IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 September 2025, accepted 11 October 2025, date of publication 20 October 2025, date of current version 6 November 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3623953

RESEARCH ARTICLE

Sparse MoE Students for Efficient Knowledge Distillation

JONGWON RYU, MINGI KIM, AND JUNYEONG KIM, (Member, IEEE)
Department of AI, Chung-Ang University, Seoul 06974, Republic of Korea

Corresponding author: Junyeong Kim (junyeongkim@cau.ac.kr)

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2022-II220184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics) and partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [RS-2021-II211341, Artificial Intelligence Graduate School Program (Chung-Ang University)].

ABSTRACT We present a sparse Mixture-of-Experts student architecture for knowledge distillation that replaces conventional dense students with a modular design of lightweight, class-agnostic experts dynamically routed by input-conditioned gating. Our approach provides three key benefits: 1) accuracy–efficiency Pareto superiority over dense students by activating only a subset of experts per input; 2) loss-agnostic plug-and-play compatibility, consistently improving performance across diverse Knowledge distillation objectives; and 3) class-agnostic expert decomposition, enabling generalization without label supervision. To ensure fairness, we construct both dense and Mixture-of-Experts students with comparable parameter budgets and systematically evaluate multiple gating strategies, including soft, top-k, and attention-based variants. Experiments on CIFAR-10 and CIFAR-100 under four distillation methods show that attention-gated MoE students consistently outperform dense baselines at equal capacity, achieving up to +1.7% accuracy gains with an average 25% FLOPs reduction. These findings demonstrate the structural advantages of sparse Mixture-of-Experts students as efficient, generalizable, and loss-agnostic alternatives to dense models in knowledge distillation.

INDEX TERMS Knowledge distillation, mixture-of-experts (MoE), routing strategy, sparse models.

I. INTRODUCTION

The proliferation of large-scale neural networks has created a critical need for effective model compression techniques that enable efficient deployment on resource-constrained devices [1], [2]. A variety of strategies have been explored, including pruning [3], [4], quantization [5], and low-rank factorization [6]. Among these, knowledge distillation (KD) has emerged as a simple and powerful paradigm for compressing large neural networks into smaller, more efficient student models [7], [8], [9], [10].

Despite its effectiveness, a common limitation of existing KD research is the implicit assumption that the student must follow a dense, monolithic architecture. This assumption leads to several problems: (i) all parameters are activated for every input, resulting in unnecessary computational waste, (ii) compression inevitably sacrifices accuracy due to the fixed capacity of dense students, and (iii) structural rigidity makes it difficult to adapt to different hardware constraints or interpret failure cases. While recent works have refined distillation losses [9], [10], they remain bound to dense students and therefore cannot overcome the inefficiency–accuracy trade-off.

To address these limitations, we revisit the architecture of student models in KD and propose a sparse, modular alternative based on the Mixture-of-Experts (MoE) framework [11], [12]. Instead of relying on a single dense network, our approach employs multiple lightweight, class-agnostic experts whose outputs are dynamically routed by an input-conditioned gating network. This design introduces the notion of conditional capacity, where only a subset of experts is activated for each input. As a result, the student can maintain or even improve accuracy while reducing FLOPs(Floating Point Operations Per Second), thereby

The associate editor coordinating the review of this manuscript and approving it for publication was Francisco J. Garcia-Penalvo.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

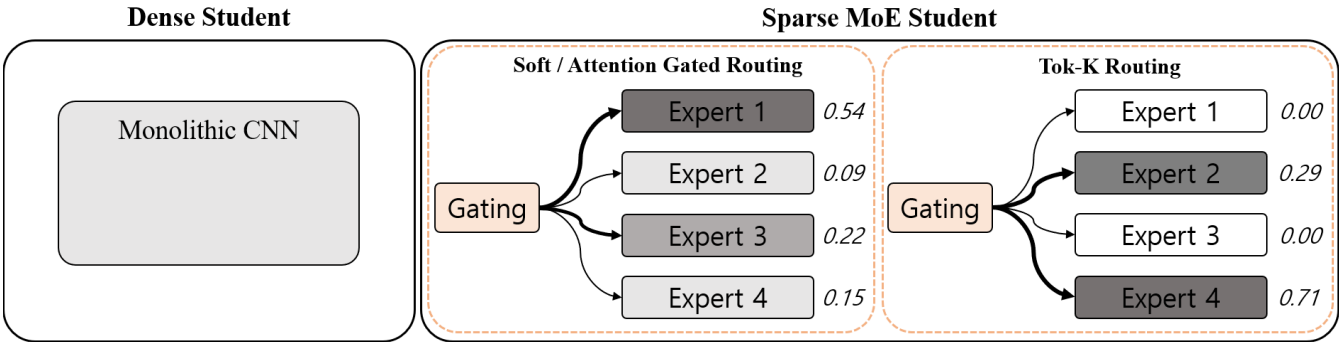


FIGURE 1. Dense vs. Sparse MoE Student.Left: Dense student (monolithic CNN, all parameters active).Middle: Soft/Attention gated routing. All experts are evaluated and aggregated, with gating probabilities shown (e.g., Expert 1:0.54, Expert 2:0.09, Expert 3:0.22, Expert 4:0.15). These values are the softmax scores from the gating network, indicating each expert's relative contribution.Right: Top-k routing (here $k = 2$). Only the top-2 scoring experts are activated(Expert 2:0.29, Expert 4:0.71), while the others are suppressed to zero.

shifting the accuracy-efficiency Pareto frontier beyond dense baselines. The modular, class-agnostic nature of experts further enables better generalization without label supervision and provides structural interpretability.

We systematically investigate different routing mechanisms-soft gating, top-k selection, and attention-based routing and construct dense and MoE students with comparable parameter budgets to ensure fairness. Experiments on image datasets such as CIFAR-10 and CIFAR-100 [13] under four KD methods (KD, DKD, TTM, and WTTM) [7], [14], [15] show that sparse MoE students consistently outperform dense baselines. In particular, attention-gated variants achieve the best trade-off, yielding up to +1.7% accuracy improvements with an average 25% FLOPs reduction at equal capacity.

Highlights of this work are summarized as follows:

- We propose a sparse Mixture-of-Experts student architecture for KD that introduces conditional capacity, enabling accuracy–efficiency Pareto superiority over dense students.
- Our design is loss-agnostic and plug-and-play, consistently improving performance under a variety of knowledge distillation methods without modifying their loss formulations.
- The experts are class-agnostic and dynamically routed, providing modular specialization, better generalization, and interpretability without label supervision.

II. RELATED WORK

A. KNOWLEDGE DISTILLATION

Knowledge Distillation (KD) [7] transfers the predictive behavior of a large teacher to a smaller student, typically by combining a Kullback–Leibler divergence on softened teacher outputs with a cross-entropy term on ground-truth labels. Classical lines of work refine the supervision signal through response, feature, and relation-level objectives [8], [16], [17] or optimize the training protocol [18], yet they largely keep the student network *dense and monolithic*.

Recent methods further enhance the target signal, such as Decoupled Knowledge Distillation (DKD) [14], which separates target and non-target alignment to preserve inter-class margins, and Transformed/Weighted Teacher Matching (TTM/WTTM) [15], which reshape or reweight teacher outputs to provide more learnable distributions. These approaches represent strong state-of-the-art KD baselines and were therefore chosen as testbeds in our study.

However, despite yielding stronger supervision, they still assume dense student architectures that activate all parameters for every input. In contrast, by replacing the dense student with our sparse MoE design, we achieve further improvements even under these SOTA objectives: at comparable parameter budgets, attention-gated MoE students consistently surpass dense baselines in accuracy (up to +1.7%) while also reducing FLOPs by an average of 25%. This demonstrates that our contribution is orthogonal to loss refinements and provides complementary gains in both accuracy and efficiency.

B. MIXTURE OF EXPERTS AND ROUTING STRATEGIES

Mixture-of-Experts (MoE) [11] introduces conditional computation by activating only a small subset of experts per input, greatly improving computational scalability. Large-scale extensions include GShard [19], which scales neural machine translation to trillion-parameter MoEs via distributed training, Switch Transformer [12], which simplifies routing by sending each token to a single expert, and GLaM [20], a generalist language model that employs top-2 expert routing for efficient large-scale pretraining.

In vision, sparse MoE designs have also been explored, most notably ViMoE [21], which applies MoE routing to vision transformers for image classification. However, these models are generally heavyweight, transformer-based, or non-distillation focused. Our work instead targets compact, convolutional student models for efficient knowledge distillation.

Routing design is key in MoE systems. Top-k routing (e.g., in Switch Transformer [12]) activates the most probable experts, improving speed but risking expert collapse. Soft routing [22] distributes responsibility across all experts,

avoiding imbalance but increasing cost. Attention-based and expert-choice routing [23], [24] aim to combine adaptivity and stability by learning content-aware expert assignment.

Our work systematically compares these strategies in a KD context with small MoE models (3)–10 experts). Despite the compactness, we find that attention-augmented soft routing outperforms top-k variants, indicating the importance of expressive gating even in lightweight settings.

C. MOE IN MODULAR AND INTERPRETABLE MODELS

Beyond scale, MoEs are attractive for their inherent modularity and interpretability. DMoE [22] introduced layer-wise modular routing, where different layers selectively activate experts to improve generalization. Recent works further explore efficient implementations and fine-tuning: MOELoRA [23] combines MoE with low-rank adaptation for parameter-efficient transfer, and MegaBlocks [25] provides a block-sparse GPU kernel design to improve training scalability.

Attention-based routing has also been investigated in ST-MoE [23], which integrates self-attention into the gating network to enable differentiable, content-aware expert assignment. Inspired by this line of work, we adopt a lightweight self-attention module in our gating network to improve expert selection without introducing significant overhead.

D. DISTILLATION INTO MODULAR ARCHITECTURES

While most KD research has focused on compressing parameters within dense student networks, relatively few works explore modularity in student design. Notable exceptions include class-specialized students [26] and Specific Expert Learning (SEL) [27], which partition the label space so that different experts are trained for subsets of classes. Although such approaches increase diversity, their reliance on class supervision or heuristic partitioning restricts their applicability and limits generalization to unseen tasks or domains.

In contrast, we propose a fully class-agnostic sparse MoE student, where experts are dynamically selected based on input semantics alone. This design removes the need for manual supervision or predefined task splits, allowing expert specialization to emerge organically from data-driven interactions. As a result, our MoE students learn modular internal structure jointly with the teacher, achieving superior generalization, efficiency, and interpretability in compact distillation settings.

III. METHOD

We propose a modular student model based on a sparse Mixture-of-Experts (MoE) architecture [11] 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 [26], [27], 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 [23].

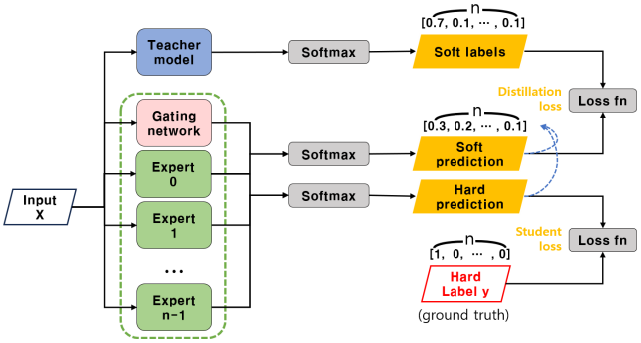


FIGURE 2. Training pipeline of the proposed MoE student for knowledge distillation.

A. 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:

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

  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 [22], where all experts are evaluated and their predictions are aggregated using the gating weights.

The overall training pipeline is visualized in Figure 2. 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 [28], 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 [25].

B. ROUTING VARIANTS

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

1) SOFT ROUTING

All experts are evaluated and combined using the gating weights $G(x)$, as in Eq. (2). This yields a smooth ensemble

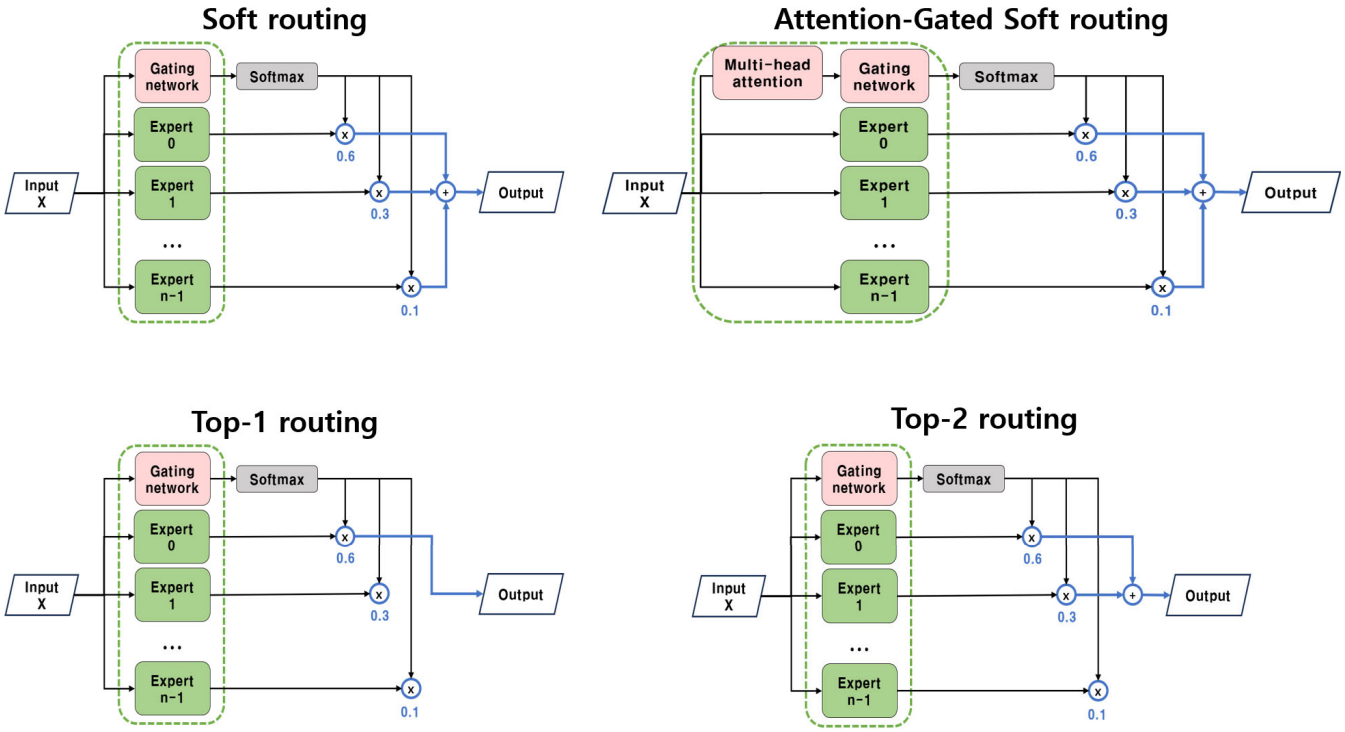


FIGURE 3. Overview of expert routing strategies in our sparse MoE student architecture.(Top left) Soft routing: all experts are evaluated and weighted by softmax scores.(Top right) Attention-based routing: the gating network is augmented with a lightweight self-attention module to improve expert selection.(Bottom left) Top-1 routing: a single expert is selected per input, maximizing sparsity.(Bottom right) Top-2 routing: only the top-2 scoring experts are used.

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.

2) TOP-K ROUTING

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

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

This strategy mimics hard expert selection [12], reducing computational cost. However, it can introduce expert imbalance or instability due to non-differentiable selection and sharp gating decisions [21].

3) ATTENTION-BASED ROUTING

We replace the standard multi layer perceptron gating with a lightweight self-attention mechanism over spatial features of x [23]. Input features are flattened into a sequence and passed through an attention block, yielding a context-aware embedding. Gating scores are then computed as:

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

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

Empirically, attention-based routing achieves the best trade-off between accuracy and efficiency in our experiments. We visualize and compare all routing behaviors in Figure 3, showing their impact on sparsity and expert assignment.

C. KNOWLEDGE DISTILLATION OBJECTIVES

We train the MoE student with a fixed teacher model $T(x) \in \mathbb{R}^K$. Following the standard KD paradigm [7], the objective combines distillation and cross-entropy losses:

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

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

1) VANILLA KD

The distillation loss is the Kullback–Leibler divergence between softened teacher and student distributions:

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

where T is the temperature parameter (fixed to $T = 2.0$). The supervised loss is the standard cross-entropy:

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

Temperature smoothing reveals relative probabilities over non-maximal classes, providing richer guidance.

2) DECOUPLED KD (DKD)

DKD [14] separates the distillation signal into *target class KD (TCKD)* and *non-target class KD (NCKD)*. Let y be the ground-truth label. Then

$$\mathcal{L}_{\text{DKD}} = \lambda_t \mathcal{L}_{\text{TCKD}} + \lambda_n \mathcal{L}_{\text{NCKD}}, \tag{8}$$

where λ_t, λ_n weight the contributions. TCKD aligns teacher and student probabilities on the true class, while NCKD aligns the re-normalized distributions over the remaining classes, preserving inter-class margins.

3) TRANSFORMED TEACHER MATCHING (TTM/WTTM)

TTM [15] proposes to transform the teacher logits $T(x)$ before computing the distillation loss:

$$\mathcal{L}_{\text{TTM}} = \text{KL}\left(\text{softmax}\left(\frac{g(T(x))}{T}\right) \middle\| \text{softmax}\left(\frac{\hat{y}}{T}\right)\right), \tag{9}$$

where $g(\cdot)$ is a transformation such as logit sharpening, masking, or normalization. The weighted variant (WTTM) additionally reweights contributions from different classes to emphasize informative parts of the teacher distribution:

$$\mathcal{L}_{\text{WTTM}} = \sum_{j=1}^{K} w_j \, p_t^{(T)}(j) \left[\log p_t^{(T)}(j) - \log q_s^{(T)}(j)\right], \tag{10}$$

where w_j are class-dependent weights, $p_t^{(T)}$ and $q_s^{(T)}$ are teacher and student softmax outputs with temperature T.

By incorporating these objectives, we evaluate our MoE students not only under vanilla KD but also under stronger supervision signals (DKD, TTM, WTTM), enabling a comprehensive assessment of how structural modularity interacts with diverse distillation criteria.

In practice, this joint distillation training stabilizes optimization and encourages experts to form complementary decision boundaries. In sparse routing regimes, where only a subset of experts is activated per input, the richer teacher signals (e.g., softened or transformed targets) further promote specialization and consistent learning across modules.

IV. EXPERIMENTS

A. EXPERIMENTAL SETUP

We conduct experiments on CIFAR-10 and CIFAR-100 to evaluate the effectiveness of sparse MoE students for knowledge distillation. CIFAR-10 serves as the primary benchmark, while CIFAR-100 is included in the Appendix to assess generalization on fine-grained classification. Both datasets are widely used, open benchmarks that allow us to examine how the number of classes affects distillation performance.

1) TEACHER

We use a ResNet-34 trained with standard cross-entropy as the teacher. We deliberately choose a medium-capacity teacher: if the teacher is too strong, the distillation task becomes trivial and the relative differences among KD methods, expert counts, and routing strategies become harder to observe.

2) DENSE BASELINES

For fair comparison, the dense student architecture is defined as a small convolutional network: Conv$(3, 32) \rightarrow$ ReLU $\rightarrow$ MaxPool $\rightarrow$ Conv$(32, 64) \rightarrow$ ReLU $\rightarrow$ MaxPool $\rightarrow$ FC$(64 \cdot 8 \cdot 8, K)$, where K is the number of classes. This design matches the architecture of a single MoE expert. The motivation is twofold: (i) to ensure parameter comparability between dense and MoE students, and (ii) to keep the student sufficiently lightweight so that the impact of different KD losses, expert counts, and routing strategies can be clearly observed. To compare fairly with MoE models containing multiple experts, we also construct wider dense baselines (`Dense student*3`, `Dense student*5`, `Dense student*10`) whose parameter counts approximate those of MoE students with 3, 5, or 10 experts.

3) MOE STUDENTS

Each MoE student consists of N experts ($N \in \{3, 5, 10\}$), where each expert has the same architecture as the dense baseline. A gating network assigns input-dependent mixture weights. The standard gating network is a lightweight CNN(convolution neural network)followed by a linear projection into N logits, normalized by softmax. We also evaluate an attention-augmented variant: after a shallow convolutional feature extractor, a multi-head self-attention layer (embed_dim $= 16, 4$ heads) models feature interactions before the final linear projection, enabling more expressive and content-aware expert assignment.

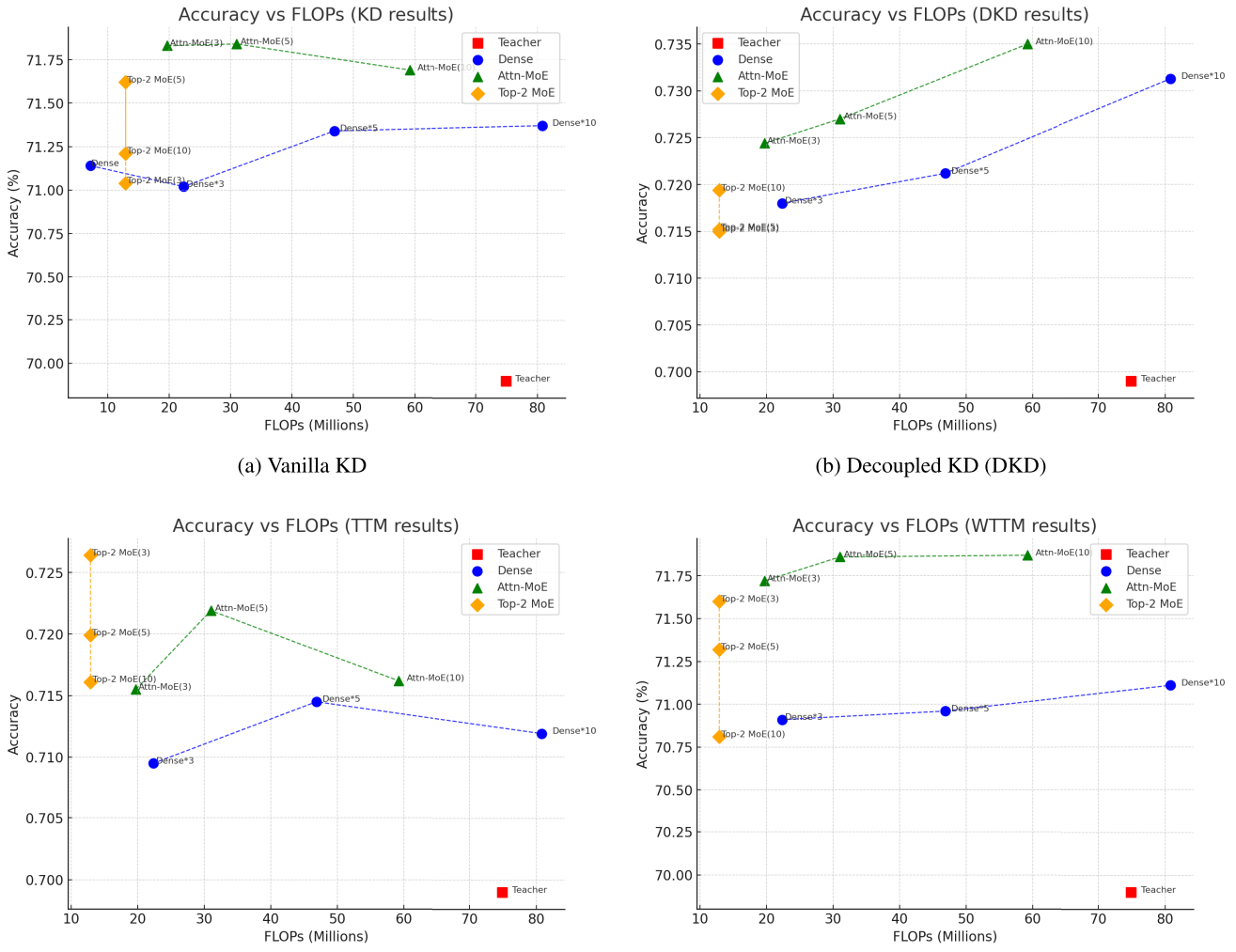
At inference, different routing strategies determine how expert outputs are combined. In *soft routing*, all experts are active and their outputs are linearly combined according to the softmax weights from the gating network. In the *attention* variant, the gating weights are computed through a self-attention mechanism over convolutional features, which allows the model to adaptively emphasize experts based on content similarity. By contrast, *top-1* and *top-2* routing activate only the single most probable expert or the top two experts, respectively, reducing active computation while maintaining conditional specialization.

4) TRAINING

All models are trained under knowledge distillation with temperature $T = 2.0$ and loss weight $\alpha = 0.5$ unless otherwise specified. We evaluate four distillation objectives: vanilla KD, Decoupled KD (DKD), Transformed Teacher Matching (TTM), and Weighted TTM (WTTM). Each model is trained jointly with distillation and cross-entropy losses.

5) METRICS

We report accuracy on the test set, total parameter count, FLOPs (measured with `fvcore`), latency (per-image forward pass measured by CUDA(Compute Unified Device



(a) Vanilla KD

(b) Decoupled KD (DKD)

(c) Transformed Teacher Matching (TTM)

(d) Weighted TTM (WTTM)

FIGURE 4. Accuracy vs. FLOPs trade-off on CIFAR-10 under four KD objectives:(a) Vanilla KD [7],(b) Decoupled KD (DKD) [14],(c) Transformed Teacher Matching (TTM) [15], and (d) Weighted TTM (WTTM) [15]. Dashed lines connect 3→5→10 expert settings within each family to show the efficiency–accuracy trend.

Architecture) timers), and peak GPU memory usage. These metrics allow us to analyze accuracy–efficiency trade-offs under different routing and distillation settings.

B. MAIN RESULTS ON CIFAR-10

Table 1 summarizes the main results on CIFAR-10 across four knowledge distillation objectives: vanilla KD, DKD, TTM, and WTTM. Several consistent trends emerge.

First, all student variants significantly outperform the teacher baseline (69.9%), confirming the effectiveness of distillation. Even the compact dense student achieves over 71% accuracy with only 0.06M parameters, validating that KD alone boosts compact models.

Second, sparse MoE students further improve upon dense baselines under comparable parameter budgets. Attention-gated MoEs consistently achieve the highest accuracy across all four objectives. For instance, under DKD, the Attn-MoE with 10 experts reaches 73.5%, surpassing the best dense student (73.1%) while requiring fewer FLOPs (59M vs. 81M). Similarly, in WTTM, Attn-MoE(10) attains 71.9%, again dominating dense counterparts.

Third, top-2 routing provides a strong accuracy–efficiency balance. With 5 experts, Top-2 MoE achieves 71.6–72.0% across objectives while maintaining only 12.9M FLOPs, yielding a ∼3–4× reduction compared to dense students of similar accuracy. This highlights the benefit of conditional capacity, where only a subset of experts is activated per input.

Finally, we observe that the accuracy–FLOPs Pareto frontier is consistently pushed forward by MoE students. Dense students show diminishing returns as parameters increase (e.g., dense*10 vs. dense*5), whereas MoE students leverage modular routing to achieve higher accuracy at lower compute. Figure 4 visualizes this trade-off: attention-based and top-2 routed MoEs lie strictly above dense baselines across all objectives.

Overall, these results confirm that our sparse MoE design provides three key advantages in KD: (i) Pareto superiority in accuracy–efficiency, (ii) loss-agnostic performance

TABLE 1. Results across KD variants. Best accuracy and lowest FLOPs within each expert-size group (3/5/10) are highlighted in bold.

Loss	Model	Accuracy (%)	FLOPs	Latency (ms)	Peak Mem (MB)	Params
–	Teacher (ResNet-34)	69.90	74.9M	1.63	94.2	21.3M
Vanilla KD [7]						
KD	Dense student*3	71.02	22.35M	0.47	9.80	165K
	Attn-MoE (3 exp)	**71.83**	19.7M	0.49	2.82	183K
	Top-2 MoE (3 exp)	71.04	**12.9M**	0.38	1.77	192K
	Dense student*5	71.34	46.87M	0.44	10.33	299K
	Attn-MoE (5 exp)	**71.84**	31.0M	0.69	3.29	303K
	Top-2 MoE (5 exp)	71.62	**12.9M**	0.37	2.25	317K
	Dense student*10	71.37	80.82M	0.43	11.81	470K
	Attn-MoE (10 exp)	**71.69**	59.2M	1.13	4.44	605K
	Top-2 MoE (10 exp)	71.21	**12.9M**	0.38	3.44	629K
Decoupled KD (DKD) [14]						
DKD	Dense student*3	71.80	22.35M	0.442	9.79	165K
	Attn-MoE (3 exp)	**72.44**	19.73M	0.941	10.95	183K
	Top-2 MoE (3 exp)	71.50	**12.92M**	0.692	9.89	192K
	Dense student*5	72.12	46.87M	0.441	10.32	299K
	Attn-MoE (5 exp)	**72.70**	31.02M	1.301	11.41	303K
	Top-2 MoE (5 exp)	71.52	**12.92M**	0.711	10.37	317K
	Dense student*10	73.13	80.82M	0.448	11.81	470K
	Attn-MoE (10 exp)	**73.50**	59.24M	2.155	12.57	605K
	Top-2 MoE (10 exp)	71.94	**12.93M**	0.703	11.57	629K
Transformed Teacher Matching (TTM) [15]						
TTM	Dense student*3	70.95	22.35M	0.450	9.80	165K
	Attn-MoE (3 exp)	71.55	19.73M	0.963	10.95	183K
	Top-2 MoE (3 exp)	**72.64**	**12.92M**	0.706	9.89	192K
	Dense student*5	71.45	46.87M	0.453	10.33	299K
	Attn-MoE (5 exp)	**72.19**	31.02M	1.283	11.41	303K
	Top-2 MoE (5 exp)	71.99	**12.92M**	0.720	10.37	317K
	Dense student*10	71.19	80.82M	0.454	11.81	470K
	Attn-MoE (10 exp)	**71.62**	59.24M	2.158	12.57	605K
	Top-2 MoE (10 exp)	71.61	**12.93M**	0.717	11.57	629K
Weighted TTM (WTTM) [15]						
WTTM	Dense student*3	70.91	22.35M	0.442	9.80	165K
	Attn-MoE (3 exp)	71.72	19.73M	0.975	10.95	183K
	Top-2 MoE (3 exp)	**71.60**	**12.92M**	0.719	9.89	192K
	Dense student*5	70.96	46.87M	0.455	10.33	299K
	Attn-MoE (5 exp)	**71.86**	31.02M	1.307	11.41	303K
	Top-2 MoE (5 exp)	71.32	**12.92M**	0.706	10.37	317K
	Dense student*10	71.11	80.82M	0.457	11.81	470K
	Attn-MoE (10 exp)	**71.87**	59.24M	2.111	12.57	605K
	Top-2 MoE (10 exp)	70.81	**12.93M**	0.695	11.57	629K

gains across diverse KD objectives, and (iii) Class-agnostic modularity: experts are dynamically routed without label supervision, enabling flexible and interpretable structure

C. GENERALIZATION ON CIFAR-100

To assess the generalization of our approach, we further evaluate on CIFAR-100 using the vanilla KD loss. Table 2 reports full results with the same configurations as CIFAR-10, and Figure 5 visualizes the accuracy–FLOPs trade-off.

Several trends mirror our CIFAR-10 findings. First, distillation improves compact dense students, but performance gains saturate as parameters grow: Dense*10 achieves only 37.4% accuracy, barely above Dense*5 (36.9%). Second, sparse MoE students consistently outperform dense baselines under comparable budgets. In particular, attention-gated MoEs always lie above the dense frontier in Figure 5, demonstrating both higher accuracy and better efficiency. For example, Attn-MoE(5) reaches 39.3% accuracy, surpassing all dense counterparts while requiring fewer FLOPs. Third, top-2 routing offers a favorable accuracy–efficiency balance:

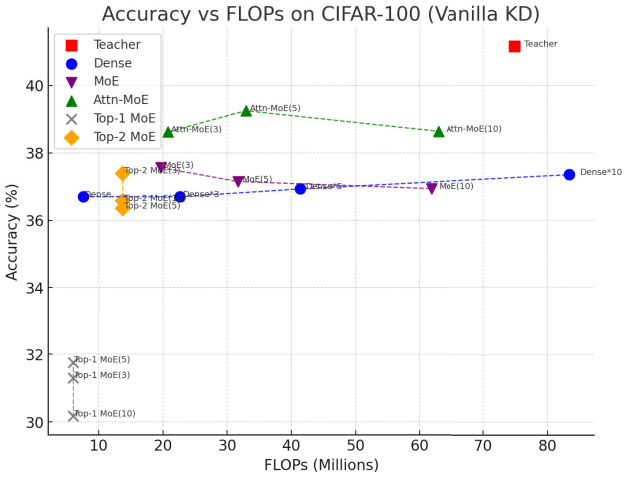


FIGURE 5. Accuracy vs. FLOPs on CIFAR-100 under vanilla KD. Colors/markers denote model families. Dashed lines connect 3→5→10 expert settings within each family to show the efficiency–accuracy trend.

with only 12.9M FLOPs, Top-2 MoE(3) attains 37.4%, competitive with Dense*5 that requires over 41M FLOPs.

Overall, these results confirm that the benefits of sparse MoE students such as conditional capacity, improved accuracy–efficiency trade-off, and loss-agnostic design extend beyond CIFAR-10 to more challenging fine-grained classification. This highlights the robustness of our approach across datasets of varying difficulty and granularity.

V. ABLATION STUDY

To better understand the impact of architectural choices in our sparse MoE student design, we conduct ablation studies focusing on two critical factors: (1) the number of experts, and (2) the routing strategy.

A. EFFECT OF THE NUMBER OF EXPERTS

We vary the number of experts from 3 to 10 while keeping the per-expert architecture fixed to match the dense baseline size. Results in Table 1 and Figure 4 show two consistent patterns. First, increasing the number of experts alone does not always guarantee accuracy gains: naively scaling to 10 experts can even hurt performance in plain MoE due to expert under-utilization. Second, when combined with attention-based routing, larger expert pools yield measurable improvements: Attn-MoE(10) achieves up to 73.5% under DKD, outperforming all dense counterparts of similar or larger size. This indicates that expert multiplicity provides a richer representation capacity, but effective routing is essential to exploit it.

B. EFFECT OF THE ROUTING STRATEGY

We compare four routing strategies: soft, attention-augmented soft, top-1, and top-2. The results highlight the trade-off between accuracy and efficiency. Top-1 routing achieves the lowest FLOPs but suffers from significant

TABLE 2. Full results on CIFAR-100. Models are grouped by expert size (3, 5, 10). Best accuracy and lowest FLOPs within each group are highlighted in bold.

Model	Accuracy (%)	FLOPs	Latency (ms)	Peak Memory (MB)	Params
Teacher (ResNet-34)	41.16	74.9M	1.75	94.4	21.3M
Dense student	36.70	7.6M	0.23	2.69	436K
Dense student*3	36.70	22.6M	0.24	2.71	1.4M
MoE (3 exp)	37.56	19.7M	0.38	5.99	1.30M
Attn-MoE (3 exp)	**38.62**	20.8M	0.51	7.04	1.29M
Top-1 MoE (3 exp)	31.30	**6.0M**	0.23	5.99	1.30M
Top-2 MoE (3 exp)	37.40	13.7M	0.36	5.99	1.30M
Dense student*5	36.93	41.4M	0.24	2.84	2.35M
MoE (5 exp)	37.15	31.7M	0.55	9.28	2.16M
Attn-MoE (5 exp)	**39.25**	32.9M	0.68	10.32	2.15M
Top-1 MoE (5 exp)	31.76	**6.0M**	0.23	9.28	2.16M
Top-2 MoE (5 exp)	36.36	13.7M	0.36	9.28	2.16M
Dense student*10	37.35	83.4M	0.24	2.88	3.73M
MoE (10 exp)	36.94	61.8M	1.02	17.51	4.32M
Attn-MoE (10 exp)	**38.64**	62.9M	1.15	18.51	4.29M
Top-1 MoE (10 exp)	30.17	**6.0M**	0.23	17.50	4.32M
Top-2 MoE (10 exp)	36.58	13.7M	0.36	17.51	4.32M

accuracy drops due to expert collapse. Top-2 routing mitigates this issue by activating multiple experts per input, striking a favorable accuracy–efficiency balance (e.g., Top-2 MoE(5) attains 71.6% accuracy with only 12.9M FLOPs). Soft routing avoids collapse but activates all experts, reducing efficiency. Attention-based routing consistently provides the best Pareto trade-off: it maintains balanced expert utilization, avoids collapse, and leverages semantic gating to push the accuracy frontier (e.g., Attn-MoE(5) reaches 72.7% under DKD).

These ablations confirm that both the number of experts and the routing mechanism play crucial roles in the success of sparse MoE students. While expert multiplicity increases representational capacity, routing quality determines how effectively that capacity is utilized. Attention-based gating emerges as the most reliable design choice, consistently outperforming dense baselines and alternative routing schemes. Moreover, the benefits hold across datasets (CIFAR-10/100) and KD objectives, demonstrating the robustness of our design.

VI. IMPLEMENTATION DETAILS

All models are implemented using PyTorch 2.0 and trained on a single NVIDIA RTX A6000 Ada GPU. We use standard preprocessing for CIFAR-10 and CIFAR-100, including per-channel normalization. No data augmentation is applied to isolate architectural effects. The teacher is a ResNet-34 trained from scratch using standard cross-entropy loss. It is optimized with SGD (learning rate 0.1, momentum 0.9, weight decay 10^{-4}) for 50 epochs. All student models are trained for 50 epochs with Adam optimizer (learning rate 0.001, batch size 64). The teacher is frozen throughout training. Table 3 reports the average training time per epoch on CIFAR-10, averaged across all evaluated knowledge distillation losses (KD, DKD, TTM, and WTTM). While dense and MoE students train in under 10 seconds per epoch, top-1 and top-2 routing incur significantly higher overhead due to sparse expert activation and inefficient GPU utilization.

TABLE 3. Average training time per epoch (in seconds) for each student model on a single NVIDIA RTX A6000 Ada GPU.

Model	Time / epoch (s)
Dense student	7
Dense student*3	7
Dense student*5	7
Dense student*10	8
MoE (3 experts)	7
MoE (5 experts)	8
MoE (10 experts)	9
Attn-MoE (3 experts)	8
Attn-MoE (5experts)	9
Attn-MoE (10 experts)	9
Top-1 MoE (3 experts)	33
Top-1 MoE (5 experts)	34
Top-1 MoE (10 experts)	35
Top-2 MoE (3 experts)	75
Top-2 MoE (5 experts)	76
Top-2 MoE (10 experts)	76

VII. LIMITATIONS AND FUTURE WORK

While our proposed sparse MoE student architecture shows promising results, several limitations remain, both in terms of experimental scope and model design.

First, all experiments are conducted on relatively small-scale datasets (CIFAR-10 and CIFAR-100). Although suitable for controlled evaluation, they do not reflect the complexity and scale of real-world benchmarks such as ImageNet. Future work will examine whether the observed benefits of sparse routing generalize to large-scale settings.

Second, our current MoE student design uses a fixed set of lightweight convolutional experts. While this ensures compactness and training stability, it may limit the model's capacity and expressiveness in more complex scenarios. We do not explore heterogeneous expert types, expert

TABLE 4. Full results on CIFAR-10 under four KD objectives: Vanilla KD [7], Decoupled KD (DKD) [14], Transformed Teacher Matching (TTM) [15], and Weighted TTM (WTTM) [15].

Loss	Model	Accuracy (%)	Params	FLOPs	Latency (ms) (ms)	Peak Mem (MB) (MB)
-	Teacher (ResNet-34)	69.90	21.3M	74.9M	1.63	94.2
Vanilla KD	Dense student	71.14	67K	7.2M	0.42	9.41
	Dense student*3	71.02	165K	22.35M	0.47	9.80
	Dense student*5	71.34	299K	46.87M	0.44	10.33
	Dense student*10	71.37	470K	80.82M	0.43	11.81
	MoE (3 exp)	70.67	192K	18.6M	0.39	1.77
	MoE (5 exp)	70.71	317K	29.8M	0.60	2.25
	MoE (10 exp)	68.93	629K	58.1M	1.00	3.45
	Attn-MoE (3 exp)	71.83	183K	19.7M	0.49	2.82
	Attn-MoE (5 exp)	71.84	303K	31.0M	0.69	3.29
	Attn-MoE (10 exp)	71.69	605K	59.2M	1.13	4.44
	Top-1 MoE (3 exp)	65.93	183K	5.6M	0.23	1.77
	Top-1 MoE (5 exp)	66.30	317K	5.6M	0.23	2.25
	Top-1 MoE (10 exp)	64.74	629K	5.6M	0.23	3.44
	Top-2 MoE (3 exp)	71.04	192K	12.9M	0.38	1.77
	Top-2 MoE (5 exp)	71.62	317K	12.9M	0.37	2.25
	Top-2 MoE (10 exp)	71.21	629K	12.9M	0.38	3.44
DKD	Dense student	70.34	67K	7.3M	0.416	9.41
	Dense student*3	71.80	165K	22.35M	0.442	9.79
	Dense student*5	72.12	299K	46.87M	0.441	10.32
	Dense student*10	73.13	470K	80.82M	0.448	11.81
	MoE (3 exp)	71.53	192K	18.56M	0.738	9.89
	MoE (5 exp)	71.59	317K	29.85M	1.055	10.37
	MoE (10 exp)	73.00	629K	58.09M	1.865	11.57
	Attn-MoE (3 exp)	72.44	183K	19.73M	0.941	10.95
	Attn-MoE (5 exp)	72.70	303K	31.02M	1.301	11.41
	Attn-MoE (10 exp)	73.50	605K	59.24M	2.155	12.57
	Top-1 MoE (3 exp)	68.08	192K	5.64M	0.450	9.89
	Top-1 MoE (5 exp)	67.52	317K	5.64M	0.435	10.37
	Top-1 MoE (10 exp)	71.15	629K	5.64M	0.435	11.57
	Top-2 MoE (3 exp)	71.50	192K	12.92M	0.692	9.89
	Top-2 MoE (5 exp)	71.52	317K	12.92M	0.711	10.37
	Top-2 MoE (10 exp)	71.94	629K	12.93M	0.703	11.57
TTM	Dense student	70.94	67K	7.3M	0.430	9.41
	Dense student*3	70.95	165K	22.35M	0.450	9.80
	Dense student*5	71.45	299K	46.87M	0.453	10.33
	Dense student*10	71.19	470K	80.82M	0.454	11.81
	MoE (3 exp)	71.75	192K	18.56M	0.753	9.89
	MoE (5 exp)	71.08	317K	29.85M	1.103	10.37
	MoE (10 exp)	70.01	629K	58.09M	1.884	11.57
	Attn-MoE (3 exp)	71.55	183K	19.73M	0.963	10.95
	Attn-MoE (5 exp)	72.19	303K	31.02M	1.283	11.41
	Attn-MoE (10 exp)	71.62	605K	59.24M	2.158	12.57
	Top-1 MoE (3 exp)	68.71	192K	5.64M	0.458	9.89
	Top-1 MoE (5 exp)	64.55	317K	5.64M	0.448	10.37
	Top-1 MoE (10 exp)	63.66	629K	5.64M	0.448	11.57
	Top-2 MoE (3 exp)	72.64	192K	12.92M	0.706	9.89
	Top-2 MoE (5 exp)	71.99	317K	12.92M	0.720	10.37
	Top-2 MoE (10 exp)	71.61	629K	12.93M	0.717	11.57
WTTM	Dense student	70.61	67K	7.3M	0.440	9.41
	Dense student*3	70.91	165K	22.35M	0.442	9.80
	Dense student*5	70.96	299K	46.87M	0.455	10.33
	Dense student*10	70.11	470K	80.82M	0.457	11.81
	MoE (3 exp)	71.61	192K	18.56M	0.757	9.89
	MoE (5 exp)	71.24	317K	29.85M	1.095	10.37
	MoE (10 exp)	70.28	629K	58.09M	1.937	11.57
	Attn-MoE (3 exp)	71.72	183K	19.73M	0.975	10.95
	Attn-MoE (5 exp)	71.86	303K	31.02M	1.307	11.41
	Attn-MoE (10 exp)	71.87	605K	59.24M	2.111	12.57
	Top-1 MoE (3 exp)	70.44	192K	5.64M	0.445	9.89
	Top-1 MoE (5 exp)	65.57	317K	5.64M	0.450	10.37
	Top-1 MoE (10 exp)	61.65	629K	5.64M	0.447	11.57
	Top-2 MoE (3 exp)	71.60	192K	12.92M	0.719	9.89
	Top-2 MoE (5 exp)	71.32	317K	12.92M	0.706	10.37
	Top-2 MoE (10 exp)	70.81	629K	12.93M	0.695	11.57

fusion, or adaptive depth, which could further improve specialization.

Third, we do not perform in-depth analysis of expert behavior. Although routing strategies are quantitatively compared, we do not analyze expert activation entropy, usage imbalance, or specialization dynamics. More interpretability-oriented evaluations are needed to better understand how modularity emerges during training.

Fourth, the teacher model, optimizer, and training schedule are fixed across all experiments. We do not study the interaction between teacher strength, augmentation schemes, or distillation schedules, factors that may significantly affect convergence in sparse setups.

Fifth, our method is evaluated solely on image classification. Extending sparse MoE distillation to tasks such as object detection, multi-task learning, or continual learning remains an open direction, where modular student designs may offer further advantages.

Finally, we have not yet examined the practical deployment of sparse MoE students on mobile or edge devices. Although our results show clear FLOPs reductions, the latency improvements on GPUs are modest due to parallelization. Investigating system-level optimizations, device-specific accelerators, and real-time inference scenarios will be important for translating the efficiency gains into practical benefits.

Addressing these limitations will be essential for validating the generality, robustness, and real-world applicability of sparse MoE students.

VIII. CONCLUSION

In this work, we revisited the design of student models in knowledge distillation and proposed a sparse Mixture-of-Experts (MoE) student architecture. Unlike conventional dense students that activate all parameters for every input, our design leverages conditional capacity by dynamically routing inputs to a subset of lightweight, class-agnostic experts. This structural change provides three key advantages: (i) accuracy–efficiency Pareto superiority over dense baselines, (ii) loss-agnostic and plug-and-play integration with diverse KD objectives, and (iii) modularity without reliance on label supervision.

Extensive experiments on CIFAR-10 and CIFAR-100 under four distillation objectives (KD, DKD, TTM, and WTTM) demonstrate that sparse MoE students consistently outperform dense counterparts under comparable parameter budgets. Attention-gated MoEs deliver the strongest performance, achieving up to +1.7% accuracy improvements while reducing FLOPs by an average of 25%, and Top-2 routing provides the best accuracy–efficiency balance with up to 4× FLOPs reduction at similar accuracy levels. These findings highlight the robustness of our approach across datasets and loss functions.

Looking forward, we see two promising directions. First, applying sparse MoE students to larger-scale architectures such as ResNet-50 or Vision Transformers could further amplify efficiency gains. Second, analyzing expert specialization in class-agnostic settings may provide insights into interpretability and adaptive deployment in real-world systems. We hope this work encourages the community to move beyond monolithic dense students and explore modular, sparse alternatives for efficient and generalizable knowledge distillation.

APPENDIX
FULL RESULTS ON CIFAR10

See Table 4.

REFERENCES

[1] X. Zhu, J. Li, Y. Liu, C. Ma, and W. Wang, "A survey on model compression for large language models," *Trans. Assoc. Comput. Linguistics*, vol. 12, pp. 1556–1577, Nov. 2024.

[2] L. Shen, Y. Sun, Z. Yu, L. Ding, X. Tian, and D. Tao, "On efficient training of large-scale deep learning models," *ACM Comput. Surv.*, vol. 57, no. 3, pp. 1–36, Mar. 2025.
[3] Y. He, G. Kang, X. Dong, Y. Fu, and Y. Yang, "Soft filter pruning for accelerating deep convolutional neural networks," 2018, *arXiv:1808.06866*.
[4] J.-H. Luo, J. Wu, and W. Lin, "ThiNet: A filter level pruning method for deep neural network compression," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5058–5066.
[5] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," in *Low-power Computer Vision*. Boca Raton, FL, USA: CRC Press, 2022, pp. 291–326.
[6] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6655–6659.
[7] G. E. Hinton, O. Vinyals, and J. B. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS Deep Learn. Represent. Learn. Workshop*, 2015.
[8] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," in *Proc. ICLR*, 2014.
[9] F. Sarfraz, E. Arani, and B. Zonooz, "Knowledge distillation beyond model compression," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 6136–6143.
[10] T. Li, J. Li, Z. Liu, and C. Zhang, "Few sample knowledge distillation for efficient network compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14627–14635.
[11] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," 2017, *arXiv:1701.06538*.
[12] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," in *Proc. ICLR*, 2021.
[13] A. Krizhevsky, V. Nair, and G. Hinton. (2009). *CIFAR-10 and CIFAR-100 Datasets*. [Online]. Available: https://www.cs.toronto.edu/kriz/cifar.html
[14] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, "Decoupled knowledge distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11953–11962.
[15] K. Zheng and E.-H. Yang, "Knowledge distillation based on transformed teacher matching," 2024, *arXiv:2402.11148*.
[16] Z. Huang, X. Shen, J. Xing, T. Liu, X. Tian, H. Li, B. Deng, J. Huang, and X.-S. Hua, "Revisiting knowledge distillation: An inheritance and exploration framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3578–3587.
[17] Z. Hao, J. Guo, K. Han, H. Hu, C. Xu, and Y. Wang, "Revisit the power of vanilla knowledge distillation: From small scale to large scale," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 10170–10183.
[18] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," in *Proc. AAAI*, vol. 34, 2020, pp. 5191–5198.
[19] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Fırat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "GShard: Scaling giant models with conditional computation and automatic sharding," in *Proc. ICLR*, 2020.
[20] N. Du et al., "GLaM: Efficient scaling of language models with mixture-of-experts," in *Proc. ICML*, 2021, pp. 5547–5569.
[21] C. Riquelme, J. Puigcerver, B. Mustafa, M. Neumann, R. Jenatton, A. S. Pinto, D. Keysers, and N. Houlsby, "Scaling vision with sparse mixture of experts," in *Proc. NeurIPS*, 2021, pp. 8583–8595.
[22] J. W. Marc'Aurelio Ranzato and A. Szlam, "Learning deep mixtures of experts," in *Proc. ICLR Workshop*, 2014. [Online]. Available: https://openreview.net/forum?id=SkP1j9cVG
[23] B. Zoph, I. Bello, S. Kumar, N. Du, Y. Huang, J. Dean, N. Shazeer, and W. Fedus, "ST-MoE: designing stable and transferable sparse expert models," 2022, *arXiv:2202.08906*.
[24] C. M. Rosenbaum, T. Klinger, and M. Riemer, "Routing networks: Adaptive selection of non-linear functions for multi-task learning," in *Proc. ICLR*, 2017.
[25] T. Gale, D. Narayanan, C. Young, and M. Zaharia, "Megablocks: Efficient sparse training with mixture-of-experts," in *Proc. ICML*, 2023, pp. 288–304.
[26] F. Wang, Y. Sun, Y. Shao, L. Shen, X. Liang, and P. Yin, "Class-specialized knowledge distillation," in *Proc. CVPR*, Jun. 2023, pp. 247–264.
[27] W. Kao, H. Xie, C. Lin, and W. Cheng, "Specific expert learning: Enriching ensemble diversity via knowledge distillation," *IEEE Trans. Cybern.*, vol. 53, no. 4, pp. 2494–2505, Apr. 2023.
[28] W. Cai, J. Jiang, F. Wang, J. Tang, S. Kim, and J. Huang, "A survey on mixture of experts in large language models," *IEEE Trans. Knowl. Data Eng.*, 2024.



JONGWON RYU received the M.S. degree in artificial intelligence from Chung-Ang University, Seoul, Republic of Korea, in 2025, where he is currently pursuing the Ph.D. degree. His research interests include natural language processing and various multi-modal reasoning.



MINGI KIM received the B.S. degree in business administration from Chung-Ang University, Seoul, Republic of Korea, in 2023, where he is currently pursuing the M.S. degree. His research interests include fairness, natural-language processing, and various multi-modal learning.



JUNYEONG KIM (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from KAIST, Republic of Korea, in 2015, 2017, and 2021, respectively. He was a Postdoctoral Research Associate at the Artificial Intelligence and Machine Learning Laboratory, School of Electrical Engineering, KAIST. He is currently an Assistant Professor at the Department of AI, Chung-Ang University, Republic of Korea. His research interests include visual-language reasoning, visual question answering, and various video-based problems.