A Theoretical Framework for Auxiliary-Loss-Free Load-Balancing of Sparse Mixture-of-Experts in Large-Scale AI Models

X.Y. Han
Chicago Booth

Yuan Zhong
Chicago Booth

Abstract

In large-scale AI training, Sparse Mixture-of-Experts (s-MoE) layers enable scaling by activating only a small subset of experts per token. An operational challenge in this design is load-balancing: routing tokens to minimize the number of idle experts, which is important for the efficient utilization of (costly) GPUs. We provide a theoretical framework for analyzing the Auxiliary-Loss-Free Load Balancing (ALF-LB) procedure — proposed by DeepSeek's Wang et al. [2024] — by casting it as a one-step-per-iteration primal-dual method for an assignment problem. First, in a stylized deterministic setting, our framework yields several insightful structural properties: (i) a monotonic improvement of a Lagrangian objective, (ii) a preference rule that moves tokens from overloaded to underloaded experts, and (iii) an approximate-balancing guarantee. Then, we incorporate the stochastic and dynamic nature of AI training using a generalized online optimization formulation. In the online setting, we derive a strong convexity property of the objective that leads to a logarithmic regret bound under certain step-size choices. Additionally, we present real experiments on 1B-parameter DeepSeekMoE models to complement our theoretical findings. Together, these results build a principled framework for analyzing the auxiliary-loss-free load-balancing of s-MoE in AI models.

1 Introduction: s-MoEs in Large-Scale AI Training

Scaling laws continue to reward larger models [Kaplan et al., 2020, Hoffmann et al., 2022], but compute, energy, and hardware constraints [Strubell et al., 2019, Thompson et al., 2020, Sevilla et al., 2022] limit dense scaling. Sparse Mixture-of-Experts (s-MoE) layers [Shazeer et al., 2017] address these challenges by routing each token through only $K \ll E$ experts, substantially increasing parameter counts without proportional compute. As a testament to s-MoEs' utility, recent releases of OpenAI's GPT [Achiam et al., 2023], Google's Gemini [Team et al., 2024], and DeepSeek [DeepSeek-AI, 2025] have all leveraged s-MoE designs to improve efficiency and maintain performance scaling.

One design challenge in s-MoE training is *load-balancing*, which aims to ensure that per-iteration token assignments are sufficiently even across experts to avoid idling and stragglers. As surveyed in Wang et al. [2024], the traditional approach of using auxiliary balancing losses [Shazeer et al., 2017, Lepikhin et al., 2021, Fedus et al., 2022] may interfere with the optimization of the main objective. To address this issue, DeepSeek's Auxiliary-Loss-Free Load Balancing (ALF-LB) [Wang et al., 2024] takes a different path by learning expert-specific biases updated once per iteration, outside of the task gradient flow. Notably, ALF-LB was used to successfully train the recent DeepSeekV3 [DeepSeek-AI, 2024] models.

1.1 Naïve s-MoE Layers Without Load-Balancing

Figure 1 shows the naïve-form variant of a s-MoE layer with sparse gating and expert routing. Let $x_1, \ldots, x_T$ be the token embeddings (the context). A layer produces features z_i that enter an s-MoE with E experts. The affinity score between token i and expert k is given by $\zeta_{i,k} := w_k^\top z_i$, and the gate selects the Top-K experts based on the K largest affinity scores $\zeta_{i,k}$. The selected experts' outputs for token i are aggregated as $\sum_{k \in \text{ChosenExperts}_i} \gamma_{i,k} f_k(z_i)$, where $\gamma_{i,k} := \mathrm{SoftMax}(\zeta_{i,k}; \{\zeta_{i,k'}\})$.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: ML×OR.

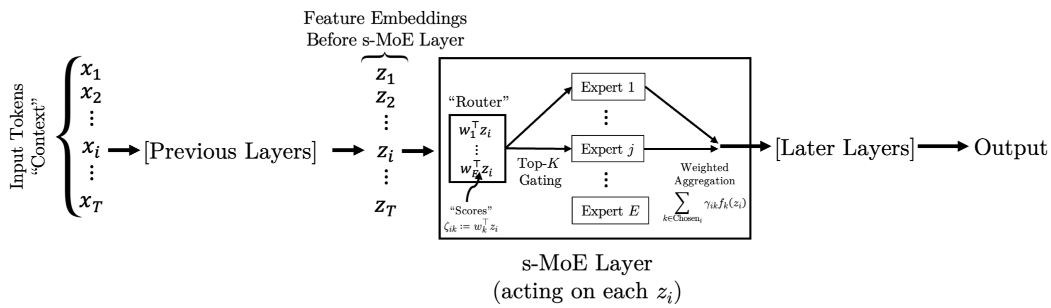


Figure 1: Schematic of a naïve s-MoE layer without load-balancing.

1.2 Load-Balancing and DeepSeek's ALF-LB

The naïve routing in Figure 1 can create overloading and underloading patterns in the MoE layers, wasting expensive computational resources. To address this problem, DeepSeek's auxiliary-loss-free (ALF-LB) [Wang et al., 2024] procedure augments each expert with a bias p_k updated once per iteration, nudging tokens toward underloaded experts — without interfering with training gradients as is done in works leveraging auxiliary balancing losses [Shazeer et al., 2017, Lepikhin et al., 2021, Fedus et al., 2022]. For some fixed constant u, the ALF-LB update is

$$p_k \leftarrow \begin{cases} p_k - u & \text{if expert } k \text{ had load } > L, \\ p_k + u & \text{if expert } k \text{ had load } < L, \\ p_k & \text{otherwise,} \end{cases} \quad \textbf{(DeepSeek ALF-LB Procedure)} \tag{1}$$

which decreases the biases of overloaded experts and increases those of underloaded experts. Tokens are then routed by applying the Top-K rule to the adjusted scores $\gamma_{ik} + p_k$. We next formalize this heuristic as a primal-dual procedure and analyze both deterministic and stochastic regimes.

2 A Primal-Dual Framework for Load-Balancing

We formalize ALF-LB as a one-step-per-iteration primal-dual method. Consider assigning T tokens to E experts with sparsity K and target load $L = KT/E$. The exact-balancing problem is

$$\begin{aligned} &\max_{\{x_{ik}\}} \sum_{i,k} \gamma_{ik} x_{ik} \\ &\text{s.t.} \sum_k x_{ik} = K \ \ \forall i, \qquad \sum_i x_{ik} = L \ \ \forall k, \qquad x_{ik} \in \{0,1\}. \end{aligned} \tag{2}$$

Relaxing $x_{ik} \in \{0,1\}$ to $x_{ik} \geq 0$ preserves the optimum. The Lagrangian is

$$\begin{aligned} \mathcal{L}(x,y,p) &= \sum_{i,k} \gamma_{ik} x_{ik} + \sum_i y_i \Big(K - \sum_k x_{ik} \Big) + \sum_k p_k \Big(\sum_i x_{ik} - L \Big) \\ &= \sum_{i,k} (\gamma_{ik} + p_k - y_i) x_{ik} + K \sum_i y_i - L \sum_k p_k. \end{aligned} \tag{3}$$

To solve this, first initialize $p_k \leftarrow 0$. Then, for iteration n, perform the following primal-dual updates:

$$\textbf{Dual: } p_k^{(n+1)} \leftarrow p_k^{(n)} + \epsilon_k^{(n)} \Big(L - \sum_i x_{ik}^{(n)} \Big) \qquad \forall\, k, \tag{4}$$

$$\textbf{Primal: } x_{ik}^{(n+1)} \leftarrow \begin{cases} 1 & \text{if } k \in \text{TopK}_{k'} \big(\gamma_{ik'}^{(n+1)} + p_{k'}^{(n+1)} \big) \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, i,k. \tag{5}$$

Instantiating the dual update (4) with the ALF-LB rule in (1) is equivalent to the step-size choice

$$\epsilon_k^{(n)} = \frac{u}{\big| L - \sum_i x_{ik}^{(n)} \big|}. \quad \textbf{(DeepSeek ALF-LB Step-Size)} \tag{6}$$

As an experimental connection to practice, Figure 2 compares the convergence and imbalance behaviors of training real 1B-parameter DeepSeekMoE models [Dai et al., 2024] with varying $\epsilon_k^{(n)}$ as well as using an auxiliary loss [Shazeer et al., 2017, Lepikhin et al., 2021, Fedus et al., 2022].

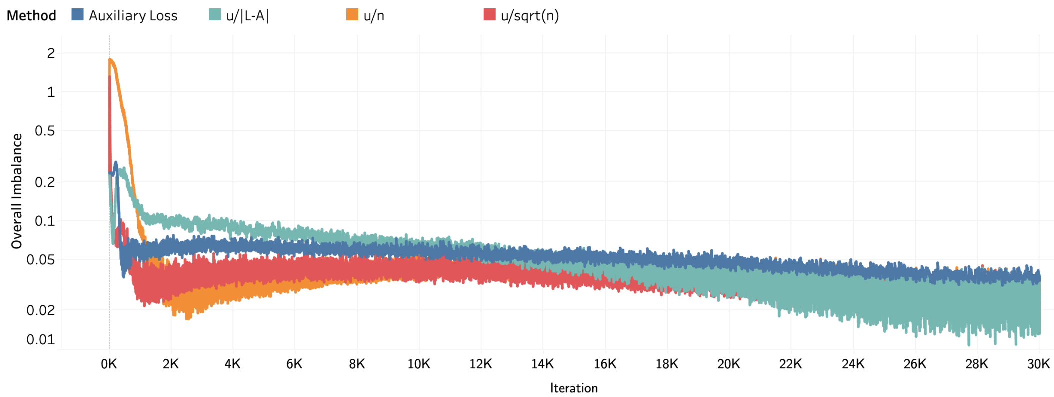


Figure 2: Overall load imbalance during the training of 1B-parameter DeepSeekMoE models [Dai et al., 2024] trained for 30K steps with different step-size choices. We use $E = 64$ experts with $K = 6$ sparsity. Architecture is the same as that in Wang et al. [2024]. Each model was trained on 8xH100/H200 GPUs with batch size 64 and 4096 tokens/batch (so, $T \approx 262K$). We use the AdamW [Loshchilov and Hutter, 2019] optimizer with cosine learning rate decay ($1e{-4}$ to $1e{-5}$) and 750 warmup steps. In the legend, $u = 0.001$ and n is the iteration number.

3 Deterministic Convergence Guarantees

For simplicity, we consider the case where $K = 1$ for Sections 3 and 4. At iteration n, denote the realized load of expert k as $A_k^{(n)} := \sum_i x_{ik}^{(n)}$ and the expert that token i is routed to as $\alpha_n(i) := \arg\max_{k'}(\gamma_{ik'} + p_{k'}^{(n)})$. Define the switching benefit

$$b^{(n+1)}(i) = \left(\gamma_{i\alpha_{n+1}(i)} + p_{\alpha_{n+1}(i)}^{(n+1)}\right) - \left(\gamma_{i\alpha_n(i)} + p_{\alpha_n(i)}^{(n+1)}\right). \tag{7}$$

Theorem 1. ***(Change in Lagrangian)*** *For updates (4)–(5),*

$$\mathcal{L}\left(x^{(n+1)}, p^{(n+1)}\right) - \mathcal{L}\left(x^{(n)}, p^{(n)}\right) = \sum_i b^{(n+1)}(i) - \sum_k \epsilon_k^{(n)} \left(A_k^{(n)} - L\right)^2.$$

Then, the gain decomposes into the sum of token-level benefits with an imbalance penalty:

Corollary 2. *With $\epsilon_k^{(n)} = u/|L - A_k^{(n)}|$,*

$$\mathcal{L}\left(x^{(n+1)}, p^{(n+1)}\right) - \mathcal{L}\left(x^{(n)}, p^{(n)}\right) = \sum_i b^{(n+1)}(i) - u \sum_k |A_k^{(n)} - L|.$$

Let $\mathcal{S}^{(n+1)}$ be the set of tokens that switch experts at iteration $n+1$. Then,

$$\mathcal{L}\left(x^{(n+1)}, p^{(n+1)}\right) - \mathcal{L}\left(x^{(n)}, p^{(n)}\right) < u\left[2|\mathcal{S}^{(n+1)}| - \sum_k |A_k^{(n)} - L|\right].$$

If the imbalanced partition does not flip between iterations, the Lagrangian strictly decreases:

Theorem 3. *If the sets of overloaded and underloaded experts stay the same between iterations n and $n{+}1$, then*

$$\mathcal{L}\left(x^{(n+1)}, p^{(n+1)}\right) - \mathcal{L}\left(x^{(n)}, p^{(n)}\right) < 0.$$

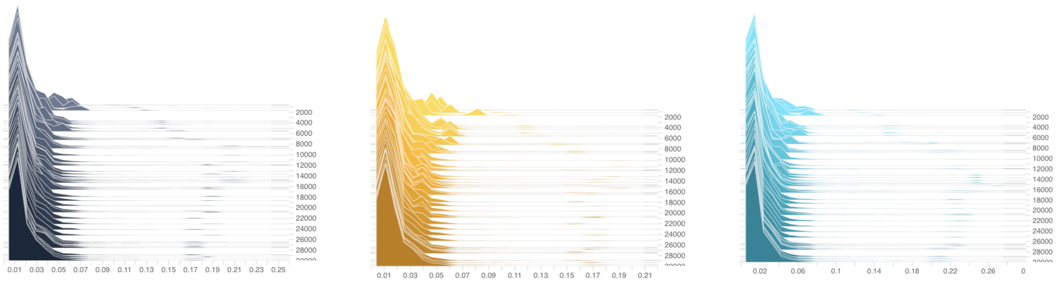
The DeepSeek step-size (6) also enforces a strict movement preference and bounds changes:

Theorem 4. *Assume the updates (4)–(5) with $\epsilon_k^{(n)}{=}u/|L - A_k^{(n)}|$, that token i switched experts between n and $n{+}1$, and that there are no ties. Then, i moves down the ordering Overloaded≻Balanced≻Underloaded, $0{<}b_i^{(n+1)}{<}2u$, and $-2u{<}\gamma_{i\alpha_{n+1}(i)}{+}p_{\alpha_{n+1}(i)}^{(n)}{-}(\gamma_{i\alpha_n(i)}{+}p_{\alpha_n(i)}^{(n)}){<}0$.*

Theorem 5. *Under the assumptions of Theorem 4, token moves are unique and each expert's load changes by at most $(E{-}1)$ per step.*

Theorem 6. ***(Guarantee of Approximate Balancing)*** *Under the assumptions of Theorem 4 — for sufficiently small, constant step-size u — the loads of all experts converge to the range $[L-(E-1), L+(E-1)]$. Moreover, once an expert's load enters that range, it remains there.*

When $T \gg E$, the $(E-1)$ deviation from $L = KT/E$ is negligible, which aligns with the observed robust performance of ALF-LB by Wang et al. [2024], Dai et al. [2024].



(a) $u/|L - A_k^{(n)}|$ Step-Size (b) u/n Step-Size (c) $u/\sqrt{n}$ Step-Size

Figure 3: Time-lapse histograms of the marginal distributions of $\gamma_{ik}^{(n)}$ during the training of 1B-parameter DeepSeekMoE models described in Figure 2.

4 Stochastic Analysis via Online Optimization

In practice, $\gamma_{ik}^{(n)}$ evolves every iteration. (See Figure 3 for experimental histograms of $\gamma_{ik}^{(n)}$ when training 1B-parameter DeepSeekMoE models.) Thus, we now assume $\gamma_{ik}^{(n)}$ are *stochastic* and drawn from expert-dependent distributions Γ_k with support on $(0, 1)$. For simplicity, for a fixed k, we assume the draws are independent across i and n. We analyze the *online objective*

$$f^{(n)}(p) = \sum_{i=1}^{T} \max_{k'} \{\gamma_{ik'}^{(n)} + p_{k'}\} - L \sum_{k=1}^{E} p_k, \tag{8}$$

whose gradient component is $\nabla_k f^{(n)}(p) = A_k^{(n)}(p) - L$. Next, note that the load-balancing router's decision is invariant to constant shifts to $p_k^{(n)}$. Thus, we can assume that all $p_k^{(n)}$ and updates lie in $\mathcal{K} = \{z : \sum_k z_k = 0\}$. Equivalently, we can generalize the dual updates to take the projected form

$$p^{(n+1)} \leftarrow \text{Proj}_{\mathcal{K}}\left(p^{(n)} - \epsilon^{(n)} \nabla f^{(n)}(p^{(n)})\right), , \tag{9}$$

which is just a computationally-negligible component-wise mean subtraction. Moreover, from experiments not reported here, we found that the range of the biases $\max_j p_j^{(n)} - \min_j p_j^{(n)}$ is typically smaller than 1 during DeepSeekMoE-1B training without explicit enforcement. Thus, we add this into our assumptions. Then, under this setting, the expected objective is strongly convex.

Lemma 7. ***(Strong Convexity)*** *Let $\mathbf{F}(p) := \mathbb{E}[\max_k\{\Gamma_k + p_k\}]$ for independent continuous Γ_k. For any direction δ, the second directional derivative satisfies*

$$D^2\mathbf{F}(p)[\delta, \delta] = \sum_{k<\ell} w_{k\ell}(p)(\delta_k - \delta_\ell)^2, \tag{10}$$

with nonnegative weights $w_{k\ell}(p)$ depending on the distributions of the Γ_k. If $\max_j p_j - \min_j p_j < 1 - \kappa$ for some $\kappa > 0$, then for $\delta \in \mathcal{K}$

$$\delta^\top \nabla^2 \mathbf{F}(p)\, \delta \geq c_\Gamma\, E\, \|\delta\|^2, \qquad \mu := T c_\Gamma E, \tag{11}$$

where $c_\Gamma > 0$ is a positive constant dependent on the distributions $\{\Gamma_k\}$. Hence, $\mathbf{f}(p) = T\mathbf{F}(p) - L\sum_k p_k$ is μ-strongly convex on $\mathcal{K}$.

This lemma leads to a logarithmic bound on the regret $R_N = \sum_{n=1}^{N}(f^{(n)}(p^{(n)}) - f^{(n)}(p^*))$.

Theorem 8. ***(Logarithmic Regret)*** *Consider the update (9) run for N iterations with $\epsilon^{(n)} = 1/(\mu n)$. Then, there exists a constant $C_\mu^{T,E}$ independent of N such that*

$$\mathbb{E}[R_N] \;\leq\; C_\mu^{T,E}\,(1 + \ln N).$$

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, and Shyamal Anadkat. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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

DeepSeek-AI. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Hendricks, Johannes Welbl, and Aidan Clark. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, Zhifeng Chen, and Yonghui Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Jaime Sevilla, Lasse Heim, Amanda Askell Ho, Noah Buchan, Alex Snell, Maruan Alhussein, Natasha Jaques McAleese, William Biles, Kevin McKee, and Joey Leung. Compute trends across three eras of machine learning. *arXiv preprint arXiv:2202.05924*, 2022.

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. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

E Strubell, A Ganesh, and A McCallum. Energy and policy considerations for deep learning in nlp. proceedings of the 57th annual meeting of the association for computational linguistics (acl). *Stroudsburg, PA, USA. Association for Computational Linguistics*, 2019.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, and Shibo Wang. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Neil Thompson, Kristjan Greenewald, Keeheon Lee, and Gustavo F. Manso. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*, 2020.

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.