ELBOW-BASED MOE ROUTING: A TRAINING-FREE INFERENCE TIME PLUGIN FOR EXPERT SELECTION

Robin Pan[1] * **Raymond Liu**[1†] **Daniel Fang**[1 †] **Adelina Andrei**[2] **Rosa Wu**[1]
[1] Harvard John A. Paulson School of Engineering and Applied Sciences, Harvard University
[2]Department of Mathematics, Harvard University

ABSTRACT

Mixture-of-Experts (MoE) models enable model scaling while maintaining low inference-time compute by activating only a subset of experts per token. However, conventional routing relies on a fixed top-k selection, forcing the model to spend the same compute regardless of how many experts are relevant. We introduce elbow-based routing, a training-free inference-time modification that dynamically adjusts the number of experts on a per-token basis. Our method examines the sorted router probability distribution and identifies an elbow point that separates high- and low-probability experts. We find that most router distributions exhibit clear inflection points suitable for this strategy, and we show both theoretically and empirically that elbow-based routing preserves expert load balance. Experiments on a state-of-the-art MoE model demonstrate an average latency reduction of 5.3% while maintaining accuracy across six benchmarks.

1 INTRODUCTION

Mixture-of-Experts (MoE) architectures scale language models efficiently by activating only a subset of experts per token, enabling models to grow in size without proportionally increasing inference cost (Shazeer et al. (2017)). During both training and inference, MoE model routers activate the top-k experts with the highest logits across all tokens regardless of token. Consequently, some tokens with only a few relevant experts consume unnecessary compute, while tokens requiring more experts may be under-routed.

Prior works in dynamic routing require either training a router, expensive hyperparameter searches, or both. Huang et al. (2024) demonstrates that some inputs benefit from increased expert capacity, and uses auxiliary losses to train a top-p router based on a threshold probability mass p. However, p is a hyperparameter that must be tuned via grid search, which is an expensive process and may be sensitive to biases in the validation data. DynMoE Guo et al. (2025) treats routing as a multi-label classification problem with each expert as a label and allows top-any, but also requires router training. This motivates our central question: *Can MoE models dynamically adjust the number of active experts per token to reduce latency without retraining or additional hyperparameter tuning?*

We introduce elbow-based routing, a simple inference-time plugin that adaptively selects k by detecting the elbow point in the router's sorted probability curve (Fig. 1), which allows us to reduce compute without significantly affecting accuracy. We evaluate our approach on OLMoE, a state-of-the-art MoE model Muennighoff et al. (2025). Our contributions are as follows:

1. Our work introduces the first training-free and inference-time plugin for dynamic routing that does not require altering model weights, architecture, or any additional losses.
2. We conduct an analysis of router probability distributions and find that the vast majority of router probabilities have distinct, sharp elbows and this characteristic is largely independent of input type or router layer.
3. We conduct a comprehensive empirical evaluation across MMLU, ARC-Easy, ARC-Challenge, HellaSwag, PIQA, and WinoGrande, demonstrating an average latency reduction of 5.3% while maintaining accuracy and having minimal effect on load balancing.

*rpan@college.harvard.edu
†These authors contributed equally

2 ELBOW-BASED ROUTING

We consider a mixture-of-experts model with N experts and a learned router that produces logits $\ell(x) \in \mathbb{R}^N$ for each token x. Applying a softmax yields router probabilities $p(x) = \mathrm{softmax}(\ell(x))$, which we sort in descending order as $p_{(1)}(x) \geq \cdots \geq p_{(N)}(x)$. For each token, we compute an elbow index $e(x)$ on the full sorted probability vector $\{p_{(i)}(x)\}_{i=1}^{N}$. The elbow is identified using a Kneedle-style Satopaa et al. (2011) criterion that selects the index of maximum deviation from a reference line after normalizing indices and probabilities to $[0, 1]$ (Algorithm 1). Intuitively, the elbow corresponds to the index at which the rapidly decaying head of the distribution transitions into a relatively flat tail.

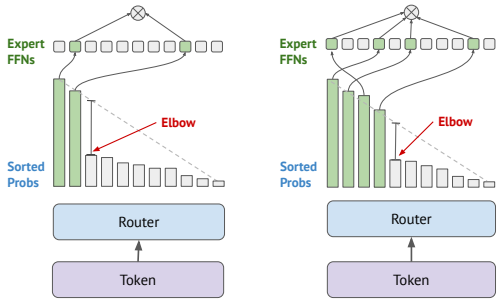

Figure 1: **Elbow-based routing**. Our method adapts to each token's probability curve, which allows us to reduce computation when a clear separation exists between high-confidence and low-confidence experts. Sharper elbows (left) select fewer experts, and more gradual elbows (right) select more experts.

2.1 CAPPED EXPERT SELECTION

The final number of active experts $k(x)$ is $k(x) = \min\big(e(x), K\big)$, where K is the number of experts activated in standard top-K routing. Given that $k(x) \leq K$ by construction, elbow-based routing is a monotone pruning rule on top of an existing router. It preserves expert ordering, never introduces new experts, and requires no retraining or additional routing hyperparameters beyond the model's fixed top-K constraint. Elbow-based routing is fast, with a time complexity of $\mathcal{O}(N \log N)$ where N, the number of experts, is typically on the order of 10 to 100.

2.2 SIGNAL–NOISE INTERPRETATION

The effectiveness of elbow-based routing can be understood through a signal–noise interpretation of the router logits. An input vector x to a router may be decomposed as $x = x_{\parallel} + x_{\perp}$, where $x_{\parallel}$ denotes components aligned with routing-relevant directions, and $x_{\perp}$ captures residual variation orthogonal to these directions. Relevant 'signal' experts aligned with $x_{\parallel}$ receive large, structured logits that form the head of the sorted routing distribution, while less relevant 'noise' experts influenced mainly by $x_{\perp}$ exhibit small, unstructured variations, yielding a flat tail.

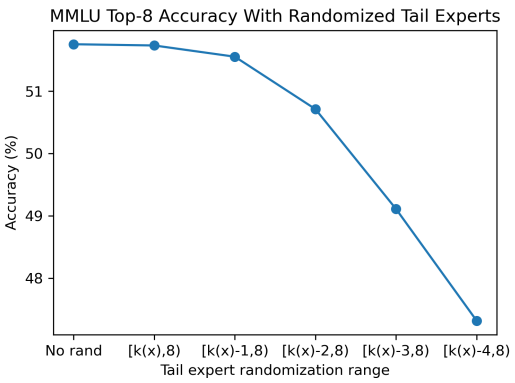

Figure 2: **Tail randomization confirm signal-noise interpretation.** Randomly exchanging tail experts with indices in $[k(x), 8)$ with experts in $[k(x), 64)$ does not affect accuracy. Accuracy drops when randomizing experts in $[0, k(x))$.

Under this view, the sorted router distribution naturally exhibits a transition between a small set of high-confidence experts and a diffuse tail. The elbow identifies this transition point, providing a token-specific estimate of how many experts are meaningfully engaged by the router. When such a transition is pronounced, the elbow corresponds to a sharp change in slope, which is reflected geometrically by a large elbow angle. In Section 2.3 we quantify this effect by measuring elbow angles across layers and tokens, and we show that the vast majority of routing distributions exhibit sharp elbows, consistent with this signal–noise interpretation.

We empirically confirm this interpretation by evaluating model accuracy when randomizing post-elbow experts. Given a router probability curve with $k(x) < 8$, we replace post-elbow experts

(ranks $k(x)$ to 8) with randomly chosen experts of rank $\geq k(x)$. As shown in Fig. 2, this tail randomization matches the baseline top-8 accuracy with no randomization on MMLU. However, accuracy decreases when replacing pre-elbow experts (ranks 1 to $k(x)$) in the same way. This confirms that the elbow marks a practical signal–noise boundary for routing.

2.3 Elbow Analysis and Characterization

We analyzed the 'elbow-ness' of router probabilities from OlMoE routers across multiple benchmarks. To characterize 'elbow-ness' we calculate an elbow angle θ between the first normalized point $(0, 0)$, the elbow point (x'_e, p'_e), and the last normalized point $(1, 1)$.

$$\theta = \arccos \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_1\|\|\mathbf{v}_2\|}, \text{where } v_1 = \begin{pmatrix} -x'_e \\ -p'_e \end{pmatrix}, \quad v_2 = \begin{pmatrix} 1 - x'_e \\ 1 - p'_e \end{pmatrix}.$$

We analyzed over 2 million router probability curves that were produced using samples from MMLU, ARC-Easy, and ARC-Challenge and found that 99.7% of them possess clear, sharp elbows ($\leq 135°$) (Fig. 3a).

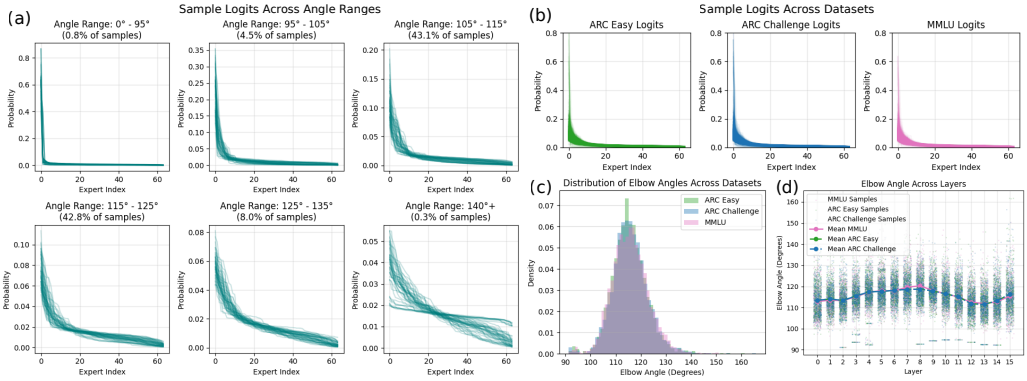

Figure 3: **Analysis of router probability distributions**. (a) Elbow angle distribution showing 99.7% of cases exhibit sharp elbows ($\leq 135°$). (b) Mean sorted router probabilities across three benchmark datasets demonstrate consistent router logit patterns. (c) Elbow angle distributions by dataset show minimal variation across datasets. (d) Mean elbow angles remain stable across all 16 router layers for all three datasets, indicating that elbow structure is a robust, layer-invariant property.

Separating these probabilities by dataset, we see that router probabilities and elbow angles of these probabilities are nearly identical across datasets (Fig. 3b-c). Additionally, mean elbow angles are consistent across router layers and across datasets (Fig. 3d). Because the MMLU dataset is categorized by subject, we analyzed router elbow angles and index by subject and found very similar elbow angle and elbow index distributions across all subjects (Supplementary Fig. 5).

2.4 Load-Balance Guarantees

For each token, elbow-based routing selects a subset of the experts chosen by standard top-K routing, reducing the number of active experts without introducing new paths or changing relative ordering. In Appendix A.3, we formalize the effect of this pruning on expert load and show that the induced changes to the normalized per-expert utilization distribution are bounded.

Let $L_i^{(\text{top-}K)}$ and $L_i^{(\text{elb})}$ denote the expected load of expert i under top-K and elbow-based routing, respectively, and let $\delta = 1 - \mathbb{E}[k(x)]/K$ denote the expected fraction of pruned expert assignments. We prove that the normalized expert utilization distributions $q^{(\text{top-}K)}$ and $q^{(\text{elb})}$ satisfy

$$\left\| q^{(\text{elb})} - q^{(\text{top-}K)} \right\|_1 \leq \frac{2\delta}{1 - \delta}.$$

Table 1: Layer-wise Load Balancing Analysis

Layer	$\ell 1$ norm	% Change in Top-1 Share	% Change in CV
1	0.049	-1.81	-5.80
2	0.055	-3.71	-8.83
3	0.043	-4.45	-4.73
4	0.042	-2.53	-3.91
5	0.047	-3.85	-4.29
6	0.030	-2.48	-2.16
7	0.020	-2.31	-2.04
8	0.019	-2.08	-1.79
9	0.033	-2.61	-1.74
10	0.023	-0.48	-0.81
11	0.020	+0.38	+0.05
12	0.032	-1.71	-1.51
13	0.036	-1.01	-1.53
14	0.048	-2.95	-3.20
15	0.032	-1.60	-1.38
16	0.020	-1.42	-1.55
Mean	**0.034**	**-2.16**	**-2.83**

We measure expert load usage averaged across six benchmarks: MMLU, ARC-Easy, ARC-Challenge, HellaSwag, PIQA, and WinoGrande. Across all benchmarks, elbow-based routing consistently selects around $\mathbb{E}[k(x)] \approx 7.6$ experts per token across all layers as seen in Table 2, which corresponds to an average pruning rate $\delta = 1 - \frac{\mathbb{E}[k(x)]}{K} = 0.05$. Empirically, we find that changes to load balance are much smaller than the worst-case theoretical bounds at this δ. For each layer, we calculate the ℓ_1 distance between the normalized expert utilization distributions under top-K and elbow-based-K routing, averaged across all six benchmarks. Table 1 reports a mean change of 0.034 in the ℓ_1 norm, which suggests that only 1.7% of the total normalized utilization mass is redistributed across experts. We calculate $\%$ Change in Top-1 Share as $\frac{\max(q^{(top-K)}) - \max(q^{(elb)})}{\max(q^{(top-K)})} \cdot 100$ and find the maximum expert load increases by 2.16% on average. $\%$ Change in CV is calculated as $\frac{CV^{top-K} - CV^{elb}}{CV^{top-K}} \cdot 100$, and CV increases by 2.83%. For all three metrics, empirical evidence is significantly smaller than theoretical guarantees in A.3.

3 Experimental Validation

Table 2: Evaluation Results

Dataset	Model	Acc (%)	k-mean	FLOPs	Latency (ms)
ARC-Easy	Base OLMoE	77.82	8	5.37E8	11.203
	Elbow-8	**78.24**	7.623	**4.86E8**	**10.497**
ARC-Challenge	Base OLMoE	61.86	8	5.37E8	11.531
	Elbow-8	**62.29**	7.634	**4.88E8**	**10.974**
MMLU	Base OLMoE	51.74	8	5.37E8	13.287
	Elbow-8	**51.75**	7.613	**4.91E8**	**12.571**
HellaSwag	Base OLMoE	**47.02**	8	5.37E8	14.630
	Elbow-8	46.86	7.655	**4.92E8**	**13.960**
PIQA	Base OLMoE	**73.18**	8	5.37E8	13.398
	Elbow-8	72.85	7.589	**4.85E8**	**12.944**
WinoGrande	Base OLMoE	**50.36**	8	5.37E8	9.659
	Elbow-8	50.28	7.576	**4.81E8**	**8.948**

We evaluate elbow-based routing on OLMoE under baseline top-8 routing and elbow-based routing capped at $k \leq 8$ (Elbow-8) across six standard multiple-choice benchmarks: MMLU, ARC-Easy, ARC-Challenge, HellaSwag, PIQA, and WinoGrande. Table 2 reports accuracy, the mean number of active experts per token (k-mean), estimated FLOPs, and wall-clock latency per forward pass. See A.4 for additional implementation details. Across datasets, Elbow-8 matches baseline accuracy (within ≤ 0.33 points) while reducing average latency by 5.3% on our setup and decreasing estimated compute, with an overall k-mean of 7.615. These results indicate that elbow-based routing provides a lightweight inference-time efficiency improvement without retraining or changes to model weights.

4 Conclusion

We introduced elbow-based routing, a training-free inference-time plugin that selects a token-specific number of experts by detecting an elbow in the router's sorted probability curve. On OLMoE with a cap of $K = 8$, this simple monotone pruning rule reduces average latency by 5.3% across six benchmarks while preserving accuracy, without retraining, architectural changes, or additional hyperparameters. Across over 2M routing decisions, we observe that router distributions exhibit sharp and consistent elbows across layers and datasets, supporting a head–tail structure in which a small set of experts carries most routing signal. We further provide load-balance guarantees and empirical evidence that the induced changes in expert utilization are small, suggesting that routers trained with fixed top-K often contain enough structure to enable lightweight test-time efficiency improvements and serve as a diagnostic lens for routing behavior.

REFERENCES

Yongxin Guo, Zhenglin Cheng, Xiaoying Tang, Zhaopeng Tu, and Tao Lin. Dynamic mixture of experts: An auto-tuning approach for efficient transformer models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. Harder tasks need more experts: Dynamic routing in moe models. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2024.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Evan Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, and et al. Olmoe: Open mixture-of-experts language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *Proceedings of the 2011 31st International Conference on Distributed Computing Systems Workshops*, ICDCSW '11, pp. 166–171, USA, 2011. IEEE Computer Society.

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, 2017. URL https://arxiv.org/abs/1701.06538.

A APPENDIX

A.1 ALGORITHM FOR DETERMINING ELBOWS

Below is an approximation of the Kneedle algorithm used to determine elbow indices:

Algorithm 1 Elbow Detection via Kneedle Approximation

Require: Router logits $l \in \mathbb{R}^N$ over N experts

$p \leftarrow \mathrm{softmax}(l)$

$p \leftarrow \mathrm{sort}_{\downarrow}(p)$ ▷ sort probabilities in descending order

$p_{\min} \leftarrow \min(p); \quad p_{\max} \leftarrow \max(p)$

for $i = 0$ to $N - 1$ **do** ▷ normalize to unit square and compute deviation

$$p'_i \leftarrow \frac{p_i - p_{\min}}{p_{\max} - p_{\min} + \epsilon}$$

$$x'_i \leftarrow \frac{i}{N-1}$$

$$D_i \leftarrow p'_i - x'_i$$

end for

$e \leftarrow \arg\max_{i \in \{0,\ldots,N-1\}} D_i$

return $k \leftarrow e + 1$

A.2 Additional Elbow Characterization

A.2.1 Sample Elbow Logits

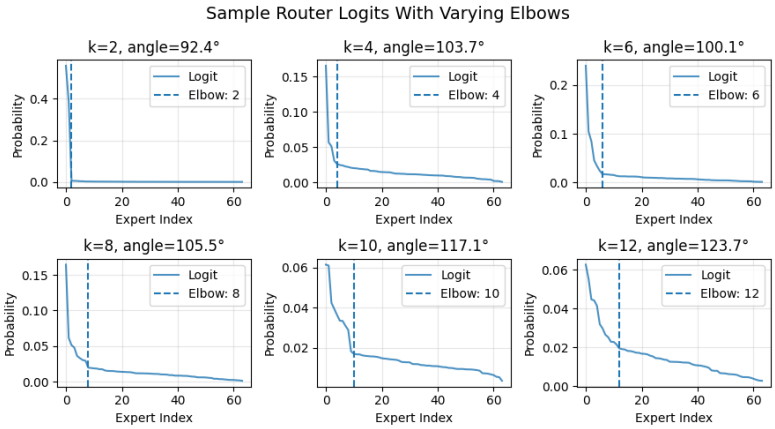

Figure 4: Plots of sorted router probabilities with elbow indexes at $k = 2, 4, 6, 8, 10, 12$ and their respective elbow angles. Our implementation of the Kneedle algorithm effectively identifies the 'elbow' of sorted router probability curves.

A.2.2 Elbow Analysis by MMLU Subject

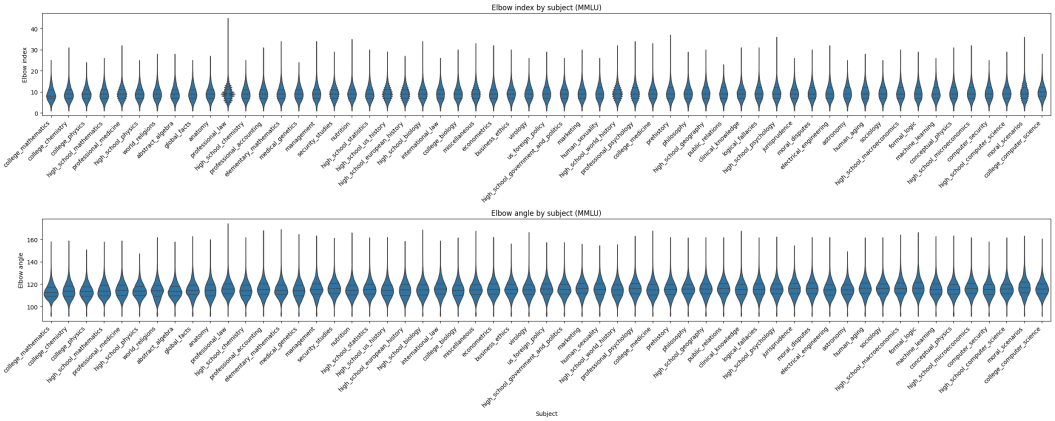

Figure 5: **Elbow angle and index vary little across subjects.** We plot the distribution of elbow index and elbow angle by subject above and conduct a one-way ANOVA test and find $p < 0.001, \eta^2 = 0.02$ for elbow indices and $p < 0.001, \eta^2 = 0.01$ with $N = 1,704,247$. This indicates that a very small fraction of variance is attributed to subject.

A.2.3 Elbow Angle and Elbow Index Correlation

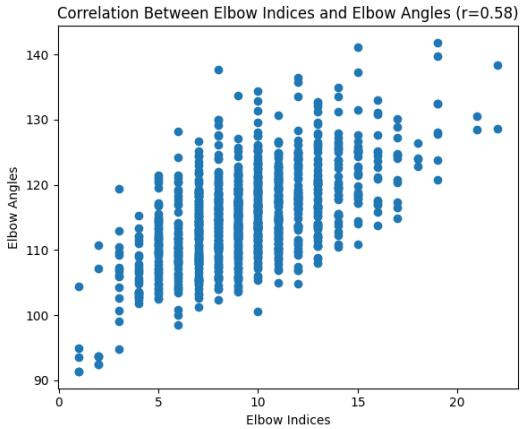

Figure 6: **Elbow indexes are correlated with elbow angle.** Elbow angle increases with elbow index, indicating that tokens requiring more active experts tend to exhibit less sharp transitions in the sorted router distribution (Pearson $r = 0.58$)

A.3 Additional Load-Balance Guarantees Details

We analyze the impact of elbow-K routing on expert load balance relative to standard top-K routing. Elbow-K routing acts as a monotone pruning of top-K routing: for each token, it selects a subset of the experts chosen by top-K. Under this structure, we show that changes in distribution of the load balance are bounded and minimal.

Notation. Let $\mathcal{S}_{\text{top-}K}(x)$ and $\mathcal{S}_{\text{elb-}K}(x)$ denote the experts selected by standard top-K routing and elbow-K routing, respectively, for token x. Let $k(x) = |\mathcal{S}_{\text{elb-}K}(x)|$ denote the number of experts selected by elbow-K routing for token x, and define the expected fraction of pruned assignments as

$$\delta = 1 - \frac{\mathbb{E}[k(x)]}{K}.$$

For a routing rule $r \in \{\text{top-}K, \text{ elb-}K\}$, define the expected per-expert load

$$L_i^{(r)} = \mathbb{E}_{x\sim\mathcal{D}}[\mathbf{1}\{i \in \mathcal{S}_r(x)\}],$$

and the corresponding normalized utilization distribution

$$q_i^{(r)} = \frac{L_i^{(r)}}{\sum_{j=1}^N L_j^{(r)}}, \qquad q^{(r)} \in \Delta^{N-1}.$$

Theorem 1 (Distribution Changes in Expert Utilization under Elbow-K Routing)**.** *For any MoE layer, let $q^{(\text{top-}K)}$ and $q^{(\text{elb-}K)}$ denote the normalized per-expert utilization distributions under top-K and elbow-K routing, respectively. Then, we claim that*

$$\left\|q^{(\text{elb-}K)} - q^{(\text{top-}K)}\right\|_1 \leq \frac{2\delta}{1-\delta}.$$

Proof. By construction, $\mathcal{S}_{\text{elb-}K}(x) \subseteq \mathcal{S}_{\text{top-}K}(x)$ for all tokens x. Therefore, working at the vector level, define the removed-mass vector $R = L^{(\text{top-}K)} - L^{(\text{elb-}K)} \in \mathbb{R}_+^N$.

Since top-K selects exactly K experts per token and elbow-K selects $k(x)$ experts, we have

$$\sum_{i=1}^N L_i^{(\text{top-}K)} = K, \qquad \sum_{i=1}^N L_i^{(\text{elb-}K)} = \mathbb{E}[k(x)] = (1-\delta)K, \qquad \sum_{i=1}^N R_i = \delta K.$$

Passing to the corresponding normalized removed-mass distribution we have

$$r = \frac{R}{\delta K}, \qquad r \in \Delta^{N-1},$$

which combined with the definitions of the normalized utilization distributions gives

$$q^{(\text{elb-}K)} = \frac{L^{(\text{elb-}K)}}{(1-\delta)K} = \frac{L^{(\text{top-}K)} - R}{(1-\delta)K} = \frac{q^{(\text{top-}K)}K - R}{(1-\delta)K} = \frac{q^{(\text{top-}K)} - r\delta}{1-\delta} \implies$$

$$\implies q^{(\text{elb-}K)} - q^{(\text{top-}K)} = \frac{\delta}{1-\delta}\big(q^{(\text{top-}K)} - r\big).$$

Since $q^{(\text{top-}K)}$ and r are probability distributions, we have that worst-case $\|q^{(\text{top-}K)} - r\|_1 \leq \sum_{i=1}^{N} |q_i^{(\text{top-}K)} - r_i| \leq \sum_{i=1}^{N} q_i^{(\text{top-}K)} + \sum_{i=1}^{N} r_i \leq 2$, which gives

$$\big\|q^{(\text{elb-}K)} - q^{(\text{top-}K)}\big\|_1 \leq \frac{2\delta}{1-\delta}.$$

□

Corollary 1. The above bound implies that the change in the load of the most-utilized expert is

$$\big|\|q^{(\text{elb}-K)}\|_\infty - \|q^{(\text{top}-K)}\|_\infty\big| \leq \|q^{(\text{elb}-K)} - q^{(\text{top}-K)}\|_1 \leq \frac{2\delta}{1-\delta}.$$

Thus, elbow-K routing cannot substantially increase the load of the most-utilized expert, which is often a determinant of inference-time latency.

Corollary 2. The above bound implies that the coefficient of variation $\text{CV}^2(q) = N\|q\|_2^2 - 1$ change is

$$\Big|\text{CV}^2(q^{(\text{elb}-K)}) - \text{CV}^2(q^{(\text{top-K})})\Big| = N\,\Big|\,\|q^{\,(\text{elb}-K)}\|_2^2 - \|q^{\,(\text{top}-K)}\|_2^2\Big| \leq$$

$$= \frac{1}{(1-\delta)^2}\Big|(2\delta - \delta^2)\|q^{(\text{top-}K)}\|_2^2 - 2\delta\langle q^{(\text{top-}K)}, r\rangle + \delta^2\|r\|_2^2\Big| \leq \frac{2N\delta}{(1-\delta)^2},$$

where we used that $0 \leq \|q^{(\text{top-}K)}\|_2^2 \leq 1$, $0 \leq \|r\|_2^2 \leq 1$, and $0 \leq \langle q^{(\text{top-}K)}, r\rangle \leq 1$ since $q^{(\text{top-}K)}, r \in \Delta^{N-1}$ to get $0 \leq (2\delta - \delta^2)\|q^{(\text{top-}K)}\|_2^2 + \delta^2\|r\|_2^2 \leq 2\delta$.

The alignment term $\langle q^{(\text{top-}K)}, r\rangle$ captures whether pruning removes assignments primarily from heavily utilized experts as opposed to low-utilized ones. Pruning aligned with high-load experts tends to reduce CV^2, while the opposite pattern can increase it.

Discussion. As we have mentioned in Section 2.4, empirically we observed across the six benchmarks that $\delta = 1 - \frac{\mathbb{E}[k(x)]}{K} = 0.05$.

Using this value, Theorem 1 guarantees that the normalized expert utilization distribution changes by at most

$$\|q^{(\text{elb-K})} - q^{(\text{top-K})}\|_1 \;\leq\; \frac{2\delta}{1-\delta} \;\approx\; 0.10.$$

This tells us that after pruning, no more than worst-case 5% of the total normalized expert utilization mass can be redistributed across experts. Empirically, the measured ℓ_1 changes in normalized utilization are around 0.05 or less across layers (Table 1), indicating that the relative expert usage profiles are preserved even more closely than required by the guarantee.

Similarly, Corollary 2 bounds the change in squared coefficient of variation as

$$\big|CV^2(q^{(\text{elb-K})}) - CV^2(q^{(\text{top-K})})\big| \;\leq\; \frac{2N\delta}{(1-\delta)^2} \;\approx\; 7.$$

This is a rather loose bound. Empirically, the observed changes in CV vary from 1 to 8% across layers (Table 1), indicating that elbow-based routing preserves not only the total utilization distribution but also its variability structure significantly more closely than required by the guarantee.

A.4 IMPLEMENTATION DETAILS

We implemented elbow-based routing on OLMoE-1B-7B-0924-Instruct (Muennighoff et al. (2025)), a sparse MoE model with 64 experts and default top-8 routing. We select this model for its state-of-the-art performance, as well as its small model size given computational resource constraints. All evaluations were performed with an NVIDIA H200 GPU. We implement elbow-based routing via a runtime monkey patch that replaces `OlmoeSparseMoeBlock.forward` with an instrumented variant, while caching the original method for reversible restoration.

FLOPs are estimated analytically per MoE block as the sum of (i) router cost—one dense linear projection ($2NdE$), softmax ($5NE$), and a top-k/sort term ($NE \log_2 E$)—and (ii) expert MLP cost—up and down projections ($2N\bar{k}dm$) each plus activation ($N\bar{k}m$), where N=batch$\times$seq, d=hidden dim, m=4d, E=experts, and $\bar{k}$ is the observed mean selected experts per token. For dynamic routing we additionally account for elbow detection overhead as per-token sort ($NE \log_2 E$) plus linear-time normalization, subtraction, and argmax terms ($\approx (4 + 1 + 1)NE$).

Latency was measured with wall-clock timing around the MoE block forward pass using `time.perf_counter()`, with an explicit `torch.cuda.synchronize()` immediately before starting and after producing the final output tensor to ensure all queued CUDA kernels completed.

All code is available at `https://github.com/rpan188/elbow-routing`.