Sparse MoE Students for Efficient Knowledge Distillation

Anonymous CVPR submission

Paper ID *****

Abstract

We propose a compact and modular student architecture for knowledge distillation (KD) based on a sparse Mixture-of-Experts (MoE) framework. Unlike conventional dense student models, our design uses a set of lightweight, class-agnostic experts whose outputs are dynamically routed via input-conditioned gating. We systematically compare multiple routing strategies—soft, top-k, and attention-enhanced variants—and evaluate their impact across accuracy, computational cost, and expert utilization. Experiments on CIFAR-10 and CIFAR-100 show that sparse MoE students not only outperform dense baselines under similar or lower resource budgets, but also achieve superior parameter-efficiency and more consistent expert usage. Notably, attention-based routing consistently yields the best trade-off between accuracy and cost. Our findings highlight the structural benefits of modular sparse students in KD, offering improved generalization, interpretability, and efficiency without requiring class supervision.

1. Introduction

Knowledge distillation (KD) is a widely adopted paradigm for compressing large neural networks into compact student models [5, 15]. Traditionally, student architectures are designed as dense, monolithic networks trained to mimic a teacher's behavior via soft label supervision. While effective, such students lack structural flexibility and often struggle to balance efficiency, interpretability, and performance.

To address this, we propose a sparse Mixture-of-Experts (MoE) architecture as an alternative student model for KD. Our design replaces the single dense backbone with a set of lightweight, class-agnostic experts, whose outputs are dynamically routed based on input-conditioned gating. This structure enables conditional computation, modular specialization, and parameter sparsity—making it well-suited for efficient knowledge transfer.

An overview of our framework is shown in Figure 1. The input is passed through both a teacher network and the modular MoE student. The student comprises a gating network and a pool of lightweight experts. Predictions are formed via expert aggregation, and the student is trained using a combination of distillation and supervised losses.

Unlike class-specific MoE students [6, 17], our model is class-agnostic and does not rely on explicit supervision or task partitioning. We evaluate multiple routing strategies—soft, top-k, and attention-based—and analyze their effects on accuracy, efficiency, and scalability.

Contributions. Our main contributions are:

- We propose a sparse MoE student architecture for KD with class-agnostic expert modules and input-dependent routing.
- We conduct a systematic comparison of routing strategies and analyze expert count and selection trade-offs.
- We show that sparse MoE students outperform dense baselines and even the teacher under comparable or lower compute.

Related Work. Knowledge distillation (KD) was first introduced by Hinton et al. [5] and has since been extended by approaches that refine loss formulations [12, 15], utilize intermediate supervision [15].

Mixture-of-Experts (MoE) architectures have been widely explored to scale model capacity using conditional computation [3, 4, 8, 16], with follow-ups improving inference [18] and training stability [9, 14]. More recently, works like OpenMoE [11] and Mixtral [7] have brought MoE into large-scale language models, while MegaBlocks [14] and MoE-LoRA [1] explore efficient MoE training and fine-tuning.

However, these MoE models are typically heavyweight and not suited for compact student architectures. Prior MoE-based KD methods such as class-specialized KD [17] and Specific Expert Learning (SEL) [6] rely on class-conditional expert assignment or label supervision. Our work departs from this by proposing a class-agnostic MoE student architecture with lightweight experts and input-conditioned routing, better suited for efficient, modular distillation.

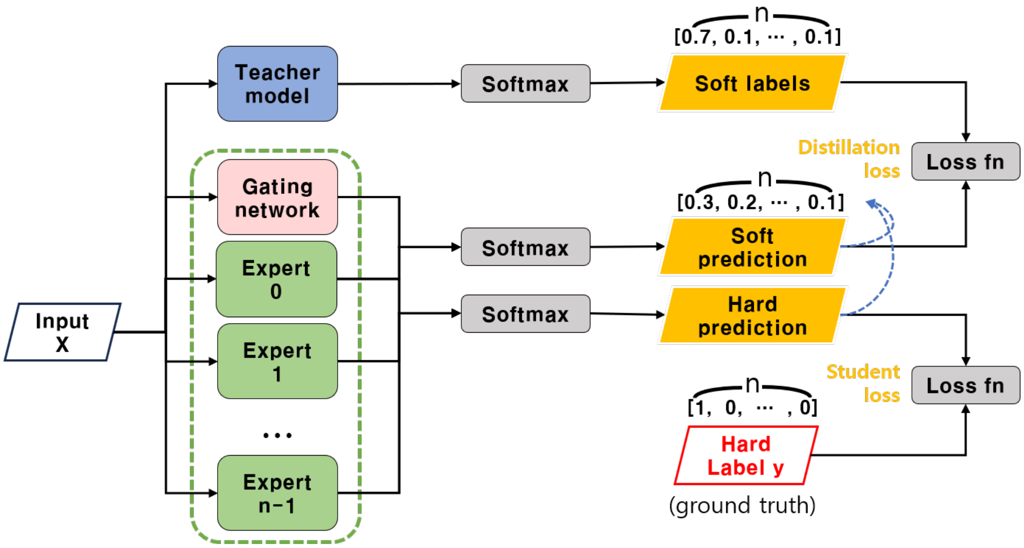

Figure 1. Overview of our knowledge distillation framework. The input is passed through both the teacher model and the sparse MoE student. The student uses a gating network to aggregate expert outputs, and is trained via both distillation loss from the teacher and standard cross-entropy with ground-truth labels.

2. Method

We propose a modular student model based on a sparse Mixture-of-Experts (MoE) architecture [16] for knowledge distillation. The design introduces a set of lightweight experts and a gating network that dynamically selects experts for each input. Unlike class-specific expert models [6, 17], all expert modules in our framework are class-agnostic and shared across categories. Routing decisions are made solely based on input features, enabling scalable and flexible expert specialization [9].

2.1. Mixture-of-Experts Student Architecture

Given an input image $x \in \mathbb{R}^{C\times H\times W}$, the MoE student consists of:

- A set of N expert networks $\{E_i\}_{i=1}^{N}$, where each expert $E_i(x) \in \mathbb{R}^K$ outputs logits over K classes.
- A gating network $G(x) \in \mathbb{R}^N$ that produces a softmax distribution over the N experts:

$$G(x) = \mathrm{softmax}(W_g f_g(x)), \tag{1}$$

  where $f_g(x)$ is a feature extractor and W_g is a linear projection layer.

The final prediction $\hat{y}$ is computed as a weighted sum over the expert outputs:

$$\hat{y} = \sum_{i=1}^{N} G_i(x) \cdot E_i(x). \tag{2}$$

This formulation corresponds to **soft routing** [10], where all experts are evaluated and their predictions are aggregated using the gating weights.

The overall training pipeline is visualized in Figure 1. The input passes through both the teacher and student networks, with gating-based expert selection and joint optimization of classification and distillation losses.

Compared to dense student models, our MoE framework introduces explicit modularity through expert decomposition [2], allowing for specialization. The gating network not only selects relevant experts, but also implements a soft mixture policy that enables ensemble-like behavior. Since experts are compact and structurally identical, the overall architecture remains lightweight and scalable [14].

2.2. Routing Variants

We explore three routing mechanisms, each trading off computational sparsity and selection expressiveness.

Soft Routing. All experts are evaluated and combined using the gating weights $G(x)$, as in Eq. (2). This yields a smooth ensemble that distributes computation and gradient flow across all experts. While we do not explicitly enforce expert balance, soft routing tends to encourage broader expert usage and stable training dynamics.

Top-k Routing. To promote sparsity, we select only the top-k experts with the highest gating scores. Let $\mathrm{TopK}(G(x), k)$ denote the top-k expert indices, and $\tilde{G}_i(x)$ the renormalized weights:

$$\hat{y} = \sum_{i\in \mathrm{TopK}(G(x),k)} \tilde{G}_i(x) \cdot E_i(x). \tag{3}$$

This strategy mimics hard expert selection [4], reducing computational cost. However, it can introduce expert imbalance or instability due to non-differentiable selection and sharp gating decisions [13].

Attention-Based Routing. We replace the standard MLP gating with a lightweight self-attention mechanism over spatial features of x [9]. Input features are flattened into a sequence and passed through an attention block, yielding a context-aware embedding. Gating scores are then computed as:

$$G(x) = \mathrm{softmax}(W_g \cdot \mathrm{Attn}(f_g(x))), \tag{4}$$

where $\mathrm{Attn}(\cdot)$ denotes the self-attention module. This input-aware routing allows expert selection to reflect spatial structure and global context.

We illustrate these routing strategies in Figure 2, which compares soft, attention-based, top-1, and top-2 routing in terms of expert activation and sparsity.

Empirically, attention-based routing achieves the best trade-off between accuracy and efficiency in our experiments. Figure 2 highlights how different mechanisms balance computation and flexibility in expert selection.

2.3. Knowledge Distillation Objective

We train the MoE student with a fixed teacher model $T(x) \in \mathbb{R}^K$. Following the standard KD paradigm [5], the loss combines distillation and cross-entropy objectives:

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{KD}} + (1 - \alpha) \cdot \mathcal{L}_{\text{CE}}, \tag{5}$$

where $\alpha \in [0, 1]$ balances the two losses.

The distillation term uses Kullback–Leibler divergence between softened outputs:

$$\mathcal{L}_{\text{KD}} = \mathrm{KL}\left(\mathrm{softmax}\left(\frac{T(x)}{T}\right) \middle\| \mathrm{log\_softmax}\left(\frac{\hat{y}}{T}\right)\right), \tag{6}$$

where T is the temperature parameter (fixed to $T = 2.0$).

The supervised loss is the standard cross-entropy:

$$\mathcal{L}_{\text{CE}} = -\sum_{j=1}^{K} y_j \log \mathrm{softmax}(\hat{y})_j. \tag{7}$$

Here, the temperature T smooths the logits to reveal relative probabilities over non-maximal classes, providing richer supervisory signals. The combined objective encourages the student to mimic teacher outputs while learning true labels.

In practice, this joint training stabilizes learning and helps experts form complementary decision boundaries. In sparse routing settings, where only a few experts are active per input, the soft targets further encourage specialization and consistent learning across modules.

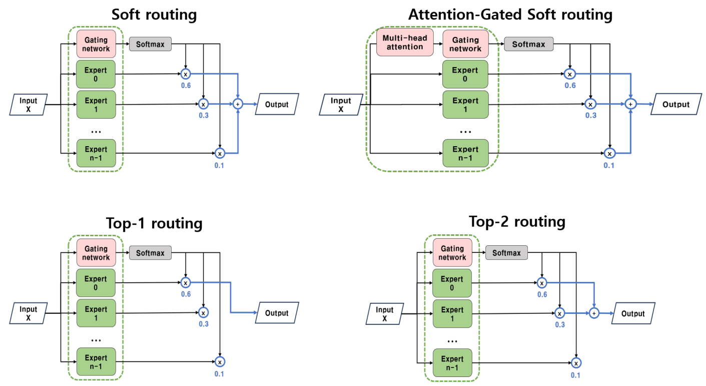

Figure 2. Overview of expert routing strategies in our sparse MoE student architecture. **(Top left)** Soft routing: all experts are evaluated and weighted by softmax scores. **(Top right)** Attention-based routing: the gating network uses spatially-aware self-attention. **(Bottom left)** Top-1 routing: only one expert is selected for each input. **(Bottom right)** Top-2 routing: the top-2 scoring experts contribute to prediction.

3. Experiments

3.1. Experimental Setup

We evaluate our sparse MoE student architecture on CIFAR-10 and extend to CIFAR-100 for generalization. The teacher is a ResNet-34 trained with cross-entropy loss. The dense student is a compact CNN with two convolutional layers and approximately 60K parameters. MoE students consist of 3, 5, or 10 lightweight experts, each identical in structure to the dense student.

We compare four routing strategies: *soft*, *attention-based*, *top-1*, and *top-2*. All models are trained for 5 epochs using knowledge distillation with a temperature of $T = 2.0$ and loss weighting factor $\alpha = 0.5$. Evaluation metrics include top-1 accuracy, parameter count, FLOPs (computed via `fvcore`), inference latency (per-image using CUDA timers), and peak memory usage.

3.2. Main Results on CIFAR-10

Table 1 presents selected CIFAR-10 results. The dense student outperforms the teacher (71.19% vs. 69.90%) despite being 340× smaller in parameter count and 9× cheaper in FLOPs. Among MoE students, **Top-1 (3)** achieves the highest accuracy (71.93%), while **Top-2 (5)** yields the best efficiency-accuracy trade-off with only 12.9M FLOPs. **Attention-based (5)** routing performs competitively across all metrics.

Figure 3 further confirms that sparse MoE students can outperform both teacher and dense baselines under lower computational cost. In particular, Top-2 (5) and Attention (5) models offer compelling trade-offs, supporting the claim that modular sparsity enables more scalable and in-

Table 1. Selected CIFAR-10 results. Full results across 13 MoE variants are in Appendix 6.

Model	Acc. (%)	Params	FLOPs
Teacher (ResNet-34)	69.90	21.3M	74.9M
Dense Student	71.19	62K	8.4M
Top-1 MoE (3)	**71.93**	183K	19.7M
Top-2 MoE (5)	71.62	317K	**12.9M**
Attn MoE (5)	71.84	303K	31.0M
Top-1 MoE (10)	64.74	629K	5.6M

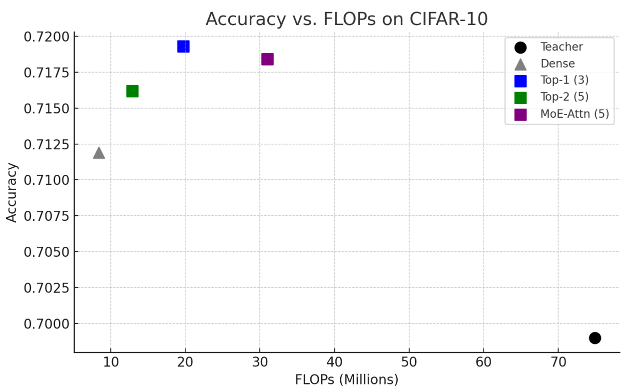

Figure 3. Accuracy vs. FLOPs on CIFAR-10. MoE students (squares) lie above the dense and teacher baselines (triangle and circle), defining a new Pareto frontier in the efficiency-accuracy space.

terpretable student designs.

3.3. Generalization to CIFAR-100

To evaluate generalization, we replicate all configurations on CIFAR-100, a more fine-grained dataset. Full results are reported in Appendix 7. Despite the increased task complexity, MoE students retain their advantages. The dense student achieves 36.70% accuracy, while the best MoE variant (Attention-based with 5 experts) reaches 39.25%, outperforming the baseline under similar compute. This trend indicates that sparse expert models remain effective even in high-label settings, validating their broader applicability.

4. Ablation and Analysis

4.1. Effect of Expert Count

We compare MoE students with 3, 5, and 10 experts under each routing strategy. From Table 1, we observe that soft and attention routing show stable or slightly improved performance as the number of experts increases. However, top-1 routing leads to performance degradation at larger N (e.g., Top-1 (10): 64.74%), indicating instability or undertraining of experts. Top-2 routing maintains competitive accuracy across expert sizes, showing robustness to scale.

4.2. Routing Trade-offs

Top-1 routing achieves the highest accuracy in small-scale setups (e.g., Top-1 (3): 71.93%), but is sensitive to expert scaling. Attention-based routing yields consistently strong performance and shows resilience to architectural variation. Top-2 routing offers a strong balance between sparsity and stability.

4.3. Efficiency Considerations

While soft routing achieves stable accuracy, it incurs the highest computational cost due to full expert activation. Top-2 routing reduces FLOPs significantly while maintaining competitive performance, especially as expert count increases. This confirms that sparse expert utilization can yield favorable trade-offs without explicit expert pruning or manual selection heuristics.

4.4. Discussion

Our findings suggest that the effectiveness of sparse MoE students depends not only on the number of experts, but also on the expressiveness of the gating mechanism. While hard routing (e.g., Top-1) can achieve high accuracy under tight budgets, it does not scale well. Attention-enhanced soft routing generalizes better across expert sizes and enables more stable training. These insights indicate that routing design is as critical as model architecture in sparse modular distillation.

5. Conclusion and Limitations

We presented a compact and modular student architecture for knowledge distillation based on a sparse Mixture-of-Experts framework. By combining class-agnostic expert modules with input-conditioned routing, our method improves accuracy under constrained compute budgets. Through extensive evaluation on CIFAR-10, we demonstrated that sparse MoE students can outperform dense baselines and even the teacher model, particularly when using attention or top-k routing strategies.

Our analysis shows that routing design plays a critical role in balancing efficiency and performance. Top-1 routing offers maximal sparsity but is sensitive to the number of experts, while attention-based gating provides robustness and consistent gains.

Limitations. Our experiments are limited to small-scale image classification tasks (CIFAR-10/100), and we do not perform interpretability or expert specialization analysis. Future work may extend this framework to larger datasets, hierarchical routing, or unsupervised expert specialization.

References

[1] Anonymous. Moe-lora: Efficient fine-tuning of mixture-of-experts models with low-rank adaptation. *arXiv preprint*, 2024. Under submission or not officially published. Placeholder entry. 1

[2] Anonymous. A survey on mixture of experts. *arXiv preprint arXiv:2401.XXXX*, 2024. 2

[3] Nan Du, Yanping Huang, Andrew M. Dai, and et al. Glam: Efficient scaling of language models with mixture-of-experts. In *ICML*, 2022. 1

[4] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. In *ICLR*, 2022. 1, 3

[5] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. 1, 3

[6] Hyeon Seong Ko, Seungju Han, Sangwoo Mo, and et al. Specific expert learning: Enriching ensemble diversity via knowledge distillation. In *NeurIPS*, 2023. 1, 2

[7] Guillaume Lample and Mistral AI. Mixtral of experts: Sparse mixture of experts model by mistral ai. *Tech Report*, 2023. `https://mistral.ai/news/mixtral-of-experts/`. 1

[8] 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*, 2021. 1

[9] Shang-Wen Li, Tianjian Meng, Hanxiao Liu, and et al. St-moe: Designing stable and transferable sparse expert models. In *NeurIPS*, 2022. 1, 2, 3

[10] Jason Weston Marc'Aurelio Ranzato, Arthur Szlam. Learning deep mixtures of experts. In *ICLR Workshop*, 2014. 2

[11] Yongfeng Min and et al. Openmoe: An early effort on open mixture-of-experts language models. *arXiv preprint arXiv:2305.14305*, 2023. 1

[12] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *AAAI*, 2020. 1

[13] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, and et al. Scaling vision with sparse mixture of experts. In *NeurIPS*, 2021. 3

[14] Jason Rolfe and et al. Megablocks: Efficient sparse training with mixture-of-experts. In *ICML*, 2023. 1, 2

[15] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015. 1

[16] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 1, 2

[17] Fangrui Wang, Yawei Sun, Yunfan Shao, Lianfeng Shen, Xiaodan Liang, and Pengcheng Yin. Class-specialized knowledge distillation. In *CVPR*, 2023. 1, 2

[18] Hang Zhao, Yao Zhang, Yunyang Xiong, and et al. Faster-moe: Accelerating mixture-of-experts inference with fine-grained routing. In *NeurIPS*, 2022. 1

Sparse MoE Students for Efficient Knowledge Distillation

Supplementary Material

6. Full CIFAR-10 Results

Table 2 provides the complete results for all 13 MoE configurations evaluated on CIFAR-10. Metrics include test accuracy, total parameter count, FLOPs per image, latency (ms), and peak GPU memory (MB).

7. Full CIFAR-100 Results

Table 3 shows the accuracy and compute statistics for all MoE variants tested on CIFAR-100. Although the classification task is more challenging, attention- and top-2-based routing consistently outperform the dense student baseline.

8. Additional Implementation Details

- **Training settings:** All models are trained for 50 epochs using the Adam optimizer with learning rate 0.001 and batch size 64.
- **Gating network:** A lightweight CNN-based gating network is used for soft and top-k routing. For attention routing, we use a single-head self-attention with 16-dimensional embedding.
- **Compute environment:** All experiments were run on a single NVIDIA A5000 GPU using PyTorch 2.0.
- **FLOPs and memory**: FLOPs are computed using `fvcore`, and peak memory is recorded using `torch.cuda.max_memory_allocated`.

Table 2. Full results on CIFAR-10. All MoE students are class-agnostic with $N \in \{3, 5, 10\}$ experts. Accuracy, latency, and FLOPs are measured on the test set.

Model	Accuracy (%)	Params	FLOPs	Latency (ms)	Peak Memory (MB)
Teacher (ResNet-34)	69.90	21.3M	74.9M	1.63	94.2
Dense Student	71.19	62K	8.4M	0.32	2.36
moe3	70.67	192K	18.6M	0.39	1.77
moe5	70.71	317K	29.8M	0.60	2.25
moe10	68.93	629K	58.1M	1.00	3.45
att3	71.83	183K	19.7M	0.49	2.82
att5	71.84	303K	31.0M	0.69	3.29
att10	71.69	605K	59.2M	1.13	4.44
top1_3	71.93	183K	19.7M	0.49	2.82
top1_5	66.30	317K	5.6M	0.23	2.25
top1_10	64.74	629K	5.6M	0.23	3.44
top2_3	71.04	192K	12.9M	0.38	1.77
top2_5	71.62	317K	12.9M	0.37	2.25
top2_10	71.21	629K	12.9M	0.38	3.44

Table 3. Full results on CIFAR-100. MoE students are evaluated with the same configurations as CIFAR-10.

Model	Accuracy (%)	Params	FLOPs	Latency (ms)	Peak Memory (MB)
Teacher (ResNet-34)	41.16	21.3M	74.9M	1.75	94.4
Dense Student	36.70	436K	7.6M	0.23	2.69
moe3	37.56	1.30M	19.7M	0.38	5.99
moe5	37.15	2.16M	31.7M	0.55	9.28
moe10	36.94	4.32M	61.8M	1.02	17.51
att3	38.62	1.29M	20.8M	0.51	7.04
att5	39.25	2.15M	32.9M	0.68	10.32
att10	38.64	4.29M	62.9M	1.15	18.51
top1_3	31.30	1.30M	6.0M	0.23	5.99
top1_5	31.76	2.16M	6.0M	0.23	9.28
top1_10	30.17	4.32M	6.0M	0.23	17.50
top2_3	37.40	1.30M	13.7M	0.36	5.99
top2_5	36.36	2.16M	13.7M	0.36	9.28
top2_10	36.58	4.32M	13.7M	0.36	17.51