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

Supplementary Material

A. Hyperparameter Details

Table S1 summarizes all hyperparameters used in our experiments. Unless otherwise specified, we use identical optimizer settings, data augmentations, training schedules, and regularization across all baselines to ensure fair comparison. Architectural configurations, such as depth and sizes of MLPs, follow the standard DeiT [9] family. For all MoE variants, we use the same configuration as VMoE [7]. For SoftMoE [6] and Expert-Choice MoE [13], we set the expert capacity (slots) so that each expert processes approximately K/E of the tokens (where E is the number of experts and K is the number of selected experts per token in VMoE), matching the computation profile of VMoE; the capacity factor is adjusted accordingly.

B. Details of the Experiment in Section 5.3

This experiment evaluates an upper-bound configuration in which the routing decisions are computed directly from the teacher's backbone features. Concretely, we attach a router to selected layers of the frozen teacher (backbone parameters are not updated). Because the teacher representations encode richer semantic knowledge than those of the randomly initialized student, the resulting routing is expected to be more stable and more specialized. This setup allows us to (i) assess whether routing learned from teacher features is indeed beneficial, and (ii) estimate the maximal accuracy achievable when expert selection is performed in the teacher's representation space, providing an upper bound for TGR-MoE.

Loss formulation. The teacher router is trained using the task loss and a load-balancing loss summed over all MoE layers:

$$\mathcal{L}_{\text{teacher}} = \mathcal{L}_{\text{task}} + \lambda_{\text{load}} \sum_{i \in \mathcal{S}_{\text{MoE}}} \mathcal{L}_{\text{load}}^{(i)}, \tag{S1}$$

which is identical to the standard VMoE objective, except that the input features are the frozen teacher representations.

In this upper-bound configuration, the student model does not perform routing during training. Its experts are trained solely through distillation from the teacher router. We therefore separate the losses applied to the student model as:

$$\mathcal{L}_{\text{student-router}} = \lambda_{\text{distill}} \sum_{i \in \mathcal{S}_{\text{MoE}}} \mathcal{L}_{\text{distill}}^{(i)}, \tag{S2}$$

$$\mathcal{L}_{\text{student}} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{student-router}}, \tag{S3}$$

which makes explicit that the student router receives no task gradient; it is optimized only via distillation. As a result, the student router is used for the first time at inference, leading to the observed performance gap reported in Table 5 of the main paper.

Difference from TGR-MoE. The architecture for this experiment is illustrated in Fig. S1. The primary difference from TGR-MoE lies in the *source of routing*: the teacher router directly selects experts during training and inference, whereas TGR-MoE uses the teacher router only for supervision, and the student router performs routing. Furthermore, unlike TGR-MoE, the student router in this upper-bound configuration receives no task loss and is therefore used for the first time only at inference. As shown in Table 5 of the main paper, this mismatch introduces a performance gap, which explains why the student model cannot fully reach the upper-bound accuracy.

In contrast, TGR-MoE eliminates this gap by always training with the student router. The router is supervised by the teacher routing prior during training, but expert selection is ultimately driven by the student router itself for both training and inference. As a result, TGR-MoE inherits the teacher's knowledge while maintaining consistency between training and inference, thereby avoiding the degradation observed in the upper-bound configuration.

C. Additional Analysis and Comparison

C.1. Training Dynamics of TGR-MoE versus VMoE Across Model Scales

To examine how the proposed TGR-MoE stabilizes expert selection and accelerates optimization, we examine the training dynamics of both TGR-MoE and VMoE during pre-training. In particular, we compare how quickly the models reduce loss and improve accuracy, as this reflects the stability of the routing decisions formed in the early stage of learning.

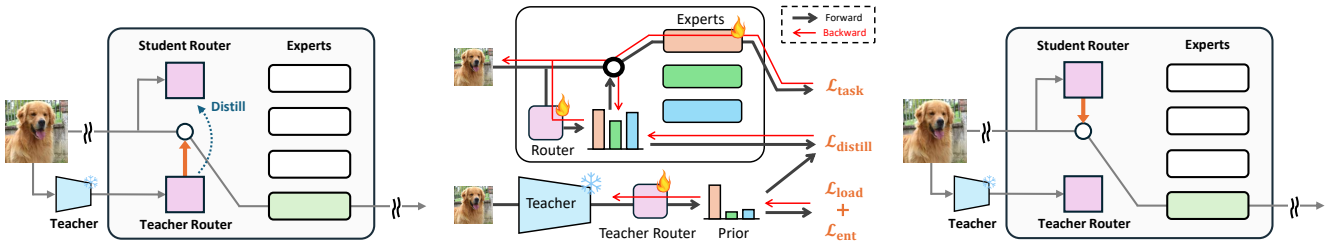
Figure S2 visualizes the training accuracy (top row) and training loss (bottom row) for the Tiny, Small, and Base models (left to right). Across all scales, TGR-MoE (blue) exhibits faster convergence than VMoE (orange), especially at the beginning of training. These results indicate that TGR-MoE produces more stable training dynamics by mitigating routing instability and accelerating the formation of well-behaved expert assignments.

C.2. Analysis of Expert Collapse

To directly assess whether the proposed method avoids degenerate routing, we report normalized routing entropy in Table S2. The results show that TGR-MoE maintains high

Table S1. Summary of hyperparameters used in all experiments.

Item	Setting
Architecture (DeiT-Tiny)	Hidden dim: 192, Heads: 3, MLP dim: 768, Layers: 12, MoE layers: {8,10,12}
Architecture (DeiT-Small)	Hidden dim: 384, Heads: 6, MLP dim: 1536, Layers: 12, MoE layers: {8,10,12}
Architecture (DeiT-Base)	Hidden dim: 768, Heads: 12, MLP dim: 3072, Layers: 12, MoE layers: {8,10,12}
Teacher architecture (DeiT-III-Small) [10]	Hidden dim: 384, Heads: 6, MLP dim: 1536, Layers: 12, Router layers: {8,10,12}
Teacher architecture (DeiT-III-Base)	Hidden dim: 768, Heads: 12, MLP dim: 3072, Layers: 12, Router layers: {8,10,12}
Image size	224 x 224
Training epochs	300 (ImageNet-1K [8]), 100 (CIFAR [3], Pets [5])
Learning rate	5×10^{-4}, warmup from 1×10^{-6} for first 5 epochs
Scheduler	Cosine annealing
Optimizer	AdamW [4] ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1.0 \times 10^{-8}$, weight decay $= 0.05$)
Data augmentation	RandAugment [1], Mixup [12], CutMix [11]
MoE noise	Gaussian noise (std 1.0) added to router logits
Load-balancing loss	$\lambda_{\text{load}} = 0.005$
Distillation loss	$\lambda_{\text{distill}} = 5.0$
Entropy regularization	$\lambda_{\text{ent}} = 0.005$ (teacher router)
z-loss [14] (comparison)	$\lambda_{\text{zloss}} = 1.0 \times 10^{-3}$



(a) Training with teacher-supervised routing (b) Teacher-routed inference (upper bound) (c) Student-routed inference (with distillation)

Figure S1. Comparison of training and inference configurations used in Section 5.3. (a) During training, routing supervision is provided by the teacher router while the teacher backbone remains frozen. (b) At inference, routing can be performed using the teacher router to obtain an upper-bound performance. (c) Alternatively, routing can be performed by the student router, which was trained only via distillation.

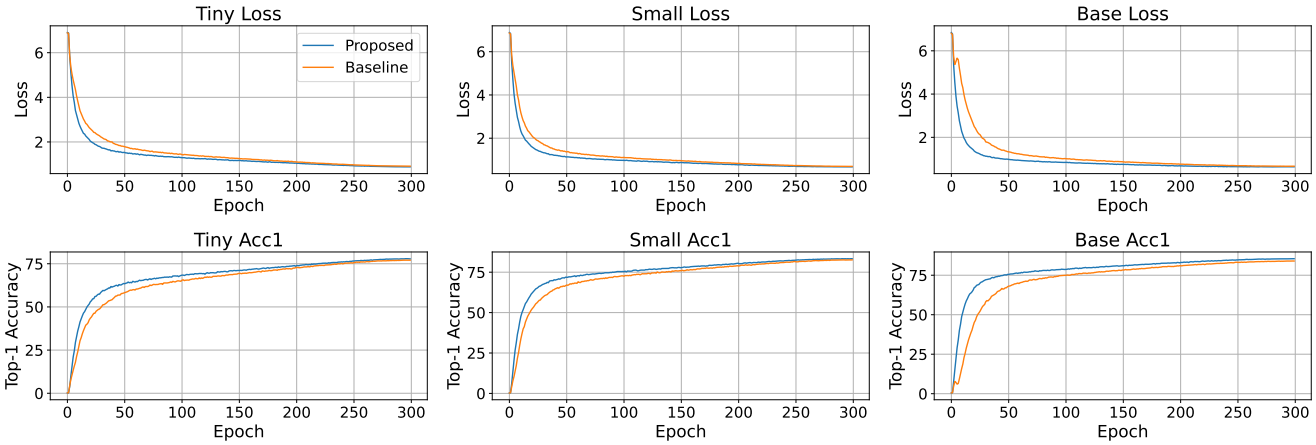


Figure S2. Training loss (top row) and training accuracy (bottom row) for the Tiny, Small, and Base models. The proposed TGR-MoE consistently converges faster and more stably than VMoE across all model scales.

Table S2. Training time (sec/epoch, single H100), trainable parameters (including teacher routers; excluding the teacher model), and normalized routing entropy.

Model	Time	Params	L8 ent.	L10 ent.	L12 ent.
VMoE-Ti	1006	19.23M	0.9098	0.7776	0.9489
TGR-MoE-Ti	1020	19.25M	0.9701	0.9627	0.9420
VMoE-S	2560	47.26M	0.9924	0.9848	0.9938
TGR-MoE-S	2570	47.28M	0.9772	0.9708	0.9878
VMoE-B	4930	186.53M	0.9677	0.9906	0.9888
TGR-MoE-B	4954	186.55M	0.9462	0.9789	0.9892

expert-utilization entropy across layers and model scales, indicating that the router does not collapse to a trivial or highly imbalanced assignment. This supports our claim that teacher-guided routing mitigates unstable expert specialization in sparse MoE training.

C.3. Additional Comparison with Routing Stabilization Baselines

We additionally compared TGR-MoE with StableMoE [2] in the Tiny setting. StableMoE was originally tested in NLP, but for verification purposes, we evaluated StableMoE using a Tiny model by copying the routing network, which is then frozen without distillation. StableMoE achieved 77.19% accuracy on ImageNet-1K (8 experts), while ours reached **77.81%** under the same setting.

C.4. Training Cost and Parameter Overhead

Table S2 also reports training time and trainable parameter counts. The additional cost of TGR-MoE is negligible: the method introduces only lightweight teacher routers and a distillation loss, resulting in almost unchanged training time and parameter count compared with VMoE. This indicates that the proposed improvement in routing stability does not come with a meaningful computational overhead.

References

[1] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *NeurIPS*, 2020. 2

[2] 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. 3

[3] 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] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 2

[5] 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

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

[7] 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

[8] 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

[9] 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. 1

[10] 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. 2

[11] 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. 2

[12] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 2

[13] 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. 1

[14] 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. 2