Distributionally Robust Mixture-of-Experts Training

Xin Teng
New York University
Center for Data Science, NYU Shanghai
xt2251@nyu.edu

Muxiao Li
New York University
Center for Data Science, NYU Shanghai
ml9007@nyu.edu

Hongyi Wen
New York University
Center for Data Science, NYU Shanghai
hongyi.wen@nyu.edu

Abstract

Mixture-of-Experts (MoE) transformers scale capacity by activating only a few experts per token, but this sparsity creates a hidden reliability problem: when routing is imperfect, load-balanced models may send tokens to experts that are insufficiently trained for the assigned inputs. We propose *Distributionally Robust MoE Training* (DRMoET), a drop-in objective that treats layer-wise experts as endogenous robustness groups and optimizes high-loss routing outcomes rather than merely equalizing traffic. DRMoET updates a per-layer expert distribution by an entropy-regularized softmax rule on EMA-smoothed, activation-weighted expert losses, strengthening plausible non-top routing paths while preserving standard MoE computation. Under the FLAME-MoE recipe at 746M-total and 10.3B-total scales, DRMoET improves downstream averages over both standard FLAME-MoE and auxiliary-loss-free balancing. At 10.3B total parameters and 67B training tokens, DRMoET improves the seven-task average from 0.6625 to 0.6767, while the auxiliary-loss-free baseline achieves 0.6431. Mechanistic analyses show lower expert-loss variance with nearly unchanged mean loss, 4.3% lower excess loss under forced mid-k misrouting, and improved domain–expert specialization. These results position routing robustness–not only utilization balance–as a practical objective for reliable sparse MoE scaling. Project page and code are available at: https://drmoet.github.io/.

1 Introduction

Scaling has been a central driver of progress in deep generative language models, but naïvely increasing parameter count quickly becomes computationally prohibitive. Mixture-of-Experts (MoE) architectures offer an appealing alternative: they increase *total* model capacity while keeping *per-token* computation nearly constant by routing each token through only a small subset of expert subnetworks. This sparse activation principle underlies widely used MoE systems, including Switch Transformer [Fedus et al., 2022] and GShard [Lepikhin et al., 2021], and continues to be adopted in modern large-scale language models.

Despite their promise, MoE models introduce a hidden reliability problem: sparse routing exposes the model to the quality of whichever experts are selected. The router need not fail catastrophically for this to matter; imperfect gating, load-balancing pressure, or distribution shift can send tokens to plausible but non-optimal experts. Under standard training, these lower-ranked or less frequently favored experts receive weaker task-aligned updates and can remain noticeably worse than top experts. As a result, occasional misrouting causes disproportionate loss. The core issue is therefore not merely uneven utilization, but uneven expert competence under imperfect routing.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

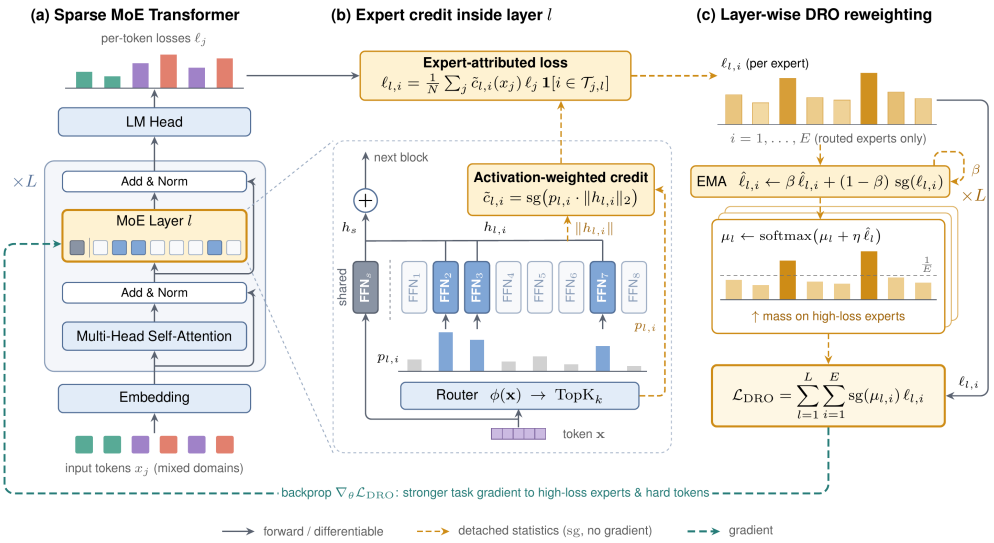


Figure 1: Distributionally Robust MoE Training workflow. DRMoET treats experts as layer-wise robustness groups and adaptively upweights high-loss routing outcomes, strengthening plausible non-optimal paths rather than only encouraging uniform utilization.

A common mitigation is to add auxiliary load-balancing terms that penalize skewed routing and encourage more even expert utilization [Fedus et al., 2022, Lepikhin et al., 2021]. While effective at preventing extreme collapse, these regularizers optimize allocation rather than *competence*: balanced traffic does not ensure that non-top experts are capable when selected. Moreover, because expert specialization is a desirable property in MoE models, uniformly pushing routing toward all experts can conflict with selective activation when certain experts are genuinely better suited to particular tokens. This motivates a complementary objective that improves worst routing outcomes without erasing useful router preferences.

Recent work has expanded MoE research beyond routing collapse and load balancing. On the expert specialization side, several approaches explicitly encourage clearer expert roles and more stable specialization, rather than only enforcing uniform utilization [Krishnamurthy et al., 2023, Dai et al., 2024]. In parallel, large-scale MoE LLMs and training recipes emphasize that expert quality and specialization remain central to scaling, not just utilization [Jiang et al., 2024]. Beyond FFN-MoE, head or attention-level MoE treats attention heads as experts, pushing sparsity and specialization into the attention mechanism itself [Jin et al., 2024, Wu et al., 2024]. For adaptation, recent work studies parameter-efficient MoE variants that preserve expert structure while substantially reducing trainable parameters [Zadouri et al., 2023].

In this work, we propose *Distributionally Robust MoE Training* (DRMoET), a training framework that makes routing robustness explicit. Unlike standard Group DRO [Sagawa et al., 2019], whose groups are predefined, DRMoET treats each layer's routed experts as endogenous groups induced by the current router. It maintains a per-layer distribution over experts and shifts mass toward high-loss expert–input interactions using EMA-smoothed, activation-weighted losses. The resulting objective sends stronger task gradients to hard-domain tokens and plausible non-top experts while leaving the router, architecture, and sparse forward/backward computation unchanged.

This formulation makes predictions beyond aggregate accuracy: it should reduce expert-quality dispersion, improve behavior when non-top experts are selected, and preserve rather than collapse specialization. We evaluate DRMoET in pretraining under the FLAME-MoE recipe at two model scales: `FLAME-MoE-290M-746M` and `FLAME-MoE-1.7B-10.3B`. Our experiments support all three: DRMoET reduces worst-expert loss and expert-loss variance while keeping mean expert loss nearly unchanged; a forced mid-k misrouting probe shows 4.3% less degradation; and functional specialization metrics, including expert–domain mutual information and competence advantage, improve under DRMoET. Across downstream benchmarks, mechanism probes, and ablations, the evidence consistently supports that improving routing robustness is a practical way to make sparse MoE capacity more reliable, not merely more balanced.

2 Related work

In this section, we discuss the differences among existing MoE training paradigms. We first review methods designed to improve load balancing in MoE training. We then provide an overview of distributionally robust optimization (DRO), which forms the foundation of DRMoET.

Mixture-of-Experts (MoE) Training. Sparsely-gated MoE layers [Shazeer et al., 2017] scale language models by routing each token to a few feed-forward *experts*. *Load-balancing* objectives enhance training stability and encourage uniform expert utilization. GShard [Lepikhin et al., 2021] adds a KL-style auxiliary loss with strict capacity limits, whereas Switch Transformer [Fedus et al., 2022] simplifies routing to top-1 selection, reducing both communication and instability. Later work formulates routing as balanced assignment: BASE Layers [Lewis et al., 2021] approximately solve a linear assignment, [Wang et al., 2024] adds a learned bias vector to the router logits, and Expert-Choice routing lets experts select tokens [Zhou et al., 2022]. Wu et al. [2024] introduce *Multi-Head MoE (MH-MoE)*, which splits each input token into h learned sub-tokens before routing. These methods make sparse MoE training practical, but primarily control *which* experts receive tokens. DRMoET is orthogonal: it leaves routing intact and improves the quality of experts that may be selected under plausible but suboptimal routing.

Distributionally Robust Optimization (DRO). DRO minimizes the *worst-case* loss over an uncertainty set $\mathcal{U}$ around the empirical distribution. f-divergence balls yield variance-regularized objectives [Duchi and Namkoong, 2019], while CVaR arises as an α-quantile special case. Group DRO [Sagawa et al., 2019] targets the hardest predefined demographic or topic group, and topic-level CVaR for language modeling [Oren et al., 2019] lowers perplexity on under-represented topics. Streaming losses can stabilize high-variance group estimates across batches [Wen et al., 2022]. DRMoET transfers this principle to MoE training, where the groups are not external labels but layer-specific expert outcomes induced by top-k routing and measured with activation-weighted attribution.

3 Preliminaries

In this section we review the core elements of sparse MoE architectures, the challenge of expert load imbalance, balancing strategies and extensions that motivate our robust training framework.

3.1 MoE Architectures

A MoE layer replaces a single dense feedforward network with a collection of n parallel expert subnetworks $\{\mathrm{FFN}_i\}_{i=1}^n$. For each token embedding $\mathbf{x} \in \mathbb{R}^d$, a router produces scores over experts. Let $\phi(\mathbf{x}) \in \mathbb{R}^n$ denote the router logits, where $\phi_i(\mathbf{x})$ is the scalar score for expert i. With optional Gumbel noise $\mathbf{g}$ and temperature $\tau > 0$, define the selected expert set and sparse routing weight as

$$\mathcal{T}(\mathbf{x}) = \mathrm{TopK}_k\big(\phi(\mathbf{x}) + \mathbf{g}\big), \qquad F_i(\mathbf{x}) = \frac{\exp((\phi_i(\mathbf{x}) + g_i)/\tau)\,\mathbf{1}[i \in \mathcal{T}(\mathbf{x})]}{\sum_{r \in \mathcal{T}(\mathbf{x})} \exp((\phi_r(\mathbf{x}) + g_r)/\tau)},$$

where $g_i = -\ln(-\ln u_i)$ with $u_i \sim \mathcal{U}(0,1)$ when noisy routing is used. Thus $F_i(\mathbf{x}) = 0$ for unselected experts and the MoE layer output is

$$\mathbf{o}(\mathbf{x}) \;=\; \sum_{i \in \mathcal{T}(\mathbf{x})} F_i(\mathbf{x})\,\mathrm{FFN}_i(\mathbf{x}).$$

Let t index the minibatch or training iteration, and let $\mathcal{X}^{(t)}$ be the multiset of all tokens in the minibatch with $N_t = |\mathcal{X}^{(t)}|$. We distinguish soft routing mass from hard selected-token frequency:

$$P_i^{(t)} \;=\; \frac{1}{N_t} \sum_{\mathbf{x} \in \mathcal{X}^{(t)}} F_i(\mathbf{x}), \qquad f_i^{(t)} \;=\; \frac{1}{N_t} \sum_{\mathbf{x} \in \mathcal{X}^{(t)}} \mathbf{1}[i \in \mathcal{T}(\mathbf{x})]\,.$$

Here $N_t P_i^{(t)}$ is the expected routing-probability mass assigned to expert i, whereas $N_t f_i^{(t)}$ is the discrete number of tokens for which expert i is selected.

3.2 Expert Load Imbalance and Balancing Strategies

In practice, the router may concentrate most tokens on a few experts, leaving the majority under-utilized. This *load imbalance* degrades both parameter efficiency and model performance, as rarely used experts receive insufficient gradient signals, and heavily used experts become bottlenecks.

A standard remedy is an *auxiliary load-balancing loss* [Fedus et al., 2022, Lepikhin et al., 2021], defined over the t-th minibatch in the common form

$$L_{\text{aux}}^{(t)} \;=\; n \sum_{i=1}^{n} f_i^{(t)} P_i^{(t)}, \qquad L^{(t)} = L_{\text{task}}^{(t)} + \lambda L_{\text{aux}}^{(t)},$$

where $f_i^{(t)}$ is the hard selected-token fraction, $P_i^{(t)}$ is the average router probability mass, and $\lambda > 0$ controls the strength of balancing. Increasing λ encourages the routing statistics to be more uniform, but can harm the primary task when balancing gradients conflict with specialization.

To avoid such interference, Wang et al. [2024] propose *Loss-Free Balancing*, which drops the auxiliary load-balancing loss. Instead, it learns a per-expert bias vector $\mathbf{b} \in \mathbb{R}^n$, adding b_i to the router logit $\phi_i(\mathbf{x})$ before top-k selection. After each batch, b_i is updated by

$$b_i \;\leftarrow\; b_i + u\,\text{sign}(\bar{n} - n_i),$$

where n_i is the hard token count for expert i, $\bar{n}$ its average, and u a small update rate. Thus overloaded experts are depressed and underloaded experts are elevated, dynamically steering the router toward balanced loads without any auxiliary gradient.

4 Method

While load balancing equalizes traffic, it can (i) interfere with the task objective and (ii) leave the model brittle when plausible but non-optimal experts are selected. We propose **DRMoET**, a *distributionally robust* training procedure that optimizes layer-wise worst expert-attributed loss *without adding additional auxiliary losses or changing the routing rule*. The key distinction from standard DRO is that the robustness groups are created by the MoE computation itself: each expert in each layer receives loss attribution through the routed tokens and activations it actually processes. This turns routing robustness into a task-loss reweighting problem, preserving the router's natural specialization while directing stronger signal to high-loss routing outcomes.

4.1 Distributionally Robust MoE Training

Experts as DRO groups. Consider a model f_θ with L MoE layers, each containing E experts and using top-k routing. For each layer l, we introduce an adversarial weight vector $\mu_{l,\cdot} \in \Delta_E$, where $\Delta_E = \{\mu \in \mathbb{R}^E_{\geq 0} : \sum_{i=1}^E \mu_i = 1\}$. Let $R_{l,i}(\theta)$ denote the population risk attributed to expert i at layer l. We study the layer-wise DRO saddle problem

$$\min_{\theta} \max_{\mu \in (\Delta_E)^L} \; F(\theta, \mu) \;\triangleq\; \sum_{l=1}^{L} \sum_{i=1}^{E} \mu_{l,i}\, R_{l,i}(\theta), \tag{1}$$

where $(\Delta_E)^L$ is the product of L simplices. Since $F(\theta, \mu)$ is linear in each $\mu_{l,\cdot}$, the inner maximization equals $\sum_l \max_i R_{l,i}(\theta)$, directly reducing the layer-wise worst expert-attributed risk.

Activation-weighted loss attribution. At iteration t, for a minibatch $\mathcal{B}$ containing N_t tokens, compute token losses $\ell_j^{(b)}(\theta) = \ell(f_\theta(x_j^{(b)}), y_j^{(b)})$. For each MoE layer l, the gate selects top-k experts $\mathcal{T}_{j,l}^{(b)} \subset [E]$ with routing probabilities $p_{l,i}(x_j^{(b)}) = \text{softmax}_{i \in \mathcal{T}_{j,l}^{(b)}}(r_{l,i}(x_j^{(b)}))$. Let $h_{l,i}(x_j^{(b)})$ be the pre-merge output activation of expert i. We define the detached activation-weighted credit

$$\tilde{c}_{l,i}(x_j^{(b)}) \;\triangleq\; \text{sg}\Big(p_{l,i}(x_j^{(b)}) \cdot \big\|h_{l,i}(x_j^{(b)})\big\|_2\Big), \tag{2}$$

where $\text{sg}(\cdot)$ denotes stop-gradient. The credit incorporates both routing probability and activation magnitude while preventing the attribution coefficient itself from becoming an additional router or activation-norm objective.

Algorithm 1 Distributionally Robust Mixture-of-Experts Training

Require: Model f_θ with L MoE layers, E experts per layer, top-k routing
Require: EMA decay β, dual step-size schedule η_t
Ensure: Learned parameters θ

Initialize $\mu_{l,i}^{(0)} \leftarrow 1/E$, $\hat{\ell}_{l,i}^{(0)} \leftarrow 0$ for all $l \in [L]$, $i \in [E]$
for $t = 1$ to T **do**
  for minibatch $\mathcal{B} = \{(x^{(b)}, y^{(b)})\}_{b=1}^{B}$ with N_t tokens **do**
    Compute token losses $\ell_j^{(b)} = \ell(f_\theta(x_j^{(b)}), y_j^{(b)})$
    Collect gate logits $r_{l,i}(x_j^{(b)})$ and expert outputs $h_{l,i}(x_j^{(b)})$
    for each MoE layer $l = 1, \ldots, L$ **do**
      $\mathcal{T}_{j,l}^{(b)} \leftarrow$ top-k experts for token j based on $r_{l,\cdot}(x_j^{(b)})$
      $p_{l,i}(x_j^{(b)}) \leftarrow \mathrm{softmax}_{i \in \mathcal{T}_{j,l}^{(b)}}(r_{l,i}(x_j^{(b)}))$
      $\tilde{c}_{l,i}(x_j^{(b)}) \leftarrow \mathrm{sg}\Big(p_{l,i}(x_j^{(b)}) \| h_{l,i}(x_j^{(b)}) \|_2\Big)$
      Compute $\ell_{l,i}^{(t)} \leftarrow \frac{1}{N_t} \sum_{b,j} \tilde{c}_{l,i}(x_j^{(b)}) \ell_j^{(b)} \mathbf{1}[i \in \mathcal{T}_{j,l}^{(b)}]$ for all i
      $\hat{\ell}_{l,\cdot}^{(t)} \leftarrow \beta \hat{\ell}_{l,\cdot}^{(t-1)} + (1-\beta)\,\mathrm{sg}(\ell_{l,\cdot}^{(t)})$
      $\mu_{l,\cdot}^{(t)} \leftarrow \mathrm{softmax}\Big(\mu_{l,\cdot}^{(t-1)} + \eta_t \hat{\ell}_{l,\cdot}^{(t)}\Big)$
    end for
    $\mathcal{L}_{\mathrm{DRO}} \leftarrow \sum_{l=1}^{L} \sum_{i=1}^{E} \mathrm{sg}(\mu_{l,i}^{(t)}) \ell_{l,i}^{(t)}$
    Backpropagate $\mathcal{L}_{\mathrm{DRO}}$ and update θ
  end for
end for

We attribute a normalized loss mass to expert (l, i):

$$\ell_{l,i}^{(t)} \;\triangleq\; \frac{1}{N_t} \sum_{(b,j) \in \mathcal{B}} \tilde{c}_{l,i}(x_j^{(b)}) \cdot \ell_j^{(b)}(\theta) \cdot \mathbf{1}\Big[i \in \mathcal{T}_{j,l}^{(b)}\Big]. \tag{3}$$

Normalizing by token count keeps the DRO term on a stable scale across micro-batch sizes and avoids changing the relative weighting of the standard task and router-regularization terms used by the training recipe. The population risk is $R_{l,i}(\theta) \triangleq \mathbb{E}[\mathcal{A}_{l,i}(x, y; \theta)]$, where $\mathcal{A}_{l,i}$ is the corresponding single-example, token-normalized analogue.

Primal–dual optimization. We maintain an EMA of attributed losses,

$$\hat{\ell}_{l,i}^{(t)} \;=\; \beta\, \hat{\ell}_{l,i}^{(t-1)} + (1-\beta)\; \mathrm{sg}(\ell_{l,i}^{(t)}), \qquad \beta \in (0,1), \tag{4}$$

and update the dual variables by the entropy-regularized softmax rule used in the implementation:

$$\mu_{l,\cdot}^{(t)} \;=\; \mathrm{softmax}\Big(\mu_{l,\cdot}^{(t-1)} + \eta_t\, \hat{\ell}_{l,\cdot}^{(t)}\Big). \tag{5}$$

This probability-space softmax rule is analyzed in Appendix A. There, we show that the implemented update can be viewed as entropic mirror ascent on a regularized robust objective, rather than as the log-space multiplicative-weights update $\mathrm{softmax}(\log \mu + \eta \hat{\ell})$. Larger EMA losses increase the relative dual mass assigned to the corresponding experts while the entropy term keeps the update smooth. Experiments use the constant parameterization $\eta_t = \eta_0/\sqrt{E}$, reported as η. The primal objective at step t is

$$\mathcal{L}_{\mathrm{DRO}}^{(t)} \;\triangleq\; \sum_{l=1}^{L} \sum_{i=1}^{E} \mathrm{sg}(\mu_{l,i}^{(t)})\, \ell_{l,i}^{(t)}, \tag{6}$$

after the EMA and dual weights have been updated outside autograd from the current detached attribution statistics. Thus DRMoET operates through normalized task-loss reweighting with detached attribution coefficients, requiring only minimal code changes with negligible overhead.

4.2 Convergence Guarantees

We analyze Algorithm 1 for the implemented probability-space softmax update

$$\mu_l^{(t+1)} = \mathrm{softmax}\Big(\mu_l^{(t)} + \eta_t \hat{\ell}_l^{(t)}\Big).$$

This update is not the log-space multiplicative-weights rule $\mathrm{softmax}(\log \mu_l^{(t)} + \eta_t \hat{\ell}_l^{(t)})$. Instead, as shown in Appendix A, for the constant-dual-step setting $\eta_t = \eta$, it is exactly entropic mirror ascent on a regularized robust objective. Define

$$\Omega(\mu_l) = -\sum_{i=1}^{E} \mu_{l,i} \log \mu_{l,i} + \frac{1}{2}\|\mu_l\|_2^2, \qquad F_\eta(\theta, \mu) = F(\theta, \mu) + \frac{1}{\eta}\sum_{l=1}^{L} \Omega(\mu_l),$$

and

$$\Phi_\eta(\theta) = \max_{\mu \in (\Delta_E)^L} F_\eta(\theta, \mu).$$

Theorem 4.1 (Convergence to stationarity of regularized DRMoET)**.** *Under the assumptions in Appendix A, let $\gamma_t = \gamma_0/\sqrt{t}$ and sample τ with probability $\Pr(\tau = t) = \gamma_t / \sum_{s=1}^{T} \gamma_s$. Then Algorithm 1 satisfies*

$$\mathbb{E}\Big[\|\nabla \Phi_\eta(\theta^{(\tau)})\|_2^2\Big] \le \widetilde{\mathcal{O}}(T^{-1/2}) + \mathcal{O}\big(\varepsilon_{\mathrm{bias}}^2 + L_{\theta\mu}^2 L^2 E^2 \eta^2 \varepsilon_{\mathrm{ema},T}^2\big)\,.$$

In the idealized population-loss setting, where $\varepsilon_{\mathrm{bias}} = 0$ and $\varepsilon_{\mathrm{ema},T} \to 0$, the expected stationarity measure vanishes at rate $\widetilde{\mathcal{O}}(T^{-1/2})$.

The guarantee is for the entropy-regularized robust objective Φ_η, which approximates the original hard worst-expert objective $\Phi(\theta) = \sum_l \max_i R_{l,i}(\theta)$ with explicit bias

$$\Phi(\theta) \le \Phi_\eta(\theta) - \frac{L}{2\eta} \le \Phi(\theta) + \frac{L \log E}{\eta}.$$

Thus DRMoET has a convergence guarantee for a smooth regularized approximation to the layer-wise DRO objective, while the approximation error to the hard max is controlled by $L \log E/\eta$.

5 Experiments

We evaluate DRMoET on large-scale, from-scratch pretraining at two model scales. All architectures, data pipelines, and evaluation protocols follow FLAME-MoE [Kang et al., 2025] to ensure direct comparability. Beyond downstream accuracy, we conduct targeted analyses to validate the routing-robustness mechanism: expert-level loss distributions, forced-misrouting probes, expert-tier/OOD checks, functional specialization, and compact hyperparameter ablations; training-throughput results are deferred to the appendix.

5.1 Pretraining Setup

Data. All models are trained on data sampled from DataComp-LM (DCLM) [Li et al., 2024], following the FLAME-MoE data pipeline with the standard next-token prediction objective. The `FLAME-MoE-290M-746M` experiments use approximately 33.5B tokens. The main `FLAME-MoE-1.7B-10.3B` comparison uses approximately 67B tokens. Within each model scale and token budget, the compared methods use the same architecture and data pipeline.

Model. We train two configurations from scratch. The `FLAME-MoE-290M-746M` model is a decoder-only Transformer with E=32 experts per MoE layer and top-k routing with k=8, of which 2 are shared (always active) experts [Kang et al., 2025]. The `FLAME-MoE-1.7B-10.3B` model uses E=64 experts to test whether gains persist at higher scale.

Optimization. We adopt the FLAME-MoE training recipe: Adam optimizer with maximum learning rate 3×10^{-4}, global batch size 1024, sequence length 2048, and a Warmup–Stable–Decay (WSD) schedule with warmup ratio 0.01 and decay ratio 0.1. The global batch is formed from per-GPU micro-batches; the 290M runs use micro-batch size 16, while the 1.7B runs use micro-batch size 8 because of memory constraints. Both the baseline and DRMoET use the same auxiliary load-balancing loss and z-loss coefficients of 1×10^{-3}.

Baselines. At each scale, we compare against (i) standard FLAME-MoE training and (ii) auxiliary-loss-free balancing baselines [Wang et al., 2024] using the Megatron-LM implementation. For the 290M model, we report $u \in \{10^{-3}, 10^{-2}\}$. At the 1.7B-active scale and 67B-token budget, we report the u=10^{-2} setting, which is independently tuned with a larger batch size.

Scale	Method	ARC-C	ARC-E	HellaSwag	PIQA	WinoGrande	SciQ	ReCoRD*	Average
`290M-746M`									Training tokens: 33.5B
	FLAME-MoE	0.2329	0.5295	0.3447	0.6697	0.5012	0.8020	0.7023	0.5403
	Aux-free (u=10^{-3})	0.2261	0.4878	0.3283	0.6670	0.4949	0.7100	0.6445	0.5084
	Aux-free (u=10^{-2})	0.2184	0.4895	0.3301	0.6627	0.4949	0.6990	0.6440	0.5055
	DRMoET (η=0.1)	**0.2338**	0.5522	**0.3472**	0.6844	**0.5209**	0.7950	**0.7115**	0.5493
	DRMoET (η=0.01)	0.2270	0.5400	0.3470	0.6839	0.4941	0.8180	0.7007	0.5444
	DRMoET (η=0.001)	0.2295	**0.5682**	0.3452	0.6888	0.5051	**0.8190**	0.7105	**0.5523**
	DRMoET (η=0.0001)	0.2193	0.5556	0.3461	**0.6970**	0.5162	0.8060	0.7037	0.5491
`1.7B-10.3B`									Training tokens: 67B
	FLAME-MoE	0.3481	0.6970	0.4942	0.7563	0.5904	0.9050	**0.8467**	0.6625
	Aux-free (u=10^{-2})	0.3148	0.6843	0.4667	0.7459	0.5722	0.8930	0.8248	0.6431
	DRMoET (η=0.01)	**0.3805**	**0.7210**	0.4973	**0.7666**	**0.6235**	0.9020	0.8457	**0.6767**
	DRMoET (η=0.001)	0.3652	0.7024	**0.4996**	0.7628	0.6148	**0.9110**	0.8452	0.6716

Table 1: **Main results at two scales.** The two blocks use approximately 33.5B and 67B training tokens, respectively. All DRMoET rows use $\beta = 0.999$. Averages are over the seven displayed tasks. Best per-task and average within each block are in **bold**. *ReCoRD is reported as F1.

5.2 Evaluation

We report 0-shot performance following FLAME-MoE [Kang et al., 2025]; ReCoRD is reported as F1, all others as accuracy. We evaluate on ARC-E / ARC-C, HellaSwag, PIQA, WinoGrande, SciQ, and ReCoRD.

ARC-Easy / ARC-Challenge [Clark et al., 2018] evaluate grade-school science knowledge and deeper scientific reasoning, respectively. **HellaSwag** [Zellers et al., 2019] tests grounded commonsense inference. **PIQA** [Bisk et al., 2020] probes physical commonsense. **WinoGrande** [Sakaguchi et al., 2021] evaluates commonsense through pronoun resolution. **SciQ** [Welbl et al., 2017] assesses scientific reasoning across biology, chemistry, and physics. **ReCoRD** [Zhang et al., 2018] evaluates reading comprehension requiring entity reasoning over long contexts.

5.3 Main Results

Tab. 1 summarizes downstream performance at both model scales. At `290M-746M`, DRMoET improves over FLAME-MoE for every η setting, with the best average ($\eta = 0.001$) increasing from 0.5403 to 0.5523. The sweep shows predictable task tradeoffs: larger η favors ReCoRD and WinoGrande, while $\eta = 0.001$ gives the strongest average and the best ARC-E/SciQ scores.

Gains persist at `1.7B-10.3B` under the 67B-token budget: DRMoET with $\eta = 0.01$ improves the seven-task average from 0.6625 to 0.6767, an absolute gain of 1.42 percentage points and a relative improvement of 2.14%. This configuration improves five of seven tasks; $\eta = 0.001$ reaches 0.6716 and improves six of seven tasks. FLAME-MoE retains the highest ReCoRD score.

Metric	Baseline	DRMoET	Δ (%)
Worst loss	2.6930	**2.6379**	-2.05
Best loss	2.1149	**2.0722**	-2.02
Mean loss	2.3712	**2.3699**	-0.05
Range	0.5781	**0.5656**	-2.16
Std.	0.1516	**0.1386**	-8.58
CV	0.0639	**0.0585**	-8.45

Table 2: Expert-loss statistics at convergence.

Comparison with auxiliary-loss-free balancing. At 290M-active scale, both auxiliary-loss-free baselines underperform FLAME-MoE baseline, as well as DRMoET results. This could be due to the small batch size we use during training. At the 1.7B-active scale, the 67B-token auxiliary-loss-free baseline with u=10^{-2}, independently tuned with a larger batch, reaches 0.6431. The result remains 1.94 points below FLAME-MoE and 3.36 points below the best DRMoET result.

5.4 Routing Robustness Analysis

The preceding results establish empirical performance, we now shift to understand how DRMoET improves robustness to suboptimal routing.

Expert-level loss distributions and forced-misrouting probe. We first examine whether DRMoET narrows the quality gap across routing outcomes on a 52M tokens validation set, and then directly

test robustness by forcing suboptimal expert selection. For the expert-level analysis, we measure per-expert loss at convergence on the best 290M model.

As shown in Tab. 2, DRMoET reduces the worst-expert loss by 2.05% while keeping mean expert loss nearly unchanged (-0.05%). The loss standard deviation and coefficient of variation decrease by over 8%, indicating substantially more uniform expert quality.

For the misrouting probe, we replace each token's top-k experts with its middle-k experts (ranked by routing probability), simulating a realistic scenario where the router makes plausible but non-optimal selections, and measure per-token loss across six evaluation domains.

Model	Normal	Misrouting	Deg.
Baseline	3.187	7.249	$2.27\times$
DRMoET	**3.182**	**7.071**	$2.22\times$

Table 3: Forced mid-k misrouting.

Tab. 3 directly validates the core claim: under forced misrouting, DRMoET exhibits 4.3% less loss degradation than the baseline, confirming that the model is more robust when routing is imperfect.

Expert tier analysis and out-of-distribution robustness. To understand which experts benefit most, we rank all experts by total routing traffic and partition them into top, mid, and bottom tiers. We further evaluate expert loss uniformity under distribution shift using 60,000 OOD examples (4.2M tokens) from six benchmark families not seen during training.

Tab. 4(a) shows that mid-tier experts improve the most, consistent with DRMoET strengthening the experts that matter most under plausible but non-optimal routing. Tab. 4(b) extends this to unseen data: mean loss is nearly unchanged, while the loss range, standard deviation, and coefficient of variation (CV=std./mean) decrease, indicating less dispersed expert losses.

Tier	FLAME-MoE	DRMoET	Improvement(%)
Top	3.176	3.188	-0.4%
Mid	3.100	3.077	$+0.7\%$
Bottom	3.072	3.058	$+0.4\%$

(a) Per-tier loss.

Metric	FLAME-MoE	DRMoET	$\Delta(\%)$
Mean loss	3.151	3.157	$+0.2\%$
Range	1.018	0.947	-7.0%
Std.	0.241	0.236	-2.1%
CV	0.076	0.075	-1.3%

(b) OOD uniformity.

Table 4: **Expert tier and OOD analysis.** Mid-tier experts improve the most, and under distribution shift DRMoET yields more uniform expert quality while keeping mean loss nearly unchanged.

Functional specialization. A natural concern is whether improved routing robustness reduces expert specialization. We test this with 10k examples per benchmark, grouped into six functional domains, and compute specialization metrics via teacher-forced inference.

Here Max Selectivity is each expert's largest domain-selection share, ΔS is the gap between its top and second domain-selection shares, ΔQ is the competence advantage on its preferred domain, and $I(E;G)$ is mutual information between selected expert E and functional domain G. All four metrics improve under DRMoET (Tab. 5), indicating stronger domain–expert association and better performance on each expert's specialized domains.

5.5 Ablation Studies

Activation-weighted vs. raw-probability credit. We compare activation-weighted credit (Eq. 2) against a variant using only raw routing probability. Activation-weighted credit improves the best average from 0.6138 to 0.6263 at matched η (Tab. 6a), supporting attribution that reflects both selection and response magnitude. This is also consistent with the MoE forward computation: routing probability scales the expert activation, so credit should reflect the routed activation rather than probability alone.

Top-k routing sparsity and EMA decay β. We ablate routing sparsity by increasing from top-6 to top-12 (excluding shared experts), and sweep β at fixed $\eta = 0.001$.

Under denser routing (Tab. 6b), DRMoET still improves, though the gain is smaller because larger k reduces sensitivity to any single expert choice. In the β sweep (Tab. 6c), β=0.9 is strongest on ARC-E and HellaSwag, β=0.99 on PIQA, and β=0.999 on SciQ and ReCoRD while also giving the best average. We therefore use β=0.999 as the default.

Ablation	Setting	ARC-E	HellaSwag	PIQA	SciQ	ReCoRD*	Average
Credit	FLAME-MoE	0.5295	0.3447	0.6697	0.8020	0.7023	0.6096
	Activation weighted, $\eta = 10^{-2}$	0.5400	**0.3470**	0.6839	0.8180	0.7007	0.6179
	Activation weighted, $\eta = 10^{-3}$	**0.5682**	0.3452	**0.6888**	**0.8190**	**0.7105**	**0.6263**
	Raw probability, $\eta = 10^{-2}$	0.5455	0.3384	0.6839	0.7920	0.6938	0.6107
	Raw probability, $\eta = 10^{-3}$	0.5421	0.3435	0.6861	0.7910	0.7065	0.6138
Top-12 routing	FLAME-MoE	0.5459	0.3506	**0.6991**	0.7950	0.7033	0.6188
	$\eta = 10^{-3}$	**0.5640**	**0.3523**	0.6910	**0.8130**	**0.7056**	**0.6252**
	$\eta = 10^{-2}$	0.5619	0.3515	**0.6991**	0.8110	0.6998	0.6247
EMA decay β	FLAME-MoE	0.5295	0.3447	0.6697	0.8020	0.7023	0.6096
	$\beta = 0.9$	0.5568	**0.3477**	0.6915	0.7970	0.7048	0.6196
	$\beta = 0.99$	0.5408	0.3455	**0.6937**	0.7940	0.7022	0.6152
	$\beta = 0.999$	**0.5682**	0.3452	0.6888	**0.8190**	**0.7105**	**0.6263**
	$\beta = 1$	0.5467	0.3465	0.6844	0.7850	0.7068	0.6139

Table 6: **Ablation studies.** We ablate credit assignment, routing sparsity, and EMA smoothing. Averages are over the five displayed tasks within each ablation block. *ReCoRD is reported as F1.

Interaction with load balancing. Removing the standard load-balancing loss from DRMoET while retaining the remaining 290M training configuration yields a seven-task average of 0.5448, compared with 0.5403 for standard FLAME-MoE and 0.5523 for DRMoET with balancing (Appendix E). The combined configuration performs best, supporting the use of DRMoET as a complement to balancing. This comparison does not include FLAME-MoE without balancing, so it does not isolate DRMoET's incremental effect within a no-balancing recipe.

Metric	Improvement (%)
Avg Max Selectivity	$+1.0\%$
Avg ΔS (specialization margin)	$+1.2\%$
Avg ΔQ (competence advantage)	$+30.2\%$
Avg MI $I(E;G)$	$+12.9\%$

Table 5: Domain–expert specialization metrics. DRMoET improves all measures.

5.6 Training throughput

Computational complexity. Let N_t be the number of tokens per step, d the hidden size, and d_{ff} the expert FFN width. Per MoE layer, the selected expert FFNs require $\mathcal{O}(N_t k d d_{\text{ff}})$ operations. DRMoET reuses their outputs and routing probabilities: computing activation norms costs $\mathcal{O}(N_t k d)$, sparse loss attribution costs $\mathcal{O}(N_t k)$, and the EMA and softmax updates cost $\mathcal{O}(E)$. Across L layers, the additional arithmetic is therefore $\mathcal{O}(L N_t k d + L E)$, compared with $\mathcal{O}(L N_t k d d_{\text{ff}})$ for the expert FFNs. The relative overhead is $\mathcal{O}(1/d_{\text{ff}} + E/(N_t k d d_{\text{ff}}))$, reducing to $\mathcal{O}(1/d_{\text{ff}})$ when $E = \mathcal{O}(N_t k d)$. The persistent EMA and dual-weight state contains only $\mathcal{O}(LE)$ scalars. Since attribution and dual weights are detached, the objective reweights existing token losses and uses a single backward pass, with no additional expert forward passes. This operation count explains the small added computation; wall-clock effects are measured below.

Measured throughput. A practical concern for distributionally robust MoE training is whether the additional robustness machinery introduces non-trivial runtime overhead. We therefore report end-to-end training throughput measured in TFLOP/s/GPU and iteration time (ms) under identical software settings and comparable hardware. As shown in Tab. 7, *DRMoET does not incur a slowdown relative to the baseline*: iteration time is slightly lower and throughput is modestly higher, with lower variance as well. The observed speedup should be interpreted as a measured systems outcome, potentially reflecting reduced straggler variance rather than an algorithmic guarantee of faster training.

Metric	Baseline	DRMoET	Δ
TFLOP/s/GPU	64.17 ± 1.81	67.19 ± 0.93	$+4.7\%$
Time/iter (ms)	6460 ± 185	6166 ± 87	-4.6%
Throughput std	1.81	0.93	$2\times$

Table 7: **Training throughput comparison.** Mean $\pm$ std across measured iterations. DRMoET introduces no slowdown and exhibits lower variance.

6 Discussion

DRMoET improves routing robustness. Our analysis suggest that DRMoET improves MoE performance not simply by changing the average expert utilization, but by reducing the penalty of imperfect routing. The forced misrouting probe shows that when tokens are deliberately assigned to plausible but lower-ranked experts, DRMoET suffers less degradation than the baseline, indicating that non-top routing paths become more reliable. This is consistent with the expert-tier and loss-dispersion analyses: DRMoET narrows expert-quality gaps while preserving useful specialization, rather than collapsing all experts toward uniform behavior. These findings support the central motivation of DRMoET: routing robustness is complementary to load balancing, because a balanced router is not sufficient if the experts selected under uncertainty remain unevenly trained.

Limitations. We evaluate DRMoET at two model scales and a fixed set of expert configurations, future work should test whether the same routing-robustness patterns hold across broader model families, expert counts, and routing depths. Our experiments focus on standard pretraining and benchmark settings; we have not yet explicitly constructed highly imbalanced pretraining mixtures or long-tail evaluation suites, where rare domains and skewed routing patterns may more directly stress expert robustness. Such settings could provide a sharper test of whether DRMoET improves reliability under distributional imbalance. Our forced-misrouting and expert-tier/OOD analyses are diagnostic probes on trained checkpoints; extending these measurements throughout training could provide a more detailed view of how expert robustness and specialization emerge.

Conclusion

We introduced DRMoET, a layer-wise expert-level DRO objective for MoE training that improves robustness to suboptimal routing. The central lesson is that sparse scaling should not be judged only by balanced utilization: real routers are imperfect, and distribution shifts can send tokens through weaker, non-top experts. DRMoET addresses this reliability gap by optimizing high-loss expert-attributed outcomes while preserving the router and the standard sparse computation path. Empirically, it reduces expert-loss standard deviation by over 8%, reduces excess loss under forced misrouting by 4.3%, preserves and even strengthens specialization, and outperforms both standard FLAME-MoE and auxiliary-loss-free balancing at the `290M-746M` and `1.7B-10.3B` scales. At 67B training tokens, the best large-model result reaches 0.6767 versus 0.6625 for FLAME-MoE and 0.6431 for auxiliary-loss-free balancing, showing that the downstream advantage persists under longer training. These results suggest that routing robustness is a practical objective for making MoE capacity gains more reliable.

Acknowledgement

This work is supported by NYU Shanghai Center for Data Science. This work is supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

References

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

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

John Duchi and Hongseok Namkoong. Variance-based regularization with convex objectives. *Journal of Machine Learning Research*, 20(68):1–55, 2019.

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.

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

Peng Jin, Bo Zhu, Li Yuan, and Shuicheng Yan. Moh: Multi-head attention as mixture-of-head attention. *arXiv preprint arXiv:2410.11842*, 2024.

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

Yamuna Krishnamurthy, Chris Watkins, and Thomas Gaertner. Improving expert specialization in mixture of experts. *arXiv preprint arXiv:2302.14703*, 2023.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, et al. Gshard: Scaling giant models with conditional computation and automatic sharding. In *ICLR*, 2021.

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

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

Yonatan Oren, Shiori Sagawa, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust language modeling. *arXiv preprint arXiv:1909.02060*, 2019.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

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*, 2024.

Johannes Welbl, Nelson F Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*, 2017.

Hongyi Wen, Xinyang Yi, Tiansheng Yao, Jiaxi Tang, Lichan Hong, and Ed H Chi. Distributionally-robust recommendations for improving worst-case user experience. In *Proceedings of the ACM Web Conference 2022*, pages 3606–3610, 2022.

Xun Wu, Shaohan Huang, Wenhui Wang, Shuming Ma, Li Dong, and Furu Wei. Multi-head mixture-of-experts. *Advances in Neural Information Processing Systems*, 37:94073–94096, 2024.

Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermiş, Acyr Locatelli, and Sara Hooker. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. *arXiv preprint arXiv:2309.05444*, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*, 2018.

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.

A Analysis of DRMoET

We analyze the detached-credit version of DRMoET stated in Algorithm 1. The analysis uses a differentiable relaxation of the router, or equivalently holds locally within fixed top-k routing regions. This is the standard way to state a smooth nonconvex guarantee for sparse routers; the exact hard top-k map is discontinuous at routing ties.

The key point is that the implemented update

$$\mu_l^{(t+1)} = \mathrm{softmax}\Big(\mu_l^{(t)} + \eta \hat{\ell}_l^{(t)}\Big)$$

is not the log-space multiplicative-weights update $\mathrm{softmax}(\log \mu_l^{(t)} + \eta \hat{\ell}_l^{(t)})$. Instead, it is exactly entropic mirror ascent on a regularized robust objective. This section gives the formal statement and convergence guarantee.

A.1 Problem formulation

Let the training distribution over examples be $\mathcal{D}$. Each example $(x, y) \sim \mathcal{D}$ contains tokens indexed by j. At iteration t on a minibatch $\mathcal{B}$ with N_t tokens, define token losses

$$\ell_j^{(b)}(\theta) = \ell\Big(f_\theta(x_j^{(b)}), y_j^{(b)}\Big).$$

For an MoE layer l with top-k routing, let $\mathcal{T}_{j,l}^{(b)}$ be the selected experts for token $x_j^{(b)}$, and define routing probabilities restricted to the selected set

$$p_{l,i}\Big(x_j^{(b)}\Big) = \begin{cases} \mathrm{softmax}_{r \in \mathcal{T}_{j,l}^{(b)}}\big(r_{l,r}(x_j^{(b)})\big)_i, & i \in \mathcal{T}_{j,l}^{(b)}, \\ 0, & i \notin \mathcal{T}_{j,l}^{(b)}. \end{cases}$$

Let $h_{l,i}(x_j^{(b)})$ denote the pre-merge output activation of expert i at layer l. We use detached activation-weighted credit

$$\tilde{c}_{l,i}\Big(x_j^{(b)}\Big) = \mathrm{sg}\Big(p_{l,i}\Big(x_j^{(b)}\Big)\, \big\|h_{l,i}(x_j^{(b)})\big\|_2\Big), \tag{7}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient.

The normalized minibatch attribution is

$$\ell_{l,i}^{(t)} = \frac{1}{N_t} \sum_{(b,j)\in\mathcal{B}} \tilde{c}_{l,i}\Big(x_j^{(b)}\Big)\, \ell_j^{(b)}(\theta) \mathbf{1}\Big[i \in \mathcal{T}_{j,l}^{(b)}\Big]. \tag{8}$$

Define the corresponding population risks as

$$R_{l,i}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[\mathcal{A}_{l,i}(x, y; \theta)\big],$$

where $\mathcal{A}_{l,i}$ is the single-example normalized analogue of (8). The unregularized layer-wise DRO objective is

$$F(\theta, \mu) = \sum_{l=1}^{L} \sum_{i=1}^{E} \mu_{l,i} R_{l,i}(\theta), \qquad \mu_l \in \Delta_E,$$

and

$$\Phi(\theta) = \max_{\mu \in (\Delta_E)^L} F(\theta, \mu) = \sum_{l=1}^{L} \max_{i\in[E]} R_{l,i}(\theta).$$

For the implemented softmax dual update, define

$$\Omega(\mu_l) = H(\mu_l) + \frac{1}{2}\|\mu_l\|_2^2 = -\sum_{i=1}^{E} \mu_{l,i} \log \mu_{l,i} + \frac{1}{2}\|\mu_l\|_2^2,$$

and the regularized robust objective

$$F_\eta(\theta, \mu) = F(\theta, \mu) + \frac{1}{\eta} \sum_{l=1}^{L} \Omega(\mu_l), \qquad \Phi_\eta(\theta) = \max_{\mu \in (\Delta_E)^L} F_\eta(\theta, \mu). \tag{9}$$

A.2 Standing assumptions

(A1) **Bounded risks and attributions.** There exists $G > 0$ such that $0 \le \ell_{l,i}^{(t)} \le G$, $0 \le \hat{\ell}_{l,i}^{(t)} \le G$, and $0 \le R_{l,i}(\theta^{(t)}) \le G$ for all l, i, t along the training trajectory.

(A2) **Smooth stop-gradient risks.** Each detached-credit risk $R_{l,i}(\theta)$ is L_θ-smooth in θ under the differentiable routing relaxation. The regularized value function Φ_η is L_Φ-smooth and bounded below by $\Phi_{\eta,\inf}$.

(A3) **Stochastic primal gradients.** Let $g^{(t)}$ denote the stochastic gradient obtained by backpropagating $\sum_{l,i} \mathrm{sg}(\mu_{l,i}^{(t)}) \ell_{l,i}^{(t)}$ with detached credits. For fixed $(\theta^{(t)}, \mu^{(t)})$,

$$\left\| \mathbb{E}\left[g^{(t)} \mid \theta^{(t)}, \mu^{(t)} \right] - \nabla_\theta F(\theta^{(t)}, \mu^{(t)}) \right\|_2 \le \varepsilon_{\mathrm{bias}},$$

and

$$\mathbb{E}\left[\|g^{(t)}\|_2^2 \mid \theta^{(t)}, \mu^{(t)} \right] \le M^2.$$

(A4) **Lipschitz dependence on the dual variable.** There exists $L_{\theta\mu} > 0$ such that for all θ and $\mu, \nu \in (\Delta_E)^L$,

$$\|\nabla_\theta F(\theta, \mu) - \nabla_\theta F(\theta, \nu)\|_2 \le L_{\theta\mu} \sum_{l=1}^{L} \|\mu_l - \nu_l\|_1.$$

(A5) **Lipschitz risks.** There exists $L_R > 0$ such that for every layer l,

$$\|R_l(\theta) - R_l(\theta')\|_\infty \le L_R \|\theta - \theta'\|_2,$$

where $R_l(\theta) = (R_{l,1}(\theta), \dots, R_{l,E}(\theta))$.

(A6) **EMA estimation error.** Define

$$\xi_t = \max_{l \in [L]} \left\| \hat{\ell}_l^{(t)} - R_l(\theta^{(t)}) \right\|_\infty.$$

For $\gamma_t = \gamma_0/\sqrt{t}$, define the weighted average EMA error

$$\varepsilon_{\mathrm{ema},T}^2 = \frac{\sum_{t=1}^{T} \gamma_t \mathbb{E}[\xi_t^2]}{\sum_{t=1}^{T} \gamma_t}.$$

This term captures both minibatch noise and EMA lag. It vanishes in the idealized population-loss setting or under a sufficiently accurate increasing-batch/variance-reduced estimate.

(A7) **Step sizes.** The primal update is

$$\theta^{(t+1)} = \theta^{(t)} - \gamma_t g^{(t)}, \qquad \gamma_t = \gamma_0/\sqrt{t}.$$

The dual update uses the implemented softmax rule

$$\mu_l^{(t+1)} = \mathrm{softmax}\left(\mu_l^{(t)} + \eta \hat{\ell}_l^{(t)} \right),$$

with fixed $\eta = \eta_0/\sqrt{E}$.

For each θ, let

$$\mu_\eta^\star(\theta) \in \arg\max_{\mu \in (\Delta_E)^L} F_\eta(\theta, \mu)$$

denote the regularized dual response.

A.3 The implemented update as mirror ascent

Lemma A.1 (Regularized mirror-ascent form)**.** *For fixed θ and layer l, one entropic mirror-ascent step on F_η with KL geometry and step size η is*

$$\mu_l^+ = \arg\max_{\nu \in \Delta_E} \{\eta \langle \nu, \nabla_{\mu_l} F_\eta(\theta, \mu)\rangle - D_{\mathrm{KL}}(\nu \| \mu_l)\}.$$

This update has the closed form

$$\mu_l^+ = \mathrm{softmax}(\mu_l + \eta R_l(\theta)).$$

Replacing $R_l(\theta)$ by the EMA estimate $\hat{\ell}_l^{(t)}$ gives the implemented update in Algorithm 1.

Proof. For the regularizer

$$\Omega(\mu_l) = -\sum_i \mu_{l,i} \log \mu_{l,i} + \frac{1}{2}\|\mu_l\|_2^2,$$

we have

$$\nabla_{\mu_l} \Omega(\mu_l) = \mu_l - \log \mu_l - \mathbf{1},$$

where the logarithm is taken elementwise. Therefore

$$\nabla_{\mu_l} F_\eta(\theta, \mu) = R_l(\theta) + \frac{1}{\eta}(\mu_l - \log \mu_l - \mathbf{1}).$$

The KKT conditions for the entropic mirror-ascent subproblem give

$$\mu_{l,i}^+ \propto \mu_{l,i} \exp(\eta R_{l,i}(\theta) + \mu_{l,i} - \log \mu_{l,i} - 1).$$

The multiplicative base $\mu_{l,i}$ cancels the $-\log \mu_{l,i}$ term, and the constant -1 is absorbed into normalization. Hence

$$\mu_l^+ = \mathrm{softmax}(\mu_l + \eta R_l(\theta)).$$

□

A.4 Dual tracking

For each fixed θ, define the population softmax response map

$$T_{\eta,\theta,l}(\mu_l) = \mathrm{softmax}(\mu_l + \eta R_l(\theta)).$$

The regularized maximizer $\mu_{\eta,l}^\star(\theta)$ is the unique fixed point of this map:

$$\mu_{\eta,l}^\star(\theta) = T_{\eta,\theta,l}\big(\mu_{\eta,l}^\star(\theta)\big).$$

Lemma A.2 (Softmax response is contractive)**.** *For any $a, b \in \mathbb{R}^E$,*

$$\|\mathrm{softmax}(a) - \mathrm{softmax}(b)\|_\infty \le \frac{1}{2}\|a - b\|_\infty.$$

Consequently, for any θ, θ' and any $\mu, \nu \in \Delta_E$,

$$\|T_{\eta,\theta,l}(\mu) - T_{\eta,\theta',l}(\nu)\|_\infty \le \frac{1}{2}\|\mu - \nu\|_\infty + \frac{\eta}{2}\|R_l(\theta) - R_l(\theta')\|_\infty.$$

Proof. Let $s = \mathrm{softmax}(z)$. The Jacobian of softmax satisfies

$$\frac{\partial s_i}{\partial z_j} = s_i(\mathbf{1}[i = j] - s_j).$$

For each row,

$$\sum_j \left|\frac{\partial s_i}{\partial z_j}\right| = 2s_i(1 - s_i) \le \frac{1}{2}.$$

Hence softmax is $1/2$-Lipschitz from ℓ_∞ to ℓ_∞. Applying this to $a = \mu + \eta R_l(\theta)$ and $b = \nu + \eta R_l(\theta')$ gives the result. □

Lemma A.3 (Tracking of the regularized dual maximizer)**.** *Define*

$$e_t = \max_{l\in[L]} \left\| \mu_l^{(t)} - \mu_{\eta,l}^\star(\theta^{(t)}) \right\|_\infty .$$

Then the implemented update satisfies

$$e_{t+1} \le \frac{1}{2} e_t + \frac{\eta}{2}\xi_t + \eta L_R \gamma_t \|g^{(t)}\|_2.$$

Consequently, for $\gamma_t = \gamma_0/\sqrt{t}$,

$$\frac{\sum_{t=1}^T \gamma_t \mathbb{E}[e_t^2]}{\sum_{t=1}^T \gamma_t} \le C \left(\frac{e_1^2}{\sum_{t=1}^T \gamma_t} + \eta^2 \varepsilon_{\mathrm{ema},T}^2 + \eta^2 L_R^2 M^2 \frac{\sum_{t=1}^T \gamma_t^3}{\sum_{t=1}^T \gamma_t} \right)$$

for a universal constant C.

Proof. Using the implemented update and the fixed-point identity for $\mu_{\eta,l}^\star(\theta^{(t)})$,

$$\begin{aligned}
&\left\| \mu_l^{(t+1)} - \mu_{\eta,l}^\star(\theta^{(t)}) \right\|_\infty \\
&\le \left\| \mathrm{softmax}\Big(\mu_l^{(t)} + \eta \hat{\ell}_l^{(t)}\Big) - \mathrm{softmax}\Big(\mu_{\eta,l}^\star(\theta^{(t)}) + \eta R_l(\theta^{(t)})\Big) \right\|_\infty \\
&\le \frac{1}{2} \left\| \mu_l^{(t)} - \mu_{\eta,l}^\star(\theta^{(t)}) \right\|_\infty + \frac{\eta}{2} \left\| \hat{\ell}_l^{(t)} - R_l(\theta^{(t)}) \right\|_\infty .
\end{aligned}$$

It remains to account for the movement of the target $\mu_{\eta,l}^\star(\theta)$ between $\theta^{(t)}$ and $\theta^{(t+1)}$. By Lemma A.2 and the Lipschitzness of R_l,

$$\left\| \mu_{\eta,l}^\star(\theta^{(t+1)}) - \mu_{\eta,l}^\star(\theta^{(t)}) \right\|_\infty \le \eta \left\| R_l(\theta^{(t+1)}) - R_l(\theta^{(t)}) \right\|_\infty \le \eta L_R \gamma_t \|g^{(t)}\|_2.$$

Combining the two inequalities and maximizing over l gives the one-step recursion.

Unrolling the geometric recursion, squaring, using $(a+b+c)^2 \le 3(a^2+b^2+c^2)$, and applying $\mathbb{E}\|g^{(t)}\|_2^2 \le M^2$ gives

$$\mathbb{E}[e_t^2] \le C \left(2^{-t} e_1^2 + \eta^2 \sum_{s=1}^{t-1} 2^{-(t-1-s)} \mathbb{E}[\xi_s^2] + \eta^2 L_R^2 \sum_{s=1}^{t-1} 2^{-(t-1-s)} \gamma_s^2 M^2 \right).$$

Multiplying by γ_t, summing over t, using the monotonicity of γ_t, and dividing by $\sum_t \gamma_t$ yields the stated weighted tracking bound. □

A.5 Connection to the original hard DRO objective

Lemma A.4 (Approximation to the hard worst-expert objective)**.** *Let*

$$\Phi(\theta) = \max_{\mu\in(\Delta_E)^L} F(\theta,\mu) = \sum_{l=1}^L \max_i R_{l,i}(\theta).$$

Then, for all θ,

$$\Phi(\theta) \le \Phi_\eta(\theta) - \frac{L}{2\eta} \le \Phi(\theta) + \frac{L\log E}{\eta}.$$

Proof. For each layer, if $e_{i^\star}$ is the one-hot vector on a maximizer of $R_{l,i}(\theta)$, then

$$\Omega(e_{i^\star}) = \frac{1}{2}.$$

Therefore

$$\Phi_\eta(\theta) \ge \Phi(\theta) + \frac{L}{2\eta}.$$

Conversely, for every $\mu_l \in \Delta_E$,

$$F(\theta,\mu) \le \Phi(\theta), \qquad \Omega(\mu_l) = H(\mu_l) + \frac{1}{2}\|\mu_l\|_2^2 \le \log E + \frac{1}{2}.$$

Thus

$$\Phi_\eta(\theta) \le \Phi(\theta) + \frac{L}{\eta}\left(\log E + \frac{1}{2}\right).$$

Subtracting $L/(2\eta)$ gives the result. □

A.6 Convergence to stationarity

Theorem A.5 (Convergence to stationarity of regularized DRMoET)**.** *Under Assumptions (A1)–(A7), Algorithm 1 satisfies*

$$\frac{\sum_{t=1}^{T}\gamma_t\mathbb{E}\left[\|\nabla\Phi_\eta(\theta^{(t)})\|_2^2\right]}{\sum_{t=1}^{T}\gamma_t} \le \widetilde{\mathcal{O}}(T^{-1/2}) + \mathcal{O}\left(\varepsilon_{\text{bias}}^2 + L_{\theta\mu}^2L^2E^2\eta^2\varepsilon_{\text{ema},T}^2\right).$$

Equivalently, if τ is sampled with $\Pr(\tau=t)=\gamma_t/\sum_{s=1}^{T}\gamma_s$, then

$$\mathbb{E}\left[\|\nabla\Phi_\eta(\theta^{(\tau)})\|_2^2\right] \le \widetilde{\mathcal{O}}(T^{-1/2}) + \mathcal{O}\left(\varepsilon_{\text{bias}}^2 + L_{\theta\mu}^2L^2E^2\eta^2\varepsilon_{\text{ema},T}^2\right).$$

In the idealized population-loss setting, where $\varepsilon_{\text{bias}}=0$ and $\varepsilon_{\text{ema},T}\to 0$, the expected stationarity measure vanishes at rate $\widetilde{\mathcal{O}}(T^{-1/2})$.

Proof. By Danskin's theorem applied to the regularized value function,

$$\nabla\Phi_\eta(\theta^{(t)}) = \nabla_\theta F(\theta^{(t)}, \mu_\eta^\star(\theta^{(t)})),$$

because the regularizer in F_η does not depend on θ.

Let

$$\delta_t = \mathbb{E}\left[g^{(t)} \mid \theta^{(t)}, \mu^{(t)}\right] - \nabla\Phi_\eta(\theta^{(t)}).$$

Using Assumption (A3) and the Lipschitz dependence of $\nabla_\theta F$ on μ,

$$\|\delta_t\|_2 \le \varepsilon_{\text{bias}} + L_{\theta\mu}\sum_{l=1}^{L}\left\|\mu_l^{(t)} - \mu_{\eta,l}^\star(\theta^{(t)})\right\|_1.$$

Since $\|\cdot\|_1 \le E\|\cdot\|_\infty$,

$$\|\delta_t\|_2 \le \varepsilon_{\text{bias}} + L_{\theta\mu}LEe_t.$$

Using L_Φ-smoothness of Φ_η,

$$\Phi_\eta(\theta^{(t+1)}) \le \Phi_\eta(\theta^{(t)}) - \gamma_t\left\langle\nabla\Phi_\eta(\theta^{(t)}), g^{(t)}\right\rangle + \frac{L_\Phi\gamma_t^2}{2}\|g^{(t)}\|_2^2.$$

Taking conditional expectation and applying $-\langle a, a+\delta\rangle \le -\frac{1}{2}\|a\|_2^2 + \frac{1}{2}\|\delta\|_2^2$ gives

$$\mathbb{E}\left[\Phi_\eta(\theta^{(t+1)})\right] \le \mathbb{E}\left[\Phi_\eta(\theta^{(t)})\right] - \frac{\gamma_t}{2}\mathbb{E}\left[\|\nabla\Phi_\eta(\theta^{(t)})\|_2^2\right] + \frac{\gamma_t}{2}\mathbb{E}[\|\delta_t\|_2^2] + \frac{L_\Phi\gamma_t^2}{2}M^2.$$

Summing over $t=1,\dots,T$, lower bounding by $\Phi_{\eta,\inf}$, and using

$$\|\delta_t\|_2^2 \le 2\varepsilon_{\text{bias}}^2 + 2L_{\theta\mu}^2L^2E^2e_t^2$$

gives

$$\frac{\sum_{t=1}^{T}\gamma_t\mathbb{E}\left[\|\nabla\Phi_\eta(\theta^{(t)})\|_2^2\right]}{\sum_{t=1}^{T}\gamma_t} \le \frac{2(\Phi_\eta(\theta^{(1)}) - \Phi_{\eta,\inf})}{\sum_{t=1}^{T}\gamma_t} + L_\Phi M^2\frac{\sum_{t=1}^{T}\gamma_t^2}{\sum_{t=1}^{T}\gamma_t} + 2\varepsilon_{\text{bias}}^2 + 2L_{\theta\mu}^2L^2E^2\frac{\sum_{t=1}^{T}\gamma_t\mathbb{E}[e_t^2]}{\sum_{t=1}^{T}\gamma_t}.$$

Substituting the tracking bound from Lemma A.3 yields

$$\frac{\sum_{t=1}^{T}\gamma_t\mathbb{E}\left[\|\nabla\Phi_\eta(\theta^{(t)})\|_2^2\right]}{\sum_{t=1}^{T}\gamma_t} \le \widetilde{\mathcal{O}}(T^{-1/2}) + \mathcal{O}\left(\varepsilon_{\text{bias}}^2 + L_{\theta\mu}^2L^2E^2\eta^2\varepsilon_{\text{ema},T}^2\right),$$

because for $\gamma_t = \gamma_0/\sqrt{t}$,

$$\sum_{t=1}^{T}\gamma_t = \Theta(\sqrt{T}), \qquad \sum_{t=1}^{T}\gamma_t^2 = \Theta(\log T), \qquad \sum_{t=1}^{T}\gamma_t^3 = \mathcal{O}(1).$$

This proves the theorem. □

B Expert utilization per layer

We track the Coefficient of Variation (CV) of expert routing weights across training steps for individual layers. The Baseline model exhibits severe transient load instability, characterized by sharp CV spikes in early layers (Layers 2–4) midway through training. In contrast, DRMoET effectively stabilizes the routing dynamics, maintaining a consistent load distribution throughout the entire training trajectory and preventing the early-layer load collapse observed in the baseline.

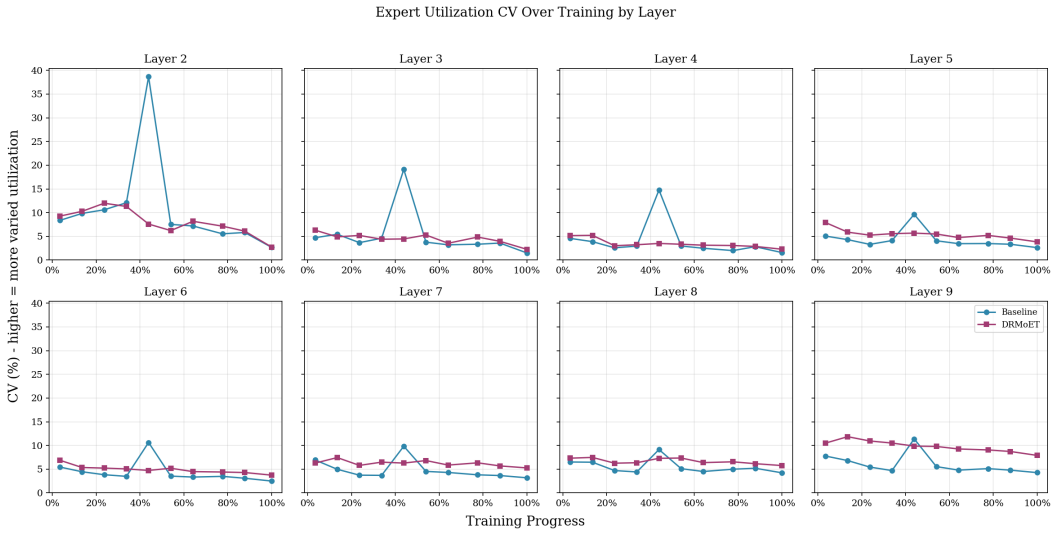


Figure 2: **Per-layer expert utilization variability over training.** DRMoET stabilizes routing dynamics in early layers and avoids the transient CV spikes observed in the baseline.

C Expert Utilization CV Over Training

Step	Baseline CV	DRMoET CV	Diff
540	2.75%	3.32%	+0.57%
2160	2.46%	3.04%	+0.58%
3780	1.78%	2.73%	+0.95%
5400	2.00%	2.64%	+0.64%
7020	6.47%	2.21%	-4.26%
8640	1.60%	2.07%	+0.46%
10260	1.32%	2.07%	+0.75%
12420	1.43%	2.37%	+0.94%
14040	1.44%	2.04%	+0.59%
16000	0.91%	1.58%	+0.67%
Average	**2.22%**	**2.41%**	

Table 8: **Expert utilization CV over training** (six routed experts plus two shared experts). Higher CV indicates greater relative variation in load across experts. DRMoET shows slightly higher average CV but avoids the baseline's transient mid-training imbalance spike (step 7k), consistent with controlled exploration early and modest specialization later.

D Coactivation Concentration Metrics

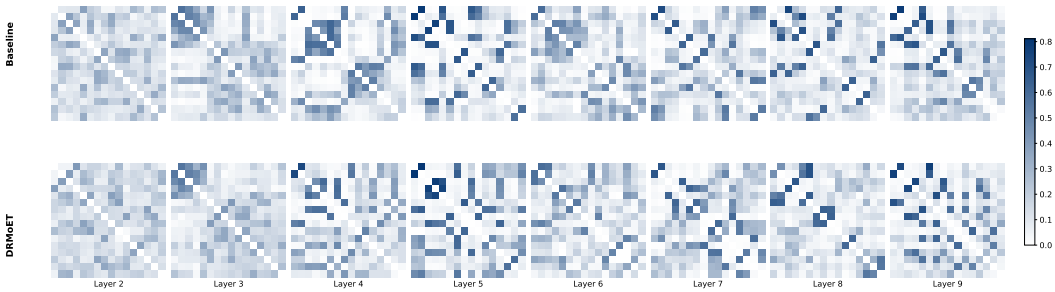


Figure 3: **Expert co-activation heatmaps** (layer-wise) comparing baseline vs. DRMoET. DRMoET produces more uniform co-activation in early/mid layers (L2–L5) and slightly more concentrated patterns in later layers (L6–L9).

Layer	Variance			Gini			Top-3 Conc.		
	Baseline	DRMoET	Δ	Baseline	DRMoET	Δ	Baseline	DRMoET	Δ
2	0.00518	0.00483	-6.6%	0.248	0.238	-4.2%	17.7%	17.5%	-1.2%
3	0.00800	0.00678	-15.3%	0.301	0.275	-8.4%	19.7%	19.3%	-2.4%
4	0.01534	0.01395	-9.0%	0.413	0.393	-4.9%	24.3%	23.8%	-2.4%
5	0.02487	0.02188	-12.0%	0.500	0.475	-5.0%	30.7%	28.7%	-6.4%
6	0.01477	0.01549	+4.9%	0.422	0.427	+1.1%	24.0%	25.1%	+4.3%
7	0.01694	0.01719	+1.5%	0.430	0.438	+1.9%	26.3%	26.0%	-1.2%
8	0.01886	0.01862	-1.2%	0.443	0.451	+1.6%	28.4%	27.7%	-2.4%
9	0.01902	0.02105	+10.7%	0.451	0.465	+3.2%	27.7%	29.4%	+5.9%

Table 9: **Coactivation concentration metrics (DRMoET vs. Baseline).** Negative Δ = more uniform partnerships under DRMoET; positive Δ = increased specialization.

E Interaction with the standard load-balancing loss

At the 290M-active/746M-total scale, we remove the standard load-balancing loss from DRMoET while keeping the remaining reported training configuration unchanged. Table 10 uses the same seven-task evaluation as the main results. The no-balancing DRMoET configuration reaches 0.5448, exceeding standard FLAME-MoE's 0.5403 on the average and on five tasks. DRMoET with balancing remains stronger on the average at 0.5523.

Method	**ARC-C**	**ARC-E**	**HellaSwag**	**PIQA**	**WinoGrande**	**SciQ**	**ReCoRD***	**Average**
FLAME-MoE (with balancing)	0.2329	0.5295	0.3447	0.6697	0.5012	0.8020	0.7023	0.5403
DRMoET without balancing	**0.2389**	0.5539	0.3430	0.6877	**0.5067**	0.7770	0.7066	0.5448
DRMoET with balancing	0.2295	**0.5682**	**0.3452**	**0.6888**	0.5051	**0.8190**	**0.7105**	**0.5523**

Table 10: **Load-balancing interaction at 290M-active/746M-total scale.** Averages are over seven tasks. Standard FLAME-MoE includes balancing; a FLAME-MoE-without-balancing control is not reported. *ReCoRD is F1.

The robust objective is defined on expert outcomes that occur under the router. It does not require a separate balancing penalty to be evaluated, but cannot by itself ensure adequate observations for experts that receive almost no traffic. Allocation and observed competence therefore remain distinct concerns. The results support using the two objectives together in this recipe; they do not establish that DRMoET replaces balancing or quantify its incremental effect relative to an unreported no-balancing baseline.

F Implementation Details

Training-token budgets. The `FLAME-MoE-290M-746M` experiments use approximately 33.5B training tokens. The `FLAME-MoE-1.7B-10.3B` comparison uses approximately 67B tokens. Within each scale and budget, the compared methods use the same architecture, data mixture, and training-token count.

We used 1×10^{-3} for both auxiliary loss, and z-loss as training hyperparameters that all runs share.

Data. The `FLAME-MoE-290M-746M` experiments use the same data the original Kang et al. [2025] paper has used. For the `FLAME-MoE-1.7B-10.3B` run, due to the increment on #tokens trained, we downloaded and processed more chunks of DCLM to guarantee we stick to the 1-epoch only training setup, which is commonly used while training LLMs.

NeurIPS Paper Checklist

(A1) **Claims**

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The abstract and introduction explained how does DRMoET work, and to what scope it answers the problem the paper is trying to address.

Guidelines:

- The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

(A2) **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The paper mentioned some of the limitations, such as lack of larger scale models, different expert counts, and also train time analysis on OOD data.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

(A3) **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: For the proof of convergence, the whole proof is included in the appendix. The assumptions, and lemma, were provided in detail.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

(A4) **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Training configurations such as the pipeline used, what is the configuration of the model, what are the hyper-parameter used, etc. The detailed pseudocode was provided to ensure reproducibility.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

(A5) **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code is provided as supplementary material. The entire pipeline is included.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

(A6) **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The dataset used, and the reason we chose it is explained. We also provided the hyperparameters we used, as well as the optimizer chosen.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

(A7) **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: To ensure the significance of our method, we tested it across several hyperparameter settings, and in all cases, it outperforms the baseline. This show that our method's gain is consistent.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

(A8) **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We did mention the GPU we use, as well as how many steps were trained. The two together can provide a full picture of the cost for training one model in our setting.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

(A9) **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: All experiments done are on the GPU, without causing social harm.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

(A10) **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The paper introduces a pretraining algorithm, which is irrelevant to broader societal impact.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

(A11) **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not provide data/model that have a high risk for misuse. The training data is from DCLM, which is a public-available language dataset, that is safe to use/release.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

(A12) **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All previous related work is cited in the paper. We also mentioned explicitly what code base did we build our code base on top of.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

(A13) **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: No new assets was release from the paper.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

(A14) **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

(A15) **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: the paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

(A16) **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: the core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.