HiLD 2026: 4th Workshop on High-dimensional Learning Dynamics 1–60

Why Routers Freeze: Infinite Width Learning Dynamics for Mixture of Experts

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2026

Abstract

Mixture-of-Experts (MoE) models scale efficiently through sparse expert activation, but their training dynamics remain poorly understood. We study MoEs in the infinite-width limit with a fixed number of experts, a regime relevant to width-based scaling for hyperparameter transfer. Using Tensor Programs, we derive the training dynamics of soft and Top-K MoEs under SGD and Adam. We show that under the Standard Parameterisation, router logits diverge after one step of feature learning, causing softmax or sigmoid gates to saturate and router gradients to vanish. In contrast, we derive μP-MoE scaling which restores stability, but soft routing produces symmetric router dynamics: experts remain identically distributed and fail to specialise. For softmax routers, this symmetry also nullifies router gradients. We then show that Top-K routing has a qualitatively different effect: even when logits converge to a symmetric limit, finite-width fluctuations determine the selected experts, making Top-K an implicit symmetry-breaking mechanism. Experiments validate the predicted scaling laws and demonstrate hyperparameter transfer under μP-MoE.

1. Introduction

Mixture-of-Experts (MoE) architectures partition computation across multiple expert modules. A router maps each input to expert weights, allowing different experts to specialise on different regions of the input space (Jacobs et al., 1991; Eigen et al., 2013). In sparse MoEs, only a subset of experts is activated per input, decoupling parameter count from compute (Shazeer et al., 2017). This mechanism has become central to large-scale models (Team et al., 2024; Jiang et al., 2024; Dai et al., 2024; Yang et al., 2025; Han et al., 2024; Fan et al., 2022).

Despite their empirical success, MoEs remain difficult to train. Common failure modes include router instability, load imbalance, and representation collapse, where experts learn similar functions or become under-utilised (Zoph et al., 2022; Chi et al., 2022; Pham et al., 2024; Fedus et al., 2022; Do et al., 2025). Existing fixes, such as load-balancing losses, router z-losses, router noise, and alternative gates (Fedus et al., 2022; Zoph et al., 2022; Nguyen et al., 2024; Csordás et al., 2023; Wang et al., 2024, 2025), are effective but largely heuristic. This leaves open a basic question: how should MoEs be scaled so that router learning, feature learning, and expert specialisation survive at large width?

For dense neural networks, infinite-width theory has clarified the relationship between parameterisation, stability, and feature learning (Neal, 1996; Mei et al., 2018; Yang, 2020; Bordelon and Pehlevan, 2022; Yang and Hu, 2021; Yang and Littwin, 2023). In particular, Tensor Programs provide a framework for deriving finite-time training limits and have led to parameterisations such as μP, which enable stable feature learning and hyperparameter

transfer across widths (Yang and Hu, 2021; Yang et al., 2021). We extend this perspective to MoEs.

We study the limit where expert width $n \to \infty$ while the number of experts m remains fixed. This regime reflects practical settings where width can be scaled more readily than the number of experts, and it isolates expert-capacity scaling from expert-count scaling. Our main finding is a tension specific to MoEs: the Standard Parameterisation (SP) causes router saturation and frozen routers, whereas stable μP-MoE scaling removes specialisation in soft-routing MoEs by enforcing symmetric router dynamics. Sparse Top-K routing avoids this latter failure by acting as an implicit symmetry-breaking mechanism.

Our contributions are:

- We derive fixed-expert infinite-width limits for soft and Top-K MoEs under SGD and Adam using Tensor Programs.
- We show that SP causes router saturation and vanishing router gradients, while stable μP-MoE soft routing removes expert specialisation by enforcing symmetric router dynamics.
- We show that Top-K routing acts as a symmetry-breaking mechanism under μP-MoE, enabling stable feature learning and non-trivial expert differentiation; experiments confirm the predicted scaling laws and hyperparameter transfer.

2. Setup

We consider a single MoE block with input $\boldsymbol{\xi} \in \mathbb{R}^{d_{\text{in}}}$, expert width n, and fixed number of experts m. The MoE is defined as follows.

Definition 1 *We define an MoE block as*

$$\begin{aligned} \mathbf{h}^1 &= \mathbf{W}^1 \boldsymbol{\xi} \in \mathbb{R}^n, & \mathbf{h}^2 &= \mathbf{W}^2 \mathbf{h}^1 + \mathbf{b}^2 \in \mathbb{R}^m, \\ \tilde{\mathbf{h}}^2 &= \mathrm{G}(\mathbf{h}^2), & \mathbf{h}^{3,k} &= \mathbf{W}^{3,k} \mathbf{h}^1 \in \mathbb{R}^n, \\ \mathbf{h}^3 &= \sum_{k=1}^{m} \tilde{h}_k^2 \mathbf{h}^{3,k}, & \mathbf{f} &= \mathbf{W}^4 \mathbf{h}^3 \in \mathbb{R}^{d_{\text{out}}}, \end{aligned} \tag{1}$$

where G *chosen to be either softmax or sigmoid.*

We call $\mathbf{W}^2$ and $\mathbf{h}^2$ the router weights and logits, and $\mathbf{W}^{3,k}$ the kth expert weights. The bias $\mathbf{b}^2$ represents a generalisation to a learned router bias, injected router noise (Shazeer et al., 2017), or load-balancing bias (Wang et al., 2024).

For Top-K MoEs, the router logits are sparsified before applying the gate:

$$\hat{h}_k^2 = \begin{cases} h_k^2, & k \in \mathrm{TopK}(\mathbf{h}^2), \\ -\infty, & \text{otherwise}, \end{cases} \qquad \tilde{\mathbf{h}}^2 = \mathrm{G}(\hat{\mathbf{h}}^2).$$

Only experts with non-zero gate values are executed. We use the standard straight-through estimator for Top-K gradients (Shazeer et al., 2017).

We study the Tensor Program scaling regime in which $n \to \infty$ while m, depth, batch size, data dimension, and training time remain fixed. We use a layerwise *bcd* parameterisation: weights are initialised as

$$W_{ij}^{\ell} \sim \mathcal{N}(0, n^{-2b_\ell}),$$

and updates are

$$\text{SGD:} \quad \mathbf{W}^{\ell} \leftarrow \mathbf{W}^{\ell} - \eta n^{-c_\ell} \nabla_{\mathbf{W}^{\ell}} \mathcal{L}, \qquad \text{Adam:} \quad \mathbf{W}^{\ell} \leftarrow \mathbf{W}^{\ell} - \eta n^{-c_\ell} \frac{\widehat{m}_\ell}{\sqrt{\widehat{v}_\ell} + n^{-d_\ell}\epsilon}.$$

We ask whether a parameterisation yields four properties in the infinite-width limit: stable activations, non-vanishing feature updates, faithful Adam updates, and non-trivial expert specialisation. Formal definitions and the full scaling tables are deferred to Appendix B and Tables 1 and 2.

3. Soft-Routing MoEs

We first consider soft-routing MoEs, where every expert is active. The main conclusions are that SP is unstable, while the stable μP-MoE limit is too symmetric to permit expert specialisation without an additional symmetry-breaking mechanism (such as a router bias).

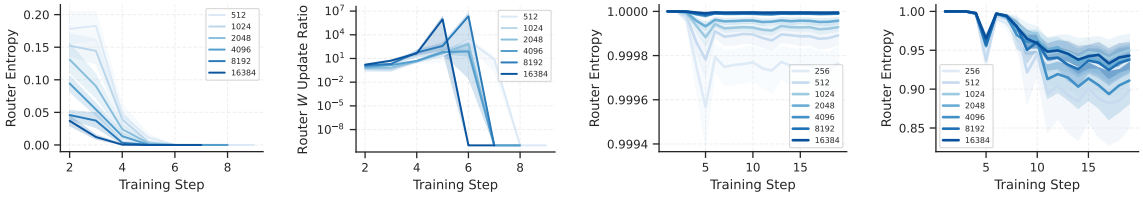
3.1. SP causes router saturation

The Standard Parameterisation initializes $b_1 = 0$ and $b_\ell = 1/2$ for $\ell > 1$, and uses a global learning-rate scaling $c_\ell = c$ (He et al., 2015; Yang and Hu, 2021). This gives $\Theta(1)$ activations at initialization, but it does not give stable MoE training dynamics.

Proposition 2 (SP router saturation) *Consider a soft MoE with softmax or sigmoid gating under SP. For any learning-rate scaling that gives $\Theta(1)$ feature updates in the first layer after one step, the router logits diverge as $n \to \infty$. Consequently, after one step the softmax gate converges to a one-hot vector, the sigmoid gate converges to a binary vector, and the router gradient vanishes.*

The mechanism is a correlated update term. Under SP, router weights have entries of scale $\Theta(n^{-1/2})$. After one step, the update to router logits contains $\sum_{j=1}^{n} (\mathbf{W}_0^2)_{:,j} (\mathbf{h}_1^1 - \mathbf{h}_0^1)_j$. The feature update $(\mathbf{h}_1^1 - \mathbf{h}_0^1)_j$ is correlated with the forward weights $(\mathbf{W}_0^2)_{:,j}$, so the sum scales as $\Theta(n^{1/2})$ rather than $\Theta(1)$. Thus $\mathbf{h}_1^2$ diverges with width. Softmax then saturates to a one-hot gate and sigmoid to a binary gate, making the gate Jacobian vanish. The router expresses a strong preference for experts, but this preference is frozen after the first step rather than learned.

This provides a partial explanation for why auxiliary losses, especially z-losses, are useful in practice: they are explicitly designed to counteract the empirically observed large router logits (Fedus et al., 2022; Zoph et al., 2022). In Figure 1(a,b), the normalized router entropy decreases toward zero under SP, and the relative router-weight update norm vanishes with width, matching Theorem 2.



(a) SP softmax entropy (b) SP router update (c) μP-MoE soft routing (d) μP-MoE Top-K

Figure 1: Empirical validation of the fixed-expert scaling predictions. Under SP, router entropy collapses and router updates vanish as width grows. Under μP-MoE with softmax soft routing, entropy remains maximal, indicating uniform expert allocation. Under μP-MoE with Top-K routing, entropy decreases without an explicit router bias, showing that Top-K breaks the soft-routing symmetry.

3.2. μP-MoE (without router bias) gives stability without specialisation

Deriving the stable infinite-width limit gives a μP-MoE scaling: the router is scaled as an output-like layer, while each expert is scaled as a matrix-like $n \times n$ layer. Full SGD and Adam scalings are given in Tables 1 and 2. This scaling resolves the SP instability, but soft routing has a different failure mode, without explicit symmetry breaking from a router bias/noise term.

Proposition 3 (μP-MoE soft-routing symmetry) *Consider a soft MoE without router bias $\mathbf{b}^2$, parameterised according to μP-MoE. As $n \to \infty$, the router scores are identical across experts at initialization: $\tilde{\mathbf{h}}_0^2 \to m^{-1}\mathbf{1}$ for softmax and $\tilde{\mathbf{h}}_0^2 \to 2^{-1}\mathbf{1}$ for sigmoid. For every finite training step, the router remains symmetric, $\tilde{\mathbf{h}}_t^2 = C_t\mathbf{1}$, and all experts remain identically distributed. With softmax gating, the router gradient also vanishes.*

The proof is in Appendix D.2. The key point is that router feature learning requires router weights of scale $\Theta(n^{-1})$. In the fixed-expert limit, this makes the router logits deterministic and identical across experts. Since expert weights are also initialised from the same distribution, the downstream gradient into every router logit converges to the same deterministic limit. Router updates therefore remain synchronous, and the experts receive identical gradients. Softmax routing is even more restrictive: when all logits are equal, the softmax Jacobian cancels the symmetric gradient signal, so the router gradient vanishes.

Thus, μP-MoE solves stability but not specialisation for soft-routing MoEs. Sigmoid routing permits non-zero router updates, but these updates are identical across experts; softmax routing removes even this router feature learning. A router bias or injected router noise avoids this degeneracy by breaking symmetry at initialisation. Empirically, Figure 1(c,d) shows that under μP-MoE without router bias, softmax soft-routing entropy remains maximal and the router update norm follows the predicted vanishing width scaling.

4. Sparse MoEs

We now consider Top-K sparse routing, the standard mechanism used to reduce MoE compute cost (Shazeer et al., 2017; Jiang et al., 2024). Top-K routing inherits the SP instability:

sparsifying the gradient changes which experts are updated, but not the $\Theta(n^{1/2})$ magnitude of the correlated router-logit update. Hence the router saturation mechanism of Theorem 2 still applies under SP.

The μP-MoE case is qualitatively different. In soft routing and without a router bias, symmetric logits at initialisation lead to symmetric gradients which persisted throughout training. In Top-K routing, however, the selected expert set depends on the ordering of logits. Even at initialisation when logits converge to the same deterministic limit, their finite-width fluctuations can determine the Top-K ordering.

Proposition 4 (Top-K symmetry breaking) *Consider the router logits $\mathbf{h}^2(n) \in \mathbb{R}^m$ under μP-MoE. At initialisation $\mathbf{h}^2(n) \to C\mathbf{1}$ almost surely and, for finite n, $\mathbf{h}^2(n) = C\mathbf{1} + \sigma_n \epsilon(n)$ with $\sigma_n \to 0$, where $\epsilon(n)$ converges in distribution to a non-degenerate Gaussian vector ϵ. Then*

$$\mathrm{TopK}(\mathbf{h}^2(n)) \Rightarrow \mathrm{TopK}(\epsilon).$$

Thus the Top-K index set remains random.

The proof is in Appendix E. This result shows that Top-K is not merely a computational device, but also a symmetry-breaking mechanism. Under μP-MoE, router logits can converge to a symmetric deterministic limit, but the Top-K operator acts on the rescaled finite-width fluctuations. Different experts are therefore selected, yielding expert-dependent gradients and preventing the synchronous soft-routing dynamics of Theorem 3.

We derive the infinite width limit for Top-K MoEs in Appendix F and show that μP-MoEs satisfy stability, feature learning and faithfulness (for Adam). As the router does not remain symmetric, like in the soft case, it also allows for expert differentiation.

We also test the practical implication of the stable scaling: hyperparameter transfer. Following the usual μP methodology (Yang et al., 2021), we tune hyperparameters on the smallest model and transfer them across widths. For a Mixtral-style Top-K MoE trained with Adam, μP-MoE transfers learning rates across widths, whereas SP does not. These results, together with additional coordinate checks and experiments with router bias/noise, are reported in Appendices H and H.4.

5. Discussion

We analysed MoE training in the fixed-expert infinite-width limit and identified two distinct degeneracies. Under SP, router logits diverge after one step of feature learning, saturating the gate and freezing router updates. Under stable μP-MoE scaling, soft routing avoids this instability but becomes symmetric across experts: experts remain identically distributed and fail to specialise. For softmax routing, the same symmetry also nullifies the router gradient.

Top-K routing changes the picture. Even when router logits converge to a symmetric limit, the Top-K operator can act on finite-width fluctuations and produce non-deterministic expert selections. This makes Top-K an implicit symmetry-breaking mechanism, explaining a benefit of sparse routing beyond computational efficiency. Overall, stable large-width MoE training requires more than correct feature-learning scaling: it also requires controlled symmetry breaking, through Top-K, router bias, router noise, or related mechanisms.

References

Patrick Billingsley. *Convergence of probability measures*. John Wiley & Sons, 2013.

Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. *Advances in Neural Information Processing Systems*, 35, 2022.

Blake Bordelon, Hamza Chaudhry, and Cengiz Pehlevan. Infinite limits of multi-head transformer dynamics. *Advances in Neural Information Processing Systems*, 37, 2024a.

Blake Bordelon, Lorenzo Noci, Mufan Bill Li, Boris Hanin, and Cengiz Pehlevan. Depthwise hyperparameter transfer in residual networks: Dynamics and scaling limit. In *The Twelfth International Conference on Learning Representations*, 2024b.

Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, et al. On the representation collapse of sparse mixture of experts. *Advances in Neural Information Processing Systems*, 35, 2022.

Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. Approximating two-layer feedforward networks for efficient transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Damai Dai, Chengqi Deng, Chenggang Zhao, Rx Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

Nolan Dey, Bin Claire Zhang, Lorenzo Noci, Mufan Li, Blake Bordelon, Shane Bergsma, Cengiz Pehlevan, Boris Hanin, and Joel Hestness. Don't be lazy: Completep enables compute-efficient deep transformers. *arXiv preprint arXiv:2505.01618*, 2025.

Giang Do, Hung Le, and Truyen Tran. Simsmoe: Toward efficient training mixture of experts via solving representational collapse. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025.

David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013.

Zhiwen Fan, Rishov Sarkar, Ziyu Jiang, Tianlong Chen, Kai Zou, Yu Cheng, Cong Hao, Zhangyang Wang, et al. M^{3}vit: Mixture-of-experts vision transformer for efficient multi-task learning with model-accelerator co-design. *Advances in Neural Information Processing Systems*, 35, 2022.

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), 2022.

Moritz Haas, Jin Xu, Volkan Cevher, and Leena Chennuru Vankadara. $\boldsymbol{\mu}\mathbf{P^2}$: Effective sharpness aware minimization requires layerwise perturbation scaling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Moritz Haas, Sebastian Bordt, Ulrike von Luxburg, and Leena Chennuru Vankadara. On the surprising effectiveness of large learning rates under standard width scaling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Xing Han, Huy Nguyen, Carl Harris, Nhat Ho, and Suchi Saria. Fusemoe: Mixture-of-experts transformers for fleximodal fusion. *Advances in Neural Information Processing Systems*, 37, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 2015.

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

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

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.

Tianze Jiang, Blake Bordelon, Cengiz Pehlevan, and Boris Hanin. Hyperparameter transfer with mixture-of-expert layers. *arXiv preprint arXiv:2601.20205*, 2026.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

Jan Małaśnicki, Kamil Ciebiera, Mateusz Boruń, Maciej Pióro, Jan Ludziejewski, Maciej Stefaniak, Michał Krutul, Sebastian Jaszczur, Marek Cygan, Kamil Adamczewski, et al. μ-parametrization for mixture of experts. *arXiv preprint arXiv:2508.09752*, 2025.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33), 2018.

Radford M Neal. Bayesian learning for neural networks, 1996.

Huy Nguyen, Nhat Ho, and Alessandro Rinaldo. Sigmoid gating is more sample efficient than softmax gating in mixture of experts. *Advances in Neural Information Processing Systems*, 37, 2024.

Quang Pham, Giang Do, Huy Nguyen, TrungTin Nguyen, Chenghao Liu, Mina Sartipi, Binh T Nguyen, Savitha Ramasamy, Xiaoli Li, Steven Hoi, et al. Competesmoe–effective training of sparse mixture of experts via competition. *arXiv preprint arXiv:2402.02526*, 2024.

Joan Puigcerver, Carlos Riquelme Ruiz, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. In *The Twelfth International Conference on Learning Representations*, 2024.

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.

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

Leena Chennuru Vankadara, Jin Xu, Moritz Haas, and Volkan Cevher. On feature learning in structured state space models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

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.

Ziteng Wang, Jun Zhu, and Jianfei Chen. Remoe: Fully differentiable mixture-of-experts with reLU routing. In *The Thirteenth International Conference on Learning Representations*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Greg Yang. Wide feedforward or recurrent neural networks of any architecture are gaussian processes. *Advances in Neural Information Processing Systems*, 32, 2019.

Greg Yang. Tensor programs iii: Neural matrix laws. *arXiv preprint arXiv:2009.10685*, 2020.

Greg Yang and Edward J Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*. PMLR, 2021.

Greg Yang and Etai Littwin. Tensor programs ivb: Adaptive optimization in the infinite-width limit. *arXiv preprint arXiv:2308.01814*, 2023.

Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: tuning large neural networks via zero-shot hyperparameter transfer. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, 2021.

Greg Yang, Dingli Yu, Chen Zhu, and Soufiane Hayou. Tensor programs VI: Feature learning in infinite depth neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.

Appendix Roadmap: The Logic of Router Dynamics

This appendix provides the rigorous derivations for the infinite-width limits of Mixture-of-Experts (MoE) architectures. To navigate the theoretical contributions, we structure the analysis into a narrative arc: establishing the Tensor Program machinery, demonstrating the inevitable failure of standard soft routing, and deriving the conditional limit for Top-K routing.

- **Full Related Work**
- **Desirable Asymptotics**
  Desirable behaviour as width increases.
- **Tensor Program for Soft MoE Architectures (Appendix C)**
  Develops the tensor-program representation of the forward/backward passes and training dynamics for soft-routed MoEs.
  - **Tensor Program for Soft MoE with SGD (Appendix C.1)**
    Specifies the NETSOR$\top^+$ program for SGD training, including all primitives and scaling conventions.
  - **Corresponding infinite-width limit for Soft MoE with SGD (Appendix C.2)**
    Derives the deterministic and distribution limit objects (Master Theorem recursion) induced by the SGD tensor program.
  - **Tensor Program for Soft MoE with Adam (Appendix C.3)**
    Extends the program to Adam using NE$\otimes$OR$\top$, introducing the OuterNonlin structure needed for adaptive updates.
  - **Corresponding infinite-width limit for Soft MoE with Adam (Appendix C.4)**
    Computes the Adam infinite-width dynamics, including the additional limit rule for OuterNonlin and faithfulness constraints.
- **Deriving Results for Soft MoE (Appendix D)**
  Uses the above limits to prove the main qualitative claims about soft routing: collapse under SP and lack of expert specalisation under μP-style stability.
  - **Router Collapse with Standard Parameterisation (Appendix D.1)**
    Shows that under SP scaling the router logits diverge, causing gating saturation and vanishing router gradients.
  - **Derivation of μP-MoE and Lack of Expert Specialiaation (Appendix D.2)**
    Proves that under stable μP-style scaling the router updates become identical across experts, preventing specialisation in the limit.
- **Top-K Mask Depends on Finite-Width Noise (Appendix E)**
  Establishes that when logits collapse to a common value (in the first forward pass under μP-MoE), the Top-K selection remains random in the limit because it is determined by vanishing-but-ordering noise.

- **Tensor Program for Top-K MoE (Appendix F)**
  Develops the tensor-program representation of the forward/backward passes and training dynamics for Top-K routed MoEs.
  - **Tensor Program for Top-K MoE (Appendix F.1)**
    Defines the Top-K training program.
  - **Corresponding infinite-width limit for Top-K MoE (Appendix F.2)**
    Derives the resulting infinite-width limit.
- **Additional Experiments (Appendix H)**
  Collects supplementary empirical results that validate the theoretical predictions across parametrisations.

Appendix A. Full Related Work

MoEs were originally introduced to encourage specialisation by decomposing the parameter space into modular experts (Jacobs et al., 1991). Lately, their application is driven by the ability to decouple parameter count from compute cost via sparse routing (Shazeer et al., 2017; Jiang et al., 2024). Despite their efficiency, MoEs are notoriously difficult to train compared to their dense counterparts. Practitioners frequently encounter exploding router logits (Zoph et al., 2022; Puigcerver et al., 2024) and representation collapse — where experts become redundant or underutilised (Chi et al., 2022; Pham et al., 2024). Mitigation of these issues has largely come from empirical heuristics, including auxiliary losses such as the router z-loss (Zoph et al., 2022) and load-balancing regularisers (Fedus et al., 2022), and alternative gating mechanisms, such as sigmoid and ReLU routing (Nguyen et al., 2024; Csordás et al., 2023; Wang et al., 2025). While effective in practice, these solutions are often ad-hoc, and the underlying theoretical mechanisms driving these instabilities remain poorly understood.

In contrast to the heuristic fixes used in MoEs, the training dynamics of dense neural networks have been rigorously characterised through infinite width limits. Early work focused on the "lazy" regime (e.g. NNNGP, NTK), where weights stay close to initialization and feature learning is suppressed (Jacot et al., 2018; Lee et al., 2019; Neal, 1996). More recently, the Tensor Programs framework (Yang, 2020; Yang and Hu, 2021) and related mean-field approaches (Mei et al., 2018; Bordelon and Pehlevan, 2022) have characterized the "feature learning" (maximal update) limit. This theory led to the μP, which prescribes layer-wise scaling rules to ensure stable feature learning and enables zero-shot hyperparameter transfer across model scales (Yang and Hu, 2021; Yang et al., 2021). These limits have been extended to various architectures (Bordelon et al., 2024a; Haas et al., 2024; Vankadara et al., 2024), and for scaling other quantities such as depth (Dey et al., 2025; Yang et al., 2024). An important empirical consequence of such analysis is the ability for zero shot hyperparameter transfer across width and depth (Yang et al., 2021; Bordelon et al., 2024b).

While independent and concurrent empirical work has attempted to apply μP directly to MoEs (Małaśnicki et al., 2025), they do not derive the limiting dynamics and thus do not characterise the breakdown of expert specialisation without the router bias, or that Top-K acts as a symmetry breaker. Concurrent work by Jiang et al. (2026) studies hyperparameter transfer for transformer MoEs and proposes scaling rules for transferring learning rates and initialisation scales as width, depth, expert hidden size, and the number of experts are varied. Our work is complementary. First, the two limits are not interchangeable: Jiang et al. (2026) take a joint proportional limit under *gradient flow dynamics* in which width, depth, expert size, and expert count diverge together, whereas we study a width-only Tensor Program limits for both soft and Top-K, sigmoid and softmax routing under SGD and Adam with a fixed number of experts. Since iterated limits do not commute in general, their joint limit does not recover the fixed-expert width limit we study. Second, the two limits are qualitatively different in their predictions: in the proportional limit of Jiang et al. (2026), experts become asymptotically independent, whereas in our width limit experts remain strongly correlated. This fixed-expert regime lets us isolate router-specific degeneracies: under SP the router logits saturate and router gradients vanish, while under stable μP-heuristic scaling soft routing remains symmetric across experts unless an explicit symmetry-breaking mechanism is present.

Appendix B. Desirable Asymptotics

Building on the asymptotic framework of Yang and Hu (2021); Yang and Littwin (2023), which formalises stability, maximal feature learning, and faithfulness, we introduce a fourth desideratum specific to MoE architectures: expert specialisation. We aim to identify the class of *bcd*-parameterizations under which network training dynamics satisfy these asymptotic properties. We denote the value of a quantity at training step t using a subscript, e.g. $\mathbf{h}_t^\ell$.

We first require that activations do not diverge with increasing width.

Definition 5 (Stability) *A network is stable if, for every layer ℓ, the activations satisfy $\mathbf{h}^\ell = \mathcal{O}(1)$ at initialisation and throughout training.*

Next, to ensure feature learning is taking place, we require that the change in activations $\mathbf{h}_t - \mathbf{h}_{t-1}$ at each time step do not vanish with increasing width (Yang and Hu, 2021). Activation updates can be decomposed into two terms

$$\mathbf{h}_t^\ell - \mathbf{h}_{t-1}^\ell = \underbrace{\mathbf{W}_{t-1}^\ell(\mathbf{h}_t^{\ell-1} - \mathbf{h}_{t-1}^{\ell-1})}_{\text{Propagating Updates}} + \underbrace{(\mathbf{W}_t^\ell - \mathbf{W}_{t-1}^\ell)\mathbf{h}_t^{\ell-1}}_{\text{Effective Updates}},$$

where the propagating updates are contributions from the update to the previous layer features, and effective updates are the contribution due to the current weight updates. *Maximal* feature learning requires that both the above terms do not vanish with increasing width. For conciseness we refer to this as simply feature learning throughout.

Definition 6 (Maximal Feature Learning) *A layer ℓ exhibits maximal feature learning if both effective and propagating updates after one step of training scale $\Omega(1)$ (Yang and Hu, 2021).*

For Adam updates, we require that the gradients have a non-trivial $\Theta(1)$ effect on the updates, a condition referred to as *faithfulness* (Yang and Littwin, 2023).

Definition 7 (Adam Faithfulness) *We say a bcd-parametrisation is faithful if the inputs to the Adam update function have scale $\Theta(1)$ for every layer's parameter updates.*

Finally, we introduce a desideratum specific to MoEs. Expert specialisation is a central tenet of MoE architectures; accordingly, we require that such specialisation does not cease at infinite width. A *necessary* condition for expert specialisation is that, for a given input, the router exhibits a preference for certain experts over others. We use this to diagnose regimes where the experts cannot specialise.

Definition 8 (Expert Specialisation) *We say that a bcd-parametrisation exhibits expert specialisation only if $\mathbf{h}_t^2 \neq C_t\mathbf{1}$ for any constants C_t for all steps $t > 0$.*

Appendix C. Tensor Program for Soft MoE

Here, we derive the Tensor program and the corresponding infinite width limit for soft MoEs (Theorem 1) for both softmax and sigmoid gating. We start by deriving the TP for SGD (Appendix C.1) and the infinite width limit (Appendix C.2). Then, we derive the TP for Adam (Appendix C.3) and the infinite width limit (Appendix C.4).

C.1. Tensor Program for Soft MoE with SGD

Matrices, Vectors and Scalars for the program: As we consider the limit of $n \to \infty$, we refer to quantities that have both dimensions scaled ($\in \mathbb{R}^{n \times n}$) as 'matrices' in the program. Similarly, if only one of the dimensions is taken to infinity, with the other being held constant (e.g. $\in \mathbb{R}^{\bullet \times n}$ for some constant dimension $\bullet$), then we refer to the quantity as a 'vector' in the program. Finally, if both dimensions of a quantity are held constant, we refer to it as a 'scalar' in the program. Objects like $\mathbf{W}^2 \in \mathbb{R}^{m \times n}$, are treated as multiple (m) vector objects.

Notation: For all vectors in the program $\mathbf{h}$, we write $\delta \mathbf{h}_t := \theta_{\delta h_t}^{-1}(\mathbf{h}_t - \mathbf{h}_{t-1})$ as the normalised change in the features, where $\theta_{\delta h_t}$ is the scale of $(\mathbf{h}_t - \mathbf{h}_{t-1})$. Thus, $\delta \mathbf{h}_t$ scales $\Theta(1)$. We denote scalars and vectors at training step t as $\mathbf{h}_t = \mathbf{h}_0 + (\mathbf{h}_1 - \mathbf{h}_0) + \ldots + (\mathbf{h}_t - \mathbf{h}_{t-1})$. We define $\boldsymbol{\chi}_t := \frac{\partial \mathcal{L}_t}{\partial f_t} \in \mathbb{R}^{d_{\text{out}} \times 1}$ which we assume is always $\Theta(1)$. This holds for common loss functions (such as mean squared error and cross entropy) as long as the function is stable (does not blow up with width). For a gradient term $\frac{\partial \mathbf{f}}{\partial \mathbf{h}}$ with scale $\theta_{\partial h}$, we define the normalised gradient $\mathrm{d}\mathbf{h} := \theta_{\partial h}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{h}}$, such that $\mathrm{d}\mathbf{h}$ scales $\Theta(1)$. We also denote $\theta'_x := n\theta_x$. We reserve subscript for indexing training time $\mathbf{h}_t$, and index a component with brackets, for example $(\mathbf{h}_t)_\alpha$ denotes the α component of $\mathbf{h}$ at step t.

Tensor Program Operations: For the SGD limit, we use NETSOR$\top^+$ (Yang, 2020; Yang and Hu, 2021). The program contains a sequence of $\mathbb{R}^n$ vectors and $\mathbb{R}$ scalars generated from one of the following ways from initial vectors and scalars

- `MatMul`: Given $\mathbf{W} \in \mathbb{R}^{n \times n}$ and $\mathbf{v} \in \mathbb{R}^n$, new vectors are generated as $\mathbf{W}\mathbf{v}$ or $\mathbf{W}^T\mathbf{v}$.
- `Nonlin`: Given $\phi : \mathbb{R}^k \times \mathbb{R}^l \to \mathbb{R}$, scalars $\theta_1, \ldots, \theta_l \in \mathbb{R}$ and vectors $\mathbf{v}^1, \ldots, \mathbf{v}^k \in \mathbb{R}^n$, a new vector can be generated as

  $$\phi(\mathbf{v}^1, \ldots, \mathbf{v}^k; \theta_1, \ldots, \theta_l) \in \mathbb{R}^n$$

  where ϕ applies coordinate wise on $\mathbf{v}^1_\alpha$.
- `Moment`: Given the same as above, a new scalar can be generated

  $$\frac{1}{n} \sum_{\alpha=1}^{n} \phi(\mathbf{v}^1_\alpha, \ldots, \mathbf{v}^k_\alpha; \theta_1, \ldots, \theta_l) \in \mathbb{R}.$$

The only assumption on ϕ operations are that they are pseudo-Lipschitz (Yang and Hu, 2021, Assumption F.4).

We first make the assumption that the parametrisation values are chosen such that the network is stable throughout training — no quantity diverges with increasing width.

Setting 9 (Stable parametrisation) *We only consider values of $\{b_\ell, c_\ell\}_\ell$ for each layer ℓ such that the network is stable throughout training.*

This implies that no quantity diverges with increasing width, leading to well-defined limits.

We will write the forward pass, backward pass, and SGD updates as tensor programs (Yang and Hu, 2021). This requires all initial matrices ($n \times n$ shape) to be samples from $\mathcal{N}\left(0, \frac{1}{n}\right)$, and all initial vectors from $\mathcal{N}(0, 1)$. All $\theta_\bullet$ parameters are defined as initial scalars.

Network: We define the network as following,

$$\mathbf{h}^1 = \mathbf{W}^1\boldsymbol{\xi}, \qquad \in \mathbb{R}^{n\times 1} \tag{2}$$
$$\mathbf{h}^2 = \mathbf{W}^2\mathbf{h}^1 + \mathbf{b}^2, \qquad \in \mathbb{R}^{m\times 1} \tag{3}$$
$$\tilde{\mathbf{h}}^2 = \mathrm{G}(\mathbf{h}^2), \qquad \in [0,1]^{m\times 1} \tag{4}$$
$$\mathbf{h}^{3,k} = \mathbf{W}^{3,k}\mathbf{h}^1, \qquad \in \mathbb{R}^{n\times 1} \tag{5}$$
$$\mathbf{h}^3 = \sum_{k=1}^{m} \mathbf{h}^{3,k}\tilde{h}^2_k, \qquad \in \mathbb{R}^{n\times 1} \tag{6}$$
$$\mathbf{f} = \mathbf{W}^4\mathbf{h}^3, \qquad \in \mathbb{R}^{d_{\text{out}}\times 1} \tag{7}$$

where G is the gating function (either softmax or sigmoid), we refer to $\mathbf{h}^2$ as the router, each $\mathbf{h}^{3,k}$ as expert k, and $\mathbf{f}$ as the function output. We follow a simpler variation of the *abc* parametrisation (Yang and Hu, 2021). The initial scalars, vectors and matrices in the program are defined below,

$$\mathbf{W}^1 = n^{-b_1}\overline{\mathbf{W}^1} = \theta_{W_0^1}\overline{\mathbf{W}^1}, \qquad \mathbf{W}^1 \in \mathbb{R}^{n\times d_{\text{in}}} \qquad \overline{W^1_{ij}} \sim \mathcal{N}(0,1), \tag{8}$$
$$\mathbf{W}^2 = n^{-b_2}\overline{\mathbf{W}^2} = \theta_{W_0^2}\overline{\mathbf{W}^2}, \qquad \mathbf{W}^2 \in \mathbb{R}^{m\times n} \qquad \overline{W^2_{ij}} \sim \mathcal{N}(0,1), \tag{9}$$
$$\mathbf{W}^{3,k} = n^{\frac{1}{2}-b_3}\overline{\mathbf{W}^{3,k}} = \theta_{W_0^3}\overline{\mathbf{W}^3_k}, \qquad \mathbf{W}^{3,k} \in \mathbb{R}^{n\times n} \qquad \overline{W^{3,k}}_{ij} \sim \mathcal{N}(0,n^{-1}), \tag{10}$$
$$\mathbf{W}^4 = n^{-b_4}\overline{\mathbf{W}^4} = \theta_{W_0^4}\overline{\mathbf{W}^4}, \qquad \mathbf{W}^4 \in \mathbb{R}^{d_{\text{out}}\times n} \qquad \overline{W^4_{ij}} \sim \mathcal{N}(0,1). \tag{11}$$

Note that we only consider the limit $n \to \infty$ with other factors held fixed, and thus $\{\mathbf{W}^{3,k}\}_{k=1}^m$ are the only initial matrices in the program ($n \times n$).

Initial Matrices: $\overline{\mathbf{W}^{3,k}}$.

Initial Vectors: $\overline{\mathbf{W}^1}, \overline{\mathbf{W}^2}, \overline{\mathbf{W}^4}$, where $\overline{\mathbf{W}^1}$ is a set of d_{in} initial vectors, $\overline{\mathbf{W}^2}$ is a set of m initial vector and $\overline{\mathbf{W}^4}$ is a set of d_{out} initial vectors.

Initial Scalars: All $\theta_\bullet$ parameters. The router bias $\mathbf{b}^2$ is a set of m initial scalars sampled from some density such that $\mathbb{P}(b_i = b_j) = 0$. In practice, this is satisfied by sampling from$\mathcal{N}(0, \sigma_b^2)$ where σ_b^2 is some hyperparameter.

Noise and Learnable Bias: We consider two separate cases here:

1. Case 1: A different noise is added at each time step $\{\mathbf{b}^2_t\}_{t=1}^T$ (Shazeer et al., 2017).
2. Case 2: The term $\mathbf{b}^2_0$ is learned, either through gradients of the loss or load balancing considerations (Wang et al., 2024).

   Note that the updates below are written only for learnable biases from gradient of the loss. For the noise case, a new noise is sampled every forward pass.

Conditioned on router bias/noise: The standard Tensor Programs Master Theorem assumes initial scalars converge almost surely to deterministic limits. A router bias/noise vector sampled from a continuous distribution does not satisfy this assumption unconditionally. Instead, we use a conditional formulation: we first sample the finite collection of router bias/noise scalars and condition on their realised values. Conditional on these values, they are treated as fixed scalar constants in the tensor program. Thus, for any fixed finite time

horizon T, we condition on $\{\mathbf{b}_t^2\}_{t=0}^T$ in the noise case, or on the initialisation of $\mathbf{b}_0^2$ in the learnable-bias case. The resulting infinite-width limits are deterministic conditional on the realised bias/noise values, but random unconditionally through their dependence on those values. Practically, this corresponds to coupling widths by using the same sampled router bias/noise values across widths.

First Forward Pass:

$$\mathbf{h}_0^1 = \theta_{W_0^1}\overline{\mathbf{W}_0^1}\boldsymbol{\xi}, \qquad (\mathbf{h}_0^1)_\alpha = \phi\left((\overline{\mathbf{W}_0^1})_{\alpha 1}, \ldots, (\overline{\mathbf{W}_0^1})_{\alpha d_{\text{in}}}; \xi_1, \ldots, \xi_{d_{\text{in}}}, \theta_{W_0^1}\right) = \sum_{j=1}^{d_{\text{in}}} \theta_{W_0^1}(\overline{\mathbf{W}_0^1})_{\alpha j}\xi_j, \quad \texttt{(Nonl}$$

$$\mathbf{h}_0^2 = \theta'_{W_0^2}\frac{1}{n}\overline{\mathbf{W}_0^2}\mathbf{h}_0^1 + \mathbf{b}_0^2, \qquad (\mathbf{h}_0^2)_k = \frac{1}{n}\sum_{\alpha=1}^{n}\phi\left((\overline{\mathbf{W}_0^2})_{k\alpha}, (\mathbf{h}_0^1)_\alpha; \theta'_{W_0^2}, \mathbf{b}_0^2\right) = \frac{1}{n}\theta'_{W_0^2}\sum_{\alpha=1}^{n}(\overline{\mathbf{W}_0^2})_{k\alpha}(\mathbf{h}_0^1)_\alpha + \mathrm{b}_0^2, \quad \texttt{(Mome}$$

$$\tilde{\mathbf{h}}_0^2 = \mathrm{G}(\mathbf{h}_0^2), \qquad \texttt{(Moment, See G)}$$

$$\hat{\mathbf{h}}_0^{3,k} = \overline{\mathbf{W}^{3,k}}\mathbf{h}_0^1, \qquad \texttt{(MatMul)}$$

$$\mathbf{h}_0^{3,k} = \theta_{W_0^3}\hat{\mathbf{h}}_0^{3,k}, \qquad (\mathbf{h}_0^{3,k})_\alpha = \phi\left((\hat{\mathbf{h}}_0^{3,k})_\alpha; \theta_{W_0^3}\right) = \theta_{W_0^3}(\hat{\mathbf{h}}_0^{3,k})_\alpha \quad \texttt{(Nonl}$$

$$\mathbf{h}_0^3 = \sum_{k=1}^{m}\mathbf{h}_0^{3,k}\tilde{h}_{0,k}^2, \qquad (\mathbf{h}_0^3)_\alpha = \phi\left((\mathbf{h}_0^{3,1})_\alpha, \ldots, (\mathbf{h}_0^{3,m})_\alpha; \tilde{h}_{0,1}^2, \ldots, \tilde{h}_{0,m}^2\right) = \sum_{k=1}^{m}(\mathbf{h}_0^{3,k})_\alpha\tilde{h}_{0,k}^2 \quad \texttt{(Nonl}$$

$$\mathbf{f}_0 = \theta'_{W_0^4}\frac{1}{n}\overline{\mathbf{W}_0^4}\mathbf{h}_0^3, \qquad (\mathbf{f}_0)_\beta = \frac{1}{n}\sum_{\alpha=1}^{n}\phi\left((\overline{\mathbf{W}_0^4})_{\beta\alpha}, (\mathbf{h}_0^3)_\alpha; \theta'_{W_0^4}\right) = \frac{1}{n}\theta'_{W_0^4}\sum_{\alpha=1}^{n}(\overline{\mathbf{W}_0^4})_{\beta\alpha}(\mathbf{h}_0^3)_\alpha. \quad \texttt{(Mome}$$

Stability at Initialisation: Satisfying stability at initialisation (Theorem 5) fixes the values of b_1, b_2, b_3, b_4, and the corresponding θ parameters. We find the values that satisfy stability at initialisation and fix them for the rest of the program. $\mathbf{h}_0^1$ is trivially a Gaussian with scale $\theta_{W_0^1} = n^{-b_1}\|\boldsymbol{\xi}\|$. Hence, we set $b_1 = 0$ to satisfy stability. Using the Master theorem (Yang and Hu, 2021), $\mathbb{E}\left[(Z^{h_0^2})^2\right] = n^{(1-2b_2)}\mathbb{E}\left[(Z^{h_0^1})^2\right] = n^{(1-2b_2)}\|\boldsymbol{\xi}\|^2$. Stability is satisfied for $b_2 \geq \frac{1}{2}$. We can use the Master theorem to find the distribution of $\mathbf{h}_0^3$ as $Z^{h^{3_0}} = \mathcal{N}(0, n^{(1-2b_3)}\mathbb{E}\left[(Z^{h_0^1})^2\right])$. We thus set $b_3 = \frac{1}{2}$. Finally, using similar reasoning as for $\mathbf{h}_0^2$, setting $b_4 \geq \frac{1}{2}$ ensures that $\mathbf{f}_0$ is stable. Note that $\tilde{\mathbf{h}}_0^2$ is always stable due to the bounded softmax operation.

Remark 10 (Feature Learning considerations to constrain output-like initialisation) *Stability constrains the output-like layers to initialise with $b_2 \geq 1/2$ and $b_4 \geq 1/2$. For ease of exposition, we use the Tensor Program derived below to constrain these values. Looking at the first feature updates of the router*

$$\mathbf{h}_1^2 - \mathbf{h}_0^2 = \theta_{\delta h_1^1}\theta'_{W_0^2}\frac{\overline{\mathbf{W}_0^2}\delta\mathbf{h}_1^1}{n} - \theta'_{\delta W_1^2}\eta\mathrm{d}\mathbf{h}_0^2\boldsymbol{\chi}_0\frac{(\mathbf{h}_0^1)^T\mathbf{h}_1^1}{n}.$$

By the Master Theorem, $\frac{\overline{\mathbf{W}_0^2}\delta\mathbf{h}_1^1}{n}$ which is a Moment operation converges to an expectation (Appendix C.2). If we have feature learning for $\mathbf{h}_1^1 - \mathbf{h}_0^1$, we require that $\theta_{\delta h_1^1} = 1$. Thus, for the first term to scale $\Theta(1)$, we require $\theta'_{W_0^2} = n \cdot n^{-b_2} = 1$, which implies setting $b_2 = 1$. A similar arguement holds for setting $b_4 = 1$.

The final values for stability at initialisation (and feature learning for 'output-like' layers) are

$$b_1 = 0, \quad b_2 = 1, \quad b_3 = \frac{1}{2}, \quad b_4 = 1, \implies \theta_{W_0^1} = 1, \quad \theta_{W_0^2} = n^{-1}, \quad \theta_{W_0^3} = 1, \quad \theta_{W_0^4} = n^{-1}.$$

Setting 11 (Stability at initialisation) *For stability at initialisation, and feature learning considerations for 'output-like' layers (Theorem 10), we choose*

$$b_1 = 0, \quad b_2 = 1, \quad b_3 = \frac{1}{2}, \quad b_4 = 1,$$

which gives

$$\theta_{W_0^1} = 1, \quad \theta_{W_0^2} = n^{-1}, \quad \theta_{W_0^3} = 1, \quad \theta_{W_0^4} = n^{-1}.$$

We set these values in the program for ease of exposition.

Note that under the assumptions made in Theorem 9 and Theorem 11 certain vectors have scale $\Theta(1)$ throughout training.

Remark 12 *Under the assumption of stable parametrisations and the values selected for stability at initialisation, the vectors $\mathbf{h}_t^1, \{\mathbf{h}_t^{3,k}\}_{k=1}^m, \mathbf{h}_t^3$ have scale $\Theta(1)$ throughout training. As the softmax and sigmoid operations are bounded, the softmax output $\tilde{\mathbf{h}}_t^2$ has scale $\Theta(1)$ throughout training.*

First Backward Pass:

$$\mathrm{d}\mathbf{h}_0^3 = (\overline{\mathbf{W}_0^4})^T, \quad \in \mathbb{R}^{n \times d_{\text{out}}}. \tag{12}$$

$$n \cdot \mathrm{d}\tilde{h}_{0,k}^2 = (\mathbf{h}_0^{3,k})^T \mathrm{d}\mathbf{h}_0^3, \quad \in \mathbb{R}^{1 \times d_{\text{out}}}, \qquad \forall k \in [m] \tag{13}$$

$$(\mathrm{d}\tilde{h}_{0,k}^2)_\beta = \frac{1}{n} \sum_{\alpha=1}^{n} \phi\left((\mathbf{h}_0^{3,k})_\alpha, (\mathrm{d}\mathbf{h}_0^3)_{\alpha\beta}; -\right), \qquad \texttt{(Moment)} \tag{14}$$

$$\phi(\ldots) := (\mathbf{h}_0^{3,k})_\alpha (\mathrm{d}\mathbf{h}_0^3)_{\alpha\beta}, \tag{15}$$

where we repeat the operation for each expert to give $\mathrm{d}\tilde{h}_{0,1}^2, \ldots, \mathrm{d}\tilde{h}_{0,m}^2$.

$$\mathrm{d}h_{0,k}^2 = \sum_{j=1}^{m} J(\tilde{\mathbf{h}}_0^2)_{jk} \mathrm{d}\tilde{h}_{0,j}^2, \quad \in \mathbb{R}^{1 \times d_{\text{out}}}, \qquad \forall k \in [m] \tag{16}$$

$$(\mathrm{d}h_{0,k}^2)_\beta = \frac{1}{n} \sum_{\alpha=1}^{n} \phi\left(-; \{J(\tilde{\mathbf{h}}_0^2)_{jk}\}_{j=1}^m, \{(\mathrm{d}\tilde{h}_{0,j}^2)_\beta\}_{j=1}^m\right), \qquad \texttt{(Moment)} \tag{17}$$

$$\phi(\ldots) := \sum_{j=1}^{m} J(\tilde{\mathbf{h}}_0^2)_{jk} (\overline{\mathrm{d}\tilde{h}_{0,j}^2})_\beta, \tag{18}$$

where $J(\tilde{\mathbf{h}}_0^2)_{ij} = \frac{\partial(\tilde{\mathbf{h}}_0^2)_i}{\partial(\mathbf{h}_0^2)_j}$.

Again, we repeat this operation for each expert to give $\mathrm{d}h_{0,1}^2, \ldots, \mathrm{d}h_{0,m}^2$. We also define

$$\mathrm{d}\mathbf{h}_0^2 = \begin{pmatrix} \mathrm{d}h_{0,1}^2 \\ \vdots \\ \mathrm{d}h_{0,m}^2 \end{pmatrix} \in \mathbb{R}^{m \times d_{\text{out}}}. \tag{19}$$

$$\mathrm{d}\mathbf{h}_0^{3,k} = \tilde{h}_{0,k}^2 \mathrm{d}\mathbf{h}_0^3, \quad \in \mathbb{R}^{n \times d_{\text{out}}}, \qquad \forall k \in [m] \tag{20}$$

$$(\mathrm{d}\mathbf{h}_0^{3,k})_{\alpha\beta} = \phi\left((\mathrm{d}\mathbf{h}_0^3)_{\alpha\beta}; \tilde{h}_{0,k}^2\right), \qquad \texttt{(Nonlin)} \tag{21}$$

$$\phi(\ldots) := \tilde{h}_{0,k}^2 \overline{\mathrm{d}\mathbf{h}_0^3}, \tag{22}$$

We repeat this for each expert to give $\mathrm{d}\mathbf{h}_0^{3,1}, \ldots, \mathrm{d}\mathbf{h}_0^{3,m}$.

$$\mathrm{d}\mathbf{h}_0^1 = \sum_{k=1}^m (\overline{\mathbf{W}_0^2})_{:,k}^T \mathrm{d}h_{0,k}^2 + \sum_{k=1}^m (\mathbf{W}_0^{3,k})^T \mathrm{d}\mathbf{h}_0^{3,k}, \quad \in \mathbb{R}^{n \times d_{\text{out}}}, \tag{23}$$

$$(\mathrm{d}\mathbf{h}_0^1)_{\alpha\beta} = \phi\left(\{(\overline{\mathbf{W}_0^2})_{\alpha k}^T, (\mathbf{V}_0^k)_{\alpha\beta}\}_{k=1}^m, ; \{(\mathrm{d}h_{0,k}^2)_\beta\}_{k=1}^m\right), \qquad \texttt{(Nonlin)} \tag{24}$$

$$(\mathbf{V}_0^k)_{:,\beta} := (\mathbf{W}_0^{3,k})^T (\mathrm{d}\mathbf{h}_0^{3,k})_{:,\beta}, \qquad \texttt{(MatMul)} \tag{25}$$

$$\phi(\ldots) := \sum_{k=1}^m (\overline{\mathbf{W}_0^2})_{:,k}^T \mathrm{d}h_{0,k}^2 + \sum_{k=1}^m \mathbf{V}_0^k. \tag{26}$$

Note that the final gradient has two branches. The scale of this term thus will be the maximum of the the scale of the twp branches. There is a $\theta'_{W_0^4}\theta_{W_0^2}$ scale coming from the router and a $\theta_{W_0^4}$ coming from the experts. The scale from the branches is the same here.

First Weight Updates: Recall that we define the gradient of the loss with respect to the function as $\boldsymbol{\chi}_t$. For the vectors $\mathbf{W} \in \{\mathbf{W}^1, \mathbf{W}^2, \mathbf{W}^4\}$ we write $\delta\mathbf{W}_t := \theta_{\delta W_t}^{-1}(\mathbf{W}_t - \mathbf{W}_{t-1})$, and thus the weights at step t as $\mathbf{W}_t = \theta_{W_0}\overline{\mathbf{W}_0} + \ldots + \theta_{\delta W_t}\delta W_t$.

$$\mathbf{W}_1^4 - \mathbf{W}_0^4 = -n^{-c_4}\eta\boldsymbol{\chi}_0(\mathbf{h}_0^3)^T \tag{27}$$

$$\theta_{\delta W_1^4} := n^{-c_4}. \tag{28}$$

$$\mathbf{W}_1^{3,k} - \mathbf{W}_0^{3,k} = -n^{-c_3}\theta_{W_0^4}\eta\mathrm{d}\mathbf{h}_0^{3,k}\boldsymbol{\chi}_0(\mathbf{h}_0^1)^T \tag{29}$$

$$\theta_{\delta W_1^3} := n^{-(c_3+1)}. \tag{30}$$

$$\mathbf{W}_1^2 - \mathbf{W}_0^2 = -n^{-c_2}n\theta_{W_0^4}\eta\mathrm{d}\mathbf{h}_0^2\boldsymbol{\chi}_0(\mathbf{h}_0^1)^T \tag{31}$$

$$\theta_{\delta W_1^2} := n^{-c_2}. \tag{32}$$

$$\mathbf{W}_1^1 - \mathbf{W}_0^1 = -n^{-c_1}n\theta_{W_0^4}\theta_{W_0^2}\eta\mathrm{d}\mathbf{h}_0^1\boldsymbol{\chi}_0(\boldsymbol{\xi})^T \tag{33}$$

$$\theta_{\delta W_1^1} := n^{-(c_1+1)}. \tag{34}$$

The router bias in the learnable case updates as

$$\mathbf{b}_1^2 - \mathbf{b}_0^2 = -\eta\theta'_{W_0^4}\mathrm{d}\mathbf{h}_0^2\boldsymbol{\chi}_0, \tag{35}$$

where due to stability at initialisation (Theorem 11) $\theta'_{W_0^4} = 1$.

First Feature Updates: The feature updates are affected by the change in the previous layers features (*propagating updates*) and the update in the weights at the current layer (*effective updates*). We write the feature updates for the vectors $\mathbf{h} \in \{\mathbf{h}^1, \{\mathbf{h}^{3,k}\}_{k=1}^m, \mathbf{h}^3\}$ as $\delta\mathbf{h}_t := \theta_{\delta h_t}^{-1}(\mathbf{h}_t - \mathbf{h}_{t-1})$. This gives each features for the vectors at step t as $\mathbf{h}_t = \mathbf{h}_0 + \theta_{\delta h_1}\delta\mathbf{h}_1 + \ldots + \theta_{\delta h_t}\delta\mathbf{h}_t$. For the scalars $\mathbf{h} \in \{\mathbf{h}^2, \tilde{\mathbf{h}}^2, \mathbf{f}\}$ we simply write $\delta\mathbf{h}_t := (\mathbf{h}_t - \mathbf{h}_{t-1})$ and $\mathbf{h}_t = \mathbf{h}_0 + \ldots + \delta\mathbf{h}_t$.

$$\mathbf{h}_1^1 - \mathbf{h}_0^1 = (\mathbf{W}_1^1 - \mathbf{W}_0^1)\boldsymbol{\xi} \tag{36}$$

$$= -\theta_{\delta W_1^1}\eta \mathrm{d}\mathbf{h}_0^1 \boldsymbol{\chi}_0 \boldsymbol{\xi}^T \boldsymbol{\xi} \tag{37}$$

$$(\delta\mathbf{h}_1^1)_\alpha = \phi\left(\{(\mathrm{d}\mathbf{h}_0^1)_{\alpha\beta}\}_{\beta=1}^{d_{\text{out}}}; -\eta, \{(\boldsymbol{\chi}_0\boldsymbol{\xi}^T\boldsymbol{\xi})_\beta\}_{\beta=1}^{d_{\text{out}}}\right) \qquad \texttt{(Nonlin)} \tag{38}$$

$$\phi(\ldots) := -\eta \sum_{\beta=1}^{d_{\text{out}}} (\mathrm{d}\mathbf{h}_0^1)_{\alpha\beta}(\boldsymbol{\chi}_0\boldsymbol{\xi}^T\boldsymbol{\xi})_\beta, \tag{39}$$

$$\theta_{\delta h_1^1} := \theta_{\delta W_1^1}, \tag{40}$$

$$\mathbf{h}_1^2 - \mathbf{h}_0^2 = \mathbf{W}_0^2(\mathbf{h}_1^1 - \mathbf{h}_0^1) + (\mathbf{W}_1^2 - \mathbf{W}_0^2)\mathbf{h}_1^1 + (\mathbf{b}_1^2 - \mathbf{b}_0^2) \tag{41}$$

$$= \theta_{\delta h_1^1}\theta'_{W_0^2}\frac{\overline{\mathbf{W}_0^2}\delta\mathbf{h}_1^1}{n} - \theta'_{\delta W_1^2}\eta \mathrm{d}\mathbf{h}_0^2 \boldsymbol{\chi}_0 \frac{(\mathbf{h}_0^1)^T\mathbf{h}_1^1}{n} - \eta \mathrm{d}\mathbf{h}_0^2\boldsymbol{\chi}_0, \tag{42}$$

$$\theta_{\delta h_1^2}(\delta\mathbf{h}_1^2)_k = \frac{1}{n}\sum_{\alpha=1}^n \phi\left(\overline{\mathbf{W}_0^2}_{k\alpha}, (\delta\mathbf{h}_1^1)_\alpha, (\mathbf{h}_0^1)_\alpha, (\mathbf{h}_1^1)_\alpha; \theta'_{W_0^4}, \theta_{\delta h_1^1}, \theta'_{W_0^2}, \theta'_{\delta W_1^2}, \eta, \{(\mathrm{d}\mathbf{h}_0^2)_{k\beta}, (\boldsymbol{\chi}_0)_\beta\}_\beta^{d_{\text{out}}}\right) \qquad \texttt{(Moment)} \tag{43}$$

$$\phi(\ldots) := \theta_{\delta h_1^1}\theta'_{W_0^2}\overline{\mathbf{W}_0^2}_{k\alpha}(\delta\mathbf{h}_1^1)_\alpha - \theta'_{\delta W_1^2}\eta\sum_{\beta=1}^{d_{\text{out}}}(\mathrm{d}\mathbf{h}_0^2)_{k\beta}(\boldsymbol{\chi}_0)_\beta(\mathbf{h}_0^1)_\alpha(\mathbf{h}_1^1)_\alpha - \eta\theta'_{W_0^4}\sum_{\beta=1}^{d_{\text{out}}}\mathrm{d}(\mathbf{h}_0^2)_{k\beta}(\boldsymbol{\chi}_0)_\beta, \tag{44}$$

$$\theta_{\delta h_1^2} := \max(\theta_{\delta h_1^1}\theta'_{W_0^2}, \theta'_{\delta W_1^2}), \tag{45}$$

$$\tilde{\mathbf{h}}_1^2 - \tilde{\mathbf{h}}_0^2 = \mathrm{G}(\mathbf{h}_1^2) - \mathrm{G}(\mathbf{h}_0^2) \tag{46}$$

$$= \mathrm{G}\left(\mathbf{h}_0^2 + \theta_{\delta h_1^2}\delta\mathbf{h}_1^2\right) - \mathrm{G}(\mathbf{h}_0^2) \tag{47}$$

$$\theta_{\delta\tilde{h}_1^2}(\delta\tilde{\mathbf{h}}_1^2)_k = \frac{1}{n}\sum_{\alpha=1}^n \phi_k\left(-; \{(\mathbf{h}_0^2)_k, (\delta\mathbf{h}_1^2)_k\}_{k=1}^m, \theta_{\delta h_1^2}\right), \qquad \texttt{(Moment)}, \tag{48}$$

$$\theta_{\delta\tilde{h}_1^2} := \theta_{\delta h_1^2}. \tag{49}$$

where G is the router gating function (either softmax or sigmoid). Both softmax and sigmoid operations can be represented as a Moment (see Appendix G).

$$\mathbf{h}_1^{3,k} - \mathbf{h}_0^{3,k} = \mathbf{W}_0^{3,k}(\mathbf{h}_1^1 - \mathbf{h}_0^1) + (\mathbf{W}_1^{3,k} - \mathbf{W}_0^{3,k})\mathbf{h}_1^1, \tag{50}$$

$$= \theta_{\delta h_1^1}\mathbf{W}_0^{3,k}\delta\mathbf{h}_1^1 - \theta'_{\delta W_1^3}\eta \mathrm{d}\mathbf{h}_0^{3,k}\boldsymbol{\chi}_0\frac{(\mathbf{h}_0^1)^T(\mathbf{h}_1^1)}{n}, \tag{51}$$

$$(\delta\mathbf{h}_1^{3,k})_\alpha = \phi\left(\Upsilon_\alpha, \{(\mathrm{d}\mathbf{h}_0^{3,k})_{\alpha l}\}_{l=1}^{d_{\text{out}}}; \theta_{\delta h_1^1/\delta h_1^{3,k}}, \theta'_{\delta W_1^3/\delta h_1^{3,k}}, \{(\boldsymbol{\chi}_0)_l\}_{l=1}^{d_{\text{out}}}, \eta, c_0,\right), \quad \texttt{(Nonlin)} \tag{52}$$

$$\Upsilon := \mathbf{W}_0^{3,k}\delta\mathbf{h}_1^1, \quad \texttt{(MatMul)} \tag{53}$$

$$c_0 := \frac{1}{n}\sum_{\alpha=1}^{n}\psi\left((\mathbf{h}_0^1)_\alpha, (\mathbf{h}_1^1)_\alpha; -\right), \quad \texttt{(Moment)} \tag{54}$$

$$\phi(\ldots) := \theta_{\delta h_1^1/\delta h_1^{3,k}}\Upsilon_\alpha - \theta'_{\delta W_1^3/\delta h_1^{3,k}}\eta c_0\sum_{l=1}^{d_{\text{out}}}(\mathrm{d}\mathbf{h}_0^{3,k})_{\alpha l}(\boldsymbol{\chi}_0)_l, \tag{55}$$

$$\theta_{\delta h_1^{3,k}} := \max(\theta_{\delta h_1^1}, \theta'_{\delta W_1^3}), \tag{56}$$

$$\theta_{\delta h_1^1/\delta h_1^{3,k}} := \theta_{\delta h_1^1}/\theta_{\delta h_1^{3,k}}, \tag{57}$$

$$\theta'_{\delta W_1^3/\delta h_1^{3,k}} := \theta'_{\delta W_1^3}/\theta_{\delta h_1^{3,k}} \tag{58}$$

We repeat the above process for each expert k to give $\{\delta\mathbf{h}_1^{3,k}\}_{k=1}^m$.

$$\mathbf{h}_1^3 - \mathbf{h}_0^3 = \sum_{k=1}^{m}\left(\tilde{h}_{0,k}^2(\mathbf{h}_1^{3,k} - \mathbf{h}_0^{3,k}) + (\tilde{h}_{1,k}^2 - \tilde{h}_{0,k}^2)\mathbf{h}_1^{3,k}\right) \tag{59}$$

$$= \sum_{k=1}^{m}\left(\theta_{\delta h_1^{3,k}}\tilde{h}_{0,k}^2\delta\mathbf{h}_1^{3,k} + \theta_{\delta\tilde{h}_1^2}\delta\tilde{h}_{1,k}^2\mathbf{h}_1^{3,k}\right), \tag{60}$$

$$(\delta\mathbf{h}_1^3)_\alpha = \phi\left(\{(\delta\mathbf{h}_1^{3,k})_\alpha\}_{k=1}^m, \{(\mathbf{h}_1^{3,k})_\alpha\}_{k=1}^m; \theta_{\delta h_1^{3,k}/\delta h_1^3}, \theta_{\delta\tilde{h}_1^2/\delta h_1^3}, \{\tilde{h}_{0,k}^2\}_{k=1}^m, \{\delta\tilde{h}_{1,k}^2\}_{k=1}^m\right), \quad \texttt{(Nonlin)} \tag{61}$$

$$\phi(\ldots) := \sum_{k=1}^{m}\left(\theta_{\delta h_1^{3,k}/\delta h_1^3}\tilde{h}_{0,k}^2(\delta\mathbf{h}_1^{3,k})_\alpha + \theta_{\delta\tilde{h}_1^2/\delta h_1^3}\delta\tilde{h}_{1,k}^2(\mathbf{h}_1^{3,k})_\alpha\right), \tag{62}$$

$$\theta_{\delta h_1^3} := \max(\theta_{\delta h_1^{3,k}}, \theta_{\delta\tilde{h}_1^2}), \tag{63}$$

$$\theta_{\delta h_1^{3,k}/\delta h_1^3} := \theta_{\delta h_1^{3,k}}/\theta_{\delta h_1^3}, \tag{64}$$

$$\theta_{\delta\tilde{h}_1^2/\delta h_1^3} := \theta_{\delta\tilde{h}_1^2}/\theta_{\delta h_1^3}. \tag{65}$$

$$\mathbf{f}_1 - \mathbf{f}_0 = \mathbf{W}_0^4(\mathbf{h}_1^3 - \mathbf{h}_0^3) + (\mathbf{W}_1^4 - \mathbf{W}_0^4)\mathbf{h}_1^3 \tag{66}$$

$$= \theta_{\delta h_1^3}\frac{\overline{\mathbf{W}_0^4}\delta\mathbf{h}_1^3}{n} - \theta'_{\delta W_1^4}\eta\boldsymbol{\chi}_0\frac{(\mathbf{h}_0^3)^T(\mathbf{h}_1^3)}{n}, \tag{67}$$

$$\theta_{\delta f_1}(\delta\mathbf{f}_1)_\beta = \frac{1}{n}\sum_{\alpha=1}^{n}\phi\left(\overline{\mathbf{W}_0^4}_{\beta\alpha}, (\delta\mathbf{h}_1^3)_\alpha, (\mathbf{h}_0^3)_\alpha, (\mathbf{h}_1^3)_\alpha; \theta_{\delta h_1^3}, \theta'_{\delta W_1^4}, \eta, (\boldsymbol{\chi}_0)_\beta\right), \quad \texttt{(Moment)} \tag{68}$$

$$\phi(\ldots) := \theta_{\delta h_1^3}(\overline{\mathbf{W}_0^4})_{\beta\alpha}(\delta\mathbf{h}_1^3)_\alpha - \theta'_{\delta W_1^4}\eta(\boldsymbol{\chi}_0)_\beta(\mathbf{h}_0^3)_\alpha(\mathbf{h}_1^3)_\alpha, \tag{69}$$

$$\theta_{\delta f_1} := \max(\theta_{\delta h_1^3}, \theta'_{\delta W_1^4}). \tag{70}$$

tth Backward Pass: Recall that we write the weights at step t as $\mathbf{W}_t = \theta_{W_0}\overline{\mathbf{W}_0} + \ldots + \theta_{\delta W_t}\delta W_t$.

$$\mathrm{d}\mathbf{h}_t^3 := (\overline{\mathbf{W}_t^4})^T, \quad \in \mathbb{R}^{n\times d_{\text{out}}}. \tag{71}$$

$$n\cdot\mathrm{d}\tilde{h}_{t,k}^2 = (\mathbf{h}_t^{3,k})^T\mathrm{d}\mathbf{h}_t^3, \quad \in \mathbb{R}^{1\times d_{\text{out}}}, \qquad \forall k\in[m] \tag{72}$$

$$(\mathrm{d}\tilde{h}_{t,k}^2)_\beta = \frac{1}{n}\sum_{\alpha=1}^{n}\phi\left((\mathbf{h}_t^{3,k})_\alpha, (\mathrm{d}\mathbf{h}_t^3)_{\alpha\beta}; -\right), \qquad \texttt{(Moment)} \tag{73}$$

$$\phi(\ldots) := (\mathbf{h}_t^{3,k})_\alpha(\mathrm{d}\mathbf{h}_t^3)_{\alpha\beta}, \tag{74}$$

where we repeat this operation for each expert $\mathrm{d}\tilde{h}_{t,1}^2, \ldots, \mathrm{d}\tilde{h}_{t,m}^2$.

$$\mathrm{d}h_{t,k}^2 = \sum_{j=1}^{m} J(\tilde{\mathbf{h}}_t^2)_{jk}\mathrm{d}\tilde{h}_{t,j}^2, \quad \in \mathbb{R}^{1\times d_{\text{out}}}, \qquad \forall k\in[m] \tag{75}$$

$$(\mathrm{d}h_{t,k}^2)_\beta = \frac{1}{n}\sum_{\alpha=1}^{n}\phi\left(-; \{J(\tilde{\mathbf{h}}_t^2)_{jk}\}_{j=1}^m, \{(\mathrm{d}\tilde{h}_{t,j}^2)_\beta\}_{j=1}^m\right), \qquad \texttt{(Moment)} \tag{76}$$

$$\phi(\ldots) := \sum_{j=1}^{m} J(\tilde{\mathbf{h}}_t^2)_{jk}(\mathrm{d}\tilde{h}_{t,j}^2)_\beta, \tag{77}$$

where $J(\tilde{\mathbf{h}}_t^2)_{ij} = \frac{\partial(\tilde{\mathbf{h}}_t^2)_i}{\partial(\mathbf{h}_t^2)_j}$. It is useful to consider this terms for all the experts,

$$(\mathrm{d}\mathbf{h}_t^2)_\beta = \begin{pmatrix}(\mathrm{d}h_{t,1}^2)_\beta\\ \vdots \\ (\mathrm{d}h_{t,m}^2)_\beta\end{pmatrix} \quad \in \mathbb{R}^{m\times d_{\text{out}}}. \tag{78}$$

$$\mathrm{d}\mathbf{h}_t^{3,k} = \tilde{h}_{t,k}^2\mathrm{d}\mathbf{h}_t^3, \quad \in \mathbb{R}^{n\times d_{\text{out}}}, \qquad \forall k\in[m] \tag{79}$$

$$(\mathrm{d}\mathbf{h}_t^{3,k})_{\alpha\beta} = \phi\left((\mathrm{d}\mathbf{h}_t^3)_{\alpha\beta}; \tilde{h}_{t,k}^2\right), \qquad \texttt{(Nonlin)} \tag{80}$$

$$\phi(\ldots) := \tilde{h}_{t,k}^2(\overline{\mathrm{d}\mathbf{h}_t^3})_{\alpha\beta}, \tag{81}$$

where we repeat this for each expert to give $\mathrm{d}\mathbf{h}_t^{3,1}, \ldots, \mathrm{d}\mathbf{h}_t^{3,m}$.

$$\mathrm{d}\mathbf{h}_t^1 = \sum_{k=1}^{m} (\overline{\mathbf{W}_t^2})_{:,k}^T \mathrm{d}h_{t,k}^2 + \sum_{k=1}^{m} (\mathbf{W}_0^{3,k})^T \mathrm{d}\mathbf{h}_t^{3,k} - \sum_{k=1}^{m} \sum_{s=0}^{t-1} \eta \mathbf{h}_s^1 (\boldsymbol{\chi}_s)^T \frac{(\mathrm{d}\mathbf{h}_s^{3,k})^T \mathrm{d}\mathbf{h}_t^{3,k}}{n}, \quad \in \mathbb{R}^{n \times d_{\text{out}}}, \tag{82}$$

$$(\mathrm{d}\mathbf{h}_t^1)_{\alpha\beta} = \phi\Big(\{(\overline{\mathbf{W}_t^2})_{\alpha k}^T, (\mathbf{V}_t^k)_{\alpha\beta}\}_{k=1}^m, \{(\mathbf{h}_s^1)_\alpha\}_{s=1}^{t-1}; \{(\mathrm{d}h_{t,k}^2)_\beta\}_{k=1}^m, \{(\boldsymbol{\chi}_s)_\beta, \{C_{s;\gamma\beta}^k\}_{k=1}^m\}_{s=1}^{t-1} \Big), \tag{Nonlin}$$

$$(\mathbf{V}_t^k)_{:,\beta} := (\mathbf{W}_0^{3,k})^T (\mathrm{d}\mathbf{h}_t^{3,k})_{:,\beta}, \tag{MatMul}$$

$$C_{s;\gamma\beta}^k := \frac{1}{n} \sum_{\alpha=1}^{n} \psi\left((\mathrm{d}\mathbf{h}_s^{3,k})_{\alpha\gamma}, (\mathrm{d}\mathbf{h}_t^{3,k})_{\alpha\beta}; - \right), \tag{Moment}$$

$$\phi(\ldots) := \sum_{k=1}^{m} (\overline{\mathbf{W}_t^2})_{\alpha k}^T (\mathrm{d}h_{t,k}^2)_\beta + \sum_{k=1}^{m} (\mathbf{V}_t^k)_{\alpha\beta} - \sum_{k=1}^{m} \sum_{s=0}^{t-1} \sum_{\gamma=1}^{d_{\text{out}}} \eta (\mathbf{h}_s^1)_\alpha (\boldsymbol{\chi}_s)_\gamma C_{s;\gamma\beta}^k. \tag{86}$$

tth Weight Updates: Recall that we define the gradient of the loss with respect to the function as $\boldsymbol{\chi}_t \in \mathbb{R}^{d_{\text{out}} \times 1}$.

$$\mathbf{W}_t^4 - \mathbf{W}_{t-1}^4 = -n^{-c_4} \eta \boldsymbol{\chi}_{t-1} (\mathbf{h}_{t-1}^3)^T \tag{87}$$

$$\theta_{\delta W_t^4} := n^{-c_4}. \tag{88}$$

$$\mathbf{W}_t^{3,k} - \mathbf{W}_{t-1}^{3,k} = -n^{-c_3} \theta_{W_{t-1}^4} \eta \mathrm{d}\mathbf{h}_{t-1}^{3,k} \boldsymbol{\chi}_{t-1} (\mathbf{h}_{t-1}^1)^T \tag{89}$$

$$\theta_{\delta W_t^3} := n^{-c_3} \theta_{W_{t-1}^4} \tag{90}$$

$$\mathbf{W}_t^2 - \mathbf{W}_{t-1}^2 = -n^{-c_2} \theta'_{W_{t-1}^4} \eta \mathrm{d}\mathbf{h}_{t-1}^2 \boldsymbol{\chi}_{t-1} (\mathbf{h}_{t-1}^1)^T \tag{91}$$

$$\theta_{\delta W_t^2} := n^{-c_2} \theta'_{W_{t-1}^4}. \tag{92}$$

$$\mathbf{W}_t^1 - \mathbf{W}_{t-1}^1 = -n^{-c_1} \theta'_{W_{t-1}^2} \theta_{W_{t-1}^4} \eta \mathrm{d}\mathbf{h}_{t-1}^1 \boldsymbol{\chi}_{t-1} (\boldsymbol{\xi})^T \tag{93}$$

$$\theta_{\delta W_t^1} := n^{-c_1} \theta'_{W_{t-1}^2} \theta_{W_{t-1}^4}. \tag{94}$$

For the learnable bias we have,

$$\mathbf{b}_t^2 - \mathbf{b}_{t-1}^2 = -\eta \theta'_{W_{t-1}^4} \mathrm{d}\mathbf{h}_{t-1}^2 \boldsymbol{\chi}_{t-1}, \tag{95}$$

tth Feature Updates:

$$\mathbf{h}_t^1 - \mathbf{h}_{t-1}^1 = (\mathbf{W}_t^1 - \mathbf{W}_{t-1}^1)\boldsymbol{\xi} \tag{96}$$

$$= -\theta_{\delta W_t^1}\eta \mathrm{d}\mathbf{h}_{t-1}^1 \boldsymbol{\chi}_{t-1}\boldsymbol{\xi}^T\boldsymbol{\xi} \tag{97}$$

$$(\delta\mathbf{h}_t^1)_\alpha = \phi\left(\{(\mathrm{d}\mathbf{h}_{t-1}^1)_{\alpha j}\}_{j=1}^{d_{\text{out}}}; -\eta, \{(\boldsymbol{\chi}_{t-1}\boldsymbol{\xi}^T\boldsymbol{\xi})_j\}_{j=1}^{d_{\text{out}}}\right) \qquad \texttt{(Nonlin)} \tag{98}$$

$$\phi(\ldots) := -\eta\sum_{j=1}^{d_{\text{out}}}(\mathrm{d}\mathbf{h}_{t-1}^1)_{\alpha j}(\boldsymbol{\chi}_{t-1}\boldsymbol{\xi}^T\boldsymbol{\xi})_j \tag{99}$$

$$\theta_{\delta h_t^1} := \theta_{\delta W_t^1}. \tag{100}$$

$$\mathbf{h}_t^2 - \mathbf{h}_{t-1}^2 = \mathbf{W}_{t-1}^2(\mathbf{h}_t^1 - \mathbf{h}_{t-1}^1) + (\mathbf{W}_t^2 - \mathbf{W}_{t-1}^2)\mathbf{h}_t^1 + (\mathbf{b}_t^2 - \mathbf{b}_{t-1}^2) \tag{101}$$

$$= \theta'_{W_{t-1}^2}\theta_{\delta h_t^1}\frac{\overline{\mathbf{W}_{t-1}^2}\delta\mathbf{h}_t^1}{n} - \theta'_{\delta W_t^2}\eta\mathrm{d}\mathbf{h}_{t-1}^2\boldsymbol{\chi}_{t-1}\frac{(\mathbf{h}_{t-1}^1)^T\mathbf{h}_t^1}{n} - \eta\theta'_{W_{t-1}^4}\mathrm{d}\mathbf{h}_{t-1}^2\boldsymbol{\chi}_{t-1}, \tag{102}$$

$$\theta_{\delta h_t^2}(\delta\mathbf{h}_t^2)_k = \frac{1}{n}\sum_{\alpha=1}^{n}\phi\Bigg((\overline{\mathbf{W}_{t-1}^2})_{k\alpha}, (\delta\mathbf{h}_t^1)_\alpha, (\mathbf{h}_{t-1}^1)_\alpha, (\mathbf{h}_t^1)_\alpha;$$

$$\theta'_{W_{t-1}^4}, \theta'_{W_{t-1}^2}, \theta_{\delta h_t^1}, \theta'_{\delta W_t^2}, \eta, \{(\mathrm{d}h_{t-1,k}^2)_\beta, (\boldsymbol{\chi}_{t-1})_\beta\}_{\beta=1}^{d_{\text{out}}}\Bigg), \qquad \texttt{(Moment)} \tag{103}$$

$$\phi(\ldots) := \theta'_{W_{t-1}^2}\theta_{\delta h_t^1}(\overline{\mathbf{W}_{t-1}^2})_{k\alpha}(\delta\mathbf{h}_t^1)_\alpha - \theta'_{\delta W_t^2}\eta\sum_{\beta=1}^{d_{\text{out}}}(\mathrm{d}h_{t-1,k}^2)_\beta(\boldsymbol{\chi}_{t-1})_\beta(\mathbf{h}_{t-1}^1)_\alpha(\mathbf{h}_t^1)_\alpha$$

$$- \eta\theta'_{W_{t-1}^4}\sum_{\beta=1}^{d_{\text{out}}}(\mathrm{d}\mathbf{h}_{t-1}^2)_{k\beta}(\boldsymbol{\chi}_{t-1})_\beta. \tag{104}$$

$$\theta_{\delta h_t^2} := \max(\theta'_{W_{t-1}^2}\theta_{\delta h_t^1}, \theta'_{\delta W_t^2}, \theta'_{W_{t-1}^4}). \tag{105}$$

$$\tilde{\mathbf{h}}_t^2 - \tilde{\mathbf{h}}_{t-1}^2 = \mathrm{G}(\mathbf{h}_t^2) - \mathrm{G}(\mathbf{h}_{t-1}^2) \tag{106}$$

$$= \mathrm{G}\left(\mathbf{h}_{t-1}^2 + \theta_{\delta h_t^2}\delta\mathbf{h}_t^2\right) - \mathrm{G}(\mathbf{h}_{t-1}^2) \tag{107}$$

$$\theta_{\delta\tilde{h}_t^2}(\delta\tilde{\mathbf{h}}_t^2)_k = \frac{1}{n}\sum_{\alpha=1}^{n}\phi_k\left(-; \{(\mathbf{h}_{t-1}^2)_k, (\delta\mathbf{h}_t^2)_k\}_{k=1}^m, \theta_{\delta h_t^2}\right), \qquad \texttt{(Moment)}, \tag{108}$$

$$\theta_{\delta\tilde{h}_t^2} := \theta_{\delta h_t^2}, \tag{109}$$

where G is the gating function (either softmax or sigmoid) (see Appendix G) (Yang and Hu, 2021).

$$\mathbf{h}_t^{3,k} - \mathbf{h}_{t-1}^{3,k} = \mathbf{W}_{t-1}^{3,k}(\mathbf{h}_t^1 - \mathbf{h}_{t-1}^1) + (\mathbf{W}_t^{3,k} - \mathbf{W}_{t-1}^{3,k})\mathbf{h}_t^1, \tag{110}$$

$$= \theta_{\delta h_t^1}\mathbf{W}_0^{3,k}\delta\mathbf{h}_t^1 - \sum_{s=0}^{t-2}\theta'_{\delta W_{s+1}^3}\theta_{\delta h_t^1}\eta \mathrm{d}\mathbf{h}_s^{3,k}\boldsymbol{\chi}_s\frac{(\mathbf{h}_s^1)^T(\delta\mathbf{h}_t^1)}{n} - \theta'_{\delta W_t^3}\eta \mathrm{d}\mathbf{h}_{t-1}^{3,k}\boldsymbol{\chi}_{t-1}\frac{(\mathbf{h}_{t-1}^1)^T(\mathbf{h}_t^1)}{n}, \tag{111}$$

$$(\delta\mathbf{h}_t^{3,k})_\alpha = \phi\left(\Upsilon_\alpha, \{\Lambda_{s,\alpha}\}_{s=0}^{t-1}, \mathbf{V}_\alpha; \theta_{\delta h_t^1/\delta h_t^{3,k}}, \{\theta'_{\delta W_s^3 \delta h_t^1/\delta h_t^{3,k}}\}_{s=0}^{t-1}, \theta'_{\delta W_t^3/\delta h_t^{3,k}}\right) \tag{Nonlin}$$

$$\tag{112}$$

$$\phi(\ldots) := \theta_{\delta h_t^1/\delta h_t^{3,k}}\Upsilon_\alpha - \sum_{s=0}^{t-2}\theta'_{\delta W_{s+1}^3 \delta h_t^1/\delta h_t^{3,k}}\Lambda_{s,\alpha} - \theta'_{\delta W_t^3/\delta h_t^{3,k}}\mathbf{V}_\alpha, \tag{113}$$

$$\Upsilon := \mathbf{W}_0^{3,k}\delta\mathbf{h}_t^1, \tag{MatMul}$$

$$\tag{114}$$

$$\Lambda_{s,\alpha} := \psi\left(\{(\mathrm{d}\mathbf{h}_s^{3,k})_{\alpha l}\}_{l=1}^{d_{\text{out}}}; \eta, \{(\boldsymbol{\chi}_s)_l\}_{l=1}^{d_{\text{out}}}, d_s\right) = \eta d_s \sum_{l=1}^{d_{\text{out}}}(\mathrm{d}\mathbf{h}_s^{3,k})_{\alpha l}(\boldsymbol{\chi}_s)_l, \tag{Nonlin}$$

$$\tag{115}$$

$$d_s := \frac{1}{n}\sum_{\alpha=1}^{n}\nu\left((\mathbf{h}_s^1)_\alpha, (\delta\mathbf{h}_t^1)_\alpha; -\right), \tag{Moment}$$

$$\tag{116}$$

$$\mathbf{V}_\alpha := \psi\left(\{(\mathrm{d}\mathbf{h}_{t-1}^{3,k})_{\alpha l}\}_{l=1}^{d_{\text{out}}}; \{(\boldsymbol{\chi}_{t-1})_l\}_{l=1}^{d_{\text{out}}}, \eta, c_{t-1}\right) = \eta c_{t-1}\sum_{l=1}^{d_{\text{out}}}(\mathrm{d}\mathbf{h}_{t-1}^{3,k})_{\alpha l}(\boldsymbol{\chi}_{t-1})_l, \tag{Nonlin}$$

$$\tag{117}$$

$$c_{t-1} := \frac{1}{n}\sum_{\alpha=1}^{n}\psi\left((\mathbf{h}_{t-1}^1)_\alpha, (\mathbf{h}_t^1)_\alpha; -\right), \tag{Moment}$$

$$\tag{118}$$

$$\theta_{\delta h_t^{3,k}} = \max\left(\theta_{\delta h_t^1}, \max_{0\le s\le t-2}\left(\theta'_{\delta W_{s+1}^3}\theta_{\delta h_t^1}\right), \theta'_{\delta W_t^3}\right), \tag{119}$$

$$\theta_{\delta h_t^1/\delta h_t^{3,k}} := \theta_{\delta h_t^1}/\theta_{\delta h_t^{3,k}}, \tag{120}$$

$$\theta'_{\delta W_s^3 \delta h_t^1/\delta h_t^{3,k}} := \theta'_{\delta W_s^3}\theta_{\delta h_t^1}/\theta_{\delta h_t^{3,k}}, \tag{121}$$

$$\theta'_{\delta W_t^3/\delta h_t^{3,k}} := \theta'_{\delta W_t^3}/\theta_{\delta h_t^{3,k}}. \tag{122}$$

We repeat the above process for each expert k.

$$\mathbf{h}_t^3 - \mathbf{h}_{t-1}^3 = \sum_{k=1}^{m} \left(\tilde{h}_{t-1,k}^2 (\mathbf{h}_t^{3,k} - \mathbf{h}_{t-1}^{3,k}) + (\tilde{h}_{t,k}^2 - \tilde{h}_{t-1,k}^2) \mathbf{h}_t^{3,k} \right), \tag{123}$$

$$= \sum_{k=1}^{m} \theta_{\delta h_t^{3,k}} \tilde{h}_{t-1,k}^2 \delta \mathbf{h}_t^{3,k} + \theta_{\delta \tilde{h}_t^2} \delta \tilde{h}_{t,k}^2 \mathbf{h}_t^{3,k}, \tag{124}$$

$$(\delta \mathbf{h}_t^3)_\alpha = \phi \left(\{(\delta \mathbf{h}_t^{3,k})_\alpha\}_{k=1}^m, \{(\mathbf{h}_t^{3,k})_\alpha\}_{k=1}^m; \theta_{\delta h_t^{3,k}/\delta h_t^3}, \theta_{\delta \tilde{h}_t^2/\delta h_t^3}, \{\tilde{h}_{t-1,k}^2\}_{k=1}^m, \{\delta \tilde{h}_{t,k}^2\}_{k=1}^m \right), \quad \texttt{(Nonlin)} \tag{125}$$

$$\theta_{\delta h_t^3} := \max(\theta_{\delta h_t^{3,k}}, \theta_{\delta \tilde{h}_t^2}), \tag{126}$$

$$\theta_{\delta h_t^{3,k}/\delta h_t^3} := \theta_{\delta h_t^{3,k}} / \theta_{\delta h_t^3}, \tag{127}$$

$$\theta_{\delta \tilde{h}_t^2/\delta h_t^3} := \theta_{\delta \tilde{h}_t^2} / \theta_{\delta h_t^3}. \tag{128}$$

Note that in the above, all experts have the same scale by symmetry so we do not write a maximum over experts.

$$\mathbf{f}_t - \mathbf{f}_{t-1} = \mathbf{W}_{t-1}^4 (\mathbf{h}_t^3 - \mathbf{h}_{t-1}^3) + (\mathbf{W}_t^4 - \mathbf{W}_{t-1}^4) \mathbf{h}_t^3, \tag{129}$$

$$= \theta'_{W_{t-1}^4} \theta_{\delta h_t^3} \frac{\overline{\mathbf{W}_{t-1}^4} \delta \mathbf{h}_t^3}{n} - \theta'_{\delta W_t^4} \eta \boldsymbol{\chi}_{t-1} \frac{(\mathbf{h}_{t-1}^3)^T (\mathbf{h}_t^3)}{n}, \tag{130}$$

$$\theta_{\delta f_t} (\delta \mathbf{f}_t)_\beta = \frac{1}{n} \sum_{\alpha=1}^{n} \phi \left((\overline{\mathbf{W}_{t-1}^4})_{\beta\alpha}, (\delta \mathbf{h}_t^3)_\alpha, (\mathbf{h}_{t-1}^3)_\alpha, (\mathbf{h}_t^3)_\alpha; \theta'_{W_{t-1}^4}, \theta_{\delta h_t^3}, \theta'_{\delta W_t^4}, \eta, (\boldsymbol{\chi}_{t-1})_\beta \right) \quad \texttt{(Moment)} \tag{131}$$

$$\phi(\ldots) := \theta'_{W_{t-1}^4} \theta_{\delta h_t^3} (\overline{\mathbf{W}_{t-1}^4})_{\beta\alpha} (\delta \mathbf{h}_t^3)_\alpha - \theta'_{\delta W_t^4} \eta (\boldsymbol{\chi}_{t-1})_\beta (\mathbf{h}_{t-1}^3)_\alpha (\mathbf{h}_t^3)_\alpha \tag{132}$$

$$\theta_{\delta f_t} := \max(\theta'_{W_{t-1}^4} \theta_{\delta h_t^3}, \theta'_{\delta W_t^4}). \tag{133}$$

C.2. Infinite Width Limit for Soft MoE with SGD

Under stable parametrisations (Theorem 9), all the θ parameters in the program will converge to either zero or one.

Z random variables: According to the Master Theorem (Yang and Hu, 2021), the distribution of the coordinates of each vector z in the program will follow an iid distribution which we refer to as Z^z. Each scalar in the program $\bullet$ converges to a deterministic limit that we label $\mathring{\bullet}$. We recursively define each Z distribution based on the operation in the program used to create it.

Z Init: Let $\mathcal{V}_I$ be the set of initial vectors. For each $z \in \mathcal{V}_I$, we set $\hat{Z}^z := Z^z$ and $\dot{Z}^z := 0$ (defined below).

Z Nonlin: Given $\phi : \mathbb{R}^k \times \mathbb{R}^l \to \mathbb{R}$, scalars $\theta_1, \ldots, \theta_l \in \mathbb{R}$, and vectors $x^1, \ldots, x^k \in \mathbb{R}^n$, we have

$$Z^{\phi(x^1,\ldots,x^k;\theta_1,\ldots,\theta_l)} := \phi(Z^{z^1}, \ldots, Z^{x^k}; \mathring{\theta}_1, \ldots, \mathring{\theta}_l). \tag{134}$$

Z Moment: Given a scalar $\theta = \frac{1}{n} \sum_{\alpha=1}^{n} \phi(x_\alpha^1, \ldots, x_\alpha^k; \theta_1, \ldots, \theta_l)$, we have

$$\mathring{\theta} := \mathbb{E}[\phi(Z^{z^1}, \ldots, Z^{x^k}; \mathring{\theta}_1, \ldots, \mathring{\theta}_l)], \tag{135}$$

where the expectation is over $Z^{z^1}, \ldots, Z^{x^k}$.

Z MatMul: $Z^{Wx} := \hat{Z}^{Wx} + \dot{Z}^{Wx}$ for every matrix $W \sim \mathcal{N}(0, \sigma_W^2/n)$ and vector x.

- $\hat{Z}^{Wx}$ is Gaussian with zero mean. Let $\mathcal{V}_W := \{Wy \text{ for some vector } y\}$. Then $\{\hat{Z}^{Wy} : Wy \in \mathcal{V}_W\}$ is jointly Gaussian with zero mean and covariance,

$$\text{Cov}\left(\hat{Z}^{Wx}, \hat{Z}^{Wy}\right) := \sigma_W^2 \mathbb{E}[Z^x Z^y] \text{ for any } Wx, Wy \in \mathcal{V}_W. \tag{136}$$

- Z^x has been computed by some operation that we unwind $Z^x = \phi(\{\hat{Z}^{W^T y^i}\}_{i=1}^k), \{\hat{Z}^{z^i}\}_{i=1}^j; \{\mathring{\theta}_i\}_{i=1}^l)$ with $z^i \notin \mathcal{V}_{W^T}$[1]. Define $\partial Z^x / \partial \hat{Z}^{W^T y^i} := \partial_i \phi(\ldots)$. Then

$$\dot{Z}^{Wx} := \sigma_W^2 \sum_{i=1}^k Z^{y^i} \mathbb{E}\left[\frac{\partial Z^x}{\partial \hat{Z}^{W^T y^i}}\right]. \tag{137}$$

Initial vectors: All our initial vectors have coordinates distributed according to $\mathcal{N}(0,1)$ by construction. The set of initial vectors are $\mathcal{V} := \{\overline{\mathbf{W}^1}, \overline{\mathbf{W}^2}, \overline{\mathbf{W}^4}\}$. Thus, $Z^W = \mathcal{N}(0,1)$ for $W \in \mathcal{V}$.

First Forward Pass:

$$Z^{h_0^1} = \sum_{j=1}^{d_{\text{in}}} \xi_j Z^{W_{0,j}^1}, \tag{138}$$

$$\mathring{\mathbf{h}}_0^2 = \mathbf{b}_0^2, \tag{139}$$

$$\mathring{\tilde{\mathbf{h}}}_0^2 = G(\mathbf{b}_0^2), \tag{140}$$

$$Z^{h_0^{3,k}} = Z^{W_0^{3,k} h_0^1} \text{ for every expert } k, \tag{141}$$

$$Z^{h_0^3} = \sum_{k=1}^m \mathring{\tilde{h}}_{0,k}^2 Z^{h_0^{3,k}} \tag{142}$$

$$\mathring{\mathbf{f}}_0 = \mathbf{0}. \tag{143}$$

Here, $Z^{W_0^1}$ is standard Gaussian by construction. $\mathring{\mathbf{h}}_0^2 = \mathbf{b}_0^2$ as the non bias router logits tend to zero $\mathbb{E}[Z^{W_0^2}]\mathbb{E}[Z^{h_0^1}] = 0$. $Z^{W_0^{3,k} h_0^1} = \hat{Z}^{W_0^{3,k} h_0^1}$ from the Z MatMul rule. Finally $\mathring{\mathbf{f}}_0 = \mathbf{0}$ as $\mathring{f}_0 = \mathbb{E}[Z^{W_0^4}]\mathbb{E}[Z^{h_0^3}] = 0$.

First Backward Pass: For $d_{\text{out}} > 1$, we index the distributions by β,

$$Z^{(\mathrm{d}h_0^3)_\beta} = Z^{(\overline{W_0^4})_\beta}, \tag{144}$$

$$\mathring{\mathrm{d}\tilde{h}}_{0,k}^2 = 0 \text{ for every expert } k, \tag{145}$$

$$\mathring{\mathrm{d}h}^2{}_{0,k} = 0 \text{ for every expert } k, \tag{146}$$

$$Z^{(\mathrm{d}h_0^{3,k})_\beta} = \mathring{\tilde{h}}_{0,k}^2 Z^{(\mathrm{d}h_0^3)_\beta} \text{ for every expert } k, \tag{147}$$

$$Z^{(\mathrm{d}h_0^1)_\beta} = \sum_{k=1}^m Z^{(W_0^{3,k})^T (\mathrm{d}h_0^{3,k})_\beta} \tag{148}$$

$$\tag{149}$$

1. z^i is not a function of W^T.

$Z^{\overline{W_0^4}}$ is a standard Gaussian by construction. $\mathring{\mathrm{d}\tilde{h}}^2_{0,k} = \mathbb{E}[Z^{h_0^{3,k}}]\mathbb{E}[\overline{W_0^4}] = 0$. From this, $\mathring{\mathrm{d}h}^2_{0,k} = 0$ follows. $Z^{(W_0^{3,k})^T \mathrm{d}h_0^3} = \hat{Z}^{(W_0^{3,k})^T \mathrm{d}h_0^3}$ from the Z MatMul rule.

First Feature Update:

$$Z^{\delta h_1^1} = -\eta \sum_{\beta=1}^{d_{\text{out}}} Z^{(\mathrm{d}h_0^1)_\beta} (\mathring{\boldsymbol{\chi}}_0)_\beta \boldsymbol{\xi}^T \boldsymbol{\xi}, \tag{150}$$

$$\delta \mathring{h}^2_{1,k} = \mathring{\theta}_{\delta h_1^1} \mathbb{E}[Z^{(\overline{W_0^2})_{k,:}} Z^{\delta h_1^1}] - \mathring{\theta}'_{\delta W_1^2} \eta \sum_{\beta=1}^{d_{\text{out}}} (\mathrm{d}\mathring{h}^2_{0,k})_\beta (\mathring{\boldsymbol{\chi}}_0)_\beta \mathbb{E}[Z^{h_0^1} Z^{h_1^1}] - \eta \mathrm{d}\mathring{\mathbf{h}}_0^2 \mathring{\boldsymbol{\chi}}_0, , \tag{151}$$

$$\delta \mathring{\tilde{\mathbf{h}}}_1^2 = \begin{cases} \mathrm{G}(\mathring{\mathbf{h}}_1^2) - \mathrm{G}(\mathring{\mathbf{h}}_0^2) \text{ if } \mathring{\theta}_{\delta h_1^2} = 1 \\ \mathrm{G}'(\mathring{\mathbf{h}}_0^2)(\delta \mathring{\mathbf{h}}_1^2) \text{ if } \mathring{\theta}_{\delta h_1^2} = 0 \end{cases}, \tag{152}$$

$$Z^{\delta h_1^{3,k}} = \mathring{\theta}_{\delta h_1^1 / \delta h_1^{3,k}} Z^{W_0^{3,k} \delta h_1^1} - \mathring{\theta}'_{\delta W_1^3 / \delta h_1^{3,k}} \eta \sum_{\beta=1}^{d_{\text{out}}} Z^{(\mathrm{d}h^{3,k})_\beta} (\mathring{\boldsymbol{\chi}}_0)_\beta \mathbb{E}[Z^{h_0^1} Z^{h_1^1}] \text{ for every expert } k, \tag{153}$$

$$Z^{\delta h_1^3} = \sum_{k=1}^{m} \mathring{\theta}_{\delta h_1^{3,k} / \delta h_1^3} \mathring{\tilde{h}}^2_{0,k} Z^{\delta h_1^{3,k}} + \mathring{\theta}_{\delta \tilde{h}_1^2} \mathring{\theta}_{h_1^{3,k} / \delta h^3} \delta \mathring{\tilde{h}}^2_{1,k} Z^{h_1^{3,k}}, \tag{154}$$

$$\delta \mathring{f}_1 = \mathring{\theta}_{\delta h_1^3} \mathbb{E}[Z^{\overline{W_0^4}} Z^{\delta h_1^3}] - \mathring{\theta}'_{\delta W_1^4} \eta \mathring{\chi}_0 \mathbb{E}[Z^{h_0^3} Z^{h_1^3}], \tag{155}$$

where G is either softmax or sigmoid and G′ is the derivative. Further, $Z^{W_0^{3,k} \delta h_1^1}$ is given by the Z MatMul rule,

$$Z^{W_0^{3,k} \delta h_1^1} = \hat{Z}^{W_0^{3,k} \delta h_1^1} + \dot{Z}^{W_0^{3,k} \delta h_1^1} \tag{156}$$

$$= \hat{Z}^{W_0^{3,k} \delta h_1^1} + \sum_{\beta=1}^{d_{\text{out}}} Z^{(\mathrm{d}h_0^{3,k})_\beta} \mathbb{E}\left[\frac{\partial Z^{\delta h_1^1}}{\partial \hat{Z}^{(W_0^{3,k})^T (\mathrm{d}h_0^{3,k})_\beta}} \right]. \tag{157}$$

We write the limit of each vector in the program as

$$Z^{h_t} = Z^{h_0} + \mathring{\theta}_{\delta h_1} Z^{\delta h_1} + \ldots + \mathring{\theta}_{\delta h_t} Z^{\delta h_t}. \tag{158}$$

Weight updates: The distribution of certain weight updates are required in the backward pass and thus the forward passes.

$$Z^{\overline{\delta W_t^4}} = -\eta \mathring{\chi}_{t-1} Z^{h_{t-1}^3}, \tag{159}$$

$$Z^{(\overline{\delta W_t^2})_{k,:}} = -\eta \sum_{\beta=1}^{d_{\text{out}}} (\mathrm{d}\mathring{h}^2_{t-1,k})_\beta (\mathring{\boldsymbol{\chi}}_{t-1})_\beta Z^{h_{t-1}^1}, \tag{160}$$

$$\delta \mathring{\mathbf{b}}^2{}_t = -\eta \mathring{\theta}'_{W_{t-1}^4} \mathrm{d}\mathring{\mathbf{h}}^2_{t-1} \mathring{\boldsymbol{\chi}}_{t-1}. \tag{161}$$

tth Backward Pass:

$$Z^{(\mathrm{d}h_t^3)_\beta} = Z^{(\overline{W_t^4})_\beta^T}, \tag{162}$$

$$(\mathrm{d}\mathring{\tilde{h}}_{t,k}^2)_\beta = \mathbb{E}[Z^{h_t^{3,k}} Z^{(\mathrm{d}h_t^3)_\beta}], \tag{163}$$

$$(\mathrm{d}\mathring{h}_{t,k}^2)_\beta = \sum_{j=1}^{m} J(\mathring{h}_t^2)_{jk} (\mathrm{d}\mathring{\tilde{h}}_{t,j}^2)_\beta, \tag{164}$$

$$Z^{(\mathrm{d}h_t^{3,k})_\beta} = \mathring{\tilde{h}}_{t,k}^2 Z^{(\mathrm{d}h_t^3)_\beta}, \tag{165}$$

$$Z^{(\mathrm{d}h_t^1)_\beta} = \sum_{k=1}^{m} \left(Z^{(\overline{W_t^2})_{k,:}} (\mathrm{d}\mathring{h}_{t,k}^2)_\beta + Z^{(W_0^{3,k})^T (\mathrm{d}h_t^{3,k})_\beta} - \sum_{s=1}^{t-1} Z^{h_s^1} \sum_{\gamma=1}^{d_{\text{out}}} (\mathring{\chi}_s)_\gamma \mathbb{E}[Z^{(\mathrm{d}h_s^{3,k})_\gamma} Z^{(\mathrm{d}h_t^{3,k})_\beta}] \right), \tag{166}$$

where $Z^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}}$ is given by the Z MatMul rule,

$$Z^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}} = \hat{Z}^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}} + \dot{Z}^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}} \tag{167}$$

$$= \hat{Z}^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}} + \sum_{v \in \mathcal{V}: W_0^{3,k} v \in \mathcal{V}} Z^v \mathbb{E}\left[\frac{\partial Z^{\mathrm{d}h_t^{3,k}}}{\partial \hat{Z}^{W_0^{3,k} v}} \right], \tag{168}$$

where $\mathcal{V} := \{v \text{ for some vector in the program } v\}$ is the set of all vectors in the program.

tth Forward Pass:

$$Z^{\delta h_t^1} = -\eta \sum_{\beta=1}^{d_{\text{out}}} Z^{(\mathrm{d}\mathbf{h}_{t-1}^1)_\beta} (\mathring{\boldsymbol{\chi}}_{t-1})_\beta \boldsymbol{\xi}^T \boldsymbol{\xi}, \tag{169}$$

$$\delta \mathring{h}_{t,k}^2 = \mathring{\theta}'_{W_{t-1}^2} \mathring{\theta}_{\delta h_t^1} \mathbb{E}[Z^{(\overline{W_{t-1}^2})_{k,:}} Z^{\delta h_t^1}] - \mathring{\theta}'_{\delta W_t^2} \eta \sum_{\beta=1}^{d_{\text{out}}} (\mathrm{d}\mathring{h}_{t-1,k}^2)_\beta (\mathring{\boldsymbol{\chi}}_{t-1})_\beta \mathbb{E}[Z^{h_{t-1}^1} Z^{h_t^1}] + \delta \mathring{\mathbf{b}}^2 t, \tag{170}$$

$$\delta \mathring{\tilde{\mathbf{h}}}_t^2 = \begin{cases} \mathrm{G}(\mathring{\mathbf{h}}_t^2) - \mathrm{G}(\mathring{\mathbf{h}}_{t-1}^2) \text{ if } \mathring{\theta}_{\delta h_t^2} = 1 \\ \mathrm{G}'(\mathring{\mathbf{h}}_{t-1}^2)(\delta \mathring{\mathbf{h}}_t^2) \text{ if } \mathring{\theta}_{\delta h_t^2} = 0 \end{cases}, \tag{171}$$

$$Z^{\delta h_t^{3,k}} = \mathring{\theta}_{\delta h_t^1 / \delta h_t^{3,k}} Z^{W_0^{3,k} \delta h_t^1} - \sum_{s=0}^{t-2} \mathring{\theta}'_{\delta W_{s+1}^{3,k} \delta h_t^1 / \delta h_t^{3,k}} \eta \sum_{\beta}^{d_{\text{out}}} Z^{(\mathrm{d}h_s^{3,k})_\beta} (\mathring{\boldsymbol{\chi}}_s)_\beta \mathbb{E}[Z^{h_s^1} Z^{\delta h_t^1}] \tag{172}$$

$$- \mathring{\theta}'_{\delta W_t^{3,k} / \delta h_t^{3,k}} \eta \sum_{\beta}^{d_{\text{out}}} Z^{(\mathrm{d}h_{t-1}^{3,k})_\beta} (\mathring{\boldsymbol{\chi}}_{t-1})_\beta \mathbb{E}[Z^{h_{t-1}^1} Z^{h_t^1}], \tag{173}$$

$$Z^{\delta h_t^3} = \sum_{k=1}^{m} \mathring{\theta}_{\delta h_t^{3,k} / \delta h_t^3} \mathring{\tilde{h}}_{k,t-1}^2 Z^{\delta h_t^{3,k}} + \mathring{\theta}_{\delta \tilde{h}_t^2 / \delta h_t^3} \delta \mathring{\tilde{h}}_{t,k}^2 Z^{h_t^{3,k}}, \tag{174}$$

$$(\delta \mathring{f}_t)_\beta = \mathring{\theta}'_{W_{t-1}^4} \mathring{\theta}_{\delta h_t^3} \mathbb{E}[Z^{(\overline{W_{t-1}^4})_\beta} Z^{\delta h_t^3}] - \mathring{\theta}'_{\delta W_t^4} \eta (\mathring{\chi}_{t-1})_\beta \mathbb{E}[Z^{h_{t-1}^3} Z^{h_t^3}], \tag{175}$$

where G is either softmax or sigmoid and G′ is the derivative, and

$$Z^{W_0^{3,k}\delta h_t^1} = \hat{Z}^{W_0^{3,k}\delta h_t^1} + \dot{Z}^{W_0^{3,k}\delta h_t^1} \tag{176}$$

$$= \hat{Z}^{W_0^{3,k}\delta h_t^1} + \sum_{s=0}^{t-1}\sum_{\beta=1}^{d_{\text{out}}} Z^{(\mathrm{d}h_s^{3,k})_\beta}\mathbb{E}\left[\frac{\partial Z^{\delta h_t^1}}{\partial \hat{Z}^{(W_0^{3,k})^T(\mathrm{d}h_s^{3,k})_\beta}}\right] \tag{177}$$

C.3. Tensor Program for Soft MoE with Adam

We consider Adam optimiser updates here.

Tensor Program: We use NE⊗OR⊤ for the Adam limit (Yang and Littwin, 2023). This has the same operations as NETSOR⊤$^+$ (Appendix C.1), but with a generalisation of the `Nonlin`.

- `OuterNonlin`: Given $r \geq 0$, vectors $\mathbf{x}^0, \ldots, \mathbf{x}^r$, and function $\psi : \mathbb{R}^{(r+1)+\ell} \to \mathbb{R}$, generate a new vector ($\mathbb{R}^n$) as

$$y_\alpha = \frac{1}{n^r}\sum_{\beta_1,\ldots,\beta_r=1}^{n} \psi(x_\alpha^0; x_{\beta_1}^1; \ldots; x_{\beta_r}^r; \theta_1, \ldots, \theta_l)$$

Definition 13 (Adam update) *Given gradients $g_0, \ldots, g_t$ for a parameter $\mathbf{W}$, the Adam update at step t with learning rate η is,*

$$\hat{m} \leftarrow \frac{1}{1-\beta_1^{t+1}}\sum_{s=0}^{t}(1-\beta_1)\beta_1^{t-s}g_s,$$

$$\hat{v} \leftarrow \sqrt{\frac{1}{1-\beta_2^{t+1}}\sum_{s=0}^{t}(1-\beta_2)\beta_2^{t-s}g_s^2 + \epsilon^2},$$

$$\mathbf{W} \leftarrow \mathbf{W} - \eta\left(\frac{\hat{m}}{\sqrt{\hat{v}+\epsilon^2}}\right),$$

where (β_1, β_2) are momentum parameters and $\epsilon > 0$ is a constant.

bcd parametrisation for Adam updates: The parameters b and c play the same role as in Appendix C.1. We parametrise the weights for layer l as $\mathbf{W}^l = n^{-b_l}\overline{\mathbf{W}}^l$. The learning rate for the parameter $\mathbf{W}^l$ is ηn^{-c_l}, where η is some width independent constant. The update function for layer l at time t is Q_t^l, which takes in scaled gradients $Q_t^l(n^{d_l}g_0, \ldots, n^{d_l}g_t)$. For Adam the update function is

$$Q_t^l(n^{d_l}g_0, \ldots, n^{d_l}g_t) = \frac{\frac{1}{1-\beta_1^{t+1}}\sum_{s=0}^{t}(1-\beta_1)\beta_1^{t-s}n^{d_l}g_s}{\sqrt{\frac{1}{1-\beta_2^{t+1}}\sum_{s=0}^{t}(1-\beta_2)\beta_2^{t-s}n^{2d_l}g_s^2 + \epsilon^2}}. \tag{178}$$

Note that instead of scaling the gradients g by n^{d_l}, we can equivalently scale $\epsilon \to n^{-d_l}\epsilon$.

Faithfulness (Theorem 7): To ensure that the inputs to the update function Q are neither too large (causing diverging behavior) or too small (trivialising the update function), we require that the inputs to the update function scale $\Theta(1)$. Equivalently, for our Adam update function, we require that the scale of $n^{-d_l}\epsilon$ is the same as the scale of the gradients $g_0, \ldots, g_t$.

We assume that the parametrisation is stable (Theorem 9), and parameter values are chosen such that the network is stable at initialisation (Theorem 11).

First Backward Pass: Same as Appendix C.1.

First Weight Updates: Recall that we define the gradient of the loss with respect to the function as $\boldsymbol{\chi}_t$. For the vectors $\mathbf{W} \in \{\mathbf{W}^1, \mathbf{W}^2, \mathbf{W}^4\}$ we write $\delta\mathbf{W}_t := \theta^{-1}_{\delta W_t}(\mathbf{W}_t - \mathbf{W}_{t-1})$, and thus the weights at step t as $\mathbf{W}_t = \theta_{W_0}\overline{\mathbf{W}_0} + \ldots + \theta_{\delta W_t}\delta W_t$.

$$\mathbf{W}^4_1 - \mathbf{W}^4_0 = -n^{-c_4}\eta Q^4_1\left(n^{d_4}\boldsymbol{\chi}_0(\mathbf{h}^3_0)^T\right) \tag{179}$$

$$\theta_{\delta W^4_1} := n^{-c_4}. \tag{180}$$

$$\mathbf{W}^{3,k}_1 - \mathbf{W}^{3,k}_0 = -n^{-c_3}\eta Q^3_1\left(n^{(d_3-1)}\mathrm{d}\mathbf{h}^{3,k}_0\boldsymbol{\chi}_0(\mathbf{h}^1_0)^T\right), \tag{181}$$

$$\theta_{\delta W^3_1} := n^{-c_3}. \tag{182}$$

$$\mathbf{W}^2_1 - \mathbf{W}^2_0 = -n^{-c_2}\eta Q^2_1\left(n^{d_2}n\theta_{W^4_0}\mathrm{d}\mathbf{h}^2_0\boldsymbol{\chi}_0(\mathbf{h}^1_0)^T\right) \tag{183}$$

$$= -n^{-c_2}\eta Q^2_1\left(n^{d_2}\mathrm{d}\mathbf{h}^2_0\boldsymbol{\chi}_0(\mathbf{h}^1_0)^T\right) \tag{184}$$

$$\theta_{\delta W^2_1} := n^{-c_2}. \tag{185}$$

$$\mathbf{W}^1_1 - \mathbf{W}^1_0 = -n^{-c_1}\eta Q^1_1\left(n^{d_1}n\theta_{W^2_0}\theta_{W^4_0}\mathrm{d}\mathbf{h}^1_0\boldsymbol{\chi}_0(\boldsymbol{\xi})^T\right) \tag{186}$$

$$= -n^{-c_1}\eta Q^1_1\left(n^{(d_1-1)}\mathrm{d}\mathbf{h}^1_0\boldsymbol{\chi}_0(\boldsymbol{\xi})^T\right) \tag{187}$$

$$\theta_{\delta W^1_1} := n^{-c_1}. \tag{188}$$

The router bias in the learnable case updates as

$$\mathbf{b}^2_1 - \mathbf{b}^2_0 = -\eta\theta'_{W^4_0}\mathrm{d}\mathbf{h}^2_0\boldsymbol{\chi}_0, \tag{189}$$

where due to stability at initialisation (Theorem 11) $\theta'_{W^4_0} = 1$.

First Feature Updates: The feature updates are affected by the change in the previous layers features (*propagating updates*) and the update in the weights at the current layer (*effective updates*). We write the feature updates for the vectors $\mathbf{h} \in \{\mathbf{h}^1, \{\mathbf{h}^{3,k}\}^m_{k=1}, \mathbf{h}^3\}$ as $\delta\mathbf{h}_t := \theta^{-1}_{\delta h_t}(\mathbf{h}_t - \mathbf{h}_{t-1})$. This gives each features for the vectors at step t as $\mathbf{h}_t = \mathbf{h}_0 + \theta_{\delta h_1}\delta\mathbf{h}_1 + \ldots + \theta_{\delta h_t}\delta\mathbf{h}_t$. For the scalars $\mathbf{h} \in \{\mathbf{h}^2, \tilde{\mathbf{h}}^2, \mathbf{f}\}$ we simply write $\delta\mathbf{h}_t := (\mathbf{h}_t - \mathbf{h}_{t-1})$ and $\mathbf{h}_t = \mathbf{h}_0 + \ldots + \delta\mathbf{h}_t$.

$$\mathbf{h}_1^1 - \mathbf{h}_0^1 = (\mathbf{W}_1^1 - \mathbf{W}_0^1)\boldsymbol{\xi} \tag{190}$$

$$= -\theta_{\delta W_1^1}\eta Q_1^1 \left(n^{d_1} n^{-1} \mathrm{d}\mathbf{h}_0^1 \boldsymbol{\chi}_0 \boldsymbol{\xi}^T\right) \boldsymbol{\xi} \tag{191}$$

$$(\delta \mathbf{h}_1^1)_\alpha = \phi\left(\{(\mathrm{d}\mathbf{h}_0^1)_{\alpha\beta}\}_{\beta=1}^{d_{\text{out}}}; -\eta, n^{(d_1-1)}, \{(\boldsymbol{\chi}_0)_\beta\}_{\beta=1}^{d_{\text{out}}}, \boldsymbol{\xi}\right) \qquad \texttt{(Nonlin)} \tag{192}$$

$$\phi(\ldots) := -\eta Q_1^1 \left(n^{(d_1-1)} \sum_{\beta=1}^{d_{\text{out}}} (\mathrm{d}\mathbf{h}_0^1)_{\alpha\beta} (\boldsymbol{\chi}_0)_\beta \boldsymbol{\xi}^T\right) \boldsymbol{\xi}, \tag{193}$$

$$\theta_{\delta h_1^1} := \theta_{\delta W_1^1}, \tag{194}$$

$$\mathbf{h}_1^2 - \mathbf{h}_0^2 = \mathbf{W}_0^2(\mathbf{h}_1^1 - \mathbf{h}_0^1) + (\mathbf{W}_1^2 - \mathbf{W}_0^2)\mathbf{h}_1^1 + (\mathbf{b}_1^2 - \mathbf{b}_0^2) \tag{195}$$

$$= \theta_{\delta h_1^1}\theta'_{W_0^2} \frac{\overline{\mathbf{W}_0^2}\delta \mathbf{h}_1^1}{n} - \theta'_{\delta W_1^2}\eta \frac{1}{n} Q_1^2 \left(n^{d_2} \mathrm{d}\mathbf{h}_0^2 \boldsymbol{\chi}_0 (\mathbf{h}_0^1)^T\right) \mathbf{h}_1^1 - \eta \mathrm{d}\mathbf{h}_0^2 \boldsymbol{\chi}_0, \tag{196}$$

$$\theta_{\delta h_1^2}(\delta \mathbf{h}_1^2)_k = \frac{1}{n}\sum_{\alpha=1}^{n} \phi\left(\overline{\mathbf{W}_0^2}_{k\alpha}, (\delta \mathbf{h}_1^1)_\alpha, (\mathbf{h}_0^1)_\alpha, (\mathbf{h}_1^1)_\alpha; \theta_{\delta h_1^1}, \theta'_{W_0^2}, \theta'_{\delta W_1^2}, n^{d_2}, \eta, \{(\mathrm{d}\mathbf{h}_0^2)_{k\beta}, (\boldsymbol{\chi}_0)_\beta\}_\beta^{d_{\text{out}}}\right) \quad \texttt{(Moment)} \tag{197}$$

$$\phi(\ldots) := \theta_{\delta h_1^1}\theta'_{W_0^2} \overline{\mathbf{W}_0^2}_{k\alpha} (\delta \mathbf{h}_1^1)_\alpha - \theta'_{\delta W_1^2}\eta \frac{1}{n} Q_1^2 \left(n^{d_2} \sum_{\beta=1}^{d_{\text{out}}} (\mathrm{d}\mathbf{h}_0^2)_{k\beta} (\boldsymbol{\chi}_0)_\beta (\mathbf{h}_0^1)_\alpha\right) (\mathbf{h}_1^1)_\alpha \tag{198}$$

$$- \eta \theta'_{W_0^4} \mathrm{d}\mathbf{h}_0^2 \boldsymbol{\chi}_0, \tag{199}$$

$$\theta_{\delta h_1^2} := \max(\theta_{\delta h_1^1}\theta'_{W_0^2}, \theta'_{\delta W_1^2}), \tag{200}$$

$$\tilde{\mathbf{h}}_1^2 - \tilde{\mathbf{h}}_0^2 = \mathrm{G}(\mathbf{h}_1^2) - \mathrm{G}(\mathbf{h}_0^2) \tag{201}$$

$$= \mathrm{G}\left(\mathbf{h}_0^2 + \theta_{\delta h_1^2}\delta \mathbf{h}_1^2\right) - \mathrm{G}(\mathbf{h}_0^2) \tag{202}$$

$$\theta_{\delta \tilde{h}_1^2}(\delta \tilde{\mathbf{h}}_1^2)_k = \frac{1}{n}\sum_{\alpha=1}^{n} \phi_k\left(-; \{(\mathbf{h}_0^2)_k, (\delta \mathbf{h}_1^2)_k\}_{k=1}^m\right), \qquad \texttt{(Moment)}, \tag{203}$$

$$\theta_{\delta \tilde{h}_1^2} := \theta_{\delta h_1^2}. \tag{204}$$

where G is either softmax or sigmoid and can be represented as a Moment (see Appendix G) (Yang and Hu, 2021).

$$\mathbf{h}_1^{3,k} - \mathbf{h}_0^{3,k} = \mathbf{W}_0^{3,k}(\mathbf{h}_1^1 - \mathbf{h}_0^1) + (\mathbf{W}_1^{3,k} - \mathbf{W}_0^{3,k})\mathbf{h}_1^1, \tag{205}$$

$$= \theta_{\delta h_1^1}\mathbf{W}_0^{3,k}\delta\mathbf{h}_1^1 - \theta'_{\delta W_1^3}\eta\frac{1}{n}Q_1^3\left(n^{(d_3-1)}\mathrm{d}\mathbf{h}_0^{3,k}\boldsymbol{\chi}_0(\mathbf{h}_0^1)^T\right)(\mathbf{h}_1^1), \tag{206}$$

$$(\delta\mathbf{h}_1^{3,k})_\alpha = \frac{1}{n}\sum_{\gamma=1}^{n}\phi\Big(\Upsilon_\alpha, \{(\mathrm{d}\mathbf{h}_0^{3,k})_{\alpha l}\}_{l=1}^{d_{\text{out}}}; (\mathbf{h}_0^1)_\gamma, (\mathbf{h}_1^1)_\gamma \tag{OuterNonlin}$$

$$; \theta_{\delta h_1^1/\delta h_1^{3,k}}, \theta'_{\delta W_1^3/\delta h_1^{3,k}}, n^{(d_3-1)}, \{(\boldsymbol{\chi}_0)_l\}_{l=1}^{d_{\text{out}}}, \eta\Big), \tag{207}$$

$$\Upsilon := \mathbf{W}_0^{3,k}\delta\mathbf{h}_1^1, \tag{MatMul}$$

$$\tag{208}$$

$$\phi(\ldots) := \theta_{\delta h_1^1/\delta h_1^{3,k}}\Upsilon_\alpha - \theta'_{\delta W_1^3/\delta h_1^{3,k}}\eta Q_1^3\left(n^{(d_3-1)}\sum_{l=1}^{d_{\text{out}}}(\mathrm{d}\mathbf{h}_0^{3,k})_{\alpha l}(\boldsymbol{\chi}_0)_l(\mathbf{h}_0^1)_\gamma\right)(\mathbf{h}_1^1)_\gamma, \tag{209}$$

$$\theta_{\delta h_1^{3,k}} = \max(\theta_{\delta h_1^1}, \theta'_{\delta W_1^3}), \tag{210}$$

$$\theta_{\delta h_1^1/\delta h_1^{3,k}} := \theta_{\delta h_1^1}/\theta_{\delta h_1^{3,k}}, \tag{211}$$

$$\theta'_{\delta W_1^3/\delta h_1^{3,k}} := \theta'_{\delta W_1^3}/\theta_{\delta h_1^{3,k}}. \tag{212}$$

We repeat the above process for each expert k to give $\{\delta\mathbf{h}_1^{3,k}\}_{k=1}^m$.

$$\mathbf{h}_1^3 - \mathbf{h}_0^3 = \sum_{k=1}^{m}\left(\tilde{h}_{0,k}^2(\mathbf{h}_1^{3,k} - \mathbf{h}_0^{3,k}) + (\tilde{h}_{1,k}^2 - \tilde{h}_{0,k}^2)\mathbf{h}_1^{3,k}\right) \tag{213}$$

$$= \sum_{k=1}^{m}\left(\theta_{\delta h_1^{3,k}}\tilde{h}_{0,k}^2\delta\mathbf{h}_1^{3,k} + \theta_{\delta\tilde{h}_1^2}\delta\tilde{h}_{1,k}^2\mathbf{h}_1^{3,k}\right), \tag{214}$$

$$(\delta\mathbf{h}_1^3)_\alpha = \phi\left(\{(\delta\mathbf{h}_1^{3,k})_\alpha\}_{k=1}^m, \{(\mathbf{h}_1^{3,k})_\alpha\}_{k=1}^m; \theta_{\delta h_1^{3,k}/\delta h_1^3}, \theta_{\delta\tilde{h}_1^2/\delta h_1^3}, \{\tilde{h}_{0,k}^2\}_{k=1}^m, \{\delta\tilde{h}_{1,k}^2\}_{k=1}^m\right), \tag{Nonlin}$$

$$\tag{215}$$

$$\phi(\ldots) := \sum_{k=1}^{m}\left(\theta_{\delta h_1^{3,k}/\delta h_1^3}\tilde{h}_{0,k}^2(\delta\mathbf{h}_1^{3,k})_\alpha + \theta_{\delta\tilde{h}_1^2/\delta h_1^3}\delta\tilde{h}_{1,k}^2(\mathbf{h}_1^{3,k})_\alpha\right), \tag{216}$$

$$\theta_{\delta h_1^3} := \max(\theta_{\delta h_1^{3,k}}, \theta_{\delta\tilde{h}_1^2}), \tag{217}$$

$$\theta_{\delta h_1^{3,k}/\delta h_1^3} := \theta_{\delta h_1^{3,k}}/\theta_{\delta h_1^3}, \tag{218}$$

$$\theta_{\delta\tilde{h}_1^2/\delta h_1^3} := \theta_{\delta\tilde{h}_1^2}/\theta_{\delta h_1^3}. \tag{219}$$

$$\mathbf{f}_1 - \mathbf{f}_0 = \mathbf{W}_0^4(\mathbf{h}_1^3 - \mathbf{h}_0^3) + (\mathbf{W}_1^4 - \mathbf{W}_0^4)\mathbf{h}_1^3 \tag{220}$$

$$= \theta'_{W_0^4}\theta_{\delta h_1^3}\frac{\overline{\mathbf{W}_0^4}\delta\mathbf{h}_1^3}{n} - \theta'_{\delta W_1^4}\eta\frac{1}{n}Q_1^4\left(n^{d_4}\boldsymbol{\chi}_0(\mathbf{h}_0^3)^T\right)\mathbf{h}_1^3, \tag{221}$$

$$\theta_{\delta f_1}(\delta\mathbf{f}_0)_\beta = \frac{1}{n}\sum_{\alpha=1}^{n}\phi\left(\overline{\mathbf{W}_0^4}_{\beta\alpha}, (\delta\mathbf{h}_1^3)_\alpha, (\mathbf{h}_0^3)_\alpha, (\mathbf{h}_1^3)_\alpha; \theta'_{W_0^4}, \theta_{\delta h_1^3}, \theta'_{\delta W_1^4}, n^{d_4}, \eta, (\boldsymbol{\chi}_0)_\beta\right), \quad \texttt{(Moment)} \tag{222}$$

$$\phi(\ldots) := \theta'_{W_0^4}\theta_{\delta h_1^3}(\overline{\mathbf{W}_0^4})_{\beta\alpha}(\delta\mathbf{h}_1^3)_\alpha - \theta'_{\delta W_1^4}\eta Q_1^4\left(n^{d_4}(\boldsymbol{\chi}_0)_\beta(\mathbf{h}_0^3)_\alpha\right)(\mathbf{h}_1^3)_\alpha, \tag{223}$$

$$\theta_{\delta f_1} := \max(\theta'_{W_0^4}\theta_{\delta h_1^4}, \theta'_{\delta W_1^4}). \tag{224}$$

tth Backward Pass: Same as Appendix C.1 except for the gradient $\mathrm{d}\mathbf{h}_t^1$.

$$\mathrm{d}\mathbf{h}_t^1 = \sum_{k=1}^{m}(\overline{\mathbf{W}_t^2})_{:,k}^T\mathrm{d}h_{t,k}^2 + \sum_{k=1}^{m}(\mathbf{W}_0^{3,k})^T\mathrm{d}\mathbf{h}_t^{3,k}$$
$$- \sum_{k=1}^{m}\sum_{s=0}^{t-1}\eta\frac{1}{n}Q_s^1\left(n^{(d_3-1)}\mathbf{h}_s^1(\boldsymbol{\chi}_s)^T(\mathrm{d}\mathbf{h}_s^{3,k})^T\right)\mathrm{d}\mathbf{h}_t^{3,k}, \quad \in \mathbb{R}^{n\times d_{\text{out}}}, \tag{225}$$

$$(\mathrm{d}\mathbf{h}_t^1)_{\alpha\beta} = \frac{1}{n}\sum_{\gamma=1}^{n}\phi\Big(\{(\overline{\mathbf{W}_t^2})_{\alpha k}^T, (\mathbf{V}_t^k)_{\alpha\beta}\}_{k=1}^m, \{(\mathbf{h}_s^1)_\alpha\}_{s=1}^{t-1};$$
$$\{(\mathrm{d}\mathbf{h}_s^{3,k})_{\gamma\beta}\}_{s=1}^{t-1}, (\mathrm{d}\mathbf{h}_t^{3,k})_{\gamma\beta}; n^{(d_3-1)}, \{(\mathrm{d}h_{t,k}^2)_\beta\}_{k=1}^m, \{(\boldsymbol{\chi}_s)_\beta\}_{s=1}^{t-1}\Big), \quad \texttt{(Outer} \tag{226}$$

$$(\mathbf{V}_t^k)_{:,\beta} := (\mathbf{W}_0^{3,k})^T(\mathrm{d}\mathbf{h}_t^{3,k})_{:,\beta}, \quad \texttt{(MatMu} \tag{227}$$

$$\phi(\ldots) := \sum_{k=1}^{m}(\overline{\mathbf{W}_t^2})_{\alpha k}^T(\mathrm{d}h_{t,k}^2)_\beta + \sum_{k=1}^{m}(\mathbf{V}_t^k)_{\alpha\beta} - \sum_{k=1}^{m}\sum_{s=0}^{t-1}\eta\frac{1}{n}Q_s^1\left(n^{(d_3-1)}(\mathbf{h}_s^1)_\alpha(\boldsymbol{\chi}_s)_\beta(\mathrm{d}\mathbf{h}_s^{3,k})_{\gamma\beta}\right)(\mathrm{d}\mathbf{h}_t^{3,k})_{\gamma\beta}, \tag{228}$$

tth Weight Updates:

$$\mathbf{W}_t^4 - \mathbf{W}_{t-1}^4 = -n^{-c_4}\eta Q_t^4\left(n^{d_4}\boldsymbol{\chi}_{t-1}(\mathbf{h}_{t-1}^3)^T\right) \tag{229}$$

$$\theta_{\delta W_t^4} := n^{-c_4}. \tag{230}$$

$$\mathbf{W}_t^{3,k} - \mathbf{W}_{t-1}^{3,k} = -n^{-c_3}\eta Q_t^3\left(n^{d_3}\theta_{W_{t-1}^4}\mathrm{d}\mathbf{h}_{t-1}^{3,k}\boldsymbol{\chi}_{t-1}(\mathbf{h}_{t-1}^1)^T\right) \tag{231}$$

$$\theta_{\delta W_t^3} := n^{-c_3}. \tag{232}$$

$$\mathbf{W}_t^2 - \mathbf{W}_{t-1}^2 = -n^{-c_2}\eta Q_t^2\left(n^{d_2}n\theta_{W_{t-1}^4}\mathrm{d}\mathbf{h}_{t-1}^2\boldsymbol{\chi}_{t-1}(\mathbf{h}_{t-1}^1)^T\right) \tag{233}$$

$$\theta_{\delta W_t^2} := n^{-c_2}. \tag{234}$$

$$\mathbf{W}_t^1 - \mathbf{W}_{t-1}^1 = -n^{-c_1}\eta Q_t^1\left(n^{d_1}n\theta_{W_{t-1}^4}\theta_{W_{t-1}^2}\mathrm{d}\mathbf{h}_{t-1}^1\boldsymbol{\chi}_{t-1}(\boldsymbol{\xi})^T\right) \tag{235}$$

$$\theta_{\delta W_t^1} := n^{-c_1}. \tag{236}$$

$$\mathbf{b}_t^2 - \mathbf{b}_{t-1}^2 = -\eta\theta'_{W_{t-1}^4}\mathrm{d}\mathbf{h}_{t-1}^2\boldsymbol{\chi}_{t-1}, \tag{237}$$

tth Feature Updates:

$$\begin{aligned}
\mathbf{h}_t^1 - \mathbf{h}_{t-1}^1 &= (\mathbf{W}_t^1 - \mathbf{W}_{t-1}^1)\boldsymbol{\xi} && (238)\\
&= -\theta_{\delta W_t^1}\eta Q_t^1\left(n^{d_1}\mathrm{d}\mathbf{h}_{t-1}^1\boldsymbol{\chi}_{t-1}\boldsymbol{\xi}^T\boldsymbol{\xi}\right) && (239)\\
(\delta\mathbf{h}_t^1)_\alpha &= \phi\left(\{(\mathrm{d}\mathbf{h}_{t-1}^1)_{\alpha\beta}\}_{\beta=1}^{d_{\text{out}}}; n^{d_1}, -\eta, \{(\boldsymbol{\chi}_{t-1}\boldsymbol{\xi}^T\boldsymbol{\xi})_\beta\}_{\beta=1}^{d_{\text{out}}}\right) \qquad \texttt{(Nonlin)} && (240)\\
\phi(\ldots) &:= -\eta Q_t^1\left(n^{d_1}\theta_{\mathrm{d}h_{t-1}^1}\sum_{\beta=1}^{d_{\text{out}}}(\mathrm{d}\mathbf{h}_{t-1}^1)_{\alpha\beta}(\boldsymbol{\chi}_{t-1}\boldsymbol{\xi}^T\boldsymbol{\xi})_\beta\right), && (241)\\
\theta_{\delta h_t^1} &:= \theta_{\delta W_t^1}, && (242)
\end{aligned}$$

$$\begin{aligned}
\mathbf{h}_t^2 - \mathbf{h}_{t-1}^2 &= \mathbf{W}_{t-1}^2(\mathbf{h}_t^1 - \mathbf{h}_{t-1}^1) + (\mathbf{W}_t^2 - \mathbf{W}_{t-1}^2)\mathbf{h}_1^t + (\mathbf{b}_t^2 - \mathbf{b}_{t-1}^2) && (243)\\
&= \theta'_{W_t^2}\theta_{\delta h_t^1}\frac{\overline{\mathbf{W}_{t-1}^2}\delta\mathbf{h}_t^1}{n} - \theta'_{\delta W_t^2}\eta\frac{1}{n}Q_t^2\left(n^{d_2}\theta'_{W_{t-1}^4}\mathrm{d}\mathbf{h}_{t-1}^2\boldsymbol{\chi}_{t-1}(\mathbf{h}_{t-1}^1)^T\right)\mathbf{h}_t^1 && (244)\\
&\quad - \eta\theta'_{W_{t-1}^4}\mathrm{d}\mathbf{h}_{t-1}^2\boldsymbol{\chi}_{t-1} && (245)\\
\theta_{h_t^2}(\delta\mathbf{h}_t^2)_k &= \frac{1}{n}\sum_{\alpha=1}^{n}\phi\Big(\overline{\mathbf{W}_{t-1}^2}_{k\alpha}, (\delta\mathbf{h}_t^1)_\alpha, (\mathbf{h}_{t-1}^1)_\alpha, (\mathbf{h}_{t-1}^1)_\alpha;\\
&\qquad \theta'_{W_t^2}, \theta_{\delta h_t^1}, \theta'_{\delta W_t^2}, n^{d_2}, \theta'_{W_{t-1}^4}, \eta, \{(\mathrm{d}\mathbf{h}_{t-1}^2)_{k\beta}, (\boldsymbol{\chi}_{t-1})_\beta\}_\beta^{d_{\text{out}}}\Big) \qquad \texttt{(Moment)} && (246)\\
\phi(\ldots) &:= \theta'_{W_{t-1}^2}\theta_{\delta h_t^1}\overline{\mathbf{W}_{t-1}^2}_{k\alpha}(\delta\mathbf{h}_t^1)_\alpha && (247)\\
&\quad - \theta'_{\delta W_t^2}\eta Q_t^2\left(n^{d_2}\theta'_{W_{t-1}^4}\sum_{\beta=1}^{d_{\text{out}}}(\mathrm{d}\mathbf{h}_{t-1}^2)_{k\beta}(\boldsymbol{\chi}_{t-1})_\beta(\mathbf{h}_{t-1}^1)_\alpha\right)(\mathbf{h}_{t-1}^1)_\alpha && (248)\\
&\quad - \eta\theta'_{W_{t-1}^4}\mathrm{d}\mathbf{h}_{t-1}^2\boldsymbol{\chi}_{t-1} && (249)\\
\theta_{\delta h_t^2} &:= \max(\theta'_{W_t^2}\theta_{\delta h_t^1}, \theta'_{\delta W_t^2}, \theta'_{W_{t-1}^4}), && (250)
\end{aligned}$$

$$\begin{aligned}
\tilde{\mathbf{h}}_t^2 - \tilde{\mathbf{h}}_{t-1}^2 &= \mathrm{G}(\mathbf{h}_t^2) - \mathrm{G}(\mathbf{h}_{t-1}^2) && (251)\\
&= \mathrm{G}\left(\mathbf{h}_{t-1}^2 + \theta_{h_t^2}\delta\mathbf{h}_t^2\right) - \mathrm{Softmax}(\mathbf{h}_{t-1}^2) && (252)\\
\theta_{\tilde{h}_t^2}(\delta\tilde{\mathbf{h}}_t^2)_k &= \frac{1}{n}\sum_{\alpha=1}^{n}\phi_k\left(-; \{(\mathbf{h}_{t-1}^2)_k, (\delta\mathbf{h}_t^2)_k\}_{k=1}^m\right), \qquad \texttt{(Moment)}, && (253)\\
\theta_{\delta\tilde{h}_t^2} &:= \theta_{\delta h_t^2}. && (254)
\end{aligned}$$

$$\mathbf{h}_t^{3,k} - \mathbf{h}_{t-1}^{3,k} = \mathbf{W}_{t-1}^{3,k}(\mathbf{h}_t^1 - \mathbf{h}_{t-1}^1) + (\mathbf{W}_t^{3,k} - \mathbf{W}_{t-1}^{3,k})\mathbf{h}_t^1, \tag{255}$$

$$= \theta_{\delta h_t^1}\mathbf{W}_0^{3,k}\overline{\delta\mathbf{h}_t^1} - \sum_{s=0}^{t-2}\theta'_{\delta W_s^3}\theta_{\delta h_t^1}\frac{1}{n}Q_t^3\left(n^{d_3}\theta_{W_s^4}\eta\mathrm{d}\mathbf{h}_s^{3,k}\boldsymbol{\chi}_s(\mathbf{h}_s^1)^T\right)\delta\mathbf{h}_t^1 \tag{256}$$

$$- \theta'_{\delta W_t^3}\frac{1}{n}Q_t^3\left(n^{d_3}\theta_{W_{t-1}^4}\eta\mathrm{d}\mathbf{h}_{t-1}^{3,k}\boldsymbol{\chi}_{t-1}(\mathbf{h}_{t-1}^1)^T\right)(\mathbf{h}_t^1), \tag{257}$$

$$(\delta\mathbf{h}_t^{3,k})_\alpha = \frac{1}{n}\sum_{\gamma=1}^{n}\phi\Big(\Upsilon_\alpha, \{\{(\mathrm{d}\mathbf{h}_s^{3,k})_{\alpha l}\}_{l=1}^{d_{\text{out}}}\}_{s=0}^{t-2}, \{(\mathrm{d}\mathbf{h}_{t-1}^{3,k})_{\alpha l}\}_{l=1}^{d_{\text{out}}};$$

$$\{(\mathbf{h}_s^1)_\gamma\}_{s=0}^{t-2}, (\delta\mathbf{h}_t^1)_\gamma, (\mathbf{h}_{t-1}^1)_\gamma, (\mathbf{h}_t^1)_\gamma; \qquad \texttt{(OuterNonlin)}$$

$$\theta_{\delta h_t^1/\delta h_t^{3,k}}, \{\theta'_{\delta W_s^3\delta h_t^1/\delta h_t^{3,k}}\}_{s=0}^{t-2}, \theta'_{\delta W_t^3/\delta h_t^{3,k}}, n^{d_3}, \{\theta_{W_s^4}\}_{s=0}^{t-1}\Big) \tag{258}$$

$$\Upsilon := \mathbf{W}_0^{3,k}\delta\mathbf{h}_1^1, \qquad \texttt{(MatMul)} \tag{259}$$

$$\theta_{\delta h_t^{3,k}} = \max\left(\theta_{\delta h_t^1}, \max_s\left(\theta'_{\delta W_s^3}\theta_{\delta h_t^1}\right), \theta'_{\delta W_t^3}\right), \tag{260}$$

$$\theta_{\delta h_t^1/\delta h_t^{3,k}} := \theta_{\delta h_t^1}/\theta_{\delta h_t^{3,k}}, \tag{261}$$

$$\theta'_{\delta W_s^3\delta h_t^1/\delta h_t^{3,k}} := \theta'_{\delta W_s^3}\theta_{\delta h_t^1}/\theta_{\delta h_t^{3,k}}, \tag{262}$$

$$\theta'_{\delta W_t^3/\delta h_t^{3,k}} := \theta'_{\delta W_t^3}/\theta_{\delta h_t^{3,k}}. \tag{263}$$

We repeat the above process for each expert k.

$$\mathbf{h}_t^3 - \mathbf{h}_{t-1}^3 = \sum_{k=1}^{m}\left(\tilde{h}_{t-1,k}^2(\mathbf{h}_t^{3,k} - \mathbf{h}_{t-1}^{3,k}) + (\tilde{h}_{t,k}^2 - \tilde{h}_{t-1,k}^2)\mathbf{h}_t^{3,k}\right) \tag{264}$$

$$= \sum_{k=1}^{m}\left(\theta_{\delta h_t^{3,k}}\tilde{h}_{t-1,k}^2\delta\mathbf{h}_t^{3,k} + \theta_{\delta\tilde{h}_t^2}\delta\tilde{h}_{t,k}^2\mathbf{h}_t^{3,k}\right), \tag{265}$$

$$(\delta\mathbf{h}_t^3)_\alpha = \phi\left(\{(\delta\mathbf{h}_t^{3,k})_\alpha\}_{k=1}^m, \{(\mathbf{h}_t^{3,k})_\alpha\}_{k=1}^m; \theta_{\delta h_t^{3,k}/\delta h_t^3}, \theta_{\delta\tilde{h}_t^2/\delta h_t^3}, \{\tilde{h}_{t-1,k}^2\}_{k=1}^m, \{\delta\tilde{h}_{t,k}^2\}_{k=1}^m\right), \quad \texttt{(Nonlin)} \tag{266}$$

$$\phi(\ldots) := \sum_{k=1}^{m}\left(\theta_{\delta h_t^{3,k}}\tilde{h}_{t-1,k}^2(\delta\mathbf{h}_t^{3,k})_\alpha + \theta_{\delta\tilde{h}_t^2}\delta\tilde{h}_{t,k}^2(\mathbf{h}_t^{3,k})_\alpha\right), \tag{267}$$

$$\theta_{\delta h_t^3} := \max(\theta_{\delta h_t^{3,k}}, \theta_{\delta\tilde{h}_t^2}), \tag{268}$$

$$\theta_{\delta h_t^{3,k}/\delta h_t^3} := \theta_{\delta h_t^{3,k}}/\theta_{\delta h_t^3}, \tag{269}$$

$$\theta_{\delta\tilde{h}_t^2/\delta h_t^3} := \theta_{\delta\tilde{h}_t^2}/\theta_{\delta h_t^3}. \tag{270}$$

$$\mathbf{f}_t - \mathbf{f}_{t-1} = \mathbf{W}^4_{t-1}(\mathbf{h}^3_t - \mathbf{h}^3_{t-1}) + (\mathbf{W}^4_t - \mathbf{W}^4_{t-1})\mathbf{h}^3_t \tag{271}$$

$$= \theta'_{W^4_{t-1}} \theta_{\delta h^3_t} \frac{\overline{\mathbf{W}^4_{t-1}} \delta \mathbf{h}^3_t}{n} - \theta'_{\delta W^4_t} \eta \frac{1}{n} Q^4_t \left(n^{d_4} \boldsymbol{\chi}_{t-1} (\mathbf{h}^3_{t-1})^T \right) \mathbf{h}^3_t, \tag{272}$$

$$\theta_{f_t}(\delta \mathbf{f}_{t-1})_\beta = \frac{1}{n} \sum_{\alpha=1}^{n} \phi \left(\overline{\mathbf{W}^4_{t-1}}_{\beta\alpha}, (\delta \mathbf{h}^3_t)_\alpha, (\mathbf{h}^3_{t-1})_\alpha, (\mathbf{h}^3_t)_\alpha; \theta'_{W^4_{t-1}}, \theta_{\delta h^3_t}, \theta'_{\delta W^4_t}, n^{d_4}, \eta, (\boldsymbol{\chi}_{t-1})_\beta \right), \quad \texttt{(Moment)} \tag{273}$$

$$\phi(\dots) := \theta'_{W^4_{t-1}} \theta_{\delta h^3_t} (\overline{\mathbf{W}^4_{t-1}})_{\beta\alpha} (\delta \mathbf{h}^3_t)_\alpha - \theta'_{\delta W^4_t} \eta Q^4_t \left(n^{d_4} (\boldsymbol{\chi}_{t-1})_\beta (\mathbf{h}^3_{t-1})_\alpha \right) (\mathbf{h}^3_t)_\alpha, \tag{274}$$

$$\theta_{\delta f_t} := \max(\theta'_{W^4_{t-1}} \theta_{\delta h^4_t}, \theta'_{\delta W^4_t}). \tag{275}$$

C.4. Infinite Width Limit for Soft MoE with Adam

See Appendix C.2 for an explanation of the limit notation.

Adam requires defining the limit of an extra operation, `OuterNonlin` (Yang and Littwin, 2023):

Z OuterNonlin: Given $r \geq 0$ and function $\psi : \mathbb{R}^{(r+1)+\ell} \to \mathbb{R}$ such that $y_\alpha = \frac{1}{n^r} \sum_{\beta_1,\dots,\beta_r=1}^{n} \psi(x^0_\alpha; x^1_{\beta_1}; \dots; x^r_{\beta_r}; \theta_1$ then

$$Z^y := \mathbb{E}_{Z^{x^1}_1, \dots Z^{x^r}_1} \left[\psi \left(Z^{x^0}; Z^{x^1}_1; \dots; Z^{x^r}_1; \mathring{\theta}_1, \dots, \mathring{\theta}_l \right) \right], \tag{276}$$

where $(Z^{x^1}_1, \dots Z^{x^r}_1) \overset{d}{=} (Z^{x^1}, \dots Z^{x^r})$ is an iid copy of $(Z^{x^1}, \dots Z^{x^r})$ and is independent of Z^{x^0}.

Assumption 14 (Pseudo Lipschitz Q, Assumption 2.3.2 (Yang and Littwin, 2023)) *Assume that Q^ℓ_t, ϵ are pseudo-Lipschitz for all ℓ, t.*

Assumption 15 (Faithfulness at initialisation) *If we want faithfulness in the first weight updates, it already restricts the values of d to $d_1 = 1, d_2 = 0, d_3 = 1, d_4 = 0$.*

First Forward Pass:

$$Z^{h^1_0} = \sum_{j=1}^{d_{\text{in}}} \xi_j Z^{W^1_{0,j}}, \tag{277}$$

$$\mathring{\mathbf{h}}^2_0 = \mathbf{b}^2_0, \tag{278}$$

$$\mathring{\tilde{\mathbf{h}}}^2_0 = G(\mathbf{b}^2_0) \tag{279}$$

$$Z^{h^{3,k}_0} = Z^{W^{3,k}_0 h^1_0} \text{ for every expert } k, \tag{280}$$

$$Z^{h^3_0} = \sum_{k=1}^{m} \mathring{\tilde{h}}^2_{0,k} Z^{h^{3,k}_0} \tag{281}$$

$$\mathring{\mathbf{f}}_0 = \mathbf{0}. \tag{282}$$

Here, $Z^{W^1_0}$ is standard Gaussian by construction. $\mathring{\mathbf{h}}^2_0 = \mathbf{0}$ as $\mathring{h}^2_0 = \mathbb{E}[Z^{W^2_0}]\mathbb{E}[Z^{h^1_0}] = 0$. $\mathring{\tilde{\mathbf{h}}}^2_0$ then follows by the definition of Softmax($\mathbf{0}$). $Z^{W^{3,k}_0 h^1_0} = \hat{Z}^{W^{3,k}_0 h^1_0}$ from the Z MatMul rule. Finally $\mathring{\mathbf{f}}_0 = \mathbf{0}$ as $\mathring{f}_0 = \mathbb{E}[Z^{W^4_0}]\mathbb{E}[Z^{h^3_0}] = 0$.

First Backward Pass:

$$Z^{\mathrm{d}h_0^3} = Z^{\overline{W_0^4}}, \tag{283}$$

$$\mathring{\mathrm{d}\tilde{h}}^2_{0,k} = 0 \text{ for every expert } k, \tag{284}$$

$$\mathring{\mathrm{d}h}^2{}_{0,k} = 0 \text{ for every expert } k, \tag{285}$$

$$Z^{\mathrm{d}h_0^{3,k}} = \mathring{\tilde{\mathbf{h}}}^2_{0,k} Z^{\mathrm{d}h_0^3} \text{ for every expert } k, \tag{286}$$

$$Z^{\mathrm{d}h_0^1} = \sum_{k=1}^{m} Z^{(W_0^{3,k})^T \mathrm{d}h_0^{3,k}} \tag{287}$$

$$\tag{288}$$

$Z^{\overline{W_0^4}}$ is a standard Gaussian by construction. $\mathring{\mathrm{d}\tilde{h}}^2_{0,k} = \mathbb{E}[Z^{h_0^{3,k}}]\mathbb{E}[\overline{W_0^4}] = 0$. From this, $\mathring{\mathrm{d}h}^2{}_{0,k} = 0$ follows. $Z^{(W_0^{3,k})^T \mathrm{d}h_0^3} = \hat{Z}^{(W_0^{3,k})^T \mathrm{d}h_0^3}$ from the Z MatMul rule.

First Feature Update:

$$Z^{\delta h_1^1} = -\eta Q_1^1 \left(\sum_{\beta=1}^{d_{\text{out}}} Z^{(\mathrm{d}h_0^1)_\beta} (\mathring{\boldsymbol{\chi}}_0)_\beta \boldsymbol{\xi}^T \right) \boldsymbol{\xi}, \tag{289}$$

$$\delta \mathring{h}^2_{1,k} = \mathring{\theta}_{\delta h_1^1} \mathbb{E}[Z^{(\overline{W_0^2})_{k,:}} Z^{\delta h_1^1}] - \mathring{\theta}'_{\delta W_1^2} \eta \mathbb{E}\left[Q_1^2 \left(\sum_{\beta=1}^{d_{\text{out}}} (\mathrm{d}\mathring{h}^2_{0,k})_\beta (\mathring{\boldsymbol{\chi}}_0)_\beta Z^{h_0^1} \right) Z^{h_1^1} \right], \tag{290}$$

$$\delta \mathring{\tilde{\mathbf{h}}}^2_1 = \begin{cases} \mathrm{G}(\mathring{\mathbf{h}}^2_1) - \mathrm{G}(\mathring{\mathbf{h}}^2_0) \text{ if } \mathring{\theta}_{\delta h_1^2} = 1 \\ \mathrm{G}'(\mathring{\mathbf{h}}^2_0)(\delta \mathring{\mathbf{h}}^2_0) \text{ if } \mathring{\theta}_{\delta h_1^2} = 0 \end{cases}, \tag{291}$$

$$\begin{aligned} Z^{\delta h_1^{3,k}} &= \mathring{\theta}_{\delta h_1^1 / \delta h_1^{3,k}} Z^{W_0^{3,k} \delta h_1^1} \\ &\quad - \mathring{\theta}'_{\delta W_1^3 / \delta h_1^{3,k}} \eta \mathbb{E}_{\left(Z_1^{h_0^1}, Z_1^{h_1^1}\right)} \left[Q_1^3 \left(\sum_{\beta=1}^{d_{\text{out}}} Z^{(\mathrm{d}h^{3,k})_\beta} (\mathring{\boldsymbol{\chi}}_0)_\beta Z_1^{h_0^1} \right) Z_1^{h_1^1} \right], \text{ for every expert } k, \end{aligned} \tag{292}$$

$$Z^{\delta h_1^3} = \sum_{k=1}^{m} \mathring{\theta}_{\delta h_1^{3,k} / \delta h_1^3} \mathring{\tilde{h}}^2_{0,k} Z^{\delta h_1^{3,k}} - \mathring{\theta}_{\delta \tilde{h}_1^2} \mathring{\theta}_{h_1^{3,k} / \delta h^3} \delta \mathring{\tilde{h}}^2_{1,k} Z^{h_1^{3,k}}, \tag{293}$$

$$\delta \mathring{f}_1 = \mathring{\theta}_{\delta h_1^3} \mathbb{E}[Z^{\overline{W_0^4}} Z^{\delta h_1^3}] - \mathring{\theta}'_{\delta W_1^4} \eta \mathbb{E}\left[Q_1^4 \left(\mathring{\chi}_0 Z^{h_0^3} \right) \right] Z^{h_1^3}, \tag{294}$$

where $Z^{W_0^{3,k} \delta h_1^1}$ is given by the Z MatMul rule,

$$Z^{W_0^{3,k} \delta h_1^1} = \hat{Z}^{W_0^{3,k} \delta h_1^1} + \dot{Z}^{W_0^{3,k} \delta h_1^1} \tag{295}$$

$$= \hat{Z}^{W_0^{3,k} \delta h_1^1} + Z^{\mathrm{d}h_0^{3,k}} \mathbb{E}\left[\frac{\partial Z^{\delta h_1^1}}{\partial \hat{Z}^{(W_0^{3,k})^T \mathrm{d}h_0^{3,k}}} \right]. \tag{296}$$

Furthermore, the random variables $(Z_1^{h_0^1}, Z_1^{h_1^1}) \overset{d}{=} (Z^{h_0^1}, Z^{h_1^1})$ is an iid copy and is independent from $Z^{\mathrm{d}h^{3,k}}$.

We write the limit of each vector in the program as

$$Z^{h_t} = Z^{h_0} + \mathring{\theta}_{\delta h_1} Z^{\delta h_1} + \ldots + \mathring{\theta}_{\delta h_t} Z^{\delta h_t}. \tag{297}$$

Weight updates: The distribution of certain weight updates are required in the backward pass and thus the forward passes.

$$Z^{\overline{\delta W_t^4}} = -Q_t^4 \left(\eta \mathring{\chi}_{t-1} Z^{h_{t-1}^3} \right), \tag{298}$$

$$Z^{(\overline{\delta W_t^2})_{k,:}} = -Q_t^2 \left(\eta \sum_{\beta=1}^{d_{\text{out}}} (\mathrm{d}\mathring{h}_{t,k}^2)_\beta (\mathring{\boldsymbol{\chi}}_{t-1})_\beta Z^{h_{t-1}^1} \right). \tag{299}$$

tth Backward Pass:

$$Z^{\mathrm{d}h_t^3} = Z^{(\overline{W_t^4})^T}, \tag{300}$$

$$\mathrm{d}\mathring{\tilde{h}}_{t,k}^2 = \mathbb{E}[Z^{h_t^{3,k}} Z^{\mathrm{d}h_t^3}], \tag{301}$$

$$\mathrm{d}\mathring{h}_{t,k}^2 = \sum_{j=1}^m J(\mathring{h}_t^2)_{kj} \mathrm{d}\mathring{\tilde{h}}_{t,k}^2, \tag{302}$$

$$Z^{\mathrm{d}h_t^{3,k}} = \mathring{\tilde{h}}_{t,k}^2 Z^{\mathrm{d}h_t^3}, \tag{303}$$

$$Z^{\mathrm{d}h_t^1} = \sum_{k=1}^m Z^{(\overline{W_t^2})_{k,:}} \mathrm{d}\mathring{h}_{t,k}^2 + Z^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}} - \eta \sum_{s=1}^{t-1} \mathbb{E}_{\left(Z_1^{\mathrm{d}h_s^{3,k}}, Z_1^{\mathrm{d}h_t^{3,k}}\right)} \left[Q_s^1 \left(Z^{h_s^1} \mathring{\chi}_s Z_1^{\mathrm{d}h_s^{3,k}} \right) Z_1^{\mathrm{d}h_t^{3,k}} \right], \tag{304}$$

where $Z^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}}$ is given by the Z MatMul rule,

$$Z^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}} = \hat{Z}^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}} + \dot{Z}^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}} \tag{305}$$

$$= \hat{Z}^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}} + \sum_{v \in \mathcal{V}: W_0^{3,k} v \in \mathcal{V}} Z^v \mathbb{E}\left[\frac{\partial Z^{\mathrm{d}h_t^{3,k}}}{\partial \hat{Z}^{W_0^{3,k} v}} \right], \tag{306}$$

where $\mathcal{V} := \{v \text{ for some vector in the program } v\}$ is the set of all vectors in the program. Further the random variables $(Z_1^{\mathrm{d}h_s^{3,k}}, Z_1^{\mathrm{d}h_t^{3,k}}) \overset{d}{=} (Z^{\mathrm{d}h_s^{3,k}}, Z^{\mathrm{d}h_t^{3,k}})$ is an iid copy and is independent of $Z^{h_s^1}$.

tth Forward Pass:

$$Z^{\delta h_t^1} = -\eta Q_t^1 \left(\sum_{\beta=1}^{d_{\text{out}}} Z^{(\mathrm{d}h_{t-1}^1)_\beta} (\mathring{\boldsymbol{\chi}}_{t-1})_\beta \boldsymbol{\xi}^T \right) \boldsymbol{\xi}, \tag{307}$$

$$\delta \mathring{h}_{t,k}^2 = \mathring{\theta}'_{W_{t-1}^2} \mathring{\theta}_{\delta h_t^1} \mathbb{E}[Z^{\overline{(W_{t-1}^2)_{k,:}}} Z^{\delta h_t^1}] - \mathring{\theta}'_{\delta W_t^2} \eta \mathbb{E}\left[Q_t^2 \left(\sum_{\beta=1}^{d_{\text{out}}} (\mathrm{d}\mathring{h}_{t-1,k}^2)_\beta (\mathring{\boldsymbol{\chi}}_{t-1})_\beta Z^{h_{t-1}^1} \right) Z^{h_t^1} \right], \tag{308}$$

$$\delta \mathring{\tilde{\mathbf{h}}}_t^2 = \begin{cases} \mathrm{G}(\mathring{\mathbf{h}}_t^2) - \mathrm{G}(\mathring{\mathbf{h}}_{t-1}^2) \text{ if } \mathring{\theta}_{\delta h_t^2} = 1 \\ \mathrm{G}'(\mathring{\mathbf{h}}_{t-1}^2)(\delta \mathring{\mathbf{h}}_{t-1}^2) \text{ if } \mathring{\theta}_{\delta h_t^2} = 0 \end{cases}, \tag{309}$$

$$Z^{\delta h_t^{3,k}} = \mathring{\theta}_{\delta h_t^1 / \delta h_t^{3,k}} Z^{W_0^{3,k} \delta h_t^1} \tag{310}$$

$$- \sum_{s=0}^{t-2} \mathring{\theta}'_{\delta W_s^{3,k} \delta h_t^1 / \delta h_t^{3,k}} \eta \mathbb{E}_{\left(Z_1^{h_s^1}, Z_1^{\delta h_t^1}\right)} \left[Q_t^3 \left(\sum_{\beta}^{d_{\text{out}}} Z^{(\mathrm{d}h_s^{3,k})_\beta} (\mathring{\boldsymbol{\chi}}_s)_\beta Z_1^{h_s^1} \right) Z_1^{\delta h_t^1} \right] \tag{311}$$

$$- \mathring{\theta}'_{\delta W_t^{3,k} / \delta h_t^{3,k}} \eta \mathbb{E}_{\left(Z_1^{h_{t-1}^1}, Z_1^{h_t^1}\right)} \left[Q_t^3 \left(\sum_{\beta}^{d_{\text{out}}} Z^{(\mathrm{d}h_{t-1}^{3,k})_\beta} (\mathring{\boldsymbol{\chi}}_{t-1})_\beta Z_1^{h_{t-1}^1} \right) Z_1^{h_t^1} \right], \tag{312}$$

$$Z^{\delta h_t^3} = \sum_{k=1}^{m} \mathring{\theta}_{\delta h_t^{3,k} / \delta h_t^3} \mathring{\tilde{h}}_{k,t-1}^2 Z^{\delta h_t^{3,k}} - \mathring{\theta}_{\delta \tilde{h}_t^2 / \delta h_t^3} \delta \mathring{\tilde{h}}_{t,k}^2 Z^{h_t^{3,k}}, \tag{313}$$

$$\delta \mathring{f}_t = \mathring{\theta}'_{W_{t-1}^4} \mathring{\theta}_{\delta h_t^3} \mathbb{E}[Z^{\overline{W_{t-1}^4}} Z^{\delta h_t^3}] - \mathring{\theta}'_{\delta W_t^4} \eta \mathbb{E}\left[Q_t^4 \left(\mathring{\chi}_{t-1} Z^{h_{t-1}^3} \right) Z^{h_t^3} \right], \tag{314}$$

where

$$Z^{W_0^{3,k} \delta h_t^1} = \hat{Z}^{W_0^{3,k} \delta h_t^1} + \dot{Z}^{W_0^{3,k} \delta h_t^1} \tag{315}$$

$$= \hat{Z}^{W_0^{3,k} \delta h_t^1} + \sum_{s=0}^{t-1} Z^{\mathrm{d}h_s^1} \mathbb{E}\left[\frac{\partial Z^{\mathrm{d}h_s^{3,k}}}{\partial \hat{Z}^{(W_0^{3,k})^T \mathrm{d}h_s^{3,k}}} \right] \tag{316}$$

Appendix D. Derivation of Soft MoE Results

We derive the results from Section 3 here. The results rely on the Tensor Program derivation in Appendix C.

D.1. SP: Router Collapse

In this section, we derive the results for the soft MoE network with SP parametrisation. We use the Tensor Program derivations in Appendix C.1 and Appendix C.3 to do this.

Recall that the SP parametrisation refers to initialisation parameters $b_1 = 0$ and $b_\ell = 0.5$ for all layers $\ell \neq 1$. Furthermore, the learning rate and epsilon scaling parameters are the same for each layer $c = c_\ell$ and $d = d_\ell$ for all layers ℓ. The initial values of b_ℓ set the scalars

for the network in Theorem 1

$$\theta_{W_0^1} = 1, \tag{317}$$
$$\theta_{W_0^2} = n^{-0.5}, \tag{318}$$
$$\theta_{W_0^{3,k}} = 1, \tag{319}$$
$$\theta_{W_0^4} = n^{-0.5}. \tag{320}$$
$$\tag{321}$$

We write the Tensor Program for the first forward pass with SP as follows (the rest of the tensor program is similar to Appendix C.1). We write thie without the bias as it does not affect the result.

First Forward Pass:

$$\mathbf{h}_0^1 = \overline{\mathbf{W}_0^1}\boldsymbol{\xi}, \qquad (\mathbf{h}_0^1)_\alpha = \phi\left((\overline{\mathbf{W}_0^1})_{\alpha 1}, \ldots, (\overline{\mathbf{W}_0^1})_{\alpha d_{\text{in}}}; \xi_1, \ldots, \xi_{d_{\text{in}}}\right) = \sum_{j=1}^{d_{\text{in}}} (\overline{\mathbf{W}_0^1})_{\alpha j}\xi_j, \qquad \texttt{(Nonlin)}$$

$$\mathbf{h}_0^2 = \sqrt{n}\overline{\mathbf{W}_0^2}\mathbf{h}_0^1, \qquad (\mathbf{h}_0^2)_k = \frac{1}{n}\sum_{\alpha=1}^{n} \phi\left((\overline{\mathbf{W}_0^2})_{k\alpha}, (\mathbf{h}_0^1)_\alpha; \sqrt{n}\right) = \frac{1}{n}\sqrt{n}\sum_{\alpha=1}^{n} (\overline{\mathbf{W}_0^2})_{k\alpha}(\mathbf{h}_0^1)_\alpha, \qquad \texttt{(Moment)}$$

$$\tilde{\mathbf{h}}_0^2 = \mathrm{G}(\mathbf{h}_0^2), \qquad \texttt{(Moment, See G)}$$

$$\hat{\mathbf{h}}_0^{3,k} = \overline{\mathbf{W}^{3,k}}\mathbf{h}_0^1, \qquad \texttt{(MatMul)}$$

$$\mathbf{h}_0^{3,k} = \hat{\mathbf{h}}_0^{3,k}, \qquad (\mathbf{h}_0^{3,k})_\alpha = \phi\left((\hat{\mathbf{h}}_0^{3,k})_\alpha; -\right) = (\hat{\mathbf{h}}_0^3)_\alpha \qquad \texttt{(Nonlin)}$$

$$\mathbf{h}_0^3 = \sum_{k=1}^{m} \mathbf{h}_0^{3,k}\tilde{h}_{0,k}^2, \qquad (\mathbf{h}_0^3)_\alpha = \phi\left((\mathbf{h}_0^{3,1})_\alpha, \ldots, (\mathbf{h}_0^{3,m})_\alpha; \tilde{h}_{0,1}^2, \ldots, \tilde{h}_{0,m}^2\right) = \sum_{k=1}^{m} (\mathbf{h}_0^{3,k})_\alpha \tilde{h}_{0,k}^2 \qquad \texttt{(Nonlin)}$$

$$\mathbf{f}_0 = \sqrt{n}\frac{1}{n}\overline{\mathbf{W}_0^4}\mathbf{h}_0^3, \qquad (\mathbf{f}_0)_\beta = \frac{1}{n}\sum_{\alpha=1}^{n} \phi\left((\overline{\mathbf{W}_0^4})_{\beta\alpha}, (\mathbf{h}_0^3)_\alpha; \sqrt{n}\right) = \frac{1}{n}\sqrt{n}\sum_{\alpha=1}^{n} (\overline{\mathbf{W}_0^4})_{\beta\alpha}(\mathbf{h}_0^3)_\alpha., \qquad \texttt{(Moment)}$$

where $G(\cdot)$ is the gating function, either Softmax or Sigmoid..

To find the scale of the output-like layers $\mathbf{h}^2$ and $\mathbf{f}$ we use Master's theorem on the variance of the coordinates

$$\mathbb{E}\left[(\mathbf{h}_0^2)_k^2\right] = \sum_{\alpha=1}^{n} \frac{1}{n}\mathbb{E}[(\overline{\mathbf{W}_0^2})_{k\alpha}^2]\mathbb{E}[(\mathbf{h}_0^1)_\alpha^2] = \mathbb{E}\left[\frac{1}{n}\sum_{\alpha=1}^{n}(\mathbf{h}_0^1)_\alpha^2\right], \qquad \texttt{(Moment)} \tag{322}$$

$$\mathbb{E}\left[(\mathbf{f}_0)_\beta^2\right] = \sum_{\alpha=1}^{n} \frac{1}{n}\mathbb{E}[(\overline{\mathbf{W}_0^4})_{\beta\alpha}^2]\mathbb{E}[(\mathbf{h}_0^3)_\alpha^2] = \mathbb{E}\left[\frac{1}{n}\sum_{\alpha=1}^{n}(\mathbf{h}_0^3)_\alpha^2\right]. \qquad \texttt{(Moment)} \tag{323}$$

We consider SGD updates here for ease of exposition. The result trivially holds for Adam updates as well (see Theorem 18).

Using the Master's theorem (Yang and Hu, 2021), the coordinates of each layer will have scale $\Theta(1)$,

$$\theta_{h_0^1} = 1, \tag{324}$$

$$\theta_{h_0^2} = 1, \tag{325}$$

$$\theta_{\tilde{h}_0^2} = 1, \tag{326}$$

$$\theta_{h_0^{3,k}} = 1, \tag{327}$$

$$\theta_{h_0^3} = 1, \tag{328}$$

$$\theta_{f_0} = 1. \tag{329}$$

Without loss of generality, we assume that values of c is chosen such that $\delta\mathbf{h}_1^1$ and $\mathbf{h}_1^1 = \mathbf{h}_0^1 + \delta\mathbf{h}_1^1$ are $\Theta(1)$. That is $\theta_{\delta h_1^1} = \theta_{h_1^1} = 1$. It is safe to make this assumption as $\delta\mathbf{h}_1^1$ exploding or vanishing with width are both undesirable. The update to $\mathbf{h}^2$ then has the form (Appendix C.1)

$$\delta\mathbf{h}_1^2 = \theta'_{W_0^2}\frac{\overline{\mathbf{W}_0^2}\delta\mathbf{h}_1^1}{n} - \theta'_{\delta W_1^2}\eta \mathrm{d}\mathbf{h}_0^2\boldsymbol{\chi}_0\frac{(\mathbf{h}_0^1)^T\mathbf{h}_1^1}{n}, \tag{330}$$

where $\overline{\mathbf{W}_0^2}, \delta\mathbf{h}_1^1$ are the unscaled ($\Theta(1)$) terms. The above operation is written as a Moment in the Tensor Program. By the TP Master Theorem, $\frac{\overline{\mathbf{W}_0^2}\delta\mathbf{h}_1^1}{n} \to \mathbb{E}[Z^{\overline{W_0^2}}Z^{\delta h_1^1}]$ (Appendix C.1). As the gradients required to calculate δh_1^1 depend on $\overline{W_0^2}$, the limiting random variables $Z^{\overline{W_0^2}}, Z^{\delta h_1^1}$ are correlated, thus the expectation is non-zero and the term scales $\Theta(1)$. Thus, the first term has the scale $\theta'_{W_0^2} = n^{0.5}$, and the second scales $\theta'_{\delta W_0^2} = n^{1-c} = n^{1-c}$. Hence, the first term scales $\Theta(n^{0.5})$ irrespective of the scaling of the second term, which means that the entire term scales $\Omega(n^{0.5})$. As $\mathbf{h}_1^2 = \delta\mathbf{h}_1^2 + \mathbf{h}_0^2$, where the first term scales $\Omega(n^{0.5})$, and the second term scales $\Theta(1)$, hence $\mathbf{h}_1^2$ scales $\Omega(n^{0.5})$.

Saturation of Softmax $\tilde{\mathbf{h}}^2$: In the softmax network, $\tilde{\mathbf{h}}_1^2 = \mathrm{Softmax}(\mathbf{h}_1^2)$ where $\mathbf{h}_1^2$ scales $\Omega(n^{0.5})$. Using the well known property that softmax of diverging inputs results in a one-hot distribution, we get,

$$\lim_{n\to\infty}\frac{e^{(h_t^2)_k}}{\sum_{j=1}^m e^{(h_t^2)_j}} \to 1,$$

for some expert index $k \in [m]$. The gradient of the softmax layer is

$$\frac{\partial(\tilde{\mathbf{h}}_1^2)_i}{\partial(\mathbf{h}_1^2)_j} = (\tilde{h}_1^2)_i(\delta_{ij} - (\tilde{h}_1^2)_j),$$

where $(\tilde{\mathbf{h}}_1^2)_i$ is the i^{th} component of the vector $\tilde{\mathbf{h}}_1^2$, and δ_{ij} is the Kronecker delta function. If $\tilde{\mathbf{h}}_1^2$ is one-hot, then the gradient $\frac{\partial\tilde{\mathbf{h}}_1^2}{\partial\mathbf{h}_1^2}$ is a zero matrix of size $m \times m$. As $\frac{\partial\tilde{\mathbf{h}}_1^2}{\partial\mathbf{h}_1^2}$ is required to calculate the gradient of the router $\frac{\partial\mathcal{L}}{\partial\mathbf{W}^2}$,

$$\frac{\partial\mathcal{L}}{\partial\mathbf{W}_1^2} = \frac{\partial\mathcal{L}}{\partial\tilde{\mathbf{h}}_1^2}\frac{\partial\tilde{\mathbf{h}}_1^2}{\partial\mathbf{h}_1^2}\frac{\partial\mathbf{h}_1^2}{\partial\mathbf{W}_1^2}$$

this implies that the router gradient will approach a matrix of zeroes as the width is increased. Hence $\mathbf{W}_t^2 = \mathbf{W}_1^2$ for all subsequent steps of training.

Saturation of Sigmoid $\tilde{\mathbf{h}}^2$: In the sigmoid network, $\tilde{\mathbf{h}}_1^2 = \sigma(\mathbf{h}_1^2)$ where $\mathbf{h}_1^2$ scales $\Omega(n^{0.5})$. With n increasing we get

$$\lim_{n\to\infty} \sigma((h_t^2)_k) \to \begin{cases} 1 & \text{if } (h_t^2)_k > 0 \\ 0 & \text{if } (h_t^2)_k < 0 \end{cases},$$

for every expert $k \in [m]$. The gradient of the sigmoid layer is

$$\frac{\partial(\tilde{\mathbf{h}}_1^2)_i}{\partial(\mathbf{h}_1^2)_i} = \sigma\left((\mathbf{h}_1^2)_i\right)\left(1 - \sigma((\mathbf{h}_1^2)_i)\right),$$

which approaches a zero matrix as n increases. Similarly to the softmax case, this results in a vanishing gradient for the router.

Remark 16 (Zero initialisation of router with SGD) *Zero intitialisation of the router with SGD ensures that the interaction between $\mathbf{W}_0^2$ and $\delta\mathbf{h}_1^1$ does not occur. However, here the results from Yang and Hu (2021) hold, primarily that it is not possible for the "output-like" layers (router and function) to feature learn. This can easily be seen by the fact that $\theta_{\delta h_1^1} = n^{-(c+1)}$, which means that $c = -1$ is required for $\mathbf{h}_1^1$ to feature learn. This however causes the second term (effective updates) in $\delta\mathbf{h}_1^2$ to diverge as $\theta'_{\delta W_1^2} = n \cdot n^{-c} \cdot n \cdot \theta_{W_0^4} = n^3$.*

Remark 17 (Results hold for Top-K) *The router logits diverge due to the interaction between $\mathbf{W}_0^2$ and $\delta\mathbf{h}_1^1$, and thus holds irrespective of the gating function, or sparse execution. If Softmax or Sigmoid activations are used with Top-K, the router probabilities will still saturate with SP.*

Remark 18 *(Router diverges with Adam) The analysis above relies on the propagating updates $\mathbf{W}_0^2\delta\mathbf{h}_1^1$ of the router blowing up. Adam updates only affect the effective updates of the router $\delta\mathbf{W}_1^2\mathbf{h}_1^2$. Hence, the same analysis holds for Adam.*

D.2. μP-MoE: Stability but with Router Stagnation without Router Bias

Table 1: *bcd* parametrisation vlaue for μP-MoE SGD

Layer (ℓ)	$\mathbf{b}_\ell$	$\mathbf{c}_\ell$
1	0	-1
2	1	1
3	0.5	0
4	1	1

Stability and Feature Learning for SGD: We show that the values in Table 1 for the network in Theorem 1 satisfy our desired properties of stability and feature learning for SGD. The values in Theorem 11 already satisfy stability. To show feature learning, we show that

the limiting values of the scalars $\mathring{\theta}_\bullet$ in the feature updates converge to one. Stability ($\mathcal{O}(1)$ features) combined with $\Theta(1)$ feature updates then imply that all the features scale $\Theta(1)$.

$$\mathring{\theta}_{\delta h_1^1} = 1, \qquad \mathring{\theta}'_{\delta W_1^2} = 1, \tag{331}$$

$$\mathring{\theta}_{\delta h_1^2} = 1, \tag{332}$$

$$\mathring{\theta}_{\delta h_1^1/\delta h_1^{3,k}} = 1, \qquad \mathring{\theta}'_{\delta W_1^3/\delta h_1^{3,k}} = 1, \tag{333}$$

$$\mathring{\theta}_{\delta h_1^{3,k}/\delta h_1^3} = 1, \qquad \mathring{\theta}_{\delta \tilde{h}_1^2/\delta h_1^3} = 1, \tag{334}$$

$$\mathring{\theta}_{\delta h_1^3} = 1, \qquad \mathring{\theta}'_{\delta W_1^4} = 1. \tag{335}$$

For the next forward passes we have $\theta_{W_t^2} = \theta_{W_0^2}$ and $\theta_{W_t^4} = \theta_{W_0^4}$, thus we have

$$\mathring{\theta}'_{W_{t-1}^2}\mathring{\theta}_{\delta h_t^1} = 1, \qquad \mathring{\theta}'_{\delta W_t^2} = 1, \tag{336}$$

$$\mathring{\theta}_{\delta h_t^2} = 1, \tag{337}$$

$$\mathring{\theta}_{\delta h_t^1/\delta h_t^{3,k}} = 1, \qquad \mathring{\theta}'_{\delta W_s^3 \delta h_t^1/\delta h_t^{3,k}} = 1 \qquad \mathring{\theta}'_{\delta W_t^3/\delta h_t^{3,k}} = 1, \tag{338}$$

$$\mathring{\theta}_{\delta h_t^{3,k}/\delta h_t^3} = 1, \qquad \mathring{\theta}_{\delta \tilde{h}_t^2/\delta h_t^3} = 1, \tag{339}$$

$$\mathring{\theta}'_{W_{t-1}^4}\mathring{\theta}_{\delta h_t^3} = 1, \qquad \mathring{\theta}'_{\delta W_t^4} = 1. \tag{340}$$

For a formal arguement to show feature learning (such that the limits are not degenerate), we refer to (Yang and Hu, 2021, H.7).

Table 2: *bcd* parametrisation vlaue for μP-MoE Adam

Layer (ℓ)	$\mathbf{b}_\ell$	$\mathbf{c}_\ell$	$\mathbf{d}_\ell$
1	0	0	1
2	1	1	0
3	0.5	1	1
4	1	1	0

Stability, Feature Learning, and Faithfulness for Adam: The faithfulness follows from the values at the first weight updates Theorem 15. Similar to above, it is easy to see that all $\mathring{\theta}_\bullet$ converge to one.

Lack of specialisation: We now show that for μP-MoE without router bias, specialisation of the experts fails. We first show that, in the infinite-width limit, the gradient received by each router logit converges to the same deterministic value. Next, we show that this implies that the router logits, and hence the probabilities evolve synchronously, keeping the same value for each expert. In Softmax-Soft MoEs, due to the form of the softmax gradient, the above results imply that the gradient received by the router is zero. We finish with remarks that these theorems hold for Adam, and we find the finite-width scaling of the router gradient in Softmax-Soft MoEs.

Proposition 19 *Consider a soft MoE network without a router bias with m experts with either Softmax or Sigmoid gating parametrised such that stability holds throughout training (Theorem 9), and such that the router logits converge to zero in the infinite width limit, $\mathring{\mathbf{h}}_0^2 = 0$ (Theorem 11, Table 1). For any time step $t \geq 0$, each router activation $\tilde{\mathbf{h}}^2$ component receives the same deterministic unscaled ($\Theta(1)$) gradient. That is,*

$$\mathrm{d}\mathring{\tilde{h}}_{t,k}^2 := \mathbb{E}\left[Z^{h_t^{3,k}} Z^{\mathrm{d}h_t^3}\right], \tag{341}$$

is identical for all experts $k \in \{1, \ldots, m\}$.

Proof First, observe that $Z^{\mathrm{d}h_t^3} = Z^{\overline{W_t^4}}$ is a global variable shared by the entire network; it depends only on the output layer weights, making it independent of the specific expert index k. Therefore, it suffices to prove that the marginal distribution of the expert features $Z^{h_t^{3,k}}$ is identical for all k. We proceed by induction.

Base Case ($t = 0$): At initialization, the expert features are defined by the random variable:

$$Z^{h_0^{3,k}} = Z^{W_0^{3,k} h_0^1}. \tag{342}$$

The input features $\mathbf{h}_0^1$ are shared across all experts. The weights $\mathbf{W}_0^{3,k}$ are initialized i.i.d. such that $(\hat{W}_k^3)_{ij} \sim \mathcal{N}(0, n^{-1})$. Thus $Z^{W_0^{3,k} h_0^1}$ (derived by ZMatMul rule) is the same for every expert. Consequently, the random variables $Z^{h_0^{3,k}}$ are identically distributed for all k:

$$Z^{h_0^{3,k}} \overset{d}{=} Z^{h_0^{3,j}} \quad \forall k, j. \tag{343}$$

Thus, the expectation $\mathbb{E}[Z^{h_0^{3,k}} Z^{\mathrm{d}h_0^3}]$ is constant with respect to k , and thus $\mathrm{d}\mathring{\tilde{h}}_{0,k}^2$ is the same for each expert.

Note that for softmax or sigmoid gating, the router probabilities are the same for each expert at initialisation due to $\mathring{h}_{0,k}^2 = 0$ and without a router bias,

$$\tilde{h}_{0,k}^2 = \begin{cases} \frac{1}{m}\mathbf{1} & \text{if G = Softmax} \\ \frac{1}{2}\mathbf{1} & \text{if G = Sigmoid} \end{cases}.$$

Hence $\tilde{h}_{0,i}^2 = \tilde{h}_{0,j}^2$ for every $i, j \in [m]$.

Inductive Step: Assume that for all time steps $s < t$, the expert features are identically distributed across experts:

$$Z^{h_s^{3,k}} \overset{d}{=} Z^{h_s^{3,j}} \quad \Longrightarrow \quad \mathrm{d}\mathring{\tilde{h}}_{s,k}^2 = \mathring{\tilde{h}}_{s,j}^2 \quad \forall k, j. \tag{344}$$

The router logits at a given time step $s < t$ are

$$\mathring{h}_{s,k}^2 = \mathring{h}_{0,k}^2 + \sum_{p=1}^{s} \delta\mathring{h}_{p,k}^2,$$

where $\mathring{h}^2_{0,k} = \mathring{h}^2_{0,j}$ for any experts k, j due to the initialisation assumption. We now show that the updates $\delta\mathring{h}^2_{p,k}$ are the same for each expert index k. The infinite width limit router weight updates and corresponding logit updates for each expert is

$$Z^{(\delta W^2_p)}_{k,:} = -\eta \sum_{\beta=1}^{d_{\text{out}}} (\mathring{\tilde{h}}^2_{p,k})_\beta (\mathring{\boldsymbol{\chi}}_{p-1})_\beta Z^{h^1_{p-1}}, \tag{345}$$

$$\delta\mathring{h}^2_{p,k} = \mathring{\theta}_{W^2_{p-1}} \mathring{\theta}_{\delta h^1_p} \mathbb{E}[Z^{(\overline{W^2_{p-1}})_{k,:}} Z^{\delta h^1_p}] - \mathring{\theta}'_{\delta W^2_p} \eta \sum_{\beta=1}^{d_{\text{out}}} (\mathring{\tilde{h}}^2_{p-1,k})_\beta (\mathring{\boldsymbol{\chi}}_{p-1})_\beta \mathbb{E}[Z^{h^1_{p-1}} Z^{h^1_p}]. \tag{346}$$

As the inductive hypothesis implies $\mathring{\tilde{h}}^2_{p,k} = \mathring{\tilde{h}}^2_{p,j} \forall k, j$, we see that updates are the same for each expert a every time step $p < s < t$ and hence that

$$\mathring{h}^2_{s,k} = \mathring{h}^2_{s,j} \quad \forall k, j.$$

As the gradient for each expert k at time step s is

$$Z^{\mathrm{d}h^{3,k}_s} = \mathring{\tilde{h}}^2_{s,k} Z^{\mathrm{d}h^3_s},$$

where $Z^{\mathrm{d}h^3_s}$ is shared for all the experts. Using the fact that $\mathring{\tilde{h}}^2_{s,k} = \mathring{\tilde{h}}^2_{s,j}$ for all experts k, j, we get

$$Z^{\mathrm{d}h^{3,k}_s} \overset{d}{=} Z^{\mathrm{d}h^{3,j}_s} \quad \forall k, j.$$

Consider the update at step t, defined in the infinite width limit as:

$$\mathbf{h}^{3,k}_t = \mathbf{h}^{3,k}_{t-1} + \delta\mathbf{h}^{3,k}_t, \tag{347}$$

$$\begin{aligned} Z^{\delta h^{3,k}_t} &= \mathring{\theta}_{\delta h^1_t / \delta h^{3,k}_t} Z^{W^{3,k}_0 \delta h^1_t} - \sum_{s=0}^{t-2} \mathring{\theta}'_{\delta W^{3,k}_s \delta h^1_t / \delta h^{3,k}_t} \eta \sum_{\beta=1}^{d_{\text{out}}} Z^{(\mathrm{d}h^{3,k}_s)}_\beta (\mathring{\boldsymbol{\chi}}_s)_\beta \mathbb{E}[Z^{h^1_s} Z^{\delta h^1_t}] \\ &\quad - \mathring{\theta}'_{\delta W^{3,k}_t / \delta h^{3,k}_t} \eta \sum_{\beta=1}^{d_{\text{out}}} Z^{(\mathrm{d}h^{3,k}_{t-1})_\beta} (\mathring{\boldsymbol{\chi}}_{t-1})_\beta \mathbb{E}[Z^{h^1_{t-1}} Z^{h^1_t}], \end{aligned} \tag{348}$$

Note that $\mathring{\theta}'_{\delta W^{3,k}_t}$ is the same for each expert. As we have shown, inductive hypothesis implies that $Z^{\mathrm{d}h^{3,k}_s} \overset{d}{=} Z^{\mathrm{d}h^{3,j}_s}$, for all experts k, j at every time step $s < t$. Furthermore, $Z^{W^{3,k}_0 \delta h^1_t}$ (from the ZMatMul rule) is the same for each expert as $Z^{W^{3,k}_0}$ is the same for each expert and $Z^{\delta h^1_t}$ is shared by all the experts. Thus, using the inductive hypothesis that $Z^{h^{3,k}_{t-1}} \overset{d}{=} Z^{h^{3,j}_{t-1}}$ we have,

$$Z^{\delta h^{3,k}_t} \overset{d}{=} Z^{\delta h^{3,j}_t} \implies Z^{h^{3,k}_t} \overset{d}{=} Z^{h^{3,j}_t} \quad \forall k, j. \tag{349}$$

This is turn implies that

$$\mathrm{d}\mathring{\tilde{h}}^2_{t,k} = \mathrm{d}\mathring{\tilde{h}}^2_{t,j} \quad \forall k, j. \tag{350}$$

■

Corollary 20 *Consider a soft MoE network without a router bias with m experts with either softmax or sigmoid gating parametrised such that stability holds throughout training (Theorem 9), and such that the router logits converge to zero in the infinite width limit, $\mathring{\mathbf{h}}_0^2 = 0$ (Theorem 11, Table 1). For any time step $t \geq 0$, the router probability is the same for every expert in the infinite width limit. That is,*

$$\mathring{\tilde{h}}_{t,i}^2 = \mathring{\tilde{h}}_{t,j}^2 \tag{351}$$

for every expert $i, j \in [m]$.

Proof Result follows from the inductive proof in Theorem 19. For the last step of the induction, the result of Theorem 19, which implies that the gradient of the router logits are the same for each expert index $\mathrm{d}\mathring{h}_{t,k}^2 = \mathrm{d}\mathring{h}_{t,j}^2$. This, combined with the initialisation $\mathring{\mathbf{h}}_0^2 = 0$ implies that $\mathring{h}_{t,i}^2 = \mathring{h}_{t,j}^2$ for every expert $i, j \in [m]$ and time step t. The result for $\mathring{\tilde{h}}_{t,i}^2 = \mathring{\tilde{h}}_{t,j}^2$ for every expert $i, j \in [m]$ follows. ■

Corollary 21 *Consider a Softmax-Soft MoE netwok without a router bias. For any time step $t \geq 0$, in the infinite width limit, the gradient with respect to the router vanishes,*

$$\mathrm{d}\mathring{h}_{t,k}^2 = 0, \quad \text{for every expert } k. \tag{352}$$

Consequently, the router weights do not evolve from initialisation,

$$Z^{(\delta W_t^2)_{k,:}} = 0, \tag{353}$$

$$\implies \mathring{\tilde{h}}_{t,k}^2 = \mathbb{E}[Z^{(\overline{W_0^2})_{k,:}} Z^{\delta h_t^1}], \tag{354}$$

for every k. That is, in the limit, the router is stuck at initialisation.

Proof The limiting value of the gradient with respect to the router is

$$\mathrm{d}\mathring{\tilde{h}}_{t,k}^2 = \sum_{j=1}^{m} J(\mathring{h}_t^2)_{kj} \mathrm{d}\mathring{\tilde{h}}_{t,j}^2, \tag{355}$$

where $J(h^2)_{kj} = \tilde{h}_k^2(\delta_{kj} - \tilde{h}_j^2)$ is the Jacobian of the Softmax and $\tilde{h}^2 = \mathrm{Softmax}(h^2)$. From Theorem 19, $\mathrm{d}\mathring{\tilde{h}}_{t,j}^2 = \mathrm{d}\mathring{\tilde{h}}_{t,k}^2$ for all experts k, j. Let $C := \mathrm{d}\mathring{\tilde{h}}_{t,j}^2$. For every k, we have

$$\mathrm{d}\mathring{\tilde{h}}_{t,k}^2 = \sum_{j=1}^{m} J(\mathring{h}_t^2)_{kj} C \tag{356}$$

$$= C\mathring{\tilde{h}}_{t,k}^2 \sum_{j=1}^{m} \left(\delta_{kj} - \mathring{\tilde{h}}_{t,j}^2\right) \tag{357}$$

$$= 0, \tag{358}$$

where we use the fact that $\sum_{j=1}^{m} \mathring{\tilde{h}}_{t,j}^2 = 1$. The rest follows from looking at the infinite width limits of $\delta \mathbf{W}_t^2$ and $\delta \mathbf{h}_{t,k}^2$. ■

Remark 22 (Router bias prevents the stagnation in Theorem 20) *The reason for the router stagnation is the symmetric initialisation due to μP-MoE and the resulting identical deterministic gradients for each expert. A router bias (or noise) ensures that at initialisation the router is not symmetric, hence leading to varied deterministic gradient for each expert. The router thus does not stagnate.*

Remark 23 (Results hold for any stable parametrisation) *For the above results, we only assume stability and the results are shown using unscaled quantities (quantities that scale $\Theta(1)$). As such, any values of bcd do not change our results. We also assume $\mathring{\tilde{\mathbf{h}}}_0^2 = 0$ — the only stable parametrisation where this is not true is with $b_2 = 0.5$. We refer to (Yang and Hu, 2021, H.4.1) for discussion that. Furthermore, $b_2 = 1$ is required for feature learning considerations (Theorem 11).*

Remark 24 *(Extension to Adam) The above theorems hold trivially for Adam. The only change required is in the proof for Theorem 19. Here, the infinite width limit update to $\mathbf{h}_t^{3,k}$ will be (Appendix C.4)*

$$Z^{\delta h_t^{3,k}} = \mathring{\theta}_{\delta h_t^1 / \delta h_t^{3,k}} Z^{W_0^{3,k} \delta h_t^1} \tag{359}$$

$$- \sum_{s=0}^{t-2} \mathring{\theta}'_{\delta W_s^{3,k} \delta h_t^1 / \delta h_t^{3,k}} \eta \mathbb{E}_{\left(Z_1^{h_s^1}, Z_1^{\delta h_t^1}\right)} \left[Q_t^3 \left(\sum_\beta^{d_{out}} Z^{(\mathrm{d}h_s^{3,k})_\beta} (\mathring{\boldsymbol{\chi}}_s)_\beta Z_1^{h_s^1} \right) Z_1^{\delta h_t^1} \right] \tag{360}$$

$$- \mathring{\theta}'_{\delta W_t^{3,k} / \delta h_t^{3,k}} \eta \mathbb{E}_{\left(Z_1^{h_{t-1}^1}, Z_1^{h_t^1}\right)} \left[Q_t^3 \left(\sum_\beta^{d_{out}} Z^{(\mathrm{d}h_{t-1}^{3,k})_\beta} (\mathring{\boldsymbol{\chi}}_{t-1})_\beta Z_1^{h_{t-1}^1} \right) Z_1^{h_t^1} \right], \tag{361}$$

where Q is the Adam update function (Theorem 13). Here it is easy to see that $Z^{\mathrm{d}h_s^{3,k}} \overset{d}{=} Z^{\mathrm{d}h_s^{3,j}}$, for all experts k, j at every time step $s < t$ implies the same result as Theorem 19. The result from Theorem 21 follows.

Remark 25 (Scale of $\mathrm{d}h_{t,k}^2$ for finite n) *In the tensor program, for each output coordinate β we have*

$$(\mathrm{d}\tilde{h}_{t,k}^2)_\beta = \frac{1}{n} \sum_{\alpha=1}^n \theta'_{W_t^4} \theta_{h_t^{3,k}} (h_t^{3,k})_\alpha (\mathrm{d}h_t^3)_{\alpha\beta} = \theta'_{W_t^4} \theta_{h_t^{3,k}} \frac{1}{n} \sum_{\alpha=1}^n (Y_{t,k})_{\alpha\beta}, \qquad \texttt{(Moment)} \tag{362}$$

where we defined

$$(Y_{t,k})_{\alpha\beta} := (h_t^{3,k})_\alpha (\mathrm{d}h_t^3)_{\alpha\beta}. \tag{363}$$

The gradient with respect to the router logits is

$$(\mathrm{d}h_{t,k}^2)_\beta = \sum_{j=1}^m J(\tilde{h}_t^2)_{kj} (\mathrm{d}\tilde{h}_{t,j}^2)_\beta = \theta'_{W_t^4} \theta_{h_t^{3,k}} \frac{1}{n} \sum_{\alpha=1}^n X_{t,k,\alpha\beta}, \qquad \texttt{(Moment)} \tag{364}$$

where

$$X_{t,k,\alpha\beta} := \sum_{j=1}^m J(\tilde{h}_t^2)_{kj} (Y_{t,j})_{\alpha\beta}. \tag{365}$$

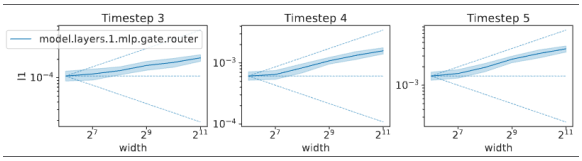


Figure 2: Average norm of router features as width is increased with router $d = 0.5$. Shows that for high ϵ, this causes the router logits to diverge.

In the infinite-width limit, we showed above that $\mathrm{d}\mathring{\tilde{h}}^2_{t,k} = 0$*, so in particular* $\mathbb{E}[X_{t,k,\alpha\beta}] = 0$*. To capture the scale, consider the second-moment "Moment" variable*

$$\theta'_{W_t^4}\theta_{h_t^{3,k}}\frac{1}{n}\sum_{\alpha=1}^{n}(X_{t,k,\alpha\beta})^2 = \theta'_{W_t^4}\theta_{h_t^{3,k}}\sum_{j=1}^{m}\sum_{i=1}^{m}J(\tilde{h}_t^2)_{kj}J(\tilde{h}_t^2)_{ki}\frac{1}{n}\sum_{\alpha=1}^{n}(Y_{t,j})_{\alpha\beta}(Y_{t,i})_{\alpha\beta}, \tag{366}$$

whose infinite-width limit, by the Master theorem, is

$$\mathring{\theta}'_{W_t^4}\mathring{\theta}_{h_t^{3,k}}\sum_{j=1}^{m}\sum_{i=1}^{m}J(\mathring{\tilde{h}}_t^2)_{kj}J(\mathring{\tilde{h}}_t^2)_{ki}\mathbb{E}\big[Z^{Y_{t,j}}Z^{Y_{t,i}}\big]. \tag{367}$$

Thus the variance of $(\mathrm{d}h^2_{t,k})_\beta = n^{-1}\sum_{\alpha=1}^{n}X_{t,k,\alpha\beta}$ *scales as*

$$\mathrm{Var}\big[(\mathrm{d}h^2_{t,k})_\beta\big] = \frac{1}{n}(\theta'_{W_t^4}\theta_{h_t^{3,k}})^2\mathrm{Var}(X_{t,k,\alpha\beta}), \tag{368}$$

where $\mathrm{Var}(X_{t,k,\alpha\beta})$ *is some non-zero scalar. Consequently, we have that* $\mathrm{d}\tilde{h}^2_{t,k}$ *scales* $\Theta(n^{-\frac{1}{2}})$*.*

Remark 26 (Scale of $\mathbf{h}_t^2 - \mathbf{h}_{t-1}^2$ for Softmax-Soft MoE networks under μP-Moe parametrisation) *Here, we find the scale of* $\mathbf{h}_t^2 - \mathbf{h}_{t-1}^2$ *for Softmax-Soft MoE networks under the μP-MoE parametrisation. We find the scale under zero-initisalisation of* $\mathbf{W}_0^2$*, so that the initialisation of the network matches the infinite-width limit, as recommended in Yang et al. (2021). Here, only the effective updates decide the scale of the feature update. The scale of the router weight update is* $\theta_{\delta W_t^2} = n^{-1}\cdot n^{-0.5} = n^{-1.5}$*, where the extra* $n^{-0.5}$ *comes from the finite width scale of the router gradient, due to the softmax Jacobian (Theorem 25). Thus, the scale of* $\mathbf{h}_t^2 - \mathbf{h}_{t-1}^2$ *is* $\theta'_{\delta W_t^2} = n\cdot n^{-1.5} = n^{-0.5}$*.*

Remark 27 (Benign Scaling for Adam) *In Softmax-Soft MoEs, the scale of the gradient has an extra* $n^{-0.5}$ *factor. This implies setting* $d = 0.5$ *for the router can be used to rescale the variance such that it is* $\Theta(1)$*. However, as this results in a random scalar in the program, this limit is not expressible in Tensor Programs (Yang and Hu, 2021, Theorem 7.4). In practice we find this can lead to diverging logits Figure 2.*

Appendix E. Proof of Theorem 4

From Appendix C.2, we see that at initialisation the router logits converge deterministically to $C\mathbf{1}$ (with $C = 0$) under μP-MoE scaling without a bias. A simple CLT argument, (formalised

in Yang (2019)) shows that $\sqrt{n}(\mathbf{h}^2(n) - C\mathbf{1})$ converges in distribution to a Gaussian with variance proportional to $\mathbb{E}[(Z^{h^1})^2]$. Thus, we can write

$$\mathbf{h}^2(n) = C\mathbf{1} + \sigma_n \epsilon_n,$$

with $\epsilon_n = \sqrt{n}(\mathbf{h}^2(n) - C\mathbf{1})$ and $\sigma_n = \frac{1}{\sqrt{n}}$.

Theorem 28 (Limit of the Top-K masked vector) *Let*

$$S_n := \mathrm{TopK}(\varepsilon_n) = \mathrm{TopK}\big(\{h_i^2(n)\}_{i=1}^m\big),$$

and define

$$\tilde{h}_i^2(n) := h_i^2(n)\,\mathbf{1}\{i \in S_n\}, \qquad \tilde{\mathbf{h}}^2(n) := (\tilde{\mathbf{h}}_1^2(n), \ldots, \tilde{\mathbf{h}}_M^2(n)).$$

Then,

$$\tilde{\mathbf{h}}^2(n) \overset{d}{\Rightarrow} C\,\mathbf{1}_S \quad \text{in } \mathbb{R}^M, \text{ where } S := \mathrm{TopK}(\varepsilon)$$

where 1_S is the indicator function indicating whether index $j \in S$.

Proof Define the Top-K indicator map

$$T : \mathbb{R}^M \to \{0,1\}^M, \qquad T(z) := \mathbf{1}_{\mathrm{TopK}(z)}.$$

This map is discontinuous only at points where ties occur among the order statistics. As the probability of ties when sampling from ϵ is zero (as Gaussian as a continuous joint density),

$$\mathbb{P}(\varepsilon \in D_T) = 0,$$

where D_T denotes the discontinuity set of T. Hence, by the extended continuous mapping theorem (Billingsley, 2013),

$$T(\varepsilon_n) \overset{d}{\Rightarrow} T(\varepsilon), \qquad \text{i.e. } \mathbf{1}_{S_n} \overset{d}{\Rightarrow} \mathbf{1}_S.$$

Now write

$$\tilde{\mathbf{h}}^2(n) = \big(C\mathbf{1} + \sigma_n \varepsilon_n\big) \odot T(\varepsilon_n) = C\,T(\varepsilon_n) + \sigma_n\big(\varepsilon_n \odot T(\varepsilon_n)\big).$$

Since $\varepsilon_n \overset{d}{\Rightarrow} \varepsilon$, the sequence $\{\varepsilon_n\}$ is tight, and therefore

$$\|\varepsilon_n\|_\infty = O_p(1).$$

As $\sigma_n \to 0$, it follows that

$$\Big\|\sigma_n\big(\varepsilon_n \odot T(\varepsilon_n)\big)\Big\|_\infty \le \sigma_n \|\varepsilon_n\|_\infty \overset{p}{\to} 0.$$

Combining this with $\mathbf{1}_{S_n} \overset{d}{\Rightarrow} \mathbf{1}_S$ and applying Slutsky's theorem yields

$$\tilde{\mathbf{h}}^2(n) \overset{d}{\Rightarrow} C\,\mathbf{1}_S.$$

■

Thus, in the first forward pass (at initialisation), where the router is symmetric, the Top-K index set is not deterministic as $S := \mathrm{TopK}(\epsilon)$.

Appendix F. Tensor Program for Top-K MoE

In this section we derive the Tensor program and the corresponding infinite width limit for Top-K MoEs (**??**). Note we only derive this for SGD, corresponding limit to Adam is analogous to Appendix C.3.

F.1. Tensor Program for Top-K MoE with SGD

We use the same notation as in Appendix C.1. Throughout we make the assumption that the parametrisation is stable (Theorem 9).

Network:

$$\mathbf{h}^1 = \mathbf{W}^1 \boldsymbol{\xi}, \qquad \in \mathbb{R}^{n\times 1} \tag{369}$$

$$\mathbf{h}^2 = \mathbf{W}^2 \mathbf{h}^1 + \mathbf{b}^2, \qquad \in \mathbb{R}^{m\times 1} \tag{370}$$

$$\mathbf{s} = \mathrm{T}(\mathbf{h}^2) \qquad \text{where } \mathrm{T}_k(\mathbf{h}^2) := \mathbf{1}\{k \in \mathrm{TopK}(\mathbf{h}^2)\} \tag{371}$$

$$\tilde{\mathbf{h}}^2 = \mathrm{G}(\mathbf{h}^2, \mathbf{s}), \qquad \in [0,1]^{m\times 1} \tag{372}$$

$$\mathbf{h}^{3,k} = \mathbf{W}^{3,k} \mathbf{h}^1, \qquad \in \mathbb{R}^{n\times 1} \tag{373}$$

$$\mathbf{h}^3 = \sum_{k=1}^{m} \mathbf{h}^{3,k} \tilde{h}^2_k, \qquad \in \mathbb{R}^{n\times 1} \tag{374}$$

$$\mathbf{f} = \mathbf{W}^4 \mathbf{h}^3, \qquad \in \mathbb{R}^{d_{\text{out}}\times 1} \tag{375}$$

where $\mathbf{b}^2 \in \mathbb{R}^m$ is a bias parameter that is represented as an initial scalar in the program, $\mathbf{s}$ is the Top-K mask, and $\mathrm{G}(\cdot, \mathbf{s})$ is the activations applied to the index set $\mathbf{s}$. For Softmax

$$\mathrm{G}_k(u, s) := \frac{s_k \exp(u_k)}{\sum_{j=1}^m s_j \exp(u_j)}, \tag{376}$$

and for Sigmoid

$$\mathrm{G}_k(u, s) := s_k \, \sigma(u_k).$$

Conditioned on noise/bias: We derive the Tensor program conditioned on the bias term noises $\{\mathbf{b}^2_t\}_{t=0}^T$ or the learnable bias $\mathbf{b}^2$. We assume that they have been sampled from a distribution such that $\mathbb{P}(b^2_i = b^2_j) = 0$. Practically, we sample $b_i \sim \mathcal{N}(0,1)$.

Remark 29 *Top-K MoEs are not admissible in the Tensor program framework without a router bias. This is because the router history is not deterministic, whereas the Tensor program Master Theorem requires scalars (such as router mask) to converge to deterministic constants.*

First Forward Pass:

$$\mathbf{h}_0^1 = \theta_{W_0^1}\overline{\mathbf{W}_0^1}\boldsymbol{\xi}, \qquad (\mathbf{h}_0^1)_\alpha = \phi\left(\{(\overline{\mathbf{W}_0^1})_{\alpha\beta}\}_{\beta=1}^{d_{\text{in}}}; \xi_1, \ldots, \xi_{d_{\text{in}}}, \theta_{W_0^1}\right) = \sum_{j=1}^{d_{\text{in}}} \theta_{W_0^1}(\overline{\mathbf{W}_0^1})_{\alpha j}\xi_j,$$

$$\mathbf{h}_0^2 = \theta'_{W_0^2}\frac{1}{n}\overline{\mathbf{W}_0^2}\mathbf{h}_0^1 + \mathbf{b}_0^2, \qquad (\mathbf{h}_0^2)_k = \frac{1}{n}\sum_{\alpha=1}^{n}\phi\left((\overline{\mathbf{W}_0^2})_{k\alpha}, (\mathbf{h}_0^1)_\alpha; \theta'_{W_0^2}, \{(b_0^2)_k\}_{k=1}^m\right) = \frac{1}{n}\theta'_{W_0^2}\sum_{\alpha=1}^{n}(\overline{\mathbf{W}_0^2})_{k\alpha}(\mathbf{h}_0^1)_\alpha + (\mathbf{b}$$

$$\mathbf{s} = \mathrm{T}(\mathbf{h}^2), \qquad (\texttt{Moment})$$

$$\tilde{\mathbf{h}}_0^2 = \mathrm{G}(\hat{\mathbf{h}}_0^2, \mathbf{s}), \qquad (\texttt{Moment}, \text{ See G})$$

$$\hat{\mathbf{h}}_0^{3,k} = \overline{\mathbf{W}^{3,k}}\mathbf{h}_0^1, \qquad (\texttt{MatMul})$$

$$\mathbf{h}_0^{3,k} = \theta_{W_0^3}\hat{\mathbf{h}}_0^{3,k}, \qquad (\mathbf{h}_0^{3,k})_\alpha = \phi\left((\hat{\mathbf{h}}_0^{3,k})_\alpha; \theta_{W_0^3}\right) = \theta_{W_0^3}(\hat{\mathbf{h}}_0^3)_\alpha$$

$$\mathbf{h}_0^3 = \sum_{k=1}^{m}\mathbf{h}_0^{3,k}\tilde{h}_{0,k}^2, \qquad (\mathbf{h}_0^3)_\alpha = \phi\left((\mathbf{h}_0^{3,1})_\alpha, \ldots, (\mathbf{h}_0^{3,m})_\alpha; \tilde{h}_{0,1}^2, \ldots, \tilde{h}_{0,m}^2\right) = \sum_{k=1}^{m}(\mathbf{h}_0^{3,k})_\alpha\tilde{h}_{0,k}^2$$

$$\mathbf{f}_0 = \theta'_{W_0^4}\frac{1}{n}\overline{\mathbf{W}_0^4}\mathbf{h}_0^3, \qquad (\mathbf{f}_0)_\beta = \frac{1}{n}\sum_{\alpha=1}^{n}\phi\left((\overline{\mathbf{W}_0^4})_{\beta\alpha}, (\mathbf{h}_0^3)_\alpha; \theta'_{W_0^4}\right) = \frac{1}{n}\theta'_{W_0^4}\sum_{\alpha=1}^{n}(\overline{\mathbf{W}_0^4})_{\beta\alpha}(\mathbf{h}_0^3)_\alpha.$$

We assume stability at initialisation and thus the initialisation values in Theorem 11. For these values, the scale of $\mathbf{W}_0^2\mathbf{h}_0^1$ is $n^{-0.5}$.

Remark 30 (Initial mask is deterministic conditioned on the bias) *With the initialisation values in Theorem 11, as the width increases $\mathbf{W}_0^2\mathbf{h}_0^1 \to \mathbf{0}$, with vanishing finite-width fluctuation (Yang, 2019). Consequently, if $\mathring{\mathbf{h}}_0^2$ denotes the limit of $\mathbf{h}_0^2$*

$$\mathrm{T}(\mathring{\mathbf{h}}_0^2) = \mathrm{T}(\mathbf{b}_0^2).$$

Hence the Top-K operator is determined by the initial bias and is deterministic in the limit. This can be shown with the result in Appendix E.

Noise and Learnable Bias: We consider two cases here:

1. Case 1: A different noise is added at each time step $\{\mathbf{b}_t^2\}_{t=1}^T$.
2. Case 2: The term $\mathbf{b}_0^2$ is learned. Note that the updates below are written only for learnable biases.

First Backward Pass: Same as Appendix C.1, except the gradient $\mathrm{d}h_{0,k}^2$ requires gradient of the Top-K mask.

$$\mathrm{d}h_{0,k}^2 = \mathbf{s}_0\sum_{j\in\mathcal{S}}J(\tilde{\mathbf{h}}_0^2)_{jk}\mathrm{d}\tilde{h}_{0,j}^2, \qquad \in \mathbb{R}^{1\times d_{\text{out}}} \quad (\texttt{Moment}) \tag{377}$$

where $J(\tilde{\mathbf{h}}_0^2)_{jk} = \frac{\partial(\tilde{\mathbf{h}}_0^2)_j}{\partial(\hat{\mathbf{h}}_0^2)_k}$ is the Jacobian of the activation function on the sparse logits.

First Weight Updates: We need to consider the update to the learnable $\mathbf{b}^2$

$$\mathbf{b}_1^2 - \mathbf{b}_0^2 = -\eta\theta'_{W_0^4}\mathrm{d}\mathbf{h}_0^2\boldsymbol{\chi}_0. \tag{378}$$

The rest is the same as Appendix C.1.

First Feature Updates: The only feature update that will change from Appendix C.1 is $\mathbf{h}^2$ and $\tilde{\mathbf{h}}^2$:

$$\mathbf{h}_1^2 - \mathbf{h}_0^2 = \mathbf{W}_0^2(\mathbf{h}_1^1 - \mathbf{h}_0^1) + (\mathbf{W}_1^2 - \mathbf{W}_0^2)\mathbf{h}_1^1 + \mathbf{b}_1^2 - \mathbf{b}_0^2 \tag{379}$$

$$= \theta_{\delta h_1^1}\theta'_{W_0^2}\frac{\overline{\mathbf{W}_0^2}\delta\mathbf{h}_1^1}{n} - \theta'_{\delta W_1^2}\eta \mathrm{d}\mathbf{h}_0^2\boldsymbol{\chi}_0\frac{(\mathbf{h}_0^1)^T\mathbf{h}_1^1}{n} - \eta\theta'_{W_0^4}\mathrm{d}\mathbf{h}_0^2\boldsymbol{\chi}_0 \tag{380}$$

$$\theta_{h_1^2}(\delta\mathbf{h}_1^2)_k = \frac{1}{n}\sum_{\alpha=1}^{n}\phi\left(\overline{\mathbf{W}_0^2}_{k\alpha}, (\delta\mathbf{h}_1^1)_\alpha, (\mathbf{h}_0^1)_\alpha, (\mathbf{h}_1^1)_\alpha; \theta_{\delta h_1^1}, \theta'_{W_0^2}, \theta'_{\delta W_1^2}, \eta, \{(\mathrm{d}\mathbf{h}_0^2)_{k\beta}, (\boldsymbol{\chi}_0)_\beta\}_\beta^{d_{\text{out}}}\right) \quad \texttt{(Moment)} \tag{381}$$

$$\phi(\ldots) := \theta_{\delta h_1^1}\theta'_{W_0^2}\overline{\mathbf{W}_0^2}_{k\alpha}(\delta\mathbf{h}_1^1)_\alpha - \theta'_{\delta W_1^2}\eta\sum_{\beta=1}^{d_{\text{out}}}(\mathrm{d}\mathbf{h}_0^2)_{k\beta}(\boldsymbol{\chi}_0)_\beta(\mathbf{h}_0^1)_\alpha(\mathbf{h}_1^1)_\alpha - \eta\theta'_{W_0^4}\sum_{\beta=1}^{d_{\text{out}}}(\mathrm{d}\mathbf{h}_0^2)_{k\beta}(\boldsymbol{\chi}_0)_\beta, \tag{382}$$

$$\theta_{\delta h_1^2} := \max(\theta_{\delta h_1^1}\theta'_{W_0^2}, \theta'_{\delta W_1^2}, \theta'_{W_0^4}). \tag{383}$$

$$\tilde{\mathbf{h}}_1^2 - \tilde{\mathbf{h}}_0^2 = \mathrm{G}(\mathbf{h}_1^2, \mathbf{s}_1) - \mathrm{G}(\mathbf{h}_0^2, \mathbf{s}_0), \quad \texttt{(Moment)} \tag{384}$$

The rest of the feature updates are the same as Appendix C.1.

***t*th Backward Pass:** Same as Appendix C.1, except for the mask

$$\mathrm{d}h_{t,k}^2 = \mathbf{s}_t\sum_{j\in\mathcal{S}}J(\tilde{\mathbf{h}}_t^2)_{jk}\mathrm{d}\tilde{h}_{t,j}^2, \quad \in\mathbb{R}^{1\times d_{\text{out}}} \quad \texttt{(Moment)} \tag{385}$$

where $J(\tilde{\mathbf{h}}_t^2)_{jk} = \frac{\partial(\tilde{\mathbf{h}}_t^2)_j}{\partial(\hat{\mathbf{h}}_t^2)_k}$ is the Jacobian of the activation function on the sparse logits.

***t*th Weight Updates:** We need to consider the update to $\mathbf{b}^2$

$$\mathbf{b}_t^2 - \mathbf{b}_{t-1}^2 = -\eta\theta'_{W_{t-1}^4}\mathrm{d}\mathbf{h}_{t-1}^2\boldsymbol{\chi}_{t-1}. \tag{386}$$

The rest is the same as Appendix C.1.

***t*th Feature Updates:** The only feature update that will change from Appendix C.1 is $\mathbf{h}^2$ and $\hat{\mathbf{h}}^2$:

$$\mathbf{h}_t^2 - \mathbf{h}_{t-1}^2 = \mathbf{W}_{t-1}^2(\mathbf{h}_t^1 - \mathbf{h}_{t-1}^1) + (\mathbf{W}_t^2 - \mathbf{W}_{t-1}^2)\mathbf{h}_t^1 + \mathbf{b}_t^2 - \mathbf{b}_{t-1}^2 \tag{387}$$

$$= \theta_{\delta h_t^1}\theta'_{W_{t-1}^2}\frac{\overline{\mathbf{W}_{t-1}^2}\delta\mathbf{h}_t^1}{n} - \theta'_{\delta W_t^2}\eta\mathrm{d}\mathbf{h}_{t-1}^2\boldsymbol{\chi}_{t-1}\frac{(\mathbf{h}_{t-1}^1)^T\mathbf{h}_t^1}{n} - \eta\theta'_{W_{t-1}^4}\mathrm{d}\mathbf{h}_{t-1}^2\boldsymbol{\chi}_{t-1} \tag{388}$$

$$\theta_{h_1^2}(\delta\mathbf{h}_t^2)_k = \frac{1}{n}\sum_{\alpha=1}^{n}\phi\left(\overline{\mathbf{W}_{t-1}^2}_{k\alpha}, (\delta\mathbf{h}_t^1)_\alpha, (\mathbf{h}_{t-1}^1)_\alpha, (\mathbf{h}_t^1)_\alpha; \theta_{\delta h_t^1}, \theta'_{W_{t-1}^2}, \theta'_{\delta W_t^2}, \eta, \{(\mathrm{d}\mathbf{h}_{t-1}^2)_{k\beta}, (\boldsymbol{\chi}_{t-1})_\beta\}_\beta^{d_{\text{out}}}\right) \tag{389}$$

$$\phi(\ldots) := \theta_{\delta h_t^1}\theta'_{W_{t-1}^2}\overline{\mathbf{W}_{t-1}^2}_{k\alpha}(\delta\mathbf{h}_t^1)_\alpha - \theta'_{\delta W_t^2}\eta\sum_{\beta=1}^{d_{\text{out}}}(\mathrm{d}\mathbf{h}_{t-1}^2)_{k\beta}(\boldsymbol{\chi}_{t-1})_\beta(\mathbf{h}_{t-1}^1)_\alpha(\mathbf{h}_t^1)_\alpha - \eta\theta'_{W_{t-1}^4}\sum_{\beta=1}^{d_{\text{out}}}(\mathrm{d}\mathbf{h}_{t-1}^2)_{k\beta}(\boldsymbol{\chi}_{t-} \tag{390}$$

$$\theta_{\delta h_t^2} := \max(\theta_{\delta h_t^1}\theta'_{W_{t-1}^2}, \theta'_{\delta W_t^2}, \theta'_{W_{t-1}^4}). \tag{391}$$

$$\tilde{\mathbf{h}}_t^2 - \tilde{\mathbf{h}}_{t-1}^2 = \mathrm{G}(\mathbf{h}_t^2) - \mathrm{G}(\mathbf{h}_{t-1}^2), \qquad \texttt{(Moment)} \tag{392}$$

The rest will be the same as Appendix C.1.

F.2. Infinite Width Limit for TopK MoE with SGD

We use the same notation as in Appendix C.2.

Remark 31 (Deterministic Limiting Mask Conditioned on Noise/Bias) *To find the limit, we need to show that the Top-K mask is deterministic in both cases: 1) where noise is added at each step to the router logits, 2) where a learnable bias is added to the router logits.*

1. *Note that the noise $\mathbf{b}_t^2$ is drawn independently of $\mathbf{W}_t\mathbf{h}_t^1$ at each step. This implies that the probability of ties $(\mathbf{h}^2)_i = (\mathbf{h}^2)_j$ is 0. $\mathbf{W}_t\mathbf{h}_t^1$ converges deterministically according to the Master Theorem: If $\mathbf{W}_t\mathbf{h}_t^1$ converges to the same value for each dimension such that the setting of Appendix E holds for $\mathbf{W}_t\mathbf{h}_t^1$, then the mask is $\mathrm{T}(\mathbf{b}_t^2)$ in the limit and thus deterministic.*

2. *With a learnable bias, the mask is deterministic at initialisation (Theorem 30). However, for subsequent steps, we must make the additional assumption that for widths $n > N$ for some $N > 0$, in the resulting logits $h_{t,(K)}^2 - h_{t,(K+1)}^2 > \epsilon$ where ϵ is width independent, for all t. As TP only allows for pseudo-Lipschitz non-linearities, scalars in the program converge to a deterministic value, and the logits are bounded under μP-MoE, this can be proven using induction. However, we leave this for future work.*

First Forward Pass:

$$Z^{h_0^1} = \boldsymbol{\xi} Z^{W_0^1}, \tag{393}$$

$$\mathring{\mathbf{h}}_0^2 = \mathbf{b}_0^2, \tag{394}$$

$$\mathring{\mathbf{s}}_0 = \mathrm{T}(\mathring{\mathbf{h}}_0^2) \tag{395}$$

$$\mathring{\tilde{\mathbf{h}}}_0^2 = \mathrm{G}(\mathring{\hat{\mathbf{h}}}_0^2, \mathring{\mathbf{s}}_0) \tag{396}$$

$$Z^{h_0^{3,k}} = Z^{W_0^{3,k} h_0^1} \text{ for every expert } k, \tag{397}$$

$$Z^{h_0^3} = \sum_{k=1}^{m} \mathring{\tilde{h}}_{0,k}^2 Z^{h_0^{3,k}} \tag{398}$$

$$\mathring{\mathbf{f}}_0 = \mathbf{0}. \tag{399}$$

Here, $Z^{W_0^1}$ is standard Gaussian by construction. The gating function G is either a Softmax or Sigmoid. $Z^{W_0^{3,k} h_0^1} = \hat{Z}^{W_0^{3,k} h_0^1}$ from the Z MatMul rule. Finally $\mathring{\mathbf{f}}_0 = \mathbf{0}$ as $\mathring{f}_0 = \mathbb{E}[Z^{W_0^4}]\mathbb{E}[Z^{h_0^3}] = 0$.

First Backward Pass:

$$Z^{\mathrm{d}h_0^3} = Z^{\overline{W_0^4}}, \tag{400}$$

$$\mathring{\mathrm{d}\tilde{h}}^2_{0,k} = 0 \text{ for every expert } k, \tag{401}$$

$$\mathring{\mathrm{d}h}^2_{0,k} = 0 \text{ for every expert } k, \tag{402}$$

$$Z^{\mathrm{d}h_0^{3,k}} = \tilde{h}^2_{0,k} Z^{\mathrm{d}h_0^3} \text{ for every expert } k, \tag{403}$$

$$Z^{\mathrm{d}h_0^1} = \sum_{k=1}^{m} Z^{(W_0^{3,k})^T \mathrm{d}h_0^{3,k}} \tag{404}$$

$$\tag{405}$$

$Z^{\overline{W_0^4}}$ is a standard Gaussian by construction. $\mathring{\mathrm{d}\tilde{h}}^2_{0,k} = \mathbb{E}[Z^{h_0^{3,k}}]\mathbb{E}[\overline{W_0^4}] = 0$. From this, $\mathring{\mathrm{d}h}^2_{0,k} = 0$ follows. $Z^{(W_0^{3,k})^T \mathrm{d}h_0^3} = \hat{Z}^{(W_0^{3,k})^T \mathrm{d}h_0^3}$ from the Z MatMul rule.

First Feature Update:

$$Z^{\delta h_1^1} = -\eta \sum_{\beta=1}^{d_{\text{out}}} Z^{(\mathrm{d}h_0^1)_\beta} (\mathring{\boldsymbol{\chi}}_0)_\beta \boldsymbol{\xi}^T \boldsymbol{\xi}, \tag{406}$$

$$\delta \mathring{h}^2_{1,k} = \mathring{\theta}_{\delta h_1^1} \mathbb{E}[Z^{(\overline{W_0^2})_{k,:}} Z^{\delta h_1^1}] - \mathring{\theta}'_{\delta W_1^2} \eta \sum_{\beta=1}^{d_{\text{out}}} (\mathrm{d}\mathring{h}^2_{0,k})_\beta (\mathring{\boldsymbol{\chi}}_0)_\beta \mathbb{E}[Z^{h_0^1} Z^{h_1^1}] - \eta \mathring{\theta}'_{W_0^4} \sum_{\beta=1}^{d_{\text{out}}} (\mathrm{d}\mathring{h}^2_{0,k})_\beta (\mathring{\boldsymbol{\chi}}_0)_\beta, \tag{407}$$

$$\delta \mathring{\mathbf{s}}_1^2 = \mathrm{T}(\mathring{\mathbf{h}}_1^2) - \mathrm{T}(\mathring{\mathbf{h}}_0^2) \text{ if } \mathring{\theta}_{\delta h_1^2} = 1, \tag{408}$$

$$\delta \mathring{\tilde{\mathbf{h}}}_1^2 = \mathrm{G}(\mathring{\mathbf{s}}_1^2, \mathring{\mathbf{s}}_1) - \mathrm{G}(\mathring{\mathbf{h}}_0^2, \mathring{\mathbf{s}}_0) \text{ if } \mathring{\theta}_{\delta h_1^2} = 1, \tag{409}$$

$$Z^{\delta h_1^{3,k}} = \mathring{\theta}_{\delta h_1^1 / \delta h_1^{3,k}} Z^{W_0^{3,k} \delta h_1^1} - \mathring{\theta}'_{\delta W_1^3 / \delta h_1^{3,k}} \eta \sum_{\beta=1}^{d_{\text{out}}} Z^{(\mathrm{d}h^{3,k})_\beta} (\mathring{\boldsymbol{\chi}}_0)_\beta \mathbb{E}[Z^{h_0^1} Z^{h_1^1}] \text{ for every expert } k, \tag{410}$$

$$Z^{\delta h_1^3} = \sum_{k=1}^{m} \mathring{\theta}_{\delta h_1^{3,k} / \delta h_1^3} \mathring{\tilde{h}}^2_{0,k} Z^{\delta h_1^{3,k}} - \mathring{\theta}_{\delta \tilde{h}_1^2} \mathring{\theta}_{h_1^{3,k} / \delta h^3} \delta \mathring{\tilde{h}}^2_{1,k} Z^{h_1^{3,k}}, \tag{411}$$

$$\delta \mathring{f}_1 = \mathring{\theta}_{\delta h_1^3} \mathbb{E}[Z^{\overline{W_0^4}} Z^{\delta h_1^3}] - \mathring{\theta}'_{\delta W_1^4} \eta \mathring{\chi}_0 \mathbb{E}[Z^{h_0^3} Z^{h_1^3}], \tag{412}$$

where G is either softmax or sigmoid and G′ is the derivative. Further, $Z^{W_0^{3,k} \delta h_1^1}$ is given by the Z MatMul rule,

$$Z^{W_0^{3,k} \delta h_1^1} = \hat{Z}^{W_0^{3,k} \delta h_1^1} + \dot{Z}^{W_0^{3,k} \delta h_1^1} \tag{413}$$

$$= \hat{Z}^{W_0^{3,k} \delta h_1^1} + Z^{\mathrm{d}h_0^{3,k}} \mathbb{E}\left[\frac{\partial Z^{\delta h_1^1}}{\partial \hat{Z}^{(W_0^{3,k})^T \mathrm{d}h_0^{3,k}}} \right]. \tag{414}$$

We write the limit of each vector in the program as

$$Z^{h_t} = Z^{h_0} + \mathring{\theta}_{\delta h_1} Z^{\delta h_1} + \ldots + \mathring{\theta}_{\delta h_t} Z^{\delta h_t}. \tag{415}$$

Weight updates: The distribution of certain weight updates are required in the backward pass and thus the forward passes. Note that $\mathbf{b}$ is a scalar in the program ($\in \mathbb{R}^m$) and hence does not have a limit.

$$Z^{\overline{\delta W_t^4}} = -\eta \mathring{\chi}_{t-1} Z^{h_{t-1}^3}, \tag{416}$$

$$Z^{(\overline{\delta W_t^2})_{k,:}} = -\eta \sum_{\beta=1}^{d_{\text{out}}} (\mathrm{d}\mathring{h}_{t,k}^2)_\beta (\mathring{\boldsymbol{\chi}}_{t-1})_\beta Z^{h_{t-1}^1}, \tag{417}$$

$$\mathbf{b}_t^2 - \mathbf{b}_{t-1}^2 = -\eta \mathring{\theta}'_{W_{t-1}^4} \sum_{\beta=1}^{d_{\text{out}}} (\mathrm{d}\mathring{h}_{t-1,k}^2)_\beta (\mathring{\boldsymbol{\chi}}_{t-1})_\beta. \tag{418}$$

tth Backward Pass: Let $\mathcal{S} = \mathrm{TopK}(\mathbf{h}_t^2)$ be the set of active experts,

$$Z^{\mathrm{d}h_t^3} = Z^{(\overline{W_t^4})^T}, \tag{419}$$

$$\mathrm{d}\mathring{\tilde{h}}_{t,k}^2 = \mathbb{E}[Z^{h_t^{3,k}} Z^{\mathrm{d}h_t^3}], \tag{420}$$

$$\mathrm{d}\mathring{h}_{t,k}^2 = \mathring{s}_{t,k} \sum_{j=1}^{m} J(\mathring{h}_t^2)_{kj} \mathrm{d}\mathring{\tilde{h}}_{t,k}^2, \tag{421}$$

$$Z^{\mathrm{d}h_t^{3,k}} = \mathring{\tilde{h}}_{t,k}^2 Z^{\mathrm{d}h_t^3}, \tag{422}$$

$$Z^{\mathrm{d}h_t^1} = \sum_{k=1}^{m} Z^{(\overline{W_t^2})_{k,:}} \mathrm{d}\mathring{h}_{t,k}^2 + Z^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}} - \sum_{s=1}^{t-1} Z^{h_s^1} \mathring{\chi}_s \mathbb{E}[Z^{\mathrm{d}h_s^{3,k}} Z^{\mathrm{d}h_t^{3,k}}], \tag{423}$$

where $Z^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}}$ is given by the Z MatMul rule,

$$Z^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}} = \hat{Z}^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}} + \dot{Z}^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}} \tag{424}$$

$$= \hat{Z}^{(W_0^{3,k})^T \mathrm{d}h_t^{3,k}} + \sum_{v \in \mathcal{V}: W_0^{3,k} v \in \mathcal{V}} Z^v \mathbb{E}\left[\frac{\partial Z^{\mathrm{d}h_t^{3,k}}}{\partial \hat{Z} W_0^{3,k} v}\right], \tag{425}$$

where $\mathcal{V} := \{v \text{ for some vector in the program } v\}$ is the set of all vectors in the program.

tth Forward Pass:

$$Z^{\delta h_t^1} = -\eta \sum_{\beta=1}^{d_{\text{out}}} Z^{(\mathrm{d}\mathbf{h}_{t-1}^1)_\beta} (\mathring{\boldsymbol{\chi}}_{t-1})_\beta \boldsymbol{\xi}^T \boldsymbol{\xi}, \tag{426}$$

$$\delta \mathring{h}_{t,k}^2 = \mathring{\theta}'_{W_{t-1}^2} \mathring{\theta}_{\delta h_t^1} \mathbb{E}[Z^{(\overline{W_{t-1}^2})_{k,:}} Z^{\delta h_t^1}] - \mathring{\theta}'_{\delta W_t^2} \eta \sum_{\beta=1}^{d_{\text{out}}} (\mathrm{d}\mathring{h}_{t-1,k}^2)_\beta (\mathring{\boldsymbol{\chi}}_{t-1})_\beta \mathbb{E}[Z^{h_{t-1}^1} Z^{h_t^1}] \tag{427}$$

$$- \eta \mathring{\theta}'_{W_{t-1}^4} \sum_{\beta=1}^{d_{\text{out}}} (\mathrm{d}\mathring{h}_{t-1,k}^2)_\beta (\mathring{\boldsymbol{\chi}}_{t-1})_\beta, \tag{428}$$

$$\delta \mathring{\mathbf{s}}_t^2 = \mathrm{T}(\mathring{\mathbf{h}}_t^2) - \mathrm{T}(\mathring{\mathbf{h}}_{t-1}^2) \text{ if } \mathring{\theta}_{\delta h_t^2} = 1, \tag{429}$$

$$\delta \mathring{\tilde{\mathbf{h}}}_t^2 = \mathrm{G}(\mathring{\mathbf{h}}_t^2, \mathbf{s}_t) - \mathrm{G}(\mathring{\mathbf{h}}_{t-1}^2, \mathbf{s}_{t-1}) \text{ if } \mathring{\theta}_{\delta h_t^2} = 1, \tag{430}$$

$$Z^{\delta h_t^{3,k}} = \mathring{\theta}_{\delta h_t^1 / \delta h_t^{3,k}} Z^{W_0^{3,k} \delta h_t^1} - \sum_{s=0}^{t-2} \mathring{\theta}'_{\delta W_s^{3,k} \delta h_t^1 / \delta h_t^{3,k}} \eta \sum_{\beta}^{d_{\text{out}}} Z^{(\mathrm{d}h_s^{3,k})_\beta} (\mathring{\boldsymbol{\chi}}_s)_\beta \mathbb{E}[Z^{h_s^1} Z^{\delta h_t^1}] \tag{431}$$

$$- \mathring{\theta}'_{\delta W_t^{3,k} / \delta h_t^{3,k}} \eta \sum_{\beta}^{d_{\text{out}}} Z^{(\mathrm{d}h_{t-1}^{3,k})_\beta} (\mathring{\boldsymbol{\chi}}_{t-1})_\beta \mathbb{E}[Z^{h_{t-1}^1} Z^{h_t^1}], \tag{432}$$

$$Z^{\delta h_t^3} = \sum_{k=1}^{m} \mathring{\theta}_{\delta h_t^{3,k} / \delta h_t^3} \mathring{\tilde{h}}_{k,t-1}^2 Z^{\delta h_t^{3,k}} - \mathring{\theta}_{\delta \tilde{h}_t^2 / \delta h_t^3} \delta \mathring{\tilde{h}}_{t,k}^2 Z^{h_t^{3,k}}, \tag{433}$$

$$\delta \mathring{f}_t = \mathring{\theta}'_{W_{t-1}^4} \mathring{\theta}_{\delta h_t^3} \mathbb{E}[Z^{\overline{W_{t-1}^4}} Z^{\delta h_t^3}] - \mathring{\theta}'_{\delta W_t^4} \eta \mathring{\chi}_{t-1} \mathbb{E}[Z^{h_{t-1}^3} Z^{h_t^3}], \tag{434}$$

where G is either softmax or sigmoid and G′ is the derivative, and

$$Z^{W_0^{3,k} \delta h_t^1} = \hat{Z}^{W_0^{3,k} \delta h_t^1} + \dot{Z}^{W_0^{3,k} \delta h_t^1} \tag{435}$$

$$= \hat{Z}^{W_0^{3,k} \delta h_t^1} + \sum_{s=0}^{t-1} Z^{\mathrm{d}h_s^1} \mathbb{E}\left[\frac{\partial Z^{\mathrm{d}h_s^{3,k}}}{\partial \hat{Z}^{(W_0^{3,k})^T \mathrm{d}h_s^{3,k}}} \right] \tag{436}$$

Remark 32 (μP-MoE gives feature learning and stability) *As the infinite-width limit does not change much for the Top-K case. It is easy to see that the μP-MoE values in Table 1 for SGD and Table 2 for Adam hold for Top-K MoEs.*

Appendix G. Expressing softmax and sigmoid over a multi-scalar as a moment operation in TP

We can express the sum of l scalars $\theta_1, \ldots, \theta_k \in \mathbb{R}$ using the **Moment** operations with no vectors ($k = 0$):

$$\phi : \mathbb{R}^0 \times \mathbb{R}^l \to \mathbb{R}, \qquad \phi(-; \theta_1, \ldots, \theta_k) = \sum_{i=1}^{k} \theta_i.$$

Then

$$\frac{1}{n} \sum_{\alpha=1}^{n} \phi(-; \theta_1, \ldots, \theta_l) = \frac{1}{n} \sum_{\alpha=1}^{n} \Big(\sum_{i=1}^{l} \theta_i \Big) = \sum_{i=1}^{l} \theta_i,$$

since the expression inside the sum is constant with respect to α.

Softmax: We can do the same with a softmax. Simply define the function ϕ as the softmax function over the scalars.

$$\phi : \mathbb{R}^0 \times \mathbb{R}^l \to \mathbb{R}, \qquad \phi(-; \theta_1, \ldots, \theta_k) = \frac{\exp \theta_i}{\sum_{i=1}^k \exp \theta_i}.$$

Then

$$\frac{1}{n} \sum_{\alpha=1}^{n} \phi(-; \theta_1, \ldots, \theta_l) = \frac{1}{n} \sum_{\alpha=1}^{n} \frac{\exp \theta_i}{\sum_{i=1}^k \exp \theta_i} = \frac{\exp \theta_i}{\sum_{i=1}^k \exp \theta_i},$$

so each of these operations generates a valid scalar. **Sanity check.** Note that if a sequence of random vectors (multi-scalars in TP language)

$$\theta^{(n)} = (\theta_1^{(n)}, \theta_2^{(n)}, \cdots \theta_l^{(n)}) \longrightarrow \theta^{(*)} = (\theta_1^{(*)}, \theta_2^{(*)}, \cdots \theta_l^{(*)})$$

where the convergence may be in probability, distribution, or almost sure, then the continuous mapping theorem states that if $\phi : \mathbb{R}^l \to \mathbb{R}$ is continuous at $\theta^{(*)}$, then

$$\phi(\theta^{(n)}) = \phi(\theta_1^{(n)}, \theta_2^{(n)}, \cdots \theta_l^{(n)}) \longrightarrow \phi(\theta^{(*)}) = \phi(\theta_1^{(*)}, \theta_2^{(*)}, \cdots \theta_l^{(*)})$$

in the same mode of convergence (probability, distribution, a.s). So continuity at the limit alone suffices for a.s convergence to hold.

TP requires that the non-linearities have polynomially bounded derivatives which are of course much stronger and clearly satisfied by the softmax.

Sigmoid: We can simply express the sigmoid of a scalar with the function ϕ defined as

$$\phi : \mathbb{R}^0 \times \mathbb{R} \to \mathbb{R}, \qquad \phi(-; \theta_1) = \frac{1}{1 + \exp(-\theta_1)}.$$

Then

$$\frac{1}{n} \sum_{\alpha=1}^{n} \phi(-; \theta_1) = \frac{1}{1 + \exp(-\theta_1)}.$$

Appendix H. Additional Experiments and Details

H.1. Details for Figures

Figures in the main paper: All figures in the main paper are created by training for $t = 20$ steps on Fashion-MNIST. The network used is the same as Theorem 1 but with non-linear experts with ReLU activation. The models contained 4 experts, and $K = 2$ were executed in the Top-K case. Hyperparameters of the model such as multiplier (for μP-MoE) and learning rate were tuned so that the figures showed effective learning. The router and function output are zero-initialised to show limiting behaviour (Yang et al., 2021).

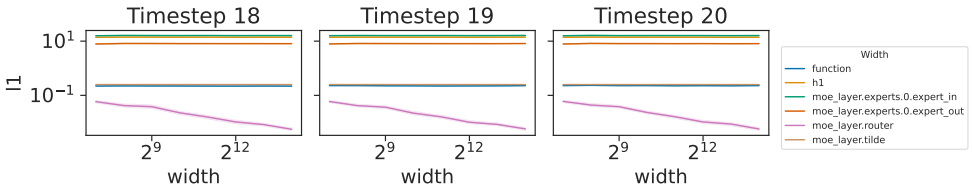


Figure 3: Coordinate check for Softmax-Soft MoE. Shows that the router feature scales $n^{-0.5}$ as predicted in Theorem 25. Tilde signifies the router probability.

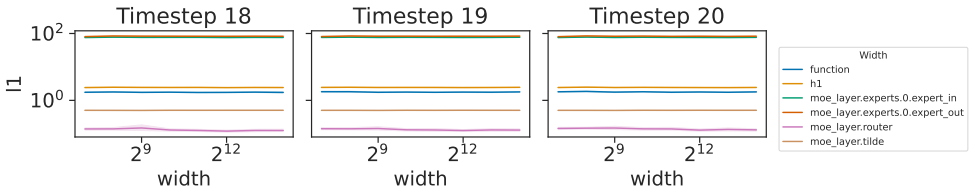


Figure 4: Coordinate check for Sigmoid-Soft MoE. Shows that the router feature learns in comparison with Softmax gating. Tilde signifies the router probability.

HP Transfer in Appendix H.4: HP transfer of learning rate in Appendix H.4 was carried out using a Mixtral style transformer language model (Jiang et al., 2024) trained on Wikitext-103-v1 for 1 epoch. The model was made smaller due to compute constraints. The model contained three transformer layers with feedforward width of 4 times the width dimension. A default load balancing loss coefficient of 0.01 was fixed. Hyperparameters such as the multipliers (for output, router, inputs) and learning rate were randomly sampled 100 times and were tuned based on the validation loss. The model consisted of 8 experts and 2 experts were executed per token. QK Layernorm was implemented as it has been shown to be more stable for HP transfer (Haas et al., 2025).

Compute Resources: The experiments for this paper (including the Mixtral run) were run on a single H100 GPU.

H.2. Coordinate Checks for Soft Routing

We show additional coordinate checks in for sigmoid soft in Figure 4 and softmax soft in Figure 3.

H.3. Coordinate Checks for Top-K Routing

We show additional coordinate checks for sigmoid Top-K in Figure 6 and Figure 5. We additionally verify that μP-MoE scaling (Figure 8) allows for feature learning in Mixtral style transformer MoEs compare to SP-MoE (Figure 7).

H.4. Additional HP Transfer Experiments

We show in Algorithm 1 how to transfer hyperparameters conditional on noise terms/bias.

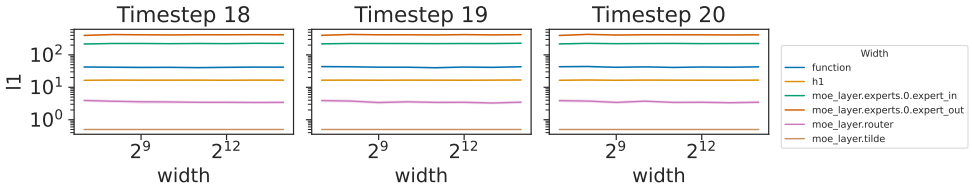


Figure 5: Coordinate check for Softmax-Top-K MoE with 2/4 active experts per forward pass. Shows that in comparison to the soft routing case, the Top-K operation can allow for expert specialisation and hence route feature learning. Tilde signifies the router probability.

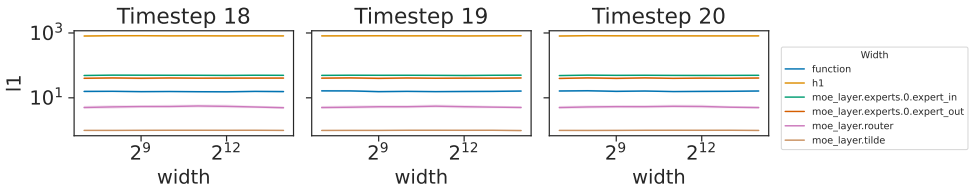


Figure 6: Coordinate check for Sigmoid-Top-K MoE with 2/4 active experts per forward pass. Shows that like in the soft routing case, the router exhibits feature learning. Tilde signifies the router probability.

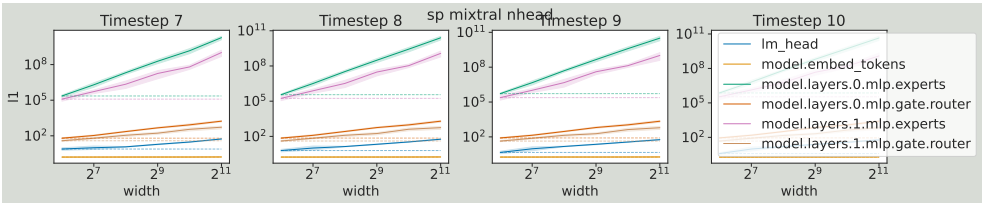


Figure 7: Coordinate check for Top-K Mixtral MoE with SP. Shows that the network diverges with increasing width.

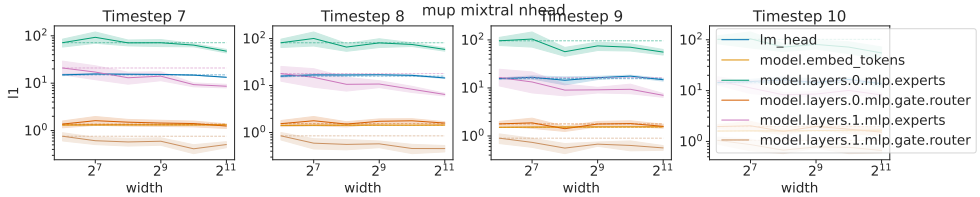
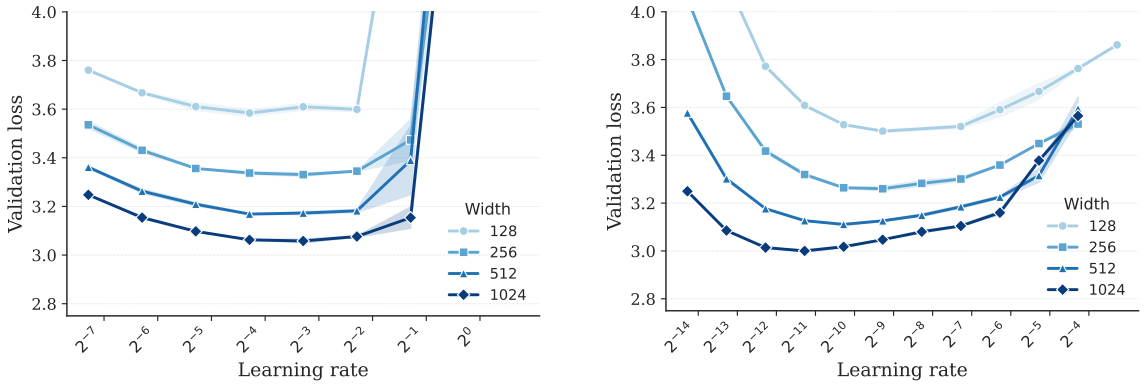


Figure 8: Coordinate check for Top-K Mixtral MoE with μP-MoE. Shows that the network fearture learns as with increases.

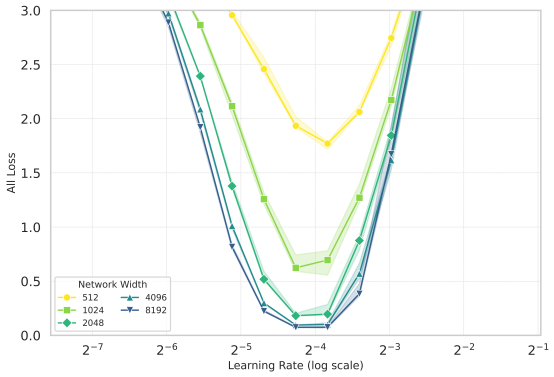


(a) μP-MoE (b) SP-MoE

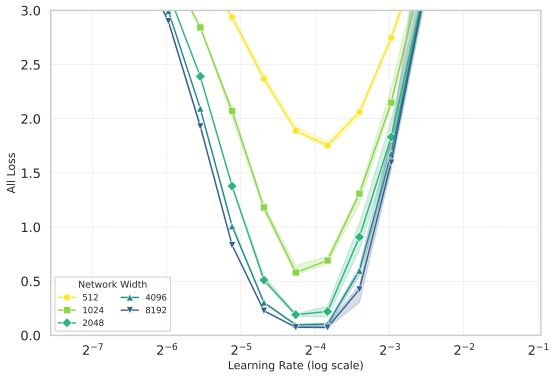
Figure 9: Learning rate transfer for Top-K MoE models parametrised according to (a) μP-MoE and (b) SP-MoE. Hyperparameters are tuned on the smallest model.

Algorithm 1: μP HP-transfer for Top-K MoE with a shared router-bias schedule
Input : base width n_0, target width n, steps T, experts m, Top-K, bias scale σ_{b^2}
Bias schedule (fix once, reuse for all widths)
  Sample $\{b_t^2\}_{t=0}^{T}$ with $b_t^2 \sim \mathcal{N}(0, \sigma_{b^2} I_m)$ using a fixed seed.
end
Training loop at width $w \in \{n_0, n\}$**, given HPs** $\mathcal{H}(\cdot)$
  for $t = 0$ **to** T **do**
    $h_t^2 \leftarrow W_t^2 h_t^1 + b_t^2 \quad I_t \leftarrow \mathrm{TopK}(h_t^2)$ Route to experts in I_t and update parameters.
  end
end
Tune at width n_0
  Run the training loop with μP-heuristic scaling $\Rightarrow$ tuned HPs $\mathcal{H}(n_0)$.
end
Transfer to width n
  Initialise width-n model with μP-heuristic and $\mathcal{H}(n)$; run the same training loop.
end

We carry out additional HP transfer with a similar network to Theorem 1, but with non-linear experts and Top-2 out of 4 expert activation. In Figure 10 we show HP transfer with and without the bias term on tinyimagenet. In Figure 11 we show HP transfer with and without the bias term on emnist.

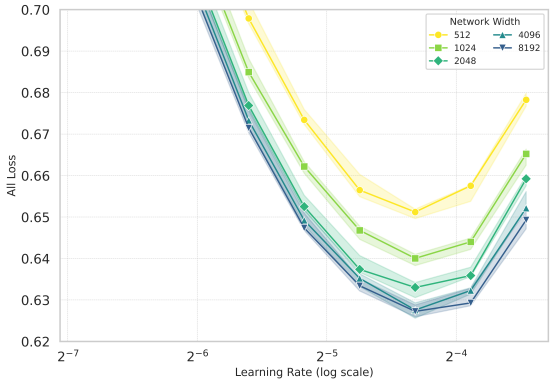


(a) With Bias

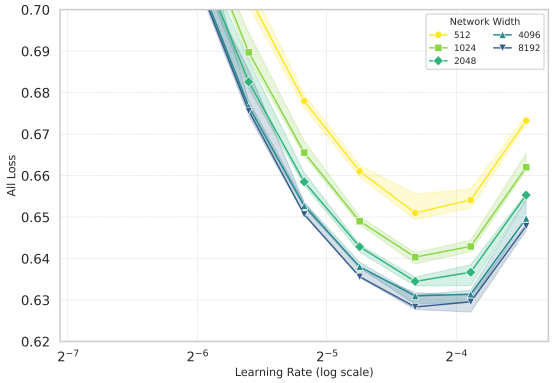


(b) Without Bias

Figure 10



(a) With Bias



(b) Without Bias

Figure 11