Balanced Knowledge Distillation for Large Language Models with Mix-of-Experts

Jiajun Liu[1], Yao He[2], Wenjun Ke[1, 3], Peng Wang[1, 3*], Ziyu Shang[1], Guozheng Li[1], Zijie Xu[1]

[1]School of Computer Science and Engineering, Southeast University
[2]School of Institute of Collaborate Innovation, University of Macau
[3]Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China
{jiajliu, kewenjun, pwang, ziyus1999, gzli, zijiexu}@seu.edu.cn, mc46477@um.edu.mo

Abstract

Mixture-of-Experts (MoE) architectures have recently become a more prevalent choice for large language models (LLMs) than dense architectures due to their superior performance. However, billions of parameters bring MoE LLMs a huge cost for deployment and inference. To address these issues, knowledge distillation (KD) has become a widely adopted technique to compress LLMs. Existing KD methods for LLMs can be divided into *dense-to-dense* and *moe-to-dense* distillation. *Dense-to-dense* distillation transfers knowledge between single dense LLMs, while *moe-to-dense* distillation attempts to transfer knowledge between the MoE LLMs and the dense LLMs. However, the architectural mismatch prevents the student from fully absorbing knowledge when distilling MoE LLMs. To address this limitation, we investigate a new distillation setting, *moe-to-moe*, which aims to fully leverage expert knowledge of teachers and enable the student to absorb it more effectively. Compared to *dense-to-dense* and *moe-to-dense*, *moe-to-moe* suffers from two imbalance issues. First, expert-coverage deficiency reflects an imbalanced knowledge transfer of teacher experts: traditional distillation utilizes only the few experts activated by the teacher router. Second, routing imbalance appears when the student routing distribution drifts from the teacher, which makes it difficult for students to learn how to distribute different experts. To overcome these issues, we propose a novel distillation framework for *moe-to-moe*, **B**alanced **Distill**ation (**B-Distill**), which equally spreads teacher expertise across student experts while regularizing the student router toward teacher-consistent balance. First, to mitigate expert-coverage deficiency, we introduce Monte Carlo exploration, which stochastically perturbs router probabilities so every teacher and student expert is sampled without enlarging the search space. Second, to correct routing imbalance and avert load collapse, we propose an entropy-aware router distillation mechanism that aligns the student router with the teacher while curbing over-concentration. Experiments show that B-Distill outperforms baselines by up to 6.6% in Rouge-L.

Introduction

Large language models (LLMs) (Achiam et al. 2023; Yang et al. 2024; Team et al. 2025) have achieved impressive performance in natural language processing (NLP) tasks, such as text generation (Liu et al. 2025), code completion (Eghbali and Pradel 2024), and information extraction (Xu et al. 2024b). Across modern architectures of LLMs, mixture-of-experts (MoE) (Lepikhin et al. 2021; Fedus, Zoph, and Shazeer 2022) have been widely adopted because they combine vast representational capacity with reduced per-token computation. However, this benefit involves extensive parameter counts that complicate deployment and inference. For example, the MoE variant of the Qwen3 series contains up to 235 billion parameters (Yang et al. 2025), imposing severe memory and latency burdens in resource-constrained environments. Although some lightweight frameworks, such as LoRAMoE (Dou et al. 2024) and HydraLora (Tian et al. 2024), have implemented lightweight MoE models, the number of parameters in the base model is still large. Consequently, effective compression for LLMs based on MoE architectures has become an urgent research priority.

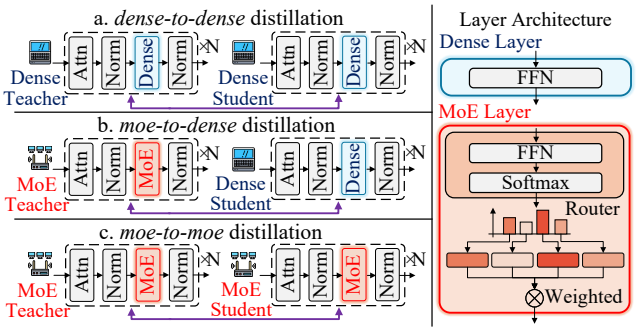

Figure 1: Illustration of *dense-to-dense* distillation, *moe-to-dense* distillation, and *moe-to-moe* distillation. FFN denotes the feed-forward neural network in the transformer block.

As a model compression technique, knowledge distillation (KD) (Hinton 2014) has been widely used in LLMs, in which a compact student model is trained to imitate a larger teacher model (Li, Liu, and Wang 2025). Current LLM distillation approaches fall into *dense-to-dense* and *moe-to-dense*. First, *dense-to-dense* KD transfers logits or hidden states between fully dense networks, as shown in Figure 1 (a), which ignores the expert structure of MoE teachers (Ko et al. 2025; Wang et al. 2025). Second, *moe-to-dense* KD attempts to fuse the knowledge of all teacher experts into a sin-

*Corresponding author.

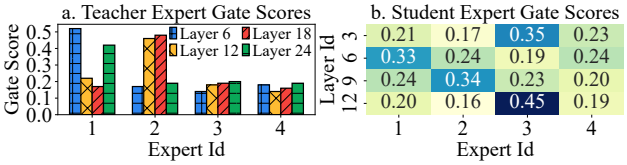

Figure 2: An example of distilling from the teacher (MoE architecture modified from Llama3.1-8B) to the student (MoE architecture modified from Llama3.2-1B). We test the expert routing of layers 6, 12, 18, and 24 of the teacher model and layers 3, 6, 9, and 12 of the student model, respectively.

gle dense student (Kim, Chu, and Yang 2025), as shown in Figure 1 (b), or distribute the knowledge of a dense teacher to an MoE student for learning (Xu et al. 2024a). However, the architectural mismatch severely limits the ability of the student to internalize diverse expertise (Hao et al. 2023). To address this limitation, we dive into the *moe-to-moe* distillation as shown in Figure 1 (c), which aims to compress MoE LLMs under the same architecture.

Although *moe-to-moe* distillation preserves architectural compatibility, it introduces two imbalance phenomena that obstruct knowledge transfer. First, expert-coverage deficiency arises because conventional KD objectives are computed only on the subset of teacher experts activated in each mini-batch. The remaining experts, therefore, receive negligible gradients, and their knowledge is not transferred to the student. As shown in Figure 2 (a), during distillation, only one expert (such as Expert 1 in Layer 6) in a teacher MoE layer is activated the most. The other experts are assigned very little weight, resulting in knowledge not being adequately distilled to the student model. Second, routing imbalance appears when the student's probabilistic router departs from the teacher's distribution. This divergence induces load collapse, where a few student experts process most tokens while the rest remain underutilized. As shown in Figure 2 (b), after distillation, the gate weights of student experts are also concentrated on a small number of experts (such as Expert 3 in Layer 12), resulting in load collapse and decreasing the effectiveness of distillation. Previous works have shown that load collapse can severely degrade the performance of MoE LLMs (Xie et al. 2024). Thus, the load collapse should be avoided during the distillation.

To resolve both imbalances, we present Balanced Distillation (B-Distill), a *moe-to-moe* framework that explicitly balances expert transfer and routing. Firstly, to enhance expert coverage in the teacher model, inspired by Monte Carlo tree search (Silver et al. 2016), we propose a Monte Carlo expert exploration mechanism, which injects controlled stochasticity into router probabilities so that every expert in both teacher and student is sampled during training. Secondly, to correct routing imbalance in the student model, we propose an entropy-aware router distillation mechanism, which guides the student router toward the distribution of the teacher while penalizing over-concentration to prevent load collapse. Both the Monte Carlo expert exploration and the entropy-aware router distillation integrate seamlessly with standard MoE training pipelines, which can be applied to the *moe-to-moe* distillation for MoE LLMs naturally.

We evaluate B-Distill with ten baselines based on Llama3 (Grattafiori et al. 2024) and Qwen3 (Yang et al. 2025) models, with feed-forward neural networks replaced by MoE layers. Experimental results on the general and specific benchmarks show that our B-Distill outperforms all baselines by up to 6.6% in Rouge-L and 8% in accuracy scores. More ablation and exploration results demonstrate the effectiveness of the Monte Carlo expert exploration and entropy-aware router distillation, which balances the expert choices in teachers and the gate scores of distilled students.

Our main contributions are as follows:

- We are among the first to investigate the *moe-to-moe* distillation for LLMs, and propose a novel *moe-to-moe* framework, B-Distill, which relieves the imbalance of expert-coverage deficiency and routing collapse.
- We propose a Monte Carlo expert exploration mechanism, which samples teacher expert representation during distillation to mitigate expert-coverage deficiency.
- We propose the entropy-aware router distillation, which supplements standard gates matching with an entropy regularizer to prevent routing collapse.
- Experiments on diverse benchmarks demonstrate that B-Distill outperforms state-of-the-art distillation baselines while balancing expert utilization for student models.

Related Work

LLMs based on MoEs. The Mixture-of-Experts (MoE) framework begins as a conditional-computation model that routes inputs to specialised experts (Jacobs et al. 1991) and is later extended to deep networks (Jordan and Jacobs 1994). Sparse-gated MoE layers (Shazeer et al. 2017) improve efficiency, while routing strategies evolved from token-level top-k expert selection (Lepikhin et al. 2021; Fedus, Zoph, and Shazeer 2022) to expert-selected tokens (Zhou et al. 2022) and globally optimised assignments (Lewis et al. 2021; Clark et al. 2022). Although these innovations have pushed LLMs beyond the hundred-billion-parameter mark (Chen et al. 2023; Jiang et al. 2024; Yang et al. 2025), they also intensify memory and latency bottlenecks, underscoring the urgent need to compress MoE-based LLMs. Currently, some lightweight MoE architectures, such as LoRAMoE (Dou et al. 2024) and HydraLora (Tian et al. 2024), have been proposed, but they cannot compress base models.

Knowledge Distillation. Knowledge distillation (KD) remains the standard approach for compressing large models (Hinton 2014; Liu et al. 2024), which can be categorized by *dense-to-dense* and *moe-to-dense* distillation. In *dense-to-dense*, black-box methods train students on teacher-generated chains of thought (Hsieh et al. 2023; Ho, Schmid, and Yun 2023), whereas white-box techniques align hidden representations (Zhang et al. 2024) or refine divergence losses (Sanh 2019; Agarwal et al. 2024; Ko et al. 2024). In *moe-to-dense*, recent methods resample or reweight experts for effective distilling (Xu et al. 2024a; Kim, Chu, and Yang 2025). However, little work studies *moe-to-moe* distillation.

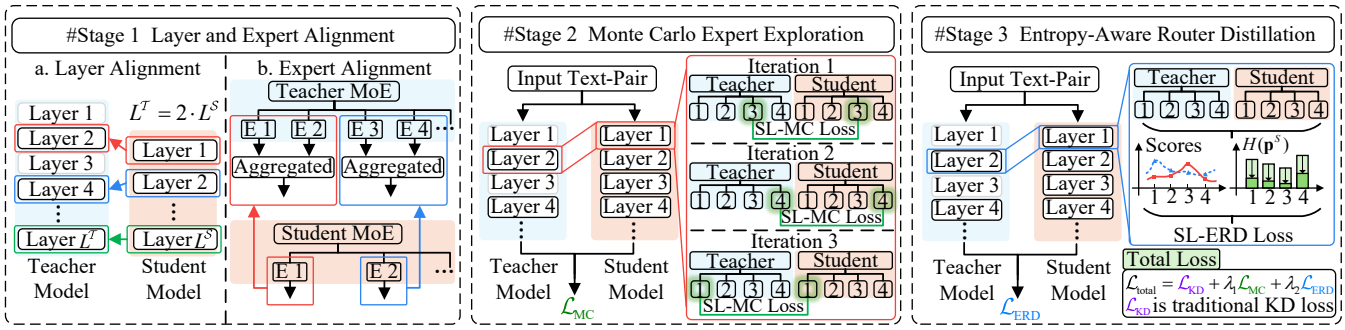

Figure 3: An overview of the B-Distill framework.

Methodology

Preliminary and Problem Statement

Knowledge Distillation for LLMs. Given an input-response pair $(\boldsymbol{x}, \boldsymbol{y})$, let $p(\boldsymbol{y}|\boldsymbol{x})$ be the predictive distribution of a teacher model $\mathcal{T}$ and $q(\boldsymbol{y}|\boldsymbol{x})$ that of a student model $\mathcal{S}$ with parameters θ. Knowledge distillation for LLMs seeks to align the two distributions p and q. The most common alignment method is Kullback-Leibler (KL) divergence, and the distillation loss with KL can be denoted as follows:

$$\mathcal{L}_{\mathrm{KL}} = \mathrm{KL}(p||q) = \sum_{t=1}^{|\boldsymbol{y}|} p(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) \log \frac{p(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})}{q(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})} \tag{1}$$

where y_t and $\boldsymbol{y}_{<t}$ denote the t-th token and the first t-th prefix tokens in y, respectively. Minimizing Eq. (1) encourages the student to reproduce the token-level predictive behavior of the teacher. The supervised fine-tuning (SFT) loss can be calculated as follows:

$$\mathcal{L}_{\mathrm{SFT}} = -\sum_{t=1}^{|\boldsymbol{y}|} \log q(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}) \tag{2}$$

The overall distillation loss $\mathcal{L}_{KD}$ can be denoted as follows:

$$\mathcal{L}_{\mathrm{KD}} = \mathcal{L}_{\mathrm{SFT}} + \mathcal{L}_{\mathrm{KL}} \tag{3}$$

Mixture-of-Experts (MoE) LLMs. The MoE LLMs replace each feed-forward sub-layer in the transformer block with a bank of E expert networks $\{\mathrm{FFN}_e\}_{e=1}^{E}$ and a learnable router $g(\cdot)$. For a token representation $\mathbf{h} \in \mathbb{R}^d$, the router produces a gating vector as follows:

$$\mathbf{p} = \mathrm{softmax}\big(g(\mathbf{h})\big) \in [0,1]^E, \qquad \|\mathbf{p}\|_1 = 1. \tag{4}$$

Then, the layer output $\mathrm{MoE}(\cdot)$ for each layer is a weighted sum over all outputs of experts:

$$\mathrm{MoE}(\mathbf{h}) = \sum_{e=1}^{E} p_e \, \mathrm{FFN}_e(\mathbf{h}) \tag{5}$$

where $p_e \in \mathbf{p}$. Therefore, each expert contributes proportionally to its routing score, yielding high representational capacity while preserving deterministic differentiability throughout training. In this paper, we develop distillation techniques tailored to MoE LLMs, transferring knowledge from a full-capacity MoE teacher to a compact MoE student.

Framework

The framework of B-Distill is shown in Figure 3. In Stage 1, we align the teacher and student models at both the layer and expert levels, enabling structured knowledge transfer despite differences in architectural scale. In Stage 2, we introduce Monte Carlo expert exploration, which injects controlled stochasticity into router probabilities to diversify expert sampling and enhance knowledge coverage. In Stage 3, we apply entropy-aware router distillation to align student routing behavior with the teacher while preventing expert over-concentration with entropy-aware regularization loss.

Layer and Expert Alignment

Before applying the Monte Carlo expert exploration, considering that student models have fewer transformer layers or experts, we first align the layers and the experts of the teacher and the student for better distillation. To align the layers, assume the teacher model and the student model contain $L^{\mathcal{T}}$ and $L^{\mathcal{S}}$ Transformer blocks, respectively. Inspired by the previous layer distillation works (Liang et al. 2023), we align them by the linear rule $\ell^{\mathcal{S}} = \left\lfloor \ell^{\mathcal{T}} \frac{L^{\mathcal{S}}}{L^{\mathcal{T}}} \right\rfloor$ and apply SL-MC on each aligned layer pair $(\ell^{\mathcal{S}}, \ell^{\mathcal{T}})$. Similarly, to align the experts in one layer, assume the teacher layer contains $E^{\mathcal{T}}$ experts and the student layer contains $E^{\mathcal{S}}$ experts. We construct an equal-width mapping π given layer j:

$$\pi(j) = \left\lfloor \frac{(j-1)\, E^{\mathcal{S}}}{E^{\mathcal{T}}} \right\rfloor + 1 \tag{6}$$

In other words, the teacher experts are divided into $E^{\mathcal{S}}$ consecutive index intervals of equal size so that every $\lceil E^{\mathcal{T}}/E^{\mathcal{S}} \rceil$ teacher expert is assigned to one student expert. The teacher gate is then aggregated to student dimensionality via $\tilde{\mathbf{p}}_e^{\mathcal{T}} = \sum_{j:\pi(j)=e} \mathbf{p}_j^{\mathcal{T}}$ and the aggregated teacher state is $\tilde{\mathbf{h}}_e^{\mathcal{T}} = \sum_{j:\pi(j)=e} \big(p_j^{\mathcal{T}} / \tilde{p}_e^{\mathcal{T}}\big) \mathbf{h}_j^{\mathcal{T}}$. In this way, we distill the expert knowledge of the teacher to the experts of the corresponding layer of the student in the Monte Carlo expert exploration.

Monte Carlo Expert Exploration

To raise the expected gradient received by each expert, inspired by Monte Carlo tree search (Silver et al. 2016), we stochastically widen teacher expert coverage in MoE layers during training with a Monte Carlo (MC) expert exploration.

Algorithm 1: Single Layer MC Expert Exploration

Input: $\tilde{\mathbf{p}}^{\mathcal{T}}, \tilde{\mathbf{h}}^{\mathcal{T}}, \mathbf{h}_e^{\mathcal{S}}, K, \alpha$
Output: SL-MC loss in the ℓ-th layer $\mathcal{L}_{\text{SL-MC}}^{\ell}$

1 Initialize gate: $\mathbf{p}^{(0)} \leftarrow \tilde{\mathbf{p}}^{\mathcal{T}}$;
2 Initialize loss: $\mathcal{L}_{\text{SL-MC}}^{\ell} \leftarrow 0$;
3 **for** $t = 1$ **to** K **do**
4 Sample teacher expert $e_t \sim \mathbf{p}^{(t-1)}$;
5 $w_t \leftarrow \mathbf{p}_{e_t}^{(t-1)}$;
6 $\mathcal{L}_{\text{SL-MC}}^{\ell} \leftarrow \mathcal{L}_{\text{SL-MC}}^{\ell} + w_t \left\| \mathbf{h}_{e_t}^{\mathcal{S}} - \tilde{\mathbf{h}}_{e_t}^{\mathcal{T}} \right\|_2^2$;
7 $\mathbf{d} \leftarrow \mathbf{1}$; $\mathbf{d}_{e_t} \leftarrow \alpha$;
8 $\mathbf{p}^{(t)} \leftarrow \text{NORMALIZE}\big(\mathbf{p}^{(t-1)} \odot \mathbf{d}\big)$;
9 **return** $\mathcal{L}_{\text{SL-MC}}$;

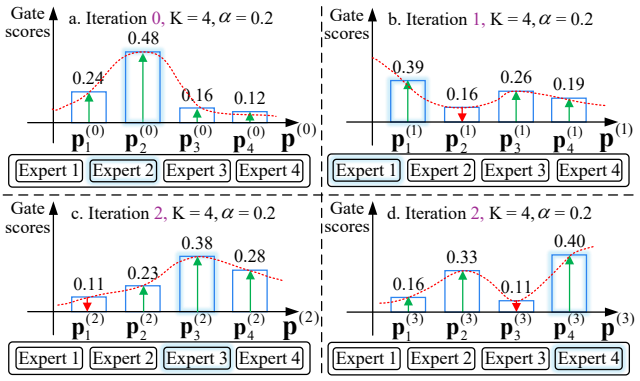

Figure 4: Illustration of MC expert exploration in one MoE layer, where MC draws expert 2, 1, 3, and 4 sequentially.

The MC expert exploration process is shown in Algorithm 1 and Figure 4, where $\text{NORMALIZE}(\mathbf{v}) = \mathbf{v} / \sum_e v_e$, and $\odot$ denotes element-wise product.

During distillation, MC receives the aggregated teacher gate $\tilde{\mathbf{p}}^{\mathcal{T}} = \{\tilde{\mathbf{p}}_e^{\mathcal{T}}\}_{e=1}^{E}$, the aggregated teacher hidden states $\tilde{\mathbf{h}}^{\mathcal{T}} = \{\tilde{\mathbf{h}}_e^{\mathcal{T}}\}_{e=1}^{E}$, the student hidden states $\mathbf{h}^{\mathcal{S}} = \{\mathbf{h}_e^{\mathcal{S}}\}_{e=1}^{E}$, the exploration length K, and the damping α as input. Specifically, SL-MC copies $\tilde{\mathbf{p}}^{\mathcal{T}}$ to $\mathbf{p}^{(0)}$ and performs K Monte Carlo iterations (Line 1-2 in Algorithm 1). At iteration t, SL-MC draws one expert e_t from the distribution $\mathbf{p}^{(t-1)}$, accumulates an importance-weighted SL-MC loss $w_t \|\mathbf{h}_{e_t}^{\mathcal{S}} - \tilde{\mathbf{h}}_{e_t}^{\mathcal{T}}\|_2^2$ with $w_t = \mathbf{p}_{e_t}^{(t-1)}$ (Line 4-6 in Algorithm 1). To bias future sampling toward yet-unused experts, the selected probability mass is damped by α. The resulting vector is passed through $\text{NORMALIZE}(\cdot)$, which rescales it to sum to 1 and thereby maintains the simplex constraint (Line 7-8 in Algorithm 1). After K iterations, the MC loss in the ℓ-th layer can be calculated as follows:

$$\mathcal{L}_{\text{SL-MC}}^{\ell} = \sum_{t=1}^{K} w_t \left\| \mathbf{h}_{e_t}^{\mathcal{S}} - \tilde{\mathbf{h}}_{e_t}^{\mathcal{T}} \right\|_2^2 \tag{7}$$

By the MC expert exploration, we update the gating probabilities in real time during sampling. This enables us to sample experts with low initial probabilities during distillation as well, as shown in Figure 4, enabling more comprehensive learning for the student model. By accumulating $\mathcal{L}_{\text{SL-MC}}$ of all layers, we obtain the total loss $\mathcal{L}_{\text{MC}}$ as follows:

$$\mathcal{L}_{\text{MC}} = \sum_{\ell=1}^{N} \mathcal{L}_{\text{SL-MC}}^{\ell} \tag{8}$$

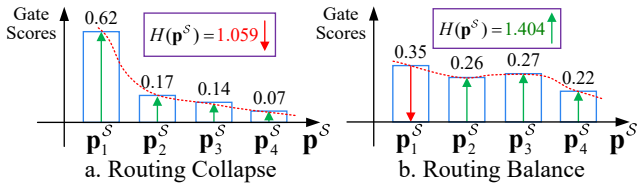

Figure 5: Relation between routing collapse and entropy of gates. Balanced routing brings higher routing entropy.

Entropy-Aware Router Distillation

In order to enable student experts to learn the knowledge utilization capabilities of teacher experts, we propose router distillation for the MoE architecture to align the gate units between teacher experts and student experts. Specifically, for the teacher gate $\tilde{\mathbf{p}}^{\mathcal{T}}$ and the student gate $\mathbf{p}^{\mathcal{S}}$ corresponding to the equal-width mapping π, we calculate the single-layer router distillation loss $\mathcal{L}_{\text{SL-RD}}^{\ell}$ in layer ℓ as follows:

$$\mathcal{L}_{\text{SL-RD}}^{\ell} = \text{KL}\big(\tilde{\mathbf{p}}^{\mathcal{T}} \parallel \mathbf{p}^{\mathcal{S}}\big) \tag{9}$$

Moreover, in order to balance student expert routing choices and prevent routing collapse, inspired by previous works (Shen et al. 2023; Pan et al. 2024), we propose entropy-aware routing (ER) distillation. Specifically, in the routing collapse case, the gate scores of a few student experts tend to be close to 1, while those of other student experts tend to be close to 0, resulting in low overall gate entropy, as shown in Figure 5. To balance the gates of student experts, we add routing entropy during router distillation and use entropy increase to alleviate routing collapse. The entropy-aware router distillation loss in layer ℓ is denoted as:

$$\mathcal{L}_{\text{SL-ERD}}^{\ell} = \text{KL}\big(\tilde{\mathbf{p}}^{\mathcal{T}} \parallel \mathbf{p}^{\mathcal{S}}\big) - \beta\, H\big(\mathbf{p}^{\mathcal{S}}\big) \tag{10}$$

$$H(\mathbf{p}^{\mathcal{S}}) = -\sum_{e} \mathbf{p}_e^{\mathcal{S}} \log \mathbf{p}_e^{\mathcal{S}} \tag{11}$$

where β is the hyperparameter to calculate the entropy. By accumulating $\mathcal{L}_{\text{SL-ERD}}^{\ell}$ of all layers, we obtain the total loss $\mathcal{L}_{\text{ERD}}$ as follows:

$$\mathcal{L}_{\text{ERD}} = \sum_{\ell=1}^{N} \mathcal{L}_{\text{SL-ERD}}^{\ell} \tag{12}$$

where $\mathcal{L}_{\text{SL-ERD}}^{\ell}$ denotes the SL-ERD loss in the ℓ-th layer. The final distillation loss $\mathcal{L}_{\text{total}}$ is defined as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{KD}} + \lambda_1 \mathcal{L}_{\text{MC}} + \lambda_2 \mathcal{L}_{\text{ERD}} \tag{13}$$

where λ_1 and λ_2 are used to balance $\mathcal{L}_{\text{MC}}$ and $\mathcal{L}_{\text{ERD}}$.

Experiments

Experimental Setup

Datasets and Benchmarks. For general fields, we distill on Databricks-Dolly-15k (**Dolly**) (Conover et al. 2023) train sets. Furthermore, we test the results on **Dolly** test sets, Self-Instruct (**SelfInst**) (Wang et al. 2022), Vicuna Evaluation (**Vicuna**) (Chiang et al. 2023), Super Natural Instructions (**S-NI**) (Wang et al. 2022), and Unnatural Instructions (**UnNI**) (Honovich et al. 2023) for a comprehensive evaluation. For specific fields, (1) **Medical**: we perform distillation on GenMedGPT (Li et al. 2023) and evaluate on medical tasks (clinical knowledge (**CK**), professional medicine (**PM**), and college medicine (**CM**)) in MMLU (Hendrycks et al. 2021). (2) **Legal**: we distill on Lawyer-Instruct (Alignment Lab AI 2024) and US-Terms (Chalkidis et al. 2023) and evaluate on law tasks (international law (**IL**) and professional law (**PL**)) in MMLU.

Models. We select LLMs based on MoE for both student models and teacher models. Due to limited computational resources, following (Dou et al. 2024; Tian et al. 2024), we construct lightweight MoE models from existing dense models with varying sizes. For the teacher models, we choose Llama3.1-8B (Grattafiori et al. 2024) and Qwen3-4B (Yang et al. 2025) as base models, then add MoE layers to each layer of the original models to obtain **Llama3.1-8B-MoE** and **Qwen3-4B-MoE**. For the student models, we select Llama3.2-1B (Grattafiori et al. 2024) and Qwen3-0.6B (Yang et al. 2025) as the base models and add MoE layers to obtain **Llama3.2-1B-MoE** and **Qwen3-0.6B-MoE**.

Baselines. We compare our method with 10 baselines. (1) 8 *dense-to-dense* methods: **KD** (Hinton 2014), **SeqKD** (Kim and Rush 2016), **ImitKD** (Lin et al. 2020), **GKD** (Agarwal et al. 2024), **MiniLLM** (Gu et al. 2024), **Distillm** (Ko et al. 2024), **Distillm-2** (Ko et al. 2025), and **ABKD** (Wang et al. 2025). (2) 2 *moe-to-dense* methods: **MoE-KD** (Xu et al. 2024a) and **SAR** (Kim, Chu, and Yang 2025).

Settings. We fine-tune teacher models on each dataset and then distill student models. We evaluate with Rouge-L, GPT-4, and accuracy scores. We set the hyperparameters λ_1, λ_2, and α to 1, 1, and 0.2, respectively. In the main experiments, we set the exploration length K to 3, the number of experts in each model to 4, and the weight of entropy β to 0.1, and we investigate them in exploratory experiments. All experiments are performed with eight NVIDIA A100 GPUs. Our code is available at https://github.com/ljj-007/moedistill.

Experimental Results

Main Results on General Fields. The results of Rouge-L and GPT-4 scores in general fields are shown in Table 1 and Figure 6. First, we observe that our B-Distill achieves the highest Rouge-L scores compared to all baselines, as shown in Table 1. Specifically, in the distillation of Llama3.2-1B-MoE, B-Distill achieves 1.4 to 3.9 Rouge-L scores higher than the *dense-to-dense* methods and 2.0 to 2.1 scores higher than the *moe-to-dense* methods. The advantage is even more pronounced in the distillation of Qwen3-0.6B-MoE, where our B-Distill outperforms the *dense-to-dense* methods by 1.7-6.6 scores and the *moe-to-dense* methods by 3.1-4.2 scores. This demonstrates that our balanced *moe-to-moe* transfer yields more efficient distillation across architectures (Llama3 and Qwen3) and scales (8B → 1B and 4B → 0.6B).

Methods	Dolly	SelfInst	S-NI	UnNI	Vicuna	Avg.
Llama3.1-8B-MoE → Llama3.2-1B-MoE						
Teacher	31.64	21.31	16.95	31.21	24.66	25.15
Student	24.66	16.97	15.91	27.62	20.91	21.21
KD	25.63	18.61	15.11	30.23	18.98	21.71
SeqKD	26.44	18.43	15.42	30.51	18.47	21.85
ImitKD	19.79	17.83	15.63	25.14	20.81	19.84
GKD	26.59	18.11	15.93	30.62	18.81	22.01
MiniLLM	25.43	16.17	16.47	30.91	19.62	21.72
Distillm	27.87	16.69	16.79	31.18	17.15	21.94
Distillm-2	28.43	17.83	16.77	30.35	18.21	22.32
ABKD	28.01	17.95	15.37	28.57	19.75	21.94
MoE-KD	26.15	18.74	14.78	29.72	19.24	21.73
SAR	27.21	18.39	14.94	28.41	19.83	21.76
B-Distill	**29.62**	**19.89**	**16.91**	**31.62**	**20.76**	**23.76**
Qwen3-4B-MoE → Qwen3-0.6B-MoE						
Teacher	22.43	19.22	11.33	11.21	21.27	17.09
Student	17.81	14.63	2.01	2.39	16.35	10.64
KD	18.72	15.49	1.77	1.32	18.77	11.21
SeqKD	19.13	14.85	2.44	2.61	17.42	11.29
ImitKD	19.78	17.83	7.63	5.11	20.77	14.22
GKD	17.88	17.05	9.48	6.97	20.57	14.39
MiniLLM	14.28	12.87	1.53	1.78	19.63	9.82
Distillm	19.16	17.03	9.21	8.58	19.41	14.68
Distillm-2	18.34	16.74	6.48	8.07	18.94	13.71
ABKD	14.23	11.68	4.53	6.54	17.86	10.97
MoE-KD	19.23	15.17	3.73	4.66	17.94	12.15
SAR	19.47	15.64	5.72	7.27	18.13	13.25
B-Distill	**21.71**	**18.65**	**9.52**	**10.53**	**21.43**	**16.37**

Table 1: Main results on general fields with Rouge-L scores (%). The bold scores are the highest scores among baselines.

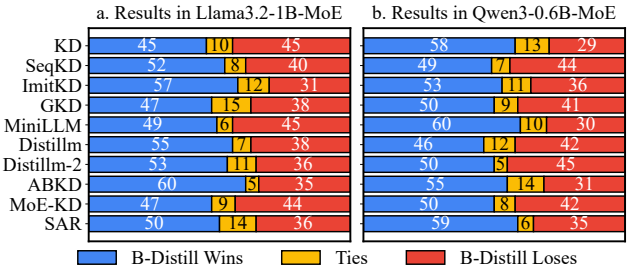

Figure 6: Main results on general fields with GPT-4 scores.

Second, to further validate that our B-Distill generates higher-quality responses, we test GPT-4 scores, as shown in Figure 6. Specifically, we randomly sample 100 prompts from all datasets in the general fields. For each prompt, we collect the responses from B-Distill and each baseline, then ask the `gpt4-turbo` evaluator (temperature 0) to select the better answer or declare a tie. As summarized in Figure 6, B-Distill wins the majority of comparisons against all *dense-to-dense* baselines (e.g., 53 wins against Distillm-2 in Llama3.2-1B-MoE) and likewise dominates the *moe-to-*

Methods	CK	PM	CM	PL	IL	Avg.
Llama3.1-8B-MoE → Llama3.2-1B-MoE						
Teacher	0.66	0.65	0.55	0.42	0.66	0.59
Student	0.29	0.24	0.27	0.23	0.41	0.29
KD	0.33	0.29	0.32	0.28	0.47	0.34
SeqKD	0.34	0.30	0.27	0.28	0.52	0.34
ImitKD	0.33	0.27	0.32	0.27	0.45	0.33
GKD	0.32	0.25	0.29	0.28	0.35	0.30
MiniLLM	0.34	0.28	0.31	0.29	0.54	0.35
Distillm	0.33	0.28	0.29	0.28	0.55	0.35
Distillm-2	0.35	0.29	0.32	0.29	0.55	0.36
ABKD	0.32	0.28	0.29	0.28	0.53	0.34
MoE-KD	0.34	0.24	0.31	0.27	0.48	0.33
SAR	0.31	0.26	0.29	0.29	0.51	0.33
B-Distill	**0.37**	**0.31**	**0.34**	**0.32**	**0.58**	**0.38**
Qwen3-4B-MoE → Qwen3-0.6B-MoE						
Teacher	0.69	0.63	0.68	0.42	0.66	0.62
Student	0.43	0.34	0.38	0.25	0.51	0.38
KD	0.46	0.36	0.41	0.27	0.54	0.41
SeqKD	0.45	0.36	0.43	0.28	0.52	0.41
ImitKD	0.46	0.38	0.43	0.27	0.53	0.41
GKD	0.47	0.37	0.44	0.28	0.52	0.42
MiniLLM	0.46	0.36	0.43	0.29	0.53	0.41
Distillm	0.46	0.37	0.45	0.28	0.54	0.42
Distillm-2	0.47	0.39	0.39	0.29	0.54	0.42
ABKD	0.46	0.36	0.42	0.28	0.52	0.41
MoE-KD	0.45	0.36	0.41	0.27	0.54	0.41
SAR	0.46	0.37	0.43	0.27	0.54	0.41
B-Distill	**0.51**	**0.41**	**0.48**	**0.32**	**0.58**	**0.46**

Table 2: Main results on specific fields with accuracy.

dense baselines (e.g., 59 wins against SAR in Qwen3-0.6B-MoE). It confirms that the qualitative gains recognized by Rouge-L are also validated by a strong referee and underscoring the effectiveness of our B-Distill.

Main Results on Specific Fields. The main results on specific fields are shown in Table 2. Across all three medical tasks (**CK**, **PM**, and **CM**) and two legal tasks (**PL** and **IL**), our B-Distill consistently delivers the highest accuracy scores, surpassing ten strong baselines on both model lines (Llama3.2-1B-MoE and Qwen3-0.6B-MoE). On average, our B-Distill narrows the performance gap between the teacher and the student by roughly one-third, raising absolute scores by about 9 scores for the Llama3 students and 6 points for the Qwen3 students. Specifically, in the medical field, our B-Distill gains of 2-4 scores over the best prior method, with the lighter Qwen3-0.6B benefiting most (3 scores on average). In the legal field, our B-Distill boosts PL and IL by roughly 3 scores for Llama3 students and about 2 scores for the Qwen3 students. These consistent improvements indicate that balanced expert coverage and stabilized routing are particularly vital for knowledge-intensive settings. Meanwhile, it underscores the promise of B-Distill for deploying compact, reliable MoE models in safety-critical fields such as medicine and law domains.

Ablation Results. The ablation results are shown in Table 3, where MC and ERD denote the Monte Carlo expert

Methods	Dolly	SelfInst	S-NI	UnNI	Vicuna	Avg.
Llama3.1-8B-MoE → Llama3.2-1B-MoE						
B-Distill	**29.62**	**19.89**	**16.91**	**31.62**	**20.76**	**23.76**
w/o MC	28.06	18.95	16.78	30.52	19.71	22.80
w/o ERD	27.92	18.71	15.82	29.78	19.61	22.37
Qwen3-4B-MoE → Qwen3-0.6B-MoE						
B-Distill	**21.71**	**18.65**	**9.52**	**10.53**	**21.43**	**16.37**
w/o MC	16.69	17.88	6.49	6.72	17.03	12.96
w/o ERD	12.21	13.51	4.42	4.48	17.15	10.35

Table 3: Ablation results with Rouge-L scores (%).

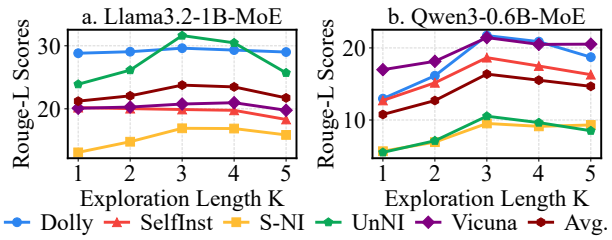

Figure 7: Exploration results of different exploration lengths K from 1 to 5 in general fields with Rouge-L scores (%).

exploration and entropy-aware router distillation, respectively. First, removing either MC or ERD leads to a clear degradation of Rouge-L. Specifically, when MC is removed, the average score drops from 23.76 to 22.80 in Llama3.2-1B-MoE and from 16.37 to 12.96 in Qwen3-0.6B-MoE. When ERD is removed, the average scores decline to 22.37 and 10.35, respectively. This demonstrates the effectiveness of our proposed module, MC and ERD, which can boost the distillation for MoE LLMs. Second, we notice that eliminating ERD produces the steeper losses (1.39↓ vs. 0.96↓ for Llama3.2-1B-MoE and 6.02↓ vs. 3.41↓ for Qwen3-0.6B-MoE), which indicates that mitigating routing imbalance and preventing load collapse is even more critical than expanding expert coverage, making ERD the dominant contributor to the overall effectiveness of our B-Distill.

Effect of Exploration Length. Figure 7 reports the effect of varying the Monte Carlo exploration length. Distillation performance increases monotonically as the length K grows from 1 to 3, but declines once $K > 3$. For Llama3.2-1B-MoE, the average Rouge-L rises from 21 to 25, with the **UnNI** dataset contributing more than seven scores of that gain. The **Dolly** dataset still achieves a modest improvement of about one score. The smaller Qwen3-0.6B-MoE is even more sensitive, whose mean Rouge-L climbs from 16 to 21 at $K = 3$ and then falls as K increases to 4 and 5. These bell-shaped curves indicate that a moderate exploration length ($K = 3$) offers the best trade-off, exposing the student to a richer set of teacher experts while avoiding the noise and load imbalance introduced by overly long random walks.

Effect of Expert Number. We test the distillation performance with different numbers of experts, as shown in Figure 8. Firstly, across both Llama and Qwen models, distilla-

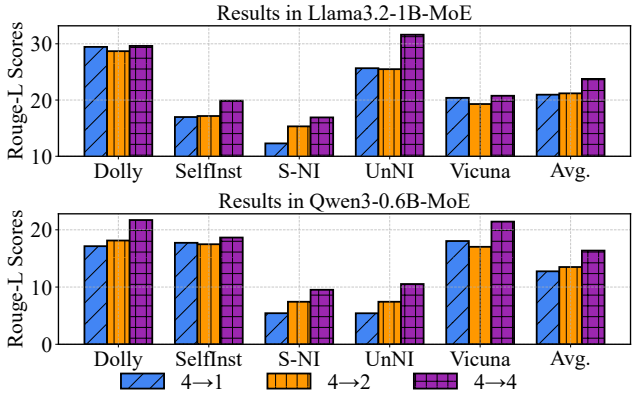

Figure 8: Exploration results of different expert numbers for student models in general fields with Rouge-L scores (%). $a \rightarrow b$ denotes distilling from the teacher model with a experts to the student model with b experts in each layer.

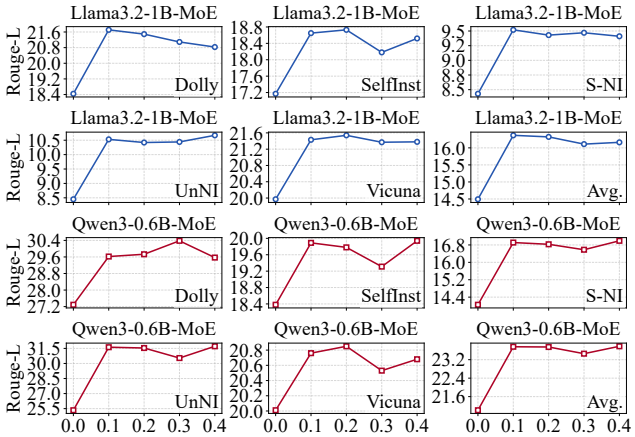

Figure 9: Exploration results of different entropy weights β from 0 to 0.4 in general fields with Rouge-L scores (%).

tion performance improves monotonically with the number of experts on four of five benchmarks (**Dolly**, **S-NI**, **UnNI**, **Vicuna**) and on average. For example, on **UnNI**, the $4 \rightarrow 4$ student surpasses $4 \rightarrow 1$ by 5.2↑ scores, and the average scores rise from 20.9 to 23.8 in Llama3.2-1B-MoE and from 12.4 to 15.6 in Qwen3-0.6B-MoE. These trends indicate that our B-Distill scale positively correlates with expert capacity: a larger expert palette enables the student to capture finer teacher specialization, thereby transferring more diverse knowledge. Consequently, *moe-to-moe* distillation ($4 \rightarrow 2$ and $4 \rightarrow 4$) is demonstrably more beneficial than *moe-to-dense* ($4 \rightarrow 1$). Secondly, while the $4 \rightarrow 4$ variant is always best on average, the incremental gain from $4 \rightarrow 2$ to $4 \rightarrow 4$ is smaller than that from $4 \rightarrow 1$ to $4 \rightarrow 2$ (e.g., 1.3↑ vs 2.7↑ scores in Qwen3-0.6B-MoE). This suggests a mild law of diminishing returns: doubling the expert count improves coverage but also raises routing overhead. In practice, $4 \rightarrow 2$ strikes an attractive balance that captures about 80% of the full MoE gain while cutting the number of experts by half.

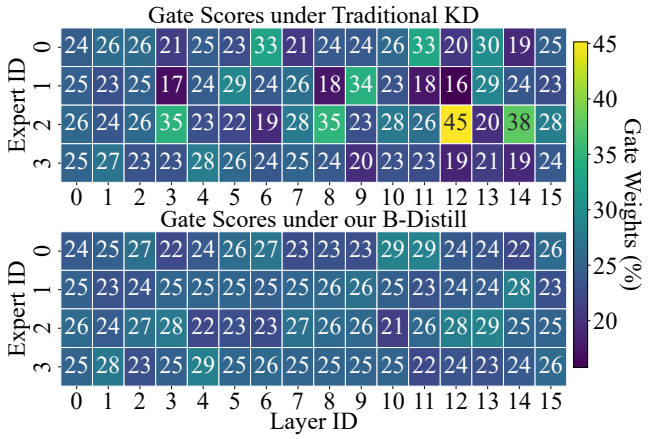

Figure 10: The average gate scores (%) in different layers of distilled Llama3.2-1B-MoE tested on the Dolly dataset.

Effect of Entropy Weights. Figure 9 shows the effect of different entropy weights β. Firstly, raising β from 0 to 0.1 consistently yields a marked performance increase on each dataset. Specifically, the average scores climb from 14.5% to 16.0% with Llama3.2-1B-MoE and from 21.5% to 23.3% with Qwen3-0.6B-MoE. It confirms that entropy-aware routing alleviates expert collapse and helps the student harvest diverse teacher knowledge. Secondly, performance remains essentially flat as β grows from 0.1 to 0.4, fluctuations stay within 0.4 scores on both Llama and Qwen backbones, indicating that once a minimal amount of entropy regularization is present, our entropy-aware routing distillation is robust to further tuning. Consequently, we fix β to 0.1 in all subsequent experiments, achieving most of the attainable gain.

Visualization of Gate Scores for Students. Figure 10 compares the gate scores in each layer learned by the four-expert student Llama3.2-1B-MoE under traditional KD (top) and our B-Distill (bottom). With traditional KD, the router suffers clear load collapse. Specifically, at Layer 3, Expert 2 absorbs 0.35 of the traffic while Expert 1 receives only 0.17; Layer 7 concentrates 0.33 on Expert 0; Layer 9 directs 0.34 to Expert 1; and the most extreme case appears at Layer 12, where Expert 2 dominates with 0.45 and the remaining experts fall below 0.24. These sharp imbalances show that naive logit matching cannot maintain a balanced expert allocation. By contrast, our B-Distill keeps each expert share tightly clustered between 0.24 and 0.29 across all 16 layers of Llama3.2-1B-MoE, indicating that our entropy-aware router distillation effectively suppresses routing collapse.

Conclusion

In this work, we study *moe-to-moe* distillation and propose a novel distillation framework B-Distill, which balances the expert coverage and routing distribution. Specifically, to enhance the expert coverage, we propose a Monte Carlo exploration to sample teacher experts. To alleviate the imbalance in gate scores, we propose an entropy-aware router distillation to mitigate the load collapse. In the future, we will explore distilling LLMs based on larger-scale MoEs.

Acknowledgments

We thank the anonymous reviewers for their insightful comments. This work was supported by National Science Foundation of China (Grant Nos.62376057), SEU Innovation Capability Enhancement Plan for Doctoral Students (No. CXJH_SEU 25131), the Start-up Research Fund of Southeast University (RF1028623234), and the Big Data Computing Center of Southeast University.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Agarwal, R.; Vieillard, N.; Zhou, Y.; Stanczyk, P.; Garea, S. R.; Geist, M.; and Bachem, O. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *ICLR*.

Alignment Lab AI. 2024. Lawyer-Instruct. https://huggingface.co/datasets/Alignment-Lab-AI/Lawyer-Instruct. Accessed: 2025-08-01.

Chalkidis, I.; Garneau, N.; Goanta, C.; Katz, D.; and Søgaard, A. 2023. LeXFiles and LegalLAMA: Facilitating English Multinational Legal Language Model Development. In *ACL*.

Chen, T.; Zhang, Z.; JAISWAL, A. K.; Liu, S.; and Wang, Z. 2023. Sparse MoE as the New Dropout: Scaling Dense and Self-Slimmable Transformers. In *ICLR*.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. https://lmsys.org/blog/2023-03-30-vicuna/. Accessed: 2025-08-01.

Clark, A.; de Las Casas, D.; Guy, A.; Mensch, A.; Paganini, M.; Hoffmann, J.; Damoc, B.; Hechtman, B.; Cai, T.; Borgeaud, S.; et al. 2022. Unified scaling laws for routed language models. In *ICML*.

Conover, M.; Hayes, M.; Mathur, A.; Xie, J.; Wan, J.; Shah, S.; Ghodsi, A.; Wendell, P.; Zaharia, M.; and Xin, R. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Dou, S.; Zhou, E.; Liu, Y.; Gao, S.; Shen, W.; Xiong, L.; Zhou, Y.; Wang, X.; Xi, Z.; Fan, X.; et al. 2024. LoRAMoE: Alleviating world knowledge forgetting in large language models via MoE-style plugin. In *ACL*.

Eghbali, A.; and Pradel, M. 2024. De-hallucinator: Iterative grounding for llm-based code completion. *arXiv preprint arXiv:2401.01701*.

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

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Gu, Y.; Dong, L.; Wei, F.; and Huang, M. 2024. MiniLLM: Knowledge Distillation of Large Language Models. In *ICLR*.

Hao, Z.; Guo, J.; Han, K.; Tang, Y.; Hu, H.; Wang, Y.; and Xu, C. 2023. One-for-all: Bridge the gap between heterogeneous architectures in knowledge distillation. In *NeurIPS*.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring Massive Multitask Language Understanding. In *ICLR*.

Hinton, G. 2014. Distilling the Knowledge in a Neural Network. In *NIPS*.

Ho, N.; Schmid, L.; and Yun, S.-Y. 2023. Large Language Models Are Reasoning Teachers. In *ACL*.

Honovich, O.; Scialom, T.; Levy, O.; and Schick, T. 2023. Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor. In *ACL*.

Hsieh, C.-Y.; Li, C.-L.; Yeh, C.-k.; Nakhost, H.; Fujii, Y.; Ratner, A.; Krishna, R.; Lee, C.-Y.; and Pfister, T. 2023. Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes. In *Findings of ACL*.

Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J.; and Hinton, G. E. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1): 79–87.

Jiang, A. Q.; Sablayrolles, A.; Roux, A.; Mensch, A.; Savary, B.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Hanna, E. B.; Bressand, F.; et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Jordan, M. I.; and Jacobs, R. A. 1994. Hierarchical mixtures of experts and the EM algorithm. *Neural computation*, 6(2): 181–214.

Kim, G.; Chu, G.; and Yang, E. 2025. Every Expert Matters: Towards Effective Knowledge Distillation for Mixture-of-Experts Language Models. *arXiv preprint arXiv:2502.12947*.

Kim, Y.; and Rush, A. M. 2016. Sequence-level knowledge distillation. In *EMNLP*.

Ko, J.; Chen, T.; Kim, S.; Ding, T.; Liang, L.; Zharkov, I.; and Yun, S.-Y. 2025. DistiLLM-2: A Contrastive Approach Boosts the Distillation of LLMs. In *ICML*.

Ko, J.; Kim, S.; Chen, T.; and Yun, S.-Y. 2024. DistiLLM: Towards Streamlined Distillation for Large Language Models. In *ICLR*.

Lepikhin, D.; Lee, H.; Xu, Y.; Chen, D.; Firat, O.; Huang, Y.; Krikun, M.; Shazeer, N.; and Chen, Z. 2021. GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding. In *ICLR*.

Lewis, M.; Bhosale, S.; Dettmers, T.; Goyal, N.; and Zettlemoyer, L. 2021. Base layers: Simplifying training of large, sparse models. In *ICML*.

Li, X.; Liu, J.; and Wang, P. 2025. Can large models teach student models to solve mathematical problems like human beings? a reasoning distillation method via multi-LoRA interaction. In *IJCAI*.

Li, Y.; Li, Z.; Zhang, K.; Dan, R.; Jiang, S.; and Zhang, Y. 2023. ChatDoctor: A Medical Chat Model Fine-Tuned on a Large Language Model Meta-AI (LLaMA) Using Medical Domain Knowledge. *Cureus*, 15(6).

Liang, C.; Zuo, S.; Zhang, Q.; He, P.; Chen, W.; and Zhao, T. 2023. Less is more: Task-aware layer-wise distillation for language model compression. In *ICML*.

Lin, A.; Wohlwend, J.; Chen, H.; and Lei, T. 2020. Autoregressive Knowledge Distillation through Imitation Learning. In *EMNLP*.

Liu, J.; Ke, W.; Wang, P.; Shang, Z.; Gao, J.; Li, G.; Ji, K.; and Liu, Y. 2024. Towards continual knowledge graph embedding via incremental distillation. In *AAAI*.

Liu, M.; Ma, Y.; Yang, Z.; Dan, J.; Yu, Y.; Zhao, Z.; Hu, Z.; Liu, B.; and Fan, C. 2025. Llm4gen: Leveraging semantic representation of llms for text-to-image generation. In *AAAI*.

Pan, B.; Shen, Y.; Liu, H.; Mishra, M.; Zhang, G.; Oliva, A.; Raffel, C.; and Panda, R. 2024. Dense training, sparse inference: Rethinking training of mixture-of-experts language models. *arXiv preprint arXiv:2404.05567*.

Sanh, V. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *NeurIPS*.

Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; and Dean, J. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *ICLR*.

Shen, Y.; Zhang, Z.; Cao, T.; Tan, S.; Chen, Z.; and Gan, C. 2023. ModuleFormer: Modularity Emerges from Mixture-of-Experts. *arXiv preprint arXiv:2306.04640*.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

Team, G.; Kamath, A.; Ferret, J.; Pathak, S.; Vieillard, N.; Merhej, R.; Perrin, S.; Matejovicova, T.; Ramé, A.; Rivière, M.; et al. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Tian, C.; Shi, Z.; Guo, Z.; Li, L.; and Xu, C. 2024. HydraLoRA: An Asymmetric LoRA Architecture for Efficient Fine-Tuning. In *NeurIPS*.

Wang, G.; Yang, Z.; Wang, Z.; Wang, S.; Xu, Q.; and Huang, Q. 2025. ABKD: Pursuing a Proper Allocation of the Probability Mass in Knowledge Distillation via α-β Divergence. In *ICML*.

Wang, Y.; Mishra, S.; Alipoormolabashi, P.; Kordi, Y.; Mirzaei, A.; Naik, A.; Ashok, A.; Dhanasekaran, A. S.; Arunkumar, A.; Stap, D.; et al. 2022. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. In *EMNLP*.

Xie, Z.; Zhang, Y.; Zhuang, C.; Shi, Q.; Liu, Z.; Gu, J.; and Zhang, G. 2024. Mode: A mixture-of-experts model with mutual distillation among the experts. In *AAAI*.

Xu, H.; Liu, H.; Gong, W.; Deng, X.; and Wang, H. 2024a. Sparse Mixture of Experts Language Models Excel in Knowledge Distillation. In *NLPCC*.

Xu, Z.; Wang, P.; Ke, W.; Li, G.; Liu, J.; Ji, K.; Chen, X.; and Wu, C. 2024b. Incorporating schema-aware description into document-level event extraction. In *IJCAI*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Zhang, S.; Zhang, X.; Sun, Z.; Chen, Y.; and Xu, J. 2024. Dual-Space Knowledge Distillation for Large Language Models. In *EMNLP*.

Zhou, Y.; Lei, T.; Liu, H.; Du, N.; Huang, Y.; Zhao, V.; Dai, A. M.; Le, Q. V.; Laudon, J.; et al. 2022. Mixture-of-experts with expert choice routing. In *NeurIPS*.