Selective Sinkhorn Routing for Improved Sparse Mixture of Experts

Duc Anh Nguyen [1] **Huu Binh Ta** [2†] **Duc Nhuan Le** [1‡] **Tan Minh Nguyen** [1,3] **Toan Tran** [1]

Abstract

Sparse Mixture-of-Experts (SMoE) models are scalable and computationally efficient, enabling large increases in model capacity with limited inference overhead. Existing SMoE methods often depend on auxiliary objectives, such as load-balancing loss and z-loss, or additional trainable components such as noisy gating. While these techniques encourage expert diversity, they can introduce objective misalignment, increase model complexity, or incur substantial training overhead, especially in Sinkhorn-based routing methods. In this paper, we revisit the token-to-expert assignment as an optimal transport problem. We add constraints to ensure balanced expert utilization. We show that even minimal optimal transport-based routing improves SMoE performance without requiring auxiliary balancing losses. Unlike prior approaches, our method derives gating scores directly from the transport map, leading to more balanced and effective token-to-expert assignments. Building on this insight, we introduce Selective Sinkhorn Routing (SSR), a lightweight routing mechanism that replaces complex auxiliary losses with efficient Sinkhorn-based routing while preserving flexible expert selection. Experiments on language modeling and image classification show that SSR improves training efficiency, accuracy, and robustness to input corruption.

[1]Qualcomm AI Research. Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc. [2]University of Virginia, USA [3]Department of Mathematics, National University of Singapore. [†]Work done at Qualcomm AI Research. [‡]Partially done at Ho Chi Minh City University of Science, VNU-HCM, Vietnam. Correspondence to: Duc Anh Nguyen <nguyenducanh909.bkhn@gmail.com>, Toan Tran <toatran@qti.qualcomm.com>.

Published at ICML 2026 Workshop on Resource-Adaptive Foundation Model Inference, AdaptFM, Seoul, South Korea.

1. Introduction

Foundation models have rapidly advanced across language (Vaswani et al., 2017; Brown et al., 2020; Raffel et al., 2020), vision (Dosovitskiy et al., 2020; Liu et al., 2021; Riquelme et al., 2021), and multimodal learning (Lin et al., 2024; Rasheed et al., 2024). A common way to improve their performance is to scale model capacity, but dense scaling substantially increases computational cost and inference latency. Sparse Mixture-of-Experts (SMoE) models (Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2022) provide an efficient alternative by activating only a small subset of specialized experts for each token, thereby increasing capacity while limiting computation.

Conventional SMoE models use Softmax-based routing, assigning each token to the top-k experts based on gating scores. This can cause routing collapse, where only a small subset of experts is frequently selected while others remain underutilized (Chi et al., 2022). Prior work mitigates this issue with auxiliary load-balancing losses (Fedus et al., 2022; Lepikhin et al., 2020), but such losses may introduce training instability (Zoph et al., 2022). Alternatively, OT-based routing methods use Sinkhorn algorithms to obtain balanced expert assignments without auxiliary losses (Liu et al., 2022), though they may reduce routing flexibility because the gating matrix is not directly optimized through gradient-based learning (Liu et al., 2024).

In contrast to prior Sinkhorn-based approaches (Kool et al., 2021; Clark et al., 2022; Liu et al., 2024), which use the transport map only to select the top-k experts, we also use its values to assign weights to each selected expert. We derive routing weights from transport-map values rather than from the conventional gating-score matrix, leveraging Sinkhorn's built-in expert balancing in SMoE. This design improves expert balancing, supported by both theory and empirical results. We introduce Selective Sinkhorn Routing (SSR), a lightweight routing mechanism that replaces auxiliary losses with minimal Sinkhorn-based optimization. Applying SSR to only 0.1%–1% of training steps per epoch yields faster convergence, higher accuracy, and greater robustness to input corruption. We also show that although Sinkhorn routing and the noise-addition trick improve training performance, they degrade inference performance.

This work makes the following principal contributions:

- We propose a routing framework for SMoE models that integrates entropy-regularized optimal transport with stochastic noise injection to promote balanced expert utilization and improve training stability.
- We provide theoretical results showing that Sinkhorn-based routing and noise injection aid training by encouraging exploration and expert balancing, but should be disabled at inference to ensure consistent, deterministic routing.
- We conduct extensive evaluations on language modeling and vision tasks, demonstrating that the proposed approach outperforms existing methods.

2. Related Work

Sparse Mixture of Experts Sparse Mixture-of-Experts (Shazeer et al., 2017; Du et al., 2022; Fedus et al., 2022) has become a core backbone in deep learning, increasing model capacity while preserving computational efficiency. Compared to densely activated models, SMoE can improve performance across tasks without excessive compute (Lepikhin et al., 2020; Zhou et al., 2022). By activating only a subset of experts per input, SMoE substantially increases parameter count without increasing FLOPs per example. This sparsity supports large-scale training that remain cost-effective at inference, making SMoE attractive for applications such as language modeling and machine translation.

Routing in SMoE In SMoE architectures, the router assigns tokens to experts. A common approach uses a gating network with a softmax to produce a distribution over experts for each token (Shazeer et al., 2017). To balance token load, load-balancing losses (Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2022) or injected noise (Shazeer et al., 2017) are often used. Despite improving utilization, they introduce extra hyperparameters and training complexity. Sinkhorn-based routing instead optimizes token–expert assignments via the Sinkhorn algorithm (Clark et al., 2022; Cai et al., 2024). ST-MoE (Zoph et al., 2022) adds a z-loss penalizing overly large gating logits for numerical stability.

3. Preliminaries

This section provides the foundation on SMoE models, followed by the background of the Sinkhorn-Knopp Algorithm.

3.1. Sparse Mixture of Experts

An SMoE model comprises multiple MoE blocks, each containing a set of experts. Within each block, experts process different aspects of the input, and their outputs are combined to form the block output. Let $\mathbf{X} \in \mathbb{R}^{m\times d}$ denote the matrix of m input token embeddings, and let $\mathbf{W}_g \in \mathbb{R}^{n\times d}$ denote the gating weight matrix for n experts. The gating score matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$ is computed as $\mathbf{S} = \mathbf{X}\mathbf{W}_g^\top$. Each entry $s_{i,j}$ in $\mathbf{S}$ is a compatibility score between token i and expert j, indicating how suitable expert j is for processing token i. Larger scores imply a stronger preference for routing token i to expert j. For each token $i \in \{1,\dots,m\}$, we select the top-k experts with the highest scores, denoted by the index set $\mathcal{T}_i \subseteq \{1,\dots,n\}$ with $|\mathcal{T}_i| = k$. The routing weight from token i to expert j is defined as

$$w_{i,j} = \begin{cases} \dfrac{\exp(s_{i,j})}{\sum_{j'\in\mathcal{T}_i}\exp(s_{i,j'})}, & \text{if } j \in \mathcal{T}_i, \\ 0, & \text{otherwise.} \end{cases}$$

Each expert j is represented by a feedforward network $f_j : \mathbb{R}^d \to \mathbb{R}^d$. Token i's output is aggregated as $\mathbf{y}_i = \sum_{j\in\mathcal{T}_i} w_{i,j}\, f_j(\mathbf{x}_i)$.

3.2. Sinkhorn-Knopp Algorithm

Given two probability vectors $r \in \mathbb{R}^m, s \in \mathbb{R}^n$ and a cost matrix $\mathbf{C} \in \mathbb{R}^{m\times n}$, the entropy-regularized optimal transport (OT) problem (with entropic regularization parameter $\xi > 0$) seeks a transport plan $\hat{\mathbf{\Pi}} \in \mathbb{R}^{m\times n}$ that minimizes:

$$\hat{\mathbf{\Pi}} = \underset{\mathbf{\Pi}\in\mathbb{R}^{m\times n}}{\arg\min} \langle \mathbf{\Pi}, \mathbf{C}\rangle + \xi\langle \mathbf{\Pi}, \log \mathbf{\Pi}\rangle, \tag{1}$$

$$\text{subject to: } \mathbf{\Pi} > 0,\ \ \mathbf{\Pi}\mathbf{1}_n = r,\ \ \mathbf{\Pi}^\top \mathbf{1}_m = s. \tag{2}$$

This can be solved efficiently by the Sinkhorn-Knopp algorithm (Cuturi, 2013) (see Algorithm 1), which alternates between scaling rows and columns of a kernel matrix $\mathbf{K} = \exp(-\mathbf{C}/\xi)$ to match the marginals r and s.

Algorithm 1 Iterative Sinkhorn-Knopp Algorithm

Input: Cost matrix $\mathbf{C} \in \mathbb{R}^{m\times n}$, $r \in \mathbb{R}^m, s \in \mathbb{R}^n$, $\xi, \delta > 0$, number of iterations $\eta \in \mathbb{N}^+$.
Output: Optimal matrix $\mathbf{\Pi}^* \in \mathbb{R}^{m\times n}$.
$\mathbf{u}^{(0)} = 1_m, \mathbf{K} = \exp(-\mathbf{C}/\xi)$.
for *each* $k = 1$ *to* η **do**
  $\mathbf{v}^{(k)} = s \oslash \mathbf{K}^\top \mathbf{u}^{(k-1)}$; // Element-wise division
  $\mathbf{u}^{(k)} = r \oslash \mathbf{K}\mathbf{v}^{(k)}$; // Element-wise division
  if $\|\mathbf{\Pi}\mathbf{1}_n - r\| < \delta$ *and* $\|\mathbf{\Pi}^\top \mathbf{1}_m - s\| < \delta$ **then**
    break;
end
$\hat{\mathbf{\Pi}} = \mathrm{diag}(\mathbf{u}^{(k)})\, \mathbf{K}\, \mathrm{diag}(\mathbf{v}^{(k)})$.

4. Methodology

In this section, we propose a novel method to balance token allocation in SMoE models via an optimal-transport-based token-to-expert assignment mechanism. A key challenge is that it requires reasonably good gating scores to produce meaningful transport maps. To address this, we concurrently employ standard Softmax gating during training to update the gating weight matrices. This joint training mechanism lets transport-based routing leverage updated gating scores while preserving its assignment process. Unless otherwise stated, all notation follows Section 3. Proofs are provided in the Appendix.

4.1. Token-to-expert Assignment as an Entropy-regularized Optimal Transport Problem

To achieve effective load balancing, we formulate token-to-expert assignment as an entropy-regularized maximum-cost optimal transport (OT) problem that assigns tokens to experts by maximizing overall compatibility. We propose two approaches to construct the transport cost matrix $\mathbf{C} \in \mathbb{R}^{m\times n}$ from the gating score matrix $\mathbf{S}$: (1) **Raw gating scores (linear cost)**: $\mathbf{C} = \mathbf{S}$; (2) **Normalized scores (softmax cost)**: apply a row-wise softmax to $\mathbf{S}$, i.e., $\mathbf{C}_{i,:} = \mathrm{softmax}(\mathbf{S}_{i,:})$. The softmax cost prevents kernel entries from becoming excessively large, since unbounded values can cause exponential growth and numerical overflow during Sinkhorn updates.

Building on this motivation, we further impose balancing constraints across experts, leading to an entropy-regularized maximum-cost OT problem. This problem can be efficiently solved using the Sinkhorn algorithm, formulated as:

$$\hat{\mathbf{\Pi}} = \underset{\mathbf{\Pi}\in\mathbb{R}^{m\times n}}{\arg\max} \left\{ \langle \mathbf{\Pi}, \mathbf{C}\rangle - \xi \langle \mathbf{\Pi}, \log \mathbf{\Pi}\rangle \right\}, \tag{3}$$

subject to

$$\text{(C1) } \mathbf{\Pi} > 0, \quad \text{(C2) } \mathbf{\Pi}\mathbf{1}_n = \mathbf{1}_m, \quad \text{(C3) } \mathbf{\Pi}^\top \mathbf{1}_m = \tfrac{m}{n}\mathbf{1}_n. \tag{4}$$

We clarify those constraints as follows. (C1): All routing probabilities must be positive to form well-defined entropy term. (C2): Each token routes its entire mass to experts; each row of $\mathbf{\Pi}$ sums to 1, so all token information is preserved. (C3): Each expert receives the same expected total load; each column of $\mathbf{\Pi}$ sums to m/n.

Unlike existing Sinkhorn Token Choice routers (Kool et al., 2021; Clark et al., 2022), which use the transport plan only for expert selection, we directly use $\hat{\mathbf{\Pi}}$ to compute routing weights. This approach is theoretically supported by Proposition 4.1. Specifically, for each token i, we select the top-k experts with the largest entries in the i-th row of $\hat{\mathbf{\Pi}}$. Let token i be assigned to experts $E_{i_1}, E_{i_2}, \ldots, E_{i_k}$, corresponding to the k largest entries in the i-th row of $\hat{\mathbf{\Pi}}$. The compatibility score between token i and expert E_{i_r} is given by $\hat{\mathbf{\Pi}}_{i,i_r}$, where $r \in \{1, 2, \ldots, k\}$.

Proposition 4.1. *Let $\hat{\mathbf{\Pi}} \in \mathbb{R}^{m\times n}$ be the solution to the entropy-regularized optimal transport problem in Eq. 3. For each token $i \in \{1, \ldots, m\}$, suppose we are allowed to assign it to at most k experts, with routing weights $\alpha_i \in \mathbb{R}^n$ satisfying:*

$$\alpha_{i,j} \geq 0, \quad \mathit{supp}(\alpha_i) \leq k, \quad \sum_{j=1}^{n} \alpha_{i,j} = 1. \tag{5}$$

We select the top-k experts $E_{i_1}, E_{i_2}, \ldots, E_{i_k}$ with the highest transport scores $\hat{\mathbf{\Pi}}_{i,i_1}, \hat{\mathbf{\Pi}}_{i,i_2}, \ldots, \hat{\mathbf{\Pi}}_{i,i_k}$, and set the weights as: $\alpha_{i,i_r} = \frac{\hat{\mathbf{\Pi}}_{i,i_r}}{\sum_{j=1}^{k} \hat{\mathbf{\Pi}}_{i,i_j}}$ for $r = 1, \ldots, k$. Then, α_i is the optimal solution of the optimization problem $\min_{\alpha_i} \mathrm{KL}(\alpha_i \,\|\, \hat{\mathbf{\Pi}}_i)$.

4.2. Selective Sinkhorn Routing for Sparse Mixture of Experts

A key limitation of Sinkhorn-based token-to-expert assignment is that it does not update the gating weight matrix $\mathbf{W}_g$. While this preserves token information during routing, $\mathbf{W}_g$ is decoupled from the computational graph because it is not included in the OT objective. As a result, the gating score matrix $\mathbf{S}$ is not directly optimized for token–expert compatibility, weakening the semantic meaning of the transport map and its role as a compatibility-aware supervision signal. Thus, the OT formulation is meaningful only when $\mathbf{S}$ reliably captures token–expert compatibility.

Proposition 4.2 (Load Balancing Is Valid During Training but Not Inference)**.** *Let $\mathcal{X}$ be the input space and $p(x)$ the data distribution. Let $g : \mathcal{X} \to \mathbb{R}^n$ be a gating function producing expert scores $\mathbf{S}(x) \in \mathbb{R}^n$, and let $f : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ be a routing function based on entropy-regularized optimal transport. For a batch $\{x_1, \ldots, x_m\}$, the transport plan $\mathbf{\Pi} \in \mathbb{R}^{m\times n}$ satisfies:*

$$\textit{(C1) } \mathbf{\Pi} > 0, \quad \textit{(C2) } \mathbf{\Pi}\mathbf{1}_n = \mathbf{1}_m, \quad \textit{(C3) } \mathbf{\Pi}^\top \mathbf{1}_m = \tfrac{m}{n}\mathbf{1}_n.$$

Then:

***(1) (Training)** When the batch $\{x_1, \ldots, x_m\}$ consists of i.i.d. samples from $p(x)$, the average routing approximates the expected routing $\mathbb{E}_{x\sim p(x)}[\pi(x)]$. Constraint (C3) encourages uniform expert usage under $p(x)$.*

***(2) (Inference)** When a single input x is processed, there is no meaningful approximation to $p(x)$, so enforcing (C3) forces uniform routing regardless of the actual score $\mathbf{S}(x)$, which leads to distorted or suboptimal expert assignment.*

To address this issue, we propose *Selective Sinkhorn Routing* (SSR). During training, each MoE block uses Sinkhorn routing with probability $p \in [0, 1]$ and Softmax gating otherwise. This hybrid strategy allows Softmax gating to train $\mathbf{W}_g$, enabling $\mathbf{S}$ to learn meaningful token–expert compatibilities, while Sinkhorn routing promotes balanced expert utilization and improves training stability. We denote the Linear-cost and Softmax-cost variants as SSR-L and SSR-S, respectively. The overall procedure is illustrated in Figure 1.

However, the Softmax-based routing may still suffer from expert collapse, in which only a small subset of experts is consistently selected due to high gating scores (Cai et al., 2024). To address this issue, during training, we can add Gaussian noise to the cost matrix (SSR with noise) to encourage exploration and prevent expert underutilization. In particular, the noisy cost matrix is computed as:

$$\tilde{\mathbf{C}} = \mathbf{C} + \alpha_{\text{noise}} \cdot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \in \mathbb{R}^{m\times n}, \quad \boldsymbol{\epsilon}_{i,j} \sim \mathcal{N}(0, \sigma^2). \tag{6}$$

Algorithm 2 desribes the training-time behavior of an SMoE block under SSR. Compared with auxiliary-loss methods,

Algorithm 2 Selective Sinkhorn Routing (SSR) in an MoE block during training

Input: Gating scores $\mathbf{S} \in \mathbb{R}^{m \times n}$, Sinkhorn probability p, top-k experts k.
Output: Routing weights $\boldsymbol{\alpha} \in \mathbb{R}^{m \times n}$.
Set $\alpha_{i,j} \leftarrow 0$ for all $i = 1, \ldots, m$ and $j = 1, \ldots, n$
Draw $\tau \sim \mathcal{U}(0, 1)$
if $\tau < p$ **then**
  Compute transport plan $\hat{\mathbf{\Pi}}$ from $\mathbf{S}$ using Sinkhorn algorithm
  for *each token* $i = 1$ *to* m **do**
    Select k experts $E_{i_1}, \ldots, E_{i_k}$ with highest $\left\{\hat{\mathbf{\Pi}}_{i,j}\right\}_{j=1}^{n}$
    for $r = 1$ *to* k **do**
      $\alpha_{i,i_r} \leftarrow \frac{\hat{\mathbf{\Pi}}_{i,i_r}}{\sum_{j=1}^{k} \hat{\mathbf{\Pi}}_{i,i_j}}$
    end
  end
else
  for *each token* $i = 1$ *to* m **do**
    Select k experts $E_{i_1}, \ldots, E_{i_k}$ with highest $\{\mathbf{S}_{i,j}\}_{j=1}^{n}$
    for $r = 1$ *to* k **do**
      $\alpha_{i,i_r} \leftarrow \frac{e^{\mathbf{S}_{i,i_r}}}{\sum_{j=1}^{k} e^{\mathbf{S}_{i,i_j}}}$
    end
  end
end

SSR avoids objective misalignment by removing additional balancing losses and instead encouraging balanced expert utilization through entropy-regularized optimal transport. Compared with existing Sinkhorn-based routing, SSR preserves the OT interpretation while reducing overhead via sparse Sinkhorn updates. Unlike trainable noise-injection methods, SSR w/ noise introduces no additional parameters, maintaining simplicity without sacrificing performance.

Proposition 4.3 (Noise Ensures Every Expert Has Nonzero Selection Probability)**.** *For each token, let* $g = (g_1, \ldots, g_n) \in \mathbb{R}^n$ *be the cost to* n *experts, and let the perturbed costs be:* $\tilde{g}_i = g_i + \alpha_{noise}\epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$. *Let* $P_i = \mathbb{P}(\tilde{g}_i > \tilde{g}_j \text{ for all } j \neq i)$ *denote the probability that expert* i *is selected as the top-1 expert after noise is added.*

Then:

$$P_i = \prod_{j \neq i} \Phi\left(\frac{g_i - g_j}{\sqrt{2}\sigma\alpha_{noise}}\right),$$

where $\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$ *is the CDF of the standard normal distribution.*

By Propositions 4.2 and 4.3, both Sinkhorn routing and noise injection improve training by promoting expert load balancing. In SSR-S, Proposition 4.3 gives $P_i \geq \Phi^n\left(-\frac{1}{\sqrt{2}\,\sigma\,\alpha_{\text{noise}}}\right)$, which depends only on the noise hyperparameter. At inference, we disable both mechanisms and use deterministic Softmax routing to avoid stochastic or batch-dependent behavior.

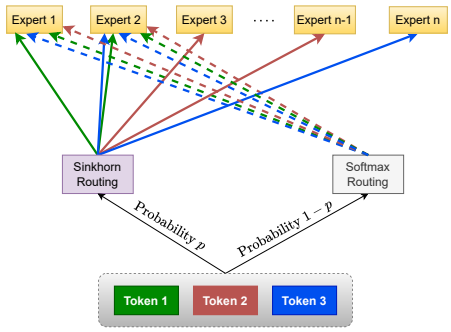


Figure 1. **Training of SSR.** At each layer during the forward pass, a fixed probability $p \in [0, 1]$ is introduced. We randomly choose Sinkhorn-based routing with probability p and Softmax-based routing with probability $1 - p$.

5. Experiments

This section presents the experimental setup, compares SSR with current state-of-the-art balancing baselines, and provides ablation studies. Unless otherwise specified, all experiments are conducted under the conventional SMoE setting. We additionally report Momentum-enhanced results on WikiText-103 and selected ablations to evaluate the compatibility of SSR with Momentum dynamics.

5.1. Settings

Datasets We evaluate SSR on both text and vision tasks using six datasets: WikiText-103 (Merity et al., 2017) and enwik8 (Mahoney, 2011) for language modeling, and ImageNet-1K (Deng et al., 2009), ImageNet-A, ImageNet-O (Hendrycks et al., 2021b), and ImageNet-R (Hendrycks et al., 2021a) for vision robustness evaluation.

Baselines For text tasks, we use the Switch Transformer architecture (Fedus et al., 2022); for vision tasks, we use Swin Transformer (Liu et al., 2021). We compare against Vanilla SMoE (SMoE w/o balancing), SMoE with load balancing loss (SMoE w/ lb loss) (Lepikhin et al., 2020), SMoE with (trainable) noise injection and load balancing loss (SMoE w/ noise) (Shazeer et al., 2017), SMoE with load balancing loss and z-loss (SMoE w/ z-loss) (Zoph et al., 2022), and Sinkhorn-based SMoE (Liu et al., 2024).

Evaluation Metrics and Analysis For text domain, we report token-level perplexity (PPL) on WikiText-103 & byte-level bits-per-character (BPC) on Enwik-8, using zero-shot evaluation on adversarial inputs. For images, models trained on ImageNet-1K are evaluated on a clean validation set and on adversarial, OOD, and real-world filtered sets, using Top-1/Top-5 accuracy and AUPR, see Appendix for details.

5.2. Main results

We compare our methods with several baselines across multiple benchmarks. In each evaluation, the **best result is in bold**, and the second-best is underlined. In the

Table 1. Perplexity (PPL) of SSR variants and baseline models on clean and attacked WikiText-103 under conventional and momentum settings. The Δ column reports the test PPL difference relative to Vanilla SMoE within the corresponding setting. Time overhead is measured relative to Vanilla SMoE in each setting.

Model/Metric	Clean WikiText-103			Attacked WikiText-103		Efficiency
	Valid PPL ↓	Test PPL ↓	Δ Test PPL vs. Baseline	Valid PPL ↓	Test PPL ↓	Training Time Overhead ↓
Conventional setting						
Vanilla SMoE	33.760	35.550	–	42.240	44.190	0.00%
SMoE w/ lb loss	33.248	34.952	-0.598	41.763	43.758	2.04 %
SMoE w/ z-loss	33.242	35.091	-0.459	42.298	44.406	2.12%
SMoE w/ noise	33.196	35.072	-0.478	41.432	43.692	2.50%
Sinkhorn-based SMoE	33.301	35.316	-0.234	42.056	44.351	72.47%
SSR-L (Ours)	32.809	34.573	-0.977	41.230	43.318	0.33%
SSR-S (Ours)	**32.610**	34.744	-0.806	**41.012**	43.283	0.37%
SSR-L w/ noise (Ours)	32.767	**34.367**	-1.183	41.164	43.159	0.62%
SSR-S w/ noise (Ours)	32.881	34.557	-0.993	41.037	**42.941**	0.65%
Momentum setting						
Vanilla SMoE	31.861	33.712	–	39.721	41.756	0.00%
SMoE w/ lb loss	32.561	34.490	+0.778	40.811	42.998	2.49%
SMoE w/ z-loss	32.542	34.235	+0.523	40.665	42.614	4.44%
SMoE w/ noise	32.490	34.077	+0.365	40.299	42.204	6.75%
Sinkhorn-based SMoE	32.576	34.305	+0.593	40.810	42.803	67.52%
SSR-L (Ours)	31.834	33.305	-0.407	39.690	**41.351**	0.36%
SSR-S (Ours)	31.888	33.559	-0.153	40.259	42.104	0.61%
SSR-L w/ noise (Ours)	31.911	**33.223**	-0.489	39.847	41.627	1.58%
SSR-S w/ noise (Ours)	**31.820**	33.338	-0.374	**39.677**	41.639	1.76%

Table 2. Top-1 and Top-5 accuracy on ImageNet-1K and robustness benchmarks: Top-1 accuracy on ImageNet-A (IM-A) and ImageNet-R (IM-R), and AUPR on ImageNet-O (IM-O).

Model	ImageNet-1K			Robustness Benchmarks		
	Top-1 ↑	Δ Top-1 vs. Vanilla	Top-5 ↑	IM-A ↑	IM-O ↑	IM-R ↑
Vanilla SMoE	75.052	–	92.302	6.852	50.690	30.713
SMoE w/ noise	75.148	+0.096	92.356	7.000	50.730	30.657
Swin-MoE	75.322	+0.270	92.578	7.093	50.460	31.743
SSR-L (Ours)	75.402	+0.350	92.528	6.600	**51.040**	30.863
SSR-L w/ noise (Ours)	**77.420**	+2.368	**93.566**	**9.760**	50.530	**33.903**

main tables, the Δ column reports the gap relative to Vanilla SMoE: $\Delta(\text{method}) = \text{Performance(method)} - \text{Performance(Vanilla SMoE)}$. A negative Δ indicates improvement for PPL and BPC, while a positive Δ indicates improvement for vision tasks (Top-1 accuracy). Thus, the sign of Δ depends on the metric: lower is better for PPL/BPC, whereas higher is better for Top-1 accuracy. The symbol "–" denotes no change (i.e., Vanilla SMoE).

We first evaluate SSR on WikiText-103 under both the conventional and Momentum settings (Teo & Nguyen, 2024). In the conventional setting, the backbone follows an interleaved Attention–SMoE architecture with separate residual connections for each block. As shown in Table 1, SSR consistently outperforms Vanilla SMoE and existing balancing methods. In particular, SSR reduces test perplexity by 1.183 PPL relative to Vanilla SMoE, achieving **more than twice** the improvement of prior balancing approaches, while incurring **substantially lower** training-time overhead.

Under the Momentum setting, SMoE can be viewed through a multi-objective optimization lens, where the output norm of each SMoE block generally decreases across layers, with a slight increase in the final layer due to gradient-descent overshooting (Teo & Nguyen, 2024). As shown in Figure 2, SSR preserves this characteristic norm evolution and closely matches the layer-wise trajectory of Vanilla SMoE, indicating compatibility with Momentum dynamics. Notably, SSR improves Vanilla SMoE by 0.489 PPL, while prior balancing methods degrade performance by up to 0.778 PPL. Together with the norm analysis in Figure 2, this suggests that existing balancing strategies may disrupt Momentum-induced optimization dynamics, leading to less stable trajectories, whereas SSR preserves the norm behavior of Vanilla SMoE while improving perplexity. Overall, SSR improves performance in both settings and trains faster than competing balancing mechanisms.

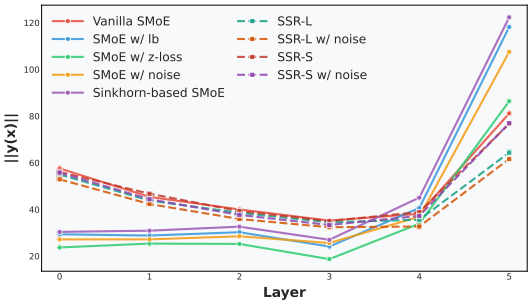


Figure 2. Average output norms across layers on the WikiText-103 validation set under the momentum setting.

Table 3 compares four inference strategies: inference without balancing (*No balancing*), inference with Sinkhorn-based routing using a fixed probability p (*w/ Sinkhorn*), inference with Gaussian noise (*w/ noise*), and inference with both Sinkhorn routing and noise (*w/ both*). We find that, at inference time, disabling load balancing yields the best results: both SSR variants consistently outperform the other inference strategies, consistent with Proposition 4.2.

Table 3. Comparison of different Inference techniques of SSR-L/S w/ noise on clean/attacked WikiText-103.

Inference	Clean WikiText-103		Attacked WikiText-103	
	Valid PPL ↓	Test PPL ↓	Valid PPL ↓	Test PPL ↓
SSR-L w/ noise				
No balancing (Ours)	**31.871**	**33.395**	**40.083**	**41.885**
w/ Sinkhorn	31.883	33.411	40.095	41.905
w/ noise	31.883	33.407	40.099	41.904
w/ both	31.894	33.424	40.111	41.926
SSR-S w/ noise				
No balancing (Ours)	**32.371**	**33.444**	**40.842**	**42.312**
w/ Sinkhorn	32.377	33.447	40.847	42.315
w/ noise	32.392	33.452	40.870	42.317
w/ both	32.407	33.461	40.886	42.325

Next, we evaluate SSR on byte-level language modeling with enwik8 (Table 4). Particularly, SSR variants consistently outperform the baselines. SSR-S w/ noise achieves the best test BPC of 1.128, improving Vanilla SMoE by 0.010 BPC, which is twice the gain of the strongest prior baseline, Sinkhorn-based SMoE. It is also **2.28**× faster than Sinkhorn-based SMoE, improving both performance and training efficiency (Figure 5). SSR-L w/ noise obtains the second-best result, further confirming the effectiveness of SSR for byte-level modeling. In contrast, SMoE w/ noise

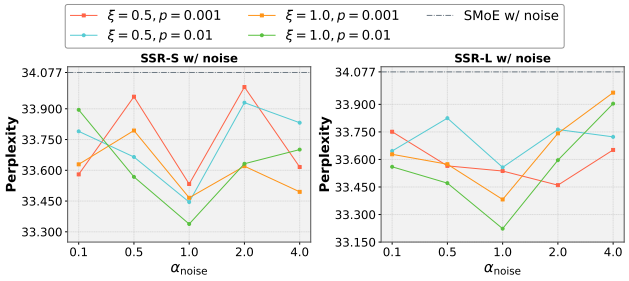


Figure 3. Performance of SSR w/ noise on Wikitext-103 under varying ξ and α_{noise} values in comparison with SMoE w/ noise.

underperforms Vanilla SMoE, highlighting the limited robustness of existing balancing methods when considering both token- and byte-level language modeling.

Finally, we evaluate SSR in the vision domain (See Table 2). Due to the high computational cost of Sinkhorn-based methods, we compare against Vanilla SMoE, SMoE with noise, and Swin-MoE. SSR-L achieves strong results on ImageNet-1K and the best performance on ImageNet-O. Adding noise further improves performance across the remaining metrics, underscoring the benefit of noise in the cost matrix.

Table 4. Bits-per-character (BPC) of SSR-L/S w/ noise vs. SMoE-based & Sinkhorn-based baselines on the Enwik-8 datasets.

Model/Metric	Enwik-8		Δ Test BPC vs. vanilla SMoE
	Valid BPC ↓	Test BPC ↓	
Vanilla SMoE	1.147	1.138	–
SMoE w/ noise	1.154	1.147	+0.009
SMoE w/ aux. loss	1.145	1.136	-0.002
Sinkhorn-based SMoE	1.143	1.133	-0.005
SSR-L w/ noise (Ours)	1.141	1.133	-0.005
SSR-S w/ noise (Ours)	**1.136**	**1.128**	-0.010

5.3. Ablation Studies

5.3.1. Effect of probability p

We evaluate SSR-L w/ noise under different routing probabilities p to study their effect on SMoE performance in Table 5. As shown in the table, p has a clear impact on performance. On the clean WikiText-103 split, $p = 0.001$ achieves the best validation and test PPL, suggesting that a very small fraction of Sinkhorn-based routing is sufficient to provide an effective balancing signal without the conventional auxiliary loss, while on the attacked split, the best performance is obtained with $p = 0.0001$. Overall, SSR-L w/ noise benefits from a small but carefully chosen routing probability: overly large p may over-constrain routing, while overly small p weakens the balancing effect.

Table 5. Perplexity (PPL) of 1-Head SSR-L w/ noise with different probability p on clean/attacked WikiText-103.

p	Clean WikiText-103		Attacked WikiText-103	
	Valid PPL ↓	Test PPL ↓	Valid PPL ↓	Test PPL ↓
0.1	32.874	34.519	41.100	43.183
0.03	33.254	35.194	41.898	44.211
0.01	32.942	34.563	41.159	43.143
10^{-3}	**32.767**	**34.367**	41.164	43.159
10^{-4}	32.845	34.677	**40.861**	**42.905**
10^{-4}	33.164	34.935	41.569	43.690

5.3.2. Effect of α_{noise} in SSR-L/S w/ noise

We further study the effect of gating noise in SSR-L/S w/ noise by varying $p \in \{0.001, 0.01\}$, $\xi \in \{0.5, 1\}$, and $\alpha_{\text{noise}} \in \{0.1, 0.5, 1, 2, 4\}$. We compare SSR-L/S w/ noise against SMoE w/ noise, the strongest baseline combining load balancing with Momentum in Table 1. As shown in Figure 3, both SSR-L and SSR-S w/ noise consistently outperform SMoE w/ noise across all tested settings, achieving the best test PPLs of 33.223 and 33.338, respectively, compared to 34.007 for SMoE w/ noise. These results demonstrate that SSR remains stable and effective under diverse noise and momentum configurations.

5.3.3. Effect of ξ in Sinkhorn Algorithm

Next, we examine the effect of the regularization parameter ξ on the performance of SSR w/ noise variants, with fixed settings of $p = 0.001$ and $\delta = 0.0001$. As discussed in Section 4.1, employing a Softmax-based cost in regularized OT improves stability compared to a linear cost by bounding the cost matrix, thereby mitigating overflow during the Sinkhorn iterations. As shown in Table 6, smaller ξ values (e.g., 0.05, 0.1) cause numerical overflow in SSR-L w/ noise (yielding `NaN`), whereas SSR-S w/ noise remains stable and continues to achieve competitive performance.

Table 6. Perplexity (PPL) of 1-Head SSR-L/S w/ noise with different ξ on clean/attacked WikiText-103.

ξ	Clean WikiText-103		Attacked WikiText-103	
	Valid PPL ↓	Test PPL ↓	Valid PPL ↓	Test PPL ↓
SSR-L w/ noise				
0.05	NaN	NaN	NaN	NaN
0.1	NaN	NaN	NaN	NaN
0.5	32.767	**34.367**	41.164	43.159
1	32.867	34.783	41.556	43.708
SSR-S w/ noise				
0.05	32.927	34.605	41.138	43.022
0.1	**32.647**	34.442	**40.834**	43.059
0.5	32.881	34.557	41.037	**42.941**
1	33.053	34.564	41.056	42.945

6. Conclusion

This paper proposes Selective Sinkhorn Routing, a novel method for Sparse Mixture-of-Experts models that improves expert utilization with minimal overhead. We reformulate routing as a regularized optimal transport problem with a constraint on the number of tokens per expert. Unlike prior methods, we derive gating scores directly from the transport map, leading to more balanced and effective token-to-expert assignments. Our theoretical and empirical results show that applying Sinkhorn intermittently and injecting noise into the cost matrix reduces training time and improves performance compared to existing routing methods, while the modifications should be disabled at inference and add no extra cost. These results highlight the practicality and effectiveness of our approach for efficient SMoE design across both training and inference deployment.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Cai, W., Jiang, J., Wang, F., Tang, J., Kim, S., and Huang, J. A survey on mixture of experts. *arXiv preprint arXiv:2407.06204*, 2024.

Chi, Z., Dong, L., Huang, S., Dai, D., Ma, S., Patra, B., Singhal, S., Bajaj, P., Song, X., Mao, X.-L., et al. On the representation collapse of sparse mixture of experts. *Advances in Neural Information Processing Systems*, 35: 34600–34613, 2022.

Clark, A., de Las Casas, D., Guy, A., Mensch, A., Paganini, M., Hoffmann, J., Damoc, B., Hechtman, B., Cai, T., Borgeaud, S., et al. Unified scaling laws for routed language models. In *International conference on machine learning*, pp. 4057–4086. PMLR, 2022.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

Du, N., Huang, Y., Dai, A. M., Tong, S., Lepikhin, D., Xu, Y., Krikun, M., Zhou, Y., Yu, A. W., Firat, O., et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pp. 5547–5569. PMLR, 2022.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8340–8349, 2021a.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15262–15271, 2021b.

Kool, W., Maddison, C. J., and Mnih, A. Unbiased gradient estimation with balanced assignments for mixtures of experts. *arXiv preprint arXiv:2109.11817*, 2021.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., and Chen, Z. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

Lin, B., Tang, Z., Ye, Y., Cui, J., Zhu, B., Jin, P., Zhang, J., Ning, M., and Yuan, L. Moe-llava: Mixture of experts for large vision-language models. *CoRR*, abs/2401.15947, 2024. URL https://doi.org/10.48550/arXiv.2401.15947.

Liu, T., Puigcerver, J., and Blondel, M. Sparsity-constrained optimal transport. *arXiv preprint arXiv:2209.15466*, 2022.

Liu, T., Blondel, M., Riquelme, C., and Puigcerver, J. Routers in vision mixture of experts: An empirical study. *arXiv preprint arXiv:2401.15969*, 2024.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Mahoney, M. Large text compression benchmark, 2011.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Morris, J., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., and Qi, Y. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In Liu, Q. and Schlangen, D. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 119–126, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.16. URL https://aclanthology.org/2020.emnlp-demos.16/.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Rasheed, H., Maaz, M., Shaji, S., Shaker, A., Khan, S., Cholakkal, H., Anwer, R. M., Xing, E., Yang, M.-H., and Khan, F. S. Glamm: Pixel grounding large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13009–13018, June 2024.

Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Pinto, A. S., Keysers, D., and Houlsby, N. Scaling vision with sparse mixture of experts. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Teo, R. and Nguyen, T. M. MomentumSMoe: Integrating momentum into sparse mixture of experts. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=y929esCZNJ.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Zhou, Y., Lei, T., Liu, H., Du, N., Huang, Y., Zhao, V., Dai, A. M., Le, Q. V., Laudon, J., et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

Zoph, B., Bello, I., Kumar, S., Du, N., Huang, Y., Dean, J., Shazeer, N., and Fedus, W. St-moe: Designing stable and transferable sparse expert models, 2022. URL https://arxiv.org/abs/2202.08906.

A. Theoretical results

A.1. Proposition 4.1

Proof. Let $\hat{\mathbf{\Pi}}_i = (\hat{\mathbf{\Pi}}_{i,1}, \dots, \hat{\mathbf{\Pi}}_{i,n}) \in \mathbb{R}^n$ be the i-th row of the solution to the entropy-regularized optimal transport problem. We aim to minimize the Kullback-Leibler (KL) divergence:

$$\mathrm{KL}(\alpha_i \,\|\, \hat{\mathbf{\Pi}}_i) = \sum_{j=1}^{n} \alpha_{i,j} \log\left(\frac{\alpha_{i,j}}{\hat{\mathbf{\Pi}}_{i,j}}\right),$$

where $\alpha_{i,j} \geq 0$, $\sum_{j=1}^{n} \alpha_{i,j} = 1$, and $\mathrm{supp}(\alpha_i) \leq k$. This means we want to select at most k non-zero entries for α_i such that the KL divergence is minimized.

Let $\mathcal{T}_i \subset \{1, \dots, n\}$ be the support of α_i, i.e., the indices where $\alpha_{i,j} > 0$. We know that $|\mathcal{T}_i| \leq k$. For fixed $\mathcal{T}_i$, the problem can be simplified to the following convex optimization problem:

$$\min_{\substack{\alpha_j \geq 0,\, j \in \mathcal{T}_i \\ \sum_{j \in \mathcal{T}_i} \alpha_j = 1}} \sum_{j \in \mathcal{T}_i} \alpha_j \log\left(\frac{\alpha_j}{\hat{\mathbf{\Pi}}_{i,j}}\right).$$

The Lagrangian for this problem is:

$$\mathcal{L}(\alpha, \lambda) = \sum_{j \in \mathcal{T}_i} \alpha_j \log\left(\frac{\alpha_j}{\hat{\mathbf{\Pi}}_{i,j}}\right) + \lambda \left(\sum_{j \in \mathcal{T}_i} \alpha_j - 1\right),$$

where λ is the Lagrange multiplier. Taking the gradient with respect to α_j and setting it to zero gives:

$$\frac{\partial \mathcal{L}}{\partial \alpha_j} = \log\left(\frac{\alpha_j}{\hat{\mathbf{\Pi}}_{i,j}}\right) + 1 + \lambda = 0 \quad \Rightarrow \quad \alpha_j = \hat{\mathbf{\Pi}}_{i,j} e^{-1-\lambda}.$$

The KL divergence is given by:

$$\mathrm{KL}(\alpha_i \,\|\, \hat{\mathbf{\Pi}}_i) = \sum_{j \in \mathcal{T}_i} \alpha_j \log\left(\frac{\alpha_j}{\hat{\mathbf{\Pi}}_{i,j}}\right).$$

By substituting $\alpha_j = \frac{\hat{\mathbf{\Pi}}_{i,j}}{\sum_{l \in \mathcal{T}_i} \hat{\mathbf{\Pi}}_{i,l}}$, we get:

$$\mathrm{KL}(\alpha_i \,\|\, \hat{\mathbf{\Pi}}_i) = \sum_{j \in \mathcal{T}_i} \frac{\hat{\mathbf{\Pi}}_{i,j}}{\sum_{l \in \mathcal{T}_i} \hat{\mathbf{\Pi}}_{i,l}} \log\left(\frac{\hat{\mathbf{\Pi}}_{i,j}}{\sum_{l \in \mathcal{T}_i} \hat{\mathbf{\Pi}}_{i,l}} \cdot \frac{1}{\hat{\mathbf{\Pi}}_{i,j}}\right).$$

This simplifies to:

$$\mathrm{KL}(\alpha_i \,\|\, \hat{\mathbf{\Pi}}_i) = -\log\left(\sum_{j \in \mathcal{T}_i} \hat{\mathbf{\Pi}}_{i,j}\right),$$

To minimize the KL divergence, we need to maximize the sum $\sum_{j \in \mathcal{T}_i} \hat{\mathbf{\Pi}}_{i,j}$. This is achieved by selecting the top-k largest values of $\hat{\mathbf{\Pi}}_{i,j}$. Therefore, the optimal support set is:

$$\mathcal{T}_i = \mathrm{TopK}(\hat{\mathbf{\Pi}}_i, k).$$

Thus, the optimal α_i is given by:

$$\alpha_{i,j} = \begin{cases} \dfrac{\hat{\mathbf{\Pi}}_{i,j}}{\sum_{l \in \mathcal{T}_i} \hat{\mathbf{\Pi}}_{i,l}}, & \text{if } j \in \mathcal{T}_i, \\ 0, & \text{otherwise.} \end{cases}$$

This completes the proof for Proposition 4.1.

□

A.2. Proposition 4.2

Proof. **(1) Training.** Let $\{x_1, \dots, x_m\} \sim p(x)$ be a batch drawn i.i.d. Let $\pi(x_i) = f(g(x_i)) \in \Delta^n$ be the routing distribution for input x_i, and define the batch-average routing vector:

$$\bar{\pi} = \frac{1}{m} \sum_{i=1}^{m} \pi(x_i).$$

By the law of large numbers, as $m \to \infty$, we have:

$$\bar{\pi} \xrightarrow{\text{a.s.}} \mathbb{E}_{x \sim p(x)}[\pi(x)].$$

Thus, enforcing constraint *(C3)*:

$$\mathbf{\Pi}^\top \mathbf{1}_m = \tfrac{m}{n} \mathbf{1}_n.$$

is equivalent to enforcing:

$$\bar{\pi} = \frac{1}{n} \mathbf{1}_n,$$

which promotes uniform expert usage across the data distribution. This helps prevent expert underuse and encourages specialization during training.

(2) Inference. At inference time, only a single input x is available. Enforcing *(C3)* with $m = 1$ yields:

$$\pi(x) = \frac{1}{n} \mathbf{1}_n,$$

which forces uniform routing, ignoring the input-specific score $\mathbf{S}(x)$. This contradicts the goal of expert specialization and leads to suboptimal predictions. Furthermore, a single sample cannot approximate $p(x)$, so the population-level balancing constraints become meaningless.

Conclusion. Training with Sinkhorn routing over batches supports expert balancing over the data distribution. In contrast, inference-time routing should be purely based on the input without enforcing expert balance constraints. □

A.3. Proposition 4.3

Proof. For any fixed i, we observe that:

$$\tilde{g}_i - \tilde{g}_j = (g_i - g_j) + \alpha_{\text{noise}}(\epsilon_i - \epsilon_j).$$

Since $\epsilon_i, \epsilon_j \sim \mathcal{N}(0, \sigma^2)$; we have $\epsilon_i - \epsilon_j \sim \mathcal{N}(0, 2\sigma^2)$. Thus:

$$\mathbb{P}(\tilde{g}_i > \tilde{g}_j) = \mathbb{P}(\alpha_{\text{noise}}\epsilon_i - \alpha_{\text{noise}}\epsilon_j > g_j - g_i) \quad (7)$$

$$= \mathbb{P}\left(\epsilon_i - \epsilon_j > \frac{g_j - g_i}{\alpha_{\text{noise}}}\right) \quad (8)$$

$$= \Phi\left(\frac{g_i - g_j}{\sqrt{2}\sigma\alpha_{\text{noise}}}\right). \quad (9)$$

Independence of the noise implies:

$$P_i = \prod_{j \neq i} \mathbb{P}(\tilde{g}_i > \tilde{g}_j) = \prod_{j \neq i} \Phi\left(\frac{g_i - g_j}{\sqrt{2}\sigma\alpha_{\text{noise}}}\right).$$

Since each $g_i - g_j$ is fixed and finite, and $\Phi(z) \in (0, 1)$ for all real z, it follows that:

$$P_i > 0.$$

Hence, P_i is a positive constant that depends only on g and σ, and is independent of any input-dependent softmax transformation. Moreover, in the SSR-S variant, the probability $P_i \geq \Phi^n\left(\frac{-\sqrt{2}}{\sigma\alpha_{\text{noise}}}\right)$, which depends solely on the noise hyperparameter. □

B. Experimental Details

B.1. Dataset Details

WikiText-103 (Merity et al., 2017) is a large-scale collection comprising over 100 million tokens sourced from 23,805 "Good" articles and 4,790 "Featured articles". Specifically, the training set includes over 103 million tokens, while the validation and test sets consist of 217,646 and 245,569 tokens, respectively. This dataset retains its original case, punctuation, and numerical values, resulting in a diverse vocabulary of 267,735 unique tokens. A clean version corresponds to the original dataset, while the attacked version is generated using TextAttack's word-swap attack (Morris et al., 2020), where words in the validation and test sets are randomly replaced with the generic token "AAA". This modification increases the difficulty for the model to predict the next word in the sequence accurately.

Enwik-8 (Mahoney, 2011) is a byte-level dataset comprising 100 million bytes sourced from Wikipedia. It includes not only English text but also markup, special characters, and multilingual content. The dataset is divided into 90 million bytes for training, 5 million for validation, and 5 million for testing.

ImageNet-1K contains 1.28 million training images and 50,000 validation images. The model is trained to classify each input image into one of 1,000 categories. Top-1 and Top-5 accuracy are reported across all experiments.

ImageNet-A (Hendrycks et al., 2021b) contains real-world images that have been adversarially filtered to mislead existing ImageNet classifiers. A subset of 200 classes is selected from the original 1,000 ImageNet-1K categories, focusing on those where misclassifications would be particularly severe. These 200 classes broadly represent the diversity of categories found in ImageNet-1K.

ImageNet-O (Hendrycks et al., 2021b) contains adversarially filtered examples designed to challenge ImageNet out-of-distribution detectors. It consists of samples drawn from ImageNet-22K that are not part of ImageNet-1K, specifically selected because a ResNet-50 model incorrectly classifies them as ImageNet-1K categories with high confidence.

Imagenet-R (Hendrycks et al., 2021a) contains a variety of artistic renditions of object classes originally found in ImageNet, which are typically discouraged by the standard ImageNet guidelines. ImageNet-R includes 30,000 such renditions spanning 200 classes, selected as a subset of the ImageNet-1K categories.

B.2. Model Architecture and Training Configurations

For language modeling, we adopt a medium-scale configuration with 6 layers for Wikitext-103 and 8 layers for Enwik-8. Each layer consists of a multi-head self-attention (MHA) block followed by a SMoE block, both with residual connections. Training is performed with a batch size of 48 for 80,000 steps, using a learning rate of 0.0007 with 4,000 warm-up steps and a dropout rate of 0.1. The model uses 8 attention heads for each MHA block and processes sequences of 512 tokens in each batch, with attention spans of 1,024 for Wikitext-103 and 2,048 for Enwik-8. The SMoE module has 16 experts with top-2 routing. The hidden and expert dimensions are 352 . The resulting model sizes are 216M parameters for WikiText-103 and 36M for Enwik8,

aligning with commonly explored scales in recent work (Teo & Nguyen, 2024).

The baseline settings are customized fairly to the paper report. Specifically, for the SMoE w/ noise, we follow the noise initialization in (Shazeer et al., 2017). The auxiliary loss coefficient is set to 0.01 (Shazeer et al., 2017; Lepikhin et al., 2020), and the z-loss coefficient to 0.001 (Zoph et al., 2022). For the Sinkhorn-based SMoE, we set $\xi = 1, \delta = 0.0001$ and $\eta = 100$. With SSR-variants, we consider $\xi \in \{0.05, 0.5, 1\}, \delta = 0.0001$, $\eta = 100$, $p \in \{0.0001, 0.001, 0.01\}$ and $\alpha_{\text{noise}} \in \{0.3, 1, 4\}$.

For image classification, we use a compact 4-stage architecture with depths [2, 2, 18, 2]. The first two stages each have 2 blocks (self-attention + feed-forward); the third stage has 18 blocks, where self-attention alternates between feed-forward and MoE layers; and the final stage includes a self-attention–feed-forward block followed by a self-attention–MoE block. The embedding dimension is 96 with attention heads [3, 6, 12, 24]. We employ 32 experts for MoE layers with top-2 routing (550M parameters) and train for 60 epochs using AdamW (base LR 1.25e-4, min LR 1.25e-7, weight decay 0.1, cosine schedule), batch size 96, and an auxiliary loss coefficient of 0.1. For SSR-L and SSR-L w/ noise, we consider $\xi = 0.5$, $\delta = 0.0001$, $\eta = 100$, $p = 0.01$ and $\alpha_{\text{noise}} = 1$. While prior work (Teo & Nguyen, 2024) has explored model sizes ranging from 36M to 388M parameters, in this paper we extend the scale further to 550M parameters, demonstrating that SSR continues to provide strong performance gains at larger scale.

B.3. Compute Resources

All models are trained and evaluated using 2 NVIDIA A100 GPUs with 40GB of memory each.

C. Additional Experimental Results

Table 7. Perplexity (PPL) of SSR-L with different probability p on clean/attacked WikiText-103.

p	Clean WikiText-103		Attacked WikiText-103	
	Valid PPL ↓	Test PPL ↓	Valid PPL ↓	Test PPL ↓
0.5	33.321	34.752	41.748	43.587
0.1	**32.754**	34.815	**40.984**	43.386
0.03	32.859	34.712	41.281	43.489
0.01	33.051	34.913	41.209	43.423
10^{-3}	33.029	34.710	41.403	43.437
10^{-4}	32.809	**34.573**	41.230	**43.318**
10^{-5}	32.992	34.794	41.573	43.821

Table 7 reports the effect of varying the probability p of applying Sinkhorn routing in SSR-L. Across all tested values, SSR-L achieves lower Test PPL than Vanilla SMoE, showing that sparse Sinkhorn routing provides a useful balancing signal. The best Test PPL is obtained at $p = 0.0001$ on both clean and attacked WikiText-103, suggesting that only a very small fraction of Sinkhorn-based routing is needed to improve performance. However, SSR-L still underperforms SSR-L w/ noise in Table 1, indicating that gating noise further improves the effectiveness of SSR. Overall, these results show that p should remain small: using Sinkhorn routing too frequently may over-constrain the router, while using it too rarely weakens the balancing effect.

Table 8. Perplexity (PPL) of SSR-S with different probability p on clean/attacked WikiText-103.

p	Clean WikiText-103		Attacked WikiText-103	
	Valid PPL ↓	Test PPL ↓	Valid PPL ↓	Test PPL ↓
0.1	33.112	35.112	41.483	43.728
0.03	33.033	35.002	41.434	43.802
0.01	33.249	35.060	42.014	44.187
10^{-3}	**32.610**	**34.744**	**41.012**	**43.283**
10^{-4}	33.533	35.240	42.036	44.108
10^{-5}	32.905	34.804	41.523	43.895

Table 9. Perplexity (PPL) of SSR-S w/ noise with different probability p on clean/attacked WikiText-103.

p	Clean WikiText-103		Attacked WikiText-103	
	Valid PPL ↓	Test PPL ↓	Valid PPL ↓	Test PPL ↓
0.1	33.212	34.768	41.568	43.411
0.03	33.470	35.143	41.960	43.962
0.01	33.193	34.918	41.548	43.764
10^{-3}	**32.881**	**34.557**	**41.037**	**42.941**
10^{-4}	33.160	34.908	41.476	43.498
10^{-5}	33.326	34.809	41.804	43.514

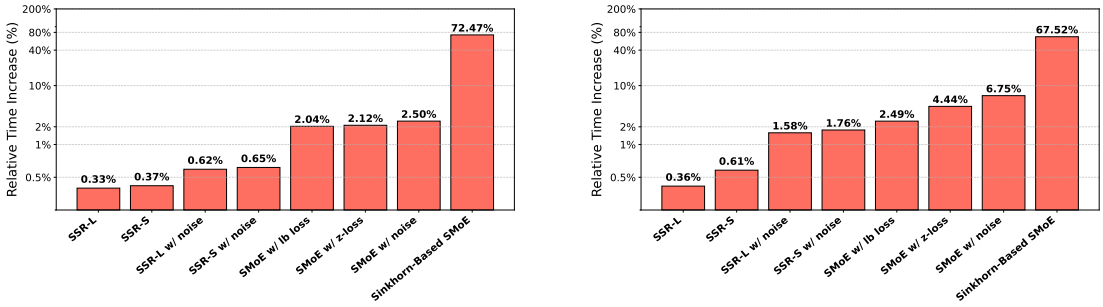
Similarly, we evaluate SSR-S and SSR-S w/ noise across different probabilities p, as reported in Table 8 and Table 9. For both variants, the best performance is consistently achieved at $p = 10^{-3}$ across clean and attacked WikiText-103. This suggests that a small amount of Sinkhorn routing is sufficient to provide an effective balancing signal for SSR-S.

Compared with SSR-S, adding gating noise further improves performance, reducing the clean Test PPL from 34.744 to 34.557 and the attacked Test PPL from 43.283 to 42.941 at the same probability $p = 10^{-3}$. These results indicate that gating noise complements sparse Sinkhorn routing by improving routing robustness. Overall, p should be carefully controlled: overly large values may over-constrain routing, while overly small values may weaken the balancing effect.

D. Detailed Derivation: Entropy-Regularized Maximum-Cost OT Solution

Recall the entropy-regularized maximum-cost OT problem in Eq. 3:

$$\hat{\mathbf{\Pi}} := \arg\max_{\mathbf{\Pi}} [\langle \mathbf{\Pi}, \mathbf{C} \rangle - \xi \langle \mathbf{\Pi}, \log \mathbf{\Pi} \rangle],$$
$$\text{s.t.} \begin{cases} \mathbf{\Pi} > 0, \\ \mathbf{\Pi} \mathbf{1}_n = \mathbf{1}_m, \\ \mathbf{\Pi}^\top \mathbf{1}_m = (m/n) \mathbf{1}_n, \end{cases}$$



(a) Conventional setting

(b) Momentum setting

Figure 4. Training time overhead relative to SMoE (Vanilla) on WikiText-103 under conventional and momentum settings.

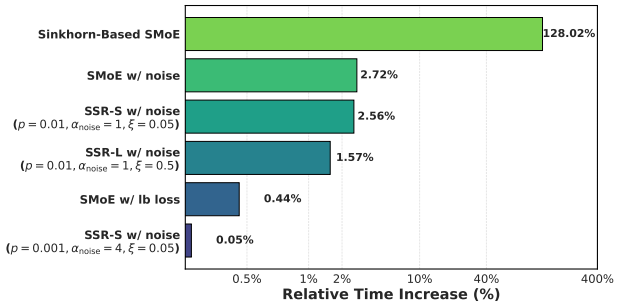


Figure 5. Training time Overhead vs. Vanilla SMoE on Enwik-8 dataset.

where $\mathbf{\Pi}, \mathbf{C} \in \mathbb{R}^{m\times n}$, $\xi > 0$ and $\langle \mathbf{A}, \mathbf{B}\rangle := \sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{A}_{i,j}\mathbf{B}_{i,j}$. Let us consider the Lagrange function:

$$\mathbb{L}\left(\mathbf{\Pi}, \lambda, \mu\right) = \sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{\Pi}_{i,j}\mathbf{C}_{i,j} - \xi \sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{\Pi}_{i,j}\log(\mathbf{\Pi}_{i,j}) + \sum_{i=1}^{m} \lambda_i \left(\sum_{j=1}^{n} \mathbf{\Pi}_{i,j} - 1\right) + \sum_{j=1}^{n} \mu_j \left(\sum_{i=1}^{m} \mathbf{\Pi}_{i,j} - (m/n)\right),$$

where $\mathbf{\Pi} = [\mathbf{\Pi}_{i,j}]_{i,j} \in \mathbb{R}^{m\times n}$, $\lambda = [\lambda_1, \ldots, \lambda_m], \mu = [\mu_1, \ldots, \mu_n]$ such that $\lambda_i, \mu_j \geq 0$ and $\sum_{i=1}^{m} \lambda_i = 1, \sum_{j=1}^{n} \mu_j = 1$ for $i = \overline{1, m}, j = \overline{1, n}$. Then we have some following assertions:

$$\frac{\partial \mathbb{L}\left(\mathbf{\Pi}, \lambda\right)}{\partial \mathbf{\Pi}_{i,j}} = \mathbf{C}_{i,j} - \xi \log \mathbf{\Pi}_{i,j} - \xi + \lambda_i + \mu_j = 0,$$

$$\lambda_i \left(\sum_{j=1}^{n} \mathbf{\Pi}_{i,j} - 1\right) = 0, \text{for } i = \overline{1, m},$$

$$\mu_j \left(\sum_{i=1}^{m} \mathbf{\Pi}_{i,j} - m/n\right) = 0, \text{for } j = \overline{1, n}.$$

Hence, we have

$$\mathbf{\Pi}_{i,j} = e^{(\mathbf{C}_{i,j} - \epsilon + \lambda_i + \mu_j)/\epsilon}.$$

Let denote kernel matrix $\mathbf{K} = \exp(\mathbf{C}/\xi - 1_{m\times n})$, then we obtain the solution of the problem in Eq. 3:

$$\hat{\mathbf{\Pi}} = \begin{bmatrix} e^{\lambda_1/\xi} & 0 & \ldots & 0 \\ 0 & e^{\lambda_2/\xi} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & e^{\lambda_m/\xi} \end{bmatrix} \mathbf{K} \begin{bmatrix} e^{\mu_1/\xi} & 0 & \ldots & 0 \\ 0 & e^{\mu_2/\xi} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & e^{\mu_n/\xi} \end{bmatrix} = \text{diag}\left(e^{\lambda/\xi}\right) \mathbf{K} \, \text{diag}\left(e^{\mu/\xi}\right).$$

Conclusion. We obtained the solution of the entropy-regularized maximum-cost OT problem. It is evident that the value range of $\hat{\mathbf{\Pi}}$ is directly influenced by the matrix $\mathbf{K}$, which in turn depends on the cost matrix $\mathbf{C}$. Therefore, employing either a linear or softmax cost formulation will inherently affect the solution of Eq. 3.