

Dual-Bias Load Balancing: Mitigating Expert-Bias Staleness in Auxiliary-Loss-Free Mixture-of-Experts

Yuxin Tian^{1*}, Hailin Zhao^{1*}, Jia Liu¹, Zhiqiang Zhang^{1†},

¹Ling Team, Ant Group

cs.yuxintian@outlook.com, lingyao.zzq@antgroup.com

Abstract

Auxiliary-loss-free load balancing (ALF-LB) achieves promising performance in Mixture-of-Experts large language model (MoE-LLM) pre-training as it decouples load balancing from the main objective. However, we observe an unexplored phenomenon of it: routing imbalance is concentrated in shallow MoE layers. We attribute this phenomenon to the interaction among homogeneous shallow layer representations, limited router discrimination, and stale expert biases, which together make token assignments fragile and load corrections poorly aligned with the current routing state. To break the cycle, we propose **Dual-Bias Load Balancing (DLB)**, which augments the standard global expert bias with an intra-step micro-bias residual. DLB introduces the effective bias $\mathbf{b}_{\text{eff}} = \mathbf{b}_g + \Delta\mathbf{b}$, where $\mathbf{b}_g$ is updated once per gradient step and $\Delta\mathbf{b}$ is updated across micro-batches within the same step. DLB does not add forward or backward passes and integrates naturally into existing large-scale MoE training pipelines. Experiments on large-scale MoE pre-training show that DLB mitigates expert-bias staleness, alleviates shallow-layer imbalance, and yields lower test perplexity and stronger average downstream performance.

1 Introduction

Mixture-of-Experts (MoE) decouples parameter scale from per-token computational cost through sparse activation, establishing itself as a compelling paradigm for parameter scaling (Shazeer et al., 2017; Riquelme et al., 2021; Mustafa et al., 2022; DeepSeek-AI et al., 2025; Yang et al., 2025). A vital challenge for MoE pre-training is routing balancing: ensuring that tokens are distributed to the most suitable experts and avoiding both expert overload and expert starvation (Fedus et al., 2022; Zoph et al., 2022). Otherwise, unbalanced routing can

cause severe hardware under-use, unstable optimization dynamics, and degraded model performance.

Existing load-balancing methods for MoE models largely fall into two choices: auxiliary-loss-based methods (AL-LB) (Zoph et al., 2022; Qiu et al., 2025b) and auxiliary-loss-free methods (ALF-LB) (Wang et al., 2024; DeepSeek-AI et al., 2025). ALF-LB is especially attractive for large-scale pre-training because it adjusts routing scores through an independent expert bias term, avoiding direct gradient interference between the load-balancing objective and the language-modeling objective. However, this advantage still comes with a severe issue: routing imbalance is uneven across layers, with shallow MoE layers consistently the most imbalanced (Lo et al., 2025; Mouzouni, 2026; Manzoni, 2026). As a result, expert starvation concentrates in shallow layers, wasting computation, underutilizing model capacity, and destabilizing training.

We attribute this phenomenon in ALF-LB to the interaction of three coupled factors: representation homogeneity, router, and expert-bias staleness. In shallow layers, hidden states are highly similar (Riquelme et al., 2021; Hu et al., 2025; Wang et al., 2026) and router cannot discriminate each other, which reduces routing-logit margins and makes routing decisions highly sensitive to bias perturbations. Meanwhile, as shown in Fig. 2, the expert bias in ALF-LB is updated via primal-dual optimization (Han and Zhong, 2025). The bias used at a gradient step is computed from earlier routing statistics, whereas the routing scores are produced by the current model state, creating a temporal mismatch between the load-correction signal and the routing state to which it is applied. These factors form a vicious cycle: poor router and homogeneous shallow representations make routing fragile, while stale expert bias can amplify this fragility into persistent load imbalance. Since MoE

*These authors contribute equally.

†Correspondence author

router and representation homogeneity is intrinsic to training, we focus on expert-bias staleness as a more tractable intervention point.

To break the vicious cycle, we propose **Dual-Bias Load Balancing (DLB)**, which augments the standard global expert bias with a secondary micro-bias residual updated within each training step. Leveraging the micro-batch splitting already present in gradient accumulation and pipeline parallelism (Narayanan et al., 2019; Huang et al., 2019), DLB maintains two complementary terms: a global bias updated once per gradient step and a micro residual updated after each micro-batch forward pass. Since model weights remain fixed across micro-batches within the same gradient optimization step, the micro residual stays aligned with the current model state and provides a timely correction to stale expert bias. Furthermore, DLB does not add extra forward or backward passes and integrates naturally into existing large-scale MoE training pipelines. In summary, our contributions are as follows:

- We identify a phenomenon of ALF-LB, namely shallow-layer routing imbalance, and show that shallow layers are substantially more imbalanced than deeper ones, reducing computational efficiency, underutilizing model capacity, and destabilizing training.
- We show that shallow-layer routing imbalance is associated with three coupled factors, *i.e.*, representation homogeneity, router and expert-bias staleness, which can form a vicious cycle. We further demonstrate empirically that replacing the stale bias with the look-ahead bias induces substantial routing changes, especially in shallow layers.
- We propose DLB, a simple and efficient load-balancing method that compensates for stale expert bias with an online-updated micro-bias residual. DLB does not introduce extra forward or backward passes, integrates naturally with pipeline parallelism, and improves load balancing, training stability, and downstream performance in our experiments.

2 Related Work

Mixture of Experts (MoE) scales the LLM by sparsely activating only a subset parameters for each token, thereby separating total model parameters from per token computation. Its effective-

ness has been demonstrated by recent LLM families, including Mixtral $8\times 7B$ (Jiang et al., 2024), DeepSeek V3 and V4 (DeepSeek-AI et al., 2025; Xu et al., 2026a), the Ling series (Team et al., 2025b,a; Li et al., 2026), the Kimi K series (Team et al., 2026b,a), the GLM 5 series (Zeng et al., 2026), and Qwen3 MoE (Yang et al., 2025). However, MoE usually suffers from load balancing during the training, which not only causes inefficient training but also performance degradation. To this end, methods such as ALF-LB, global batch AL-LB, and similarity preserving routers improve the load balancing while preserving expert specialization (Wang et al., 2024; Qiu et al., 2025b; Omi et al., 2025). More recent approaches, including QB (Su, 2026) and BIP (Sun, 2025), formulate load balancing from an optimal transport perspective. Complementary architectural designs improve routing stability by incorporating routing decisions from preceding layers, constraining expert paths, or eliminating centralized routing (Qiu et al., 2025a; Gu et al., 2026; Liu et al., 2026). Despite this progress, growing evidence suggests that routing imbalance varies substantially across layers. To be specific, shallow layers tend to converge more slowly, develop weaker expert specialization, and exhibit greater sensitivity to token identity, language identity, and perturbations in hidden representations (Xue et al., 2024; Muennighoff et al., 2025; Bandarkar et al., 2026; Zheng et al., 2026). These observations motivate the use of layer specific expert budgets and initialization strategies (Lo et al., 2025; Kang et al., 2025; Yoon et al., 2026; Huang et al., 2024; Nakamura et al., 2025; Xu et al., 2026b).

Nevertheless, the mechanism that produces persistent imbalance in shallow layers under ALF-LB remains insufficiently understood. This study explore two factors behind this behavior and shows that expert bias staleness is one of a central mechanism. Based on this analysis, we propose DLB, which introduces an intra step micro bias residual to provide more timely corrections to routing decisions. DLB improves load balance without requiring additional forward or backward passes.

3 Methodology

3.1 Preliminary

The MoE-LLMs replace the dense FFN with a sparse MoE module consisting of a learned router and E parallel FFN experts (DeepSeek-AI et al.,

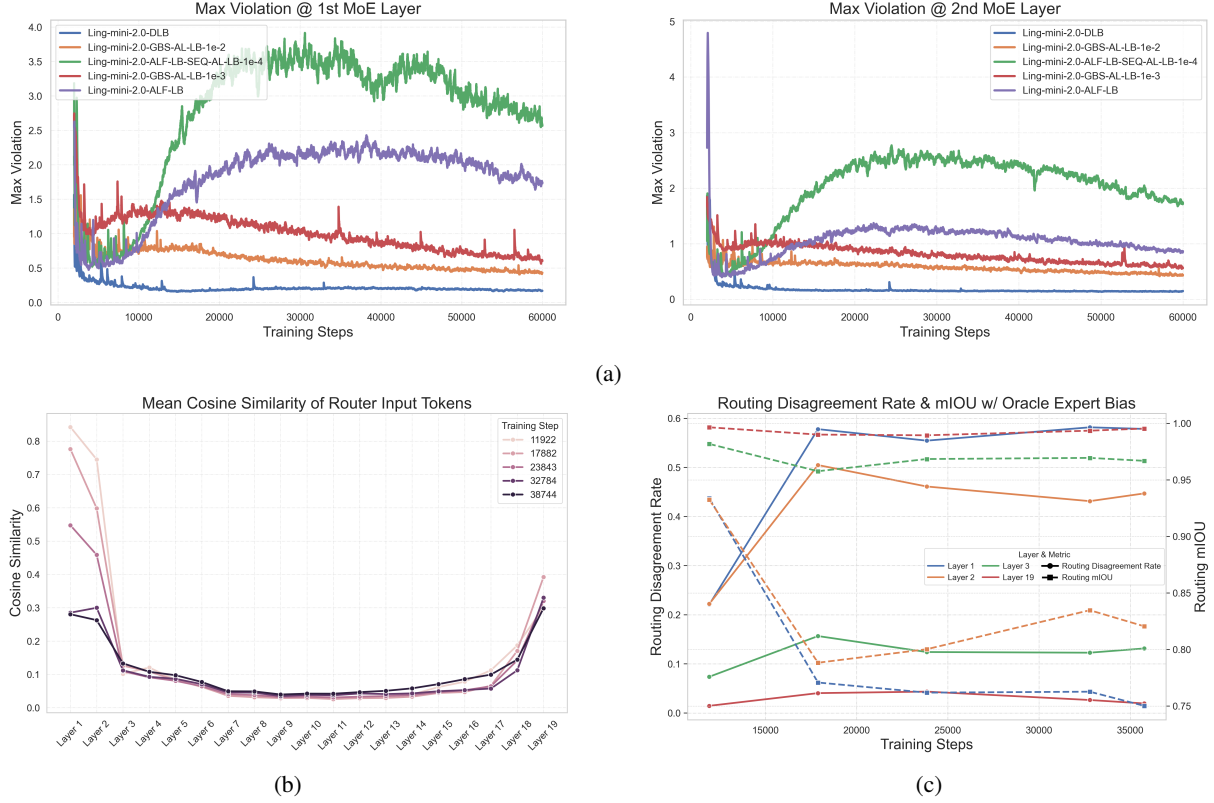


Figure 1: Shallow-layer routing imbalance in ALF-LB. (a) shows the per-layer MaxVio (Wang et al., 2024) of a 16B-A1.4B model (Team et al., 2025a), revealing that shallow MoE layers suffer disproportionately severe imbalance. We attribute this phenomenon to three compounding factors: representation homogeneity, router, and expert-bias staleness. (b) shows that router inputs in shallow layers exhibit high cosine similarity, especially early in training. (c) quantifies the impact of stale expert bias correction through two metrics: (1) the *routing disagreement rate*, the fraction of tokens whose top- k assignments change when the current bias is replaced by the look-ahead bias computed from the same step, and (2) *routing mIoU*, the mean intersection over union of top- k assignments under the current bias versus the look-ahead bias.

2025; Dai et al., 2024; Team et al., 2025c; Yang et al., 2025). Formally, given tokens $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^B$, the router produces gating scores $\mathbf{s}_i \in \mathbb{R}^E$ and selects a top- k subset of experts per token¹:

$$\begin{aligned} \mathbf{s}_i &= \delta(\mathbf{x}_i \mathbf{W}_{router}), \mathcal{A}_i = \text{TopK}(\mathbf{s}_i, k), \\ \mathbf{h}_i &= \mathbf{x}_i + \sum_{j \in \mathcal{A}_i} g_{ij} E_j(\mathbf{x}_i), \end{aligned} \quad (1)$$

where $\delta(\cdot)$ is the gating activation, $\mathbf{W}_{router} \in \mathbb{R}^{d \times E}$ is the router weight, $E_j(\cdot)$ is the j -th expert, and g_{ij} denotes the normalized weighting score

Above routing is prone to load imbalance and training instability. Compared with AL-LB, the ALF-LB avoids gradient interference between the balance and language-modeling objectives, yielding stronger performance in practice (Wang et al., 2024; Team et al., 2025c,a). Specifically, ALF-LB introduces a per-expert bias $\mathbf{b} \in \mathbb{R}^E$ to correct

the routing scores before top- k selection, while the original unbiased scores are used for expert output weighting:

$$\begin{aligned} \mathcal{A}_i(\mathbf{b}) &= \text{TopK}(\mathbf{s}_i + \mathbf{b}, k), \\ g_{ij}(\mathbf{b}) &= \frac{s_{ij}}{\sum_{r \in \mathcal{A}_i(\mathbf{b})} s_{ir}}, \quad j \in \mathcal{A}_i(\mathbf{b}), \\ \mathbf{h}_i &= \mathbf{x}_i + \sum_{j \in \mathcal{A}_i(\mathbf{b})} g_{ij}(\mathbf{b}) E_j(\mathbf{x}_i), \\ \mathbf{s}_i &= \text{sigmoid}(\mathbf{x}_i \mathbf{W}_{router}). \end{aligned} \quad (2)$$

The bias $\mathbf{b}$ is initialized to $\mathbf{0}$ and updated at the end of each gradient step via a sign-based rule:

$$\mathbf{b}^{t+1} = \mathbf{b}^t + \mu \cdot \text{sign}(\mathbf{e}^t), \quad \mathbf{e}^t = \bar{\mathbf{c}} - \mathbf{c}^t, \quad (3)$$

where μ is the update rate, $\mathbf{c}^t \in \mathbb{R}^E$ counts the tokens assigned to each expert at step t , $\bar{\mathbf{c}}$ repeats the average expert load across experts, and $\mathbf{e}^t$ is the per-expert load violation.

¹For simplicity, we omit the MoE layer index throughout.

3.2 Shallow Layer Routing Imbalance

Although ALF LB is effective, routing imbalance persists in shallow layers, as shown in Fig. 1(a). We attribute this behavior to the interaction between representation homogeneity, MoE router and expert bias staleness. Homogeneous representations make expert rankings fragile, while stale biases apply delayed corrections to an evolving router. Together, these effects create a feedback loop that sustains shallow layer imbalance.

As shown in Fig. 1(b), hidden states in shallow layers are highly similar, particularly early in training. Because the shallow layer router remains weakly discriminative, competing experts receive scores with narrow margins, making top k assignments sensitive to small perturbations. This instability, captured by routing mIoU and the disagreement rate, makes bias based correction unreliable. Hidden states in deeper layers may also be similar, but better trained routers distinguish them more effectively², resulting in more stable routing and balanced expert loads.

Expert bias staleness introduces a complementary temporal mismatch, as illustrated in Fig. 1. The current bias is computed from earlier load statistics but applied to an updated router state. To quantify this effect, we replace the current bias with the look ahead bias for the next optimization step and compare the resulting routes using routing mIoU and the disagreement ratio, as shown in Fig. 1(c). The substantial difference confirms that bias staleness materially affects routing. Once imbalance emerges, delayed corrections lag behind changing assignments, limiting the recovery of underused experts and reinforcing imbalance across training steps. Since representation homogeneity and MoE router are intrinsic to training, we target expert bias staleness as the more tractable intervention point.

We next formalize expert-bias staleness and the diagnostic used in Fig. 1(c). At gradient step t , let $\mathbf{W}^t$ be the current model weights and let $\mathbf{b}^t$ be the expert bias available before routing the current batch. The token-expert assignment matrix is

$$R_{ij}^t(\mathbf{b}^t) = \mathbb{1}[j \in \text{TopK}(\mathbf{s}_i(\mathbf{W}^t) + \mathbf{b}^t, k)] . \quad (4)$$

After all routing decisions in step t are observed,

²See the Appendix for the layer-wise diagnosis of router discriminative ability.

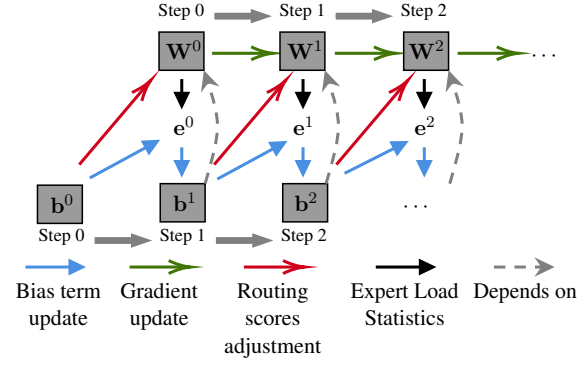


Figure 2: The illustration of the expert-bias staleness. The $\mathbf{W}^i$, $\mathbf{b}^i$, and $\mathbf{e}^i$ denote the weights, expert bias, and expert load violation at the i -th gradient step, respectively.

ALF-LB updates the bias as

$$\begin{aligned} \mathbf{b}^{t+1} &= \mathbf{b}^t + \mu \cdot \text{sign}(\mathbf{e}^t) , \\ \mathbf{e}^t &= \bar{\mathbf{c}} - \mathbf{c}^t(\mathbf{X}; \mathbf{W}^t, \mathbf{b}^t) . \end{aligned} \quad (5)$$

Thus, $\mathbf{b}^{t+1}$ is a look-ahead bias estimated from the current routing statistics, but it is only available after the current routing decisions have already been made. At the next gradient step, this bias is applied to a new router state after the model weights and input batch have changed. This delayed dual update is what we refer to as expert-bias staleness. In Fig. 1(c), we perform a sensitivity diagnostic by re-computing routing under the look-ahead bias $\mathbf{b}^{t+1}$ and comparing it with routing under $\mathbf{b}^t$ using the routing disagreement ratio and routing mIoU. This diagnostic should not be read as a complete causal decomposition of weight changes, batch variation, and bias-update frequency. Rather, it shows that stale bias correction is large enough to substantially alter routing decisions, especially in shallow MoE layers, where routing disagreement reaches nearly 60% in our analysis.

3.3 Dual-Bias Load Balancing

The analysis above suggests that the key limitation of ALF-LB is the temporal mismatch between stale expert bias and the current routing state. We therefore propose DLB, which augments the standard expert bias with an online-updated micro-bias residual. Specifically, DLB splits a global batch $\mathbf{X}$ into multiple micro-batches $\mathbf{X}_m$ and maintains two complementary terms: a global bias $\mathbf{b}_g$, updated once per gradient step, and a micro residual $\Delta \mathbf{b}_m$, updated after each micro-batch forward pass. The effective bias used for routing micro-batch m is

$$\mathbf{b}_{\text{eff}}^{t,m} = \mathbf{b}_g^t + \Delta \mathbf{b}_m^t , \quad \Delta \mathbf{b}_0^t = \mathbf{0} . \quad (6)$$

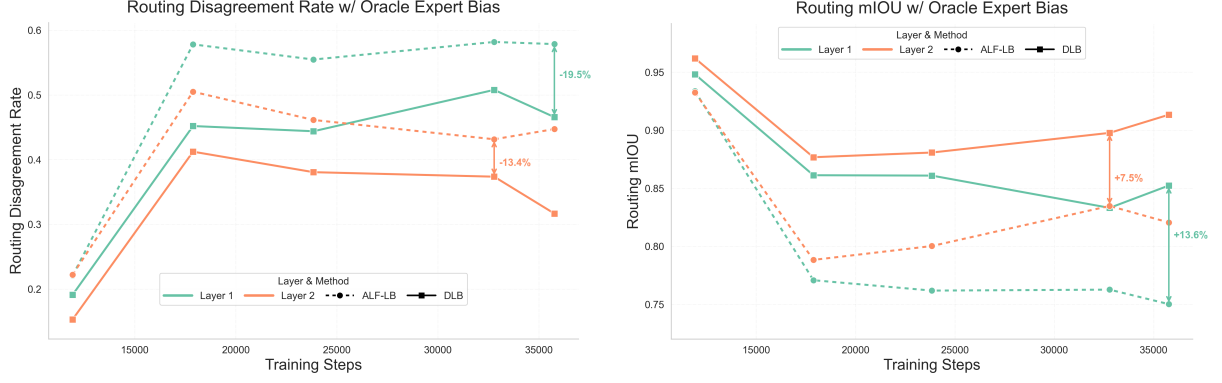


Figure 3: DLB reduces sensitivity to delayed bias correction. Following the diagnostic in Fig. 1(c), we replace the current bias with the look-ahead bias and compare the resulting routing decisions using routing disagreement ratio and routing mIoU.

With this effective bias, the MoE module in Eq. 2 becomes:

$$\begin{aligned} \mathcal{A}_i^{t,m} &= \text{TopK} \left(\mathbf{s}_i + \mathbf{b}_{\text{eff}}^{t,m}, k \right), \\ g_{ij}^{t,m} &= \frac{s_{ij}}{\sum_{r \in \mathcal{A}_i^{t,m}} s_{ir}}, \quad j \in \mathcal{A}_i^{t,m}, \\ \mathbf{h}_i^{t,m} &= \mathbf{x}_i + \sum_{j \in \mathcal{A}_i^{t,m}} g_{ij}^{t,m} E_j(\mathbf{x}_i). \end{aligned} \quad (7)$$

The global term captures long-term load correction across gradient steps, while the micro residual provides a short-term correction within the current step. As shown in Algorithm 1, model weights remain fixed while $\Delta \mathbf{b}_m$ is updated across micro-batches. Therefore, unlike the delayed global bias, the micro residual is estimated and applied under the same model state, allowing Eq. 7 to reduce the temporal mismatch in routing decisions. From a primal-dual optimization perspective (Han and Zhong, 2025), the model weights $\mathbf{W}$ and expert bias $\mathbf{b}$ correspond to the primal and dual variables, respectively. Standard ALF-LB performs only one delayed dual update per gradient step, whereas DLB adds several inner dual updates through $\Delta \mathbf{b}_m$ before the next weight update. This design makes the bias correction more responsive while preserving the original global bias update.

DLB also aligns naturally with existing large-scale training systems. Micro-batch splitting is already used in gradient accumulation and pipeline parallelism (Narayanan et al., 2019; Huang et al., 2019), so DLB only reuses the routing statistics collected at this granularity and does not require additional forward or backward passes. Depending on the synchronization scope used for micro statistics, DLB may introduce different communication

overheads; we discuss this trade-off in Section 4.

4 Experiments

We evaluate the proposed DLB from two complementary perspectives. First, we compare DLB against four representative load-balancing methods for MoE-LLM pre-training: global-batch-size auxiliary-loss load balancing (GBS-AL-LB) (Qiu et al., 2025b; Yang et al., 2025), ALF-LB (Wang et al., 2024), and a hybrid strategy that combines auxiliary-loss-free load balancing with sequence-level auxiliary-loss load balancing (ALF-LB + SEQ-AL-LB) (DeepSeek-AI et al., 2025). Notably, the load balancing constraints are applied to all the MoE layers. Second, we perform ablation studies to disentangle the effects of the balancing batch size and the update rate of the micro-bias residual. The appendix provides additional evaluations under alternative LLM settings, including a hybrid linear-attention MoE model and the Muon (Liu et al., 2025) optimizer and more ablations.

Experimental Setups. Unless otherwise specified, the main experiments and ablation studies are conducted in a large-scale pre-training setting using an MoE model with 16B total parameters and 1.4B activated parameters, following Ling (Team et al., 2025a). Concretely, the model adopts a decoder-only architecture with 20 transformer layers and 1 MTP layer (DeepSeek-AI et al., 2025). For self-attention, we use grouped-query attention (GQA) with 16 query heads and 8 key-value heads. To mitigate routing collapse in shallow layers, the feed-forward network (FFN) in the first Transformer layer is kept dense, whereas all subsequent layers are instantiated as MoE layers. Each MoE layer em-

16B-A1.4B 500B Tokens								
LB Method	G-LB-BSZ (λ/μ)	M-LB-BSZ (λ/ν)	Exp. ID	Com. ID	Smoothed Loss Diff. ↓	$\Delta \downarrow$ @ 1st MoE Layer	$\Delta \downarrow$ @ Global	Final Test PPL
GBS-AL-LB	$2^{11}(\lambda=1e-2)$	-	1	5	<u>+0.0035</u>	<u>+223.42%</u>	<u>+47.15%</u>	3.635
GBS-AL-LB	$2^{11}(\lambda=1e-3)$	-	2	5	+0.0037	+430.80%	+339.96%	3.638
ALF-LB	$2^{11}(\mu=1e-3)$	-	3	5	+0.0059	+860.76%	+91.93%	<u>3.632</u>
ALF-LB	$2^{11}(\mu=1e-3)$	$2^0(\lambda=1e-4)$	4	5	+0.0055	+1381.10%	+180.35%	3.637
+SEQ-AL-LB								
DLB	$2^{11}(\mu=1e-3)$	$2^2(\nu=1e-3)$	5	-	0.0000	0.00%	0.00%	3.619

Table 1: Training dynamics of different load-balancing methods. Here, “G/M-LB-BSZ”, “ λ ”, “ μ/ν ”, “Exp. ID”, “Com. ID”, and “ Δ ” denote the global/micro load-balancing batch size, the loss coefficient for AL-LB, the update rate for global/micro expert bias, the experimental ID, the compared experimental ID, and the smoothed relative difference ratio in MaxVio, respectively. 1st MoE Layer refers to the first MoE block, *i.e.*, the second Transformer block when the first Transformer FFN is dense.

16B-A1.4B 500B Tokens						
Exp. ID	Avg.	Knowledge			Math	
		AGIEval	MMLU	CMMLU	MATH	GSM8K
1	52.64	37.23	55.49	52.55	34.58	63.91
2	<u>53.11</u>	35.40	<u>55.72</u>	52.96	<u>38.10</u>	63.08
3	52.10	<u>36.41</u>	55.53	53.03	35.96	59.51
4	51.04	34.75	55.43	<u>53.38</u>	31.36	54.44
5	53.56	36.35	56.47	53.92	38.60	<u>63.31</u>

Exp. ID	Code		Reasoning		Understanding	
	MBPP	CruxEval	PIQA	BBH	HellaSwag	SQuAD2.0
1	44.80	35.81	76.28	45.52	66.00	66.88
2	48.00	<u>38.31</u>	77.15	<u>45.53</u>	65.59	64.33
3	45.80	36.25	77.31	44.94	65.98	62.39
4	<u>46.60</u>	34.31	78.02	43.03	<u>66.26</u>	63.91
5	44.60	38.38	<u>77.64</u>	48.98	66.58	<u>64.38</u>

Table 2: Performance on different downstream tasks. The “Exp. ID” refers to Table 1. The best and second-best scores are **boldfaced** and *italic underlined*, respectively.

employs token-choice routing and contains 256 fine-grained routed experts (Dai et al., 2024); for each token, 8 routed experts and 1 shared expert are activated. Across all experiments, models are trained from scratch with 2000 linear warmup steps on high-quality corpora covering multilingual, STEM, reasoning, and general-knowledge content, under the 300B-token or 500B-token settings. Unless noted otherwise, all methods share the same hyperparameters and parallelization setup. We use AdamW with a learning rate of 4.78e-4, and set the batch size and sequence length to 2048 and 4096, respectively. In addition, all compared methods employ router z-loss (Zoph et al., 2022) with a loss weight of 3.5e-6 to stabilize the magnitude of the MoE gating logits and apply zero-mean normalization to the bias updates (Su, 2025).

Evaluation. We mainly evaluate the proposed DLB from two perspectives: training dynamics

and zero/few-shot downstream performance. For training dynamics, we use the smoothed training-loss difference and the smoothed relative difference ratio of MaxVio (Wang et al., 2024) at first MoE Layer and across all MoE layers (*i.e.*, Δ @ 1st MoE Layer/Global) to assess the effectiveness over the long-term pre-training period. Given expert loads $\mathbf{c} \in \mathbb{R}^E$ and their average $\bar{c} = \frac{1}{E} \sum_{j=1}^E c_j$, the MaxVio is defined as

$$\text{MaxVio}(\mathbf{c}) = \max_j \left| \frac{c_j}{\bar{c}} - 1 \right|. \quad (8)$$

The smoothed relative difference ratio Δ compares each baseline’s smoothed MaxVio against DLB under the same setting. We also report perplexity on the held-out test set. For downstream evaluation, we consider 11 widely used benchmarks: AGIEval (Zhong et al., 2024), MMLU (Hendrycks et al., 2021a), and CMMLU (Li et al., 2024) for general knowledge; MATH (Hendrycks et al.,

Algorithm 1 Dual-Bias Load Balancing

Input: MoE model $\mathbf{W}$, global/micro update rates

μ/ν , # of experts E

- 1: Initialize global bias $\mathbf{b}_g^0 \leftarrow \mathbf{0}$
- 2: **for** gradient step $t = 0$ to $T - 1$ **do**
- 3: Get current global batch $\mathbf{X}$
- 4: Initialize global counter $\mathbf{c}_g^t \leftarrow \mathbf{0}$ and micro residual $\Delta \mathbf{b}_0^t \leftarrow \mathbf{0}$
- 5: **for** micro-batch index $m = 0$ to $M - 1$ **do**
- 6: Set effective bias $\mathbf{b}_{\text{eff}}^{t,m} \leftarrow \mathbf{b}_g^t + \Delta \mathbf{b}_m^t$
- 7: Route $\mathbf{X}_m$ with $\mathbf{b}_{\text{eff}}^{t,m}$ using Eq. 7 and obtain assignment map $\mathbf{R}^{t,m} \in \{0, 1\}^{B_m \times E}$
- 8: Count expert load $\mathbf{c}_m^t \leftarrow \mathbf{1} \cdot \mathbf{R}^{t,m}$
- 9: Synchronize micro-batch counter $\mathbf{c}_m^t$ (optional)
- 10: Compute average micro load $\bar{\mathbf{c}}_m \leftarrow \mathbf{1} \cdot \frac{1}{E} \sum_j \mathbf{c}_{m,j}^t$
- 11: Compute micro violation $\mathbf{e}_m^t \leftarrow \bar{\mathbf{c}}_m - \mathbf{c}_m^t$
- 12: Update micro residual $\Delta \mathbf{b}_{m+1}^t \leftarrow \Delta \mathbf{b}_m^t + \nu \cdot \text{sign}(\mathbf{e}_m^t)$
- 13: Accumulate global counter $\mathbf{c}_g^t \leftarrow \mathbf{c}_g^t + \mathbf{c}_m^t$
- 14: **end for**
- 15: Synchronize global counter $\mathbf{c}_g^t$
- 16: Compute average global load $\bar{\mathbf{c}}_g \leftarrow \mathbf{1} \cdot \frac{1}{E} \sum_j \mathbf{c}_{g,j}^t$
- 17: Compute global violation $\mathbf{e}_g^t \leftarrow \bar{\mathbf{c}}_g - \mathbf{c}_g^t$
- 18: Update global bias $\mathbf{b}_g^{t+1} \leftarrow \mathbf{b}_g^t + \mu \cdot \text{sign}(\mathbf{e}_g^t)$
- 19: **end for**

Output: MoE model $\mathbf{W}^T$ with global bias $\mathbf{b}_g^T$

2021b) and GSM8K (Cobbe et al., 2021) for mathematical capability; MBPP (Austin et al., 2021) and CruxEval (Gu et al., 2024) for code generation; PIQA (Bisk et al., 2020), BBH (Suzgun et al., 2023), and HellaSwag (Zellers et al., 2019) for reasoning; and SQuAD2.0 (Rajpurkar et al., 2018) for language understanding.

4.1 Main Results

The main results are summarized in Fig. 1(a), Fig. 3, Table 1, and Table 2. Taken together, DLB reduces sensitivity to expert-bias staleness, substantially improves shallow-layer load balancing, stabilizes pre-training, and leads to better average downstream performance in our runs.

DLB reduces sensitivity to expert-bias staleness.

We first test whether DLB addresses the mechanism

that motivates our method, namely the sensitivity of routing decisions to expert-bias staleness. Following the protocol in Fig. 1(c), we replace the current expert bias with the look-ahead bias and quantify the resulting routing changes using the routing disagreement ratio and routing mIoU. As shown in Fig. 3, DLB consistently achieves a lower routing disagreement ratio and a higher routing mIoU than ALF-LB. This result indicates that DLB produces routing decisions that are less affected by the look-ahead bias, suggesting that the proposed micro-bias residual reduces the temporal mismatch caused by stale expert bias.

DLB alleviates shallow-layer routing imbalance.

Fig. 1(a) and Table 1 reveal two key findings. First, unlike ALF-LB, AL-LB does not exhibit pronounced shallow-layer routing imbalance; under ALF-LB, experts in first MoE Layer may bear nearly twice the average load. Second, DLB not only markedly alleviates shallow-layer routing imbalance, but also yields smoother load-balancing dynamics with fewer spikes than the competing methods. Compared with DLB, baseline methods exhibit 223%–1381% higher smoothed MaxVio at first MoE Layer. In addition, the balanced performance of DLB is not attributed to reducing the balancing batch size from global-batch scope to a micro-batch scope, shown in Exp.4 in Table 1. These findings are consistent with our hypothesis that mitigating expert-bias staleness is especially beneficial in shallow layers, by breaking the vicious cycle between representation homogeneity and expert-bias staleness.

DLB improves both training and downstream performance.

Table 1 shows that DLB achieves the best training loss and the lowest test perplexity, indicating that improved load balancing translates into better optimization quality rather than merely more uniform expert utilization. Table 2 further reports zero/few-shot performance on 11 downstream benchmarks, where DLB achieves the best average score among the compared load-balancing methods. The gains are concentrated on MMLU, CMMLU, MATH, CruxEval, PIQA, BBH, and HellaSwag, while results are mixed or comparable on AGIEval, GSM8K, MBPP, and SQuAD2.0. This pattern is consistent with the hypothesis that improved shallow-layer load balancing can improve optimization quality and may contribute to better downstream performance.

16B-A1.4B 300B Tokens G-LB-BSZ = 2^{11} M-LB-BSZ = 2^2 $\mu = 1e-3$						
Setup	ν	Avg.	Knowledge			Math
			AGIEval	MMLU	CMMLU	MATH
DLB GBS-MBS	1e-4	50.45	33.00	53.36	50.89	35.02
DLB GBS-MBS	5e-4	50.59	32.16	52.61	50.51	34.26
DLB GBS-MBS	1e-3	51.48	32.98	53.40	50.92	35.02
Math		Code		Reasoning		Understanding
GSM8K	MBPP	CruxEval	PIQA	BBH	HellaSwag	SQuAD2.0
60.27	42.20	34.00	76.77	42.65	64.84	61.98
61.79	40.00	36.88	76.99	44.91	64.78	61.61
62.77	43.20	33.88	76.44	47.33	64.51	65.86

Table 3: Ablation studies on the update rate ν of the micro-bias residual $\Delta\mathbf{b}$. The best scores are **boldfaced**.

16B-A1.4B 300B Tokens G-LB-BSZ = 2^{11} $\mu = \nu = 1e-3$						
Setup	M-LB-BSZ	Avg.	Knowledge			Math
			AGIEval	MMLU	CMMLU	MATH
DLB GBS-DP	2^{11}	50.03	33.04	53.48	51.32	34.74
DLB GBS-EP	2^5	50.53	33.77	52.68	49.59	34.06
DLB GBS-MBS	2^2	51.48	32.98	53.40	50.92	35.02
Math		Code		Reasoning		Understanding
GSM8K	MBPP	CruxEval	PIQA	BBH	HellaSwag	SQuAD2.0
58.45	40.40	34.19	76.77	39.25	64.97	63.71
59.82	43.20	34.94	77.04	42.82	64.33	63.54
62.77	43.20	33.88	76.44	47.33	64.51	65.86

Table 4: Ablation studies on the balancing batch size of the micro-bias residual $\Delta\mathbf{b}$. The best scores are **boldfaced**.

Method	Iteration time	Relative Delay
ALF-LB	2.476 s	-
DLB GBS-MBS	2.521 s	+1.8%
DLB GBS-EP	2.507 s	+1.3%
DLB GBS-DP	3.312 s	+33.8%

Table 5: Computation cost of different balancing scope.

4.2 Ablation Study

In this section, we investigate how the update rate of the micro-bias residual and its balancing batch size affect downstream performance. The ablation results are reported in Table 3 and Table 4. Overall, they suggest that DLB is reasonably robust to moderate hyperparameter changes, while still benefiting from a sufficiently frequent and sufficiently local micro-bias update.

Effect of the micro-bias update rate. Table 3 studies the update rate ν of the micro-bias residual while fixing the global setup. A larger update rate generally yields better overall performance, with $1e-3$ achieving the best average score. The gains are consistent but moderate across $1e-4$ to $1e-3$,

suggesting that DLB benefits from more responsive updates without requiring sensitive tuning.

Effect of the micro balancing batch size. Table 4 examines different balancing batch sizes for $\Delta\mathbf{b}$. The global bias $\mathbf{b}_g$ is always updated from routing statistics accumulated over the full global batch (GBS). By contrast, $\Delta\mathbf{b}$ is updated from a smaller scope within each step: 1) GBS-MBS computes the update from the routing statistics of a micro-batch within each data-parallel (DP) rank, 2) GBS-EP computes it from the routing statistics of a micro-batch step across all expert-parallel (EP) ranks, and 3) GBS-DP computes it from the routing statistics of a micro-batch step across all DP ranks.

Among these settings, GBS-MBS achieves the best average performance. This suggests that a small balancing scope can be enough to mitigate shallow layer imbalance while avoiding the communication required for cross rank synchronization. On other hand, expanding the scope also introduces system costs. Larger micro batches consume more GPU memory, while broader synchronization increases communication overhead and iteration time,

as shown in Table 5. Thus, GBS-MBS is a good trade-off choice.

5 Conclusion

We studied a previously underexplored limitation of ALF-LB: shallow-layer routing imbalance. We attributed this behavior to the interaction between representation homogeneity and expert-bias staleness, which together make shallow-layer routing especially fragile. To address this issue, we proposed DLB, which complements the standard global expert bias with a micro-bias residual updated within each gradient step. Experiments show that DLB reduces sensitivity to delayed bias correction, improves shallow-layer load balancing, stabilizes training, and yields better perplexity and average downstream performance, without adding forward or backward passes. This suggests that timely bias updates are an effective way to improve large-scale MoE training.

Limitations

Our empirical conclusions are based on a finite set of pre-training runs and downstream evaluations, and some benchmark results may be affected by potential random factors. Although the overall trends are consistent across the metrics we report, individual benchmark scores should still be interpreted with appropriate caution. In addition, MoE routing balance can also be influenced by other data, architectural or optimization choices, such as training data orchestration, expert configuration, parallelism strategy, and optimizer dynamics. When a model already achieves highly balanced routing under such settings, the marginal benefit of DLB may become less pronounced. Finally, DLB does not add forward or backward passes, but system overhead can depend on how micro-batch routing statistics are synchronized across DP and EP ranks.

References

- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. [Program Synthesis with Large Language Models](#). *Preprint*, arXiv:2108.07732.
- Lucas Bandarkar, Chenyuan Yang, Mohsen Fayyaz, Junlin Hu, and Nanyun Peng. 2026. [Multilingual Routing in Mixture-of-Experts](#). *Preprint*, arXiv:2510.04694.
- Yonatan Bisk, Rowan Zellers, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7432–7439.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training Verifiers to Solve Math Word Problems](#). *Preprint*, arXiv:2110.14168.
- Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 2024. [DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models](#). *Preprint*, arXiv:2401.06066.
- DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. [DeepSeek-V3 Technical Report](#). *Preprint*, arXiv:2412.19437.
- William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.
- Alex Gu, Baptiste Rozière, Hugh Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida I. Wang. 2024. [CRUXEval: A Benchmark for Code Reasoning, Understanding and Execution](#). *Preprint*, arXiv:2401.03065.
- Zijin Gu, Tatiana Likhomanenko, Vimal Thilak, Jason Ramapuram, and Navdeep Jaitly. 2026. [Path-Constrained Mixture-of-Experts](#). *Preprint*, arXiv:2603.18297.
- X. Y. Han and Yuan Zhong. 2025. [A Theoretical Framework for Auxiliary-Loss-Free Load Balancing of Sparse Mixture-of-Experts in Large-Scale AI Models](#). *arXiv preprint arXiv:2512.03915*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. [Measuring Massive Multitask Language Understanding](#). *Preprint*, arXiv:2009.03300.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. [Measuring Mathematical Problem Solving With the MATH Dataset](#). *Preprint*, arXiv:2103.03874.
- Gang Hu, Yinglei Teng, Pengfei Wu, and Nan Wang. 2025. [FFT-MoE: Efficient Federated Fine-Tuning for Foundation Models via Large-scale Sparse MoE under Heterogeneous Edge](#). *Preprint*, arXiv:2508.18663.

- Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. 2024. [Harder Tasks Need More Experts: Dynamic Routing in MoE Models](#). *Preprint*, arXiv:2403.07652.
- Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le, and Yonghui Wu. 2019. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in neural information processing systems*, 32.
- Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L  lio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. [Mixtral of Experts](#). *Preprint*, arXiv:2401.04088.
- Hao Kang, Zichun Yu, and Chenyan Xiong. 2025. [FLAME-MoE: A Transparent End-to-End Research Platform for Mixture-of-Experts Language Models](#). *Preprint*, arXiv:2505.20225.
- Ang Li, Ben Liu, Bin Han, Bin Hu, Bin Jing, Binbin Hu, Bing Li, Cai Chen, Caizhi Tang, Changxin Tian, and 1 others. 2026. Ling and ring 2.6 technical report: Efficient and instant agentic intelligence at trillion-parameter scale. *arXiv preprint arXiv:2606.15079*.
- Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024. Cmm-lu: Measuring massive multitask language understanding in chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11260–11285.
- Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, and 9 others. 2025. [Muon is Scalable for LLM Training](#). *Preprint*, arXiv:2502.16982.
- Yilun Liu, Jinru Han, Sikuan Yan, Volker Tresp, and Yunpu Ma. 2026. [Routing-Free Mixture-of-Experts](#). *Preprint*, arXiv:2604.00801.
- Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. 2025. A closer look into mixture-of-experts in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4427–4447.
- Andrea Manzoni. 2026. [MoE-Sieve: Routing-Guided LoRA for Efficient MoE Fine-Tuning](#). *arXiv preprint arXiv:2603.24044*.
- Charafeddine Mouzouni. 2026. [Three Phases of Expert Routing: How Load Balance Evolves During Mixture-of-Experts Training](#). *arXiv preprint arXiv:2604.04230*.
- Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, and Nathan Lambert. 2025. Olmoe: Open mixture-of-experts language models. In *International Conference on Learning Representations*, volume 2025, pages 62061–62121.
- Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. 2022. Multi-modal contrastive learning with limoe: The language-image mixture of experts. *Advances in Neural Information Processing Systems*, 35:9564–9576.
- Taishi Nakamura, Takuya Akiba, Kazuki Fujii, Yusuke Oda, Rio Yokota, and Jun Suzuki. 2025. [Drop-Upcycling: Training Sparse Mixture of Experts with Partial Re-initialization](#). *Preprint*, arXiv:2502.19261.
- Deepak Narayanan, Aaron Harlap, Amar Phanishayee, Vivek Seshadri, Nikhil R. Devanur, Gregory R. Ganger, Phillip B. Gibbons, and Matei Zaharia. 2019. PipeDream: Generalized pipeline parallelism for DNN training. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, pages 1–15, Huntsville Ontario Canada. ACM.
- Nabil Omi, Siddhartha Sen, and Ali Farhadi. 2025. [Load Balancing Mixture of Experts with Similarity Preserving Routers](#). *Preprint*, arXiv:2506.14038.
- Zhen Qin, Dong Li, Weigao Sun, Weixuan Sun, Xuyang Shen, Xiaodong Han, Yunshen Wei, Baohong Lv, Xiao Luo, Yu Qiao, and Yiran Zhong. 2024a. [TransNormerLLM: A Faster and Better Large Language Model with Improved TransNormer](#). *Preprint*, arXiv:2307.14995.
- Zhen Qin, Weigao Sun, Dong Li, Xuyang Shen, Weixuan Sun, and Yiran Zhong. 2024b. [Lightning Attention-2: A Free Lunch for Handling Unlimited Sequence Lengths in Large Language Models](#). *Preprint*, arXiv:2401.04658.
- Zihan Qiu, Zeyu Huang, Shuang Cheng, Yizhi Zhou, Ivan Titov, and Jie Fu. 2025a. Layerwise recurrent router for mixture-of-experts. In *International Conference on Learning Representations*, volume 2025, pages 14179–14205.
- Zihan Qiu, Zeyu Huang, Bo Zheng, Kaiyue Wen, Zekun Wang, Rui Men, Ivan Titov, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025b. [Demons in the Detail: On Implementing Load Balancing Loss for Training Specialized Mixture-of-Expert Models](#). *Preprint*, arXiv:2501.11873.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don’t know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.
- Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, Andr   S  sano Pinto, Daniel Keysers, and Neil Houlsby. 2021.

- Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarczyk, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. [Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer](#). *Preprint*, arXiv:1701.06538.
- Jianlin Su. 2025. [Moe travelogue: 3. a different approach to allocation](#).
- Jianlin Su. 2026. [Moe travelogue: 6. optimal allocation promotes equilibrium](#).
- Yuan Sun. 2025. [Binary-Integer-Programming Based Algorithm for Expert Load Balancing in Mixture-of-Experts Models](#). *arXiv preprint arXiv:2502.15451*.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, and Denny Zhou. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.
- Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, and 1 others. 2026a. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*.
- Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, S. H. Cai, Yuan Cao, Y. Charles, H. S. Che, Cheng Chen, and Guanduo Chen. 2026b. [Kimi K2. 5: Visual Agentic Intelligence](#). *arXiv preprint arXiv:2602.02276*.
- Ling Team, Ang Li, Ben Liu, Binbin Hu, Bing Li, Bingwei Zeng, Borui Ye, Caizhi Tang, Changxin Tian, Chao Huang, Chao Zhang, Chen Qian, Chenchen Ju, Chenchen Li, Chengfu Tang, Chilin Fu, Chunshao Ren, Chunwei Wu, Cong Zhang, and 123 others. 2025a. [Every Activation Boosted: Scaling General Reasoner to 1 Trillion Open Language Foundation](#). *Preprint*, arXiv:2510.22115.
- Ling Team, Binwei Zeng, Chao Huang, Chao Zhang, Changxin Tian, Cong Chen, Dingnan Jin, Feng Yu, Feng Zhu, Feng Yuan, Fakang Wang, Gangshan Wang, Guangyao Zhai, Haitao Zhang, Huizhong Li, Jun Zhou, Jia Liu, Junpeng Fang, Junjie Ou, and 55 others. 2025b. [Every FLOP Counts: Scaling a 300B Mixture-of-Experts LING LLM without Premium GPUs](#). *Preprint*, arXiv:2503.05139.
- Meituan LongCat Team, Bayan, Bei Li, Bingye Lei, Bo Wang, Bolin Rong, Chao Wang, Chao Zhang, Chen Gao, Chen Zhang, Cheng Sun, Chengcheng Han, Chenguang Xi, Chi Zhang, Chong Peng, Chuan Qin, Chuyu Zhang, Cong Chen, Congkui Wang, and 163 others. 2025c. [LongCat-Flash Technical Report](#). *Preprint*, arXiv:2509.01322.
- Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. 2024. [Auxiliary-Loss-Free Load Balancing Strategy for Mixture-of-Experts](#). *Preprint*, arXiv:2408.15664.
- Xi Wang, Soufiane Hayou, and Eric Nalisnick. 2026. [The Myth of Expert Specialization in MoEs: Why Routing Reflects Geometry, Not Necessarily Domain Expertise](#). *arXiv preprint arXiv:2604.09780*.
- Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, and 1 others. 2026a. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*.
- Yuqi Xu, Rizhen Hu, Zihan Liu, Mou Sun, and Kun Yuan. 2026b. [Grouter: Decoupling Routing from Representation for Accelerated MoE Training](#). *Preprint*, arXiv:2603.06626.
- Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. 2024. [OpenMoE: An Early Effort on Open Mixture-of-Experts Language Models](#). *Preprint*, arXiv:2402.01739.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 Technical Report](#). *Preprint*, arXiv:2505.09388.
- Youngsik Yoon, Siwei Wang, Wei Chen, and Jungseul Ok. 2026. [When Are Experts Misrouted? Counterfactual Routing Analysis in Mixture-of-Experts Language Models](#). *Preprint*, arXiv:2605.07260.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.
- Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, and Chengxing Xie. 2026. [Glm-5: From vibe coding to agentic engineering](#). *arXiv preprint arXiv:2602.15763*.
- Yu Zhang, Zongyu Lin, Xingcheng Yao, Jiayi Hu, Fanqing Meng, Chengyin Liu, Xin Men, Songlin Yang, Zhiyuan Li, Wentao Li, Enzhe Lu, Weizhou Liu, Yanru Chen, Weixin Xu, Longhui Yu, Yejie Wang, Yu Fan, Longguang Zhong, Enming Yuan, and 40 others. 2025. [Kimi linear: An expressive, efficient attention architecture](#). *CoRR*, abs/2510.26692.
- Kening Zheng, Wei-Chieh Huang, Jiahao Huo, Zhonghao Li, Henry Peng Zou, Yibo Yan, Xin Zou, Jungang Li, Junzhuo Li, Hanrong Zhang, Xuming Hu, and Philip S. Yu. 2026. [Unveiling Language Routing Isolation in Multilingual MoE Models](#)

for Interpretable Subnetwork Adaptation. *Preprint*, arXiv:2604.03592.

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

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. 2022. *ST-MoE: Designing Stable and Transferable Sparse Expert Models*. *Preprint*, arXiv:2202.08906.

A Appendix

Layer	Averaged routing probs std
1	0.0018103
2	0.0030487
3	0.0044945
17	0.017193
18	0.018439
19	0.019926

Table 6: Average standard deviation of token-level routing probabilities (ALF-LB).

Routing diversity diagnosis. The Table 6 shows the std diagnosis of the routing probabilities based on the Exp.3 and 5 in the Table 1 denotes that the diversity of the routing logits is positively related to the layer depth. Thus the MoE router at deep layers is able to discriminate the hidden states.

Implementation details. Unless otherwise specified, all experiments use $EP = 8$, $TP = 1$, $PP = 1$, a micro batch size of 4, a global batch size of 2048, and two gradient accumulation steps.

Single-bias mirco-updated Ablation. Here we conduct a natural ablation, where the original ALF-LB expert bias is directly updated after each micro-batch without decomposing it into $\mathbf{b}_{\text{eff}} = \mathbf{b}_g + \Delta\mathbf{b}$. From Table 7, one could find such a operation not only decrease the global balance batch size and is harmful for the downstream performance due the worse expert specialization.

DLB with alternative architectural and optimization choices. We further evaluate whether DLB remains effective under alternative architectural and optimization choices. Specifically, we consider a hybrid linear-attention MoE model and

compare results under both AdamW and Muon (Liu et al., 2025) optimizers. Unless otherwise specified, most experimental configurations are the same as the main experiments and ablation studies. The model has 8.6B total parameters and 0.9B activated parameters, with 25 transformer layers, 1 MTP layer and hidden size 1664. It applies full softmax attention every four layers and uses Lightning Linear Attention (Qin et al., 2024a,b) in the remaining layers. Each MoE layer contains 128 routed experts and 1 shared expert, with 2 routed experts activated per token. For Muon experiments, we follow the optimization strategy of Liu et al. (2025).

Table 8 reports the downstream results. Under AdamW, DLB slightly improves the average score from 47.67 to 47.77, indicating that it remains competitive on the hybrid linear-attention architecture. Under Muon, DLB also slightly improves the average score from 49.24 to 50.70, with improvements on AGIEval, MMLU, MATH, GSM8K, MBPP, CruxEval, and BBH. These results suggest that DLB is not tied to a specific attention architecture or optimizer.

Other MoE setup. Following above hybrid-linear model setup, we include additional results that independently vary the number of routed experts and the top-(k) routing configuration, as shown in Table 9. Together, these evaluations demonstrate that DLB remains effective across various MoE architectures and training settings.

Layer-wise DLB. DLB is applied to all MoE layers in the main experiments. Here, we conduct a shallow-only/deep-only/all-layer ablation to offer a further investigation. To be specific, the tested model has 88.8B total parameters and 3.2B activated parameters, with 72 transformer layers, 1 MTP layer and hidden size 1536. It applies full softmax attention after every three linear layers and uses KDA (Zhang et al., 2025) in the remaining layers, like Ling-3.0-tiny³. Each MoE layer contains 128 routed experts and 1 shared expert, with 8 routed experts activated per token. And we adopt Muon as optimizer with learning rate 5.08e-4 and global batch size 2048. From Table 10, one could find that enabling DLB only in first 4 MoE layers remain its effectiveness. Notably, the training corpus of the above experiments have been changed due the computing cluster relocating and these results can not be comparable against other tables.

³<https://huggingface.co/inclusionAI/Ling-3.0-tiny-base>

16B-A1.4B 150B Tokens G-LB-BSZ = 2^{11} M-LB-BSZ = 2^2 $\mu = 1e-3$						
Method	Avg.	Knowledge			Math	
		AGIEval	MMLU	CMMLU	MATH	
DLB MBS	39.68	27.48	44.03	41.25	17.42	
DLB GBS-MBS	46.09	31.17	48.44	45.46	30.60	
Math		Code		Reasoning		Understanding
GSM8K	MBPP	CruxEval	PIQA	BBH	HellaSwag	SQuAD2.0
36.92	29.60	20.94	76.28	31.86	60.82	49.91
54.06	37.00	23.94	76.33	36.25	62.11	61.65

Table 7: Ablation of single-bias mirco updating. The best scores are **boldfaced**.

Hybrid-Linear 8.6B-A0.9B 500B Tokens G-LB-BSZ = 2^{11} M-LB-BSZ = 2^2 $\mu = \nu = 1e-3$						
Setup	Optimizer	Avg.	Knowledge			Math
			AGIEval	MMLU	CMMLU	MATH
ALF-LB	AdamW	47.67	31.80	51.92	47.83	34.38
DLB	AdamW	47.77	33.18	52.56	49.62	30.04
ALF-LB	Muon	49.24	33.53	54.94	52.88	33.02
DLB	Muon	50.70	35.29	55.05	52.14	35.44
Math		Code		Reasoning		Understanding
GSM8K	MBPP	CruxEval	PIQA	BBH	HellaSwag	SQuAD2.0
54.59	34.40	29.19	76.39	40.83	63.75	59.26
53.90	35.80	31.38	75.95	38.72	63.50	60.92
59.06	39.00	23.00	76.82	39.36	65.14	64.90
59.97	44.20	31.69	76.12	42.40	64.89	60.50

Table 8: Performance of DLB with hybrid-linear attention under different optimizer configurations. The best scores are **boldfaced**.

Hybrid-Linear 300B Tokens G-LB-BSZ = 2^{11} M-LB-BSZ = 2^2 $\mu = \nu = 1e-3$						
Method	MoE Setup	Avg.	Knowledge			Math
			AGIEval	MMLU	CMMLU	MATH
ALF-LB	E128A2	47.98	32.73	52.32	49.42	36.88
DLB	E128A2	48.18	34.87	52.73	49.87	40.92
ALF-LB	E128A8	51.86	34.18	55.75	52.76	38.06
DLB	E128A8	52.19	35.93	55.93	52.88	37.14
ALF-LB	E256A8	54.64	37.38	57.71	55.38	39.38
DLB	E256A8	54.86	36.65	57.28	56.10	41.20
Math		Code		Reasoning		Understanding
GSM8K	MBPP	CruxEval	PIQA	BBH	HellaSwag	SQuAD2.0
58.30	39.00	18.38	77.04	38.74	63.15	61.84
57.01	37.80	19.81	77.64	39.98	63.72	55.63
62.32	41.40	30.88	76.77	46.61	65.38	66.37
62.02	43.20	30.50	77.69	43.95	66.14	68.69
66.41	46.00	37.06	77.75	45.48	67.97	70.52
67.48	46.60	34.31	78.13	47.68	67.95	70.04

Table 9: Performance comparison of ALF-LB and DLB under different MoE configurations. The best scores within each MoE configuration are **boldfaced**.

Hybrid-KDA 88.8B-A3.2B 100B Tokens G-LB-BSZ = 2^{11} M-LB-BSZ = 2^2 $\mu = \nu = 1e-3$						
Method	Avg.	Knowledge			Math	
		AGIEval	MMLU	CMMLU	MATH	
ALF-LB	42.87	28.24	48.91	44.61	20.96	
DLB (First 4)	43.59	28.81	48.72	45.26	23.58	
DLB (All)	43.87	29.12	49.73	46.85	20.58	
Math	Code		Reasoning			Understanding
GSM8K	MBPP	CruxEval	PIQA	BBH	HellaSwag	SQuAD2.0
29.34	31.80	24.31	76.44	37.00	65.89	64.03
28.43	29.40	27.00	77.80	38.50	66.44	65.56
31.77	32.40	25.44	76.88	38.46	66.32	65.00

Table 10: Performance comparison of ALF-LB and layer-wise DLB configurations. The best scores are **boldfaced**.