

Staged Knowledge Distillation Through Least-to-Most Prompting: Optimizing Teacher Guidance via Difficulty-Aware Training

Mengxiang Zhang

The University of Hong Kong
mxzhang6@connect.hku.hk

Lingyuan Liu*

City University of Hong Kong
ly.liu@my.cityu.edu.hk

Abstract

Knowledge distillation (KD) enables the compression of large language models (LLMs) by transferring knowledge from a high-capacity teacher model to a resource-efficient student model, maintaining competitive performance for tasks such as instruction following. However, conventional white-box KD methods often suffer from training-inference mismatches and suboptimal performance due to the asymmetric nature of Kullback-Leibler divergence (KLD) and reliance on computationally expensive student-generated outputs. To address these challenges, we propose Least-to-Most Prompting Knowledge Distillation (L2M-KD), a novel white-box KD method grounded in curriculum learning (CL) and adaptive loss design. L2M-KD employs a two-pronged approach: (1) a CL strategy that ranks training samples by difficulty using Rouge-L scores, partitioning them into easy-to-hard subsets across multiple stages, and (2) an adaptive KD loss that transitions from KLD to skew KLD, dynamically adjusting teacher guidance to mitigate mode-averaging and over-smoothing. Extensive experiments on instruction-following tasks demonstrate that L2M-KD outperforms existing white-box KD methods, achieving superior student model performance with reduced computational overhead by leveraging ground-truth outputs exclusively. Our findings underscore the efficacy of difficulty-aware training and adaptive teacher guidance, offering a computationally efficient and robust approach to LLM compression. The code for our method is publicly available at <https://github.com/liuliyuan6/L2M-KD>.

1 Introduction

Large language models (LLMs) have achieved significant progress in text generation, language understanding, and inference, driven by increased parameter scales and high-quality training data (Ouyang

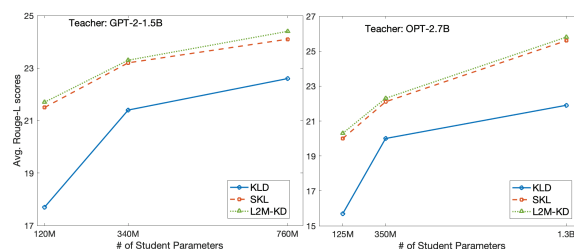


Figure 1: Comparison of the proposed L2M-KD method with white-box KD methods, KLD (off-policy) (Hinton et al., 2015) and SKL (on-policy) (Ko et al., 2024), based on average Rouge-L scores across five evaluation sets. **Left:** GPT-2-1.5B as the teacher with GPT-2 (125M, 340M, 760M) as student models. **Right:** OPT-2.7B as the teacher with OPT (125M, 350M, 1.3B) as student models.

et al., 2022). However, their substantial computational and memory requirements during inference limit practical deployment, particularly in resource-constrained settings like edge computing (Aryan et al., 2023). This has spurred demand for smaller, efficient language models (LMs) that maintain competitive performance in tasks such as text generation (Li et al., 2024b) and tool learning (Gao et al., 2024). Knowledge distillation (KD) (Hinton et al., 2015), a technique that transfers knowledge from a high-capacity teacher model to a smaller student model, has become a cornerstone for compressing LLMs into small LMs, as evidenced by models like Llama 3.2 (Meta, 2024) and DeepSeek-R1 (DeepSeek-AI, 2025).

KD methods for LLMs are broadly classified into black-box and white-box approaches (Yang et al., 2024b). Black-box KD, which leverages only teacher predictions (Kim and Rush, 2016), gained traction due to the proprietary nature of models like GPT-4o (Hurst et al., 2024) and Claude 3.5 (Anthropic, 2024). However, the emergence of open-source LLMs, such as DeepSeek-v3 (Liu et al., 2024) and Qwen 2.5 (Yang et al., 2024a), with performance rivaling proprietary models (Maslej et al., 2025), has shifted focus toward white-box

*Corresponding author.

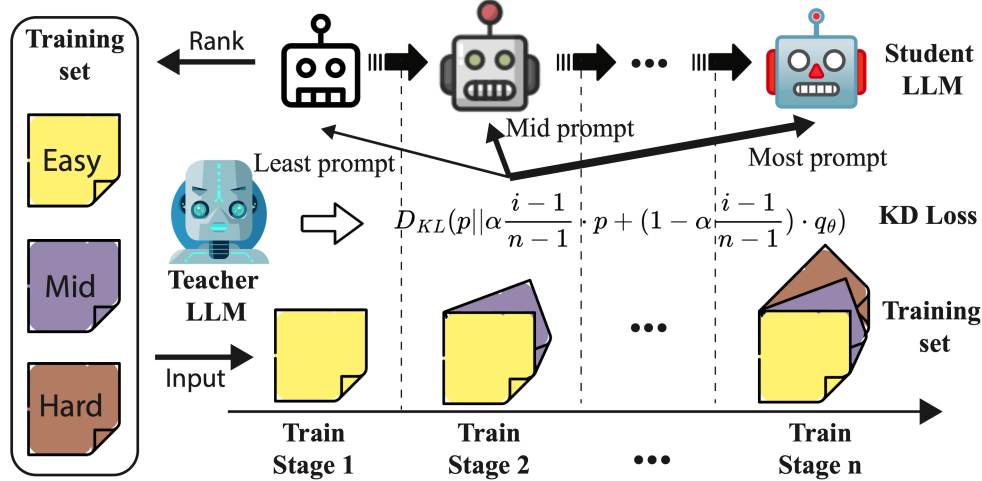


Figure 2: Overview of Least-to-Most Prompting Knowledge Distillation (L2M-KD), illustrating the curriculum learning component that ranks training samples by difficulty using Rouge-L scores and partitions them into easy-to-hard subsets across multiple stages, alongside the adaptive KD loss function that progressively transitions from KLD to SKL to optimize teacher guidance in alignment with the least-to-most prompting principle.

KD. White-box KD offers enhanced student performance and greater control over the distillation process. Existing white-box KD methods primarily optimize the divergence function $D(\cdot \parallel \cdot)$ in the KD loss (Eq. 2), with Kullback-Leibler divergence (KLD) (Hinton et al., 2015) as a standard. Yet, KLD’s asymmetric nature often induces mode-averaging (Gu et al., 2023), prompting alternatives like reverse KLD (Gu et al., 2023), Jensen-Shannon divergence (JSD) (Agarwal et al., 2024), and skew KLD (Ko et al., 2024). These variants, while effective in specific contexts, lack generalizability across tasks and datasets, resulting in inconsistent performance (Agarwal et al., 2024; Ko et al., 2024).

Data curation strategies further influence KD efficacy, with options including ground-truth outputs (GTOs) (Hinton et al., 2015), teacher-generated outputs (TGOs) (Kim and Rush, 2016), and student-generated outputs (SGOs) (Lin et al., 2020; Agarwal et al., 2024). GTOs, though widely used, can cause training-inference mismatches due to the student’s limited capacity, while SGO-based on-policy methods (Gu et al., 2023; Agarwal et al., 2024) mitigate this but incur significant computational costs and risk teacher misguidance (Ko et al., 2024). Thus, a central challenge in white-box KD for LLMs is to design an effective KD loss and leverage GTOs exclusively to boost student performance without excessive computational overhead.

Despite the prevalence of KLD and GTOs in white-box KD, their full potential remains under-exploited, particularly for LLMs. KL’s mode-averaging tendencies and the computational ineffi-

ciencies of SGO-based methods highlight the need for a refined approach. We posit that structuring the training process and dynamically adapting the KD loss can enhance the effectiveness of KLD and GTOs, improving student model performance while sidestepping the limitations of existing methods. Empirical evidence suggests that student models benefit from progressive training on samples of increasing difficulty, paired with tailored teacher guidance (Bengio et al., 2009).

Drawing inspiration from curriculum learning (CL) (Bengio et al., 2009) and the educational strategy of least-to-most prompting (Libby et al., 2008), where guidance scales with task complexity, we propose a staged KD method. This approach mirrors human learning by starting with simpler tasks and incrementally introducing challenges, adjusting the KD loss to provide optimal teacher guidance at each stage. Our motivation is rooted in the observation that such a structured process can stabilize knowledge transfer, mitigate mode-averaging, and maximize the utility of GTOs, offering a computationally efficient alternative to on-policy methods.

We present **Least-to-Most Prompting Knowledge Distillation (L2M-KD)**, a novel white-box KD method that combines a CL framework with an adaptive KD loss to optimize teacher guidance based on training difficulty (See Fig. 2). L2M-KD operates in two key phases: (1) a curriculum learning framework ranks training samples by difficulty using Rouge-L scores between GTOs and SGOs, partitioning the dataset into easy-to-hard

subsets across multiple stages; (2) an adaptive KD loss transitions from KLD to skew KLD (SKL) (Ko et al., 2024), increasing teacher prompting as difficulty rises, following the least-to-most prompting principle.

Our contributions are threefold:

- **Staged KD Method:** L2M-KD introduces a difficulty-aware, multi-stage training process inspired by least-to-most prompting, enhancing knowledge transfer efficiency.
- **Adaptive Loss Design:** By dynamically adjusting the KD loss from KLD to SKL, L2M-KD addresses mode-averaging and over-smoothing, aligning teacher guidance with the student’s learning progression.
- **Superior Performance:** Extensive evaluations show that L2M-KD achieves superior performance for student LMs on instruction following tasks across multiple white-box KD approaches, using only GTOs to minimize computational costs.

2 Background and Rethinking

2.1 Background

White-Box KD for Auto-regressive LMs. In white-box KD, a student LM is trained to mimic a larger teacher LM using a dataset of source-target sequence pairs (x, y) . The student optimizes two objectives: (1) minimizing the cross-entropy loss between the ground-truth target sequence y and the student’s conditional distribution $q_\theta(y|x)$, and (2) minimizing the KD loss, which measures the divergence between the teacher’s token-level distribution $p(y|x)$ and the student’s distribution $q_\theta(y|x)$.

The cross-entropy loss is defined as:

$$L_{ce} = - \sum_{i=1}^{|y|} \log q_\theta(y_i|x, y_{<i}), \quad (1)$$

where $q_\theta(y_i|x, y_{<i})$ is the student’s probability for token y_i given input x and prior tokens $y_{<i}$.

The KD loss is formulated as:

$$L_{kd} = \sum_{i=1}^{|y|} D(p(y_i|x, y_{<i}) \parallel q_\theta(y_i|x, y_{<i})), \quad (2)$$

where $D(\cdot \parallel \cdot)$ quantifies the divergence between the teacher and student distributions.

The total loss combines both objectives:

$$L_s = (1 - \beta)L_{ce} + \beta L_{kd}, \quad (3)$$

where $\beta \in [0, 1]$ balances the contributions of the cross-entropy and KD losses.

Limitations of the Current Methods. Current white-box KD methods primarily focus on optimizing the divergence function $D(\cdot \parallel \cdot)$ in Eq. (2), such as KLD (Hinton et al., 2015). However, KLD’s asymmetric nature can lead to mode-averaging, where the student model learns an overly smooth distribution to cover the teacher’s support set (Gu et al., 2023). Variants like reverse KLD (Gu et al., 2023), JSD (Agarwal et al., 2024), and skew KLD (Ko et al., 2024) show task-specific improvements but lack systematic evaluation, resulting in suboptimal performance and variability across tasks (Agarwal et al., 2024; Ko et al., 2024). Data curation strategies also impact KD effectiveness. Ground-truth outputs (Hinton et al., 2015) can cause training-inference mismatches due to the student’s limited capacity compared to the teacher (Gu et al., 2023). Student-generated outputs (SGOs) (Lin et al., 2020; Agarwal et al., 2024) mitigate this but introduce challenges: overuse increases computational costs (up to 80% of training time), while underuse degrades performance (Ko et al., 2024). Balancing SGO usage and selecting an effective KD loss remain critical challenges for efficient and robust KD.

2.2 Rethinking KD Loss and GTO Utilization

Empirical Analysis. Despite limitations, KLD and GTOs remain foundational in white-box KD due to their stability across tasks (Hinton et al., 2015). To explore their potential, we conducted experiments using the Dolly dataset (Conover et al., 2023), with GPT-2 (1.5B) as the teacher and GPT-2 (0.1B) as the student (Radford et al., 2019), evaluating ROUGE-L scores (Lin, 2004) on the validation set (details in Section 4).

Inspired by CL (Wang et al., 2021), we ranked Dolly training samples by difficulty based on ROUGE-L scores between SGOs and GTOs, selecting the top 25% (2.5K) as an easy subset and the bottom 25% (2.5K) as a hard subset. We compared KD performance using KLD and skew KLD (SKL) (Ko et al., 2024) as loss functions, and off-policy (100% GTOs) and on-policy (50% GTOs, 50% SGOs) (Agarwal et al., 2024) data curation strategies.

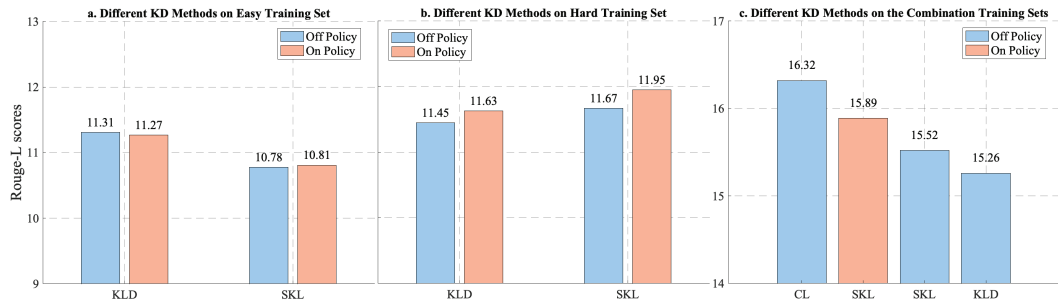


Figure 3: Results of empirical analysis on the Dolly dataset, comparing ROUGE-L scores for KLD and SKL loss functions under off-policy and on-policy data curation strategies across easy and hard subsets, alongside the performance of the proposed two-stage CL-based KD method.

On the easy subset, KLD outperformed SKL under both data strategies, with no significant difference between off- and on-policy methods (Figure 3(a)). Conversely, on the hard subset, SKL outperformed KLD, and on-policy methods surpassed off-policy ones (Figure 3(b)). Motivated by these findings, we proposed a two-stage CL-based KD method: (1) train with KLD on the easy subset, and (2) fine-tune with SKL on a mixed easy-hard subset, using the first-stage checkpoint as initialization. This approach outperformed traditional white-box KD methods, with SKL and on-policy strategies showing superior ROUGE-L scores (Figure 3(c)).

Theoretical Analysis. The results in Figure 3(c) are supported by theoretical insights into the capacity gap between teacher and student models in white-box KD. The student’s limited parameters often lead to a divergence between the teacher’s and student’s distributions, causing parameter overwriting, degraded performance, or biased outputs (Zhang et al., 2024). The CL-based KD approach mitigates this by progressing from easy to hard samples. In the first stage, KLD’s flatter loss landscape stabilizes learning on easy samples, reducing mode collapse (Li et al., 2024a). In the second stage, SKL’s sharper landscape on harder samples prevents over-smoothing, leveraging prior training to maintain stability (Ko et al., 2024). This staged approach, using KLD and GTOs, optimizes knowledge transfer without relying on computationally costly on-policy strategies (Wang et al., 2021). Furthermore, theoretical insights from Shing et al. (2025) suggest that SKL—which introduces an intermediate teacher model that interpolates between the output distributions of the teacher and student—provides a target distribution of moderate capacity. This intermediate target effectively narrows the capacity gap between teacher and student, thereby mitigating

distributional divergence and reducing the risk of mode collapse during distillation.

3 Methodology

3.1 Motivation

Inspired by the observations in Section 2.2, we propose a novel approach to enhance KD by leveraging a multi-stage training process inspired by educational psychology. Specifically, we draw on the concept of *least-to-most prompting* (Libby et al., 2008), where a teacher progressively increases the complexity of guidance to facilitate student learning. In KD, this translates to a structured curriculum that prioritizes easier training examples in early stages and gradually introduces more challenging ones, coupled with adaptive loss functions that evolve from minimal to maximal teacher guidance.

Our approach is motivated by the distinct properties of two KD loss functions: KLD and Skew SKL. KLD enables the student model to directly learn the teacher’s output distribution without smoothing, which is effective for simpler tasks. In contrast, SKL introduces smoothing by mixing the teacher’s and student’s distributions, providing more robust guidance for complex tasks (Ko et al., 2024). As demonstrated by Shing et al. (2025), combining KLD and SKL within a least-to-most prompting framework facilitates a gradual and difficulty-aware transfer of knowledge, aligning loss function dynamics with the evolving capacity of the student model across training stages. By the above analysis, we hypothesize that a staged KD process—where training examples are ranked by difficulty and loss functions are adapted to the training stage—can improve the student model’s performance while maintaining stability and avoiding computationally expensive on-policy strategies.

To this end, we introduce *Least-to-Most Prompting Knowledge Distillation* (L2M-KD), a novel KD method that integrates a CL strategy with adaptive loss functions. L2M-KD operates in three key steps: (1) ranking training examples by difficulty to create easy-to-hard subsets, (2) conducting multi-stage training with progressively harder subsets, and (3) employing adaptive loss functions that transition from KLD to SKL to simulate least-to-most prompting.

3.2 Least-to-Most Prompting Knowledge Distillation

L2M-KD combines a CL framework with an adaptive KD loss function to optimize the distillation process. The method consists of two core components: a CL strategy for difficulty-aware sample ranking and a dynamic loss function that adjusts the degree of teacher guidance across training stages. The overall process is outlined in Algorithm 1.

Algorithm 1 L2M-KD: Least-to-Most Prompting Knowledge Distillation

```

1: Input: Training dataset  $D$ , student model  $lm_s$ , teacher model  $lm_t$ 
2: Output: Distilled student model  $lm_s^*$ 
3:  $D' = \{d_1, d_2, \dots, d_n\} \leftarrow \text{sort}(D)$  by Rouge-L score
4:  $D_{train} \leftarrow \emptyset, \alpha \leftarrow 0.1, \beta_1 \leftarrow 0.7, \beta_n \leftarrow 1$ 
5:  $lm_s^* \leftarrow lm_s$ 
6: for  $i \leftarrow 1$  to  $n$  do
7:    $D_{train} \leftarrow D_{train} \cup d_i$ 
8:    $\alpha_i \leftarrow \alpha \cdot \frac{i-1}{n-1}$ 
9:    $\beta_i \leftarrow \beta_1 - (\beta_1 - \beta_n) \cdot \frac{i-1}{n-1}$ 
10:  while not converged for  $p$  epochs do
11:     $lm_s^* \leftarrow \text{train}(lm_s^*, lm_t, D_{train}, \alpha_i, \beta_i)$  using Eq. 6
12:  end while
13: end for
14: return  $lm_s^*$ 

```

3.2.1 CL Framework

The CL framework ranks training samples by difficulty to create a sequence of subsets ranging from easy to hard. Difficulty is measured using the Rouge-L score (Lin, 2004) between the SGOs from a pre-distilled student model and the GTOs from the given training set, supplemented by the cross-entropy loss with respect to the ground-truth response. A higher Rouge-L score indicates an easier sample, as it reflects greater similarity between the student’s and teacher’s outputs.

Given a training dataset D with N samples, we partition D into n subsets $\{d_1, d_2, \dots, d_n\}$, ordered from easiest to hardest based on the Rouge-L scores. Each subset contains approximately $\lfloor N/n \rfloor$

or $\lfloor N/n \rfloor + 1$ samples, with any remainder distributed evenly across the subsets starting from the easiest. Empirically, we find that $n = 4$ subsets provide a robust balance across datasets without requiring dataset-specific tuning.

Training proceeds in n stages, following a *Baby Step* scheduler (Bengio et al., 2009). In stage 1, the easiest subset d_1 is used for training. After a fixed number of epochs or convergence, the next subset d_2 is merged with d_1 , and training continues. This process repeats until all subsets are included, culminating in training on the full dataset D . The progressive inclusion of harder samples ensures that the student model builds foundational knowledge before tackling more complex examples, aligning with the least-to-most prompting principle.

3.2.2 Adaptive KD Loss Function

The L2M-KD method employs a novel adaptive KD loss function that dynamically combines KLD and SKL to modulate the degree of teacher guidance. The standard KLD is defined as:

$$D_{KLD}(p \parallel q_\theta) = \mathbb{E}_{y \sim p} \left[\log \frac{p(y|x, y_{<i})}{q_\theta(y|x, y_{<i})} \right], \quad (4)$$

where p is the teacher’s output distribution, q_θ is the student’s output distribution, and y is sampled from p . The SKL, which introduces smoothing, is expressed as:

$$D_{SKL}(p \parallel q_\theta) = D_{KLD}(p \parallel \alpha \cdot p + (1 - \alpha) \cdot q_\theta), \quad (5)$$

where $\alpha \in [0, 1]$ controls the mixing ratio of the teacher and student distributions (Lee, 2001).

In L2M-KD, we introduce an adaptive parameter α_i that varies across training stages to simulate least-to-most prompting. The adaptive KD loss is defined as:

$$D_{L2M-KD}(p \parallel q_\theta) = D_{KLD}(p \parallel \alpha_i \cdot p + (1 - \alpha_i) \cdot q_\theta), \quad (6)$$

where $\alpha_i = \alpha \cdot \frac{i-1}{n-1}$, and $i \in [1, n]$ denotes the current training stage. We set $\alpha = 0.1$ based on empirical evaluation, ensuring that α_i ranges from 0 (equivalent to KLD) in the first stage to α (approximating SKL) in the final stage. This progression mimics the increasing guidance in least-to-most prompting, starting with minimal teacher influence and gradually incorporating more.

To further optimize training, we introduce an adaptive KD ratio β_i to balance the KD loss and the cross-entropy loss in the total objective:

$$\beta_i = \beta_1 - (\beta_1 - \beta_n) \cdot \frac{i-1}{n-1}, \quad (7)$$

where $\beta_1 = 0.7$ and $\beta_n = 1$ are empirically determined. This ensures that early stages prioritize learning from ground-truth outputs, while later stages emphasize teacher knowledge, aligning with the curriculum’s progression.

4 Experiments

4.1 Experimental Setup

We evaluate the L2M-KD method on instruction-following tasks (Ouyang et al., 2022), where models generate task-compliant responses from given instructions. The pipeline involves fine-tuning a large language model (LLM) as the teacher on instruction-response pairs, followed by knowledge distillation (KD) into smaller student models and performance comparison (Gu et al., 2023).

Base Models. We use two LLM families: GPT-2 (Radford et al., 2019) (teacher: 1.5B; students: 120M, 340M, 760M) and OPT (Zhang et al., 2022) (teacher: 2.7B; students: 125M, 350M, 1.3B). These models are selected for their open availability and representativeness. Details are in Appendix 9.1.

Training. We employ the databricks-dolly-15k dataset (Conover et al., 2023), comprising 15K human-written instruction-response pairs. After filtering samples exceeding context length limits, we split the dataset into 11.5K training, 1K validation, and 0.5K test samples. Training samples are ranked by difficulty using Rouge-L scores (Lin, 2004) and partitioned into $n = 4$ subsets. All models are trained for equal total iterations, with on-policy KD using 50% student-generated outputs (SGOs) and ground-truth responses, while off-policy methods apply adaptive balancing with $\alpha_0 = 0.3$ and $\alpha_n = 0$. Hyperparameters are tuned using validation Rouge-L scores. Full training details are in Appendix 9.2.

Evaluation. We assess distilled models on five instruction-following datasets: DollyEval (Conover et al., 2023), SelfInst (Wang et al., 2022a), VicunaEval (Chiang et al., 2023), S-NI (Wang et al., 2022b), and UnNI (Honovich et al., 2022). Performance is measured using Rouge-L scores, averaged over five generations per prompt with five random seeds, at a temperature of 1.0. Evaluation details are in Appendix 9.3.

Baselines. We compare L2M-KD against four baselines: (1) SFT (supervised fine-tuning on ground-truth responses), (2) SeqKD (Lin et al.,

2020) (uses teacher-generated outputs), (3) KLD (Hinton et al., 2015) (Kullback-Leibler divergence loss), and (4) SKL (Ko et al., 2024) (skew KLD loss), with an on-policy SKL variant using 50% SGOs. Baseline details are in Appendix 9.4.

4.2 Results

As shown in Table 1, L2M-KD consistently outperforms baseline white-box KD methods across both GPT-2 and OPT teacher-student setups. For GPT-2 (teacher: 1.5B; students: 120M, 340M, 760M), L2M-KD achieves average Rouge-L scores of 21.7, 23.3, and 24.4, respectively, surpassing KLD by up to 4.0 points and SKL (on-policy) by up to 1.6 points, while exceeding the teacher on multiple datasets, notably VicunaEval and S-NI. Similarly, for OPT (teacher: 2.7B; students: 125M, 350M, 1.3B), L2M-KD yields average scores of 20.3, 22.3, and 25.6, outperforming KLD by up to 3.7 points and SKL (on-policy) by up to 0.5 points, with significant gains over the teacher on S-NI and UnNI, particularly with the 1.3B student where L2M-KD achieves 30.1 and 35.8, respectively, compared to the teacher’s 19.2 and 22.7.

L2M-KD’s superior performance across both model families underscores the effectiveness of its difficulty-aware curriculum learning strategy and adaptive KD loss function, as introduced in Section 3. By ranking training samples using Rouge-L scores and transitioning from KLD to SKL, L2M-KD mitigates mode-averaging and over-smoothing, aligning teacher guidance with the student’s learning progression. Notably, L2M-KD achieves these gains using GTOs exclusively, avoiding the computational overhead of on-policy methods like SKL (on-policy), which rely on student-generated outputs. The consistent outperformance of the teacher model, particularly with larger student models (e.g., GPT-2-760M, OPT-1.3B), highlights L2M-KD’s ability to maximize knowledge transfer efficiency. These results validate our hypothesis that a staged KD process with adaptive teacher guidance can significantly enhance student model performance while maintaining computational efficiency, as posited in Section 1.

5 Analysis and Discussion

5.1 Ablation analysis

To elucidate the contributions of L2M-KD’s core components, we conduct three ablation studies using the GPT-2 (1.5B teacher, 0.1B student) and

Table 1: Evaluation of the L2M-KD. Rouge-L scores on several benchmarks.

Model	# Params	KD Methods	DollyEval	SelfInst	VicunaEval	S-NI	UnNI	Avg.
GPT-2	1.5B	Teacher	27.6	14.3	16.3	27.6	31.8	23.5
		SFT	23.3	10.0	14.7	16.3	18.5	16.6
	120M	SeqKD	22.7	10.1	14.3	16.4	18.8	16.5
		KLD	22.8	10.8	13.4	19.7	22.0	17.7
		SKL	24.2	12.3	15.7	24.3	24.0	20.1
		SKL (on-policy)	24.0	12.2	16.9*	27.0	27.3	21.5
		L2M-KD	25.0	12.3	16.5*	27.2	27.4	21.7
	340M	SFT	25.5	13.0	16.0	25.1	32.0	22.3
		SeqKD	25.3	12.6	16.9*	22.9	30.2	21.6
		KLD	25.0	12.0	15.4	23.7	31.0	21.4
		SKL	26.4	15.9*	16.4*	27.2	29.8	23.1
		SKL (on-policy)	26.7	15.3*	17.6*	26.3	30.2	23.2
		L2M-KD	26.5	16.0*	17.4*	26.7	30.2	23.3
	760M	SFT	25.4	12.4	16.1	21.5	27.1	20.5
		SeqKD	25.6	14.0	15.9	26.1	32.9*	22.9
		KLD	25.9	13.4	16.9*	25.3	31.7	22.6
		SKL	26.8	15.2*	16.4*	28.1*	32.0*	23.7*
		SKL (on-policy)	27.1	15.7*	16.9*	28.9*	32.1*	24.1*
L2M-KD		27.8*	15.6*	17.1*	29.1*	32.3*	24.4*	
OPT	2.7B	Teacher	26.2	11.2	15.5	19.2	22.7	18.9
		SFT	21.8	8.1	14.4	13.5	15.1	14.6
	125M	SeqKD	20.7	8.9	13.6	16.8	18.7	15.7
		KLD	20.5	9.2	14.7	15.8	18.2	15.7
		SKL	24.0	11.5*	15.6*	23.0*	24.5*	19.7*
		SKL (on-policy)	24.3	11.7*	15.9*	23.5*	24.6*	20.0*
		L2M-KD	24.7	12.0*	16.3*	23.4*	25.1*	20.3*
	350M	SFT	22.6	11.1	15.1	19.3*	21.7	18.0
		SeqKD	24.3	10.7	15.5	19.9*	22.6	18.6
		KLD	24.0	12.0*	16.1*	22.5*	25.4*	20.0*
		SKL	25.0	12.7*	16.3*	26.2*	28.7*	21.8*
		SKL (on-policy)	25.3	13.1*	16.9*	26.4*	28.9*	22.1*
		L2M-KD	25.7	13.4*	17.4*	26.2*	28.8*	22.3*
	1.3B	SFT	25.0	13.1*	15.5	25.0*	27.2*	21.1*
		SeqKD	26.3*	13.2*	16.7*	24.6*	27.8*	21.7*
		KLD	25.4	13.0*	16.2*	25.3*	29.4*	21.9*
		SKL	28.0*	16.5*	17.4*	29.8*	35.0*	25.3*
		SKL (on-policy)	28.3*	16.6*	17.7*	30.1*	35.5*	25.6*
L2M-KD		28.4*	16.5*	17.5*	30.9*	35.8*	25.8*	

Note: All KD methods not otherwise indicated are off-policy. Results represent the mean performance across five random seeds. The highest scores among student models are **bolded**, while instances where the student model outperforms the teacher are marked with a superscript *. **Avg.** refers to the average ROUGE-L score across the five evaluation datasets.

OPT (2.7B teacher, 0.1B student) model families across five evaluation datasets. These studies examine the impact of the adaptive KD ratio β_i , the least-to-most prompting strategy, and the control parameter α , respectively, with performance measured via Rouge-L scores.

Impact of Adaptive KD Ratio. We evaluate the adaptive KD ratio β_i (Eq. 7), which balances cross-entropy and KD losses across stages. Table 2 shows that L2M-KD with β_i consistently outperforms both the variant without β_i and the SKL baseline across all datasets, particularly on VicunaEval and UnNI. This highlights β_i 's role in dynamically adjusting teacher guidance, optimiz-

ing the student's learning from both ground-truth and teacher knowledge.

Table 2: Performance comparison of L2M-KD with and without adaptive KD ratio on the five evaluation datasets.

Model # Params	KD Methods	Evaluation Dataset				
		DE	SI	VE	S-NI	UnNI
GPT-2 (1.5B/0.1B)	SKL	24.2	12.3	15.7	24.3	24.0
	L2M-KD (w/o β)	24.6	12.3	15.9	26.6	27.0
	L2M-KD	25.0	12.3	16.5	27.2	27.4
OPT (2.7B/0.1B)	SKL	24.0	11.5	15.6	23.0	24.5
	L2M-KD (w/o β)	24.5	11.9	15.8	23.3	24.7
	L2M-KD	24.7	12.0	16.3	23.4	25.1

Note: 'w/o β ' indicates that the adaptive KD ratio (β_i , see Eq. 7) were not used in the L2M-KD methods. 'DE', 'SI', 'VE', 'S-NI' and 'UnNI' indicates five evaluation datasets - DollyEval, SelfInst, VicunaEval, S-NI, and UnNI, respectively.

Impact of Prompting Strategy. We compare the least-to-most (L2M) prompting strategy (KLD to SKL) against a most-to-least (M2L) variant. As shown in Table 3, L2M-KD (L2M) surpasses both L2M-KD (M2L) and SKL, with notable gains on S-NI and UnNI for GPT-2, validating that progressively increasing teacher guidance aligns with the curriculum learning framework (Section 3.2) and enhances knowledge transfer.

Table 3: Results showing the impact of different prompting strategies on the performance of the distilled student model in L2M-KD.

Model # Params	KD Methods	Evaluation Dataset				
		DE	SI	VE	S-NI	UnNI
GPT-2 (1.5B/0.1B)	SKL	24.2	12.3	15.7	24.3	24.0
	L2M-KD (M2L)	24.8	12.3	16.0	24.8	25.2
	L2M-KD (L2M)	25.0	12.3	16.5	27.2	27.4
OPT (2.7B/0.1B)	SKL	24.0	11.5	15.6	23.0	24.5
	L2M-KD (M2L)	24.6	11.5	15.9	23.3	25.1
	L2M-KD (L2M)	24.7	12.0	16.3	23.4	25.1

Note: ‘M2L’ and ‘L2M’ denote two variants of L2M-KD, where ‘L2M’ employs a least-to-most prompting strategy (progressing from KLD to SKL), and ‘M2L’ uses a most-to-least prompting strategy (reversing from SKL to KLD). ‘DE’, ‘SI’, ‘VE’, ‘S-NI’, and ‘UnNI’ refer to the five evaluation datasets: DollyEval, SelfInst, VicunaEval, S-NI, and UnNI, respectively.

Impact of Control Parameter α . We analyze the sensitivity to α , which governs the KLD-to-SKL transition in the adaptive KD loss (Eq. 6). Table 4 indicates that $\alpha = 0.15$ yields optimal performance for both GPT-2 and OPT, while higher values (e.g., $\alpha = 0.90$) degrade results due to over-smoothing, underscoring the need for a balanced α to maintain effective teacher guidance without excessive blending of the student’s distribution.

Table 4: Results showing the impact of different control parameter α on the performance of the distilled student model in L2M-KD.

Model # Params	α	Evaluation Dataset				
		DE	SI	VE	S-NI	UnNI
GPT-2 (1.5B/0.1B)	0.15	25.0	12.3	16.5	27.2	27.4
	0.30	25.1	12.1	15.7	27.7	26.9
	0.45	24.7	12.0	16.3	23.4	25.1
	0.60	24.5	12.4	14.7	21.6	27.7
	0.75	24.1	11.0	14.5	23.6	27.6
	0.90	23.9	12.1	16.2	19.9	24.2
OPT (2.7B/0.1B)	0.15	24.7	12.0	16.3	23.4	25.1
	0.30	24.8	11.0	15.3	22.7	21.7
	0.45	23.2	11.2	14.9	20.1	19.9
	0.60	23.2	9.3	14.7	18.9	21.2
	0.75	23.0	11.7	15.9	19.6	19.0
	0.90	21.8	11.0	15.6	21.1	19.8

Note: ‘DE’, ‘SI’, ‘VE’, ‘S-NI’, and ‘UnNI’ refer to the five evaluation datasets: DollyEval, SelfInst, VicunaEval, S-NI, and UnNI, respectively.

5.2 Compare with other curriculum-based KD methods

We compared L2M-KD with DistiLLM-2 (Ko et al., 2025) (off-and on-policy) using GPT-2 and OPT

models across five datasets (See Table 5). L2M-KD outperforms off-policy DistiLLM-2, while on-policy DistiLLM-2 slightly leads, though L2M-KD excels on DollyEval and S-NI.

Table 5: Results comparing our approach with other curriculum-based methods.

Model	KD Methods	DE	SI	VE	S-NI	UnNI
GPT-2 (1.5B/0.1B)	L2M-KD	25.0	12.3	16.5	27.2	27.4
	Distillm-2 (off)	24.1	12.1	16.3	25.2	25.1
	Distillm-2 (on)	24.9	12.5	16.9	27.6	27.7
OPT (2.7B/0.1B)	L2M-KD	24.7	12.0	16.3	23.4	25.1
	Distillm-2 (off)	24.1	11.5	15.6	23.1	24.7
	Distillm-2 (on)	25.1	12.2	16.5	23.3	25.5

5.3 Effectiveness of L2M-KD on larger models and other metrics

We extended our experiments to include OpenL-LaMA2 (7B teacher, 2B student). We compared L2M-KD against the same baselines (SFT, SeqKD, KLD, SKL off-policy, and SKL on-policy) across the five instruction-following datasets, measuring performance with both Rouge-L and winning rates (WR) using pairwise comparison (Zheng et al., 2023). The baseline is OpenLLaMA2 (7B) fine-tuned on databricks-dolly-15k, with DeepSeek-V3-0324 (Liu et al., 2024) as the judge for WR. The results (Table 6) show that L2M-KD slightly outperforms baselines in both Rouge-L and WR, consistent with our findings on smaller models. This confirms L2M-KD’s effectiveness for larger-size LLMs.

5.4 Discussion

The ablation studies collectively highlight the synergistic contributions of L2M-KD’s components. The adaptive KD ratio β_i enhances performance by balancing learning objectives, the least-to-most prompting strategy ensures effective knowledge transfer by aligning with the curriculum learning progression, and an optimal α value (0.15) maximizes the benefits of the adaptive KD loss. These findings reinforce the design choices in L2M-KD (Section 3), demonstrating its robustness and efficacy in improving student model performance while maintaining computational efficiency through the exclusive use of GTOs.

6 Related Work

White-box KD for LLMs leverages teacher model internals, outperforming black-box KD (Yang et al., 2024b). Methods focus on KD loss functions like

Table 6: Effectiveness of L2M-KD on Larger Models and Diverse Metrics

Model	Methods	DollyEval		SelfInst		VicunaEval		SNI		UnNI	
		R-L	WR(%)	R-L	WR(%)	R-L	WR(%)	R-L	WR(%)	R-L	WR(%)
OpenLLaMA2 (7B/2B)	SFT	25.1	48.60	16.2	46.44	16.3	48.60	29.3	48.53	29.1	48.38
	SeqKD	24.7	47.24	15.8	45.70	17.1	49.70	29.1	49.15	28.6	48.14
	KLD	21.0	44.15	16.1	46.84	15.4	47.09	27.9	47.91	25.2	44.88
	SKL	27.9	51.37	19.4	51.05	17.3	51.00	33.9	52.46	32.8	51.57
	SKL (on-policy)	28.5	51.74	19.2	50.83	18.4	52.17	33.7	52.43	33.1	51.86
	L2M-KD	28.7	52.29	19.2	51.14	18.2	51.92	34.2	53.02	33.5	52.45

KLD (Hinton et al., 2015), reverse KLD (Gu et al., 2023), JSD (Agarwal et al., 2024), and skew KLD (Ko et al., 2024), alongside data strategies using TGOs (Kim and Rush, 2016), SGOs (Agarwal et al., 2024), or mixtures (Ko et al., 2024) to address mode-averaging and training-inference mismatches (Section 2.1). However, optimal combinations remain challenging. Curriculum learning (CL) in KD, primarily in computer vision (Xiang et al., 2020; Li et al., 2023), is underexplored for LLMs. Confucius (Gao et al., 2024) applies CL to black-box KD for tool learning, but general white-box KD lacks such frameworks, which L2M-KD addresses (Section 3.2).

7 Conclusion

In this work, we introduce L2M-KD, a novel white-box KD method designed to address critical challenges in compressing LLMs for auto-regressive tasks. By integrating a CL strategy with an adaptive KD loss function, L2M-KD optimizes knowledge transfer from a high-capacity teacher to a resource-efficient student model. The method comprises two core components: (1) a CL strategy that ranks training samples by difficulty using Rouge-L scores, enabling a staged training process from easy to hard subsets inspired by least-to-most prompting, and (2) an adaptive KD loss that transitions from KLD to SKL, dynamically adjusting teacher guidance to mitigate mode-averaging and over-smoothing. Extensive evaluations on instruction-following tasks demonstrate that L2M-KD achieves superior student model performance compared to existing white-box KD methods, while exclusively utilizing GTOs to minimize computational costs. These results highlight the effectiveness of difficulty-aware training and adaptive loss design in enhancing KD efficiency and robustness. Our work provides a scalable and practical approach to LLM compression, paving the way for broader deployment in resource-constrained environments. Future research will explore the generalizability of L2M-KD across diverse model architectures and

tasks, as well as its efficacy for larger student models.

8 Limitations

While L2M-KD demonstrates significant advancements in white-box KD for LLMs, our study is subject to certain limitations due to computational constraints. First, our experiments were conducted using two model families, GPT-2 and OPT, focusing exclusively on instruction-following tasks. This scope limits the assessment of L2M-KD’s generalizability across other model architectures, such as Qwen 2.5 (Yang et al., 2024a) or Llama 3.2 (Meta, 2024), and diverse natural language processing tasks, including text generation (Li et al., 2024b) and summarization (Zhang et al., 2019). Further investigation is needed to validate the method’s effectiveness in these contexts. Second, the student models evaluated in our experiments have relatively small parameter sizes. Larger student models, with greater inherent capacity, may exhibit reduced variability in perceived sample difficulty when ranked using Rouge-L scores, potentially diminishing the effectiveness of the CL strategy. This could impact the staged training process central to L2M-KD. Future work will address these limitations by evaluating L2M-KD on a broader range of model architectures, tasks, and larger student models to ensure its robustness and scalability across diverse settings.

References

- Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*.
- Anthropic. 2024. Claude 3.5 Sonnet. <https://www.anthropic.com/news/claude-3-5-sonnet>. Published: 21 Jun 2024.
- Abi Aryan, Aakash Kumar Nain, Andrew McMahon, Lucas Augusto Meyer, and Harpreet Singh Sahota.

2023. The costly dilemma: generalization, evaluation and cost-optimal deployment of large language models. *arXiv preprint arXiv:2308.08061*.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.
- Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023), 2(3):6.
- Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. [Free dolly: Introducing the world’s first truly open instruction-tuned llm](#).
- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). Preprint, arXiv:2501.12948.
- Shen Gao, Zhengliang Shi, Minghang Zhu, Bowen Fang, Xin Xin, Pengjie Ren, Zhumin Chen, Jun Ma, and Zhaochun Ren. 2024. Confucius: Iterative tool learning from introspection feedback by easy-to-difficult curriculum. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18030–18038.
- Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Minillm: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2022. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1317–1327.
- Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. 2025. Distillm-2: A contrastive approach boosts the distillation of llms. *arXiv preprint arXiv:2503.07067*.
- Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*.
- Lillian Lee. 2001. On the effectiveness of the skew divergence for statistical language analysis. In *International workshop on artificial intelligence and statistics*, pages 176–183. PMLR.
- Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. 2024a. Revisiting catastrophic forgetting in large language model tuning. *arXiv preprint arXiv:2406.04836*.
- Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024b. Pre-trained language models for text generation: A survey. *ACM Computing Surveys*, 56(9):1–39.
- Zheng Li, Xiang Li, Lingfeng Yang, Borui Zhao, Renjie Song, Lei Luo, Jun Li, and Jian Yang. 2023. Curriculum temperature for knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1504–1512.
- Myrna E Libby, Julie S Weiss, Stacie Bancroft, and William H Ahearn. 2008. A comparison of most-to-least and least-to-most prompting on the acquisition of solitary play skills. *Behavior analysis in practice*, 1:37–43.
- Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. 2020. Autoregressive knowledge distillation through imitation learning. *arXiv preprint arXiv:2009.07253*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Nestor Maslej, Loredana Fattorini, Raymond Perrault, Yolanda Gil, Vanessa Parli, Njenga Kariuki, Emily Capstick, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, Tobi Walsh, Armin Hamrah, Lapo Santarasci, Julia Betts Lotufo, Alexandra Rome, Andrew Shi, and Sukrut Oak. 2025. The ai index 2025 annual report. Technical report, Stanford University, Institute for Human-Centered Artificial Intelligence (HAI), Stanford, CA. AI Index Steering Committee.
- Meta. 2024. [Llama 3.2: Revolutionizing edge ai and vision with open, customizable models](#). Accessed: 2025-04-21 (current date from system).
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al.

2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Neal Parikh, Stephen Boyd, et al. 2014. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Makoto Shing, Kou Misaki, Han Bao, Sho Yokoi, and Takuya Akiba. 2025. Taid: Temporally adaptive interpolated distillation for efficient knowledge transfer in language models. *arXiv preprint arXiv:2501.16937*.
- Xin Wang, Yudong Chen, and Wenwu Zhu. 2021. A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4555–4576.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022a. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.
- Yizhong Wang, Swaroop Mishra, Pegah Alipoor-molabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022b. Benchmarking generalization via in-context instructions on 1,600+ language tasks. *arXiv preprint arXiv:2204.07705*, 2:2.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- Liuyu Xiang, Guiguang Ding, and Jungong Han. 2020. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 247–263. Springer.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024a. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.
- Chuanpeng Yang, Yao Zhu, Wang Lu, Yidong Wang, Qian Chen, Chenlong Gao, Bingjie Yan, and Yiqiang Chen. 2024b. Survey on knowledge distillation for large language models: methods, evaluation, and application. *ACM Transactions on Intelligent Systems and Technology*.
- Haoyu Zhang, Jianjun Xu, and Ji Wang. 2019. Pretraining-based natural language generation for text summarization. *arXiv preprint arXiv:1902.09243*.
- Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. 2024. Dual-space knowledge distillation for large language models. *arXiv preprint arXiv:2406.17328*.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

9 Technical Appendices

This appendix provides detailed specifications to ensure reproducibility of our experiments. It covers the base models (Section 9.1), training procedures (Section 9.2), evaluation protocols (Section 9.3), and baseline methods (Section 9.4).

9.1 Base Models

We utilize two prominent families of decoder-only transformer-based large language models (LLMs): GPT-2 (Radford et al., 2019) and OPT (Zhang et al., 2022). These models are selected for their open-source availability, widespread adoption in research, and architectural representativeness of modern LLMs, facilitating reproducibility and fair comparison.

- **GPT-2** (Radford et al., 2019): Developed by OpenAI, GPT-2 is a decoder-only transformer model optimized for autoregressive language modeling. We use a 1.5B-parameter GPT-2 model as the teacher, which has 48 layers, 25 attention heads, and a hidden size of 1600. The student models are smaller GPT-2 variants with 120M (12 layers, 12 heads, hidden size 768), 340M (24 layers, 16 heads, hidden size 1024), and 760M (36 layers, 20 heads, hidden size 1280) parameters, enabling evaluation across a range of model scales.
- **OPT** (Zhang et al., 2022): Developed by Meta AI, OPT is a family of open-source LLMs designed for research, also following a decoder-only transformer architecture. The teacher model is OPT-2.7B, with 32 layers, 32 attention heads, and a hidden size of 2560. The student models are OPT-125M (12 layers, 12

heads, hidden size 768), OPT-350M (24 layers, 16 heads, hidden size 1024), and OPT-1.3B (24 layers, 32 heads, hidden size 2048). These configurations allow us to assess the scalability of L2M-KD across different parameter sizes.

All models are sourced from the Hugging Face Transformers library (Wolf et al., 2019), ensuring standardized implementations. Pre-trained weights are used as initialization for both teacher and student models, with teacher models fine-tuned on the training dataset prior to distillation.

9.2 Training Details

All experiments are conducted on a cluster equipped with four NVIDIA A800 80GB GPUs and an Intel(R) Xeon(R) Platinum 8350C CPU, using PyTorch 2.1.0 and CUDA 12.1 for implementation. We employ the databricks-dolly-15k dataset (Conover et al., 2023), which contains 15,000 human-written instruction-response pairs across diverse categories, including brainstorming, classification, question answering, and summarization. After filtering samples exceeding the maximum context length of 1024 tokens (to ensure compatibility with model constraints), the dataset is split into 11,500 training, 1,000 validation, and 500 test samples using a fixed random seed of 42 for reproducibility.

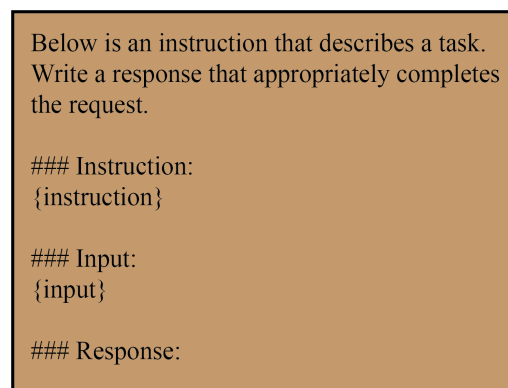
For the L2M-KD method, training samples are ranked by difficulty using Rouge-L scores (Lin, 2004) between student-generated outputs (SGOs) from a pre-distilled student model and GTOs, supplemented by cross-entropy loss with respect to the ground truth. The 11,500 training samples are partitioned into $n = 4$ difficulty-based subsets, each containing approximately 2,875 samples, with any remainder distributed starting from the easiest subset. Training proceeds in four stages, following the Baby Step scheduler (Bengio et al., 2009): each stage adds the next difficulty subset, using the previous stage’s checkpoint for initialization, and trains until convergence or a fixed number of epochs (5 per stage). The adaptive KD ratio β_i (Eq. 7) decreases linearly from $\beta_1 = 0.7$ to $\beta_n = 1$, while the control parameter α_i (Eq. 6) increases from 0 to 0.15.

Hyperparameters are tuned on the validation set using Rouge-L scores, which correlate strongly with human judgments (Agarwal et al., 2024). We explore learning rates ($\{5e-4, 1e-4, 5e-5\}$) and

batch sizes ($\{8, 16\}$), selecting $1e-4$ and 16, respectively, for optimal performance. All models (baselines and L2M-KD) are trained for an equivalent total number of iterations: baselines train for 20 epochs, while L2M-KD trains for 8 epochs across its four stages (2 epochs per stage), ensuring fairness. On-policy KD methods use a mixture of 50% SGOs and 50% GTOs, following Agarwal et al. (2024), while off-policy methods, including L2M-KD, use GTOs exclusively with adaptive balancing ($\alpha_0 = 0.3, \alpha_n = 0$). The AdamW optimizer (Parikh et al., 2014) is used with a weight decay of 0.01 and a linear learning rate scheduler with 10% warmup steps.

9.3 Evaluation Details

Evaluations are conducted on a single NVIDIA A800 80GB GPU, following the protocol of Gu et al. (2023). During inference, responses are generated with a temperature of 1.0, a maximum sequence length of 512 tokens, and top-k sampling ($k=50$). To ensure robustness, we generate five responses per prompt using random seeds $\{10, 20, 30, 40, 50\}$ and report the average Rouge-L score across these generations. A standardized prompt template (Fig. 4) is used for consistency across all models and datasets.



```

Below is an instruction that describes a task.
Write a response that appropriately completes
the request.

### Instruction:
{instruction}

### Input:
{input}

### Response:

```

Figure 4: Prompt template for instruction-following experiments, adapted from Gu et al. (2023).

The evaluation datasets are as follows:

- databricks-dolly-15k (Conover et al., 2023): 15,000 human-written instruction-response pairs covering diverse tasks such as brainstorming, classification, question answering, and summarization. We use the 500-sample test split for evaluation.
- self-instruct-eval (Wang et al., 2022a): Contains 252 expert-written tasks and 50,000

public examples for evaluation, designed to test instruction-following and generalization.

- vicuna-eval (Chiang et al., 2023): Comprises 80 challenging questions to assess complex reasoning and instruction-following capabilities.
- supernatural-instructions (Wang et al., 2022b): Includes 9,000 test samples from 119 NLP tasks, spanning 76 task types, with expert-written instructions.
- unnatural-instructions-core (Honovich et al., 2022): A 66,000-sample subset of machine-generated instructions, demonstrating the efficacy of synthetic data for training.

Performance is evaluated using Rouge-L scores (Lin, 2004), a metric widely adopted for its correlation with human judgment in instruction-following tasks (Agarwal et al., 2024).

9.4 Baseline Methods

We detail the KD loss functions used in our baselines, as defined in the KD loss method (Eq. 2), which measures the divergence between the teacher distribution p and the student distribution q_θ using a divergence function $D(\cdot||\cdot)$. The formulations are as follows:

- **KLD** (Hinton et al., 2015): The Kullback-Leibler Divergence is a standard measure in KD, defined as:

$$D_{\text{KLD}}(p||q_\theta) = \mathbb{E}_{y \sim p} \left[\log \frac{p(y)}{q_\theta(y)} \right], \quad (8)$$

where y is sampled from the teacher distribution p . KLD encourages the student to match the teacher’s output distribution, often leading to mode-averaging (Gu et al., 2023).

- **SKL** (Ko et al., 2024): The Skew KLD introduces a smoothing mechanism to mitigate over-smoothing issues in KLD, defined as:

$$D_{\text{SKL}}(p||q_\theta) = D_{\text{KLD}}(p||\beta \cdot p + (1 - \beta) \cdot q_\theta), \quad (9)$$

where $\beta \in [0, 1]$ controls the mixing ratio between teacher and student distributions. In our experiments, $\beta = 0.5$ for SKL and SKL (on-policy), balancing the influence of both distributions.

For on-policy KD (e.g., SKL (on-policy)), we use a mixture of 50% SGOs and 50% GTOs, as recommended by Agarwal et al. (2024), to mitigate training-inference mismatches. Off-policy methods (SFT, SeqKD, KLD, and SKL) rely solely on ground-truth outputs or teacher-generated outputs.