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

X.Y. Han*
Chicago Booth
XY.Han@chicagobooth.edu

Yuan Zhong*
Chicago Booth
Yuan.Zhong@chicagobooth.edu

April 24, 2026

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 and for the thorough training of architecture parameters across all experts. 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 primal-dual method using a single-shot, constant-time update per training iteration for solving an assignment problem. First, in a stylized deterministic setting, our framework yields several insightful structural properties: (i) a monotonic improvement condition for the 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 expected 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 and Load Balancing in AI Training

Scaling architecture size has been the dominant driver of modern AI performance, with larger models consistently achieving better results (Kaplan et al., 2020; Hoffmann et al., 2022; Epoch AI, 2023). However, AI development has reached the point where the scaling of computation becomes

*Authors listed alphabetically.

prohibitively expensive due to hardware and energy constraints (Strubell et al., 2019; Thompson et al., 2020; Sevilla et al., 2022). Adapting to the cost and hardware limitations of scaling, researchers have turned to sparse Mixture-of-Experts (s-MoE) architectures (Shazeer et al., 2017), which are sparse realizations within the mixture-of-experts (MoE) paradigm codified by Jacobs et al. (1991).

In modern large-scale AI architectures, s-MoE layers — consisting of several parallel subnetworks ("experts") controlled by a "sparse gate" or *router* that selects data to route to them — have largely replaced single submodules through which all data must pass. In these s-MoEs, for each input, the sparse-gating component selects a strict subset of experts (hence "sparse") to apply to that input. Thus, only a small subcomponent of an AI architecture is activated to process each piece of input data — allowing models to have significantly more parameters while keeping inference and training costs manageable. As a testament to s-MoEs' utility, recent releases of OpenAI's GPT (Achiam et al., 2023), Google's Gemini (Gemini Team et al., 2024), and DeepSeek (DeepSeek-AI et al., 2024; DeepSeek-AI, 2026) have all leveraged s-MoE designs to improve efficiency and maintain performance scaling.

However, a crucial aspect of s-MoE design — load balancing (controlling "how many inputs per expert") — is mostly developed using trial-and-error motivated by heuristic insights (see Section 1.2). Learning to precisely and mathematically balance the load across experts, which reduces monetary losses from idle GPUs, could lead to enormous monetary savings for AI training.

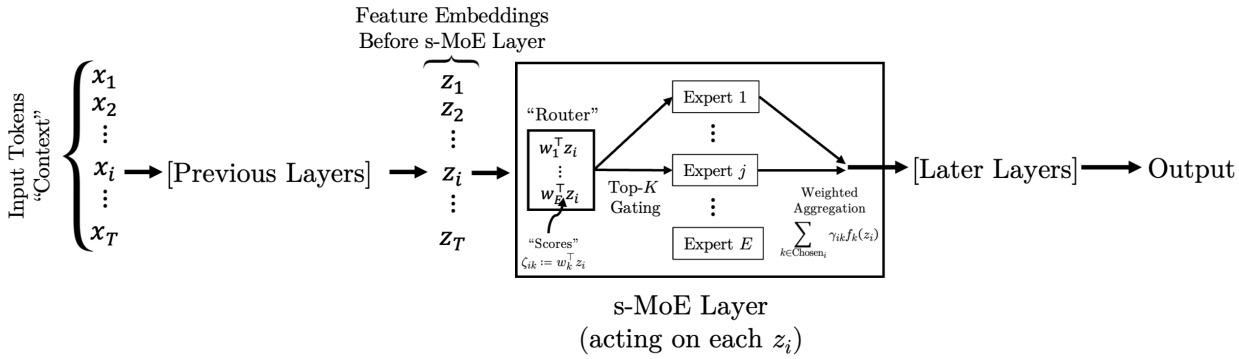


Figure 1: Schematic of a naïve s-MoE layer without load balancing.

1.1 Naïve s-MoE Layers Without Load Balancing

Figure 1 describes the "naïve" setup for s-MoE layers within transformer-based AI models. In particular, the input is a series of *token embeddings* $x_1, x_2, ..., x_T$ where each x_i is a high-dimensional vector corresponding (in language models) to a language unit such as "Hel", "lo", "world", etc. or (in vision models) a patch within an image. Each piece of input data (a sentence, an image patch, etc.) is decomposed into constituent tokens; each token is mapped to its vector embedding x_i; and those embeddings are input into the AI model. The entire tuple of vectors $\{x_i\}_{i=1}^T$ is called the *context* and T is the *context length*.

Within the AI model, each of the original token embeddings x_i is transformed into *feature embeddings* by each of the AI model's layers. In Figure 1, to describe the action of some particular s-MoE layer, we use $\{z_i\}_{i=1}^T$ to denote the feature embeddings before that s-MoE layer.

When a feature embedding "enters" an s-MoE layer with E experts, we calculate an unnormalized affinity score $\zeta_{i,k}$ between z_i and the k-th expert — usually using an inner product: $\zeta_{i,k} := w_k^\top z_i$. These scores are then normalized, typically using the "softmax" function, into the *affinity scores*:

$$\gamma_{i,k} := \text{SoftMax}\left(\zeta_{i,k};\ \{\zeta_{i,k'}\}_{k'=1}^E\right) = \frac{\exp(\zeta_{i,k})}{\sum_{k'=1}^E \exp(\zeta_{i,k'})} \tag{1}$$

The router then selects the Top-K experts based on the K largest $\gamma_{i,k}$. The final step in an s-MoE layer is to aggregate the outputs of the selected experts. This is done by computing a weighted sum of the selected experts' outputs:

$$\sum_{k \in \text{ChosenExperts}_i} \gamma_{i,k} f_k(z_i), \tag{2}$$

where f_k represents the k-th expert. Note that the softmax is taken *before* the Top-K selection, which is typically the preferred order in recent s-MoEs (Dai et al., 2024; Riquelme et al., 2021). Moreover, the softmax is monotonic, so it is equivalent to choose the Top-K experts based on the K largest $\{\gamma_{i,k}\}_{k=1}^E$ for each i, where $K < E$ for s-MoEs. This completes the description of the schematic in Figure 1.

1.2 Load Balancing of Experts: Background and Related Work

While conceptually simple, the naïve routing method of choosing the top-K experts based only on $\{\gamma_{i,k}\}_{k=1}^E$ often causes load imbalance when assigning tokens to experts. This creates critical issues for both deployment and training. During deployment, imbalance results in underutilized GPUs hosting idling experts, which wastes costly computing resources. Although one might try to balance GPU usage across *multiple requests* in high-traffic deployment scenarios (see further discussion in Section 1.4.1), across-requests balancing operate in the deployment stage (not the training stage) and remain vulnerable to inefficiencies when traffic is low and expert affinities are skewed. Moreover, during training, such imbalance prevents effective learning across network parameters across all s-MoE layer experts: This is due to the uneven training of the experts that, in turn, induces a self-perpetuating cycle where the router preferentially selects better-trained experts while others become more underutilized and undertrained. Since every expert must be hosted on costly GPUs during training such imbalance could lead to significant monetary losses during training as well.

Several fixes have been proposed. The most commonly adopted approach is adding an auxiliary "balancing loss" directly to the training loss penalizing the network parameters during training for inducing imbalanced token allocations (Fedus et al., 2022; Lepikhin et al., 2021; Shazeer et al., 2017). However, as discussed in Wang et al. (2024, Section 2.2), this method interferes with the gradient updates of the performance-focused component of the objective.

Another approach by Lewis et al. (2021) approximately solves — via a truncated auction

heuristic based on Bertsekas (1992) — an integer program that balances the load across experts in every training iteration. However, generating an AI model's outputs for even one single batch of data (a "forward pass") requires significant computation time and memory since it requires calculating matrix multiplications and non-linear transformations defined by millions to billions of parameters. During training (as opposed to inference/deployment), there is an additional computational and memory overhead for computing and storing the backpropagated gradients (the "backward pass"). Thus, it is inadvisable to spend additional time solving a multi-iterative subroutine (whether an auction algorithm or an integer program) for every s-MoE layer and every batch.

To address this problem, DeepSeek's auxiliary-loss-free (ALF-LB) (Wang et al., 2024) procedure augments each expert with a bias p_k using a *single-shot* update (as opposed to a multi-step subroutine), nudging tokens toward underloaded experts — without interfering with training gradients as is the case when using auxiliary balancing losses[1]. Notably, ALF-LB was used to successfully train the recent DeepSeekV3 (DeepSeek-AI et al., 2024) and DeepSeekV4 (DeepSeek-AI, 2026) models.

1.3 DeepSeek's ALF-LB Algorithm

DeepSeek's ALF-LB procedure (Wang et al., 2024) is as follows:

1. For each expert $k = 1, \ldots, E$, initialize a scalar shift parameter p_k to be 0.
2. Perform a forward pass on a batch. During the forward pass, route token i based on the experts with the highest shifted weights $\gamma_{ik} + p_k$.
3. Calculate the downstream network loss and update the main network parameters, treating the shifts $\{p_k\}$ as constants.
4. For each expert k, update its shift parameter as follows, where u is a small constant (e.g., 0.001):

$$p_k \leftarrow \begin{cases} p_k - u & \text{if expert } k \text{ had load } > L; \\ p_k + u & \text{if expert } k \text{ had load } < L; \\ p_k & \text{otherwise.} \end{cases} \tag{3}$$

5. Repeat steps 2-4 for each batch of input data.

In the original publication, Wang et al. (2024) chose $u = 0.001$ and exhibited empirical benefits of this procedure on 1B to 3B parameter DeepSeekMoE models (Dai et al., 2024). Notably, this procedure was subsequently used to train the DeepSeekV3 (DeepSeek-AI et al., 2024) and DeepSeekV4 (DeepSeek-AI, 2026) base models.

1.4 Contributions and Organization of Paper

Our main contribution is a rigorous theoretical framework for understanding and analyzing the ALF-LB procedure, with specific contributions detailed across different sections. First, in Section

[1]Specifically, the p_k biases in ALF-LB are considered constants during the backpropagation phase of AI training.

2, we cast the ALF-LB procedure as a single-step primal-dual method for an assignment problem, connecting a state-of-the-art heuristic from large-scale AI to the operations research and primal-dual optimization literature for resource allocation such as those in Bertsekas (1992, 1998, 2008). However, the procedure we analyze differs from the aforementioned operations research problems since, as discussed in Section 1.2, the computational and memory requirements of performing a forward pass through an AI model do not allow for one to run multi-iterative procedures as subroutines with those forward passes. Instead, s-MoE balancing routines (such as ALF-LB) must be updated in a "single-shot" manner — with computationally-minimal, constant-time updates per forward pass — instead of relying on multi-iterative subroutines.

Then, in Section 4, we analyze this procedure in a stylized deterministic setting and establish several insightful structural properties: (i) a monotonic improvement condition for the Lagrangian objective (Theorem 1), (ii) a preference rule that moves tokens from overloaded to underloaded experts (Theorem 5), and (iii) a band-stability guarantee showing that once expert loads enter an approximate-balance band, they remain there (Theorem 9). Finally, in Section 5, we extend our analysis to a more realistic online, stochastic setting by establishing a strong convexity property of the expected dual objective (Section 5.6) and using it to derive a logarithmic regret bound for the ALF-LB procedure (Theorem 17).

1.4.1 Online Resource Allocation: Connections and Related Works

It is insightful to compare this paper to another recent line of work at the intersection of AI implementation and operations research: the online resource allocation of multiple requests/queries in AI datacenters (see Zhang et al. (2024); Markovic-Voronov et al. (2026) and citations therein) where many computational requests arrive in an online, stochastic manner and must be optimally routed to a server in the datacenter to complete the job. In comparison, in the s-MoE balancing problem, for each forward pass, every individual s-MoE layer must process batches of tokens that *all arrive at once*. During a forward pass through a multi-layered s-MoE based AI architecture, each s-MoE layer within the architecture must wait for all parallel experts in the previous s-MoE layer to complete their forward passes before it can proceed. For reference, DeepSeekV3 (DeepSeek-AI et al., 2024) and DeepSeekV4 (DeepSeek-AI, 2026) both contain 61 s-MoE layers. Thus, unlike the routing of requests to datacenter servers, the allocation of tokens in an s-MoE layer must be conducted in a "single-shot", computationally-minimal manner in order to not delay the sequential progression of the forward pass through the multi-layered AI architecture itself.

Another related line of works is Balseiro et al. (2020, 2021); Agrawal and Devanur (2014); Jenatton et al. (2016) (see Balseiro et al. (2021, Section 1.2) and citations therein) that design and analyze primal-dual methods for solving online resource allocation problems by formulating them as online stochastic convex programs or regularized allocation problems. These prior works utilize dual descent and mirror descent techniques to manage global resource constraints which can be formulated as load balancing. However, the algorithms proposed in those works often require solving auxiliary optimization sub-routines such as linear programs, quadratic programs, or non-

trivial projections during their updates. As discussed earlier, in the context of s-MoE training, multi-iterative subroutines are computationally impracticable because routing must occur in every s-MoE layer of the AI architecture during already-computationally-expensive forward passes, which does not allow for the extra overhead of solving auxiliary sub-routines at every s-MoE layer. Thus, in comparison, our paper instead analyzes a "single-shot" update framework, built specially to encompass DeepSeek's ALF-LB procedure (Wang et al., 2024), that updates the load balancing parameters with negligible effect on the speed of the forward pass.

2 A Primal-Dual Framework for Optimal Load Balancing

Now, we establish a rigorous mathematical framework auxiliary-loss-free load balancing heuristics for s-MoE layers and, in particular, DeepSeek's ALF-LB method (Wang et al., 2024). In the remainder of the paper, for simplicity, we will refer to the normalized affinity scores γ_{ik} (Equation 1) as the "affinity scores" and adopt the convention of using them both for routing and aggregation.

2.1 Allocation Problem: Integer Program and Relaxation

Consider the exact-balancing primal problem for assigning T tokens to E experts. As a starting point, we make the following assumptions and stylizations:

- The number of tokens multiplied by the sparsity, KT, is exactly divisible by the number of experts E, so the perfectly balanced load is $L = KT/E$.
- The affinity scores γ_{ik} are constant from iteration to iteration[2].

Hence, the target load is $L := KT/E$ and perfect balance is characterized by the solution of the following integer program (IP):

$$\begin{aligned} \max_{\{x_{ik}\}} \ & \sum_{i,k} \gamma_{ik} x_{ik} \\ \text{s.t.} \ & \sum_{k} x_{ik} = K \quad \forall i = 1, \dots, T \\ & \sum_{i} x_{ik} = L \ \forall k = 1, \dots, E \\ & x_{ik} \in \{0, 1\} \quad \forall i, k. \end{aligned} \tag{4}$$

In practice, it is typically inadvisable (in terms of both time and memory requirements) to solve an IP for every MoE layer and on each individual batch of data[3].

Instead, we first relax the IP to a linear program (LP) by replacing the integer constraint $x_{ik} \in \{0, 1\}$ with $x_{ik} \in [0, 1]$. It is routine to show that the IP and the LP relaxation have the same

[2]This is a stylized assumption for the initial analysis in this section only. Later, in Section 5, we will consider the case where the affinity scores are new stochastic realizations from some distribution every iteration.

[3]One notable exception is the BASE layer heuristic invented by Lewis et al. (2021) which aims to approximately solve the IP using a truncated auction algorithm modeled after Bertsekas (1992).

optimal value. The Lagrangian of the LP relaxation is

$$\begin{aligned}\mathcal{L}(x, y, p) &= \sum_{i,k} \gamma_{ik} x_{ik} + \sum_i y_i \left(K - \sum_k x_{ik} \right) + \sum_k p_k \left(\sum_i x_{ik} - L \right) \\ &= \sum_{i,k} \left(\gamma_{ik} + p_k - y_i \right) x_{ik} + K \sum_i y_i - L \sum_k p_k. \end{aligned} \tag{5}$$

The corresponding dual problem is

$$\begin{aligned} \min_{\{y_i\},\{p_k\}} \quad & K \sum_i y_i - L \sum_k p_k \\ \text{s.t.} \quad & y_i - p_k \geq \gamma_{ik} \quad \forall i, k. \end{aligned}$$

However, even solving this LP relaxation to completion every iteration would still be too slow and often memory-infeasible.

2.2 Deriving ALF-LB from Primal-Dual Principles

We show that DeepSeek ALF-LB (Section 1.3) can be formulated as a primal-dual procedure that performs a single-shot update per iteration for finding a critical point of the Lagrangian (5). For conciseness, we introduce the following notation for the *load* of the k-th expert at iteration n:

$$A_k^{(n)} := \sum_i x_{ik}^{(n)}. \tag{6}$$

Indexing each training iteration with n, consider the following primal-dual scheme:

$$\textbf{Dual Update: } p_k^{(n+1)} \leftarrow p_k^{(n)} + \epsilon_k^{(n)} \left(L - A_k^{(n)} \right) \qquad \forall\ k. \tag{7}$$

$$\textbf{Primal Update: } x_{ik}^{(n+1)} \leftarrow \begin{cases} 1 & \text{if } k \in \text{TopKInd}_{k'} \left(\gamma_{ik'}^{(n+1)} + p_{k'}^{(n+1)} \right) \\ 0 & \text{otherwise} \end{cases} \qquad \forall\ i, k, \tag{8}$$

where $\left\{ \epsilon_k^{(n)} \right\}$ are step-sizes and $\text{TopKInd}_{k'}(\cdot)$ gives the indices that would induce the K-largest arguments. The Primal Update enforces $\sum_k x_{ik} = K$. Maximizing $\sum_{i,k} (\gamma_{ik} + p_k - y_i) x_{ik}$ subject to this constraint is equivalent to choosing the top K values of $\gamma_{ik} + p_k$ for each i, regardless of y_i. Thus we can simplify the Lagrangian by dropping the y_i terms, which gives:

$$\mathcal{L}(x, p) = \sum_{i,k} \left(\gamma_{ik} + p_k \right) x_{ik} - L \sum_k p_k. \tag{9}$$

Within this setup, the original DeepSeek ALF-LB update from Wang et al. (2024) described in (3) corresponds to the step-size

$$\epsilon_k^{(n)} = \frac{u}{\left|L - A_k^{(n)}\right|} \mathbb{1}\left\{\left|L - A_k^{(n)}\right| \neq 0\right\}, \quad \textbf{(DeepSeek ALF-LB Step-Size)} \tag{10}$$

where $\mathbb{1}\{...\}$ is the indicator function[4]. Figure 2 illustrates the convergence behavior of this primal-dual scheme during the training of 1B-parameter DeepSeekMoE models (Dai et al., 2024) with varying $\epsilon_k^{(n)}$ step-size choices. More experimental details are provided in Section 3.

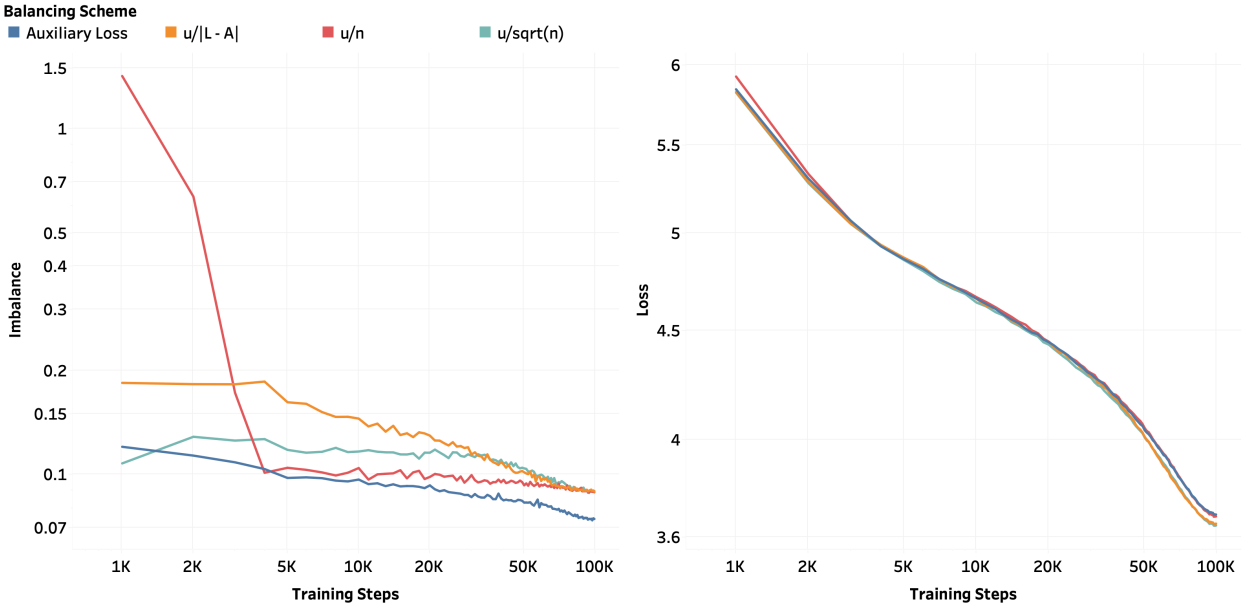


Figure 2: Validation set load imbalance and loss during the training of a 1B-parameter DeepSeekMoE model. Section 3 gives experiment details. *Left:* We measure the imbalance as the average load deviation from the target load $L = KT/E$ across all experts in the DeepSeekMoE-1B architecture. *Right:* We measure the loss on the validation set.

3 Experimental Setup and Observations

3.1 Experimental Setup

In all experiments in this paper (Figures 2-4), we train 1B-parameter DeepSeekMoE models (Dai et al., 2024) for 100K steps on the next-token prediction task on the Salesforce WikiText-103 dataset (Merity et al., 2016) with the cross-entropy loss. The text data is tokenized using the GPT-2 tokenizer (Radford et al., 2019).

Here, we will provide only a brief description of the DeepSeekMoE architecture for completeness and refer to Dai et al. (2024) for more in-depth details: The DeepSeekMoE architecture follows

[4]For conciseness, we use the short-hand "$\epsilon_k^{(n)} = \frac{u}{\left|L - A_k^{(n)}\right|}$" to refer to this step-size in the remainder of this paper.

the paradigmatic practice of stacking decoder-only transformer layers (Vaswani et al., 2017) into a full large language model. In its simplest form, the transformer layer contains several sub-layers — among them a multi-headed attention sub-layer and a multi-layer perceptron (MLP) sub-layer. For our setting of interest, modern s-MoE architectures (Shazeer et al., 2017; Jiang et al., 2024; Dai et al., 2024) replace the MLP sub-layer of each transformer layer with an s-MoE sub-layer described in Sections 1.1-1.2, where each parallel expert is typically a separate MLP. Additionally, the DeepSeekMoE architecture (Dai et al., 2024) is specifically characterized by its use of "granular segmentation" (using narrower experts but increasing the total number of experts) and the inclusion of two "shared experts" that are always chosen by the gate[5].

The architectural parameters of the 1B-parameter DeepSeekMoE models in our experiments are the same as those described in Wang et al. (2024, Table 5). For consistency with Wang et al. (2024), we also use $E = 64$ experts with sparsity level $K = 6$. During training, we optimize all 1B parameters within the transformer backbone and prediction head of the DeepSeekMoE architectures starting from random initializations. Each model was trained on 8xH100/H200 GPUs with a batch size of 64 sequences/batch and 4096 tokens/sequence (so, $T \approx 262K$). To optimize the models, we use the AdamW (Loshchilov and Hutter, 2019) optimizer.

Balancing Schemes. In our experiments, we compare three choices of the k-th expert step-sizes at iteration n (denoted $\epsilon_k^{(n)}$, see Section 2.2) in the ALF-LB balancing scheme framework. In particular, given some balancing hyperparameter u, we compare the following schemes:

- $\epsilon_k^{(n)} = \frac{u}{\left|L - A_k^{(n)}\right|}$ (Original DeepSeek ALF-LB from Wang et al. (2024))
- $\epsilon_k^{(n)} = \frac{u}{n}$
- $\epsilon_k^{(n)} = \frac{u}{\sqrt{n}}$

Additionally, we include a comparison with a fourth scheme that trains with an auxiliary loss (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022; Wang et al., 2024). We calculate the auxiliary loss with the method described in Wang et al. (2024, Section 2.2). The auxiliary loss is multiplied by a "trade-off parameter" that we will, for consistency, also denote by u and then added to the main cross-entropy loss.

Hyperparameter Search. For each of the four scheduling schemes, we conducted hyperparameter search over the following hyperparameters:

- balancing constants $u \in \{1e{-}4, 1e{-}3, 1e{-}2, 1e{-}1, 1, 10\}$,
- learning rates $\texttt{lr} \in \{1e{-}5, 1e{-}4, 1e{-}3\}$, and
- weight decay $\texttt{wd} \in \{0.01, 0.1, 0.001\}$.

Thus, we trained $4 \times 6 \times 3 \times 3 = 216$ separate 1B-parameter DeepSeekMoE models to conduct this search. Then, for each of the four scheduling schemes, we select the hyperparameter setting that

[5]We will not include the shared experts within the theoretical framework presented in this paper because the shared experts represent a fixed computational load that does not require dynamic balancing. Additionally, omitting the shared experts from our theoretical formulation leads to cleaner and more concise analyses.

achieves the best cross-entropy loss on a held-out validation set to be shown in the experimental plots and tables in this paper. We found that

- `lr` $= 1e{-}4$ and `wd` $= 1e{-}1$ consistently led to the best validation loss across all settings;
- the u/n and auxiliary loss scheduling schemes performed the best with parameter $u = 1$; and
- the $u/|L - A_k^{(n)}|$ and $u/\sqrt{n}$ scheduling schemes performed the best with $u = 1e{-}3$.

3.2 Experimental Observations

We make some interesting empirical observations from our experiments that are of separate interest from the theoretical framework proposed in this paper.

Firstly, we found that, for the original "constant update" scheme considered by Wang et al. (2024) (which corresponds to $u/|L - A_k^{(n)}|$ in our formalization), our hyperparameter search also yielded $u = 1e{-}3$ to be the optimal balancing constant, which corroborates the same observation from Wang et al. (2024).

Secondly, Table 1 reports the final validation loss and overall imbalance of the different balancing schemes at the end of training. Observe that the $u/\sqrt{n}$ scheme achieves the lowest validation loss (best predictive performance) but the highest imbalance (worst computational efficiency); in contrast, the auxiliary loss approach (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022) achieves the lowest imbalance (best computational efficiency) but the highest validation loss (worst predictive performance). The u/n scheme (which we will analyze in Section 5 through the lens of online optimization) and Wang et al. (2024)'s original $u/|L - A_k^{(n)}|$ scheme achieve a balance between validation loss and imbalance — with u/n achieving slightly better balance and $u/|L - A_k^{(n)}|$ achieving slightly better predictive performance.

Table 1: Comparison of cross-entropy loss on validation data and overall imbalance at the end of training for different scheduling schemes. Experiment details in Section 3.

Balancing Scheme	**Validation Loss**	**Overall Imbalance**
Auxiliary Loss	3.68999	**0.07443**
DeepSeek's Original ALF-LB $\left(u/\lvert L - A_k^{(n)}\rvert\right)$	3.65369	0.08928
u/n	3.68228	0.08893
$u/\sqrt{n}$	**3.64642**	0.08961

4 Convergence Analysis for the Deterministic Case

4.1 Section Assumptions

To start, we derive theoretical guarantees for the convergence of the procedure described by (7) and (8) in a stylistic setting where we **assume here in Section 4 only** that scores γ_{ik} are fixed and deterministic. We will later consider the case where the scores are stochastic in Section

5. Additionally, since the scores γ_{ik} are the output of softmax transformations (Section 1.1), we assume $\gamma_{ik} \in (0, 1)$ for all i, k.

4.2 Monotonicity of the Lagrangian

Towards showing the convergence of this procedure, we will show a monotonic improvement condition for the Lagrangian.

4.2.1 Lagrangian Characterization

We start by characterizing the change in the Lagrangian in each iteration. Two helpful abstractions are the *assignment set*

$$\alpha_n(i) := \text{TopKInd}_{k'}\left(\gamma_{ik'} + p_{k'}^{(n)}\right), \tag{11}$$

which gives the set of indices of the Top-K experts assigned to token i at iteration n; and the *switching benefit* of token i

$$b^{(n+1)}(i) := \sum_{k \in \alpha_{n+1}(i)} \left(\gamma_{ik} + p_k^{(n+1)}\right) - \sum_{k \in \alpha_n(i)} \left(\gamma_{ik} + p_k^{(n+1)}\right), \tag{12}$$

which captures the gain in the Lagrangian-affinity term when token i replaces its Top-K set $\alpha_n(i)$ by $\alpha_{n+1}(i)$ with respect to the fixed dual variables at iteration $n{+}1$. Since $\alpha_{n+1}(i)$ is chosen as a Top-K set with respect to $\gamma_{ik} + p_k^{(n+1)}$, we have $b^{(n+1)}(i) \geq 0$ for all i.

Theorem 1. ***(Change in Lagrangian)*** *Under the assumptions in Section 4.1 and using the procedure described in Steps (7)-(8), the following holds for the Lagrangian (9):*

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

Proof. Using the Lagrangian (9) definition, we have:

$$\begin{aligned}
&\mathcal{L}\left(x^{(n+1)}, p^{(n+1)}\right) - \mathcal{L}\left(x^{(n)}, p^{(n)}\right) \\
&\quad = \sum_{i,k} \left(\gamma_{ik} + p_k^{(n+1)}\right) x_{ik}^{(n+1)} - \sum_{i,k} \left(\gamma_{ik} + p_k^{(n)}\right) x_{ik}^{(n)} - L \sum_k (p_k^{(n+1)} - p_k^{(n)}) \\
&\quad = \sum_i \left[\sum_{k \in \alpha_{n+1}(i)} \left(\gamma_{ik} + p_k^{(n+1)}\right) - \sum_{k \in \alpha_n(i)} \left(\gamma_{ik} + p_k^{(n)}\right) \right] - L \sum_k \left(p_k^{(n+1)} - p_k^{(n)}\right) \\
&\qquad \text{by assignment set definition (11)} \\
&\quad = \sum_i \left[b^{(n+1)}(i) + \sum_{k \in \alpha_n(i)} \left(\gamma_{ik} + p_k^{(n+1)}\right) - \sum_{k \in \alpha_n(i)} \left(\gamma_{ik} + p_k^{(n)}\right) \right] - L \sum_k \left(p_k^{(n+1)} - p_k^{(n)}\right) \\
&\qquad \text{by switching benefit definition (12)}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_i b^{(n+1)}(i) + \sum_i \sum_{k \in \alpha_n(i)} \left(p_k^{(n+1)} - p_k^{(n)}\right) - L \sum_k \left(p_k^{(n+1)} - p_k^{(n)}\right) \\
&= \sum_i b^{(n+1)}(i) + \sum_k A_k^{(n)} \left(p_k^{(n+1)} - p_k^{(n)}\right) - L \sum_k \left(p_k^{(n+1)} - p_k^{(n)}\right) \\
&\qquad \text{by expert load definition (6)} \\
&= \sum_i b^{(n+1)}(i) + \sum_k \left(A_k^{(n)} - L\right) \left(p_k^{(n+1)} - p_k^{(n)}\right) \\
&= \sum_i b^{(n+1)}(i) - \sum_k \epsilon_k^{(n)} \left(A_k^{(n)} - L\right)^2 \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{by step (7) definition.}
\end{aligned}
$$

This completes the proof. □

Thus, Theorem 1 shows that the improvement in the Lagrangian is the difference between the total switching benefit and the squared sum of load imbalances weighted by step-sizes).

In fact, we can further characterize the switching benefit in terms of the expert choice changes between iterations. Specifically, define the sets of removed and newly selected experts, respectively, for some token i between iterations $n+1$ and n:

$$
\dot{\alpha}_-^{(n+1)}(i) := \alpha_n(i) \setminus \alpha_{n+1}(i), \qquad \dot{\alpha}_+^{(n+1)}(i) := \alpha_{n+1}(i) \setminus \alpha_n(i).
$$

We will denote the corresponding count of changed assignments as

$$
|\dot{\alpha}^{(n+1)}(i)| := |\dot{\alpha}_-^{(n+1)}(i)| = |\dot{\alpha}_+^{(n+1)}(i)|,
$$

where the index sets are necessarily equal-sized since we always select the top-K experts. As such, we can define an arbitrary bijection $\mathfrak{b}_i^{(n)}$ between $\dot{\alpha}_+^{(n+1)}(i)$ and $\dot{\alpha}_-^{(n+1)}(i)$ and define a set of entering-exiting pairs relative to $\mathfrak{b}_i^{(n)}$:

$$
\mathfrak{B}_i^{(n)} := \left\{ (k^+, k^-) \in \dot{\alpha}_+^{(n+1)}(i) \times \dot{\alpha}_-^{(n+1)}(i) \mid k^+ = \mathfrak{b}_i^{(n)}(k^-) \right\}.
$$

We can then characterize the switching benefit using $\mathfrak{B}_i^{(n)}$.

Proposition 2. ***(Switching Benefit Decomposition)*** *Assume the setting in Section 4.1 using the procedure described in Steps (7)-(8) as well as that there are no ties in bias-shifted scores $\gamma_{ik} + p_k^{(n)}$. Then, for a fixed token i at iteration n, we can write the switching benefit as*

$$
b_i^{(n+1)} = \sum_{(k^+, k^-) \in \mathfrak{B}_i^{(n)}} \left[\left(\gamma_{ik^+} + p_{k^+}^{(n+1)}\right) - \left(\gamma_{ik^-} + p_{k^-}^{(n+1)}\right) \right]. \tag{13}
$$

Proof. By definition of $b_i^{(n+1)}$ (Equation (12)),

$$\begin{aligned} b_i^{(n+1)} &= \sum_{k\in\alpha_{n+1}(i)} \left(\gamma_{ik} + p_k^{(n+1)}\right) - \sum_{k\in\alpha_n(i)} \left(\gamma_{ik} + p_k^{(n+1)}\right) \\ &= \sum_{k\in\dot{\alpha}_+^{(n+1)}(i)} \left(\gamma_{ik} + p_k^{(n+1)}\right) - \sum_{k\in\dot{\alpha}_-^{(n+1)}(i)} \left(\gamma_{ik} + p_k^{(n+1)}\right) \\ &= \sum_{(k^+,k^-)\in\mathfrak{B}_i^{(n)}} \left[\left(\gamma_{ik^+} + p_{k^+}^{(n+1)}\right) - \left(\gamma_{ik^-} + p_{k^-}^{(n+1)}\right)\right]. \end{aligned}$$

This proves the desired result. □

While Theorems 1 and Proposition 2 apply to arbitrary step-size $\epsilon_k^{(n)}$ choices, additional insights can be derived when specializing to the Wang et al. (2024)'s original step-size choice.

4.3 Analysis of DeepSeek's Original Step-Size Choice

Under Wang et al. (2024)'s original step-size choice (Equations (3) and (10)), we derive more precise behaviors for ALF-LB in our theoretical setting.

To start, we can infer the following bounds on differences of bias-shifted score for entering-exiting pairs that will be useful in the subsequent analysis. Note that we will use the $\text{Sign}(\cdot) : \mathbb{R} \to \{-1, 0, 1\}$ function with the convention $\text{Sign}(0) = 0$.

Lemma 3. ***(Pairwise Bounds)*** *Assume the setting in Section 4.1 using the procedure described in Steps (7)-(8) with DeepSeek's original step-size (10) as well as that there are no ties in bias-shifted scores $\gamma_{ik} + p_k^{(n)}$. Then, for a fixed token i, the following hold between iterations n and $n+1$:*

a. Every pair $(k^+, k^-) \in \dot{\alpha}_+^{(n+1)}(i) \times \dot{\alpha}_-^{(n+1)}(i)$ satisfies

$$0 < \left(\gamma_{ik^+} + p_{k^+}^{(n+1)}\right) - \left(\gamma_{ik^-} + p_{k^-}^{(n+1)}\right) < u\left(\text{Sign}\left(L - A_{k^+}^{(n)}\right) - \text{Sign}\left(L - A_{k^-}^{(n)}\right)\right) \tag{14}$$

$$< 2u, \tag{15}$$

and

$$-2u < \gamma_{ik^+} + p_{k^+}^{(n)} - \left(\gamma_{ik^-} + p_{k^-}^{(n)}\right) < 0. \tag{16}$$

b. If $\alpha_{n+1}(i) \neq \alpha_n(i)$, then the switching benefit of token i is bounded by the number of changed assignments:

$$0 < b_i^{(n+1)} < 2u\,|\dot{\alpha}^{(n+1)}(i)|.$$

If $\alpha_{n+1}(i) = \alpha_n(i)$, then $b_i^{(n+1)} = 0$.

Proof. Fix any pair $(k^+, k^-) \in \dot{\alpha}_+^{(n+1)}(i) \times \dot{\alpha}_-^{(n+1)}(i)$. By definition, $k^- \in \alpha_n(i)$ and $k^+ \notin \alpha_n(i)$, so

$$\gamma_{ik^+} + p_{k^+}^{(n)} - \gamma_{ik^-} + p_{k^-}^{(n)} < 0. \tag{17}$$

Similarly, $k^+ \in \alpha_{n+1}(i)$ and $k^- \notin \alpha_{n+1}(i)$, so

$$0 < \left(\gamma_{ik^+} + p_{k^+}^{(n+1)}\right) - \left(\gamma_{ik^-} + p_{k^-}^{(n+1)}\right). \tag{18}$$

Denote $s_k^{(n)} := \text{Sign}\left(L - A_k^{(n)}\right)$. Using the update (3), we can rewrite

$$\left(\gamma_{ik^+} + p_{k^+}^{(n+1)}\right) - \left(\gamma_{ik^-} + p_{k^-}^{(n+1)}\right) = \left(\gamma_{ik^+} + p_{k^+}^{(n)} - \gamma_{ik^-} - p_{k^-}^{(n)}\right) + u\left(s_{k^+}^{(n)} - s_{k^-}^{(n)}\right). \tag{19}$$

Combining (19) with (17) gives (14).

The inequality (15) follows since $s_{k^\pm}^{(n)} \in \{-1, 0, 1\}$, so sign differences are at most 2.

Next, rearranging (19) gives

$$\gamma_{ik^+} + p_{k^+}^{(n)} - \left(\gamma_{ik^-} + p_{k^-}^{(n)}\right) = \left(\gamma_{ik^+} + p_{k^+}^{(n+1)}\right) - \left(\gamma_{ik^-} + p_{k^-}^{(n+1)}\right) - u\left(s_{k^+}^{(n)} - s_{k^-}^{(n)}\right),$$

and using (18) together and again bounding sign differences by 2 yields (16).

Finally, if $\alpha_{n+1}(i) \neq \alpha_n(i)$, then $|\dot{\alpha}^{(n+1)}(i)| = |\mathfrak{B}_i^{(n)}| > 0$. By Proposition 2, $b_i^{(n+1)}$ is a sum of $|\dot{\alpha}^{(n+1)}(i)|$ differences of the form

$$\left(\gamma_{ik^+} + p_{k^+}^{(n+1)}\right) - \left(\gamma_{ik^-} + p_{k^-}^{(n+1)}\right),$$

and by Item (a) each such term lies in $(0, 2u)$. Summing gives $0 < b_i^{(n+1)} < 2u\,|\dot{\alpha}^{(n+1)}(i)|$. If $\alpha_{n+1}(i) = \alpha_n(i)$, then $b_i^{(n+1)} = 0$ trivially by definition. This proves Item (b). □

One immediate consequence of Proposition 2 and Lemma 3 is a more precise characterization of the Lagrangian change under DeepSeek's original step-size.

Theorem 4. ***(Lagrangian with DeepSeek Step-Size)*** *Consider the setting in Section 4.1 using the procedure described in Steps (7)-(8) with step-size DeepSeek's original step-size (10) as well as that there are no ties in bias-shifted scores $\gamma_{ik} + p_k^{(n)}$. Then, the Lagrangian change in Theorem 1 simplifies to*

$$\mathcal{L}\left(x^{(n+1)}, p^{(n+1)}\right) - \mathcal{L}\left(x^{(n)}, p^{(n)}\right) = \sum_i b^{(n+1)}(i) - u\sum_k \left|A_k^{(n)} - L\right|. \tag{20}$$

Furthermore, let $\mathcal{S}^{(n+1)}$ denote the index set of tokens whose assignment sets changed between iterations $(n+1)$ and n. Then,

$$\mathcal{L}\left(x^{(n+1)}, p^{(n+1)}\right) - \mathcal{L}\left(x^{(n)}, p^{(n)}\right) \leq u\left[2\sum_{i\in\mathcal{S}^{(n+1)}} |\dot{\alpha}^{(n+1)}(i)| - \sum_k \left|A_k^{(n)} - L\right|\right], \tag{21}$$

with strict inequality whenever $\mathcal{S}^{(n+1)} \neq \emptyset$.

Proof. Substituting step-size (10) into Theorem 1 yields (20).

Next, Lemma 3b states that $0 < b_i^{(n+1)} < 2u\,|\dot{\alpha}^{(n+1)}(i)|$ whenever $i \in \mathcal{S}^{(n+1)}$ and $b_i^{(n+1)} = 0$ otherwise. Summing over i gives

$$\sum_i b_i^{(n+1)} \leq 2u \sum_{i \in \mathcal{S}^{(n+1)}} |\dot{\alpha}^{(n+1)}(i)|,$$

with strict inequality if $\mathcal{S}^{(n+1)} \neq \emptyset$. Substituting this upper bound into (20) yields (21). □

Additionally, we can derive interesting implications for when a token's assignment set changes.

Theorem 5. ***(Assignment Change Implications)*** *Assume the setting of Theorem 4. Then, for any entering-exiting index pair $(k^+, k^-) \in \dot{\alpha}_+^{(n+1)}(i) \times \dot{\alpha}_-^{(n+1)}(i)$,*

$$\text{Sign}\left(L - A_{k^+}^{(n)}\right) > \text{Sign}\left(L - A_{k^-}^{(n)}\right).$$

In other words, entering experts are strictly lower in the ordering

$$\textit{Overloaded} \succ \textit{Balanced} \succ \textit{Underloaded}$$

than exiting experts.

Proof. By Lemma 3a, we have

$$0 < \left(\gamma_{ik^+} + p_{k^+}^{(n+1)}\right) - \left(\gamma_{ik^-} + p_{k^-}^{(n+1)}\right) < u\left(\text{Sign}\left(L - A_{k^+}^{(n)}\right) - \text{Sign}\left(L - A_{k^-}^{(n)}\right)\right).$$

The result follows after recalling $u > 0$. □

Next, we show that, if all experts' qualitative state (overloaded vs. balanced vs. underloaded) do not change between iterations, the Lagrangian cannot increase and, in fact, strictly decreases whenever there is any imbalanced experts.

Theorem 6. ***(Lagrangian Monotonicity)*** *Assume the setting of Theorem 4. Consider the index sets of overloaded, balanced, and underloaded experts, respectively, at iteration n:*

$$\mathbf{U}_>^{(n)} := \{k : A_k^{(n)} > L\}, \qquad \mathbf{U}_=^{(n)} := \{k : A_k^{(n)} = L\}, \qquad \mathbf{U}_<^{(n)} := \{k : A_k^{(n)} < L\}.$$

Assume the imbalance states stay unchanged between iterations n and $n+1$:

$$\mathbf{U}_>^{(n+1)} = \mathbf{U}_>^{(n)} \quad \textit{and} \quad \mathbf{U}_<^{(n+1)} = \mathbf{U}_<^{(n)}.$$

(So, necessarily, $\mathbf{U}_=^{(n+1)} = \mathbf{U}_=^{(n)}$ as well.) Then,

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

with strict inequality whenever there exists any imbalanced expert.

Proof. For conciseness, let $s_k^{(n)} := \text{Sign}\left(L - A_k^{(n)}\right)$. Combining Proposition 2 and Lemma 3a yields

$$b_i^{(n+1)} \leq \sum_{(k_i^+, k_i^-) \in \mathfrak{B}_i^{(n)}} u \left(s_{k_i^+}^{(n)} - s_{k_i^-}^{(n)} \right).$$

Theorem 5 implies $\left(s_{k_i^+}^{(n)} - s_{k_i^-}^{(n)} \right) > 0$, so equality holds if and only if $\mathfrak{B}_i^{(n)} = \emptyset$ i.e. when there are no changes in the assignment set for token i between iterations n and $n+1$.

Summing over all tokens i yields

$$\sum_i b^{(n+1)}(i) \;\leq\; u \sum_i \sum_{(k_i^+, k_i^-) \in \mathfrak{B}_i^{(n)}} \left(s_{k_i^+}^{(n)} - s_{k_i^-}^{(n)} \right), \tag{22}$$

where equality holds if and only if $\mathfrak{B}_i^{(n)} = \emptyset$ for all i.

Consider any individual expert k. Every time expert k appears as an entering expert ($k = k_i^+$) for some token i, it contributes $+s_k^{(n)}$ to the RHS of (22). Correspondingly, every time expert k appears as an exiting expert ($k = k_i^-$), it contributes $-s_k^{(n)}$ to the RHS of (22). Thus, the total contribution of each individual expert k to the RHS of (22) is

$$s_k^{(n)} \sum_i \left(x_{ik}^{(n+1)} - x_{ik}^{(n)} \right) = s_k^{(n)} \left(A_k^{(n+1)} - A_k^{(n)} \right),$$

where the equality follows from the definition (6). Summing over k allows us to rewrite (22) as

$$\sum_i b^{(n+1)}(i) \;\leq\; u \sum_{k=1}^{E} s_k^{(n)} \left(A_k^{(n+1)} - A_k^{(n)} \right). \tag{23}$$

Observe,

- Any balanced expert $k \in \mathbf{U}_=^{(n)} = \mathbf{U}_=^{(n+1)}$ satisfies $A_k^{(n+1)} = A_k^{(n)} = L$. So, they contribute 0 to the RHS sum in (23).
- Any overloaded expert $k \in \mathbf{U}_>^{(n)} = \mathbf{U}_>^{(n+1)}$ has $s_k^{(n)} = -1$ by definition. Theorem 5 implies that changes in any token's assignment set can not have an overloaded expert as the entering expert, so $A_k^{(n+1)} \leq A_k^{(n)}$ for that overloaded expert.
- Any underloaded expert $k \in \mathbf{U}_<^{(n)} = \mathbf{U}_<^{(n+1)}$ has $s_k^{(n)} = 1$ by definition. Similarly, Theorem 5 implies $A_k^{(n+1)} \geq A_k^{(n)}$ for that underloaded expert.
- Every token is assigned to K experts, so the total load is conserved: $\sum_k A_k^{(n+1)} = \sum_k A_k^{(n)} = KT$. Thus, the total gain of underloaded experts equals the total loss of overloaded experts:

$$\sum_{k \in \mathbf{U}_<^{(n)}} \left(A_k^{(n+1)} - A_k^{(n)} \right) = \sum_{k \in \mathbf{U}_>^{(n)}} \left(A_k^{(n)} - A_k^{(n+1)} \right).$$

Thus, we can rewrite the sum in (23) in terms of just the overloaded experts:

$$\sum_{k=1}^{E} s_k^{(n)} \left(A_k^{(n+1)} - A_k^{(n)} \right) = 2 \sum_{k \in \mathbf{U}_>^{(n)}} \left(A_k^{(n)} - A_k^{(n+1)} \right). \tag{24}$$

For $k \in \mathbf{U}_>^{(n)} = \mathbf{U}_>^{(n+1)}$, we have $A_k^{(n+1)} \geq L + 1$ by definition. So,

$$A_k^{(n)} - A_k^{(n+1)} \leq A_k^{(n)} - (L+1) = \left(A_k^{(n)} - L \right) - 1.$$

Summing over $k \in \mathbf{U}_>^{(n)}$ gives

$$\sum_{k \in \mathbf{U}_>^{(n)}} \left(A_k^{(n)} - A_k^{(n+1)} \right) \leq \sum_{k \in \mathbf{U}_>^{(n)}} \left(A_k^{(n)} - L \right) - |\mathbf{U}_>^{(n)}|. \tag{25}$$

Moreover, since total load is conserved and $L = KT/E$ by definition,

$$\begin{aligned} 0 &= \sum_k \left(A_k^{(n)} - L \right) \\ &= \sum_{k \in \mathbf{U}_>^{(n)}} \underbrace{\left(A_k^{(n)} - L \right)}_{>0} + \sum_{k \in \mathbf{U}_<^{(n)}} \underbrace{\left(A_k^{(n)} - L \right)}_{<0} + \sum_{k \in \mathbf{U}_=^{(n)}} \underbrace{\left(A_k^{(n)} - L \right)}_{=0} \\ &= \sum_{k \in \mathbf{U}_>^{(n)}} \left| A_k^{(n)} - L \right| - \sum_{k \in \mathbf{U}_<^{(n)}} \left| A_k^{(n)} - L \right|. \end{aligned}$$

From this, we can deduce

$$\sum_{k \in \mathbf{U}_>^{(n)}} \left(A_k^{(n)} - L \right) = \tfrac{1}{2} \sum_k \left| A_k^{(n)} - L \right|. \tag{26}$$

Combining (23), (24), (25), and (26) gives

$$\sum_i b^{(n+1)}(i) \;\leq\; u \left(\sum_k \left| A_k^{(n)} - L \right| - 2|\mathbf{U}_>^{(n)}| \right).$$

Substituting this bound into Theorem 4, yields

$$\mathcal{L}\left(x^{(n+1)}, p^{(n+1)} \right) - \mathcal{L}\left(x^{(n)}, p^{(n)} \right) \leq -2u\, |\mathbf{U}_>^{(n)}|.$$

Observe the the LHS is strictly negative whenever there exists any imbalanced experts. □

4.3.1 Gap Analysis and Maximum Token Movement

Within the same iteration, we can consider the gap between the scores or biases of two different experts for a given token. In particular,for some token i and experts k and k', define *score gap* as

$$\mathfrak{G}\gamma^i_{k'k} := \gamma_{ik'} - \gamma_{ik}.$$

Similarly, for two experts k and k', define the *bias gap* as

$$\mathfrak{G}p_{kk'} := p_k - p_{k'}. \tag{27}$$

Lemma 7. *Assume the setting of Theorem 4 and that there are also no ties between the scores* $\left\{\gamma^{(n)}_{ik}\right\}_{k=1}^{E}$ *and biases* $\left\{p^{(n)}_{k}\right\}_{k=1}^{E}$ *themselves. Suppose two tokens i and j both remove the same expert k^- and add the same expert k^+ between iterations n and $n+1$, i.e., $k^- \in \dot{\alpha}_-^{(n+1)}(i) \cap \dot{\alpha}_-^{(n+1)}(j)$ and $k^+ \in \dot{\alpha}_+^{(n+1)}(i) \cap \dot{\alpha}_+^{(n+1)}(j)$. Then, their score gaps relative to those experts satisfy*

$$\left|\mathfrak{G}\gamma^i_{k^+k^-} - \mathfrak{G}\gamma^j_{k^+k^-}\right| < 2u.$$

Proof. Since $k^- \in \dot{\alpha}_-^{(n+1)}(i)$ and $k^+ \in \dot{\alpha}_+^{(n+1)}(i)$, Lemma 3a implies

$$-2u < \gamma_{ik^+} + p^{(n)}_{k^+} - \left(\gamma_{ik^-} + p^{(n)}_{k^-}\right) < 0.$$

Writing $\mathfrak{G}\gamma^i_{k^+k^-} = \gamma_{ik^+} - \gamma_{ik^-}$ and $\mathfrak{G}p^{(n)}_{k^-k^+} = p^{(n)}_{k^-} - p^{(n)}_{k^+}$, this inequality is equivalent to

$$\mathfrak{G}p^{(n)}_{k^-k^+} - 2u < \mathfrak{G}\gamma^i_{k^+k^-} < \mathfrak{G}p^{(n)}_{k^-k^+}.$$

The same interval constraint holds for token j. Since the upper and lower bounds are independent of i, j and the interval has length at most $2u$, the result follows. □

Lemma 7 leads to an interesting implication: If we choose the step parameter u to be smaller than half the minimum difference between any two *distinct* score gaps, i.e.,

$$u < \bar{u} \quad \text{where} \quad \bar{u} := \frac{1}{2} \min_{k \neq k'} \min_{i \neq j} \left|\mathfrak{G}\gamma^i_{kk'} - \mathfrak{G}\gamma^j_{kk'}\right|,$$

and assume $\bar{u} > 0$, then movements between entering-exiting pairs of experts become unique to a particular token in any two consecutive iterations.

Proposition 8. ***(Uniqueness of Token Movements)*** *Assume the setting of Theorem 4 with step-length $u < \bar{u}$ and where there are also no ties between the scores* $\left\{\gamma^{(n)}_{ik}\right\}$ *and biases* $\left\{p^{(n)}_{k}\right\}$ *themselves. Consider the update from iteration n to $n+1$ and some fixed pair of experts (k^-, k^+). Then, at most one token can simultaneously remove expert k^- and add expert k^+. As a consequence, an expert's load cannot change by more than $(E-1)$ tokens between two consecutive iterations.*

Proof. Suppose, for contradiction, that between some iterations n and $n+1$ two distinct tokens $i \neq j$ both remove the same expert k^- and add the same expert k^+. Then Lemma 7 implies

$$\left|\mathfrak{G}\gamma^i_{k^+k^-} - \mathfrak{G}\gamma^j_{k^+k^-}\right| < 2u.$$

By the definition of $\bar{u}$, however,

$$\left|\mathfrak{G}\gamma^i_{k^+k^-} - \mathfrak{G}\gamma^j_{k^+k^-}\right| \geq 2\bar{u} > 2u,$$

which is a contradiction. Hence, for each ordered pair (k^-, k^+), at most one token can remove k^- and add k^+ between iterations n and $n+1$. There are $E-1$ possible origins (or destinations) for any fixed expert, so its load can increase or decrease by at most $(E-1)$ tokens in a single iteration. □

4.3.2 Convergence to Approximate Balance Band

Finally, we show that loads eventually converge to an *approximate balance band* where loads are within $(E-1)$ of the target load L. Specifically, the below Theorem 9 shows that once an expert's load enters the band, it remains in that band for all subsequent iterations. Later, Theorem 11 will show that all loads eventually enter the band in finite time.

Theorem 9. ***(Approximate Balance Band Stability)*** *Assume the setting of Proposition 8. Then, once an expert's load enters the range* $[L-(E-1), L+(E-1)]$*, it remains in that range for all subsequent iterations.*

Proof. Fix an expert k and suppose that for some iteration n,

$$A^{(n)}_k \in \big[L-(E-1),\, L+(E-1)\big].$$

It suffices to show that $A^{(n+1)}_k$ also lies in this interval as all subsequent iterations then follow by induction. Consider the following three cases:

- If $A^{(n)}_k = L$, then Proposition 8 yields

  $$\left|A^{(n+1)}_k - A^{(n)}_k\right| \leq E-1,$$

  so $A^{(n+1)}_k \in [L-(E-1),\, L+(E-1)]$.
- If $L+(E-1) \geq A^{(n)}_k > L$, then expert k is overloaded at iteration n and hence, by Theorem 5, it cannot be the entering expert in iteration $n+1$ in any change. Therefore,

  $$A^{(n+1)}_k \leq A^{(n)}_k \leq L+(E-1).$$

  Combining this with the per-iteration bound from Proposition 8 gives

  $$A^{(n+1)}_k \geq A^{(n)}_k - (E-1) \geq L-(E-1),$$

so $A_k^{(n+1)} \in [L-(E-1),\, L+(E-1)]$.

- If $L > A_k^{(n)} > L-(E-1)$, then expert k is underloaded at iteration n and an analogous argument to the above case (but instead using the fact that k is underloaded and so can not be the the exiting expert in any change) shows that $A_k^{(n+1)} \in [L-(E-1),\, L+(E-1)]$ as well.

This completes the proof. □

Now, we show the finite-time convergence of all expert loads to the approximate balance band.

Lemma 10. ***(Uniform Score Dominance Implies Load Dominance)*** *Assume the setting of Theorem 4. Fix an iteration n and consider two fixed experts k, k'. If*

$$\gamma_{ik} + p_k^{(n)} > \gamma_{ik'} + p_{k'}^{(n)} \qquad \forall i \in \{1, \dots, T\},$$

then, $k' \in \alpha_n(\bar{i})$ implies $k \in \alpha_n(\bar{i})$ for any token $\bar{i}$. Consequently, $A_k^{(n)} \geq A_{k'}^{(n)}$.

Proof. Consider any fixed token $\bar{i}$. If $k' \in \alpha_n(\bar{i})$, then k' is among the K largest values of $\left\{\gamma_{\bar{i}\ell} + p_\ell^{(n)}\right\}_{\ell=1}^{E}$. Since $\gamma_{ik} + p_k^{(n)}$ is strictly larger than $\gamma_{ik'} + p_{k'}^{(n)}$ for all i and there are no ties, expert k must rank above k' for token $\bar{i}$. So, $k \in \mathrm{TopKInd}(\{\gamma_{\bar{i}\ell} + p_\ell^{(n)}\}_\ell)$ and $k \in \alpha_n(\bar{i})$. Applying this argument across all tokens i for which $k' \in \alpha_n(i)$ yields $A_k^{(n)} \geq A_{k'}^{(n)}$. □

Theorem 11. ***(Finite-Time Entry into the Approximate Balance Band)*** *Assume the setting of Proposition 8. Then, for each expert k, there exists a finite iteration N_k such that its load satisfies*

$$A_k^{(N_k)} \in \big[L-(E-1),\, L+(E-1)\big].$$

Moreover, letting $N := \max_k N_k$, we have $A_k^{(m)} \in [L-(E-1),\, L+(E-1)]$ for every $m \geq N$.

Proof. Fix an expert k. We prove that the load of expert k must enter the band in finite time.

Suppose, for contradiction, that $A_k^{(n)} \notin [L-(E-1),\, L+(E-1)]$ for every $n \geq 0$. Since $A_k^{(n)}$ is an integer, for each n we must have bands in the "over-band" region $A_k^{(n)} \geq L+E$ or the "under-band" region $A_k^{(n)} \leq L-E$. Moreover, by Proposition 8, $\left|A_k^{(n+1)} - A_k^{(n)}\right| \leq E-1$, so expert k's load cannot move between the over-band region to the under-band region without landing in the band. Hence, exactly one of the following cases holds for all $n \geq 0$:

$$A_k^{(n)} \geq L+E \quad \text{for all } n \qquad \text{or} \qquad A_k^{(n)} \leq L-E \quad \text{for all } n.$$

We will show a contradiction in the the first, over-band case. The second case analogously follows a nearly identical argument.

Thus, we consider when $A_k^{(n)} \geq L+E$ for all n. So, k is overloaded at every iteration, so $\mathrm{Sign}(L - A_k^{(n)}) = -1$ and the update (3) gives $p_k^{(n+1)} = p_k^{(n)} - u$ for all n.

Next, every token is routed to exactly K experts, so $\sum_{r=1}^{E} A_r^{(n)} = KT = EL$ for all n. Since $A_k^{(n)} > L$ for all n, at each iteration, there must exist at least one underloaded expert: otherwise, all E loads would be at least L with at least one strictly larger than L, forcing the sum to exceed EL.

Because there are finitely many experts, there exists an expert $k^\star$ that is underloaded at infinitely many iterations, that is, $A_{k^\star}^{(n)} < L$ for infinitely many n.

For every n, the bias gap (27) satisfies

$$\begin{aligned}
\mathfrak{G}p_{k^\star k}^{(n+1)} - \mathfrak{G}p_{k^\star k}^{(n)} &= \left(p_{k^\star}^{(n+1)} - p_k^{(n+1)}\right) - \left(p_{k^\star}^{(n)} - p_k^{(n)}\right) \\
&= \left(p_{k^\star}^{(n+1)} - p_{k^\star}^{(n)}\right) - \left(p_k^{(n+1)} - p_k^{(n)}\right) \\
&= u\left(\mathrm{Sign}\left(L - A_{k^\star}^{(n)}\right) - \mathrm{Sign}\left(L - A_k^{(n)}\right)\right) \qquad \text{(by (3))} \\
&= u\left(\mathrm{Sign}\left(L - A_{k^\star}^{(n)}\right) + 1\right) \\
&\geq 0,
\end{aligned}$$

where, in the second-to-last line, we used $\mathrm{Sign}\left(L - A_k^{(n)}\right) = -1$ since we assumed expert k is overloaded at every iteration; and the last-line inequality holds because $\mathrm{Sign}(\cdot) \in \{-1, 0, 1\}$.

Next, whenever $A_{k^\star}^{(n)} < L$, we have $\mathrm{Sign}\left(L - A_{k^\star}^{(n)}\right) = 1$ and thus $\mathfrak{G}p_{k^\star k}^{(n+1)} - \mathfrak{G}p_{k^\star k}^{(n)} = 2u$. Since $k^\star$ is underloaded infinitely often, $\mathfrak{G}p_{k^\star k}^{(n)} \to +\infty$ and, because it is nondecreasing, there exists n_1 such that

$$\mathfrak{G}p_{k^\star k}^{(n)} > 1 \qquad \forall n \geq n_1.$$

For any token i and any $n \geq n_1$,

$$(\gamma_{ik^\star} + p_{k^\star}^{(n)}) - (\gamma_{ik} + p_k^{(n)}) = (\gamma_{ik^\star} - \gamma_{ik}) + (p_{k^\star}^{(n)} - p_k^{(n)}) > -1 + 1 = 0,$$

where we used $\gamma_{ik^\star}, \gamma_{ik} \in (0, 1)$ (Section 4.1). Hence, for all tokens i and all $n \geq n_1$,

$$\gamma_{ik^\star} + p_{k^\star}^{(n)} > \gamma_{ik} + p_k^{(n)}.$$

By Lemma 10, this implies $A_{k^\star}^{(n)} \geq A_k^{(n)} \geq L + E$ for all $n \geq n_1$, so $k^\star$ is overloaded for all $n \geq n_1$. This contradicts that $k^\star$ is underloaded infinitely often.

Therefore, the over-band case is impossible. The under-band case follows analogously.

We conclude that expert k must enter the band $[L - (E-1),\ L + (E-1)]$ at some finite time N_k. Since k was arbitrary, this holds for all experts. Letting $N := \max_k N_k$ yields simultaneous band entry, and Theorem 9 gives permanence thereafter. □

When the number of tokens is much larger than the number of experts ($T \gg E$), the deviation of $(E-1)$ from the perfectly balanced load $L = KT/E$ is negligible. This demonstrates DeepSeek's ALF-LB desirability under this section's stylized, deterministic setting. The next section will consider a more realistic stochastic setting.

5 Stochastic Analysis via Online Optimization

In practice, the affinity scores $\gamma_{ik}^{(n)}$ evolve during training. Thus, we generalize the previously considered Lagrangian (9) by now **assuming here in Section 5** that the scores $\gamma_{ik}^{(n)}$ are *stochastic* and drawn from expert-dependent distributions. In particular, at iteration n, we assume a random affinity score $\gamma_{ik}^{(n)} \in (0,1)$ is observed for each token $i \in \{1,\dots,T\}$ and expert $k \in \{1,\dots,E\}$. The algorithm updates a shift vector $p^{(n)} \in \mathbb{R}^E$ and, for each token i, selects the K experts with the largest values of $\gamma_{ik}^{(n)} + p_k^{(n)}$, where $p_k^{(n)}$ is the k-th coordinate of $p^{(n)}$.

Using the notation of Section 2.2, we note that this section will consider a step-size choice of $\epsilon_k^{(n)} = u/n$ instead of the $u/|L - A_k^{(n)}|$ step-size chosen by Wang et al. (2024). This is because, when affinity scores become stochastic and time-varying, analyzing the coordinate-dependent $u/|L - A_k^{(n)}|$ step-size sequence becomes technically intricate. In contrast, the diminishing and coordinate-independent u/n step-size connects directly with ideas from online convex analysis (Hazan, 2016, Section 3.3.1) leading to cleaner theoretical insights. This adjustment maintains experimental relevance and practicality: Figure 2 and Table 1 demonstrate that using the u/n step-size is comparable in effectiveness as using the original $u/|L - A_k^{(n)}|$ step-size. In fact, Table 1 shows that the u/n step-size leads to a slightly *better* load balancing performance than the $u/|L - A_k^{(n)}|$ step-size at the cost of a slightly higher validation loss.

5.1 Notation

For any vector $z \in \mathbb{R}^E$, define $\mathrm{TopKInd}(z) \subseteq \{1,\dots,E\}$ to be the set of indices of the K largest coordinates of z with ties broken arbitrarily. For round n and token i, denote $\Gamma^{(n),i} := \left(\gamma_{i1}^{(n)}, \dots, \gamma_{iE}^{(n)}\right) \in \mathbb{R}^E$. Routing at round n sends token i to the experts in $\mathrm{TopKInd}\big(\Gamma^{(n),i} + p^{(n)}\big)$.

Fix an integer $K \in \{1,\dots,E\}$. We frame this as minimizing the per-round online loss, which corresponds to the dual online objective:

$$f^{(n)}(p) = \sum_{i=1}^{T} \sum_{k \in \mathrm{TopKInd}\left(\Gamma^{(n),i}+p\right)} \left(\gamma_{ik}^{(n)} + p_k\right) - L \sum_{k=1}^{E} p_k, \tag{28}$$

where $L := KT/E$ is the desired per-expert load for Top-K routing. The k-th component of the loss gradient $g^{(n)}(p) \in \mathbb{R}^E$ is given by

$$g_k^{(n)}(p) := \nabla_k f^{(n)}(p) = A_k^{(n)}(p) - L, \tag{29}$$

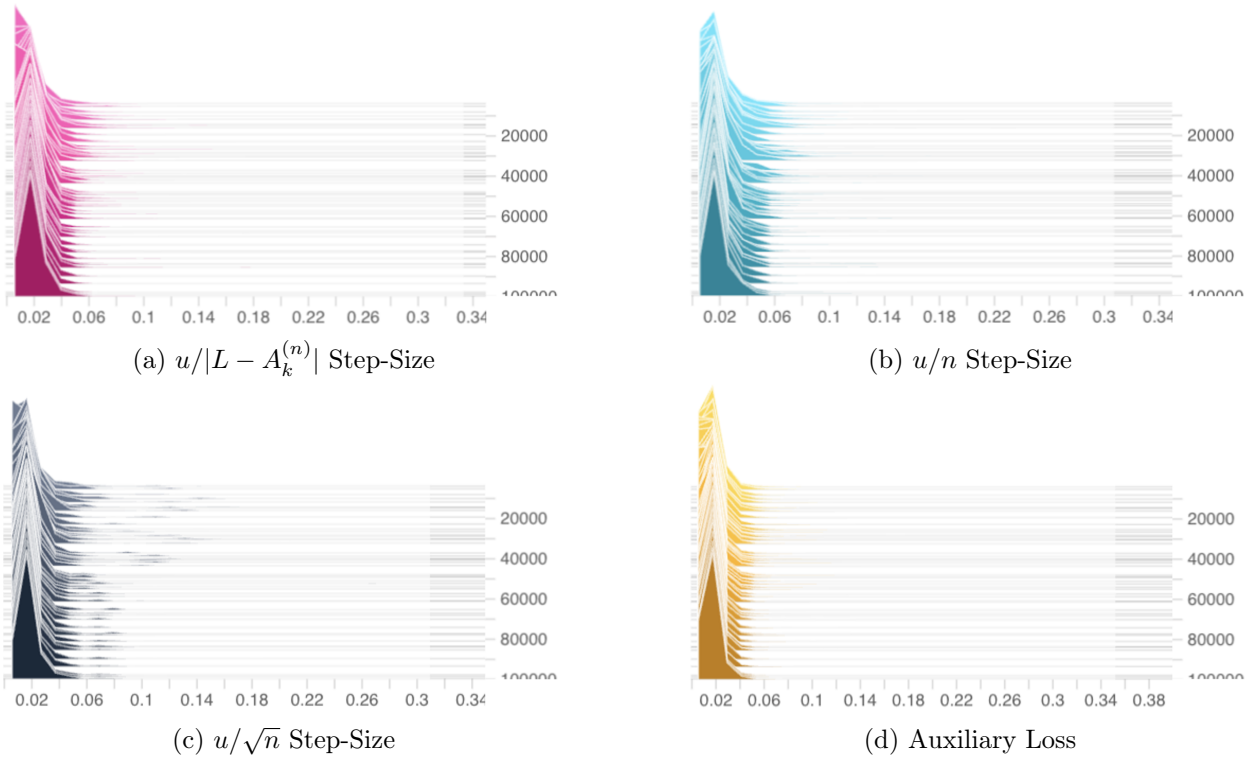
where $A_k^{(n)}(p) := \left|\left\{i : k \in \mathrm{TopKInd}\big(\Gamma^{(n),i} + p\big)\right\}\right|$ counts the number of tokens for which expert k lies in the per-token Top-K set under shifts p. The online dual update corresponding to (7) can then be rewritten as $p_k^{(n+1)} \leftarrow p_k^{(n)} - \epsilon_k^{(n)} g_k^{(n)}(p^{(n)})$.

5.2 Distributional Assumptions

Assume for the remainder of this section that each affinity $\gamma_{ik}^{(n)}$, for a fixed expert k, is drawn independently[6] from a distribution that

- depends only on the expert k,
- has bounded support on $(0, 1)$, and
- has a probability density function (pdf) φ_k that is upper bounded by some expert-independent constant.

Thus, for fixed k, the vectors of random affinity scores $\Gamma^{(n),i} = \left(\gamma_{ik}^{(n)}\right)_{k=1}^{E}$ for the i-th token in the n-th iteration are i.i.d across i and n. While this assumption may seem strong, our experiments in Figure 3 from training 1B-parameter DeepSeekMoE models suggest that it is close to reality.



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

(b) u/n Step-Size

(c) $u/\sqrt{n}$ Step-Size

(d) Auxiliary Loss

Figure 3: Time-lapse histograms of the marginal distributions of $\gamma_{ik}^{(n)}$ during the training of 1B-parameter DeepSeekMoE models using different choices of step-size (Section 2.2). Experimental details in Section 3.

[6]This independence assumption is a stylization: various mechanisms (attention, layer norm, etc.) in earlier layers could create dependencies between token embeddings. However, it gives us a starting point for building a tractable, baseline theory. Furthermore, Figure 3 demonstrates that the distributions of $\gamma_{ik}^{(n)}$ remain stable and well-behaved throughout training on DeepSeekMoE-1B models, which indicates that independence is empirically well-founded as an approximating simplification at least at the marginal distribution level.

5.3 Online Loss Gradient Analysis

5.3.1 Unbiasedness

It is easy to check that the loss $f^{(n)}$ is convex. Thus, the expected loss $\mathbf{f}(p) = \mathbb{E}\left[f^{(n)}(p)\middle| p\right]$ is also convex. We first show that the loss gradient $g^{(n)}(p)$ is an unbiased estimator of $\nabla\mathbf{f}(p)$ with an explicit form expression.

Proposition 12 (Unbiased Stochastic Gradient)**.** *For any fixed $p \in \mathbb{R}^E$,*

$$\mathbb{E}\left[g^{(n)}(p)\middle| p\right] \;=\; \nabla\mathbf{f}(p) \;=\; T\,\pi(p) - L\,\mathbf{1},$$

where $\mathbf{1}$ is the ones-vector, and $\pi(p) \in \mathbb{R}^E$ is the selection probabilities vector with k-th coordinate

$$\pi_k(p) \;:=\; \Pr(k \in \mathrm{TopKInd}(\Gamma + p))\,,$$

for some generic affinities $\Gamma \overset{d}{=} \Gamma^{(n),i}$. Necessarily, $\sum_k \pi_k(p) = K$ almost surely.

Proof. For a given token i, let

$$X_{ik}(p) := \mathbb{1}\left\{k \in \mathrm{TopKInd}\left(\Gamma^{(n),i} + p\right)\right\}. \tag{30}$$

Then, $\mathbb{E}\left[X_{ik}(p) \mid p\right] = \pi_k(p)$ by definition and the k-th expert loads are $A_k^{(n)}(p) = \sum_{i=1}^T X_{ik}(p)$. Hence, $\mathbb{E}\left[A_k^{(n)}(p) \mid p\right] = T\,\pi_k(p)$. Therefore,

$$\begin{aligned}
\mathbb{E}\left[g^{(n)}(p) \mid p\right] &= \mathbb{E}\left[\nabla f^{(n)}(p) \mid p\right] \\
&= \left(\mathbb{E}\left[A_1^{(n)}(p) \mid p\right] - L, \ldots, \mathbb{E}\left[A_E^{(n)}(p) \mid p\right] - L\right) \quad \text{by Eq. 29} \\
&= T\,\pi(p) - L\,\mathbf{1}.
\end{aligned}$$

Since the distribution of $\Gamma^{(n),i}$ is independent of i, observe that

$$\mathbf{f}(p) = T\,\mathbb{E}\left[\sum_{k \in \mathrm{TopKInd}(\Gamma+p)} (\Gamma_k + p_k)\right] - L\sum_k p_k.$$

To calculate $\nabla\mathbf{f}$, we can move the differentiation into the expectation via the dominated convergence theorem: the Γ_k have continuous densities so ties occur with probability zero; thus, for almost every realization, the partial derivative $\partial_{p_k}\sum_{m\in\mathrm{TopKInd}(\Gamma+p)}(\Gamma_m + p_m)$ exists and equals $\mathbb{1}\{k \in \mathrm{TopKInd}(\Gamma + p)\} \le 1$, yielding $\partial_{p_k}\,\mathbb{E}\left[\sum_{m\in\mathrm{TopKInd}(\Gamma+p)}(\Gamma_m + p_m)\right] = \Pr\{k \in \mathrm{TopKInd}(\Gamma + p)\} = \pi_k(p)$. The desired result follows. □

Using Proposition 12, we can compute the variance and second moment of $g^{(n)}(p)$.

Proposition 13. *(Variance and 2nd Moment) The variance and second moment of $g^{(n)}(p)$ are given by*

$$\mathrm{Var}\left[g^{(n)}(p)\middle|\, p\right] = T\left(K - \sum_{k=1}^{E} \pi_k(p)^2\right),$$

$$\mathbb{E}\left[\|g^{(n)}(p)\|^2\middle|\, p\right] = T^2\left(\sum_{k=1}^{E} \pi_k(p)^2 - \tfrac{K^2}{E}\right) + T\left(K - \sum_{k=1}^{E} \pi_k(p)^2\right).$$

Proof. Using Equation (29) and Proposition 12,

$$\begin{aligned}\mathrm{Var}\left(g^{(n)}(p)\middle|\, p\right) &= \mathbb{E}\left[\|g^{(n)}(p) - \nabla \mathbf{f}(p)\|^2 \mid p\right] \\ &= \mathbb{E}\left[\|\pi(p) - A^{(n)}(p)\|^2 \mid p\right]\end{aligned}$$

where $A^{(n)}(p) = \left(A_1^{(n)}(p), \ldots, A_E^{(n)}(p)\right)$ is the vector of expert loads. Let $X_i(p) \in \{0,1\}^E$ denote the assignment vector for token i with components as in (30). Then, $A^{(n)}(p) = \sum_{i=1}^{T} X_i(p)$ and $\mathbb{E}[X_i(p)] = \pi(p)$. So,

$$\begin{aligned}\mathrm{Var}\left(g^{(n)}(p)\middle|\, p\right) &= \mathbb{E}\left[\left\|\sum_{i=1}^{T} \pi(p) - X_i(p)\right\|^2 \middle| p\right] \\ &= \sum_{i=1}^{T} \mathbb{E}\left[\|\pi(p) - X_i(p)\|^2 \mid p\right] \quad \text{by independence across } i \\ &= T\sum_{k=1}^{E} \mathrm{Var}(X_{1k} \mid p) \quad \text{by identical distribution across } i \\ &= T\left(K - \sum_{k=1}^{E} \pi_k(p)^2\right),\end{aligned}$$

since $\sum_k X_{ik} = K$ a.s. and $\mathrm{Var}(X_{1k} \mid p) = \pi_k(p)\,(1 - \pi_k(p))$. Finally, decomposing the second moment and applying Proposition 12 gives

$$\begin{aligned}\mathbb{E}\left[\|g^{(n)}(p)\|^2 \mid p\right] &= \|\nabla \mathbf{f}(p)\|^2 + \mathbb{E}\left[\|g^{(n)}(p) - \nabla \mathbf{f}(p)\|^2 \mid p\right] \\ &= T^2\left(\sum_{k=1}^{E} \pi_k(p)^2 - \tfrac{K^2}{E}\right) + T\left(K - \sum_{k=1}^{E} \pi_k(p)^2\right).\end{aligned}$$

This completes the proof. □

Proposition 13 will be useful later to prove Theorem 17.

5.4 Second-Order Analysis of Expected Loss

In the following Sections 5.4-5.6, we show that the expectation of the Top-K objective is *strongly convex* with respect to p updates under certain (realistic) assumptions. The strong convexity then

allows us to show a logarithmic regret bound in Theorem 17. Without strong convexity, it is routine to verify that the regret bound is at best $\mathrm{O}(\sqrt{N})$ without additional assumptions. The next lemma characterizes the second directional derivative of the expected objective.

Proposition 14 (Second Directional Derivative)**.** *Let $\Gamma = (\Gamma_1, \dots, \Gamma_E)$ be a random affinity vector in $\mathbb{R}^E$ with the properties in Section 5.2. For biases $p \in \mathbb{R}^E$ define*

$$\mathbf{F}_K(p) \;=\; \mathbb{E}\left[\sum_{k \in \mathrm{TopKInd}(\Gamma + p)} (\Gamma_k + p_k)\right],$$

and let φ_k and Φ_k denote the density and CDF of Γ_k, respectively. Then, for any direction $\delta \in \mathbb{R}^E$, its second directional derivative at p is given by the formula

$$D^2\mathbf{F}_K(p)[\delta, \delta] \;=\; \sum_{k<\ell} w_{k\ell}^{(K)}(p)\,(\delta_k - \delta_\ell)^2, \tag{31}$$

where the symmetric edge weights are

$$w_{k\ell}^{(K)}(p) \;=\; \int_{-\infty}^{\infty} \varphi_k(v - p_k)\,\varphi_\ell(v - p_\ell)\,B_{k,\ell}^{(K-1)}(v; p)\,dv, \qquad w_{k\ell}^{(K)}(p) \geq 0,$$

with

$$B_{k,\ell}^{(K-1)}(v; p) \;=\; \sum_{\substack{S \subseteq [E]\setminus\{k,\ell\} \\ |S| = K-1}} \prod_{j \in S} \Phi_j^c(v - p_j) \prod_{m \in [E]\setminus(\{k,\ell\} \cup S)} \Phi_m(v - p_m).$$

Proof. For some fixed argument $\gamma \in [0,1]^E$, define the function

$$\bar{f}_{p,K}(\gamma) = \sum_{k \in \mathrm{TopKInd}(\gamma + p)} (\gamma_k + p_k),$$

so that $\mathbf{F}_K(p) = \int \bar{f}_{p,K}(\gamma)\,\varphi(\gamma)\,d\gamma$, where $\varphi(\gamma) = \prod_{k=1}^E \varphi_k(\gamma_k)$ is the joint density of Γ. For $t \in \mathbb{R}$, define $p(t) := p + t\,\delta$ and $\widetilde{\mathbf{F}}_K(t) := \mathbf{F}_K\left(p(t)\right)$. Since ties occur with probability zero, $\mathbf{F}_K$ is a.s. differentiable with gradient $\nabla\mathbf{F}_K(p) = \pi(p)$. Hence, the chain rule gives

$$\widetilde{\mathbf{F}}'_K(0) \;=\; \sum_{k=1}^E \delta_k\,\pi_k(p).$$

We will next compute $\widetilde{\mathbf{F}}''_K(0)$. For each k, using independence and conditioning on $\Gamma_k = v$,

$$\begin{aligned}
\pi_k(p) &= \Pr(k \in \mathrm{TopKInd}(\Gamma + p)) \\
&= \int_0^1 \varphi_k(v)\,\Pr(k \in \mathrm{TopKInd}(\Gamma + p) \,|\, \Gamma_k = v)\,dv \\
&= \int_0^1 \varphi_k(v)\,\Pr(|\{j \neq k : \Gamma_j + p_j > v + p_k\}| \leq K-1)\,dv
\end{aligned}$$

$$= \int_0^1 \varphi_k(v) \sum_{r=0}^{K-1} \Pr(|\{j \neq k : \Gamma_j + p_j > v + p_k\}| = r)\, dv$$
$$= \int_0^1 \varphi_k(v) \sum_{r=0}^{K-1} \sum_{\substack{S \subseteq [E]\setminus\{k\} \\ |S|=r}} \Pr(\forall j \in S : \Gamma_j + p_j > v + p_k \text{ and } \forall m \notin S \cup \{k\} : \Gamma_m + p_m \leq v + p_k)\, dv$$
$$= \int_0^1 \varphi_k(v) \sum_{r=0}^{K-1} \sum_{\substack{S \subseteq [E]\setminus\{k\} \\ |S|=r}} \prod_{j \in S} \Phi_j^c(v - p_j + p_k) \prod_{m \notin S \cup \{k\}} \Phi_m(v - p_m + p_k)\, dv.$$

where $\Phi_j^c(\cdot) := 1 - \Phi_j(\cdot)$ and S represents possible index sets within the top-K components of $\Gamma + p$ that are also larger than $v + p_k$. For notational convenience, set

$$\theta_{jk}(v,t) := v - (p_j - p_k) - t\,(\delta_j - \delta_k).$$

Then,

$$\pi_k\left(p(t)\right) = \int_0^1 \varphi_k(v) \sum_{r=0}^{K-1} \sum_{\substack{S \subseteq [E]\setminus\{k\} \\ |S|=r}} \left[\prod_{j \in S} \Phi_j^c\left(\theta_{jk}(v,t)\right) \prod_{m \notin S \cup \{k\}} \Phi_m\left(\theta_{mk}(v,t)\right) \right] dv.$$

Consider the integrand

$$\varphi_k(v) \sum_{r=0}^{K-1} \sum_{\substack{S \subseteq [E]\setminus\{k\} \\ |S|=r}} \prod_{j \in S} \Phi_j^c\left(\theta_{jk}(v,t)\right) \prod_{m \notin S \cup \{k\}} \Phi_m\left(\theta_{mk}(v,t)\right). \tag{32}$$

For each j, because Φ_j is differentiable everywhere on $\mathbb{R}$ except (possibly) at 0 or 1, the derivative of the integrand (32) with respect to t at $t = 0$ exists for all but finitely many $v \in (0,1)$. Thus, the integrand (32) is differentiable at $t = 0$ for almost all $v \in (0,1)$.

Next, observe that for each fixed $S \subseteq [E] \setminus \{k\}$, the product in (32) has the a.e. derivative

$$\frac{d}{dt}\left[\prod_{j \in S} \Phi_j^c\left(\theta_{jk}(v,t)\right) \prod_{m \notin S \cup \{k\}} \Phi_m\left(\theta_{mk}(v,t)\right) \right]$$
$$= \sum_{\ell \notin S \cup \{k\}} \underbrace{(\delta_k - \delta_\ell)\, \varphi_\ell\left(\theta_{\ell k}(v,t)\right) \prod_{j \in S} \Phi_j^c\left(\theta_{jk}(v,t)\right) \prod_{m \notin S \cup \{k,\ell\}} \Phi_m\left(\theta_{mk}(v,t)\right)}_{:= \Xi_{S,\ell}^+}$$
$$- \sum_{\ell \in S} \underbrace{(\delta_k - \delta_\ell)\, \varphi_\ell\left(\theta_{\ell k}(v,t)\right) \prod_{j \in S \setminus \{\ell\}} \Phi_j^c\left(\theta_{jk}(v,t)\right) \prod_{m \notin S \cup \{k\}} \Phi_m\left(\theta_{mk}(v,t)\right)}_{:= \Xi_{S,\ell}^-}.$$

This leads to a telescoping cancellation across r in the integrand (32). Specifically, for each fixed

$r < K-1$ and ℓ, every index set S_r such that $|S_r| = r$ corresponds to another index set S^ℓ_{r+1} such that $|S^\ell_{r+1}| = r+1$ and $S^\ell_{r+1} = S_r \cup \{\ell\}$. It is easy to check that $\Xi^+_{S_r,\ell} = \Xi^-_{S^\ell_{r+1},\ell}$.

So, the sum in (32) telescopes over r except at the $r = K-1$ boundary where $\Xi^+_{S_{K-1},\ell}$ has no corresponding "$\Xi^-_{S^\ell_K,\ell}$" term to cancel with. Hence, for almost all fixed $v \in (0,1)$,

$$\begin{aligned}
&\frac{d}{dt}\left[\varphi_k(v)\sum_{r=0}^{K-1}\sum_{\substack{S\subseteq[E]\setminus\{k\}\\|S|=r}}\prod_{j\in S}\Phi_j^c\left(\theta_{jk}(v,t)\right)\prod_{m\notin S\cup\{k\}}\Phi_m\left(\theta_{mk}(v,t)\right)\right]\Bigg|_{t=0}\\
&=\varphi_k(v)\sum_{\ell\neq k}(\delta_k-\delta_\ell)\,\varphi_\ell\left(\theta_{\ell k}(v,0)\right)\sum_{\substack{S\subseteq[E]\setminus\{k,\ell\}\\|S|=K-1}}\prod_{j\in S}\Phi_j^c\left(\theta_{jk}(v,0)\right)\prod_{m\notin S\cup\{k,\ell\}}\Phi_m\left(\theta_{mk}(v,0)\right).
\end{aligned}$$

Next, for any non-zero $h \in \mathbb{R}$, it is routine to check that the assumptions in Section 5.2 imply that the integrand of the following is uniformly bounded:

$$\begin{aligned}
&\frac{1}{h}\left[\pi_k\left(p(h)\right)-\pi_k\left(p(0)\right)\right]\\
&=\int_0^1\frac{1}{h}\Bigg[\varphi_k(v)\sum_{r=0}^{K-1}\sum_{\substack{S\subseteq[E]\setminus\{k\}\\|S|=r}}\prod_{j\in S}\Phi_j^c\left(\theta_{jk}(v,h)\right)\prod_{m\notin S\cup\{k\}}\Phi_m\left(\theta_{mk}(v,h)\right)\\
&\qquad-\varphi_k(v)\sum_{r=0}^{K-1}\sum_{\substack{S\subseteq[E]\setminus\{k\}\\|S|=r}}\prod_{j\in S}\Phi_j^c\left(\theta_{jk}(v,0)\right)\prod_{m\notin S\cup\{k\}}\Phi_m\left(\theta_{mk}(v,0)\right)\Bigg]dv
\end{aligned}$$

Therefore, by the dominated convergence theorem, we have that $\pi_k\left(p(t)\right)$ is differentiable at $t=0$, with the derivative being given (after a change of variables) by

$$\begin{aligned}
&D\pi_k(p)[\delta]\\
&\quad=\sum_{\ell\neq k}(\delta_k-\delta_\ell)\int_{-\infty}^{\infty}\varphi_k(v-p_k)\,\varphi_\ell\left(v-p_\ell\right)\sum_{\substack{S\subseteq[E]\setminus\{k,\ell\}\\|S|=K-1}}\prod_{j\in S}\Phi_j^c(v-p_j)\prod_{m\notin S\cup\{k,\ell\}}\Phi_m(v-p_m)\,dv
\end{aligned}$$

Finally, by symmetry,

$$\begin{aligned}
D^2\mathbf{F}_K(p)[\delta,\delta]=\frac{d}{dt}\sum_{k=1}^{E}\delta_k\,\pi_k(p+t\delta)\Big|_{t=0}&=\sum_{k=1}^{E}\delta_k\,D\pi_k(p)[\delta]\\
=\sum_{k=1}^{E}\sum_{\ell\neq k}\delta_k(\delta_k-\delta_\ell)\,w^{(K)}_{k\ell}(p)&=\sum_{1\le k<\ell\le E}w^{(K)}_{k\ell}(p)\,(\delta_k-\delta_\ell)^2,
\end{aligned}$$

which is exactly (31). □

5.5 Experimentally-Realistic Assumptions on p

Observe that the TopKInd decision of the MoE router is invariant to adding the same constant to all coordinates of p. Motivated by this, we define the zero-sum subspace

$$\mathcal{Z} := \left\{ z \in \mathbb{R}^E : \sum_{k=1}^{E} z_k = 0 \right\},$$

where $\mathcal{Z}$ is the linear subspace orthogonal to the all-ones vector. Thus, we assume the following about ALF-LB for some update direction δ':

$$p^{(n+1)} \leftarrow \mathrm{Proj}_{\mathcal{Z}} \left(p^{(n)} - \delta' \right). \tag{33}$$

Remark 15 (**Practicality of $\mathcal{Z}$ Assumptions**). The assumption (33) is not artificial; it arises naturally from the problem definition:

- **Zero-sum gradients.** Since $\sum_k A_k^{(n)}(p)=TK$, the components of the gradient (29) sum to zero:

$$\sum_k \nabla_k f^{(n)}(p) = \sum_k (A_k^{(n)}(p) - L) = TK - EL = 0.$$

  Thus, any update of the form

$$p^{(n+1)} \leftarrow p^{(n)} - \epsilon^{(n)} \nabla f^{(n)}(p^{(n)})$$

  automatically preserves $p^{(n+1)} \in \mathcal{Z}$ as long as $p^{(n)} \in \mathcal{Z}$. In practice, we initialize with $p^{(0)} = 0$, so the projection in (33) is just the identity mapping.

- **Explicit $\mathcal{Z}$-projection with per-coordinate step-sizes.** In the more general case where heterogeneous step-sizes $\epsilon_k^{(n)}$ are used across coordinates,

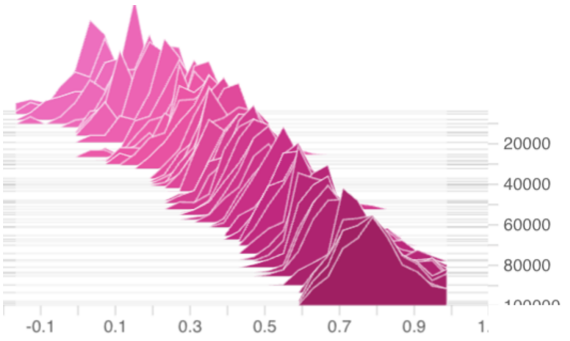
$$p^{(n+1)} \leftarrow p^{(n)} - \left(\epsilon_1^{(n)} g_1^{(n)}, \ldots, \epsilon_E^{(n)} g_E^{(n)} \right),$$

  the updated $p^{(n+1)}$ may not reside in $\mathcal{Z}$. (In fact, the difference between per-coordinate step-sizes and homogeneous step-sizes can be seen in Figure 4 where the per-coordinate $\epsilon_k^{(n)} = u/|L - A_k^{(n)}|$ step-size results in a bias distribution that shifts rightward over time while the homogeneous $\epsilon^{(n)} = u/n$ and $\epsilon^{(n)} = u/\sqrt{n}$ step-sizes result in bias distributions that stay centered around zero.) However, in this per-coordinate step-size case, it is well-known that the projection onto $\mathcal{Z}$ is equivalent to subtracting the componentwise mean:

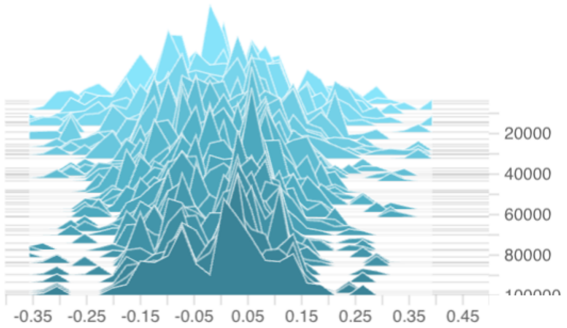
$$\mathrm{Proj}_{\mathcal{Z}}(p) = p - \left(\tfrac{1}{E} \sum_{k=1}^{E} p_k \right) \mathbf{1},$$

  which is computationally negligible.

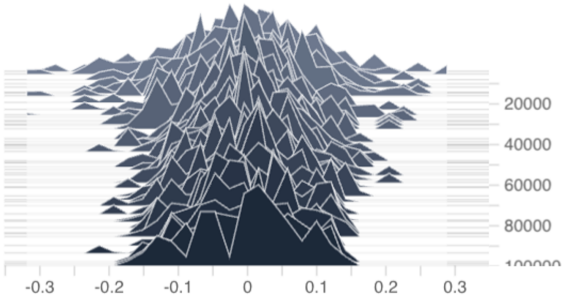
Additionally, we will make the technical assumption that $\mathrm{diam}(p) := \max_j p_j - \min_j p_j \leq 1 - \kappa$



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



(b) u/n Step-Size



(c) $u/\sqrt{n}$ Step-Size

Figure 4: Time-lapse histograms of the marginal distributions of the ALF-LB biases p during the training of 1B-parameter DeepSeekMoE models using different choices of step-size (Section 2.2). No explicit constraints were enforced on p. Section 3 provides experimental details.

for some constant $\kappa > 0$, which we found holds without explicit enforcement in our experiments on 1B-parameter DeepSeekMoE models (Figure 4). Thus, we can realistically assume

$$p \in \mathrm{dom}_{\mathcal{Z}}^{\kappa} := \{p \in \mathcal{Z} : \mathrm{diam}(p) \leq 1 - \kappa\}.$$

5.6 Strong Convexity

Next, observe that the component densities of the affinity scores Γ are continuous and strictly positive on $(0, 1)$. Thus, by the continuity of $w_{k\ell}^{(K)}(p)$ in p and compactness of $\mathrm{dom}_{\mathcal{Z}}^{\kappa}$,

$$c_K(d) \;:=\; \inf_{p \in \mathrm{dom}_{\mathcal{Z}}^{\kappa}} \min_{k<\ell} w_{k\ell}^{(K)}(p) \;>\; 0.$$

Hence, by Proposition 14,

$$\delta^\top \nabla^2 \mathbf{F}_K(p)\, \delta \;\geq\; c_K(d) \sum_{k<\ell} (\delta_k - \delta_\ell)^2. \tag{34}$$

The assumption (33) ensures that $p^{(n)} \in \mathcal{Z}$ for all n. Thus, since $\mathcal{Z}$ is a linear subspace,

$$p^{(n+1)} = \mathrm{Proj}_{\mathcal{Z}} \left(p^{(n)} - \delta' \right)$$

$$= \mathrm{Proj}_{\mathcal{Z}}\left(p^{(n)}\right) - \mathrm{Proj}_{\mathcal{Z}}\left(\delta'\right)$$
$$= p^{(n)} - \underbrace{\mathrm{Proj}_{\mathcal{Z}}\left(\delta'\right)}_{\delta}.$$

Since the update direction δ lies in $\mathcal{Z}$,

$$\sum_{k<\ell}(\delta_k - \delta_\ell)^2 \;=\; E\,\|\delta\|^2 - \left(\sum_{k=1}^{E}\delta_k\right)^2 \;=\; E\,\|\delta\|^2.$$

Combining with property (34), this yields

$$\delta^\top \nabla^2 \mathbf{F}_K(p)\,\delta \;\geq\; c_K(d) E\,\|\delta\|^2.$$

Recall the expected loss is $\mathbf{f}(p) = T\mathbf{F}_K(p) - L\sum_k p_k$ and observe the linear term does not affect curvature; thus, for all $p, \delta \in \mathcal{Z}$ with p having diameter at most d, $\mathbf{f}$ is μ_K-strongly convex with

$$\mu_K := T c_K(d) E. \tag{35}$$

5.7 Logarithmic Regret Bound for ALF-LB

Consider the minimizer of the expected loss

$$p^* = \arg\min_{p\in\mathcal{Z}} \mathbf{f}(p).$$

Since $\mathbf{f}$ is μ_K-strongly convex in $\mathcal{Z}$, p^* is necessarily unique.

Define the regret $R_N := \sum_{n=1}^{N}\left(f^{(n)}(p^{(n)}) - f^{(n)}(p^*)\right)$. We now give a logarithmic bound on the expected regret $\mathbb{E}[R_N]$ with the ALF-LB update

$$p^{(n+1)} \leftarrow \mathrm{Proj}_{\mathcal{Z}}\left(p^{(n)} - \epsilon^{(n)}\,\nabla f^{(n)}\left(p^{(n)}\right)\right). \tag{36}$$

While the details are adapted to the specific problem at hand, the proof technique is standard in online convex optimization (see, for example, Hazan (2016, Section 3.3.1)). For clarity, define the following short-hand notations:

$$\Delta_n := \mathbb{E}\left[\|p^{(n)} - p^*\|^2\right],\; s_n := \sum_{k=1}^{E}\pi_k\left(p^{(n)}\right)^2,\; a_n := \mathbb{E}\left[\mathbf{f}\left(p^{(n)}\right) - \mathbf{f}\left(p^*\right)\right],\; \sigma^2_{T,E,K} := T^2\left(K - \tfrac{K^2}{E}\right).$$

Lemma 16 (One-step accounting)**.** *Under the assumptions and notations of Section 5.1-5.6, for any $\epsilon^{(n)}>0$, the iteration (36) satisfies*

$$2\,a_n \;\leq\; \frac{\Delta_n - \Delta_{n+1}}{\epsilon^{(n)}} - \mu_K\,\Delta_n \;+\epsilon^{(n)}\,\sigma^2_{T,E,K}. \tag{37}$$

Proof. Since $\mathcal{Z}$ is a linear subspace, the projection operator is nonexpansive. Thus,

$$\begin{aligned}\|p^{(n+1)}-p^*\|^2 &= \left\|\mathrm{Proj}_{\mathcal{Z}}\left(p^{(n)}-\epsilon^{(n)}\nabla f^{(n)}\left(p^{(n)}\right)\right)-p^*\right\|^2\\ &\le \left\|p^{(n)}-\epsilon^{(n)}\nabla f^{(n)}\left(p^{(n)}\right)-p^*\right\|^2\\ &\le \|p^{(n)}-p^*\|^2-2\epsilon^{(n)}\left\langle\nabla f^{(n)}\left(p^{(n)}\right),p^{(n)}-p^*\right\rangle+\left(\epsilon^{(n)}\right)^2\left\|\nabla f^{(n)}\left(p^{(n)}\right)\right\|^2.\end{aligned}$$

Taking conditional expectation and using Proposition 12 gives

$$\begin{aligned}\mathbb{E}\left[\|p^{(n+1)}-p^*\|^2\mid p^{(n)}\right]\le\|p^{(n)}-p^*\|^2-2\epsilon^{(n)}\left\langle\nabla\mathbf{f}(p^{(n)}),p^{(n)}-p^*\right\rangle\\ +(\epsilon^{(n)})^2\,\mathbb{E}\left[\|\nabla f^{(n)}(p^{(n)})\|^2\mid p^{(n)}\right].\end{aligned}$$

Since the TopKInd decision is invariant to adding the same constant to all coordinates of p, we can assume without loss of generality that $p^*\in\mathcal{Z}$. Thus, since $\mathcal{Z}$ is a linear subspace, $p^{(n)}-p^*\in\mathcal{Z}$. Then, the μ_K-strong convexity of $\mathbf{f}$ in $\mathcal{Z}$ (Section 5.6) gives

$$2\left(\mathbf{f}\left(p^{(n)}\right)-\mathbf{f}\left(p^*\right)\right)+\mu_K\|p^{(n)}-p^*\|^2\le 2\left\langle\nabla\mathbf{f}(p^{(n)}),p^{(n)}-p^*\right\rangle.$$

Combining the last two expressions, taking total expectation, and rearranging gives

$$2\,a_n\;\le\;\frac{\Delta_n-\Delta_{n+1}}{\epsilon^{(n)}}-\mu_K\,\Delta_n\;+\epsilon^{(n)}\,\mathbb{E}\left[\|\nabla f^{(n)}(p^{(n)})\|^2\right]. \tag{38}$$

Recall from Section 5.1 that the gradient is $\nabla f^{(n)}(p)=A^{(n)}(p)-L\,\mathbf{1}$ where $\sum_k A_k^{(n)}(p)=TK$ and each $A_k^{(n)}(p)\in[0,T]$. It is then easy to check that, for any p,

$$\|\nabla f^{(n)}(p)\|^2\le\sigma^2_{T,E,K}.$$

Substituting this bound into (38) yields the desired result. □

Theorem 17. ***(Logarithmic Regret)*** *Consider the update (36) run for N iterations with $\epsilon^{(n)}=1/(\mu_K n)$. Then,*

$$\mathbb{E}[R_N]\;\le\;\frac{\sigma^2_{T,E,K}}{2\mu_K}\,(1+\ln N).$$

Proof. Observe that $\mathbb{E}[R_N]=\sum_{n=1}^N a_n$. Summing (37) over $n=1,\dots,N$ gives

$$2\sum_{n=1}^N a_n\le\sum_{n=1}^N\left(\frac{\Delta_n-\Delta_{n+1}}{\epsilon^{(n)}}-\mu_K\Delta_n\right)+\sum_{n=1}^N\epsilon^{(n)}\sigma^2_{T,E,K}.$$

The first term on the right-hand side is a telescoping sum which evaluates to

$$\begin{aligned}\sum_{n=1}^{N}\left(\frac{\Delta_n-\Delta_{n+1}}{\epsilon^{(n)}}-\mu_K\Delta_n\right) &= \sum_{n=1}^{N}\left(\mu_K n(\Delta_n-\Delta_{n+1})-\mu_K\Delta_n\right)\\ &= \mu_K\sum_{n=1}^{N}\left((n-1)\Delta_n-n\Delta_{n+1}\right)\\ &= -\,\mu_K\,N\,\Delta_{N+1}.\end{aligned}$$

Dropping this non-positive term, we are left with

$$2\sum_{n=1}^{N}a_n \;\le\; \sum_{n=1}^{N}\epsilon^{(n)}\,\sigma_{T,E,K}^2 \;=\; \frac{\sigma_{T,E,K}^2}{\mu_K}\sum_{n=1}^{N}\frac{1}{n}.$$

Invoking the classic $\sum_{n=1}^{N}\frac{1}{n}\le 1+\ln N$ inequality and dividing by 2 yields the desired result. □

Acknowledgements. The authors thank Alexandre Belloni, John Birge, Rene Caldentey, Chamsi Hssaine, and Nian Si for helpful discussions and feedback during the preparation of this paper. The authors also thank the Booth School of Business, University of Chicago for financial support that enabled this research. The empirical experiments and models training described in this paper were conducted on Chicago Booth's Pythia Supercomputer Cluster with the assistance of the Chicago Booth IT Department to whom the authors are grateful.

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.

Shipra Agrawal and Nikhil R Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 1405–1424. SIAM, 2014.

Santiago Balseiro, Haihao Lu, and Vahab Mirrokni. Dual mirror descent for online allocation problems. In *International Conference on Machine Learning*, pages 613–628. PMLR, 2020.

Santiago Balseiro, Haihao Lu, and Vahab Mirrokni. Regularized online allocation problems: Fairness and beyond. In *International Conference on Machine Learning*, pages 630–639. PMLR, 2021.

Dimitri Bertsekas. *Network optimization: continuous and discrete models*, volume 8. Athena Scientific, 1998.

Dimitri P Bertsekas. Auction algorithms for network flow problems: A tutorial introduction. *Computational optimization and applications*, 1(1):7–66, 1992.

Dimitri P Bertsekas. Auction algorithms. In *Encyclopedia of optimization*, pages 128–132. Springer, 2008.

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-v4: Towards highly efficient million-token context intelligence, 2026.

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

Epoch AI. Key trends and figures in machine learning, 2023. URL https://epoch.ai/trends. Accessed: 2025-09-27.

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.

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.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

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.

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

Rodolphe Jenatton, Jim Huang, Dominik Csiba, and Cedric Archambeau. Online optimization and regret guarantees for non-additive long-term constraints. *arXiv preprint arXiv:1602.05394*, 2016.

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.

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.

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.

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

Jelena Markovic-Voronov, Kayhan Behdin, Yuanda Xu, Zhengze Zhou, Zhipeng Wang, and Rahul Mazumder. Robust batch-level query routing for large language models under cost and capacity constraints. *arXiv preprint arXiv:2603.26796*, 2026.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021.

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.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 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.

Wenxin Zhang, Santiago R Balseiro, Robert Kleinberg, Vahab Mirrokni, Balasubramanian Sivan, and Bartek Wydrowski. Optimal and stable distributed bipartite load balancing. *arXiv preprint arXiv:2411.17103*, 2024.