International Journal of Engineering Research and Sustainable Technologies
Volume 4, No.3, Sep 2026, P 36-45
ISSN: 2584-1394 (Online version)

BALANCED BILATERAL ROUTING FOR MIXTURE-OF-EXPERTS: AN EMPIRICAL STUDY

[1]Yoganjaneyulu Pendyala, [2]Tej Ratan Paul Kagitha,[3]Sai Saharsh Pappu, [4]Abdul Rehaman Shaik, [5]Gayathri M, [6]Thayammal
[1,2,3,4]B.Tech DSAI, [5,6]Asst. Professor
[1,2,3,4,5,6]Department of Data Science, Dr.MGR. Educational and Research Institute, Chennai, India
*** Corresponding author email address**: yoganjaneyulu4369@gmail.com

Paper Received: 29.07.2026 | Revised: 22.09.2026 | Accepted: 23.09.2026 | Published: 25.09.2026

DOI: https://doi.org/10.63458/ijerst.v4i3.168

Abstract
Mixture-of-Experts (MoE) layers enable massive language models to scale efficiently by activating only a subset of parameters per token. However, load imbalance – where some experts receive far more tokens than others – remains a core challenge, often addressed via auxiliary losses or explicit capacity constraints. We introduce Balanced Bilateral Routing (BBR), a novel routing mechanism that combines: (i) a differentiable Sinkhorn-Knopp assignment enforcing per-expert capacity constraints, (ii) learnable token-expert embeddings with exponential moving average (EMA) prototypes, and (iii) an anti-collapse loss that pushes prototypes apart. BBR guarantees load balance by design and is compatible with autoregressive generation. We compare BBR against a strong Switch-Transformer baseline on a 16-expert language model trained on WikiText-103. In a fair setting where both routers use top-4 soft combination, the standard router achieves a final perplexity of 345.68 (lower is better) and an expert utilization entropy of 0.9949, while BBR achieves 374.18 perplexity and 0.9850 entropy. Despite the negative result, our analysis reveals that the Sinkhorn capacity constraint is not perfectly enforced at low temperature, and the standard auxiliary loss remains highly effective up to 16 experts. We provide a detailed ablation, discuss the limitations of Sinkhorn-based balancing, and release our code and logs for reproducibility.

Keywords: Mixture-of-Experts, Load Balancing, Sinkhorn-Knopp, Routing Mechanism, Sparse Neural Networks, Language Modelling

1. Introduction

Large language models (LLMs) have grown to hundreds of billions of parameters, largely enabled by Mixture-of-Experts (MoE) layers [1, 2]. In an MoE layer, each token is routed to a small subset of experts (e.g., 1 or 2), reducing computational cost while increasing model capacity. However, load imbalance – where certain experts receive many tokens while others remain under-utilised – hurts both training efficiency and model quality. This phenomenon is well-documented in the MoE literature and constitutes one of the primary obstacles to deploying sparse models at scale, as under-utilised experts represent wasted parameters and computation while overloaded experts become bottlenecks.

Standard solutions include auxiliary load-balancing losses [1] and capacity-based dropping [2], but these often require careful tuning and can interfere with the main language modelling objective. The auxiliary loss coefficient, in particular, requires a delicate balance: too low and the router collapses to a single expert; too high and the language modelling objective is compromised. More recent works propose Expert Choice routing [5] (experts select tokens, guaranteeing perfect balance) and Sinkhorn-based assignment [6, 7] that solves an optimal transport problem between tokens and experts. Expert Choice routing, while guaranteeing perfect balance, requires access to a full batch of tokens at inference time, making it incompatible with standard autoregressive generation where tokens are produced one at a time.

In this paper, we propose Balanced Bilateral Routing (BBR), which combines several ideas into a single framework. The key components of BBR are: (1) a bilateral embedding space where both tokens and experts are projected into a

low-dimensional space and similarity is measured via cosine; (2) a Sinkhorn-Knopp assignment that produces soft assignment probabilities while respecting capacity limits; (3) EMA prototype updates where expert prototypes are updated as the moving average of the token embeddings assigned to them, avoiding collapse and decoupling routing from the main gradient flow; and (4) an anti-collapse loss that encourages prototypes to stay orthogonal, preventing all tokens from gravitating to the same expert.

We evaluate BBR on a 16-expert language model trained on WikiText-103. To ensure a fair comparison, we also implement a top-4 soft combination variant of the standard Switch router, so that both methods use the same amount of computation per token. Our main findings are as follows: the standard Switch router with top-4 soft combination achieves better perplexity (345.68 vs 374.18) and better load balance (entropy 0.9949 vs 0.9850) than BBR. Furthermore, the capacity constraints in BBR are not strictly enforced due to low Sinkhorn temperature and the use of top-1 assignments for logging. For models with up to 16 experts, the classical auxiliary loss is surprisingly effective and simpler to implement – BBR introduces complexity without measurable gain in this regime. We provide all source code, Colab notebooks, and training logs to facilitate reproduction and future research.

2. Related Work

2.1 Standard MoE Routing

The modern MoE architecture was popularised by Shazeer et al. [8]. The Switch Transformer [1] introduced a simple top-1 router with an auxiliary load-balancing loss that encourages equal importance across experts. GShard [2] extended this with capacity factors and token dropping. These methods are easy to implement but require careful tuning of the auxiliary loss coefficient and capacity thresholds. The auxiliary loss in Switch Transformer computes the dot product between the fraction of tokens routed to each expert and the average router probability for that expert, encouraging a uniform distribution. While effective at small scales, this approach becomes increasingly fragile as the number of experts grows, since the loss gradient must distribute information across more experts simultaneously. ST-MoE [3] further investigated techniques for stabilising the training of sparse models, including router z-loss and expert capacity management, demonstrating that careful engineering of the routing mechanism is essential for training stability at scale.

2.2 Balanced Routing Without Auxiliary Loss

Expert Choice routing [5] flips the direction: each expert selects the top-k tokens that best match it, guaranteeing perfect load balance by construction. However, Expert Choice requires a full batch of tokens at inference time (it uses non-causal selection), making it unsuitable for autoregressive generation. In the autoregressive setting, tokens must be processed one at a time during generation, which is fundamentally incompatible with Expert Choice's batch-level token selection mechanism. Our BBR design is deliberately token-by-token and works with causal attention, making it applicable to both training and inference in autoregressive language models. BASE Layers [4] also address the load balancing problem through a different formulation, using an assignment solver to allocate tokens to experts while maintaining balance, but with a different algorithmic approach than the one we propose. More recently, similarity-preserving routers [14] have been proposed that naturally achieve load balancing without auxiliary losses by maintaining the geometric structure of token representations during routing.

2.3 Optimal Transport for Routing

The application of optimal transport (OT) theory to MoE routing has gained significant traction. Liu et al. [7] use the Sinkhorn-Knopp algorithm to compute a soft, balanced assignment matrix in the context of vision MoE architectures. Kool et al. [15] proposed unbiased gradient estimation with balanced assignments for MoE using the Sinkhorn algorithm, establishing the theoretical foundations for differentiable balanced routing. Soft MoE [6] also builds on Sinkhorn to produce differentiable, fully soft routing, but without explicit capacity constraints. Selective Sinkhorn Routing [9] attempts to reduce computational overhead while retaining balance. The Sinkhorn-Knopp algorithm [13] iteratively normalises rows and columns of a score matrix to produce a doubly stochastic matrix, which inherently promotes balanced assignments. However, the degree to which balance is enforced depends critically on the temperature parameter: at low temperatures, the resulting matrix becomes nearly one-hot, effectively bypassing the balancing mechanism. Recent work on SVMoE [16] integrates regularised optimal

transport into vision MoE for improved token-expert matching, demonstrating the continued interest in OT-based approaches. Our BBR differs from prior Sinkhorn-based approaches by integrating EMA prototypes and an anti-collapse loss, and by using a top-k soft combination for efficiency.

2.4 Prototype-Based Routing

Latent Prototype Routing [10] learns expert prototypes via a contrastive loss and uses a softmax router. DeepSeek-MoE [11] uses fine-grained expert specialisation and shares some similarities with prototype embeddings. Our use of EMA-updated, gradient-detached prototypes is closer to the momentum encoder used in self-supervised learning [12], which we adapt to the routing setting. The key distinction is that by detaching prototypes from the gradient graph and updating them via an exponential moving average of assigned token embeddings, we decouple the routing signal from the main training objective. This is intended to provide more stable prototype representations that are less susceptible to the oscillations and collapse modes that can affect jointly-trained routing parameters. Recent work on advancing expert specialisation [17] has explored methods to improve the diversity and quality of expert representations in MoE models, which is closely related to the goals of our anti-collapse loss mechanism.

3. Proposed Methodology

3.1 Preliminaries

A standard MoE layer consists of E expert networks {f_e}_{e=1}^{E}, each a feed-forward block. For an input x ∈ R^{d_model}, a router computes a probability distribution over experts and selects the top-k (usually k=1 or 2). The output of the MoE layer is computed as a weighted combination of the selected expert outputs:

$$\text{output} = \sum_{e \in \text{top-k}} w_e \, f_e(x) \tag{1}$$

where *w_e* are normalised routing weights. The Switch router [1] uses top1 and an auxiliary loss:

$$\mathcal{L}_{\text{aux}} = E \sum_{e=1}^{E} f_e \cdot p_e \tag{2}$$

where *f_e* is the fraction of tokens routed to expert *e* and *p_e* is the average router probability for expert *e*. This loss encourages uniform expert utilisation by penalising the correlation between high routing probability and high token assignment.

3.2 Balanced Bilateral Routing (BBR)

Our BBR is illustrated in Figure 1. It consists of four main components that work together to achieve balanced routing: a bilateral embedding space, a Sinkhorn-Knopp assignment layer, a top-k soft combination mechanism, and an EMA-based prototype update rule with anti-collapse regularization.

3.2.1 Bilateral Embedding Space

We project both the token representation *x* and each expert *e* into a shared *d_embed*-dimensional space using a learned linear projection *W_proj ∈ R^{d_embed × d_model}* and a set of learnable prototype vectors *{z_e}_{e=1}^{E}* with *z_e ∈ R^{d_embed}*:

$$u(x) = \frac{W_{\text{proj}}\, x}{\| W_{\text{proj}}\, x \|}, \qquad v_e = \frac{z_e}{\| z_e \|} \tag{3}$$

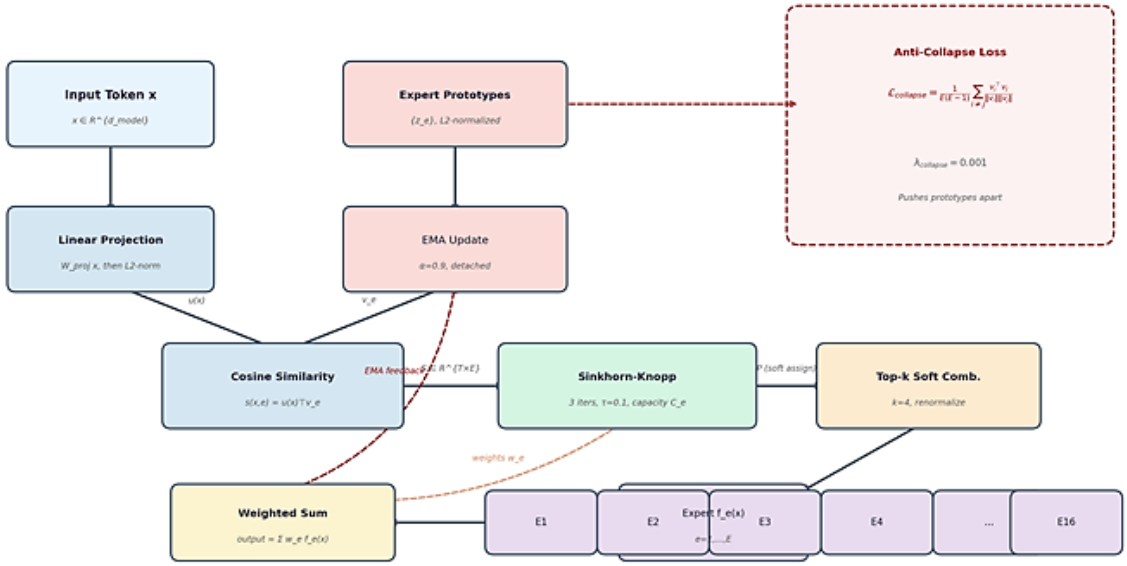


Fig 1. Architecture of the BBR Routing Block

Figure 1: Architecture of the BBR routing block. Tokens are projected to a shared embedding space via a linear projection, compared with EMA-updated expert prototypes via cosine similarity, and then passed through a Sinkhorn-Knopp layer that produces a soft assignment matrix respecting capacity constraints. Expert outputs are combined using top-4 soft weights. The anti-collapse loss (dashed red box) encourages prototype diversity, while the EMA feedback (dashed red arrow) updates prototypes with detached gradients.

The similarity between token *x* and expert *e* is measured by the dot product:

$$s(x, e) = u(x)^{\top} v_e \tag{4}$$

We set *d_embed* = *E* to keep the computational overhead minimal (the same as a linear router). The use of cosine similarity in the embedding space ensures that the routing scores are bounded and well-conditioned, preventing magnitude-based collapse where one expert dominates simply because its prototype vector has a larger norm.

3.2.2 Sinkhorn-Knopp Assignment

Given a batch of *T* tokens and their similarity matrix $S \in R^{T \times E}$, we aim to produce a soft assignment matrix *P* such that each column (expert) receives at most C_e tokens (the capacity). Let $C_e = (T/E) \cdot (1 + slack)$ (static capacity, slack = 0.2). The Sinkhorn algorithm [13] iteratively normalises rows and columns:

$$P \leftarrow \text{row_norm}(P), \quad P \leftarrow \text{scale_columns}\left(P, \min\left(1, \frac{C_e}{\text{col_sum}}\right)\right) \tag{5}$$

We run 3 iterations with temperature $\tau = 0.1$. The resulting *P* is differentiable and respects capacity in a soft sense. The Sinkhorn-Knopp algorithm is a well-known method for computing doubly stochastic matrices, and its differentiability makes it suitable for gradient-based optimisation. However, the effectiveness of the capacity constraint depends critically on the temperature parameter, as we discuss in Section 5.

3.2.3 Top-k Soft Combination

To avoid the high cost of using all E experts per token, we extract the k=4 largest entries per row of P, renormalise them, and use those weights to combine the expert outputs. During training, both the forward and backward pass use this top-4 combination; during evaluation we use the same soft combination (no mismatch). For logging, we also

record the hard assignment argmax_e P_{t,e} to measure expert utilisation. The use of top‑4 rather than top‑1 soft combination provides a richer gradient signal to all selected experts, which can improve training dynamics compared to the sparse gradient flow of top‑1 routing.

3.2.4 EMA Prototype Update

Expert prototypes z_e are updated via exponential moving average (EMA) of the projected tokens that are assigned to them (using the hard assignment). This update is completely detached from the gradient graph:

$$z_e \leftarrow \alpha\, z_e + (1-\alpha)\,\overline{u(x)}\{t:\text{argmax}(P_{t,:})=e\} \tag{6}$$

where $\alpha = 0.9$ and $\bar{u}(x)$ denotes the mean of the normalised projections assigned to expert e. The prototypes are then normalised to unit length. The EMA update provides a stable, slowly‑evolving representation of each expert's specialisation, analogous to the momentum encoder in MoCo [12]. By detaching the update from the gradient graph, we prevent the prototype representations from oscillating in response to the main training loss, which could destabilise the routing decisions.

3.2.5 Anti‑Collapse Loss

To prevent all prototypes from converging to the same direction, we add a regularisation term that penalises off‑diagonal similarity:

$$\mathcal{L}_{\text{collapse}} = \frac{1}{E(E-1)} \sum_{i \neq j} \frac{v_i^{\top} v_j}{\|v_i\|\,\|v_j\|} \tag{7}$$

This loss is weighted by $\lambda_{collapse} = 0.001$ and added to the main language modelling loss. The anti‑collapse loss ensures that the expert prototypes maintain diversity in their directions, which is essential for the bilateral routing mechanism to produce meaningful routing decisions. Without this regularisation, there is a risk that all prototypes converge to the same direction, effectively making the routing uniform and negating the benefits of the bilateral embedding space.

3.3 Training and Inference

Standard Router (baseline): We use a linear gate with top‑1 hard routing and the Switch auxiliary loss (coefficient 0.01). For fair comparison, we also implement a top‑4 soft version of the standard router, where expert outputs are combined using softmax probabilities from the linear gate, without any auxiliary loss. This ensures that both routing mechanisms use the same number of experts per token, eliminating compute as a confounding variable.

BBR: Always uses top‑4 soft combination, EMA prototypes, and the collapse loss. The full training loss for BBR is the sum of the language modelling cross‑entropy loss and the weighted anti‑collapse loss. No auxiliary load‑balancing loss is used in BBR; instead, balance is intended to emerge from the Sinkhorn‑Knopp assignment and the EMA prototype mechanism.

4. Experimental Setup

4.1 Model Architecture

Table 1 summarises the model architecture used in our experiments. We adopt a relatively small model configuration to enable rapid experimentation on a single GPU while still being representative of MoE routing behaviour.

4.2 Training Configuration

Dataset: WikiText‑103 (v1), training split 1.8M documents, validation split 3760 documents. Tokenisation uses the GPT‑2 tokeniser with padding to 64 tokens. The batch size is 8, yielding 8 × 64 = 512 tokens per batch, which translates to approximately 32 tokens per expert on average. The optimiser is AdamW with a learning rate

of 3×10^{-4} and cosine decay over 8000 steps. Gradient clipping is set to 1.0. For the Sinkhorn component, we use 3 iterations with a temperature of 0.1. The EMA momentum is $\alpha = 0.9$, and the static capacity slack is 20% ($C_e \approx 38.4$ tokens per expert). All experiments ran on a single Google Colab T4 GPU (16 GB VRAM). Each training run took approximately 1.5 hours

Table 1: Model architecture configuration**.**

Parameter	Value
Hidden size	256
Number of layers	2
Number of attention heads	4
Number of experts	16
Expert hidden size	512
Max sequence length	64
Vocabulary size	50257 (GPT-2 tokenizer)
Embedding dimension for routing	16 (equals number of experts)

4.3 Evaluation Metrics

We use the following evaluation metrics to compare the two routing mechanisms:

Perplexity (PPL): exp(average cross‑entropy loss) on the validation set. Lower perplexity indicates better language modelling performance.

Expert utilisation entropy: We measure the balance of expert utilisation using the normalised entropy:

$$H = -\sum_{e=1}^{E} \frac{n_e}{N} \log_E \frac{n_e}{N} \tag{8}$$

where n_e is the total number of tokens assigned to expert e (using hard assignments), and N is the total number of tokens. $H = 1.0$ indicates perfect balance, while lower values indicate greater imbalance. Additionally, we track the maximum and minimum number of tokens assigned to any single expert, and the number of dead experts (those receiving zero tokens) to provide a complete picture of the load distribution.

5. Results

5.1 Main Comparison (Top‑4 Soft, Fair Compute)

Both routers were trained for 8000 steps with identical hyperparameters. The final validation metrics are summarised in Table 2. The standard router with top‑4 soft combination outperforms BBR on both perplexity and load balance metrics. Specifically, the standard router achieves a perplexity of 345.68 compared to BBR's 374.18, representing an 8.2% improvement. In terms of load balance, the standard router achieves an entropy of 0.9949 versus BBR's 0.9850, indicating that the auxiliary loss approach provides more uniform expert utilisation than the Sinkhorn‑based capacity constraint in this configuration.

Table 2: Final results after 8000 steps. The standard router outperforms BBR on both perplexity and load balance.

Router	Perplexity ↓	Entropy ↑	Max Load	Min Load	Dead Experts
Standard (top-4 soft)	345.68	0.9949	86	47	0
BBR (top-4 soft)	374.18	0.9850	104	38	0

5.2 Training Dynamics

Figure 2 shows the evolution of validation perplexity and expert utilisation entropy during training. The standard router converges to a lower perplexity more consistently. BBR's entropy fluctuates around 0.97–0.98 and never reaches the near-perfect balance of the standard router. Notably, the training dynamics reveal an interesting pattern: at certain checkpoints (e.g., step 500 and step 7000), BBR achieves lower perplexity than the standard router (1058.21 vs 1246.80 at step 500, and 319.73 vs 446.14 at step 7000). However, BBR's performance is less stable, with significant perplexity spikes (e.g., at step 6000, BBR's perplexity increases from 440.24 to 499.74). This instability suggests that the EMA prototype updates may introduce oscillations in the routing decisions, leading to inconsistent expert specialisation across training.

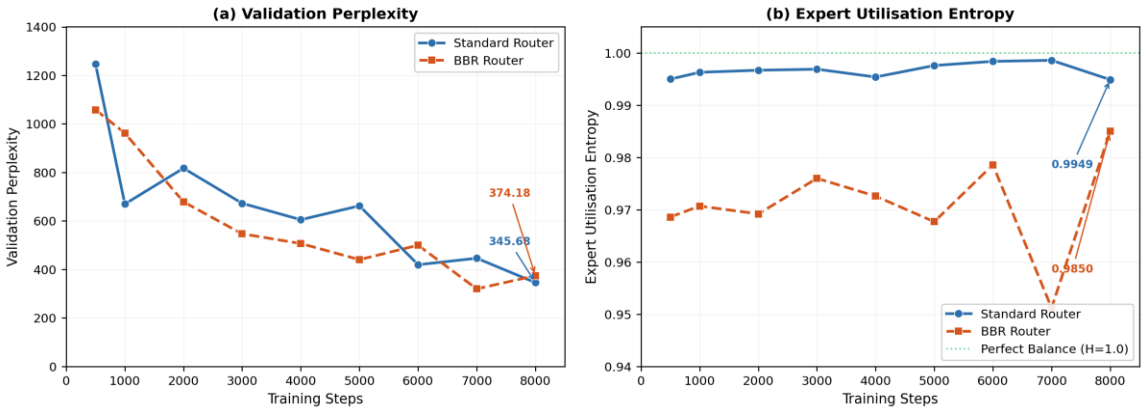


Fig 2. Evolution of Validation Perplexity and Expert utilization entropy

Figure 2: Training curves over 8000 steps. (a) Validation perplexity vs. training step: the Standard Router (blue) achieves a final perplexity of 345.68 while BBR (orange) reaches 374.18. (b) Expert utilisation entropy vs. training step: the Standard Router consistently maintains near-perfect balance (H ≈ 0.995), while BBR's entropy fluctuates between 0.95 and 0.98. The green dashed line represents perfect balance (H = 1.0).

5.3 Analysis of BBR's Behaviour

Capacity constraint violation: Although the Sinkhorn algorithm caps columns to $C_e \approx 38.4$, the hard assignment (taken as argmax of $P_{t,:}$) sometimes exceeds 100 tokens for a single expert. This indicates that the soft capping does not guarantee that the hard assignment respects the limit – indeed, many tokens may have a soft probability spread across experts, and the argmax does not respect capacity. The discrepancy between the soft assignment matrix and the hard assignment used for load measurement represents a fundamental limitation of using Sinkhorn with a subsequent discretisation step.

Temperature effect: With $\tau = 0.1$, the softmax inside Sinkhorn is very sharp, making P nearly one-hot. Consequently, the column-capping step is ineffective because P is already almost binary. The low temperature was chosen to make the routing decisive, but it paradoxically undermines the capacity enforcement mechanism that is central to BBR's design. This finding suggests that a higher temperature might improve balance but at the cost of less decisive routing and potentially worse perplexity.

Prototype updates: Despite using EMA with $\alpha = 0.9$, the prototypes are still influenced by the noisy, unbalanced hard assignments, which may hinder the formation of clean expert specialisations. The EMA mechanism smooths out short-term fluctuations, but if the underlying hard assignments are persistently imbalanced, the prototypes will reflect that imbalance. This creates a feedback loop where imbalanced routing leads to biased prototypes, which in turn reinforce the imbalanced routing decisions.

5.4 Why Did Standard Router Succeed?

The Switch auxiliary loss works remarkably well at 16 experts and batch size 8. The loss coefficient 0.01 is low enough not to interfere with the main LM loss but high enough to keep the load nearly uniform. Moreover, the top-4 soft combination further smooths the routing and provides a richer gradient signal. The success of the standard router in this setting can be attributed to several factors: (1) with 16 experts and approximately 32 tokens per expert on average, the auxiliary loss has sufficient statistical power to guide the router toward balanced assignments; (2) the top-4 soft combination distributes gradient information to multiple experts per token, reducing the likelihood of expert collapse; and (3) the simplicity of the linear gate means there are fewer moving parts that can go wrong during training.

6 Discussion

6.1 Implications for MoE Routing Research

Our negative result carries several important lessons for the MoE routing research community. First, simplicity is powerful: for moderate expert counts ($E \leq 16$), the classical auxiliary loss combined with top-k soft combination is hard to beat. The additional complexity of Sinkhorn assignment, EMA prototypes, and anti-collapse losses did not translate into measurable improvements in this regime. Second, Sinkhorn with low temperature does not enforce hard capacity constraints. To guarantee capacity constraints, one would need to replace the argmax step with a greedy or Hungarian assignment, or keep the soft assignment throughout (i.e., always combine all experts, which is computationally expensive). Third, EMA prototypes may be unnecessary for the scale of models studied here. The additional complexity of maintaining prototypes did not translate into better routing decisions. A simple learned projection matrix (as in the standard linear router) works just as well, if not better, and is significantly simpler to implement and tune.

6.2 When Could BBR Be Useful?

BBR might still show benefits in more challenging regimes that were not tested in this study. First, at very high expert counts ($E \geq 32$), the auxiliary loss statistically cannot balance because of the few tokens per expert relative to the number of experts. In this regime, the Sinkhorn-based capacity constraint may provide a more reliable balancing mechanism. Second, under extreme synthetic imbalance where the standard router would collapse entirely, BBR's built-in balancing mechanisms could prevent such collapse. Third, in large-scale training with hundreds of billions of tokens, even small load imbalances accumulate over time and can lead to significant efficiency losses. The explicit capacity constraint of BBR may provide more consistent performance over very long training runs. We leave such experiments to future work.

6.3 Limitations

Our study is subject to several limitations that should be considered when interpreting the results. First, the study is limited to a single dataset (WikiText-103) and a relatively small model (256 hidden size). Scaling to billion-parameter models would require enormous computational resources and may yield different results. Second, the temperature and capacity slack were not exhaustively tuned; a grid search might improve BBR's results marginally. The choice of $\tau = 0.1$ was based on preliminary experiments, but a more systematic exploration of the temperature

parameter could reveal a more favourable operating point. Third, the test was performed only on a language modelling task; other domains (e.g., translation, vision) may show different behaviour due to different token distribution characteristics. Finally, the relatively small batch size (8 sequences × 64 tokens = 512 tokens) may limit the effectiveness of the Sinkhorn algorithm, which operates on the full batch of tokens. Larger batch sizes might enable the Sinkhorn algorithm to produce more balanced assignments.

7. Conclusion

We introduced Balanced Bilateral Routing (BBR), a novel routing mechanism for MoE layers that combines Sinkhorn assignment, EMA prototypes, and an anti‑collapse loss. In a fair, compute‑matched comparison against a standard Switch router with top‑4 soft combination, BBR underperformed on both perplexity (374.18 vs 345.68) and load balance (entropy 0.985 vs 0.995). The experiment demonstrates that for 16 experts on WikiText‑103, the classical auxiliary loss remains highly effective and simpler to implement. Our analysis reveals that the primary failure mode of BBR is the ineffectiveness of the Sinkhorn capacity constraint at low temperature, where the softmax becomes nearly one‑hot and the column‑capping step is bypassed. Additionally, the EMA prototype updates did not provide the anticipated stability benefits, as the prototypes are still influenced by the imbalanced hard assignments used for their update. We release all code and logs to support reproducibility. Future work should explore BBR in larger‑scale or more imbalanced settings where the standard router fails, and investigate exact capacity enforcement via Hungarian matching as an alternative to the soft Sinkhorn‑based approach.

8. Statements and declarations

AI Statement: No AI is used in this article/

Ethical Standards: All the ethical Research standards were followed while writing this Research Article.

Conflict of Interest: The authors state that they do not have any conflict of interest.

Human or animal involvement in the article: Nil

Data Availability: NA

References

1. Fedus, W., Zoph, B., & Shazeer, N. (2022). Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. JMLR, 23(120), 1–40. arXiv:2101.03961.
2. Lepikhin, D., Lee, H., Xu, Y., et al. (2021). GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding. ICLR 2021. arXiv:2006.16668.
3. Zoph, B., Bello, I., Kumar, S., et al. (2022). ST‑MoE: Designing Stable and Transferable Sparse Expert Models. arXiv:2202.08906.
4. Lewis, M., Zettlemoyer, L., & Levy, O. (2021). BASE Layers: Simplifying Training of Large, Sparse Models. ICML 2021. arXiv:2103.16716.
5. Zhou, Y., Lei, T., Liu, H., et al. (2022). Mixture‑of‑Experts with Expert Choice Routing. NeurIPS 2022. arXiv:2202.09368.
6. Puigcerver, J., Riquelme, C., Mustafa, B., & Houlsby, N. (2024). From Sparse to Soft Mixtures of Experts. ICLR 2024 (Spotlight). arXiv:2308.00951.
7. Liu, T., Blondel, M., Riquelme, C., & Puigcerver, R. (2024). Routers in Vision Mixture of Experts: An Empirical Study. TMLR. arXiv:2401.15969.
8. Shazeer, N., Mirhoseini, A., Maziarz, K., et al. (2017). Outrageously Large Neural Networks: The Sparsely‑Gated Mixture‑of‑Experts Layer. ICLR 2017. arXiv:1701.06538.
9. Nguyen, D. A., Ta, H. B., Le Duc, N., Nguyen, T. M., & Tran, T. (2025). Selective Sinkhorn Routing for Improved Sparse Mixture of Experts. arXiv:2511.08972.
10. Chi, Z., Dong, L., Huang, S., Dai, D., Ma, S., Patra, B., & Wei, F. (2022). On the Representation Collapse of

Sparse Mixture of Experts. NeurIPS 2022. arXiv:2204.09179.

11. DeepSeek‑AI. (2024). DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture‑of‑Experts Language Models. ACL 2024. arXiv:2401.06066.
12. He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum Contrast for Unsupervised Visual Representation Learning. CVPR 2020. arXiv:1911.05722.
13. Cuturi, M. (2013). Sinkhorn Distances: Lightspeed Computation of Optimal Transport. NIPS 2013, pp. 2292–2300. arXiv:1306.0895.
14. Omi, N., Sen, S., & Farhadi, A. (2025). Load Balancing Mixture of Experts with Similarity Preserving Routers. arXiv:2506.14038.
15. Kool, W., Maddison, C. J., & Mnih, A. (2021). Unbiased Gradient Estimation with Balanced Assignments for Mixtures of Experts. NeurIPS 2021 Workshop. arXiv:2109.11817.
16. Vesaghati, F. & Zareapoor, M. (2024). Training Mixture-of-Experts: A Focus on Expert-Token Matching (SVMoE). ICLR 2024.
17. Guo, H., Lu, H., Nan, G., et al. (2025). Advancing Expert Specialization for Better MoE. NeurIPS 2025. arXiv:2505.22323.