Function Over Form: Distributional Orthogonalization in Mixture-of-Experts with Replica Expert Mechanism

Jinfan He[1,2], **Yunzhuo Liu**[2,*], **Kai Zhang**[2], **Weidong Han**[2], **Key**[2], **Rayying**[2]
[1]School of EECS, Peking University, China
[2]Tencent Hunyuan, China
∗ Corresponding author.

Abstract

The scaling of LLMs increasingly relies on MoE architectures to decouple active computation from total parameter count. However, the efficacy of MoE is often constrained by expert collapse and representation redundancy, both leading to underutilization of model capacity. To address these challenges, this paper proposes Distributional Orthogonalization Loss (DO-loss), an auxiliary regularization that shifts the focus from static weight diversity to dynamic routing behavior. By representing each expert's token assignment history as a high-dimensional binary load signature, DO-loss penalizes signature overlap to prevent expert collapse while encouraging functional specialization. To align this algorithmic design with system efficiency, we further introduce the Replica Expert Mechanism (REM), which improves load balancing through a two-tiered strategy: adjusting replica expert placement at the global-batch level and performing real-time token dispatching at the micro-batch level. Empirical evaluations demonstrate that our method outperforms the evaluated routing algorithms on downstream tasks for both 4.8BA0.5B and 30BA3B MoE models, while maintaining comparable training efficiency.

1 Introduction

Mixture-of-Experts (MoE) architectures have emerged as a standard design for scaling large language models Jiang et al. (2024); Liu et al. (2024a); Yang et al. (2025). By routing each token to a sparse subset of expert feed-forward networks (FFNs) instead of activating the entire parameter space, these models maintain constant per-token computational costs. This efficiency enables the training and deployment of models with trillions of parameters.

The effectiveness of such sparse activation depends on a sufficient degree of expert specialization, as routing tokens to specific experts yields better performance than averaging the processing results across different experts Shazeer et al. (2017); Lepikhin et al. (2020). However, this specialization is often impeded by routing biases that emerge early in training. As popular experts receive more tokens, they update more frequently, creating a feedback loop where a few experts dominate while others remain underutilized. Without explicit mechanisms for encouraging inter-expert diversity, standard gradient descent often fails to differentiate expert functions, resulting in redundant expert representations that cover similar domains.

Current approaches tend to address these problems in isolation. Load-balancing losses Fedus et al. (2022); Wang et al. (2024a) mitigate expert collapse by penalizing uneven token distributions. However, by encouraging uniform allocation, they push experts to process tokens irrespective of representational compatibility, thereby suppressing specialization. On the other hand, orthogonality-based approaches Baidu-ERNIE-Team (2025); Lv et al. (2025); Liu et al. (2024b) attempt to promote functional diversity by enforcing orthogonality among experts in the parameter space. However, these methods do not guarantee that all experts receive input tokens. Experts can remain mutually orthogonal while pointing toward regions outside the input manifold. Directly combining these techniques often

produces conflicting gradient directions that destabilize training Li et al. (2023). To prevent conflicts with load-balancing methods, recent approaches Hendawy et al. (2024); Wang et al. (2025) introduce orthogonality constraints within the expert output spaces. However, this incurs significant computational, communication, and memory overhead, and it integrates poorly with standard distributed training frameworks Rajbhandari et al. (2022); Shoeybi et al. (2019).

In this paper, we propose the Distributional Orthogonalization loss (DO-loss). The core idea is to record each expert's token assignment history as a binary load signature and penalize small pairwise Hamming distances between signatures, which indicate routing overlap (Fig. 1). This directly encourages specialization: experts covering similar token domains develop highly overlapping signatures and thus incur severe penalties, forcing them to diverge. The same mechanism also makes expert collapse self-correcting. When multiple experts become idle, the fixed total pairwise distance (Eq. 2) forces larger distances among the active experts, which increases distance variance. Minimizing this variance under the zero-sum constraint pushes the router to re-activate idle experts. Through a single loss term operating on routing statistics rather than model weights, DO-loss thus addresses both collapse and specialization with negligible computational overhead.

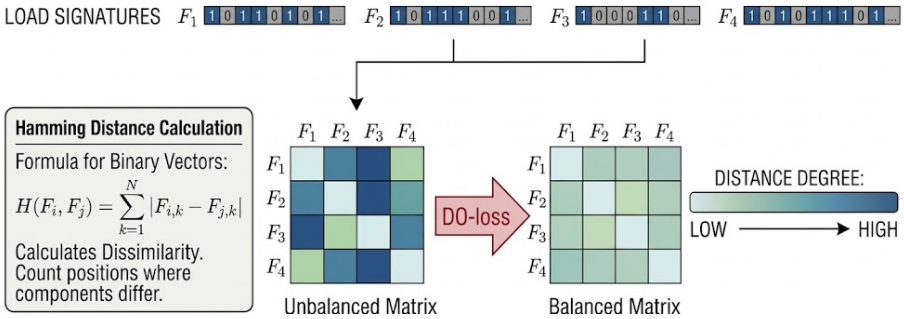


Figure 1: Token assignment histories encoded as binary load signatures. While the total sum of the pairwise Hamming distances remains constant under top-K routing, the DO-loss drives the distances between all pairs of experts toward equidistance to penalize high routing overlap.

While DO-loss prevents expert collapse, it does not guarantee strict load balancing. Because specialized experts attract different volumes of tokens, load imbalance is an inherent byproduct of specialization. In expert-parallel deployments, this imbalance has two interacting effects: overloaded ranks become stragglers that limit overall throughput, while the excess activations they must hold create memory footprint disparities that increase the risk of out-of-memory (OOM) failures at scale. To close this gap, we introduce the Replica Expert Mechanism (REM). REM first replicates hot experts onto underloaded ranks to rebalance the static workload, and then routes incoming tokens to the shortest available queue at runtime to absorb residual fluctuations. This allows the system to maintain near-uniform resource utilization without constraining expert specialization.

We evaluate our approach on 4.8BA0.5B and 30BA3B MoE models across six downstream benchmarks. By synchronizing the DO-loss distance matrix across data-parallel ranks, Global-DO-loss operates at the global-batch level and outperforms all baselines at both scales. Our method simultaneously resolves expert collapse and expert representation redundancy. REM reduces load imbalance below the level attained by traditional load balancing methods, recovering the throughput lost to specialization without any degradation in model quality.

2 Related Work

2.1 Routing and Auxiliary Load Balancing Loss

Token-choice routing Fedus et al. (2022) is standard in autoregressive LMs, where each token independently selects its Top-K experts, but is prone to collapse without constraints. The Switch Transformer Fedus et al. (2022) penalizes the product of per-expert routing fraction and mean probability; ST-MoE Zoph et al. (2022) adds a router z-loss for stability. Qiu et al. Qiu et al. (2025) show that computing these losses per device forces local uniformity that hinders specialization, and propose global synchronization to preserve semantic clustering. DeepSeek Wang et al. (2024a) uses per-expert bias terms updated by load deviation for implicit, gradient-free balancing. Expert-choice routing Zhou et al. (2022) guarantees perfect balance but introduces causal leakage in autoregressive settings Muennighoff et al. (2024).

2.2 Orthogonality in MoE

Orthogonality-based methods reduce representation redundancy by penalizing similarity among expert parameters or outputs. MOORE Hendawy et al. (2024) uses Gram-Schmidt regularization; AES-MoE Wang et al. (2025) regularizes output similarity without affecting load balance. However, the associated overhead restricts these methods to post-training or fine-tuning. ERNIE Baidu-ERNIE-Team (2025) and ERC-loss Lv et al. (2025) reduce this cost by operating on routing representations or coupling expert and router optimization.

2.3 System Efficiency and Expert Scheduling

Skewed token distributions require robust system-level scheduling. EfficientMoE Zeng et al. (2025) places replicas at the global-batch level but lacks per-step granularity. Pro-Prophet Wang et al. (2026) enables real-time micro-batch re-dispatch, though its fine-grained redistribution is time-consuming and its overlap of scheduling with computation depends on the compute-to-communication ratio.

3 Methods

3.1 Distributional Orthogonalization Loss (DO-loss)

3.1.1 Binary Load and Zero-Sum Constraint

For a batch of m tokens, the workload of expert i is a binary vector $F_i \in \{0,1\}^m$:

$$f_{it} = \mathbb{I}\left(i \in \text{Top-}K(\mathbf{x}_t)\right), \quad t \in \{1, \ldots, m\} \tag{1}$$

Let s_{it} denote the raw Sigmoid-gated routing score for expert i on token t. The normalized routing probability is $p_{it} = s_{it} / \sum_{i'} s_{i't}$, which satisfies $\sum_i p_{it} = 1$. After Top-K selection, only the p_{it} of the K chosen experts retain their normalized values; all others are set to zero and the K values are re-normalized.

With Top-K routing and n experts, exactly k bits are set per token, so the total pairwise Hamming distance over the batch is fixed:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \|F_i - F_j\|_1 = 2mk(n-k) \tag{2}$$

This zero-sum constraint is the structural key: minimizing the *variance* of pairwise distances drives all pairs toward the same distance without requiring an explicit target. It also explains DO-loss's anti-collapse property. When some experts are idle ($F_i = \mathbf{0}$), the zero-sum constraint forces the active experts to exhibit larger pairwise distances to maintain the fixed total, which increases this variance. Since DO-loss minimizes this variance, idle-expert configurations are inherently penalized: the gradient of the loss with respect to the routing probability p_{it} of an idle expert i is negative for tokens assigned to active experts, pushing

the router to re-activate the idle expert. A formal derivation is provided in Appendix C. The original loss is:

$$\mathcal{L}_{\text{origin_DO}} = \sum_{i \neq j} \|F_i - F_j\|_1^2 \tag{3}$$

3.1.2 *Equalization Loss and Gradient Derivation*

Discrete assignments block gradient flow. We apply Straight-Through Estimation (STE) by replacing each F_i with $P_i + \text{sg}[F_i - P_i]$, where P_i is the continuous routing probability vector and $\text{sg}[\cdot]$ denotes the stop-gradient operator, which yields a zero gradient. Substituting into Eq. 3:

$$\mathcal{L}_{\text{DO-STE}} = \sum_{i \neq j} \left\| P_i + \text{sg}[F_i - P_i] - P_j - \text{sg}\left[F_j - P_j\right] \right\|_1^2 \tag{4}$$

Taking the gradient and noting that $f_{it} \in \{0, 1\}$ implies $\text{sgn}(f_{it} - f_{jt}) = f_{it} - f_{jt}$:

$$\nabla \mathcal{L}_{\text{DO-STE}} = \sum_{i \neq j} 2\|F_i - F_j\|_1 \sum_{t=1}^{m} (f_{it} - f_{jt}) \nabla (p_{it} - p_{jt}) \tag{5}$$

3.1.3 *Surrogate Loss Implementation*

We factor the loss into a detached magnitude G_{ij} and a gradient direction term C_{ij}:

$$G_{ij} = \|F_i - F_j\|_1, \quad C_{ij} = \sum_{t=1}^{m} p_{it}(f_{it} - f_{jt}) \tag{6}$$

Since $C_{ii} = 0$, the $i \neq j$ constraint is automatically satisfied. With a normalization scalar $\frac{1}{m^2 k}$ (derived in Appendix A), the final loss is:

$$\mathcal{L}_{\text{DO}} = \frac{1}{m^2 k} \sum_{i=1}^{n} \sum_{j=1}^{n} G_{ij} \cdot C_{ij} \tag{7}$$

3.1.4 *Global-Level Regularization via Cross-DP Synchronization*

Qiu et al. (2025) show that routing statistics over larger batches produce more stable load distributions. DO-loss achieves the same effect by exploiting its decoupled structure (Eq. 7). We synchronize the distance matrix G_{ij} across the DP group via All-Reduce and keep the gradient direction indicator C_{ij} local:

$$G_{ij}^d = \|F_i^d - F_j^d\|_1, \qquad G_{ij}^g = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} G_{ij}^d \tag{8}$$

The global DO-loss on device d then applies the shared magnitude to the local direction indicator:

$$\mathcal{L}_{\text{global_DO}}^{(d)} = \frac{1}{m^2 k} \sum_{i=1}^{n} \sum_{j=1}^{n} G_{ij}^g \cdot C_{ij}^d \tag{9}$$

The loss computation complexity of $\mathcal{O}(mn^2)$ is practically negligible because the expert count n (64–256) is far smaller than the hidden dimension, and the quadratic dependence is on n rather than the model width.

The overall training objective is:

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \lambda \cdot \mathcal{L}_{\text{DO}} \tag{10}$$

where λ is the DO-loss coefficient and $\mathcal{L}_{CE}$ is the standard cross-entropy loss. No additional load-balancing auxiliary terms are used, as DO-loss alone addresses both collapse and specialization.

3.2 Replica Expert Mechanism (REM)

MoE training exhibits stable phases during which hot-expert identities remain consistent across steps Cong et al. (2024). REM uses this property by replicating hot experts to distribute their token load and subsequently routing tokens to the least-loaded replicas in real time.

This mechanism operates through two decoupled loops (pseudocode in Appendix B; architecture in Figure 2). A metric collector asynchronously aggregates EP load statistics using a moving average window. At each global-batch boundary, a controller evaluates these statistics to greedily place replicas on underloaded ranks based on a Benefit Score (Eq. 11). Within each micro-batch, a local dispatcher routes tokens independently of the global controller to prevent synchronization bottlenecks.

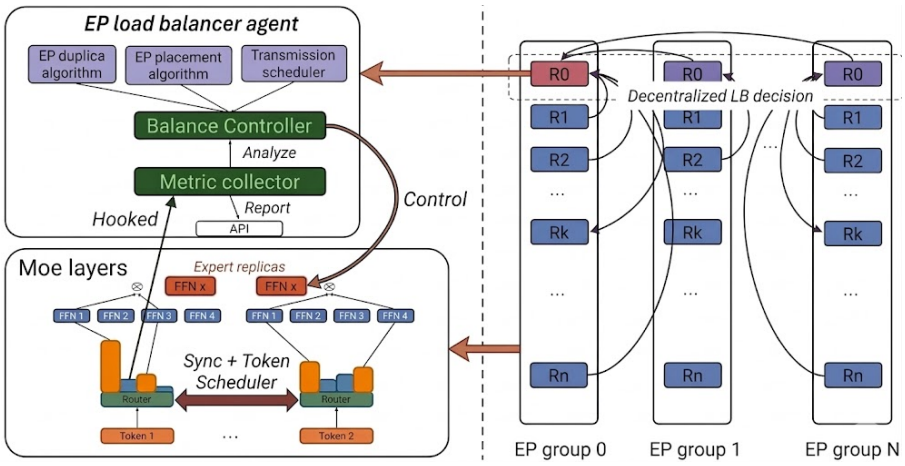


Figure 2: The architecture of REM. **Left:** The global-batch controller (top) allocates replicas once per global batch; a collector asynchronously retrieves metrics from the router. The micro-batch dispatcher (bottom) synchronizes router information and routes tokens in real time. **Right:** Load statistics are gathered hierarchically—the coordinator (R0) of each EP group aggregates intra-group rank loads locally, and R0 nodes subsequently exchange summaries across EP groups to construct a global view. This two-level aggregation avoids all-to-all collection across all ranks while enabling decentralized load-balancing decisions at every rank in real time.

3.2.1 *Global-Batch Replica Allocation*

REM calculates a Benefit Score S for replica placement. The score S represents the optimal number of tokens that the receiving replica r_r can accept from r_s to maximize load balance with replica expert e.

Assuming a capacity of D replica expert slots per rank, the scheduler iteratively identifies the least-loaded rank as the receiver r_r and greedily assigns the expert that maximizes S. This allocation process continues until all available slots across every rank are fully occupied.

We define the ideal token threshold τ as the average load across all ranks $\tau = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} L_r$. The score S is then calculated as:

$$S(e, r_s, r_r, \tau) = \begin{cases} \min(\tau - L_r, T_e, L_s - \tau), & \text{if } L_r < \tau \text{ and } L_s > \tau \\ \min\left(\left\lfloor \frac{L_s - L_r}{2} \right\rfloor, T_e\right), & \text{if } L_r \geq \tau \text{ and } L_s > L_r \\ -1, & \text{otherwise} \end{cases} \tag{11}$$

where L_s and L_r denote the historical loads of the sending and receiving ranks, respectively, while T_e represents the token volume of expert e. The scheduler identifies the (e, r_s) pair that maximizes S, updating the load estimates before the subsequent iteration.

3.2.2 *Dynamic Routing via Micro-step Scheduling*

Although token scheduling at the micro-batch level adheres to the same greedy principle as global-batch allocation, it evaluates the real-time load distribution instead of historical statistics. To obtain a globally consistent instantaneous view, all GPUs synchronize the current depths of the queues via an AllGather operation at the beginning of each micro-batch. This fine-grained signal enables the dispatcher to route tokens dynamically in response to transient load fluctuations that a sliding-window average inherently misses. The overhead of this AllGather operation is negligible compared to the AlltoAll dispatch operation.

The dispatcher executes exactly D iterations. In each iteration, it sorts the ranks according to the current queue depths and processes them in order from the lightest load to the heaviest. During this traversal, tokens are greedily routed to the expert replica that yields the maximum score for each rank. To avoid prohibitive on-device overhead, the dispatcher does not resort the ranks after each individual allocation. The entire sequence of token dispatch executes on the device without requiring intervention from the global controller.

REM operates purely as a system-level scheduler and does not alter the mathematical training trajectory. Replicas hold asynchronously synchronized copies of the original expert parameters; gradients computed on replicas are accumulated back to the original rank after each global step, with optimizer states residing exclusively on the original rank. A 10,000-step precision test confirmed gradient alignment to three decimal places. Appendix B provides the complete pseudocode and formal equivalence conditions.

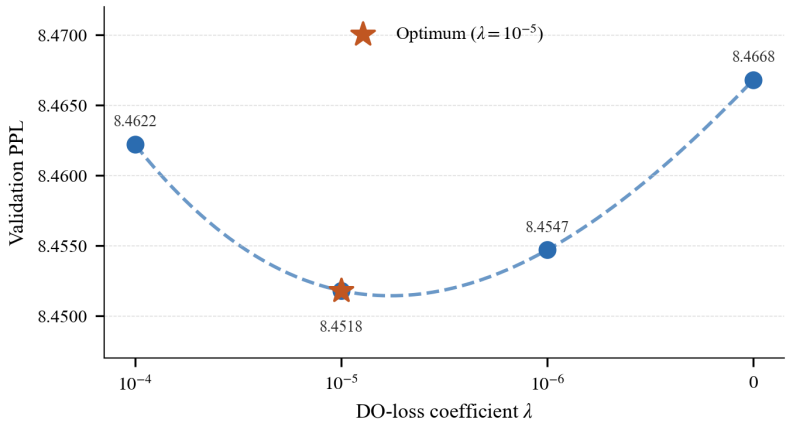


Figure 3: **Ablation of DO-loss coefficient λ.** Validation perplexity as a function of λ (4.8BA0.5B, 100B tokens). λ=0 disables DO-loss, leading to expert collapse via redundancy. Excessively large λ causes the high-dimensional pairwise regularization to overwhelm the cross-entropy signal. The optimum λ=10^{-5} is approximately two orders of magnitude smaller than typical load-balancing loss coefficients.

4 Experimental Setup

4.1 Model Architecture

We use the Qwen3MoE architecture Yang et al. (2025) at two scales, following its Active Parameter (A) naming convention. Both models use Sigmoid-gated Top-8 routing over 128 experts (K=8, N=128), a vocabulary of 120,832, and 32-head GQA with 4 query groups. The 4.8BA0.5B model has 15 layers and a hidden size of 1024; the 30BA3B model has 48 layers and a hidden size of 2048. Expert intermediate dimension is 768 for both.

4.2 Baseline Methods

We compare against five routing strategies spanning the main paradigms.

Local-LBL uses the Switch Transformer auxiliary loss Fedus et al. (2022) $\mathcal{L}_{\text{LBL}} = \sum_i f_i p_i$, where f_i is the per-expert routing fraction and p_i the mean routing probability, computed independently on each device. **Global-LBL** Qiu et al. (2025) synchronizes f_i across data-parallel ranks via All-Reduce before computing the same loss, preserving semantic clustering at the macro level while enforcing global balance. **Auxfree** Wang et al. (2024a) maintains per-expert bias terms b_i updated proportionally to the deviation from the ideal uniform load $b_i \leftarrow b_i + \eta \cdot (L_i - \bar{L})$, achieving balance through implicit feedback without gradient interference. **Orth-loss** Baidu-ERNIE-Team (2025) penalizes the cosine similarity between normalized routing weight vectors: $\mathcal{L}_{\text{orth}} = \sum_{i \neq j} (\hat{w}_i^\top \hat{w}_j)^2$. **Seq-LBL** computes $\mathcal{L}_{\text{LBL}}$ at the per-sequence level with coefficient 10^{-3}, included only in MaxVio and throughput evaluations as an extreme balancing baseline.

4.3 Training Settings

All experiments use the Megatron-Core framework on NVIDIA H20 GPUs with SwiGLU expert FFNs Shazeer (2020), AdamW optimizer (β_1=0.9, β_2=0.95, weight decay 0.1), and a cosine-decay learning rate schedule peaking at 3×10^{-4} and decaying to 3×10^{-5} after 2000 linear warmup steps. Sequence length is 4096, global batch size 512 (micro-batch 4). The 4.8BA0.5B and 30BA3B models train for 100B and 200B tokens, respectively.

Models are trained on a 2T-token proprietary corpus. Table 1 summarizes its domain composition.

Domain	Description
GitHub code	Multi-language code repositories
STEM data	Academic papers and textbooks
Chinese/English web	General web crawl
Academic books	Multi-discipline textbooks
Encyclopedias	Encyclopedia-style knowledge
Q&A banks	Exam and quiz datasets

Table 1: Domain composition of the pretraining corpus.

4.4 Evaluation Benchmarks

We evaluate on six benchmarks spanning reasoning, knowledge, and mathematics. AGIEval Zhong et al. (2024) tests human-level reasoning with problems from standardized exams. MMLU Hendrycks et al. (2021) and MMLU-Pro Wang et al. (2024b) measure broad multitask language understanding, with the latter offering a more challenging and discriminative variant. CEval Huang et al. (2023) covers Chinese-language academic knowledge across multiple disciplines. GSM8K Cobbe et al. (2021) assesses multi-step arithmetic reasoning via grade-school math problems. TriviaQA Joshi et al. (2017) evaluates open-domain factual recall from trivia questions.

5 Results and Analysis

5.1 Regularization Coefficient Sensitivity

Figure 3 reveals a U-shaped relationship between the DO-loss coefficient and validation perplexity, with an optimal value at $\lambda = 10^{-5}$. This value is orders of magnitude smaller than the 10^{-3} to 10^{-4} range typical of load-balancing losses. The gap points to a structural difference: DO-loss simultaneously regularizes all expert pairs, and this high-dimensional pairwise constraint introduces substantially larger gradient variance than a scalar auxiliary loss over n fractions. A much smaller coefficient is therefore required to prevent this variance from destabilizing training. Setting λ too high causes the pairwise objective to overwhelm the cross-entropy signal, while setting it too low fails to encourage expert specialization

Model	Method	AGIEval	MMLU	MMLU-PRO	CEval	GSM8K	TriviaQA	PPL (↓)
4.8BA0.5B	Local-LBL	22.24	36.01	10.95	36.63	12.71	23.56	8.4650
	Global-LBL	23.02	36.80	11.96	37.24	13.19	23.75	8.4604
	Orth-loss	22.60	34.44	11.42	34.63	12.25	23.47	8.4664
	Auxfree	23.55	36.85	12.02	38.63	13.57	23.06	8.4620
	DO-loss	24.05	36.33	13.50	37.10	13.51	23.89	8.4518
	Global-DO-loss	24.13	37.18	13.88	39.64	14.55	24.03	8.4433
30BA3B	Global-LBL	41.06	54.23	21.05	57.32	56.07	52.91	6.0989
	Auxfree	41.35	53.54	22.32	57.76	56.78	57.22	6.1140
	Global-DO-loss	41.51	54.96	22.75	58.35	57.14	58.27	6.0856

Table 2: Validation perplexity and benchmark accuracy at two scales (100B and 200B tokens). At 30BA3B, we additionally evaluate Global-LBL for completeness; Global-LBL underperforms Auxfree on most benchmarks, particularly on knowledge-intensive tasks such as TriviaQA, indicating that load-balancing objectives can conflict with knowledge-intensive tasks at larger scales.

and leads to expert collapse. A finer-grained sweep on a 2B/0.2B model (Appendix F.1) refines the optimum to $\lambda = 5\times10^{-6}$, roughly two orders of magnitude smaller than typical load-balancing loss coefficients.

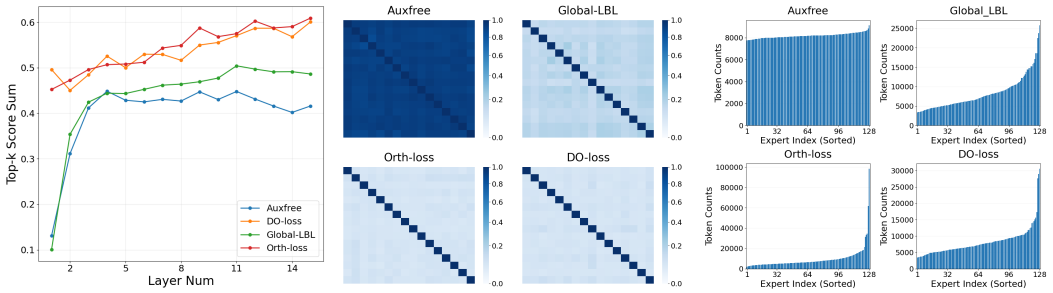
5.2 Downstream Benchmark Performance

As shown in Table 2, Global-DO-loss leads on every metric at the 4.8BA0.5B scale. Orth-loss yields the worst performance, which stems directly from expert collapse preventing the model from utilizing its full capacity. Auxfree enforces load balance through per-expert bias terms that leave the main gradient undisturbed, avoiding interference from auxiliary losses and making it the strongest baseline at the 4.8BA0.5B scale. The performance gap between DO-loss and Global-DO-loss confirms that device-local routing statistics systematically underestimate cross-expert overlap. Due to compute constraints, we compare only against Auxfree and Global-LBL at the 30BA3B scale, where Global-DO-loss maintains a consistent advantage. Global-LBL underperforms Auxfree on most metrics at this scale, indicating that the gains of Global-DO-loss stem from orthogonalization rather than global statistics alone. Validation perplexity also improves monotonically with larger effective batch sizes for G_{ij} statistics (see Appendix F.2 for the full sweep), consistent with the findings of Qiu et al. (2025).

5.3 Ablation Study

To isolate the contributions of the DO objective and global-batch statistics, we compare four variants on the 4.8BA0.5B model: local and global versions of both the load-balancing loss (Local-LBL) and DO-loss. As shown in Table 2, global statistics benefit both methods (Global-LBL $>$ Local-LBL, Global-DO-loss $>$ DO-loss). More importantly, the DO objective strictly dominates Local-LBL at both the local and global levels, with Global-DO-loss achieving the best performance across all seven metrics. This confirms that the pairwise orthogonalization objective, rather than the use of global statistics, is the primary driver of the observed improvements.

We further examine how the benefit of Global-DO-loss over Auxfree varies with expert granularity (see Appendix F.3 for full results). On a 2B/0.2B model, Global-DO-loss trails Auxfree at 32 experts but pulls ahead as granularity increases to 64 and 128 experts, consistent with the trend toward fine-grained architectures in modern LLMs.



(a) Top-K score sum (b) CKA similarity matrix (c) Load distributions

Figure 4: Expert behavior in the trained 4.8BA0.5B model. **(a)** Top-K routing score sum per token: higher values indicate the router assigns tokens with greater confidence, reflecting stronger expert specialization. **(b)** CKA similarity matrix (Appendix D) at layer 4, sampled every 8th expert (16×16 grids). CKA measures functional similarity between expert outputs on the same inputs; high off-diagonal values indicate representational redundancy, while a near-identity matrix indicates functional diversity. **(c)** Distribution of token counts per expert at layer 4.

5.4 Expert Behavior: A Comparative Analysis

Figure 4 evaluates the methods across routing decisiveness (Panel a), inter-expert representational similarity (Panel b), and per-expert token load distribution (Panel c). None of the baselines satisfies all three criteria simultaneously, reflecting inherent design trade-offs.

Auxfree produces a nearly flat load histogram but yields the highest CKA off-diagonal values and lowest Top-K sums, indicating that strict uniform token distribution suppresses the selective routing needed for expert differentiation. Global-LBL adopts global token statistics that allow some semantic clustering, reducing off-diagonal CKA and increasing Top-K sums vs. Auxfree, but its balancing objective still restricts full specialization.

Weight-space orthogonality promotes representational diversity without governing functional utilization. Orth-loss achieves the sparsest CKA and highest Top-K sums, yet its token load exhibits a severely skewed exponential tail—parameter-space differentiation lacks constraints on token allocation, risking collapse despite near-ideal representational structures. DO-loss matches Orth-loss on CKA sparsity and Top-K sums while maintaining a more controlled load distribution.

To complement the visual CKA comparison, we quantitatively measure CKA across all 128 expert pairs (Appendix D). The results confirm: Auxfree has the highest redundancy in shallow layers, Orth-loss achieves sparsest early-layer representations, and Global-DO-loss attains the best separation in deeper layers with smallest variance, consistent with its more decisive routing.

5.5 REM: Load Balancing and System Efficiency

Figure 5 shows Auxfree converging quickly to low MaxVio. Seq-LBL, which computes the auxiliary loss at the sequence level to enforce stricter per-sample balancing, settles at a slightly higher MaxVio. Unaugmented DO-loss remains persistently elevated—reflecting structural specialization, not slow convergence. REM resolves this: a single replica closes most of the throughput gap and drives MaxVio below Auxfree; two replicas achieve near-perfect balance. Both configurations maintain slightly lower throughput than Auxfree due to replica weight-copy overhead.

The memory cost of replication is modest and REM communication overhead is equally negligible: the AllGather for queue depths transmits only $|\mathcal{R}|$ integers per EP group. REM was enabled for all methods in throughput evaluations; for methods already achieving near-perfect load balance (Seq-LBL, Auxfree), REM provides only marginal additional gains.

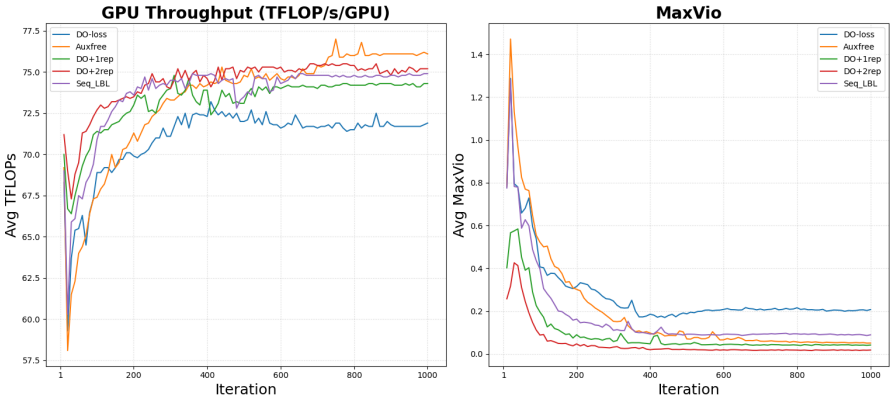


Figure 5: **Ablation of REM replica count.** GPU throughput and MaxVio over 1000 steps on H20 GPUs (EP= 8, 4.8BA0.5B) with 0, 1, and 2 replica experts. A single replica recovers most of the throughput gap; two replicas drive MaxVio to near zero. MaxVio (Appendix E) measures the fractional load excess of the busiest rank relative to the ideal average.

We further compare REM against two representative expert-scheduling methods, Efficient-MoE Zeng et al. (2025) and Pro-Prophet Wang et al. (2026), on both H20 and H800 GPU clusters under two expert-parallelism configurations; REM consistently outperforms both baselines. Appendix F.4 provides full throughput results, overhead measurements, and a detailed analysis of the scheduling comparison.

6 Conclusion

MoE architectures scale model capacity by activating a token-specific expert subset, yet their effectiveness is limited by two intertwined challenges: expert collapse reduces the effective parameter count, while representation redundancy wastes the capacity that remains. Existing load balancing methods mitigate collapse and maintain system resource efficiency, but suppress specialization. Parameter orthogonalization, on the other hand, encourages functional diversity yet provides no mechanism to prevent collapse itself. Naively combining these independent solutions introduces conflicting gradients that destabilize training, revealing a deeper issue: both attempt to govern routing behavior through the training objective, coupling concerns that operate at fundamentally different levels.

We argue that load balancing is not the only means to resolve collapse, and that system efficiency should not be enforced at the algorithmic level. To this end, we introduce DO-loss, which encourages specialization and mitigates collapse by penalizing routing trajectory overlap rather than weight similarity. For system efficiency, REM restores workload balance through adaptive replica placement and real-time token dispatching, entirely decoupled from the routing objective.

Experiments on both 4.8BA0.5B and 30BA3B MoE models show that this decoupled design consistently outperforms the evaluated routing algorithms on downstream tasks while maintaining comparable training throughput. This decoupling also makes the design adaptable to diverse hardware paradigms: on wafer-scale architectures where token routing and expert placement decouple differently from GPU expert parallelism, DO-loss alone can provide acceptable load balance, making REM optional. More broadly, our results suggest a design principle for future MoE systems: MoE effectiveness and system efficiency operate at fundamentally different abstraction levels requiring separate mechanisms. Routing objectives should focus solely on functional data partitioning, while load imbalance should be treated as an infrastructure problem addressed by system-level mechanisms such as replica scheduling and capacity-aware dispatching. Disentangling these concerns is key to ensuring that theoretical parameters reliably translate into actual model capacity.

Acknowledgements

We thank Tencent for providing the computational resources and infrastructure that made this work possible. We also thank the anonymous reviewers for their insightful comments and constructive feedback, which helped improve the quality of this paper.

References

Baidu-ERNIE-Team. Ernie 4.5 technical report. Technical report, Baidu, 2025.

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. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20820–20835, 2021.

Peizhuang Cong, Aomufei Yuan, Shimao Chen, Yuxuan Tian, Bowen Ye, and Tong Yang. Prediction is all moe needs: Expert load distribution goes from fluctuating to stabilizing. *arXiv preprint arXiv:2404.16914*, 2024.

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.

Mohamed Hendawy, Jiahao Li, and William Yang Wang. Moore: Orthogonalizing experts in mixture-of-experts models. arXiv preprint arXiv:2406.09177, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, et al. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in neural information processing systems*, 36:62991–63010, 2023.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529. PMLR, 2019.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

J. Li, H. Wang, and Y. Zhang. Diversifying the mixture-of-experts representation for language models with orthogonal optimizer. *arXiv preprint arXiv:2310.09762*, 2023.

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

Yuhang Liu, Dong Li, Xing Wei, and Huchuan Lu. Omoe: Orthogonal mixture of experts. arXiv preprint arXiv:2405.19669, 2024b.

Ang Lv, Jin Ma, Yiyuan Ma, and Siyuan Qiao. Coupling experts and routers in mixture-of-experts via an auxiliary loss. *arXiv preprint arXiv:2512.23447*, 2025.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. OLMoE: Open mixture-of-experts language models. *arXiv preprint arXiv:2409.02060*, 2024.

Zihan Qiu, Zeyu Huang, Bo Zheng, Kaiyue Wen, Zekun Wang, Rui Men, Ivan Titov, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Demons in the detail: On implementing load balancing loss for training specialized mixture-of-expert models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5005–5018, 2025.

Samyam Rajbhandari, Conglong Li, Zhewei Yao, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, Shaden Smith, and Yuxiong He. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale. In *International Conference on Machine Learning*, pp. 18343–18358. PMLR, 2022.

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. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–15, 2019.

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *arXiv preprint arXiv:2408.15664*, 2024a.

Wei Wang, Zhiquan Lai, Dongsheng Li, Shengwei Li, Weijie Liu, Keshi Ge, Ao Shen, and Huayou Su. Pro-prophet: A systematic load balancing method for efficient parallel training of large-scale moe models. *IEEE Transactions on Parallel and Distributed Systems*, 2026.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024b.

Z. Wang, Y. Li, H. Zhang, et al. Advancing expert specialization for better moe. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

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

Yan Zeng, Chengchuang Huang, Yipeng Mei, Lifu Zhang, Teng Su, Wei Ye, Wenqi Shi, and Shengnan Wang. Efficientmoe: Optimizing mixture-of-experts model training with adaptive load balance. *IEEE transactions on parallel and distributed systems*, 36(4):677–688, 2025.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. In *Findings of the association for computational linguistics: NAACL 2024*, pp. 2299–2314, 2024.

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.

A DO-loss Normalization Coefficient

We first clarify the definition of p_{it}. Let s_{it} be the raw Sigmoid-gated routing score. Then $p_{it} = s_{it} / \sum_{i'} s_{i't}$ is the normalized routing probability, satisfying $\sum_i p_{it} = 1$. After Top-K selection, only the probabilities of the K chosen experts are retained and re-normalized. Consequently, $\bar{s}_{\text{all}} = \frac{1}{m} \sum_t \sum_i p_{it} = 1$ for every token, which is why this term naturally cancels in the derivation below.

We derive the normalization coefficient $\frac{1}{m^2 k}$ in Eq. 7 by evaluating the scale of the unnormalized loss at the balanced equilibrium.

Step 1: Sum of G_{ij}. $G_{ij} = \|F_i - F_j\|_1$, so the sum over all pairs follows from the zero-sum constraint (Eq. 2):

$$\sum_{i,j} G_{ij} = 2mk(n-k) \tag{12}$$

Step 2: Sum of C_{ij}. Expanding $C_{ij} = \sum_t p_{it}(f_{it} - f_{jt})$ and summing over all pairs:

$$\begin{aligned} \sum_{i,j} C_{ij} &= \sum_t \sum_i p_{it} \big(n f_{it} - \underbrace{\sum_j f_{jt}}_{=k}\big) \\ &= n \sum_t \underbrace{\sum_i p_{it} f_{it}}_{\bar{s}_{K,t}} - km \cdot \underbrace{\frac{1}{m} \sum_t \sum_i p_{it}}_{\bar{s}_{\text{all}}} \end{aligned} \tag{13}$$

where $\bar{s}_{K,t}$ is the sum of selected routing scores for token t and $\bar{s}_{\text{all}}$ is the mean total score per token. Denoting $\bar{s}_K = \frac{1}{m} \sum_t \bar{s}_{K,t}$:

$$\sum_{i,j} C_{ij} = m(n\bar{s}_K - k) \tag{14}$$

Step 3: Scale at equilibrium. At the DO-loss minimum, all pairs share the same distance magnitude and gradient weight, so each G_{ij} and C_{ij} equals its mean:

$$\bar{G} = \frac{2mk(n-k)}{n^2}, \qquad \bar{C} = \frac{m(n\bar{s}_K - k)}{n^2} \tag{15}$$

The unnormalized sum over all n^2 pairs is:

$$\mathcal{L}_{\text{unnorm}} \approx n^2 \bar{G}\bar{C} = \frac{2m^2 k(n-k)(n\bar{s}_K - k)}{n^2} \tag{16}$$

Since $k \ll n$, we have $(n-k) \approx n$ and $(n\bar{s}_K - k) \approx n\bar{s}_K$, so:

$$\mathcal{L}_{\text{unnorm}} \approx 2m^2 k \bar{s}_K \sim \mathcal{O}(m^2 k) \tag{17}$$

Dividing by $m^2 k$ normalizes the loss to $\mathcal{O}(1)$ up to the constant $2\bar{s}_K$. Here $\bar{s}_K$ is the per-token mean of the Top-K routing score sum. As specialization intensifies, routing concentrates, and the sum converges to $\bar{s}_K \approx 1$. Together with the factor of 2 from pairwise symmetry, both are absorbed into the normalization. The resulting coefficient $\frac{1}{m^2 k}$ makes DO-loss scale-free and directly comparable to the cross-entropy loss without further tuning.

B Replica Expert Mechanism

REM uses two decoupled control loops operating at different time granularities, plus a training trajectory invariance mechanism that ensures mathematical equivalence with non-replicated execution.

B.1 Global-Batch Replica Allocation

At each global-batch boundary, the controller greedily fills replica slots by repeatedly picking the least-loaded rank and placing the expert that maximizes the Benefit Score S (Eq. 11 in the main text), which represents the optimal number of tokens that the receiving replica r_r can accept from r_s to maximize load balance. Load estimates are updated after each placement until all D slots per rank are filled or no beneficial placement remains.

Algorithm 1 REM Global-Batch Replica Allocation

Require: Historical rank loads $\{L_r\}$, expert token volumes $\{T_e\}$, per-rank capacity D
Ensure: Replica assignment $\mathcal{A} \leftarrow \emptyset$

$\tau \leftarrow \frac{1}{|\mathcal{R}|} \sum_r L_r$
repeat
  $r_r \leftarrow \arg\min_{\{r:|\mathcal{A}_r|<D\}} L_r$
  $(e^*, r_s^*) \leftarrow \arg\max_{e, r_s \neq r_r} S(e, r_s, r_r, \tau)$
  $\Delta \leftarrow S(e^*, r_s^*, r_r, \tau)$
  if $\Delta \leq 0$ **then break**
  $\mathcal{A} \leftarrow \mathcal{A} \cup \{(e^*, r_s^*, r_r)\}$
  $L_{r_r} \mathrel{+}= \Delta$
  $L_{r_s^*} \mathrel{-}= \Delta$
  $T_{e^*} \mathrel{-}= \Delta$
until all ranks full

B.2 Micro-Batch Token Dispatching

Within each micro-batch, a local dispatcher routes tokens independently of the global controller to prevent synchronization bottlenecks. After syncing queue depths via AllGather, the dispatcher repeatedly sorts all ranks by current load and traverses them from lightest to heaviest. For every rank, it assigns the highest-scoring replica tokens and updates the queue depth Q_r. Passes repeat D times. The AllGather overhead is negligible compared to the AlltoAll dispatch, as it transmits only $|\mathcal{R}|$ integers per EP group.

B.3 Training Trajectory Invariance

REM operates purely at the system level and preserves the mathematical training trajectory. The invariance relies on the following implementation design:

Parameter synchronization. Replica experts use independent memory allocations initialized to zero. Before each forward pass, replica parameters are overwritten with the current parameters of the original expert via asynchronous peer-to-peer communication. Fetching of early-layer parameters begins before the forward pass, and subsequent layers are fetched during the computation of prior layers, overlapping communication with computation.

Gradient accumulation. Each micro-batch computes gradients independently on all replicas. After each global step, the accumulated gradients from all replicas are sent back to the original expert rank and summed. Because replicas hold a frozen copy of the original parameters at the time of the forward pass, the gradients computed remotely are numerically identical to those that would have been computed locally.

Optimizer state. Optimizer states (e.g., Adam moments exp_avg and exp_avg_sq) reside exclusively on the original expert rank. Standard DDP all-reduce is applied only to the

Algorithm 2 REM Micro-Batch Token Dispatching

Require: Replica assignment $\mathcal{A}$, real-time expert token counts $\{T_e\}$, per-rank capacity D
Ensure: Token-to-rank mapping $\mathcal{M} \leftarrow \emptyset$

$T_e \leftarrow \text{AllGather}(\{T_e\})$
$Q_r \leftarrow \sum_{e \in r} T_e$
$\tau \leftarrow \frac{1}{|\mathcal{R}|} \sum_r Q_r$
for $d = 1$ **to** D **do**
    sort ranks by Q_r ascending
    for each rank r in sorted order **do**
        $(e^*, r_s^*) \leftarrow \arg\max_{(e, r_s) : (e, r_s, r) \in \mathcal{A}} S(e, r_s, r, \tau)$
        $\Delta \leftarrow S(e^*, r_s^*, r, \tau)$
        $\mathcal{M} \leftarrow \mathcal{M} \cup \{(r, e^*, \Delta)\}$
        $Q_r \mathrel{+}= \Delta$
        $Q_{r_s^*} \mathrel{-}= \Delta$
        $T_{e^*} \mathrel{-}= \Delta$
    end for
end for

accumulated gradient on the original expert. A 10,000-step precision test confirmed gradient alignment to three decimal places.

Equivalence conditions. The training trajectory equivalence holds under three assumptions: (a) replica parameters are fully synchronized with the original expert before each forward pass, (b) gradient accumulation across replicas uses summation, and (c) optimizer updates occur only on the original expert rank.

C Gradient Analysis of DO-loss

C.1 Zero-Sum Constraint and Variance Minimization

Under Top-K routing with n experts and K selections per token over a batch of m tokens, the total number of bits set across all expert load signatures is exactly mK. The sum of all pairwise Hamming distances is therefore constant:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\|F_i - F_j\|_1 = 2mK(n-K) \tag{18}$$

Minimizing the sum of squared pairwise distances $\sum_{i\neq j}\|F_i - F_j\|_1^2$ under this fixed-sum constraint is equivalent to minimizing the variance of the pairwise distances. The global minimum is achieved when all pairwise distances are equal to $d^* = \frac{2mK(n-K)}{n^2}$.

C.2 Anti-Collapse Gradient

When expert i is idle ($F_i = \mathbf{0}$), its distance to any active expert j ($F_j \neq \mathbf{0}$) is $\|F_i - F_j\|_1 = \|F_j\|_1 > 0$. Under the zero-sum constraint, these nonzero distances increase the variance of the pairwise distance distribution. The gradient of the DO-loss with respect to the routing probability p_{it} for an idle expert i is:

$$\nabla_{p_{it}}\mathcal{L}_{\text{DO}} \propto -\sum_{j:F_j\neq\mathbf{0}}\|F_i - F_j\|_1 \cdot f_{jt} \tag{19}$$

The negative sign indicates that the loss pushes p_{it} to increase for tokens that are currently assigned to active experts. As the router raises these probabilities, the idle expert begins to receive tokens, which reduces the distances between it and the previously active experts and thereby lowers the overall variance. This negative feedback loop makes expert collapse self-correcting without requiring an explicit load-balancing term.

C.3 Equidistance and Optimal Specialization

We now formalize the relationship between pairwise equidistance and expert specialization. Let $c_{ij} = \sum_t f_{it}f_{jt}$ denote the co-occurrence count of experts i and j (the number of tokens they both select). The Hamming distance can be expressed as:

$$\|F_i - F_j\|_1 = \|F_i\|_1 + \|F_j\|_1 - 2c_{ij} \tag{20}$$

When all pairwise distances are equal ($\|F_i - F_j\|_1 = d^*$ for all $i \neq j$), summing this equality over all $j \neq i$ for a fixed i yields $\|F_i\|_1 = \frac{mK}{n}$ for every expert i. That is, equidistance implies uniform per-expert token counts. Substituting into the Hamming distance formula gives $c_{ij} = \frac{mK(K-1)}{n(n-1)}$ for all $i \neq j$, which is the minimum possible co-occurrence under the constraints of Top-K routing. This means that under the equidistance condition, every pair of experts shares the smallest possible number of tokens.

Since the router naturally converges to assign semantically similar tokens to the same expert subsets, this minimal and uniform co-occurrence ensures that distinct semantic patterns are processed by maximally disjoint combinations of experts. The result is a functionally diversified expert population in which each expert develops a specialized role.

D Centered Kernel Alignment (CKA)

Feature extraction. Each expert is a SwiGLU FFN Shazeer (2020) with input projections $W_1^{(i)}, W_3^{(i)} \in \mathbb{R}^{d \times d_e}$ and output projection $W_2^{(i)} \in \mathbb{R}^{d_e \times d}$. For input $\mathbf{x} \in \mathbb{R}^d$:

$$\mathrm{Expert}_i(\mathbf{x}) = \left(\sigma(\mathbf{x}W_1^{(i)}) \odot \mathbf{x}W_3^{(i)}\right) W_2^{(i)} \tag{21}$$

where $\sigma(\cdot)$ is SiLU. We use the intermediate gated representation $\sigma(\mathbf{x}W_1^{(i)}) \odot \mathbf{x}W_3^{(i)}$ as the feature vector: it combines gating magnitude with the value branch and therefore captures each expert's functional selectivity more faithfully than either branch alone or the final output.

A fixed probe set of m tokens is passed through every expert independently (bypassing the router). For expert i:

$$\mathbf{A}_i = \left[\sigma(\mathbf{x}_1 W_1^{(i)}) \odot \mathbf{x}_1 W_3^{(i)}; \cdots ; \sigma(\mathbf{x}_m W_1^{(i)}) \odot \mathbf{x}_m W_3^{(i)}\right]^\top \in \mathbb{R}^{m \times d_e} \tag{22}$$

Linear CKA. Linear CKA (Kornblith et al., 2019) compares the centered Gram structures of two feature matrices. First, the feature matrix $\mathbf{A}_i$ is centered along the token axis:

$$\tilde{\mathbf{A}}_i = \left(\mathbf{I}_m - \tfrac{1}{m}\mathbf{1}_m\mathbf{1}_m^\top\right)\mathbf{A}_i \tag{23}$$

To better illustrate the distinction between the Global-LBL approach and orthogonalization methods, we compute the square root of the standard metric. The final evaluation value is defined as:

$$\mathrm{CKA}(\mathbf{A}_i, \mathbf{A}_j) = \sqrt{\frac{\|\tilde{\mathbf{A}}_i^\top \tilde{\mathbf{A}}_j\|_F^2}{\|\tilde{\mathbf{A}}_i^\top \tilde{\mathbf{A}}_i\|_F \|\tilde{\mathbf{A}}_j^\top \tilde{\mathbf{A}}_j\|_F}} \tag{24}$$

Within the square root, the numerator represents the Hilbert–Schmidt inner product of the centered Gram matrices, while the denominator provides normalization to ensure invariance to isotropic scaling. A resulting value of 0 indicates that the two experts produce linearly independent representations, whereas a value of 1 signifies that the representations are identical up to an invertible linear transformation. Consequently, high off-diagonal values indicate that distinct experts extract highly correlated features for the same tokens, which serves as a direct symptom of representation redundancy.

Quantitative results. To complement the qualitative CKA heatmaps in the main text, Table 3 reports the mean and standard deviation of CKA similarity across all 128 expert pairs at Layers 4 and 8. Adopting 0.1 as a threshold for excessive similarity, Auxfree exhibits the highest redundancy, particularly in the shallower layer where its mean CKA substantially exceeds the threshold. Orth-loss achieves the sparsest representations in Layer 4, while Global-DO-loss attains the lowest mean and smallest variance in the deeper Layer 8, consistent with its more decisive and diversified routing.

Method	Layer 4 CKA mean (std)	Layer 8 CKA mean (std)
Auxfree	0.263 (0.031)	0.115 (0.023)
Global-LBL	0.089 (0.016)	0.077 (0.008)
Orth-loss	**0.084** (0.010)	0.084 (0.011)
Global-DO-loss	0.086 (0.010)	**0.071** (0.007)

Table 3: Mean and standard deviation of CKA cosine similarity across all 128 expert pairs. Lower values indicate less representational redundancy. Best results in **bold**.

E MaxVio Metric

We adopt **Maximum Violation (MaxVio)** to evaluate the degree of outlier behavior at the rank level. Specifically, each rank r within an EP group has a total load L_r, calculated as the sum of tokens across all experts residing on that rank. MaxVio measures the extent to which the load of the busiest rank deviates from the ideal uniform load τ:

$$\text{MaxVio} = \max_{r \in \mathcal{R}} \left| \frac{L_r}{\tau} - 1 \right|, \qquad \tau = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} L_r \tag{25}$$

The reported value represents the average across all MoE layers and micro-batches during a single training step. This metric directly captures the straggler effect, as any rank processing more tokens than τ stalls the entire EP group. For instance, a MaxVio value of 0.1 indicates that the most heavily loaded rank processes 10% more tokens than the ideal distribution on average.

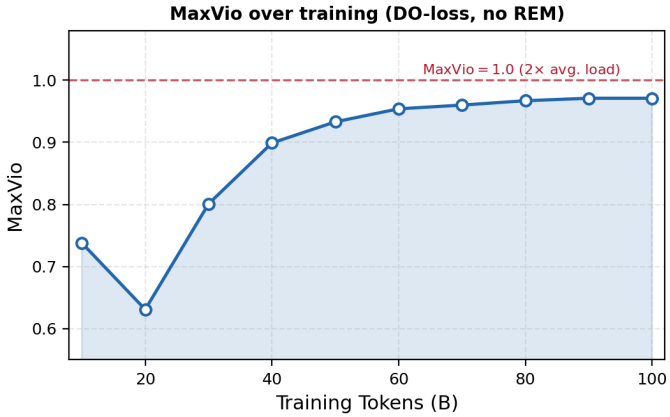


Figure 6: **Ablation of load imbalance without REM.** MaxVio over the course of 100B-token training for DO-loss alone (no replica scheduling). MaxVio stabilizes below 1.0, indicating that even without REM the busiest rank processes at most twice the average load, though it remains substantially higher than with replica scheduling enabled.

F Supplementary Experiments

This section collects auxiliary experiments and implementation details that support claims made in the main text.

F.1 Fine-grained λ Sweep

To pinpoint the optimal DO-loss coefficient at finer resolution, we sweep $\lambda \in \{1 \times 10^{-6}, 3 \times 10^{-6}, 5 \times 10^{-6}, 1 \times 10^{-5}, 3 \times 10^{-5}\}$ on a 2B/0.2B model trained for 20B tokens. Figure 7 shows that the optimum lies at $\lambda = 5 \times 10^{-6}$, roughly two orders of magnitude smaller than typical load-balancing loss coefficients, confirming the trend observed on the main 4.8B model.

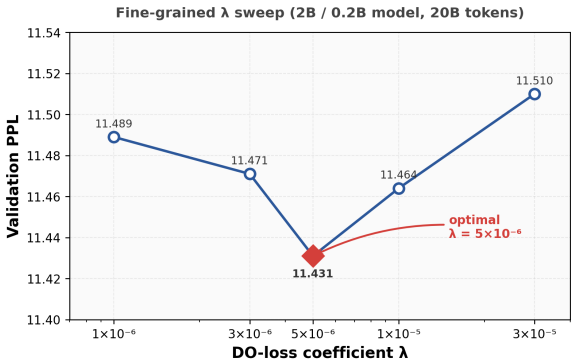


Figure 7: **Ablation of DO-loss coefficient λ at finer resolution.** Sweep on a 2B/0.2B model (20B tokens). The optimum λ=5×10^{-6} is roughly two orders of magnitude smaller than typical load-balancing loss coefficients.

F.2 DO-loss Batch Size Ablation

Table 4 ablates the effective batch size for computing the global distance matrix G_{ij} on the 4.8BA0.5B model, keeping the micro-batch size fixed at 4 and synchronizing across 8 GPUs. Validation perplexity improves monotonically as the statistical scope widens from a single-device micro-batch to the full global batch, but with diminishing marginal returns, consistent with the batch-size effects documented by Qiu et al. (2025) for load-balancing losses.

Configuration	Micro-batch	GPUs	grad_accum	Eff. Batch (G_{ij})	Val. PPL
DO-loss (local)	4	–	–	4 (no sync)	8.4518
DO-32	4	8	1	32	8.4475
DO-64	4	8	2	64	8.4458
DO-128	4	8	4	128	8.4446
Global-DO-loss	4	8	16	512	8.4433

Table 4: Effect of the effective batch size for G_{ij} statistics on validation perplexity (4.8BA0.5B, 100B tokens).

F.3 Expert Count Scaling

Figure 8 compares Global-DO-loss and Auxfree on a 2B/0.2B model as the number of experts increases from 32 to 128. At 32 experts, Auxfree holds a slight edge; however, as expert granularity increases, Global-DO-loss pulls ahead and the gap widens. This demonstrates

that the benefit of distributional orthogonalization grows with the number of experts—precisely the regime where expert collapse and representational redundancy become more severe. The result aligns with the trend toward fine-grained expert architectures in modern LLMs.

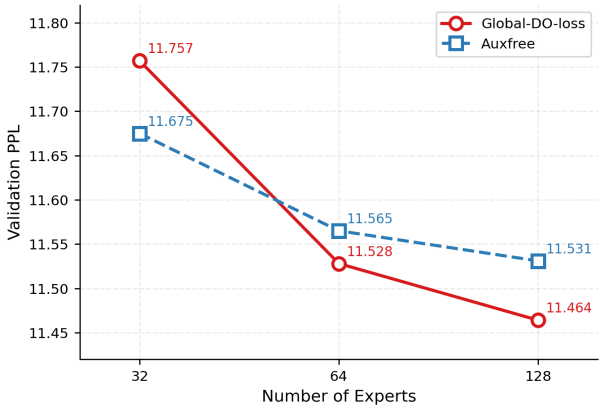


Figure 8: **Ablation of expert granularity.** Validation perplexity vs. expert count (2B/0.2B, 20B tokens). Global-DO-loss trails Auxfree at 32 experts but surpasses it as granularity increases to 64 and 128, demonstrating that DO-loss benefits more from finer expert specialization.

F.4 REM Memory and Throughput Overhead

Table 5 reports the steady-state throughput of each scheduling method on both H20 and H800 GPU clusters. All methods use two replica experts per rank on top of the DO-loss-trained 4.8BA0.5B model. REM consistently outperforms EfficientMoE and Pro-Prophet across all configurations and approaches the Auxfree baseline.

Throughput recovery. EP8 yields higher throughput than EP16 across all methods due to the absence of inter-node all-to-all communication. The DO-only baseline incurs a throughput penalty of roughly 10% relative to Auxfree under EP8, and a larger drop under EP16 due to exacerbated inter-node imbalance. REM recovers the vast majority of this gap across all configurations, bringing throughput to within a few percent of the Auxfree baseline. Under EP16, REM nearly matches Auxfree because replica experts inherently reduce costly inter-node communication by routing tokens to local replicas.

	H20		H800	
Method	DP2 EP8	DP1 EP16	DP2 EP8	DP1 EP16
Auxfree	81.2	70.1	161.3	101.8
DO-only	75.3	62.8	152.7	93.1
DO+REM	79.5	69.7	159.6	101.3
DO+EfficientMoE	77.8	67.5	155.2	95.7
DO+Prophet	76.8	65.9	152.3	92.2

Table 5: **Ablation of expert scheduling method.** Steady-state throughput (TFLOPs) averaged over 1,000 steps. All methods use two replica experts per rank on the DO-loss-trained 4.8BA0.5B model.

Scheduling comparison. EfficientMoE lags behind REM because it operates only at the global-batch level without real-time scheduling, leaving transient intra-step imbalances unresolved. Pro-Prophet introduces micro-batch-level replica replication whose communication overhead, under communication-bound conditions, can offset its scheduling benefits. In contrast, REM's two-tiered design—global replica placement plus lightweight micro-batch dispatching—avoids these pitfalls.

Memory overhead. Peak GPU memory increases modestly (single-digit percentage) with one or two replica experts, as replicas add only static parameter copies without extra activation memory. This overhead is negligible at the scales where MoE models are typically deployed.

Communication overhead. The AllGather for queue depths transmits only $|\mathcal{R}|$ integers per EP group, which is dwarfed by the AlltoAll dispatch that moves full hidden states.