Teacher-Guided Routing for Sparse Vision Mixture-of-Experts

Masahiro Kada[1] Ryota Yoshihashi[1] Satoshi Ikehata[2,3] Rei Kawakami[1] Ikuro Sato[1,2]
[1]Institute of Science Tokyo [2]DENSO IT Laboratory [3]National Institute of Informatics
Tokyo, Japan
kada@d-itlab.comp.isct.ac.jp

Abstract

Recent progress in deep learning has been driven by increasingly large-scale models, but the resulting computational cost has become a critical bottleneck. Sparse Mixture of Experts (MoE) offers an effective solution by activating only a small subset of experts for each input, achieving high scalability without sacrificing inference speed. Although effective, sparse MoE training exhibits characteristic optimization difficulties. Because the router receives informative gradients only through the experts selected in the forward pass, it suffers from gradient blocking and obtains little information from unselected routes. This limited, highly localized feedback makes it difficult for the router to learn appropriate expert-selection scores and often leads to unstable routing dynamics, such as fluctuating expert assignments during training. To address this issue, we propose TGR-MoE: Teacher-Guided Routing for Sparse Vision Mixture-of-Experts, a simple yet effective method that stabilizes router learning using supervision derived from a pretrained dense teacher model. TGR-MoE constructs a teacher router from the teacher's intermediate representations and uses its routing outputs as pseudo-supervision for the student router, suppressing frequent routing fluctuations during training and enabling knowledge-guided expert selection from the early stages of training. Extensive experiments on ImageNet-1K and CIFAR-100 demonstrate that TGR consistently improves both accuracy and routing consistency, while maintaining stable training even under highly sparse configurations.

1. Introduction

Sparse Mixture-of-Experts (MoE) architectures [37] enable compute-efficient scaling by activating only a small subset of experts per input, allowing model capacity to grow without a proportional increase in computation. Originally popularized in large-scale language models [8, 21], MoE has recently been extended to vision models for recognition and generation tasks [9, 29, 32, 41]. In this work, we focus on

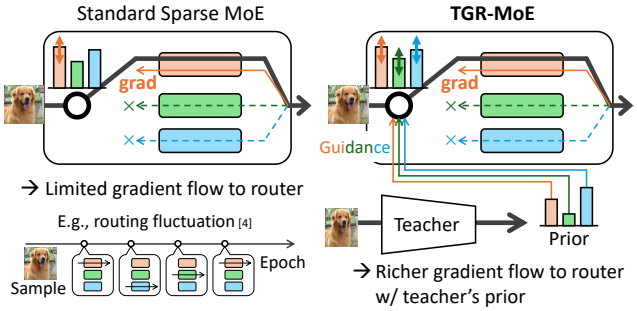

Figure 1. Comparison between standard sparse MoE routing (left) and our Teacher-Guided Routing in TGR-MoE (right). The bar plots denote the router's per-expert routing scores. In a conventional sparse MoE, the router receives gradient feedback only from the experts it selects, leaving the majority of experts without informative signals. This limited and highly localized feedback can lead to locally optimal routing behaviors, such as expert collapse or frequent changes in expert selection. TGR-MoE introduces a global prior derived from the intermediate features of a pretrained dense teacher model. This prior provides additional supervision to the router, complementing the sparse gradient signals and encouraging more coherent routing preferences even for experts that are not selected at each step.

sparse Vision MoE (VMoE) (*e.g.*, [29, 32]), where individual patch tokens are fed into MoE layers in parallel.

Despite their computational advantages, sparse MoE models remain challenging to train stably [49]. Unlike dense (*i.e.*, non-MoE) models, which update all parameters on every input, sparse MoE architectures activate and update only a small subset of experts and the router for each input. We argue that an important yet under-discussed challenge in this training regime is *gradient blocking*: as illustrated in Fig. 1 (left), because only selected experts participate in the forward and backward passes, the router receives no informative feedback from unselected routes. As a result, it is difficult for the router to learn appropriate expert-selection scores for each input, especially in the early stage of training when expert specialization has not yet emerged.

This lack of informative, data-dependent supervision can

easily lead to unstable routing dynamics. In particular, the expert assigned to a fixed input can vary in an excessively frequent fashion over the course of training, a phenomenon referred to as *routing fluctuation* [4]. Although auxiliary load-balancing losses [37] are commonly used to mitigate expert-usage imbalance, they are not sufficient to effectively suppress routing instability during training and can even amplify changes in expert assignment over the course of training. Such temporal inconsistency complicates optimization because the same sample may update different experts across iterations, even though only one or a small fixed set of experts will be used at inference. Overly frequent changes in expert assignment hinder stable expert specialization and can leave some experts undertrained, particularly in settings with a large number of experts. These considerations motivate us to provide the router with an external prior that supplies informative guidance beyond the sparsely activated routes, as illustrated in Fig. 1 (right).

We propose *TGR-MoE: Teacher-Guided Routing for Sparse Vision Mixture-of-Experts*, a framework that stabilizes sparse MoE routing by leveraging a pretrained dense teacher. Rather than modifying the task objective, TGR-MoE attaches an auxiliary *teacher router* to the frozen teacher backbone. The teacher router is trained with a load-balancing loss and an entropy loss, which were formerly used as regularizers for the target router, to produce routing distributions that are both balanced across experts and sufficiently confident. The resulting teacher router yields stable routing distributions that are used to guide the target router. The MoE student learns its router by distilling these distributions: for each MoE layer, the student gating distribution is trained to match the teacher's via a Kullback–Leibler divergence term combined with the standard task loss. Because the teacher backbone is frozen and only the lightweight teacher router is optimized, teacher routing converges quickly and remains stable, and the distillation term encourages temporally consistent, semantically meaningful expert assignments in the student. The proposed training strategy introduces a prior to the router's softmax outputs with different abstraction levels, relieving the gradient blocking problem without freezing the target router, while no additional cost is required at test time.

We evaluate TGR-MoE against the standard VMoE [32] and its variants. Across ImageNet-1K [35] pre-training and downstream tasks including CIFAR-10/100 [19] and Oxford-IIIT Pets [27], TGR-MoE consistently outperforms VMoE at all model scales (Tiny, Small, Base) under a comparable compute budget. It is competitive with, and often superior to, advanced MoE variants such as expert-choice routing [48], SoftMoE [29], and z-loss regularization [49]. Applying TGR-MoE during fine-tuning further yields the strongest transfer performance on CIFAR-100, indicating that explicit routing guidance remains beneficial beyond pre-training. TGR-MoE also scales more reliably with the number of experts, maintaining steady gains even in regimes where conventional VMoE saturates (Sec. 4.3), and routing-consistency analysis shows that it substantially reduces routing fluctuations and reaches stable expert assignments earlier than VMoE (Sec. 5.1).

2. Related Work

Mixture of Experts. While MoE was originally introduced in the early 1990s [14, 26], it has recently emerged as an efficient way to scale model capacity without increasing computation proportionally [21, 37]. Driven by scaling laws [13, 18], MoE has been widely adopted in large language models [5, 7, 8, 16, 21, 22, 43] and vision models [2, 9, 15, 17, 24, 28, 32, 38, 46]. Despite this progress, MoE architectures face several training challenges. A well-known issue is expert-usage imbalance, where only one or a few experts receive most routing traffic; auxiliary load-balancing losses [8, 32, 37] and alternative designs such as expert-choice routing [48] or normalized routing strategies [6] aim to address this. Prior work has also noted numerical sensitivities arising from interactions between router logits and expert outputs [49]. To mitigate issues caused by gradients flowing only through chosen experts, giving the router localized and discontinuous feedback, several methods introduce continuous relaxations of routing, including DSelect-k [11], differentiable top-k gating [36], and ReMoE [42]. Although these smooth the gating function, expert activations remain discrete. Fully continuous formulations such as Soft-MoE [29] eliminate discrete selection entirely, while approaches like SMEAR [25] merge experts via adaptive soft combinations.

Sparse MoE training also exhibits *routing fluctuation*, where expert assignments vary across training steps. StableMoE [4] and HashMoE [33] improve stability through decoupled or fixed routing, highlighting the value of consistent assignments. However, such methods constrain or freeze routing, limiting the model's ability to adapt routing decisions as representations evolve. In this paper, instead of fixing routing, we guide the router during training with an external, stable source of global guidance.

Knowledge Distillation. Knowledge distillation (KD) transfers predictive behavior or intermediate representations from a teacher model to a student [12, 34]. Representative vision distillation methods include DeiT [39], which introduced token-based teacher guidance for ViTs, DKD [47], which separates target and non-target logits to stabilize training, and CAT-KD [10], which leverages class-wise attention to enhance feature alignment. A few recent studies distill information directly into the router, but with objectives fundamentally different from ours. Read-ME [1] transfers the activation sparsity pattern of a pretrained dense

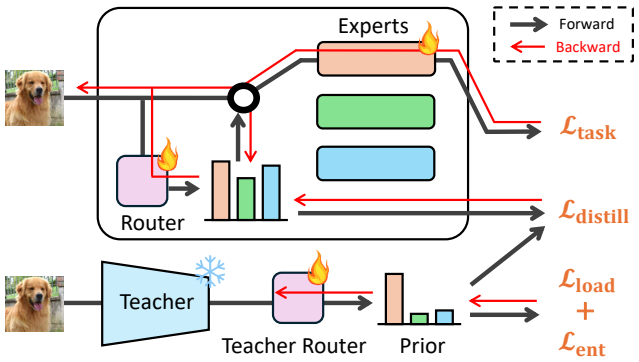

Figure 2. Detailed architecture of the proposed TGR-MoE, illustrated for layer i. The teacher provides a routing signal $\mathcal{L}_{\text{distill}}$ to guide the student router, with the additional balancing loss $\mathcal{L}_{\text{load}}$ and the regularization loss $\mathcal{L}_{\text{ent}}$.

model to an MoE router in order to reproduce the teacher's computation during model conversion, and Dynamic Expert Specialization [23] constrains a fine-tuned router to remain close to the routing distribution of a pretrained MoE to mitigate catastrophic forgetting. Both methods therefore apply distillation to preserve a pre-existing routing behavior for purposes such as model conversion or domain adaptation. However, neither approach directly addresses the instability that originates from discrete routing during pretraining.

3. Method

We propose TGR-MoE, a framework that stabilizes sparse MoE routing by leveraging a pretrained dense teacher model. The teacher router provides a stable routing distribution that supervises the student router during training.

3.1. Preliminary: MoE Routing

In Transformer architectures, the MoE layer is typically introduced as a replacement for the feed-forward network (FFN) within each block. Each MoE layer consists of E experts $\{f_e(\cdot)\}_{e=1}^{E}$, each implemented as an MLP with independent parameters. Given $\mathbf{h} \in \mathbb{R}^{N\times D}$, D-dimensional representations for N patches, the router network $R(\cdot)$ produces pre-softmax logits $\mathbf{z} = R(\mathbf{h}) \in \mathbb{R}^{N\times E}$, and the gating probabilities are obtained as

$$\mathbf{p} = \mathrm{softmax}(\mathbf{z}). \tag{1}$$

For each token representation $\mathbf{h}_b \in \mathbb{R}^d$, the router produces gating probabilities $\mathbf{p}_b = \mathrm{softmax}(R(\mathbf{h}_b)) \in \mathbb{R}^E$. The top-$K$ experts with the highest probabilities are selected, and their outputs are aggregated as a weighted sum:

$$\mathrm{MoE}(\mathbf{h}_b) = \sum_{e\in \mathrm{TopK}(\mathbf{p}_b)} \mathbf{p}_{b,e}\, f_e(\mathbf{h}_b), \tag{2}$$

where $\mathrm{TopK}(\mathbf{p}_b)$ denotes the indices of the K experts with the highest gating probabilities for token b.

Algorithm 1: Training procedure of TGR-MoE

Input: Pretrained dense teacher model $\mathcal{M}_{\text{teacher}}$, MoE student model $\mathcal{M}_{\text{student}}$, dataset $\mathcal{D}$, set of MoE layers $\mathcal{S}_{\text{MoE}}$
Output: Trained student model $\mathcal{M}_{\text{student}}$
for *each mini-batch* $(x, y) \in \mathcal{D}$ **do**
  Forward $\mathcal{M}_{\text{student}}$ and obtain routing probabilities $\{\mathbf{p}^{(i)}\}_{i\in\mathcal{S}_{\text{MoE}}}$;
  Extract teacher intermediate features $\{\mathbf{h}_t^{(i)}\}$;
  Compute teacher routing probabilities:

$$\mathbf{p}_t^{(i)} = \mathrm{softmax}\Big(R_t^{(i)}(\mathbf{h}_t^{(i)})\Big).$$

  Compute teacher router loss w.r.t. Eqn. 8;
  Compute router distillation loss w.r.t. Eqn. 9;
  Compute student final loss w.r.t. Eqn. 10;
  Update $\mathcal{M}_{\text{student}}$ and R_t using $\mathcal{L}_{\text{student}}$ and $\mathcal{L}_{\text{teacher}}$;
end

Load-Balancing Loss. A major challenge in MoE training is expert imbalance, where one or a few experts dominate computation while others remain inactive. Following VMoE [32], an auxiliary load-balancing loss is introduced to encourage even expert utilization. Given a patch of N tokens with routing probabilities $\mathbf{p} \in \mathbb{R}^{N\times E}$, the per-expert importance is defined as

$$\mathrm{Imp}_e(\mathbf{p}) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{p}_{i,e}. \tag{3}$$

The load loss $\mathcal{L}_{\text{load}}(\mathbf{p})$ minimizes the coefficient of variation of the expert importances:

$$\mathcal{L}_{\text{load}}(\mathbf{p}) = \left(\frac{\mathrm{std}(\mathrm{Imp}(\mathbf{p}))}{\mathrm{mean}(\mathrm{Imp}(\mathbf{p}))}\right)^2 \propto \mathrm{var}(\mathrm{Imp}(\mathbf{p})), \tag{4}$$

where $\mathrm{Imp} = \{\mathrm{Imp}_e\}_{e=1}^{E}$. In VMoE, two balancing terms are introduced—the *load loss* and the *importance loss*—which together promote uniform expert utilization. For simplicity, we refer to their combination as the overall load loss $\mathcal{L}_{\text{load}}(\mathbf{p})$ throughout this paper.

The overall objective for the VMoE model is expressed as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \sum_{i\in\mathcal{S}_{\text{MoE}}} \lambda_{\text{load}}\, \mathcal{L}_{\text{load}}(\mathbf{p}^{(i)}), \tag{5}$$

where λ_{load} is a balancing coefficient for the load loss, $\mathcal{S}_{\text{MoE}}$ denotes the set of layers containing MoE modules, and $\mathbf{p}^{(i)}$ represents the routing distribution produced by the router at the i-th MoE layer.

3.2. Teacher-Guided Routing for MoE (TGR-MoE)

Teacher Router Construction To leverage a non-MoE dense teacher model to guide the student's routers, we add auxiliary *teacher routers*. The teacher routers are dedicated to providing pseudo-supervision for student routers, and do not perform actual routing in the sense of MoE.

We first extract intermediate representations $\mathbf{h}_t$ from a pretrained dense teacher model. Using these features, we define a teacher router $R_t(\cdot)$ that outputs the expert assignment probabilities:

$$\mathbf{p}_t = \mathrm{softmax}(R_t(\mathbf{h}_t)) \in \mathbb{R}^{N\times E}. \tag{6}$$

The teacher router is optimized to produce stable and well-distributed routing using two loss terms: the load-balancing loss $\mathcal{L}_{\text{load}}(\mathbf{p})$ and the entropy loss $\mathcal{L}_{\text{ent}}(\mathbf{p})$. The load-balancing loss encourages uniform utilization of experts and prevents underused experts, while the entropy loss prevents the router from collapsing into an overly uniform distribution, thereby encouraging confident and distinct expert assignments. The entropy loss is defined as:

$$\mathcal{L}_{\text{ent}}(\mathbf{p}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{e=1}^{E}\mathbf{p}_{i,e}\log\mathbf{p}_{i,e}. \tag{7}$$

The pretrained dense teacher model is frozen, and only the teacher router is optimized. We optimize the teacher router using a subset of training samples $\mathcal{S}_{\text{MoE}}$ to obtain a stable and balanced routing distribution. The objective function is defined as:

$$\mathcal{L}_{\text{teacher}} = \sum_{i\in\mathcal{S}_{\text{MoE}}}\left(\lambda_{\text{load}}\mathcal{L}_{\text{load}}(\mathbf{p}_t^{(i)}) + \lambda_{\text{ent}}\mathcal{L}_{\text{ent}}(\mathbf{p}_t^{(i)})\right), \tag{8}$$

where λ_{load} and λ_{ent} are balancing coefficients. This optimization does not aim to minimize the downstream task loss directly. Instead, it constructs a teacher router that promotes desirable routing behaviors in Sparse MoE models, such as balanced expert utilization and the avoidance of overly uniform assignments. It leverages the pretrained teacher's semantically structured feature space, whose rich and organized representations provide a strong prior for guiding the router toward consistent and meaningful expert assignment.

Teacher-Guided Router Training The student router $\mathcal{R}_{\text{student}}$ in the MoE model learns to imitate the teacher router's routing distribution. Let $\mathbf{p}_{\text{t}}$ and $\mathbf{p}$ denote the output distributions of the teacher and student routers, respectively. We define the distillation loss as:

$$\mathcal{L}_{\text{distill}}(\mathbf{p}, \mathbf{p}_t) = \mathrm{KL}(\mathrm{stopgrad}(\mathbf{p}_t) \parallel \mathbf{p}), \tag{9}$$

where $\mathrm{KL}(\cdot\|\cdot)$ denotes the Kullback–Leibler divergence [20] and $\mathrm{stopgrad}(\cdot)$ indicates the stop-gradient operation, which detaches the teacher's output from the computation graph to prevent gradient backpropagation into the teacher network.

The overall objective for the MoE model is then expressed as:

$$\mathcal{L}_{\text{student}} = \mathcal{L}_{\text{task}} + \frac{\lambda_{\text{distill}}}{|\mathcal{S}_{\text{MoE}}|}\sum_{i\in\mathcal{S}_{\text{MoE}}}\mathcal{L}_{\text{distill}}(\mathbf{p}^{(i)}, \mathbf{p}_t^{(i)}) \tag{10}$$

where $\mathcal{L}_{\text{task}}$ is the main task loss (*e.g.*, classification), and λ_{distill} is a hyperparameter controlling the distillation strength.

In practice, we jointly train the teacher router together with the student MoE model. Since the teacher's backbone is frozen and only the router receives its own auxiliary losses, the teacher routing converges quickly and remains stable throughout training. We also observed that pretraining and freezing the teacher router in advance yields nearly identical final accuracy. Therefore, for simplicity and efficiency, we adopt the joint training scheme in all experiments. The overall training algorithm for TGR-MoE is summarized in Figure 2 and Algorithm 1.

4. Results

4.1. Implementation Details

We adopted the DeiT [39] architecture as the backbone to ensure a fair comparison, since our proposed TGR-MoE employs a teacher-model-based training paradigm and DeiT offers a well-established and efficient framework for incorporating teacher signals in vision transformers. To maintain fairness, all experiments, including both the proposed TGR-MoE and the baseline VMoE [32] employed the same DeiT-style distillation strategy applied to the final classification layer. The MoE configuration followed the original VMoE setup. The detailed settings are provided in Table 1. As the teacher model, we selected DeiT-III [40] pretrained on ImageNet-21K [31] because it achieves strong top-1 accuracy while sharing the same architectural family as the student models. This architectural alignment reduces the inductive gap between teacher and student, facilitating more stable and effective knowledge transfer.

Following the DeiT architectures, which consist of 12 transformer layers, the 8th, 10th, and 12th layers were replaced with MoE layers across all model variants (Tiny, Small, and Base). Since the DeiT-III teacher model does not include a distillation token, we use the routing output associated with the teacher's CLS token as the distillation target for the student router. All models were trained on ImageNet-1K [35] with an input resolution of 224×224 and fine-tuned on CIFAR-10, CIFAR-100 [19], and Oxford-IIIT Pets [27] in later experiments. We adopted AdamW optimizer with cosine learning rate scheduling, and data augmentations (RandAugment [3], Mixup [45], CutMix [44])

Table 1. Model configurations and loss coefficients used in all experiments. E denotes the number of experts.

Model	E	Teacher	Accuracy (Teacher)
Tiny	16	DeiT-III-Small	83.1%
Small, Base	8	DeiT-III-Base	85.7%
Loss coefficients: $\mathcal{L}_{\text{load}} = 0.005$, $\mathcal{L}_{\text{distill}} = 5.0$, $\mathcal{L}_{\text{ent}} = 0.005$			

applied consistently across all methods. Experiments were conducted on 2× or 4× H100 GPUs depending on the model size. Note that we trained all student models from scratch. Some experiments were run only on the Tiny and Small variants due to computational limits. Detailed hyperparameter settings are provided in Supplementary Section A.

4.2. Quantitative Evaluation

We first conducted pre-training on ImageNet-1K to compare the performance of ViT, VMoE, and our proposed TGR-MoE. In addition to standard VMoE, we evaluated Expert Choice MoE [48], z-loss regularization [49], and SoftMoE [29]. To ensure a fair comparison, Expert Choice MoE and SoftMoE were configured so that each expert processed, on average, the same number of tokens.

As shown in Table 2, TGR-MoE achieves consistently strong performance across all model scales. It reliably surpasses ViT and VMoE, and is competitive with or better than Expert Choice MoE and SoftMoE on most datasets. For example, on ImageNet-1K, TGR-MoE improves the Tiny model's Top-1 accuracy from 77.85% to 78.78% and the Small model from 82.63% to 83.34% under the $K{=}1$ setting. These gains demonstrate that providing stable global guidance during routing yields benefits beyond those achieved by existing capacity-balancing or continuous-routing approaches.

We further applied the same training scheme during fine-tuning, where TGR-MoE continued to show superior performance on downstream datasets such as CIFAR-100. These results demonstrate that teacher-guided routing leads to improved performance and more effective expert utilization in both pre-training and fine-tuning. Additional analysis and comparisons are provided in Supplementary Section C.

Effect of TGR-MoE during fine-tuning. To further examine whether the routing structure learned during pre-training is preserved during transfer, we conducted fine-tuning experiments on CIFAR-10, CIFAR-100, and Oxford-IIIT Pets, using Tiny, Small, and Base models pretrained on ImageNet-1K. We compared two configurations: (i) fine-tuning a pre-trained TGR-MoE model under the standard VMoE objective, and (ii) continuing to apply TGR-MoE during fine-tuning.

As shown in Table 3, maintaining TGR-MoE during fine-tuning consistently achieves the highest accuracy (86.95%, 90.26%, and 91.07% for Tiny, Small, and Base, respectively), whereas using the pre-trained TGR-MoE only for initialization yields limited gains (86.15%, 89.18%, and 90.05%) compared to the VMoE baselines (86.14%, 88.68%, and 89.04%). Despite strong pre-training, the performance converges close to the baseline when TGR-MoE is not applied during fine-tuning, suggesting that the routing knowledge acquired during pre-training is difficult to retain without explicit guidance. Applying TGR-MoE throughout fine-tuning helps preserve and refine this routing knowledge, leading to more stable expert utilization and consistent transfer performance.

4.3. Effect of Varying Number of Experts

We further investigated how the number of experts E influences model performance and training stability. Increasing E naturally raises the sparsity of the model, as only a small subset of experts is activated for each token. However, this heightened sparsity also amplifies the difficulty of the routing task, often resulting in unstable or suboptimal expert utilization in conventional MoE training. To assess whether the proposed method can sustain stable learning under such sparse regimes, we evaluated Tiny-scale models with $E \in \{4, 8, 16, 32, 64, 128, 256\}$. The results are summarized in Table 4.

As E increases, both VMoE and TGR-MoE exhibit consistent performance improvements, confirming that expanding expert capacity enhances representational power. However, the gains realized by VMoE taper off at larger expert counts, whereas TGR-MoE maintains steady improvements across all configurations—achieving a +1.07% boost at $E{=}16$ and sustaining meaningful gains even at $E{=}128$. These findings suggest that teacher-guided routing enables more reliable expert allocation as model capacity scales, thereby improving the overall scalability of sparse expert architectures.

5. Analytical Discussion

5.1. Analysis of Routing Consistency

To assess how routing behavior evolves during training and whether a method suppresses frequent changes in expert selection, we evaluate *routing agreement* of TGR-MoE and VMoE trained on ImageNet-1K in Table 2, which measures the proportion of tokens whose assigned expert matches a reference assignment.

As shown in Figure 3 (top), the proposed method maintains a consistently higher agreement with the final model throughout training. Around the midpoint of optimization, VMoE still differs from the final routing assignments for roughly 40% of inputs, indicating that its expert selection

Table 2. Comparison of Top-1 accuracy (%) across datasets and model scales. For V-MoE and the proposed TGR-MoE, results are reported as K=1 / K=2 (left / right) in top-K routing. Models with K = 1 and K = 2 are trained independently. For all other baselines without a slash, the reported values correspond to K=1 or an equivalent computational cost.

Model Size	Model	Top-1 Accuracy (%)			
		ImageNet-1K [35]	CIFAR-10	CIFAR-100 [19]	Pets [27]
Tiny	ViT (dense)	74.62	97.78	85.43	89.86
	VMoE [32]	77.85 / 78.21	97.98 / 97.72	86.20 / 85.75	89.82 / 90.17
	VMoE w/ z-loss [49]	77.99	97.84	86.13	90.15
	Expert Choice MoE [48]	77.82	97.83	86.60	91.21
	SoftMoE [29]	**79.31**	97.99	86.80	**91.91**
	TGR-MoE (ours)	78.78 / 79.24	**98.27 / 98.37**	**87.03 / 86.87**	91.78 / 91.63
Small	ViT (dense)	81.74	98.35	88.60	92.75
	VMoE	82.63 / 82.81	98.51 / 98.68	88.68 / 88.98	93.31 / 93.10
	VMoE w/ z-loss	82.42	98.62	89.36	93.04
	Expert Choice MoE	82.09	98.55	89.21	93.69
	SoftMoE	82.76	98.58	88.45	93.32
	TGR-MoE (ours)	**83.34 / 83.68**	**98.90 / 98.93**	**90.26 / 90.48**	**93.79 / 94.14**
Base	ViT (dense)	84.02	98.76	89.72	93.65
	VMoE	83.97 / 84.08	98.66 / 98.78	89.04 / 89.26	93.79 / 93.45
	VMoE w/ z-loss	84.03	98.81	89.99	94.21
	TGR-MoE (ours)	**85.46 / 85.34**	**98.99 / 99.05**	**91.07 / 91.07**	**94.63 / 94.97**

Table 3. Effect of applying TGR-MoE during fine-tuning (K=1, CIFAR-100). Maintaining TGR-MoE during fine-tuning yields the highest accuracy across all model scales, showing that explicit routing guidance preserves pre-trained routing knowledge and stabilizes transfer learning.

Setting	Top-1 Accuracy (%)		
	Tiny	Small	Base
ViT	85.43	88.60	89.72
VMoE	86.14	88.68	89.04
TGR-MoE (pretrained only)	86.15	89.18	90.05
TGR-MoE (during fine-tuning)	**86.95**	**90.26**	**91.07**

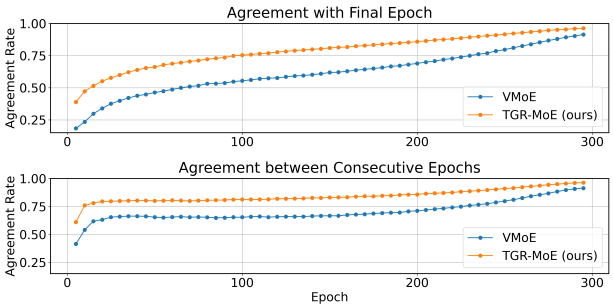

Figure 3. Comparison of routing consistency during training. (**Top**) Agreement rate with the final epoch. (**Bottom**) Agreement rate between consecutive epochs, evaluated every 5 epochs. The proposed method significantly reduces routing fluctuations and achieves more stable expert selection, averaged over all MoE layers.

continues to shift even in later stages. In contrast, our method surpasses 70% agreement within the first 50 epochs and remains comparatively stable thereafter, demonstrating that teacher-guided routing enables the router to settle into a consistent usage pattern much earlier. A similar trend is observed when examining agreement between consecutive epochs (Figure 3, bottom). Our method maintains a high and steady level of agreement, approximately 0.8 throughout training. VMoE, however, exhibits pronounced oscillations, especially in the early training phase, with agreement frequently dropping to 0.5–0.6. Overall, these results indicate that teacher-guided routing effectively suppresses routing fluctuation.

To further examine whether the routing structure learned during pre-training is preserved after transfer, we measured the agreement rate of router outputs before and after fine-tuning on CIFAR-100. For the conventional VMoE, agreement dropped to only 50.56%, indicating that fine-tuning substantially altered the routing configuration. In contrast, the proposed TGR-MoE maintained a much higher agreement of 73.75%, suggesting that the teacher-guided routing mechanism effectively preserves the pre-trained routing behavior and provides a more consistent initialization for downstream adaptation.

5.2. Interaction of Task and Distillation Losses

While TGR-MoE trains the student router using both task and distillation losses, the interaction between these two

Table 4. Comparison of Top-1 accuracy with different numbers of Experts (Tiny model, ImageNet-1K). The advantage of TGR-MoE becomes more evident as the number of experts increases.

Number of experts	1 (Dense)	4	8	16	32	64	128
VMoE	74.62%	76.18%	77.39%	77.85%	78.41%	78.74%	79.38%
TGR-MoE (ours)	–	**76.59% (+0.41%)**	**77.81% (+0.42%)**	**78.78% (+1.07%)**	**79.35% (+0.94%)**	**79.95% (+1.21%)**	**80.36% (+0.98%)**

Table 5. Comparison of different **TGR-MoE training variants** using the Tiny model with 8 experts on ImageNet-1K. While distillation alone provides sufficient supervision for stable routing, applying it primarily in the early stage and switching to task optimization later yields the best final performance.

Training Configuration	Accuracy (%)
Distillation only	77.83
Distillation (first half) + Task	**78.13**
Distillation + Task	77.81

signals remains unclear. We hypothesize that task gradients may conflict with the teacher-guided routing signal, especially in the early stages when routing decisions are still unstable. Meanwhile, it was reported that applying distillation throughout training can impede late-stage convergence, and that restricting it to earlier epochs often improves final performance [30]. Motivated by this, we investigate how the balance between task-driven and distillation-driven supervision shapes router learning in TGR-MoE. To study this, we compare three training configurations: (1) distillation-only router training, (2) distillation applied only during the first half of training, and (3) the standard TGR-MoE setup with both losses active throughout.

The results are illustrated in Table 5. Two observations emerge. First, and surprisingly, explicit task supervision is not essential for router optimization: distillation alone achieves competitive accuracy (77.83%), indicating that the teacher's routing distribution provides sufficiently strong guidance. Second, applying distillation only in the early phase yields the best performance (78.13%), suggesting that teacher-guided signals are most beneficial when routing is volatile, while task gradients become more effective once expert selection has stabilized. These findings reveal that the role of distillation in router training is inherently *phase-dependent*: strong early guidance stabilizes routing, whereas relaxing distillation pressure later improves task convergence.

5.3. Analysis of Upper-Bound Routing

To understand the potential benefits of routing based directly on teacher features, we analyze an upper-bound configuration in which the router itself is trained on top of a pretrained dense teacher backbone and used as the routing mechanism at inference time. Although this setup is not feasible in practice as the teacher model cannot be deployed during inference, it serves as an oracle that reveals the maximum achievable benefit when routing decisions are made from the teacher's rich feature representations rather than those of the student. Concretely, we train the teacher router using *task* and load-balancing losses. Unlike our proposed TGR-MoE, this configuration does not distill teacher routing into a student router; instead, the teacher router directly performs all routing decisions during both training and inference. This allows us to evaluate the upper bound of expert allocation quality that a student router could aspire to imitate.

Table 6 reports the results for the Tiny model with 8 experts. Using the teacher router for inference yields the highest accuracy (80.19%), suggesting that routing guided by the teacher's structured and semantically rich feature space can substantially improve expert utilization. This highlights that current student-side MoE routers still leave significant room for improvement in how effectively they leverage expert capacity. In contrast, the student router trained solely through imitation of the teacher distribution shows a substantial performance drop (74.84%). This indicates that it cannot fully reproduce the complex routing behaviors available to the teacher, likely due to limited representational capacity, accumulated approximation errors across layers, and the fact that it never performs routing during training and therefore cannot adjust its behavior based on task-driven signals. Overall, while direct teacher-guided routing represents an unattainable but informative upper bound, TGR-MoE provides a practical middle ground: it transfers meaningful routing structure from the teacher while remaining deployable at inference time, achieving stable and effective routing without requiring access to the teacher model. We provide additional details of this experiment in Supplementary Section B.

5.4. Further Analysis of Teacher-Guided Routing

To better understand how teacher knowledge stabilizes routing, we analyze the behavior of teacher-guided routing in detail. We first study which layer of the teacher model provides the most effective routing supervision, and then evaluate how faithfully the student router imitates the teacher router.

Table 6. Evaluation of the effectiveness of incorporating teacher knowledge into routing. Teacher-routed inference performs routing directly with the pretrained teacher during evaluation, achieving the highest accuracy and illustrating the potential benefit of teacher-informed routing. Student-routed inference (w/ distillation) learns to imitate the teacher's routing but performs inference without the teacher model, showing a performance drop due to limited adaptation capacity. TGR-MoE achieves a balance between these settings, effectively transferring teacher knowledge while remaining teacher-free at inference.

Configuration	Accuracy (%)
VMoE baseline	77.39
TGR-MoE (ours)	77.81
Student-routed inference (w/ distillation)	74.84
Teacher-routed inference (upper bound)	80.19

Which teacher layer provides the best guidance?

We evaluate several choices of teacher feature layers for constructing routing guidance, using the Tiny model with 8 experts. When the routing signal is generated from the teacher's final-layer features, accuracy drops substantially to 75.83%, considerably lower than our proposed method (77.81%) and even below the standard VMoE baseline. This suggests that the high-level representations of a pretrained teacher are too abstract and task-specialized to serve as effective routing cues for the student. In contrast, layer-aligned intermediate features provide a better structural match, yielding a more compatible and effective supervision signal for guiding expert allocation.

Agreement between teacher and student routers.

We next measure how closely the student router replicates teacher routing behavior by computing the top-1 expert selection agreement between the two routers across different MoE layers. As reported in Table 7, the agreement consistently improves with model scale, indicating that larger student models have greater capacity to approximate the teacher's routing boundaries. However, perfect alignment is not achieved even in the Base model, reflecting the inherent capacity gap between the student router and the teacher's richer representation space.

Across layers, we observe a monotonic decrease in agreement toward deeper stages. This is expected: deeper student representations naturally deviate more from the teacher's, and forcing strict alignment in these layers can conflict with task optimization. Indeed, teacher routing, while beneficial as an upper bound, is not necessarily optimal for the student due to representational mismatch. Thus, a balance between imitation and adaptation is required: the student should leverage teacher-informed structure where beneficial while retaining flexibility in later layers. This trade-off is precisely what TGR-MoE embodies, enabling effective routing transfer without over-constraining the student model.

Table 7. Top-1 expert selection agreement between teacher and student routers across different layers. The agreement improves with model size but decreases in deeper layers, suggesting that the student router approximates but does not perfectly replicate the teacher's routing behavior.

Model	Agreement (%)		
	Layer 8	Layer 10	Layer 12
Tiny	74.81	70.60	53.43
Small	80.01	75.49	59.87
Base	81.27	77.06	67.15

6. Conclusion

In this work, we introduced TGR-MoE, a teacher-guided routing framework in which a lightweight router attached to a pretrained dense teacher provides stable and informative routing distributions as supervision for the student MoE router. Through extensive experiments on ImageNet-1K and multiple downstream benchmarks, we demonstrated that TGR-MoE consistently improves accuracy across model scales and remains effective as the number of experts increases, highlighting its scalability advantages over standard VMoE and other MoE variants. Our analyses further revealed that TGR-MoE substantially stabilizes routing during training, preserves routing patterns more faithfully during fine-tuning, and reduces fluctuations that typically hinder expert specialization. Importantly, the framework requires access to the teacher model only during training, incurring no additional inference cost. Overall, our findings indicate that effective MoE routing requires supervisory signals beyond task-driven gradients, and that incorporating teacher-informed routing priors offers a simple yet powerful mechanism for achieving stable, scalable, and high-performing expert-based architectures.

Limitations and Future Work. Our experiments were primarily focused on image classification tasks; extending TGR-MoE to natural language and multimodal settings remains an important direction for future work. Since MoE models are highly sensitive to data scale, future studies should also examine the generalization and scalability of TGR-MoE on larger datasets. We believe that such extensions could further validate and enhance the potential of TGR-MoE in large-scale, real-world applications.

7. Acknowledgement

This work was supported by DENSO IT LAB Recognition, Control, and Learning Algorithm Collaborative Research Chair. All experiments were carried out using the TSUBAME4.0 supercomputer at Institute of Science Tokyo.

References

[1] Ruisi Cai, Yeonju Ro, Geon-Woo Kim, Peihao Wang, Babak Ehteshami Bejnordi, Aditya Akella, and Zhangyang Wang. Read-me: refactorizing llms as router-decoupled mixture of experts with system co-design. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2024. Curran Associates Inc. 2

[2] Wenyan Cong, Hanxue Liang, Peihao Wang, Zhiwen Fan, Tianlong Chen, Mukund Varma, Yi Wang, and Zhangyang Wang. Enhancing nerf akin to enhancing llms: Generalizable nerf transformer with mixture-of-view-experts. In *ICCV*, 2023. 2

[3] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *NeurIPS*, 2020. 4

[4] Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. StableMoE: Stable routing strategy for mixture of experts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7085–7095, Dublin, Ireland, 2022. Association for Computational Linguistics. 2

[5] Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 1280–1297. Association for Computational Linguistics, 2024. 2

[6] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. 2

[7] Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. GLaM: Efficient scaling of language models with mixture-of-experts. In *ICML*, 2022. 2

[8] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research (JMLR)*, 23(1), 2022. 1, 2

[9] Zhida Feng, Zhenyu Zhang, Xintong Yu, Yewei Fang, Lanxin Li, Xuyi Chen, Yuxiang Lu, Jiaxiang Liu, Weichong Yin, Shikun Feng, Yu Sun, Li Chen, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-vilg 2.0: Improving text-to-image diffusion model with knowledge-enhanced mixture-of-denoising-experts. In *CVPR*, pages 10135–10145, 2023. 1, 2

[10] Ziyao Guo, Haonan Yan, Hui Li, and Xiaodong Lin. Class attention transfer based knowledge distillation. In *CVPR*, pages 11868–11877, 2023. 2

[11] Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed H. Chi. Dselect-k: differentiable selection in the mixture of experts with applications to multi-task learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2021. Curran Associates Inc. 2

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 2

[13] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *NeurIPS*, Red Hook, NY, USA, 2022. Curran Associates Inc. 2

[14] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991. 2

[15] Yash Jain, Harkirat Behl, Zsolt Kira, and Vibhav Vineet. Damex: Dataset-aware mixture-of-experts for visual understanding of mixture-of-datasets. In *NeurIPS*, 2023. 2

[16] 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, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. 2

[17] Masahiro Kada, Ryota Yoshihashi, Satoshi Ikehata, Rei Kawakami, and Ikuro Sato. Robustifying routers against input perturbations for sparse mixture-of-experts vision transformers. *IEEE Open Journal of Signal Processing*, 6:276–283, 2025. 2

[18] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 2

[19] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009. 2, 4, 6

[20] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1): 79–86, 1951. 4

[21] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In *ICLR*, 2020. 1, 2

[22] Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. Base layers: Simplifying training of large, sparse models. In *ICML*, 2021. 2

[23] Junzhuo Li, Bo Wang, Xiuze Zhou, and Xuming Hu. Dynamic expert specialization: Towards catastrophic forgetting-free multi-domain MoE adaptation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18489–18504, Suzhou, China, 2025. Association for Computational Linguistics. 3

[24] Zhenxing Mi and Dan Xu. Switch-NeRF: Learning scene decomposition with mixture of experts for large-scale neural radiance fields. In *ICLR*, 2023. 2

[25] Mohammed Muqeeth, Haokun Liu, and Colin Raffel. Soft merging of experts with adaptive routing. *CoRR*, abs/2306.03745, 2023. 2

[26] Steven Nowlan and Geoffrey E Hinton. Adaptive soft weight tying using gaussian mixtures. In *NeurIPS*, 1991. 2

[27] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 2, 4, 6

[28] Joan Puigcerver, Rodolphe Jenatton, Carlos Riquelme, Pranjal Awasthi, and Srinadh Bhojanapalli. On the adversarial robustness of mixture of experts. In *NeurIPS*, 2022. 2

[29] Joan Puigcerver, Carlos Riquelme Ruiz, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. In *ICLR*, 2024. 1, 2, 5, 6

[30] Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. DeepSpeed-MoE: Advancing mixture-of-experts inference and training to power next-generation AI scale. In *ICML*, pages 18332–18346. PMLR, 2022. 7

[31] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik. Imagenet-21k pretraining for the masses. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. 4

[32] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. In *NeurIPS*, 2021. 1, 2, 3, 4, 6

[33] Stephen Roller, Sainbayar Sukhbaatar, Jason Weston, et al. Hash layers for large sparse models. In *NeurIPS*, 2021. 2

[34] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015. 2

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014. 2, 4, 6

[36] Michael E. Sander, Joan Puigcerver, Josip Djolonga, Gabriel Peyré, and Mathieu Blondel. Fast, differentiable and sparse top-k: a convex analysis perspective. In *ICML*, 2023. 2

[37] 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 *ICLR*, 2017. 1, 2

[38] Yuge Shi, Siddharth N, Brooks Paige, and Philip Torr. Variational mixture-of-experts autoencoders for multi-modal deep generative models. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 2

[39] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers amp; distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021. 2, 4

[40] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, page 516–533, Berlin, Heidelberg, 2022. Springer-Verlag. 4

[41] Mathurin VIDEAU, Alessandro Leite, Marc Schoenauer, and Olivier Teytaud. Mixture of experts for image classification: What's the sweet spot? *TMLR*, 2025. 1

[42] Ziteng Wang, Jun Zhu, and Jianfei Chen. Remoe: Fully differentiable mixture-of-experts with relu routing. In *ICLR*, 2025. 2

[43] Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. Openmoe: an early effort on open mixture-of-experts language models. In *ICML*. JMLR.org, 2025. 2

[44] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 4

[45] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 4

[46] Yihua Zhang, Ruisi Cai, Tianlong Chen, Guanhua Zhang, Huan Zhang, Pin-Yu Chen, Shiyu Chang, Zhangyang Wang, and Sijia Liu. Robust mixture-of-expert training for convolutional neural networks. In *ICCV*, 2023. 2

[47] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *CVPR*, pages 11953–11962, 2022. 2

[48] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, zhifeng Chen, Quoc V Le, and James Laudon. Mixture-of-experts with expert choice routing. In *NeurIPS*, 2022. 2, 5, 6

[49] 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. 1, 2, 5, 6