcal{R}_{\text{cp}}$ (cross-layer coupling). To verify that cross-layer coupling is both natural and worth amplifying, we pre-train a 0.4B MoE model with 80M activated parameters and monitor conditional activation probabilities between adjacent layers. As shown in Figure 1, a clear coupling structure is already present early in training and becomes more pronounced over time, confirming that structured expert paths are an intrinsic feature of MoE learning.

7 Conclusion

We proposed two plug-and-play losses that directly optimize expert specialization in MoE models. The intra-layer specialization loss ($\mathcal{R}_{\text{sp}}$) penalizes activation similarity between experts processing identical tokens, while the cross-layer coupling loss ($\mathcal{R}_{\text{cp}}$) maximizes joint routing probabilities across adjacent layers to establish coherent expert pathways. These losses require no architectural modifications, integrate seamlessly with existing objectives, and are theoretically grounded. Experiments show consistent improvements across scales and MoE variants, along with higher inference throughput from more stable expert paths.

Acknowledgments

This work is supported by Zhejiang Lab, the National Key Research and Development Program of China (No. 2024YFA1012902), National Natural Science Foundation of China (No. 12288101, 92370121, 12301392, W2441021) and AI for Science Institute, Beijing, China.

REFERENCES

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Ruisi Cai, Yeonju Ro, Geon-Woo Kim, Peihao Wang, Babak Ehteshami Bejnordi, Aditya Akella, Zhangyang Wang, et al. *Read-ME*: Refactorizing llms as router-decoupled mixture of experts with system co-design. *Advances in Neural Information Processing Systems*, 37:116126–116148, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukas Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Such, David Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H Guss, Alex Nichol, Carlo Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, S Balaji, S Jain, William Saunders, Christopher Hesse, Amariah Carr, Jan Leike, Joshua Achiam, Vedant Misra, E Morikawa, Alec Radford, M Knight, Miles Brundage, Mira Murati, B Mayer, Peter Welinder, Bob McGrew, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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.

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 *NAACL*, 2019.

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. In *arXiv preprint arXiv:1803.05457*, 2018.

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

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 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. *International Conference on Machine Learning*, pp. 5547–5569, 2022.

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.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pp. 30–45, 2022.

Hongcan Guo, Haolang Lu, Guoshun Nan, Bolun Chu, Jialin Zhuang, Yuan Yang, Wenhao Che, Sicong Leng, Qimei Cui, and Xudong Jiang. Advancing expert specialization for better moe. *arXiv preprint arXiv:2505.22323*, 2025a.

Yongxin Guo, Zhenglin Cheng, Xiaoying Tang, Zhaopeng Tu, and Tao Lin. Dynamic mixture of experts: An auto-tuning approach for efficient transformer models. In *International Conference on Learning Representations*, 2025b.

Xu Owen He. Mixture of a million experts. *arXiv preprint arXiv:2407.04153*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Rizhen Hu, Yutong He, Ran Yan, Mou Sun, Binghang Yuan, and Kun Yuan. Mecefo: Enhancing llm training robustness via fault-tolerant optimization. *arXiv preprint arXiv:2510.16415*, 2025.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

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.

Boao Kong, Junzhu Liang, Yuxi Liu, Renjia Deng, and Kun Yuan. Cr-net: Scaling parameter-efficient training with cross-layer low-rank structure. *arXiv preprint arXiv:2509.18993*, 2025.

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.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 3214–3252, 2022.

Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. Minicache: Kv cache compression in depth dimension for large language models. *Advances in Neural Information Processing Systems*, 37:139997–140031, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2017.

Matteo Pagliardini, Amirkeivan Mohtashami, Francois Fleuret, and Martin Jaggi. Denseformer: Enhancing information flow in transformers via depth weighted averaging. *Advances in neural information processing systems*, 37:136479–136508, 2024.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

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. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

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.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Manxi Sun, Wei Liu, Jian Luan, Pengzhi Gao, and Bin Wang. Mixture of diverse size experts. *arXiv preprint arXiv:2409.12210*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

An Wang, Xingwu Sun, Ruobing Xie, Shuaipeng Li, Jiaqi Zhu, Zhen Yang, Pinxue Zhao, Weidong Han, Zhanhui Kang, Di Wang, et al. Hmoe: Heterogeneous mixture of experts for language modeling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21954–21968, 2025a.

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

Jinghan Yao, Quentin Anthony, Aamir Shafi, Hari Subramoni, and Dhabaleswar K DK Panda. Exploiting inter-layer expert affinity for accelerating mixture-of-experts model inference. In *2024 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 915–925. IEEE, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Chao Zhang, Xinyu Chen, Wensheng Li, Lixue Liu, Wei Wu, and Dacheng Tao. Understanding deep neural networks via linear separability of hidden layers. *arXiv preprint arXiv:2307.13962*, 2023.

Ruijie Zhang, Ziyue Liu, Zhengyang Wang, and Zheng Zhang. Lax: Boosting low-rank training of foundation models via latent crossing. *arXiv preprint arXiv:2505.21732*, 2025.

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.

Simiao Zuo, Xiaodong Liu, Jian Jiao, Young Jin Kim, Hany Hassan, Ruofei Zhang, Tuo Zhao, and Jianfeng Gao. Taming sparsely activated transformer with stochastic experts. *arXiv preprint arXiv:2110.04260*, 2021.

Appendix

A Related works

Here we discuss prior works which are closely related to our proposed approach.

Balancing losses and specialization objectives. A primary strategy to prevent routing collapse and improve stability in MoE training is to enforce balanced expert utilization. Early systems such as GShard (Lepikhin et al., 2020) and Switch (Fedus et al., 2022) introduced auxiliary load-balancing terms to distribute tokens across experts, with router z-loss (Zoph et al., 2022) providing additional stabilization. BASE layers (Lewis et al., 2021) formulated routing as an optimal linear assignment problem, achieving perfectly balanced usage without auxiliary terms. Expert-Choice routing (Zhou et al., 2022) further reversed the assignment process, allowing experts to select their Top-k tokens, which inherently balances load. These methods primarily regulate *how much* each expert is used. In contrast, our approach is complementary: we supervise *what* experts learn and *how* their paths align, introducing a within-layer similarity penalty to discourage activation overlap and a cross-layer coupling term to enforce coherence, while leaving existing balancing mechanisms intact.

Architectural and router-Level approaches. Another line of work promotes expert specialization by redesigning MoE architectures or router mechanisms. DeepSeekMoE (Dai et al., 2024) partitions experts more finely and introduces always-active shared experts, allowing routed specialists to focus on idiosyncratic patterns. Router-centric methods also refine gating: ReMoE (Wang et al., 2025b) replaces Top-k Softmax with a differentiable ReLU router and adaptive L_1 regularization, while Dynamic MoE (Guo et al., 2025b) auto-tunes both the number of activated experts per token and the size of the expert pool. Several structural variants further expand capacity and specialization. Mixtral layers multiple FFNs with top-2 routing, achieving strong accuracy–efficiency trade-offs (Jiang et al., 2024); Mixture of a Million Experts pushes expert granularity to the extreme (He, 2024); HMoE mixes experts of different sizes and biases usage toward smaller ones to encourage division of labor (Wang et al., 2025a); MoDSE deploys diverse-sized experts with pairwise allocation to stabilize routing and balance compute across devices (Sun et al., 2024); while simpler approaches such as Hash Layers (Roller et al., 2021) and THOR (Zuo et al., 2021) enforce balanced usage through fixed or randomized routing schemes. Unlike these methods—which modify layer composition or router design and largely rely on in-layer dynamics—our approach is architecture-agnostic. We impose explicit specialization objectives, namely a within-layer similarity penalty and a cross-layer coupling loss, on top of existing designs without altering attention, FFN, or router code paths.

Cross-layer signals and information. Recent work observes that routing decisions and activations exhibit strong cross-layer correlations. Read-ME precomputes routing decisions across layers and leverages inter-layer expert affinity to optimize scheduling and caching (Cai et al., 2024; Yao et al., 2024). Meanwhile, other studies demonstrate that inter-layer residuals—especially when exhibiting low-rank or redundant structures—can be harnessed to improve efficiency (Liu et al., 2024; Pagliardini et al., 2024; Kong et al., 2025; Zhang et al., 2025; Hu et al., 2025). These methods demonstrate that cross-layer structure is a robust empirical phenomenon, but they mainly use it for system or training efficiency rather than as an explicit learning signal. In contrast, our cross-layer coupling loss turns inter-layer routing affinity into a supervised objective: it encourages tokens to follow coherent expert paths through depth, thereby reinforcing intra-layer specialization while remaining compatible with standard load-balancing objectives.

B Proofs for Gradient Decorrelation and Cross-Layer Specialization Propagation.

In this section, we present the proofs for the proposed propositions in Section 3 and 4.

B.1 Proof of Proposition 3.1

Proposition B.1 (Proposition 3.1). *For any two activated experts $e, \nu \in \mathbb{A}_i^{(\ell)}$, the cosine similarity between the gradients of the total loss $\mathcal{L}$ with respect to their down-projection matrices satisfies*

$$\cos\left(\frac{\partial \mathcal{L}}{\partial W_{\text{down}}^{(\ell,e)}}, \frac{\partial \mathcal{L}}{\partial W_{\text{down}}^{(\ell,\nu)}}\right) = \cos\left(z_i^{(\ell,e)},\, z_i^{(\ell,\nu)}\right), \tag{12}$$

where $z_i^{(\ell,e)}$ and $z_i^{(\ell,\nu)}$ are the corresponding intermediate activations.

Proof. As the routing weights do not affect the cosine, without loss of generality we assume that the activated experts contribute with equal weights. Then the output of MoE blocks for layer l can be written as

$$E(x_i^{(l)}) := \sum_{e \in \mathbb{A}_i^{(l)}} y_i^{(l,e)}. \tag{13}$$

Then for any $e \in \mathbb{A}_i^{(l)}$ it holds that

$$\frac{\partial \mathcal{L}}{\partial y^{(l,e)}} = \frac{\partial \mathcal{L}}{\partial E(x_i^{(l)})}. \tag{14}$$

As $y_i^{(l,e)} = z_i^{(l,e)} W_{\text{down}}^{(l,e)}$, thus from (14) it comes for any $e \in \mathbb{A}_i^{(l)}$ that:

$$\frac{\partial \mathcal{L}}{\partial W_{\text{down}}^{(l,e)}} = \frac{\partial \mathcal{L}}{\partial y_i^{(l,e)}} \frac{\partial y_i^{(l,e)}}{\partial W_{\text{down}}^{(l,e)}} = \frac{\partial \mathcal{L}}{\partial E(x_i^{(l)})} z_i^{(l,e)}. \tag{15}$$

Using the Frobenius inner-product identity $\langle ab^\top,\, cd^\top \rangle_F = (a^\top c)(b^\top d)$ and $\|ab^\top\|_F = \|a\|_2 \|b\|_2$, we obtain that for $e_1, e_2 \in \mathbb{A}_i^{(l)}$ it holds that

$$\begin{aligned}
\cos\left(\frac{\partial \mathcal{L}}{\partial W_{\text{down}}^{(l,e_1)}}, \frac{\partial \mathcal{L}}{\partial W_{\text{down}}^{(l,e_2)}}\right) &= \frac{\left[\left(z_i^{(l,e_1)}\right)^\top z_i^{(l,e_2)}\right] \cdot \left[\frac{\partial \mathcal{L}}{\partial \left(y_i^{(l,e)}\right)^\top} \frac{\partial \mathcal{L}}{\partial y_i^{(l,e)}}\right]}{\left\|z_i^{(l,e_1)}\right\|_2 \cdot \left\|\frac{\partial \mathcal{L}}{\partial \left(y_i^{(l,e)}\right)}\right\|_2 \cdot \left\|z_i^{(l,e_2)}\right\|_2 \cdot \left\|\frac{\partial \mathcal{L}}{\partial \left(y_i^{(l,e)}\right)}\right\|_2} \\
&= \frac{z_i^{(l,e_1)} \left(z_i^{(l,e_2)}\right)^\top}{\left\|z_i^{(l,e_1)}\right\|_2 \left\|z_i^{(l,e_2)}\right\|_2} = \cos\left(z_i^{(l,e_1)}, z_i^{(l,e_2)}\right).
\end{aligned} \tag{16}$$

When condisering the case that each expert output is scaled by a positive routing weight, i.e., $\widetilde{y}_i^{(l,e)} = \alpha_i^{(l,e)} \cdot z_i^{(l,e)} W_{\text{down}}^{(l,e)}$, where $\alpha_i^{(l,e)} \in (0, 1]$ is the routing weight. Similar to (15), we can obtain that

$$\frac{\partial \mathcal{L}}{\partial W_{\text{down}}^{(l,e)}} = \alpha_i^{(l,e)} \cdot \frac{\partial \mathcal{L}}{\partial E(x_i^{(l)})} z_i^{(l,e)}.$$

Thus the common positive factor cancels in the cosine similarity, leaving the result unchanged. □

B.2 Proof of Proposition 4.1

Proposition B.2 (Proposition 4.1). *Let $\mathbb{A}_i^{(l)}$ denote the set of activated experts for token x_i at layer l. Consider adjacent layers l and $l+1$ satisfying:*

1. Representation continuity. *For a token x_i, its representations evolve smoothly across layers: $\cos(x_i^{(l)}, x_i^{(l+1)}) \geq 1 - \delta^2$ for small $\delta \in (0, 1)$.*

2. Source-layer specialization. *Layer l exhibits expert specialization with nearly orthogonal router weights: for experts $e_1 \in \mathbb{A}_i^{(l)}$ and $e_2 \in \mathbb{A}_j^{(l)}$ processing different tokens $x_i \neq x_j$, we have $|\cos(r^{(l,e_1)}, r^{(l,e_2)})| \leq \varepsilon$ for a small $\varepsilon \in (0,1)$.*

3. Strong cross-layer coupling. *Adjacent layers exhibit stable expert pathways with high routing correlation. For any expert $e \in \mathbb{A}_i^{(l)}$ activated by token x_i, there exists a corresponding expert $\nu \in \mathbb{A}_i^{(l+1)}$ such that both routing decisions are confident: $\cos(x_i^{(l)}, r^{(l,e)}) \geq 1 - \iota^2$ and $\cos(x_i^{(l+1)}, r^{(l+1,\nu)}) \geq 1 - \iota^2$ for small $\iota \in (0,1)$.*

Under these conditions, layer $l+1$ inherits the specialization structure from layer l:

$$\left|\cos\left(r^{(l+1,\nu_1)}, r^{(l+1,\nu_2)}\right)\right| \leq \varepsilon + O(\delta, \iota) \tag{17}$$

for experts $\nu_1 \in \mathbb{A}_i^{(l+1)}$ and $\nu_2 \in \mathbb{A}_j^{(l+1)}$ processing different tokens, where the error term $O(\delta, \iota)$ vanishes as δ and ι decrease to 0.

Proof. From the first Assumption, It can be obtained that:

$$\left\| \frac{x_i^{(l,e)}}{\left\|x_i^{(l,e)}\right\|} - \frac{x_i^{(l+1,e)}}{\left\|x_i^{(l+1,e)}\right\|} \right\|^2 = 2 - 2\cos\left(x_i^{(l,e)}, x_i^{(l+1,e)}\right) \leq 2\delta^2. \tag{18}$$

Similarly, it holds that:

$$\left\| \frac{x_i^{(l,e)}}{\left\|x_i^{(l,e)}\right\|} - \frac{r^{(l,e)}}{\left\|r^{(l,e)}\right\|} \right\|^2 \leq 2\iota^2, \quad \left\| \frac{x_i^{(l+1,\nu)}}{\left\|x_i^{(l+1,\nu)}\right\|} - \frac{r^{(l+1,\nu)}}{\left\|r^{(l+1,\nu)}\right\|} \right\|^2 \leq 2\iota^2. \tag{19}$$

Then from Eq. (18) and Eq. (19), it holds that

$$\begin{aligned} &\left\| \frac{r^{(l,e)}}{\left\|r^{(l,e)}\right\|} - \frac{r^{(l+1,\nu)}}{\left\|r^{(l+1,\nu)}\right\|} \right\| \\ \leq &\left\| \frac{x_i^{(l,e)}}{\left\|x_i^{(l,e)}\right\|} - \frac{x_i^{(l+1,e)}}{\left\|x_i^{(l+1,e)}\right\|} \right\| + \left\| \frac{x_i^{(l,e)}}{\left\|x_i^{(l,e)}\right\|} - \frac{r^{(l,e)}}{\left\|r^{(l,e)}\right\|} \right\| + \left\| \frac{x_i^{(l+1,\nu)}}{\left\|x_i^{(l+1,\nu)}\right\|} - \frac{r^{(l+1,\nu)}}{\left\|r^{(l+1,\nu)}\right\|} \right\| \\ \leq &\sqrt{2}\left(\delta + 2\iota\right). \end{aligned} \tag{20}$$

Then it holds that

$$\cos\left(r^{(l,e)}, r^{(l+1,\nu)}\right) = 1 - \frac{1}{2}\left\| \frac{r^{(l,e)}}{\left\|r^{(l,e)}\right\|} - \frac{r^{(l+1,\nu)}}{\left\|r^{(l+1,\nu)}\right\|} \right\|^2 \geq 1 - (\delta + 2\iota)^2. \tag{21}$$

Then we prove (6). Let

$$\begin{aligned} \tilde{r}^{(l,e_1)} &:= \frac{r^{(l,e_1)}}{\left\|r^{(l,e_1)}\right\|}, \quad \tilde{r}^{(l+1,\nu_1)} := \frac{r^{(l+1,\nu_1)}}{\left\|r^{(l+1,\nu_1)}\right\|}, \\ \tilde{r}^{(l,e_2)} &:= \frac{r^{(l,e_2)}}{\left\|r^{(l,e_2)}\right\|}, \quad \tilde{r}^{(l+1,\nu_2)} := \frac{r^{(l+1,\nu_2)}}{\left\|r^{(l+1,\nu_2)}\right\|}. \end{aligned}$$

Then it comes that:

$$\begin{aligned} &\left|\left\langle \tilde{r}^{(l+1,\nu_1)}, \tilde{r}^{(l+1,\nu_2)} \right\rangle\right| \\ = &\left|\left\langle \tilde{r}^{(l,e_1)}, \tilde{r}^{(l,e_2)} \right\rangle\right| + \left|\left\langle \tilde{r}^{(l,e_1)} - \tilde{r}^{(l+1,\nu_1)}, \tilde{r}^{(l,e_2)} \right\rangle\right| + \left|\left\langle \tilde{r}^{(l,e_1)}, \tilde{r}^{(l,e_2)} - \tilde{r}^{(l+1,\nu_2)} \right\rangle\right| \\ &+ \left|\left\langle \tilde{r}^{(l,e_1)} - \tilde{r}^{(l+1,\nu_1)}, \tilde{r}^{(l,e_2)} - \tilde{r}^{(l+1,\nu_2)} \right\rangle\right| \\ \leq &\varepsilon + 2\sqrt{2}\left(\delta + 2\iota\right) + 2\left(\delta + 2\iota\right)^2, \end{aligned} \tag{22}$$

where the last inequality is from (20). Then we finish the proof of this lemma. □

C Theory Details for Specialization, Routing Sharpness, and Cross-Layer Coupling

In this section, we aim to establish a comprehensive theoretical framework for the proposed specialization and coupling loss.

Notations. Throughout this appendix we consider a single MoE layer with $E \geq 2$ experts and **top-**1 **routing** ($k = 1$) at inference. Let $(t, y) \sim D$ be the data distribution. Each expert e incurs per-token loss

$$\ell_e(t) \;\coloneqq\; \ell\big(f_e(t), y\big),$$

and the router produces logits $z(t) = (z_1(t), \ldots, z_E(t))$ with softmax probabilities

$$g_e(t) \;=\; \frac{\exp(z_e(t))}{\sum_{j=1}^{E} \exp(z_j(t))}, \qquad e \in \{1, \ldots, E\}.$$

The standard soft-routing objective is the mixture loss

$$L(t) \;\coloneqq\; \sum_{e=1}^{E} g_e(t)\, \ell_e(t).$$

When the best expert is unique, we write $e^\star(t) \coloneqq \arg\min_e \ell_e(t)$ and define the (token-wise) margin

$$\Delta(t) \;\coloneqq\; \min_{e \neq e^\star(t)} \big(\ell_e(t) - \ell_{e^\star(t)}(t)\big) \in [0, \infty).$$

Section Roadmap. Section 5 in the main text presents a lightweight theoretical picture of how $\mathcal{R}_{\text{sp}}$ and $\mathcal{R}_{\text{cp}}$ interact with routing and load balancing. This section provides the formal counterparts: additional definitions, auxiliary quantities, and proofs that support the claims summarized in Section 5. Concretely:

- Appendix C.1 formalizes the specialization–routing mutual reinforcement mechanism (weak loss advantage ⇒ decisive routing, and decisive routing ⇒ specialization amplification), and derives an explicit entropy corollary.
- Appendix C.2 models cross-layer coupling as a learning signal, defines a permutation-invariant coupling coefficient, and proves a backward-transfer guarantee under strong coupling.
- Appendix C.3 provides sufficient conditions and constructions establishing compatibility with standard load balancing.

C.1 Specialization–Routing Mutual Reinforcement

C.1.1 Weak specialization ⇒ decisive routing

Theorem C.1 (Weak specialization implies high-probability decisive routing)**.** *We assume that*

(A1) ***(Local uniqueness)*** *For D-almost every token t, the best expert $e^\star(t)$ is unique.*

(A2) ***(Weak specialization: high-probability margin)*** *There exist $\gamma_0 > 0$ and $\varepsilon_0 \in [0, 1)$ such that*

$$\mathbb{P}_{(t,y)\sim D}\big[\Delta(t) \geq \gamma_0\big] \;\geq\; 1 - \varepsilon_0. \tag{23}$$

Consider a router update in which $\{\ell_e(t)\}_{e=1}^{E}$ are treated as locally constant w.r.t. the router logits $z(t)$. Then:

1. *For any token t with $\Delta(t) > 0$ and $g_{e^\star(t)}(t) < 1$,*

$$\frac{\partial L(t)}{\partial z_{e^\star(t)}(t)} \;<\; 0. \tag{24}$$

Consequently, a gradient-descent update increases $z_{e^\star(t)}(t)$.

2. *For any* $\delta \in (0,1)$,

$$\mathbb{P}_{(t,y)\sim D}\big[g_{e^\star(t)}(t) \geq 1-\delta\big] \;\geq\; 1-\varepsilon_0 - \frac{\mathbb{E}_{(t,y)\sim D}\big[L(t)-\ell_{e^\star(t)}(t)\big]}{\gamma_0\,\delta}. \tag{25}$$

Proof. Fix a token t and abbreviate $\ell_e = \ell_e(t)$, $g_e = g_e(t)$, $z_e = z_e(t)$, and $L = L(t)$. For the softmax, $\dfrac{\partial g_j}{\partial z_e} = g_j(\mathbf{1}\{j=e\} - g_e)$, hence

$$\frac{\partial L}{\partial z_e} = \sum_{j=1}^{E} \ell_j \frac{\partial g_j}{\partial z_e} = \sum_{j=1}^{E} \ell_j g_j(\mathbf{1}\{j=e\} - g_e) = \ell_e g_e - g_e \sum_{j=1}^{E} g_j \ell_j = g_e(\ell_e - L). \tag{26}$$

Let $e^\star = e^\star(t)$. If $\Delta(t) > 0$ and $g_{e^\star} < 1$, then there exists at least one $e \neq e^\star$ with $g_e > 0$ and $\ell_e > \ell_{e^\star}$. As $L = \sum_j g_j \ell_j$ is a convex combination with positive mass on a value strictly larger than $\ell_{e^\star}$, we have $L > \ell_{e^\star}$. Therefore $\frac{\partial L}{\partial z_{e^\star}} = g_{e^\star}(\ell_{e^\star} - L) < 0$, proving (24).

For the high-probability sharpness bound, on any token t where $e^\star(t)$ is unique and $\Delta(t) > 0$,

$$L(t) - \ell_{e^\star(t)}(t) = \sum_{e \neq e^\star(t)} g_e(t)\big(\ell_e(t) - \ell_{e^\star(t)}(t)\big) \geq \sum_{e \neq e^\star(t)} g_e(t)\,\Delta(t) = \Delta(t)\big(1 - g_{e^\star(t)}(t)\big), \tag{27}$$

which implies

$$1 - g_{e^\star(t)}(t) \;\leq\; \frac{L(t) - \ell_{e^\star(t)}(t)}{\Delta(t)}. \tag{28}$$

Let $G \coloneqq \{\Delta(t) \geq \gamma_0\}$. By (23), $\mathbb{P}(G^c) \leq \varepsilon_0$, where G^c denotes the complement of event. Then for any $\delta \in (0,1)$,

$$\mathbb{P}\big(g_{e^\star(t)}(t) < 1-\delta\big) = \mathbb{P}\big(1 - g_{e^\star(t)}(t) > \delta\big) \leq \mathbb{P}(G^c) + \mathbb{P}\big(1 - g_{e^\star(t)}(t) > \delta,\; G\big), \tag{29}$$

For the event G, inequality (28) yields $1 - g_{e^\star(t)}(t) \leq \dfrac{L(t) - \ell_{e^\star(t)}(t)}{\gamma_0}$, then we can obtain

$$\mathbb{P}\big(1 - g_{e^\star(t)}(t) > \delta,\; G\big) \leq \mathbb{P}\left(\frac{L(t) - \ell_{e^\star(t)}(t)}{\gamma_0} > \delta\right) = \mathbb{P}\big(L(t) - \ell_{e^\star(t)}(t) > \gamma_0\delta\big). \tag{30}$$

Applying Markov's inequality to the nonnegative random variable $L(t) - \ell_{e^\star(t)}(t)$ gives

$$\mathbb{P}\big(L(t) - \ell_{e^\star(t)}(t) > \gamma_0\delta\big) \leq \frac{\mathbb{E}\big[L(t) - \ell_{e^\star(t)}(t)\big]}{\gamma_0\delta}. \tag{31}$$

Combining the last three displays and using $\mathbb{P}(G^c) \leq \varepsilon_0$ yields

$$\mathbb{P}\big(g_{e^\star(t)}(t) \geq 1-\delta\big) \geq 1 - \varepsilon_0 - \frac{\mathbb{E}\big[L(t) - \ell_{e^\star(t)}(t)\big]}{\gamma_0\delta},$$

which proves (25). □

Theorem C.1 formalizes a simple but important point: the standard MoE mixture objective already prefers *decisive* routing. As soon as one expert attains even a weak per-token loss advantage, the router gradient increases that expert's logit, so probability mass moves toward the best expert. Moreover, if the mixture loss stays close to the best-expert loss on average (a small oracle gap), then the router must place nearly all mass on the best expert for most tokens, yielding low-entropy routing. This is exactly the direction in which $\mathcal{R}_{\mathrm{sp}}$ helps: by discouraging expert overlap, it makes such loss advantages more persistent and easier to amplify.

Remark C.2 (Why is the oracle gap typically small during training?). Let $G(t) := L(t) - \ell_{e^\star(t)}(t) \geq 0$ denote the *oracle gap* between the router's soft mixture loss $L(t)$ and the best-expert loss $\ell_{e^\star(t)}(t)$ at token t. In practice, $G(t)$ tends to shrink during training for two coupled reasons. First, router updates increase the logit (and hence probability) of the current best expert, which decreases the mixture loss $L(t)$. Second, expert updates reduce $\ell_e(t)$ on the tokens they repeatedly receive, which makes best-expert advantages more pronounced over time. Once experts become even mildly differentiated, this interaction naturally drives routing to become more decisive, pushing $L(t)$ closer to the oracle loss.

We next convert the high-probability sharpness event in Theorem C.1 into an explicit upper bound on router entropy, which serves as a convenient routing-clarity diagnostic.

Corollary C.3 (Entropy bound as a consequence of decisive routing)**.** *Let* $H(g(t)) := -\sum_{e=1}^{E} g_e(t) \log g_e(t)$. *For any* $\delta \in (0, 1/2]$, *on the event* $\{g_{e^\star(t)}(t) \geq 1 - \delta\}$ *we have*

$$H(g(t)) \;\leq\; h(\delta) + \delta \log(E-1), \qquad h(\delta) := -\delta \log \delta - (1-\delta)\log(1-\delta). \tag{32}$$

Consequently, with the same lower bound as in (25),

$$\mathbb{P}_{(t,y)\sim D}[H(g(t)) \leq h(\delta) + \delta \log(E-1)] \;\geq\; 1 - \varepsilon_0 - \frac{\mathbb{E}_{(t,y)\sim D}\big[L(t) - \ell_{e^\star(t)}(t)\big]}{\gamma_0\, \delta}. \tag{33}$$

Proof. Fix a token t and let $p := g_{e^\star(t)}(t)$. Consider all probability vectors $g \in \Delta^{E-1}$ with $g_{e^\star} = p$. Among these, the Shannon entropy $H(g)$ is maximized when the remaining mass $1-p$ is spread uniformly over the other $E-1$ coordinates, i.e., $g_e = \dfrac{1-p}{E-1}$ for all $e \neq e^\star$ (this follows from the strict concavity of $x \mapsto -x \log x$ and Jensen's inequality). Therefore,

$$H(g(t)) \leq -\,p \log p \;-\; \sum_{e \neq e^\star} \frac{1-p}{E-1} \log \frac{1-p}{E-1} = -\,p \log p \;-\; (1-p) \log \frac{1-p}{E-1}. \tag{34}$$

Rewriting gives

$$H(g(t)) \leq \underbrace{\big(-\,p \log p - (1-p)\log(1-p)\big)}_{= h(1-p)} + (1-p)\log(E-1) = h(1-p) + (1-p)\log(E-1). \tag{35}$$

As $p \geq 1-\delta$, we have $1-p \leq \delta \leq 1/2$, and since $h(\cdot)$ is nondecreasing on $[0, 1/2]$, it follows that $h(1-p) \leq h(\delta)$ and $(1-p)\log(E-1) \leq \delta \log(E-1)$. This proves (32). The probability statement follows by observing that the event $\{g_{e^\star(t)}(t) \geq 1-\delta\}$ implies the event $\{H(g(t)) \leq h(\delta) + \delta \log(E-1)\}$ and applying (25). □

Corollary C.3 converts the sharpness event $g_{e^\star}(t) \geq 1-\delta$ into an explicit upper bound on the router entropy. Combined with Theorem C.1, it provides a clean "loss margin ⇒ low entropy" link without introducing any extra entropy regularizer.

C.1.2 DECISIVE ROUTING ⇒ SPECIALIZATION AMPLIFICATION

We now formalize the reverse direction: *if routing is sufficiently clear and aligned, each expert trains on a cleaner distribution concentrated on tokens where it already has advantage, which forces its performance on that region to improve.* This result is stated in a deliberately weak (but assumption-light) form that captures the positive feedback mechanism.

Let $\{S_e^\star\}_{e=1}^{E}$ be measurable subsets of $\mathcal{T}$ that partition the token space up to D-null sets (i.e., $D(\cup_e S_e^\star) = 1$ and $D(S_e^\star \cap S_{e'}^\star) = 0$ for $e \neq e'$), interpreted as the current "advantage regions". Given router weights $g(e \mid t)$ and data distribution D, define the *effective training distribution* seen by expert e:

$$D_e(A) \;:=\; \frac{\mathbb{E}_{(t,y)\sim D}\big[g(e \mid t)\mathbf{1}\{(t,y) \in A\}\big]}{\mathbb{E}_{(t,y)\sim D}[g(e \mid t)]}, \qquad A \subseteq \mathcal{T} \times \mathcal{Y}. \tag{36}$$

Define the *purity* of expert e as

$$\alpha_e \;:=\; \mathbb{P}_{(t,y)\sim D_e}\big[t \in S_e^\star\big] = D_e\big(S_e^\star \times \mathcal{Y}\big). \tag{37}$$

Finally define the effective risk and the region-conditional risk:

$$R_e^{\mathrm{eff}}(f_e) \;:=\; \mathbb{E}_{(t,y)\sim D_e}\big[\ell(f_e(t), y)\big], \qquad R_e^{S}(f_e) \;:=\; \mathbb{E}_{(t,y)\sim D_e}\big[\ell(f_e(t), y) \mid t \in S_e^\star\big].$$

Theorem C.4 (Clear routing improves region-conditional risk (a weak specialization amplifier))**.** *Fix an expert* e *and assume:*

(B1) ***(High purity)*** $\alpha_e \geq 1 - \eta$ *for some* $\eta \in (0, 1)$.

*(B2) (**Effective optimization**) After a training phase, expert e achieves*

$$R_e^{\text{eff}}(f_e^{\text{new}}) \;\le\; R_e^{\text{eff}}(f_e^{\text{old}}) - \Delta_e \qquad \textit{for some } \Delta_e > 0. \tag{38}$$

*(B3) (**Bounded loss**) $0 \le \ell(\hat{y}, y) \le B$ almost surely for some $B > 0$.*

Then the region-conditional risk on the advantage region satisfies

$$R_e^S(f_e^{\text{new}}) \;\le\; R_e^S(f_e^{\text{old}}) - (\Delta_e - \eta B). \tag{39}$$

In particular, if $\Delta_e > \eta B$ and η is small (clear, well-aligned routing), then expert e must strictly improve on its advantage region $S_e^\star$ under $D_e(\cdot \mid t \in S_e^\star)$.

Proof. Under D_e, decompose the effective risk by conditioning on whether $t \in S_e^\star$:

$$R_e^{\text{eff}}(f_e) = \mathbb{E}_{D_e}\big[\ell(f_e(t), y)\big] = \alpha_e\, \mathbb{E}_{D_e}\big[\ell(f_e(t), y) \mid t \in S_e^\star\big] + (1 - \alpha_e)\, \mathbb{E}_{D_e}\big[\ell(f_e(t), y) \mid t \notin S_e^\star\big]. \tag{40}$$

Let $R_e^{\bar{S}}(f_e) \coloneqq \mathbb{E}_{D_e}[\ell(f_e(t), y) \mid t \notin S_e^\star]$. Define the changes

$$\Delta R^{\text{eff}} \coloneqq R_e^{\text{eff}}(f_e^{\text{new}}) - R_e^{\text{eff}}(f_e^{\text{old}}), \qquad \Delta R^S \coloneqq R_e^S(f_e^{\text{new}}) - R_e^S(f_e^{\text{old}}), \qquad \Delta R^{\bar{S}} \coloneqq R_e^{\bar{S}}(f_e^{\text{new}}) - R_e^{\bar{S}}(f_e^{\text{old}}). \tag{41}$$

By the above decomposition applied to f_e^{new} and f_e^{old} and subtracting, we get

$$\Delta R^{\text{eff}} = \alpha_e\, \Delta R^S + (1 - \alpha_e)\, \Delta R^{\bar{S}}. \tag{42}$$

Assumption (38) gives $\Delta R^{\text{eff}} \le -\Delta_e$. By boundedness (B3), both $R_e^{\bar{S}}(f_e^{\text{new}})$ and $R_e^{\bar{S}}(f_e^{\text{old}})$ lie in $[0, B]$, hence $\Delta R^{\bar{S}} \le B$. Therefore,

$$\alpha_e\, \Delta R^S = \Delta R^{\text{eff}} - (1 - \alpha_e) \Delta R^{\bar{S}} \le -\Delta_e + (1 - \alpha_e) B. \tag{43}$$

Using (B1), $1 - \alpha_e \le \eta$ and $\alpha_e \le 1$, yielding

$$\Delta R^S \le -\Delta_e + \eta B. \tag{44}$$

Rearranging gives (39). □

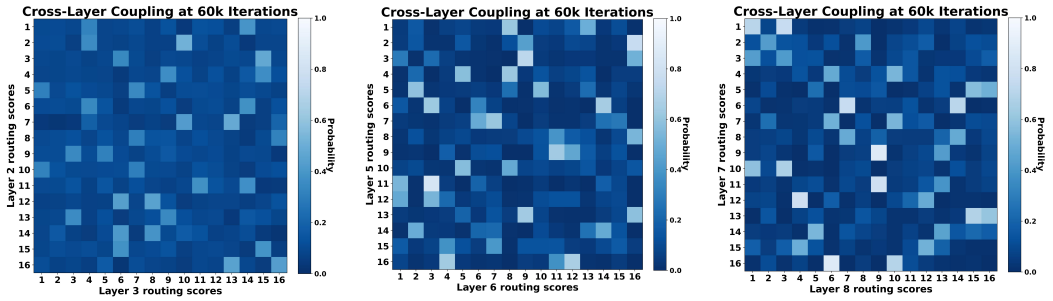
Theorem C.4 captures the reverse direction of the loop. Once routing is sharp and aligned, each expert's effective training distribution becomes *high-purity*: it is dominated by the subset of tokens where that expert already has comparative advantage. Under this higher-purity data distribution, progress on the effective risk must translate into progress on the expert's preferred region, so the expert's regional advantage widens over time. In short, decisive routing is not only about low entropy—it is a data-selection mechanism that reduces gradient interference and *amplifies* functional specialization.

C.2 Cross-layer coupling as a learning signal

So far we focused on the within-layer feedback mechanism; we next show how cross-layer coupling provides an additional learning signal that stabilizes and propagates these effects across depth.

Observation: cross-layer coupling. Recent work has identified an emergent property in MoE training known as *cross-layer coupling* (Cai et al., 2024; Yao et al., 2024): the expert activated at layer ℓ strongly predicts the expert activated at layer $\ell + 1$. Empirically, this phenomenon is visible early in training and becomes increasingly pronounced as optimization proceeds . These observations suggest that coupling is not merely an artifact of optimization, but reflects a meaningful structural regularity in how tokens are routed through depth.

A token-distribution prior view. We consider the **top-1 routing setting**, i.e., each token activates exactly one expert per layer. We provide an intuitive explanation for why such coupling naturally arises. Tokens are not uniformly distributed in semantic space; rather, they exhibit latent structure . For example, some tokens predominantly encode mathematical reasoning patterns, others linguistic



(a) Layers $2 \to 3$ (b) Layers $5 \to 6$ (c) Layers $7 \to 8$

Figure 3: Cross-layer coupling heatmaps. Coupling becomes more pronounced in deeper layers.

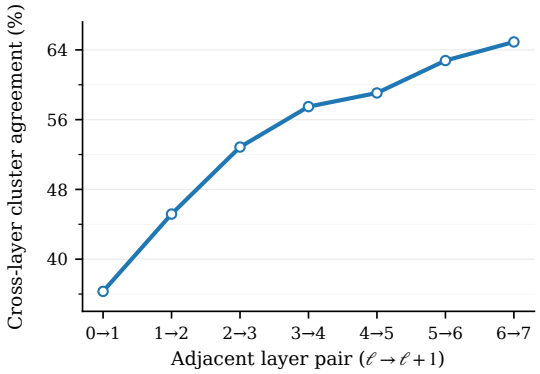


Figure 4: Cross-layer cluster agreement (%) for adjacent layer pairs $(\ell \to \ell{+}1)$ under top-1 routing.

syntax, and others domain-specific factual knowledge. This induces a *latent, relatively good token-to-expert allocation*: a partition of the token distribution into subsets for which different experts are more suitable .

Importantly, we do *not* assume this allocation is globally optimal or explicitly known. Instead, it should be viewed as a soft prior induced by the data distribution itself. During training, routers at different layers implicitly attempt to approximate this same underlying allocation. Cross-layer coupling emerges when multiple layers align to this shared token-distribution prior, up to permutations of expert identities .

Empirical justification of the prior. To justify the token-distribution prior, we approximate it by a natural partition of the pre-router token distribution at each layer and evaluate how consistent these partitions are across depth. Concretely, we use a 0.4B MoE with total experts $E{=}8$ and activated experts $k{=}1$ per layer. At every layer ℓ, we run k-means to cluster tokens into E groups and apply a distance-greedy balancing step to equalize cluster sizes. Because cluster indices are arbitrary per layer, we align clusters between layers ℓ and $\ell{+}1$ using the Hungarian algorithm (maximum bipartite matching over the inter-layer confusion matrix). We report the percentage of tokens whose cluster at layer ℓ matches the aligned cluster at layer $\ell{+}1$; the partition difference is 1 minus this agreement (Fig. 4).

Why coupling strengthens with depth. A key empirical observation is that cross-layer coupling becomes stronger in deeper layers . (See Fig. 3, where deeper-layer pairs exhibit sharper coupling patterns .) Our modeling provides a natural explanation for this phenomenon.

Let $h_\ell(t) \in \mathbb{R}^{d_\ell}$ denote the representation of token t at layer ℓ. For a given latent token-to-expert assignment $A(t)$ induced by the data distribution, define the best achievable linear routing error at

layer ℓ as

$$\alpha_\ell \;\coloneqq\; \inf_{W_\ell \in \mathbb{R}^{E \times d_\ell}} \mathbb{P}_{t\sim D}\Big[\arg\max_e \langle W_{\ell,e}, h_\ell(t)\rangle \neq A(t)\Big].$$

In deep networks, representations typically become more linearly separable with depth (Zhang et al., 2023), suggesting $\alpha_{\ell+1} \leq \alpha_\ell$ in many regimes.

Since standard MoE routers are approximately linear classifiers on $h_\ell(t)$, their routing error can be decomposed into: (i) a *representation-induced approximation error* (captured by α_ℓ), and (ii) an *optimization error* due to imperfect training. As depth increases, the approximation error decreases, allowing later-layer routers to more faithfully align with the latent token allocation. As a result, adjacent layers increasingly agree on routing decisions, leading to stronger cross-layer coupling in deeper parts of the network.

Using later-layer routing as a learning signal. The discussion above suggests that, on average, routing decisions at layer $\ell + 1$ are a more accurate proxy of the latent token allocation than those at layer ℓ. This motivates a simple but powerful idea: *use routing decisions from deeper layers as a self-supervised target to guide earlier layers*. By encouraging adjacent layers to agree on expert identities, we can propagate improved routing structure backward through depth, reducing routing ambiguity and accelerating specialization.

Coupling coefficient and backward transfer guarantee. For top-1 routing, let

$$C_\ell(t) \in \{1, \ldots, E\}, \qquad C_{\ell+1}(t) \in \{1, \ldots, E\}$$

denote the selected experts at layers ℓ and $\ell + 1$ for token t. We quantify cross-layer pathway consistency using the permutation-invariant coupling coefficient

$$\kappa_{\ell\to\ell+1} \;\coloneqq\; \max_{\pi\in S_E} \mathbb{P}_{t\sim D}\big[\pi\big(C_\ell(t)\big) = C_{\ell+1}(t)\big].$$

A value of $\kappa_{\ell\to\ell+1}$ close to 1 indicates that, after relabeling experts, most tokens follow consistent expert identities across the two layers.

C.2.1 Backward transfer via cross-layer coupling

We now formalize the above intuition with a simple *backward-transfer* guarantee: if adjacent layers exhibit strong cross-layer coupling, then aligning the earlier-layer routing to the later-layer routing bounds how far it can drift from the same latent token allocation. For clarity, we restrict attention to the top-1 routing setting ($k = 1$), where each token activates exactly one expert per layer.

Specifically, let $\mathcal{T}$ denote the token space and let $t \sim D$ be drawn from the data distribution. Assume the existence of an underlying latent assignment

$$A : \mathcal{T} \to \{1, \ldots, E\},$$

which represents an idealized data-induced expert partition.

For layer ℓ, let $C_\ell(t) \in \{1, \ldots, E\}$ denote the expert selected by the router for token t. Define the permutation-invariant cross-layer coupling coefficient

$$\kappa_{\ell\to\ell+1} \;\coloneqq\; \max_{\pi\in S_E} \mathbb{P}_{t\sim D}\big[\pi\big(C_\ell(t)\big) = C_{\ell+1}(t)\big], \tag{45}$$

where S_E is the symmetric group on $\{1, \ldots, E\}$.

Define the routing error of layer $\ell + 1$ relative to A as

$$\varepsilon_{\ell+1} \;\coloneqq\; \mathbb{P}_{t\sim D}[C_{\ell+1}(t) \neq A(t)].$$

Theorem C.5 (Backward transfer via cross-layer coupling)**.** *There exists a permutation $\pi^\star \in S_E$ such that the aligned routing decision $\pi^\star(C_\ell(t))$ satisfies*

$$\mathbb{P}_{t\sim D}\big[\pi^\star\big(C_\ell(t)\big) \neq A(t)\big] \;\leq\; \varepsilon_{\ell+1} + \big(1 - \kappa_{\ell\to\ell+1}\big). \tag{46}$$

Proof. By definition of $\kappa_{\ell\to\ell+1}$, there exists

$$\pi^\star \in \arg\max_{\pi\in S_E} \mathbb{P}_{t\sim D}\big[\pi\big(C_\ell(t)\big) = C_{\ell+1}(t)\big]$$

such that

$$\mathbb{P}_{t\sim D}\big[\pi^\star\big(C_\ell(t)\big) = C_{\ell+1}(t)\big] = \kappa_{\ell\to\ell+1}.$$

Consider the event

$$\mathcal{E} \;\coloneqq\; \big\{\pi^\star\big(C_\ell(t)\big) \neq A(t)\big\}.$$

Then

$$\mathcal{E} \subseteq \{C_{\ell+1}(t) \neq A(t)\} \;\cup\; \big\{\pi^\star\big(C_\ell(t)\big) \neq C_{\ell+1}(t)\big\}.$$

Taking probabilities and applying the union bound yields

$$\begin{aligned}\mathbb{P}(\mathcal{E}) &\le \mathbb{P}[C_{\ell+1}(t) \neq A(t)] + \mathbb{P}\big[\pi^\star\big(C_\ell(t)\big) \neq C_{\ell+1}(t)\big] \\ &= \varepsilon_{\ell+1} + \Big(1 - \mathbb{P}\big[\pi^\star\big(C_\ell(t)\big) = C_{\ell+1}(t)\big]\Big) \\ &= \varepsilon_{\ell+1} + \big(1 - \kappa_{\ell\to\ell+1}\big),\end{aligned}$$

which proves the claim. □

Theorem C.5 explains why cross-layer agreement is a meaningful training signal. If layer $\ell{+}1$ routing is closer to an underlying token allocation (small $\kappa_{\ell+1}$), then forcing g_ℓ to align with $g_{\ell+1}$ bounds how far g_ℓ can drift from that same allocation. This gives a principled view of $\mathcal{R}_{\mathrm{cp}}$: later-layer routing can act as a "teacher signal" that transfers stable pathways backward, reducing routing ambiguity early in training.

C.3 Compatibility with load balancing

In this section, we present the complete statements and proofs of Proposition C.6 and Proposition C.7 which regards the compatibility between the load balancing condition, the intra-layer specialization loss, and the cross-layer coupling loss.

Before presenting the proof, we note that *exact* load balancing is not achievable when the batch size is not divisible by the number of experts. However, since the imbalance per expert is at most one token, and the batch size in practice is large, this discrepancy is negligible. Thus, in this subsection, we assume the batch size is divisible by the number of experts without loss of generality.

C.3.1 Balanced partition exists under orthogonality-style structure

The following proposition demonstrates that load balancing can be maintained under conditions of expert orthogonality, illustrating the compatibility between the intra-layer specialization loss and load balancing:

Proposition C.6. *Suppose $k = 1$. For $e = 1, 2, \cdots, E$, denote $\mathcal{P}^{(l,e)}$ as the input space in which all the token can activate the e-th expert in layer l. Then there is always possible that the token space $\mathcal{P}^{(l,1)}, \mathcal{P}^{(l,2)}, \cdots, \mathcal{P}^{(l,E)}$ are convex, connected, and disjoint. Moreover, each $\mathcal{P}^{(l,e)}$ contains B/E elements for the batch of input tokens.*

Proof. Since $E \mid B$, let $m = \dfrac{B}{E}$. We aim to partition the input set $\{x_1^{(l,e)}, x_2^{(l,e)}, \cdots, x_B^{(l,e)}\} \subset \mathbb{R}^h$ into E convex connected subsets of equal size m. Pick any nonzero vector $a \in \mathbb{R}^h$. For each token $x_i^{(l,e)}$, compute the scalar projection

$$u_i = a^\top x_i^{(l,e)}, \quad i = 1, \ldots, N. \tag{47}$$

Without loss of generality, sort them in increasing order:

$$u_{(1)} \le u_{(2)} \le \cdots \le u_{(N)}. \tag{48}$$

As $N_E \mid N$, we can divide this ordered list into E consecutive blocks of size m. Specifically, denote

$$B_e := \{u_{((e-1)m+1)},\, u_{((e-1)m+2)},\, \ldots,\, u_{(em)}\}, \quad e = 1, \ldots, E. \tag{49}$$

Now define $E-1$ hyperplanes of the form

$$H_e = \{\, x \in \mathbb{R}^n : a^\top x + b_r = 0\}, \qquad e = 1, \ldots, E-1, \tag{50}$$

where each b_e is chosen to satisfy that $-b_e \in \big(u_{(em)},\, u_{(em+1)}\big)$, which means that the hyperplane lies strictly between the last element of block B_r and the first element of block B_{e+1}.

These hyperplanes split $\mathbb{R}^h$ into E slabs:

$$P_r = \{\, x : a^\top x \in [\alpha_{e-1}, \alpha_e]\}, \quad e = 1, \ldots, E, \tag{51}$$

where $\alpha_0 < \alpha_1 < \cdots < \alpha_E$ are thresholds satisfying

$$u_{(em)} < \alpha_e < u_{(em+1)}, \quad e = 1, \ldots, E-1, \tag{52}$$

and we set $\alpha_0 = -\infty$, $\alpha_E = +\infty$ for completeness.

Each region P_e is convex (intersection of halfspaces), connected, and by construction contains exactly $m = B/E$ tokens. Thus, if we let $\mathcal{P}^{(l,e)} = P_e$, we obtain a partition of the token space into E disjoint convex connected subset with equal token counts, which proves the theorem. □

Proposition C.6 is an existence result addressing a common concern: load balancing constrains *how much* each expert is used, while our regularizers constrain *what* experts learn. The proposition shows that even under strong orthogonality-style specialization, one can construct token partitions (of equal measure) and corresponding router directions that satisfy perfect load balance. Hence, balancing and specialization objectives need not be inherently in conflict.

C.3.2 Coupling optimum exists under perfect load balance

Then we consider the compatibility between the coupling loss. Formally, for a given input $x_i^{(l)}$ and expert $e = 1, 2, \cdots, E$, define a binary variable:

$$f_i^{(l,e)} := \chi(\text{The expert } e \text{ in layer } l \text{ is activated})$$

where χ denotes the indicator function. With the definition of $f_i^{(l,e)}$, we can present the following proposition:

Proposition C.7. *If we define the coupling loss $\mathcal{R}_{cp}(x_i)$ as Eq. (7), there exists a state that $\mathcal{L}_{cp}$ reach the optimal when satisifying the load balance condition*

$$\sum_{i=1}^{B} f_i^{(l,e)} = \sum_{i=1}^{B} f_i^{(l,\nu)}$$

for any $e, \nu \in \{1, 2, \cdots, E\}$.

Proof. Denote the coupling loss for one given token batch as $\mathcal{L}_{\text{cp}} := \sum_{i=1}^{B} \mathcal{R}_{\text{cp}}(x_i)$, then we have:

$$\mathcal{L}_{\text{cp}} = -\sum_{i=1}^{B}\sum_{l=1}^{L-1}\sum_{e=1}^{E}\sum_{\nu \in \mathbb{T}_i^{(l,e)}} s_i^{(l,e)} s_i^{(l+1,\nu)} \geq -\sum_{i=1}^{B}\sum_{l=1}^{L-1}\sum_{e=1}^{E} s_i^{(l,e)} \quad = -\left(B(L-1)\right), \tag{53}$$

where the equality condition is that for any $\nu \notin \mathbb{T}_i^{(l,e)}$ it holds

$$P_i^{(l,(e,\nu))} = 0, \tag{54}$$

for $i = 1, 2, \cdots, B$ and $e = 1, 2, \cdots, E$.

Recall the load balance condition

$$\sum_{i=1}^{B} f_i^{(l,e)} = \sum_{i=1}^{B} f_i^{(l,\nu)}, \tag{55}$$

where $e, \nu \in \{1, 2, \cdots, E\}$. We now prove that (54) and (55) can be simultaneously satisfied by explicitly constructing the desired condition.

We denote

$$[n] := \{1, 2, \ldots, n\}, \qquad [n]^k := \underbrace{[n] \times \cdots \times [n]}_{k \text{ times}}.$$

And we also define modular addition on $\{1, \ldots, n\}$ by

$$a \oplus_n b := \big((a - 1 + b) \bmod n\big) + 1.$$

Consider $\iota = (\iota_1, \ldots, \iota_k)$, any array in $[E]^k$. We define the following class of functions:

$$\mathcal{F}_{B,E,k} := \Big\{ f : [B] \to [E]^k \;\Big|\; \forall i = 1, 2, \cdots, B,\ f(i) := \big(\iota_1 \oplus_{N_E} (i-1), \ldots, \iota_k \oplus_{N_E} (i-1)\big) \Big\}.$$

Equivalently in component form, it holds that

$$f(i) = \big(f(i)_1, \ldots, f(i)_k\big), \quad (f(i))_r = \big((s_r - 1) + (i - 1)\big) \bmod N_E + 1, \quad r = 1, \ldots, k, \tag{56}$$

where $(f(i))_r$ denotes the $\mathcal{R}$-th element of $f(i)$. Then implies the recursion that

$$\forall i \in \{1, \ldots, B\},\ \forall r \in \{1, \ldots, k\}, \quad f(i+1)_r = f(i)_r \oplus_{N_E} 1, \qquad f(N_E + 1) = f(1). \tag{57}$$

Now taking any collection of parameters $\eta_i^{(l,\kappa)}$ for $i = 1, 2, \cdots, B$, $l = 1, 2, \cdots, L$, and $\kappa = 1, 2, \cdots, k$ subject to the normalization constraint

$$\sum_{\kappa=1}^{k} \eta_i^{(l,\kappa)} = 1, \quad \forall l \in \{1, 2, \cdots, L\} \text{ and } i \in \{1, 2, \cdots, B\}. \tag{58}$$

We also take $f_1, f_2, \ldots, f_L \in \mathcal{F}_{B,E,k}$. Then define the routing scores $\eta_i^{(l,e)}$ by

$$s_i^{(l,e)} = \begin{cases} \eta_i^{(l,\kappa)}, & \text{if } e = (f_l(i))_\kappa \text{ for some } \kappa \in \{1, \ldots, k\}, \\ 0, & \text{otherwise.} \end{cases} \tag{59}$$

We now verify that the term $s_i^{(l,e)}$ defined in (59) satisfies conditions (54) and (55). Specifically, for Eq. (54) we have

$$P_i^{(l,(e,\nu))} := s_i^{(l,e)} s_i^{(l+1,\nu)} = \begin{cases} \eta_i^{(l,\kappa_1)} \eta_i^{(l+1,\kappa_2)}, & \text{if } e = (f_l(i))_{\kappa_1},\ \nu = (f_{l+1}(i))_{\kappa_2}, \\ 0, & \text{otherwise.} \end{cases} \tag{60}$$

Thus $\mathbb{T}_i^{(l,e)}$ equals to the set of all the elements of $f_{l+1}(i)$. And for any $\nu \notin \mathbb{T}_i^{(l,e)}$, it holds that $P_i^{(l,(e,\nu))} = 0$.

Moreover, we consider Eq. (55). Recall the recursion property (57) of the selected function. Since $E|B$, in each layer every expert is loaded exactly $\dfrac{Bk}{R}$, which directly gives (55). □

Proposition C.7 strengthens the compatibility message for $\mathcal{R}_{\text{cp}}$: it shows that there exist routing configurations that minimize the coupling objective while maintaining perfect load balance. This supports treating $\mathcal{R}_{\text{cp}}$ as a plug-and-play auxiliary term on top of standard balancing losses, rather than a competing constraint. Empirically, we further confirm that adding $\mathcal{R}_{\text{sp}}$ and $\mathcal{R}_{\text{cp}}$ does not hinder load-balance optimization: the load-balance loss curves converge rapidly and remain nearly unchanged across ablations (Appendix D.1.5).

Takeaway: a positive feedback loop. Combining Theorem 5.1 (weak advantage sharpens routing) and Theorem 5.2 (decisive routing amplifies specialization), we obtain a positive feedback loop: even a weak loss advantage of the best expert tends to sharpen routing (lower entropy), which increases the purity of each expert's effective training data; this purer distribution then strengthens regional experts and further amplifies specialization. Our two losses directly target the bottlenecks of this loop: $\mathcal{R}_{\text{sp}}$ encourages functional differentiation within a layer, while $\mathcal{R}_{\text{cp}}$ stabilizes pathways across depth so that these specialization signals persist and propagate. A schematic summary is provided in Figure 5, while Section 7 gives a small set of experiments that directly verify the loop's empirical predictions.

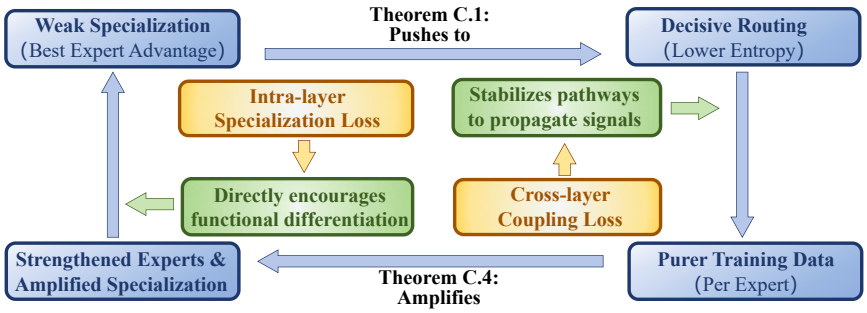


Figure 5: **The self-reinforcing cycle** illustrating how expert specialization and routing decisiveness amplify one another.

Table 1: Mixture-of-Experts (MoE) model configurations and training data volumes. 'A. Experts' denotes the activated experts and 'A. Params' denotes the activate parameters.

Model size	Experts	A. Experts	Params	A. Params	Training Tokens
Small	16	2	0.4B	80M	30B
Medium	64	4	1.1B	100M	30B
Large	96	6	7.0B	500M	50B

C.4 Summary: Closing the Theoretical Loop.

Appendix C can be read as answering three questions that together close the theory loop:

- **Why does specialization sharpen routing?** Appendix C.1 shows that even a weak best-expert advantage is sufficient to push routing toward a sharper, lower-entropy regime (Theorem C.1), making expert assignments more decisive.
- **Why does decisive routing amplify specialization?** Appendix C.1 further shows that once routing is sharp and stable, each expert is trained on higher-purity data, which strengthens experts on their advantage regions and amplifies specialization (Theorem C.4).
- **Why is cross-layer coupling a useful learning signal, and is it compatible with load balancing?** Appendix C.2 formalizes cross-layer coupling as a meaningful self-supervised signal: under strong coupling, later-layer routing can transfer pathway structure backward and stabilize token–expert identities across depth (Theorem C.5). Appendix C.3 establishes that these objectives remain compatible with standard load balancing (Propositions C.6 and C.7).

Together with the main-text propositions (proved in Appendix B), these results provide a coherent closed-loop theory for why $\mathcal{R}_{\text{sp}}$ and $\mathcal{R}_{\text{cp}}$ jointly improve both specialization quality and routing stability.

D Large-scale Experiments

In this section, we present the experimental setup and results in the pre-training tasks of MoE models with scaling model size.

D.1 Pre-train MoE models

D.1.1 Experimental setup

Infrastructure. We integrate two auxiliary loss functions into the Megatron-LM framework (Shoeybi et al., 2019) as a plug-and-play module. By setting the corresponding hyperparameters, these losses can be enabled during MoE training.

Model architecture. We evaluate two MoE variants at multiple scales. For the vanilla MoE, we adopt a mainstream design comprising RMS normalization (Zhang & Sennrich, 2019), SwiGLU

Table 2: Validation perplexity (↓) across model scales and auxiliary-loss configurations.

	Vanilla MoE			DeepSeek-style MoE		
Losses	**Small**	**Medium**	**Large**	**Small**	**Medium**	**Large**
$\mathcal{L}_{\text{lb}}$	14.01	12.50	9.68	13.54	12.33	9.56
$\mathcal{L}_{\text{lb,o,v}}$	14.27	12.71	9.84	13.76	12.51	9.81
$\mathcal{L}_{\text{lb,sp,cp}}$	13.75	12.27	9.48	13.37	12.16	9.47
$\mathcal{L}_{\text{lb,z}}$	13.80	12.33	9.52	13.40	12.07	9.46
$\mathcal{L}_{\text{lb,z,o,v}}$	14.15	12.67	9.82	13.69	12.36	9.77
$\mathcal{L}_{\text{lb,z,sp,cp}}$	13.63	12.17	9.42	13.30	11.99	9.39

activations (Shazeer, 2020), and rotary position embeddings (RoPE) (Su et al., 2024); architectural hyperparameters are listed in Table 1. For the DeepSeek-style MoE, we augment the vanilla design with **ONE** shared expert and employ the **auxiliary-loss-free** load balancing strategy (Dai et al., 2024). The hyperparameters λ_{cp} and λ_{sp} are set to 1×10^{-3} and 2×10^{-3}, respectively. Unless otherwise noted, the load-balance loss weight is set to 1×10^{-2}, the z-loss weight $\mathcal{R}_{\text{z}}$ (Zoph et al., 2022) to 1×10^{-3}, and the update step size for the coefficient b in the auxiliary-loss-free load balancing strategy to 1×10^{-3}.

Training settings. Training is performed on the C4-en dataset (Raffel et al., 2020) using the LLaMA-2 tokenizer (Touvron et al., 2023). The small and medium MoE models are trained for 30 billion tokens, and the large MoE model for 50 billion tokens. This token budget exceeds the data size suggested by MoE scaling laws (Clark et al., 2022), providing sufficient signal for convergence. We use AdamW (Loshchilov & Hutter, 2017) optimizer with moment coefficient $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay coefficient 0.1.

D.1.2 Comparison of validation perplexity.

We evaluate on the C4 dataset (Raffel et al., 2020) across three model scales for both Vanilla MoE and DeepSeek-style MoE. Table 2 summarizes validation perplexity under different auxiliary-loss configurations. Beyond standard load-balancing (and optionally the router z-loss), we include a recent auxiliary-loss baseline from Guo et al. (2025a) that regularizes orthogonality and routing-logit variance (denoted as $\mathcal{L}_{\text{lb,o,v}}$, and $\mathcal{L}_{\text{lb,z,o,v}}$ when combined with z-loss); further discussions for auxiliary-loss comparisons are deferred to Appendix E. Overall, our specialization-and-coupling regularizers consistently improve perplexity across scales and architectures, while the variance-on-logits baseline tends to degrade perplexity relative to the corresponding objectives. These trends hold both with and without z-loss, indicating that our activation/path-based regularization is complementary to standard router-stabilization and more robust than directly amplifying routing-logit dispersion. Notably, the gains are architecture-agnostic: the same objectives benefit both Vanilla MoE and DeepSeek-style MoE (with shared experts), and in some regimes the improved Vanilla MoE reaches or surpasses the DeepSeek-style baseline without increasing activated capacity, highlighting that targeted training objectives can rival architectural router modifications while remaining plug-and-play.

D.1.3 Ablation study for different regularizations

To evaluate the influence of various regularization techniques on model performance, we performed an ablation study utilizing a medium-scale architecture. The outcomes of this investigation are summarized in Table 3. Our analysis identifies several consistent trends. First, each auxiliary objective demonstrates individual efficacy: for the Vanilla MoE model, the introduction of $\mathcal{R}_{\text{sp}}$ leads to a reduction in perplexity, whereas $\mathcal{R}_{\text{cp}}$ produces a more substantial improvement. Similarly, in the DeepSeek-style MoE, both regularizers enhance performance, with $\mathcal{R}_{\text{cp}}$ yielding a greater effect. Moreover, the two losses exhibit complementarity, as their combined application results in further gains. When integrated with additional components such as $\mathcal{R}_{\text{lb}}$, the full regularization set achieves the most pronounced enhancements across both model variants. These patterns indicate that the specialization and coupling mechanisms independently contribute to refining expert behavior and, when employed together, synergize to produce cumulative reductions in perplexity.

Table 3: Ablations for two MoE architectures; metric is perplexity (↓).

Model	$\mathcal{L}_{\text{lb}}$	$\mathcal{L}_{\text{lb,sp}}$	$\mathcal{L}_{\text{lb,cp}}$	$\mathcal{L}_{\text{lb,sp,cp}}$	$\mathcal{L}_{\text{lb,z}}$	$\mathcal{L}_{\text{lb,z,sp}}$	$\mathcal{L}_{\text{lb,z,cp}}$	$\mathcal{L}_{\text{lb,z,sp,cp}}$
Vanilla MoE	12.50	12.44	12.33	12.27	12.33	12.27	12.21	12.17
DeepSeek-style MoE	12.33	12.29	12.22	12.16	12.07	12.05	12.00	11.99

Table 4: Zero-shot accuracy of *Vanilla MoE* and *DeepSeek-style MoE* across seven benchmarks (↑).

Model	Loss	BoolQ	ARC-E	ARC-C	Truthful QA-MC2	PIQA	MMLU	Hella Swag	Avg.
Vanilla MoE	$\mathcal{L}_{\text{lb}}$	0.570 (0.003)	0.452 (0.003)	0.204 (0.003)	0.432 (0.001)	0.622 (0.005)	0.247 (0.002)	0.268 (0.002)	0.399
	$\mathcal{L}_{\text{lb,sp,cp}}$	0.578 (0.003)	0.462 (0.002)	0.210 (0.004)	0.451 (0.003)	0.627 (0.002)	0.253 (0.002)	0.275 (0.004)	**0.408**
	$\mathcal{L}_{\text{lb,z}}$	0.567 (0.003	0.457 (0.004)	0.205 (0.002)	0.433 (0.003)	0.629 (0.002)	0.250 (0.001)	0.267 (0.004)	0.401
	$\mathcal{L}_{\text{lb,z,sp,cp}}$	0.589 (0.003)	0.453 (0.004)	0.206 (0.006)	0.445 (0.003)	0.637 (0.003)	0.257 (0.002)	0.274 (0.003)	**0.409**
DS-style MoE	$\mathcal{L}_{\text{lb}}$	0.578 (0.002)	0.453 (0.001)	0.205 (0.003)	0.438 (0.003)	0.631 (0.002)	0.248 (0.001)	0.269 (0.002)	0.403
	$\mathcal{L}_{\text{lb,sp,cp}}$	0.584 (0.001)	0.452 (0.003)	0.206 (0.005)	0.457 (0.002)	0.635 (0.002)	0.255 (0.003)	0.277 (0.005)	**0.410**
	$\mathcal{L}_{\text{lb,z}}$	0.564 (0.002)	0.453 (0.002)	0.205 (0.002)	0.444 (0.002)	0.628 (0.001)	0.252 (0.001)	0.270 (0.004)	0.402
	$\mathcal{L}_{\text{lb,z,sp,cp}}$	0.575 (0.002)	0.461 (0.004)	0.214 (0.004)	0.452 (0.004)	0.642 (0.003)	0.257 (0.002)	0.280 (0.002)	**0.412**

D.1.4 Downstream Task Evaluations for pre-trained MoE models.

We evaluate the pre-trained MoE models on supervised fine-tuning tasks (see Appendix for details; (Raffel et al., 2020)) and seven zero-shot benchmarks: BoolQ (Clark et al., 2019), ARC-Easy and ARC-Challenge (Clark et al., 2018), TruthfulQA-MC2 (Lin et al., 2022), PIQA (Bisk et al., 2020), MMLU (Hendrycks et al., 2020), and HellaSwag (Zellers et al., 2019) as outlined in Table 4. For each experimental setup, the process was conducted three times with different random seeds to ensure robustness.

Across both architectures, the addition of $\mathcal{R}_{\text{cp}}$ and $\mathcal{R}_{\text{sp}}$ enhances zero-shot accuracy in synergy with load-balance loss and z-loss. For the Vanilla MoE, integrating $\mathcal{R}_{\text{sp}}$ and $\mathcal{R}_{\text{cp}}$ with load-balance regularization leads to a marked improvement in average accuracy. Further gains are observed when $\mathcal{R}_{\text{sp}}$ and $\mathcal{R}_{\text{cp}}$ are applied in combination with load-balance loss and z-loss. In the DeepSeek-style MoE, a similar trend emerges: the inclusion of $\mathcal{R}_{\text{sp}}$ and $\mathcal{R}_{\text{cp}}$ alongside $\mathcal{R}_{\text{lb}}$ boosts average performance, while the loss with full set of regularization ($\mathcal{L}_{\text{lb,z,sp,cp}}$) achieves the highest overall accuracy, outperforming both $\mathcal{L}_{\text{lb,z}}$ and $\mathcal{L}_{\text{lb}}$, and yielding superior results on benchmarks such as ARC-E, ARC-C, and PIQA.

Although individual benchmarks show slight variations, the consistent upward trend in average accuracy demonstrates that $\mathcal{R}_{\text{cp}}$ and $\mathcal{R}_{\text{sp}}$ effectively complement load-balance loss and z-loss, contributing to downstream improvements across model families.

D.1.5 The Impact of Auxiliary Loss on Load Balance Loss

To investigate the impact of the proposed regularization terms $\mathcal{R}_{\text{sp}}$ and $\mathcal{R}_{\text{cp}}$ on the primary training objective, we analyze the load balance loss curves throughout the training process. Figure 6 compares the load balance loss of the full model $\mathcal{R}_{\text{lb,sp,cp}}$ with ablation studies involving the baseline configurations $\mathcal{R}_{\text{lb}}$, $\mathcal{R}_{\text{lb,cp}}$, and $\mathcal{R}_{\text{lb,sp}}$.

All model variants exhibit rapid and stable convergence. The load balance loss for each configuration declines sharply within the initial 5,000 steps and stabilizes promptly near its optimal value. This

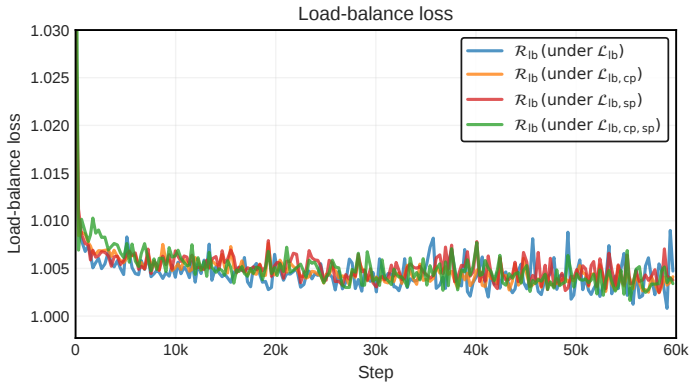


Figure 6: The impact of auxiliary loss on load balance loss

Table 5: Validation perplexity for medium model scale with three random repetitions (↓).

Losses	Vanilla MoE	DeepSeek-style MoE
$\mathcal{L}_{\text{lb}}$	12.50 (0.01)	12.33 (0.02)
$\mathcal{L}_{\text{lb,sp,cp}}$	12.26 (0.01)	12.15 (0.02)
$\mathcal{L}_{\text{lb,z}}$	12.33 (0.02)	12.07 (0.01)
$\mathcal{L}_{\text{lb,z,sp,cp}}$	12.17 (0.01)	11.98 (0.01)

demonstrates that incorporating the auxiliary objectives does not hinder the model's capacity to learn the primary load balancing task.

The load-balance loss curves are nearly identical across all configurations, with the trajectories largely overlapping throughout training. This indicates that incorporating $\mathcal{R}_{\text{sp}}$ and $\mathcal{R}_{\text{cp}}$ has a negligible effect on load balancing and does not alter the optimization of the load-balance objective.

D.1.6 Pre-training results with random seeds

To rigorously demonstrate that the reported improvements are attributable to the auxiliary regularization and are statistically significant rather than resulting from optimization noise, we conducted repeated pre-training experiments using medium-sized models for both the Vanilla MoE and DeepSeek-style architectures. The experimental configuration remains identical to that described in Appendix D.1.1.

As illustrated in the Table 5, for the Vanilla MoE, the comparison between $\mathcal{L}_{\text{lb}}$ versus $\mathcal{L}_{\text{lb,sp,cp}}$ shows an improvement from 12.50 to 12.26, corresponding to an approximately 1.9% relative reduction, with a standard deviation across seeds of only 0.01. Furthermore, when all auxiliary terms are included, $\mathcal{L}_{\text{lb,z}}$ versus $\mathcal{L}_{\text{lb,z,sp,cp}}$ improves from 12.33 to 12.17 , with standard deviations ranging from 0.01 to 0.02. These findings confirm consistent and statistically meaningful gains in validation performance.

D.1.7 Hyperparameter sensitivity in pre-training

The validation performance for the pre-training tasks, as presented in Table 2, is based on a fixed hyperparameter selection described in Appendix D.1.1. To examine the sensitivity of the hyperparameters λ_{cp} and λ_{sp}, we performed a hyperparameter sweep around the default values using a medium-sized model. Validation perplexity (where lower values indicate better performance) was measured under the following variations:

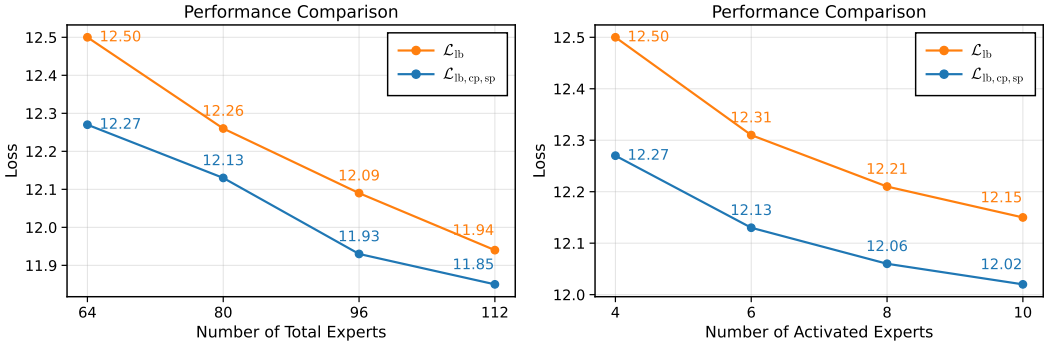
- With λ_{sp} fixed at 2×10^{-3}, we varied λ_{cp} from 0.2 to 2 times the default value of 1×10^{-3}.
- With λ_{cp} fixed at 1×10^{-3}, we varied λ_{sp} from 0.25 to 3 times the default value of 2×10^{-3}.

Table 6: Perplexity with fixed λ_{sp} and varied λ_{cp}.

λ_{cp}	2×10^{-4}	5×10^{-4}	1×10^{-3}	2×10^{-3}
PPL	12.41	12.35	**12.27**	12.30

Table 7: Perplexity with fixed λ_{cp} and varied λ_{sp}.

λ_{sp}	5×10^{-4}	1×10^{-3}	2×10^{-3}	3×10^{-3}
PPL	12.32	12.30	**12.27**	12.29



(a) varying total experts

(b) varying activated experts

Figure 7: Scalability performance on the model with medium size.

The results, shown in Tables 6 and 7, demonstrate that the model performance remains stable across a broad interval. The perplexity changes are limited to less than 1% relative to the optimum. The heuristic choice of $\lambda_{\text{cp}} = 10^{-3}$ and $\lambda_{\text{sp}} = 2 \times 10^{-3}$ yields near-optimal results, and deviations cause only minor degradation.

D.1.8 Scalability of the auxiliary loss

We evaluate scalability on *medium-sized* MoE models by varying (i) the number of activated experts (N) and (ii) the total number of experts (E). As shown in Figure 7, our auxiliary objectives consistently achieve lower perplexity than the load-balance-only baseline across both axes. Notably, with our loss, activating only N=6 experts already outperforms the baseline even with N=10 (12.13 vs. 12.15), and using a smaller expert pool E=96 surpasses the baseline with E=112 (11.93 vs. 11.94), indicating improved scaling efficiency with fewer active/total experts.

D.2 Quantitative comparison with DeepSeekMoE-style load balancing

To directly address whether training with our proposed specialization induces more expert specialization than DeepSeekMoE's auxiliary-loss-free load balancing, we compare two training objectives including $\mathcal{L}_{\text{lb}}$ and $\mathcal{L}_{\text{lb,cp,sp}}$ over the small model with configurations in Table 1. As a proxy for expert specialization and routing coherence, we measure every 1000 iterations the percentage of tokens whose top-1 expert assignment remains unchanged between consecutive checkpoints. Higher values correspond to more stable token–expert assignments, lower routing entropy, and, via Proposition C.7, more persistent expert-specific gradient directions.

The results, as detailed in Table 8, demonstrate that across all training stages, $\mathcal{L}_{\text{lb,cp,sp}}$ consistently enhances routing stability by 1–2 absolute points (approximately 2%–4% relative improvement) compared to $\mathcal{L}_{\text{lb}}$. The gains are especially significant during early training phases when routing ambiguity is most severe, which aligns precisely with the regime addressed by Proposition 2. Furthermore, the benefits persist even at later stages (e.g., 59K–60K iterations), indicating that expert assignments maintain greater consistency over time.

Table 8: The fraction of tokens that keep the same experts between checkpoints.

Iteration range	1K–2K	2K–3K	4K–5K	9K–10K	19K–20K	29K–30K	59K–60K
$\mathcal{L}_{\text{lb,sp}}$	0.4746	0.6056	0.6601	0.6987	0.7450	0.7864	0.9011
$\mathcal{L}_{\text{lb,sp,cp}}$	0.4898	0.6213	0.6757	0.7187	0.7594	0.7935	0.9067

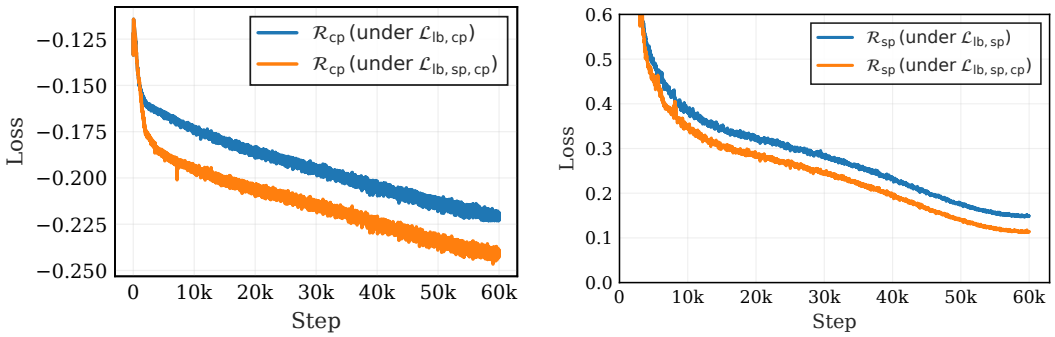


Figure 8: Training dynamics on the 0.4B MoE model. Left: cross-layer coupling loss $\mathcal{R}_{\text{cp}}$ when training with $\mathcal{L}_{\text{lb,cp}}$ vs. $\mathcal{L}_{\text{lb,cp,sp}}$; adding $\mathcal{R}_{\text{sp}}$ consistently makes $\mathcal{R}_{\text{cp}}$ more negative (stronger coupling). Right: intra-layer specialization loss $\mathcal{R}_{\text{sp}}$ when training with $\mathcal{L}_{\text{lb,sp}}$ vs. $\mathcal{L}_{\text{lb,sp,cp}}$; adding $\mathcal{R}_{\text{cp}}$ consistently reduces $\mathcal{R}_{\text{sp}}$ (stronger specialization).

Table 9: Hyperparameters for the fine-tuning task under Qwen3-30B models.

Hyperparameters	**Value**
Global batch size	64
Learning rate	8e-5
Epochs	3
Sequence length	32768
$\lambda_{\text{lb}}^{\diamond}$	1e-3
λ_{sp}	2e-4
λ_{cp}	1e-4

$^{\diamond}$ The coefficient of the regularization of load-balancing.

In conjunction with Fig 8, where $\mathcal{L}_{\text{lb,sp,cp}}$ reduces the intra-layer specialization loss R_{sp} relative to $\mathcal{L}_{\text{lb,sp}}$, these findings confirm that our method further sharpens expert differentiation beyond the capabilities of auxiliary-loss-free load balancing alone. This improvement is consistent with our theoretical framework: reduced activation similarity promotes more orthogonal gradients (as per Proposition C.7), while enhanced routing stability supports stronger and more coherent expert paths (in line with Proposition 4.1).

D.3 Finetuning tasks evaluation

We fine-tune `Qwen3-30B-A3B-Instruct-2507` model on an internal corpus of 38B tokens under identical training hyperparameters listed in Table 9. We evaluate on a broad suite of reasoning and knowledge-intensive benchmarks, including HumanEval (Chen et al., 2021), GSM8K (Cobbe et al., 2021), math500_PRM800K_dataset (Lightman et al., 2023), and MMLU (Hendrycks et al., 2020). Across nearly all settings, incorporating $\mathcal{R}_{\text{cp}}$ and $\mathcal{R}_{\text{sp}}$ outperforms the baseline, yielding consistent gains on reasoning-oriented tasks as well as aggregate knowledge measures as Table 10. While a minor fluctuation is observed on the humanities subset of MMLU, the overall trend remains positive, confirming that our objectives not only sharpen specialization in pre-training but also transfer effectively to finetuning adaptation.

Table 10: Evaluation score on Qwen3-30B-A3B-Instruct-2507 finetuning tasks. The last four rows stands for the performance for mmlu dataset with different domains.

Dataset	Metric	$\mathcal{L}_{\text{lb}}$	$\mathcal{L}_{\text{lb,sp,cp}}$
openai_humaneval	humaneval_pass@1	92.07	**95.73**
gsm8k	accuracy	93.33	**94.16**
math_prm800k_500	accuracy	94.00	**94.20**
mmlu	naive_average	78.97	**79.86**
mmlu-weighted	weighted_average	76.35	**77.10**
mmlu-humanities	naive_average	**75.90**	75.42
mmlu-stem	naive_average	87.38	**88.97**
mmlu-social-science	naive_average	75.91	**77.11**
mmlu-other	naive_average	72.59	**73.52**

Table 11: Throughput comparison (samples/s; ↑) on four standard benchmarks. Here 'SO' means the system optimization.

Model size	Loss	MMLU	GSM8K	HumanEval	Math500
Small	$\mathcal{L}_{\text{lb}}$ *w.o.* SO	161.3 (1.00×)	26.2 (1.00×)	35.7 (1.00×)	6.9 (1.00×)
	$\mathcal{L}_{\text{lb}}$ *w.* SO	164.9 (1.03×)	26.5 (1.01×)	36.6 (1.02×)	7.0 (1.01×)
	$\mathcal{L}_{\text{lb,sp,cp}}$ *w.* SO	170.4 (1.06×)	27.0 (1.03×)	37.5 (1.05×)	7.1 (1.03×)
Medium	$\mathcal{L}_{\text{lb}}$ *w.o.* SO	157.4 (1.00×)	25.9 (1.00×)	27.7 (1.00×)	6.1 (1.00×)
	$\mathcal{L}_{\text{lb}}$ *w.* SO	162.7 (1.03×)	26.2 (1.01×)	28.6 (1.03×)	6.2 (1.01×)
	$\mathcal{L}_{\text{lb,sp,cp}}$ *w.* SO	165.7 (1.05×)	26.6 (1.03×)	29.4 (1.06×)	6.3 (1.03×)
Large	$\mathcal{L}_{\text{lb}}$ *w.o.* SO	96.9 (1.00×)	15.0 (1.00×)	12.8 (1.00×)	3.9 (1.00×)
	$\mathcal{L}_{\text{lb}}$ *w.* SO	96.9 (1.00×)	15.0 (1.00×)	12.8 (1.00×)	3.9 (1.00×)
	$\mathcal{L}_{\text{lb,sp,cp}}$ *w.* SO	103.5 (1.07×)	15.7 (1.05×)	13.7 (1.07×)	4.2 (1.08×)

D.4 Inference acceleration

To leverage the benefits of the specialization loss and coupling loss during the inference period, we implement a path-aware placement and bucketing strategy. This involves estimating a cross-layer expert co-activation matrix from a held-out dataset, greedily co-locating strongly coupled experts on the same GPU shard via graph partitioning, and performing a lightweight pre-routing pass through the first MoE router to bucket sequences according to early expert decisions. These buckets are then assigned to shards hosting the corresponding experts, ensuring that most subsequent dispatches remain local.

We evaluate our approach on MoE models of varying scales under 8 Nvidia A100 80G GPUs with expert parallelism. The number of parallel devices are set to 8 and the microbatch size is set to 1. A baseline model trained solely with $\mathcal{R}_{\text{lb}}$ is compared against our variant trained with $\mathcal{R}_{\text{lb,sp,cp}}$. While the baseline uses default round-robin expert placement and uniform batching, our model employs the path-aware scheme described above. We also apply identical system-level optimizations to both the load-balancing baseline and our model. This design cleanly separates the acceleration attributable to engineering infrastructure from that enabled by structural properties—specifically, stronger cross-layer expert coupling and lower routing entropy—induced by our proposed losses.

As summarized in Table 11, throughput improves consistently across model sizes and benchmarks—without any architectural modifications or additional parameters. These results demonstrate that reducing routing ambiguity through $\mathcal{R}_{\text{sp}}$ and $\mathcal{R}_{\text{cp}}$ directly enhances system-level efficiency by streamlining token-to-expert execution paths. With the inference throughput, it can be observed that our proposed auxiliary losses improve model perplexity while simultaneously enhancing inference efficiency through reduced routing entropy. By promoting sharper expert specialization and stronger cross-layer coupling, tokens follow more consistent expert paths, which in an expert parallelism setup improves cache locality and reduces All-to-All communication overhead.

Table 12: The iteration time and peak memory with different loss

Model size	Loss	Iteration time (ms/iteration)	Peak memory (GB)
Small	$\mathcal{L}_{\text{lb}}$	405.9 (1.0000×)	43.5 (1.0000×)
	$\mathcal{L}_{\text{lb,sp,cp}}$	413.6 (1.0190×)	43.6 (1.0023×)
Medium	$\mathcal{L}_{\text{lb}}$	518.4 (1.0000×)	60.6 (1.0000×)
	$\mathcal{L}_{\text{lb,sp,cp}}$	526.5 (1.0156×)	60.7 (1.0016×)
Large	$\mathcal{L}_{\text{lb}}$	2927.8 (1.0000×)	73.1 (1.0000×)
	$\mathcal{L}_{\text{lb,sp,cp}}$	2942.4 (1.0049×)	73.3 (1.00027×)

Table 13: Downstream evaluation results (mean ± std) across multiple 16B-class MoE models after LoRA fine-tuning.

Method	Model	MMLU	MMLU-Pro	HellaSwag	BBH	GPQA-Diamond	MBPP	HumanEval	GSM8K
$\mathcal{L}_{\text{lb}}$	DeepSeek-MOE	0.4143$_{\pm 0.0040}$	0.1729$_{\pm 0.0034}$	0.5852$_{\pm 0.0049}$	0.4041$_{\pm 0.0053}$	0.2323$_{\pm 0.0301}$	**0.4180**$_{\pm 0.0221}$	0.2927$_{\pm 0.0356}$	0.2661$_{\pm 0.0122}$
$\mathcal{L}_{\text{lb,z}}$		0.3717$_{\pm 0.0040}$	0.1410$_{\pm 0.0032}$	0.5621$_{\pm 0.0050}$	0.3864$_{\pm 0.0053}$	0.2273$_{\pm 0.0299}$	0.3200$_{\pm 0.0209}$	0.2500$_{\pm 0.0339}$	0.1789$_{\pm 0.0106}$
$\mathcal{L}_{\text{lb,o,v}}$		0.4293$_{\pm 0.0041}$	0.1735$_{\pm 0.0034}$	0.5882$_{\pm 0.0049}$	0.4386$_{\pm 0.0055}$	0.2626$_{\pm 0.0314}$	0.3940$_{\pm 0.0219}$	0.2805$_{\pm 0.0352}$	0.2896$_{\pm 0.0125}$
$\mathcal{L}_{\text{lb,sp,cp}}$		**0.4586**$_{\pm 0.0041}$	**0.2276**$_{\pm 0.0034}$	**0.5906**$_{\pm 0.0049}$	**0.4558**$_{\pm 0.0053}$	**0.2828**$_{\pm 0.0321}$	0.4100$_{\pm 0.0220}$	**0.3415**$_{\pm 0.0371}$	**0.3275**$_{\pm 0.0128}$
$\mathcal{L}_{\text{lb}}$	DeepSeek-V2-Lite	0.5361$_{\pm 0.0040}$	0.2605$_{\pm 0.0039}$	0.5893$_{\pm 0.0049}$	0.4346$_{\pm 0.0056}$	0.2879$_{\pm 0.0323}$	0.3680$_{\pm 0.0216}$	0.3110$_{\pm 0.0363}$	0.4617$_{\pm 0.0137}$
$\mathcal{L}_{\text{lb,z}}$		0.5268$_{\pm 0.0040}$	0.2103$_{\pm 0.0037}$	0.5777$_{\pm 0.0049}$	0.3860$_{\pm 0.0054}$	0.2778$_{\pm 0.0319}$	0.3660$_{\pm 0.0216}$	0.3049$_{\pm 0.0363}$	0.4147$_{\pm 0.0136}$
$\mathcal{L}_{\text{lb,o,v}}$		0.5474$_{\pm 0.0040}$	0.2523$_{\pm 0.0039}$	0.5902$_{\pm 0.0049}$	0.4264$_{\pm 0.0056}$	0.3131$_{\pm 0.033}$	0.4080$_{\pm 0.0216}$	0.3049$_{\pm 0.0361}$	0.4701$_{\pm 0.0137}$
$\mathcal{L}_{\text{lb,sp,cp}}$		**0.5735**$_{\pm 0.0040}$	**0.3108**$_{\pm 0.0039}$	**0.6091**$_{\pm 0.0049}$	**0.4793**$_{\pm 0.0055}$	**0.3535**$_{\pm 0.0341}$	**0.4280**$_{\pm 0.0216}$	**0.3598**$_{\pm 0.0371}$	**0.5004**$_{\pm 0.0138}$
AuxLossFree	Ling-mini-2.0	0.6998$_{\pm 0.0036}$	0.4759$_{\pm 0.0044}$	0.7449$_{\pm 0.0044}$	0.6592$_{\pm 0.0051}$	0.3636$_{\pm 0.0343}$	0.6707$_{\pm 0.0368}$	0.6402$_{\pm 0.0376}$	0.8173$_{\pm 0.0106}$
AuxLossFree+z		0.6878$_{\pm 0.0036}$	0.4480$_{\pm 0.0044}$	0.7396$_{\pm 0.0044}$	0.6520$_{\pm 0.0054}$	0.3535$_{\pm 0.0341}$	0.6500$_{\pm 0.0214}$	0.6220$_{\pm 0.038}$	0.7885$_{\pm 0.0112}$
AuxLossFree+o+v		0.7119$_{\pm 0.0036}$	0.4892$_{\pm 0.0044}$	0.7596$_{\pm 0.0044}$	0.7016$_{\pm 0.0048}$	0.3838$_{\pm 0.0346}$	0.6620$_{\pm 0.0212}$	0.6524$_{\pm 0.0373}$	0.8309$_{\pm 0.0103}$
AuxLossFree+cp+sp		**0.7667**$_{\pm 0.0036}$	**0.5002**$_{\pm 0.0044}$	**0.7627**$_{\pm 0.0042}$	**0.7269**$_{\pm 0.0048}$	**0.3889**$_{\pm 0.0347}$	**0.6820**$_{\pm 0.0208}$	**0.6890**$_{\pm 0.0363}$	**0.8559**$_{\pm 0.0101}$

E Comparison with recent auxiliary-loss methods for specialization

Several recent studies have introduced auxiliary loss functions aimed at improving expert specialization and routing efficacy in Mixture-of-Experts (MoE) models. In this section, we present a conceptual analysis and empirical evaluation comparing our approach with a representative method by (Guo et al., 2025a), which combines an orthogonality loss with a *variance* loss applied to the routing logits.

E.1 Conceptual Comparison

Our method integrates two complementary components: the *intra-layer specialization* loss $\mathcal{L}_{\text{sp}}$, which promotes orthogonality in the representations of co-activated experts, thereby aligning their parameter gradients along orthogonal directions (see Proposition 3.1), and the *cross-layer coupling* loss $\mathcal{L}_{\text{cp}}$, which enforces consistency in expert selection across adjacent layers, reducing routing ambiguity (see Proposition 4.1). These losses operate on principles of *information geometry and path consistency* and are compatible with standard router-stabilization techniques, such as the z-loss and logit clipping.

In contrast, the approach by (Guo et al., 2025a) incorporates an orthogonality term along with a *variance-maximization* objective on the routing logits, explicitly encouraging high logit dispersion to enhance discrimination. While increased dispersion can sharpen top-k selections, it lacks inherent control over logit magnitudes, potentially leading to adverse interactions with softmax temperature and z-loss penalties. Specifically, unregulated variance amplification often causes prematurely peaked routing distributions or numerical instabilities (e.g., gradient spikes), increasing sensitivity to learning rate and initialization in large-scale pre-training. Our design mitigates these issues by regularizing *activations and paths* rather than directly inflating raw logit variance.

E.2 Experimental Evaluation

We compare our losses with $\mathcal{L}_{\text{lb,o,v}}$ in Table 2 and Table 13, we show that our consistency-based formulation demonstrates stronger accuracy and stability. The combination of $\mathcal{L}_{\text{sp}}$ and $\mathcal{L}_{\text{cp}}$ outperforms variance-maximization methods in both pre-training and downstream settings. These gains align with our theoretical framework: (i) orthogonalizing expert activations yields orthogonal gradient directions, reducing parameter interference; (ii) cross-layer coupling concentrates routing probability

mass along consistent paths, diminishing ambiguity and consolidating expert specialization. Together, these effects enhance final model quality and training dynamics for large-scale applications.

F Analysis for the computational and memory efficiency

In this section, we present a series of analysis for the computation and memory overhead for the gradient evaluation with our proposed auxiliary regularization.

F.1 Theoretical analysis for computational and memory overhead

Here we present a theoretical analysis for computational and memory overhead. From a computational perspective, both losses are lightweight relative to the model's core operations (attention mechanisms and feed-forward networks)

F.1.1 Intra-Layer Specialization Loss ($\mathcal{R}_{\text{SP}}$).

Computational Complexity: This loss requires computing pairwise cosine similarities between activations of the top-k selected experts. For a hidden dimension d and k activated experts, the per-token complexity is $\mathcal{O}(k^2 \cdot d)$. In standard MoE configurations, k typically assumes small values (e.g., 2 or 4), while d represents a large dimension (e.g., 4096). Consequently, $k^2 \ll d$, rendering the cost of $\mathcal{O}(k^2 \cdot d)$ negligible compared to the standard FFN transformation cost of $\mathcal{O}(k \cdot d^2)$.

Scalability: Crucially, this computational cost depends solely on the number of activated experts k, rather than the total number of experts E. This implies that even as the total expert count E scales to hundreds or thousands (as in "Mixture of Million Experts" architectures), as long as the activated expert count k remains small, the computational overhead of $\mathcal{R}_{sp}$ remains both constant and minimal.

Memory Requirements: No additional memory allocation is necessary, as this loss reuses intermediate activations $z^{(l,e)}$ already computed during the forward pass.

F.1.2 Cross-Layer Coupling Loss ($\mathcal{R}_{\text{CP}}$).

Computational Complexity: This loss operates exclusively on scalar routing logits. Specifically, it involves basic statistical operations on token routing scores across consecutive layers. These operations avoid complex matrix computations involving high-dimensional hidden states.

Memory Requirements: This loss requires storing a lightweight tensor of dimensions $E \times E \times L$ to track expert transition statistics. Since the number of experts E is typically much smaller than the hidden dimension d, the memory consumption of this tensor is negligible.

F.2 Empirical wall-clocked time and memory analysis

Empirical results are fully consistent with our theoretical complexity analysis. We systematically measured training throughput (in ms/iteration) and peak GPU memory consumption (in GB) across the Small (0.4B), Medium (1.1B), and Large (7.0B) model configurations used in Appendix D.1.1, with all experiments conducted on a uniform hardware configuration consisting of 8 A100 GPUs.

Our benchmarking results, as shown in Table 12, demonstrate that the overhead introduced by $\mathcal{R}_{\text{sp}}$ and $\mathcal{R}_{\text{cp}}$ is negligible: the combined auxiliary losses introduce only 0.5% to 1.9% additional latency, with the relative overhead exhibiting a decreasing trend as model scale increases (reducing to approximately 0.5% for the 7B parameter model), indicating favorable scaling characteristics of our method, while the additional memory footprint is minimal ($< 0.3\%$), empirically confirming that our approach does not impose additional hardware requirements.