SWITCH: Studying with Teacher for Knowledge Distillation of Large Language Models

Anonymous ACL submission

Abstract

Despite the success of Large Language Models (LLMs), they still face challenges related to high inference costs and memory requirements. To address these issues, Knowledge Distillation (KD) has emerged as a popular method for model compression, with student-generated outputs (SGOs) being particularly notable for reducing the mismatch between training and inference. However, SGOs often produce noisy and biased sequences, which can lead to misguidance from the teacher model, especially in long sequences. To mitigate these challenges, we propose **SWITCH** (**S**tudying **WI**th **T**ea**CH**er for Knowledge Distillation), a novel approach that strategically incorporates the teacher model during the student's sequence generation. SWITCH identifies discrepancies between the token probabilities of the teacher and student models, allowing the teacher to intervene selectively, particularly in long sequences that are more prone to teacher misguidance. Extensive experimental results across three model families and five instruction-following datasets show that SWITCH surpasses traditional KD methods, particularly excelling in the generation of long sequential data.

1 Introduction

Despite the strong performance of large language models (LLMs, Taori et al., 2023; OpenAI, 2023), their immense scale incurs high resource demands, leading to efforts to compress them while preserving performance (Hsieh et al., 2023; Jiang et al., 2023; Zhong et al., 2024). To address this issue, Knowledge Distillation (KD, Hinton et al., 2015) has emerged as a promising method for compressing LLMs, aiming to transfer knowledge from a large teacher model to a smaller student model without substantial loss in performance. While traditional KD techniques have focused on natural language understanding (Sanh et al., 2019; Mirzadeh et al., 2020), recent research has shifted towards applying KD to natural language generation (Lin et al., 2020; Wen et al., 2023). In this vein, methods utilizing Student-Generated Outputs (SGOs)—where the student model's own outputs are used as training sequences—have been proposed to improve text generation performance. These approaches help mitigate the training-inference mismatch (Bengio et al., 2015), leading to notable performance improvements (Agarwal et al., 2024; Gu et al., 2024).

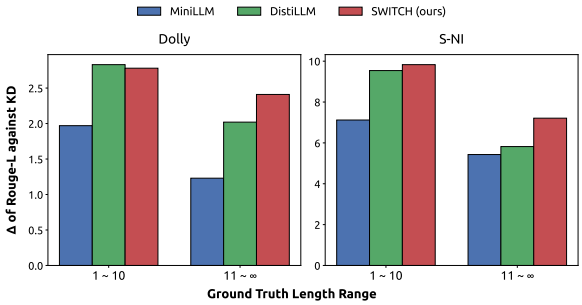

Figure 1: Rouge-L scores of the distilled models against KD. Dataset split by ground truth length.

However, SGO methods primarily emphasize efficient training of the student model while often overlooking the necessity of precise guidance from the teacher model. This approach accounts for the training-inference mismatch from the student's perspective but neglects the analogous mismatch from the teacher's perspective, potentially resulting in misguidance during training (Ko et al., 2024). This misguidance is due to the inherent capacity gap between the teacher and student models, and the problem becomes more pronounced as the gap widens. Furthermore, such misguidance is a significant concern, as the KD method is based on the assumption that the teacher offers reliable guidance. Specifically, it causes the student to receive high loss penalties for correct predictions and low penalties for incorrect ones, resulting in inaccurate knowledge transfer.

Moreover, the autoregressive nature of language models exacerbates this issue, as biases from the student model can accumulate over long sequences (Arora et al., 2022). This problem manifests in instruction-following tasks, which are a general form of sequence generation (Ouyang et al., 2022). We observe that existing KD methods utilizing SGOs struggle with long sequential data, as shown in Figure 1, underscoring the need for more sophisticated strategies when applying SGOs in KD frameworks. Therefore, excessive reliance on outputs generated solely by the student model may lead to misguidance from the teacher.

To address this issue, this paper proposes an innovative approach, termed **SWITCH** (**S**tudying **WI**th **T**ea**CH**er for knowledge distillation), which strategically incorporates the teacher model into the generation of student sequences to ensure accurate guidance from the teacher. By detecting discrepancies in the probability distributions of the next token between the teacher and the student, our method selectively generates tokens with the teacher model instead of the student model. This teacher intervention is particularly effective for long sequences, where the risk of cumulative errors and the following misguidance becomes more significant. To manage this, SWITCH leverages an exponentially decaying threshold that increases the teacher's involvement as the sequence progresses, preventing misguidance in long outputs.

Through comprehensive experiments, we demonstrate that the SWITCH method outperforms existing baseline approaches. This performance improvement has been verified across five instruction-following benchmarks and three model families, and it remains effective across different student model sizes. Notably, the performance gains achieved by SWITCH increase as the size difference between the student and teacher models grows, suggesting that our method effectively mitigates the misguidance issues caused by excessive reliance on student-generated outputs. Furthermore, we observe significant performance improvements in generating long sequences. This highlights the critical role of our method in applying KD for long sequences.

To summarize, our contributions are three-fold:

- We propose SWITCH, a novel approach that utilizes selective intervention of the teacher model to mitigate misguidance from student-generated outputs.
- We demonstrate that SWITCH achieves state-of-the-art performance across various benchmarks and model sizes.
- Our method particularly excels when there is a substantial size difference between the student and teacher models and when handling long sequences.

2 Methodology

In this section, we introduce **SWITCH**, a novel approach designed to enhance the knowledge distillation process for language models. We begin by formalizing the problem of sequence-level knowledge distillation and then delve into the specifics of SWITCH, explaining how it addresses the limitations of traditional student-generated output (SGO) methods. Our methodology emphasizes the strategic involvement of the teacher model during sequence generation, aiming to reduce the accumulation of errors and improve the overall performance of the student model.

2.1 Preliminaries

In knowledge distillation for language models, a smaller student model q learns to emulate a larger teacher model p. Given a prompt x and ground-truth sequence $y = (y_1, y_2, \ldots, y_T)$, the training objective for the student model is to minimize the divergence D of token-level distribution between the student and teacher.

$$\mathcal{D}(p \parallel q)(y|x) = \sum_{t=1}^{|y|} \sum_{y_t \in V} \mathcal{D}(p \parallel q)(y_t|y_{<t}, x) \tag{1}$$

where $y_{<t}$ denotes the sequence of tokens generated up to time $t-1$, and V is the vocabulary.

Meanwhile, in SGO methods, the student model generates sequences based on its own probability distribution:

$$y_t \sim q(\cdot \mid y_{<t}, x) \tag{2}$$

Using student-generated sequence, the student learns by comparing its output distribution to that of the teacher, aiming to minimize the divergence between their respective distributions. The rationale behind the use of SGO is to use its own outputs to reduce training-inference mismatch.

However, since the student model q typically has less capacity than the teacher model p, it tends to

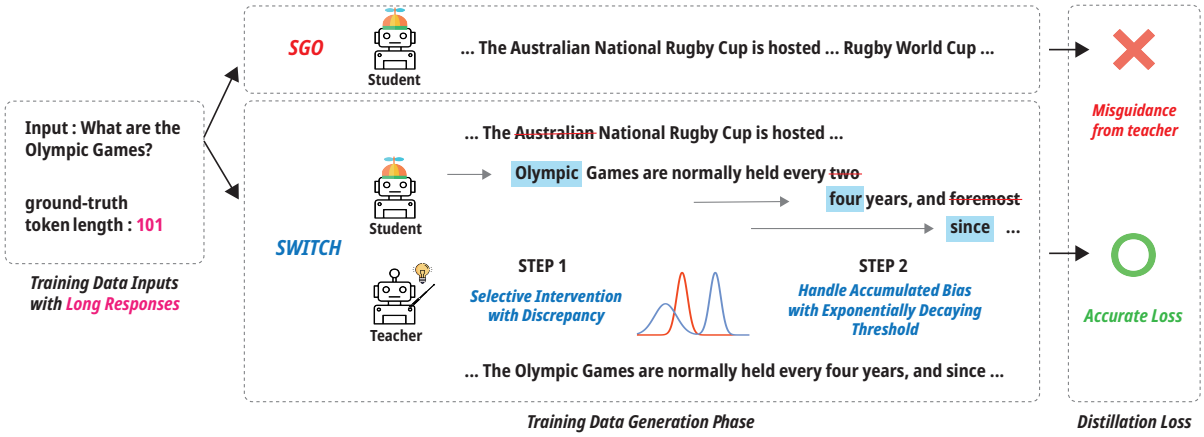

Figure 2: An overview of our SWITCH method. To mitigate misguidance, SWITCH selectively intervenes the generation process of SGO using distribution discrepancy. More intervention is given as the sequence gets longer to balance the benefits of student learning from their own outputs and the mitigation of teacher misguidance.

produce noisier and more biased sequences. Due to the autoregressive nature of sequence generation, these errors can accumulate over time, leading to significant divergence from the teacher's behavior.

2.2 SWITCH

To overcome the limitations of SGO methods, we propose SWITCH, which strategically incorporates the teacher model into the sequence generation process. The key idea is to selectively switch from the student to the teacher model to generate the next token when significant discrepancies between their distributions are detected in order to minimize distribution mismatch from the teacher model's perspective, thereby reducing misguidance. To address accumulated bias in long sequences, SWITCH increases teacher involvement as the sequence progresses. This selective involvement aims to minimize the misguidance from the teacher model caused by accumulated errors from SGO while preserving the benefits for the student to learn from its own outputs. Figure 2 provides an overview.

Measuring Distribution Discrepancy for Selective Token Generation To determine when to involve the teacher model, we need a reliable measure of the discrepancy between the student and teacher distributions for the next token. We employ the Jensen-Shannon divergence (JSD), which provides a symmetric and bounded measure of divergence between two probability distributions.

The JSD between the teacher distribution $p(\cdot \mid x_{<t})$ and the student distribution $q(\cdot \mid x_{<t})$ is defined as:

$$\mathrm{JSD}(p \parallel q) = \tfrac{1}{2} D_{\mathrm{KL}}(p \parallel m) + \tfrac{1}{2} D_{\mathrm{KL}}(q \parallel m) \quad (3)$$

where $m = \frac{1}{2}(p + q)$ is the average distribution, and D_{KL} denotes the Kullback-Leibler divergence (KLD). The JSD ranges between 0 and 1, facilitating a consistent thresholding mechanism.

At each time step t, we compute the JSD between the student and teacher distributions conditioned on the current context. If the divergence exceeds a predefined threshold τ_t, we switch from using the student model to the teacher model for generating the next token:

$$y_t = \begin{cases} \text{Sample from } q(\cdot \mid y_{<t}, x), & \text{if } \mathrm{JSD}_t \leq \tau_t \\ \text{Sample from } p(\cdot \mid y_{<t}, x), & \text{if } \mathrm{JSD}_t > \tau_t \end{cases} \quad (4)$$

This strategy allows the student to produce sequences that are reliable enough for the teacher to interpret and offer accurate guidance, as the teacher only intervenes when there is a significant deviation between the two distributions.

Handling Accumulated Bias with Exponentially Decaying Threshold To address the accumulation of bias in long sequences due to the autoregressive nature of the model, we introduce an exponentially decaying threshold for the JSD:

$$\tau_t = \tau_0 \cdot e^{-\lambda t} \quad (5)$$

where:

- τ_0 is the initial threshold at $t = 0$, set to 1.
- λ is the decay rate controlling how quickly the threshold decreases over time.

This approach encourages the student model to learn from its own outputs at the beginning of the sequence. As the sequence progresses, the threshold τ_t decreases exponentially, increasing the likelihood of teacher involvement in later tokens. Since noises of student generations accumulate in long sequences, a lower threshold in subsequent steps ensures that the teacher model can step in to maintain correct guidance. The exponentially decaying threshold thus balances the need for the student to learn from their own outputs with the necessity of preventing error accumulation in long sequences. We summarize the SWITCH algorithm in Algorithm 1

Algorithm 1 SWITCH

Input: Dataset (X, Y) of prompts and gt responses
teacher model p, initial student model q_{θ_0}
divergence $\mathcal{D}$ for training
divergence $\mathcal{D}_{\text{JSD}}$ for discrepancy
decay rate λ, learning rate η
Output: Trained student model q_{θ_I}
for x in (X, Y) **do**
  while $y_t \neq EOS$ **do**
    Compute threshold $\tau_t \leftarrow \tau_0 e^{-\lambda t}$
    Get student distribution $q_t \leftarrow q_\theta(\cdot \mid y_{<t}, x)$
    Get teacher distribution $p_t \leftarrow p(\cdot \mid y_{<t}, x)$
    Compute divergence $\mathcal{D}_{\text{JSD}} \leftarrow \mathcal{D}_{\text{JSD}}(p_t \parallel q_t)$
    if $\mathcal{D}_{\text{JSD}} \leq \tau_t$ **then**
      Sample $y_t \sim q_t$
    else
      Sample $y_t \sim p_t$
    end if
  end while
  Update student model $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{D}(p \parallel q_\theta)(y|x)$
end for

3 Experiments

3.1 Experimental Setup

Implementation Details For training, we randomly split `databricks-dolly-15k` (Conover et al., 2023) data into 14K samples for training and 500 samples for validation, following the experimental setup of Gu et al. (2024); Ko et al. (2024). Following Gu et al. (2024) and Ko et al. (2024), we incorporate a language modeling loss (Radford, 2018) using the OpenWebText (Gokaslan et al., 2019) corpus in all experiments. Similar to Agarwal et al. (2024), we use a mix of generated and ground-truth sequences, setting it at 0.5, to reduce the computational load during training. We use a decay factor of 0.1 for the main experiments. Further details can be found in Appendix A.

We evaluate our models on instruction-following tasks (Ouyang et al., 2022) across five datasets. We use Dolly (Conover et al., 2023), comprising a 500-sample test set derived from `databricks-dolly-15k`. SelfInst (Wang et al., 2022a), contains 252 samples from a user-oriented instruction-following collection. Vicuna (Chiang et al., 2023) provides 80 questions from its evaluation, spanning nine categories such as writing, roleplay, math, coding, and knowledge. S-NI (Wang et al., 2022b) features a 9K test set from Super-Natural Instructions, with samples from a wide range of tasks. UnNI (Honovich et al., 2022) consists of a 10K test set, randomly sampled from the core set of Unnatural Instructions. Similar to Gu et al. (2024), we split the Dolly and SelfInst datasets into two subsets based on the ground truth response length for further analysis, as discussed in Section 4 .

As the metric for evaluation, we report Rouge-L (Lin, 2004) for all experiments. Following Gu et al. (2024) and Ko et al. (2024), we also report GPT-4 feedback scores (Zheng et al., 2023) to DollyTest, SelfInst and VicunaEval for the results in Table 1. We report the ratio of the total score of model responses and ground truth answers, using temperature 0.7 with evaluation prompt following Zheng et al. (2023) and Ko et al. (2024). Following Ko et al. (2024), for all test sets we report the average score by sampling 5 responses with different seeds using temperature 1.0.

Models To evaluate the effectiveness of SWITCH, we examine the instruction-following capabilities across three different pre-trained model families, which include GPT-2 (Radford et al., 2019), OPT (Zhang et al., 2022), and OpenLLaMA-2 (Geng and Liu, 2023). For the GPT-2 model family, we use GPT-2 XL (1.5 billion parameters) as the teacher model, while the student models include GPT-2 Base (120M params), GPT-2 Medium (340M params), and GPT-2 Large (760M params). Regarding the OPT model family, the teacher model is OPT-2.7B, and the student models are OPT-125M, OPT-350M, and OPT-1.3B. In the OpenLLaMA-2 model family, the teacher model employs OpenLLaMA-2 with 7 billion parameters, while the student model uses OpenLLaMA-2 with 3 billion parameters.

Consistent with prior studies, we use the teacher model that is fine-tuned on the Dolly training set prior to knowledge distillation. The student model is similarly fine-tuned on the same dataset for three epochs. Evaluation is conducted using the models

Model	Parameters	Method	Dolly		SelfInst		Vicuna		S-NI	Unnatural
			GPT4	R-L	GPT4	R-L	GPT4	R-L	R-L	R-L
GPT-2	1.5B	Teacher (SFT)	49.2	27.0	38.5	15.5	39.1	16.2	25.2	33.4
	120M	SFT	34.1	22.8	23.2	10.3	24.4	15.1	17.2	21.3
		KD	34.9	23.2	23.7	11.2	24.3	15.9	17.7	22.2
		SeqKD	34.7	23.7	23.8	10.1	25.0	15.0	15.0	23.2
		GKD	35.1	23.7	25.1	12.4	27.1	17.2	23.1	26.5
		MiniLLM	35.2	23.8	25.4	12.7	27.2	17.2	24.2	27.1
		DistiLLM	36.4	25.6	26.1	13.1	29.1	18.4	27.5	28.0
		Ours	**36.9**	**26.2**	**26.9**	**13.7**	**29.6**	**18.6**	**28.1**	**28.4**
	340M	SFT	38.2	24.1	27.5	12.6	31.7	15.6	24.2	27.3
		KD	38.1	24.7	27.7	12.9	31.9	16.2	24.2	27.4
		SeqKD	38.6	24.5	27.6	13.1	32.1	16.4	25.2	27.5
		GKD	40.1	24.5	28.3	14.4	33.3	17.2	26.3	28.2
		MiniLLM	40.5	25.1	28.1	14.7	33.4	17.5	26.8	29.4
		DistiLLM	41.4	27.0	28.7	15.1	34.1	**18.2**	28.2	30.1
		Ours	**41.9**	**27.4**	**29.0**	**15.2**	**34.3**	18.0	**28.5**	**30.2**
	760M	SFT	44.3	25.4	33.1	13.7	34.0	16.5	23.2	27.5
		KD	44.6	25.7	33.4	14.1	34.5	16.7	24.5	29.4
		SeqKD	44.6	26.2	33.3	14.6	34.5	17.0	24.7	28.1
		GKD	46.1	25.1	35.1	14.9	35.1	17.3	26.6	31.2
		MiniLLM	46.5	26.4	35.2	15.2	35.8	17.9	28.0	32.3
		DistiLLM	47.1	28.4	35.8	15.5	**36.2**	**18.5**	29.2	34.1
		Ours	**47.3**	**28.5**	**36.1**	**15.7**	36.1	18.4	**29.3**	**34.5**
OPT	2.7B	Teacher (SFT)	48.7	26.2	32.1	13.3	36.9	16.6	23.4	30.4
	125M	MiniLLM	31.2	22.7	23.5	10.1	24.1	15.3	16.5	20.3
		DistiLLM	31.6	24.9	24.4	10.7	24.7	16.1	21.4	23.2
		Ours	**32.0**	**25.1**	**24.9**	**11.2**	**25.6**	**16.5**	**23.1**	**24.5**
	350M	MiniLLM	36.1	24.8	26.5	13.2	28.2	15.9	20.4	24.0
		DistiLLM	36.9	25.1	**27.1**	**14.2**	29.5	16.8	22.0	25.7
		Ours	**37.0**	**25.4**	26.9	13.7	**29.7**	**17.0**	**22.3**	**26.1**
	1.3B	MiniLLM	43.1	25.9	28.1	14.3	27.1	16.6	21.7	27.3
		DistiLLM	43.9	26.8	29.4	15.1	28.1	16.4	24.5	**30.4**
		Ours	**44.1**	**27.0**	**29.5**	**15.5**	**28.5**	**16.9**	**24.9**	30.2
OpenLLaMA2	7B	Teacher (SFT)	63.2	28.8	60.9	20.5	53.1	17.1	34.8	34.5
	3B	MiniLLM	58.2	27.4	57.1	19.8	50.9	19.4	35.4	36.2
		DistiLLM	58.9	28.3	58.4	19.7	52.0	19.5	36.1	35.8
		Ours	**59.4**	**28.6**	**58.5**	**20.0**	**52.1**	**19.6**	**36.5**	**36.3**

Table 1: Evaluation results on 5 instruction-following datasets. Each GPT4 and ROUGE-L score is averaged over 5 random seeds. The best score for each model size is highlighted in **boldface**.

with the highest ROUGE-L score on the validation set. For the training of OpenLLaMA-2, we employ low-rank adaptation (Hu et al., 2021), as in Ko et al. (2024).

Baselines We conduct a comprehensive comparison of our method against a diverse set of baseline approaches, such as SFT, which directly fine-tunes the student model on the training data without involving knowledge distillation. KD (Hinton et al., 2015) employs KLD using the teacher's output distributions as supervision at each token step. SeqKD (Kim and Rush, 2016) applies SFT to sequences generated by the teacher model. GKD (Agarwal et al., 2024) utilizes Generalized JSD and leverages both ground-truth sequences and sequences generated by the student model. MiniLLM (Gu et al., 2024) employs reverse KLD, using sequences generated by the student model and updating through policy gradient. DistiLLM (Gu et al., 2024) uses skew reverse KLD and adaptively selects whether to use ground-truth sequences or those generated by the student model.

Compared to other methods, the distinctive aspect of SWITCH is its use of sequences generated not only by the student model but also with selective involvement from the teacher model. We employ reverse skew divergence with a weight of 0.1 for our main results, following the approach

used in (Ko et al., 2024), while also demonstrating the performance enhancement of SWITCH using different divergence losses in section 4.

3.2 Experimental Results

Main Results Table 1 illustrates the instruction-following performance, showing that SWITCH consistently outperforms state-of-the-art methods across various teacher-student configurations and evaluation metrics. Notably, the performance gains are more pronounced in smaller student models that have a larger performance gap than the teacher model. This indicates that our method effectively bridges the gap between student and teacher models, especially when the student's capacity is significantly lower. By outperforming various methods that utilize SGOs, we emphasize the importance of strategic teacher intervention during the distillation process, rather than relying on outputs generated solely by students. Full table results for OPT and OpenLLaMA-2 families can be found in the Appendix C.

Performance on Different Response Lengths To further evaluate the effectiveness of our method, we analyze model performance across responses with varying ground truth lengths. In Figure 1, we compare different distillation methods using SGOs against standard KD, which does not use SGOs. The experiment, conducted with GPT-2 Small (120M) and GPT-2 XL (1.5B), reports Rouge-L scores.

It is evident that methods that utilize SGOs, such as Gu et al. (2024) and Ko et al. (2024), which rely only on SGO, experience a significant performance drop as the response length increases. In contrast, our SWITCH method maintains higher performance levels for instructions that require both short and long responses. This reduced performance degradation highlights the effectiveness of our approach in mitigating teacher misguidance in autoregressive sequence generation.

4 Analysis

Mitigating Misguidance in Lengthy SGOs In this section, we aim to determine whether teacher misguidance tends to increase with longer outputs and demonstrate that SWITCH can effectively mitigate this issue. To measure the extent of misguidance when using SGOs as training sequences, we calculate the correlation between Rouge-L scores of SGOs and training loss. To evaluate the impact of SWITCH on mitigating misguidance, we repeat the process, this time using the training loss obtained from sequences generated with SWITCH. We interpret students who require further tuning for the sample instruction based on the Rouge-L scores of the SGO—a lower coefficient indicates that students with an incorrect understanding of the given sample receive higher losses, while those with a correct understanding receive lower losses.

In this experiment, we use GPT-2 Base (120M) at the start of the distillation process and GPT-2 XL (1.5B) as the teacher. To sample, we utilize the full Dolly train split for instructions. We use Spearman correlation for the scale-invariant measures (Zar, 2005).

SGO token length	Correlation coefficient (↓)	
SWITCH	✗	✓
1 ∼10	-0.78	-0.79
11 ∼50	-0.68	-0.73
51 ∼ inf.	-0.59	-0.69

Table 2: Spearman's correlation coefficient between Rouge-L scores for SGOs and divergence loss, both with and without the use of SWITCH. Results are split by SGO token length. A lower coefficient suggests that incorrect students tend to have higher losses.

The results in Table 2 show that SGOs with longer sequences exhibit a greater tendency for misguidance. More importantly, they demonstrate that the SWITCH method effectively mitigates this misguidance, especially in longer SGOs. This underscores the crucial role of SWITCH in enhancing the training process by reducing teacher misguidance from lengthy SGOs.

Intervention Strategy	Datasets		
	Dolly	SelfInst	S-NI
Exponentially decaying threshold	26.2	13.7	28.1
w/ Linear decrease	25.1	12.7	27.1
w/ Exponential growth	23.4	12.3	24.8
Constant threshold	25.5	12.9	27.0
Teacher prob mix-in	24.6	12.7	26.3
Random teacher gen	24.8	12.5	25.2

Table 3: Comparison of the performance using different teacher involvement strategies.

Analysis of Teacher Intervention Strategies To analyze the key factors in mitigating misguidance, we conduct an ablation study to assess the impact of different teacher intervention strategies on the

student model's performance. The following strategies are compared:

Linear decrease: To explore the necessity of exponential intervention scaling, we implemented a linearly decreasing threshold. This threshold diminishes linearly from a value of 1 to 0 across the maximum response length, offering a straightforward comparison to assess whether a more constant reduction in guidance affects performance differently. *Exponential growth*: Contrasting with our primary method, this strategy employs an exponentially increasing threshold, prioritizing early sequence token selection. This approach tests whether increased early incorporation in sequence generation helps the knowledge transfer. *Constant threshold*: A threshold of 0.2 was used to determine if a fixed threshold could maintain or improve model performance compared to dynamic strategies. This constant threshold serves as a measure to examine the benefits of adaptive thresholds. *Teacher prob mix-in*: To investigate an alternative to incorporate the teacher model during the generation, we compare it against a strategy that combines the sampling distributions of teacher and student models. This method, devised in Gu et al. (2024) to address reward hacking, employs a confidence-based probability weighting as in Equation (6). We define the mix-in strength α as 0.2, consistent with the settings from Gu et al. (2024). This approach allows us to assess whether combining the probabilities can effectively train the student model compared to divergence-based thresholds.

$$\tilde{p}_t(y_t \mid y_{<t}, x) = \alpha \cdot p(y_t \mid y_{<t}, x) + (1-\alpha) \cdot q_\theta(y_t \mid y_{<t}, x) \quad (6)$$

Random Teacher Generation: This strategy randomly selects between the teacher and student models for generating each token, exploring whether arbitrary teacher involvement might either benefit or hinder the knowledge transfer.

The results shown in Table 3 indicate that our exponential decaying threshold strategy significantly outperforms other teacher intervention methods. This underscores the importance of dynamically adjusting teacher involvement based on the divergence between student and teacher distributions as the sequence progresses. Specifically, methods like the teacher probability mix-in that combines the teacher probabilities perform less effectively because they do not adjust teacher intervention based on divergence. By employing an exponentially decreasing threshold, our approach ensures that the teacher's guidance is applied when it is most beneficial, thereby enhancing knowledge transfer and effectively mitigating misguidance in the student model.

Performance Across Various Loss Functions To demonstrate the flexibility of our method, we experiment with various loss functions. Specifically, we compare performance both with and without applying our approach. We focus on various discrepancy measures commonly used in knowledge distillation for autoregressive models, including KLD, reverse KLD (RKL) (Hinton et al., 2015; Gu et al., 2024). We also experiment with Generalized JSD and skew reverse KLD (SRKL), using the weight 0.9 and 0.1 following (Agarwal et al., 2024; Ko et al., 2024).

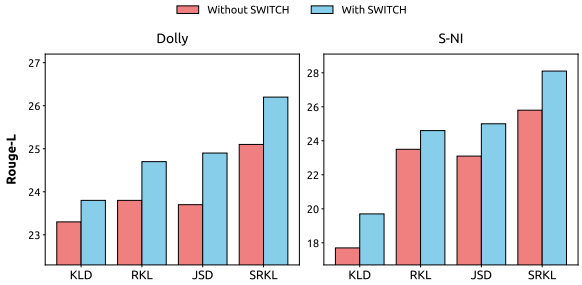

Figure 3: Application of our method SWITCH to different distillation loss

As shown in Figure 3, the consistent improvements across various loss functions highlight the robustness and adaptability of our SWITCH method. This versatility allows SWITCH to handle a wide range of distillation objectives and remain compatible with emerging loss functions aimed at capturing complex teacher distributions. Its ability to generalize across tasks ensures reliable performance under varying distillation requirements, making it a valuable contribution to autoregressive model training.

Impact of Decay Factor on Performance and Token Generation Ratio To examine the impact of various decay factors in the exponential decay threshold on performance and the ratio of tokens generated by the student versus the teacher model, we conduct experiments using different decay factors. Table 4 summarizes the results.

Decay factor	Rouge-L			Tokens generated	
	Dolly	SelfInst	S-NI	Student	Teacher
1/5	24.9	12.4	26.0	22%	78%
1/10	26.2	13.7	28.1	47%	53%
1/15	26.0	13.6	27.6	81%	19%
1/25	25.7	13.1	27.5	93%	7%

Table 4: Performance and token generation ratio using different decaying factor.

The decay factor of 1/10 yields the best performance, with a balanced proportion of tokens generated by the student and teacher models. This suggests that while our teacher intervention strategy offers significant advantages, excessive reliance on the teacher during training can negate the benefits of addressing the training-inference mismatch. Therefore, a cautious and balanced approach is necessary to optimize performance.

5 Related Works

KD for non-autoregressive models Knowledge Distillation (KD, Hinton et al., 2015) trains a smaller student model under the guidance of a more powerful teacher model, to have the student model replicate the teacher model's rich and complex representations. Various KD methods have been explored, including logit-based (Sun et al., 2019; Sanh et al., 2019), feature-based (Heo et al., 2019; Chen et al., 2021) and relation-based (Park et al., 2019; Wang et al., 2020). These methods are primarily applied to non-autoregressive architectures such as BERT and its variants for classification tasks.

KD for autoregressive models KD for autoregressive models can be broadly classified into black-box and white-box methods. Black-box methods resemble supervised fine-tuning (SFT), where the teacher model's generated outputs are used to train the student model(Hsieh et al., 2023; Jiang et al., 2023). In contrast, white-box methods utilize the teacher model's output distribution, minimizing the discrepancy between the student and teacher distributions at each time step to provide training signals (Kim and Rush, 2016).

An area of development in white-box KD involves exploring various loss functions to better align the probability distributions between teacher and student models. Due to the complexity of generative tasks and the asymmetric nature of KLD, the student distribution may fail to capture the teacher distribution (Wen et al., 2023). To address these issues, alternative loss functions such as reverse-KLD and skew divergence have been employed (Gu et al., 2024; Ko et al., 2024). In this work, we conduct experiments to assess the compatibility of our method with these loss functions, a consideration given their impact on the effectiveness of the distillation process.

Another approach involves using student-generated outputs (SGOs) as the training sequence in the distillation process. This method aims to address training-inference mismatch—often referred to as exposure bias—where autoregressive models suffer from distribution mismatch between output sequences seen during training and those generated by the student during inference (Agarwal et al., 2024). By simulating the student's own learning trajectory, SGOs have been applied to various distillation methods (Agarwal et al., 2024; Gu et al., 2024). However, the use of SGOs introduces potential risks of misguidance from the teacher, due to the distribution mismatch between the student and teacher models. Ko et al. (2024) addresses this issue with an adaptive SGO scheduler, which mixes SFT with ground-truth data while gradually increasing the usage of SGOs. Our approach differs by directly adjusting SGOs based on discrepancies with the teacher model, addressing the accumulation of errors, which hinders effective knowledge transfer in long sequences (Arora et al., 2022). This work strategically incorporates the teacher into the generation process, ensuring the sequence is reliable from the teacher's perspective for accurate guidance while preserving the benefits of SGOs.

6 Conclusion

We propose SWITCH, a novel knowledge distillation approach that strategically incorporates the teacher model during the student's training sequence generation. By detecting discrepancies in token probabilities and applying an exponentially decaying threshold for selective teacher intervention, SWITCH balances the benefits of student learning from its own outputs while effectively mitigating teacher misguidance in long sequences. Comprehensive experiments across various model families and datasets show that SWITCH outperforms baseline methods, particularly in enhancing knowledge transfer for long sequences.

Limitations

While SWITCH achieves state-of-the-art performance, it faces limitations in terms of computational resources due to the involvement of larger teacher models in the generation process. However, our approach significantly improves performance, and more importantly, the additional training time does not affect the deployment of the compressed language model. This aligns with the overall goal of knowledge distillation, which is to produce a smaller, more efficient model for deployment.

Ethics Statement

SWITCH used pre-trained language models, making it vulnerable to the inherent ethical and social risks associated with such models, as outlined by Bommasani et al. (2021); Weidinger et al. (2021). Additionally, model compression can introduce biases Hooker et al. (2020); Gonçalves and Strubell (2023). However, since most compression studies use pre-trained models, these risks are general and not specific to SWITCH.

In this work, we used publicly available datasets, ensuring they were applied in accordance with their original purposes. Our evaluations were conducted using the official API, and all models and source codes were obtained from their respective official repositories.

While writing this paper, we used an AI assistant to help draft and refine sentences at the sentence level, as well as to check for grammatical errors.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. Generalized knowledge distillation for auto-regressive language models. In *The Twelfth International Conference on Learning Representations*.

Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Chi Kit Cheung. 2022. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. *arXiv preprint arXiv:2204.01171*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. 2021. Cross-layer distillation with semantic calibration. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 7028–7036.

Wei-Lin Chiang, Zhuohan Li, Zi 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.

Xinyang Geng and Hao Liu. 2023. Openllama: An open reproduction of llama.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. Openwebtext corpus.

Gustavo Gonçalves and Emma Strubell. 2023. Understanding the effect of model compression on social bias in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2663–2675, Singapore. Association for Computational Linguistics.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. 2019. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3779–3787.

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*.

Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. 2020. Characterising bias in compressed models. *arXiv preprint arXiv:2010.03058*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger

language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. 2023. Lion: Adversarial distillation of proprietary large language models. *arXiv preprint arXiv:2305.12870*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*.

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.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5191–5198.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. 2019. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3967–3976.

Alec Radford. 2018. Improving language understanding by generative pre-training.

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.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

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 Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022b. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023. f-divergence minimization for sequence-level knowledge distillation. *arXiv preprint arXiv:2307.15190*.

Jerrold H Zar. 2005. Spearman rank correlation. *Encyclopedia of Biostatistics*, 7.

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.

Qihuang Zhong, Liang Ding, Li Shen, Juhua Liu, Bo Du, and Dacheng Tao. 2024. Revisiting knowledge distillation for autoregressive language models. *arXiv preprint arXiv:2402.11890*.

A Additional Implementation Information

Our experimental setup largely follows the implementation of (Gu et al., 2024; Ko et al., 2024), with the exception of batch size. For models with fewer than 1B parameters, we search for learning rates in {5e-4, 1e-4, 5e-5} and batch sizes of {8,16}, training for 20 epochs. For models with more than 1B parameters, we search for learning rates in {5e-5, 1e-5, 5e-6} with a batch size of 8, training for 10 epochs. We exclude samples from `databricks-dolly-15k` that exceed the models' context length. For the generation of sequences during training, we use a temperature of 1.0.

To train OpenLLaMA2, we apply low-rank adaptation (LoRA) (Hu et al., 2021) to the query and value weights with a rank of 16, as described in (Ko et al., 2024). We conduct our experiments using 2 A100 80GB GPUs. Distilling OPT-2.7B from OPT-1.3B takes less than 12 hours.

For instruction-following prompts, we use the prompt shown in Figure 4 for both training and evaluation. For GPT-4 feedback scores, we use the prompt template in Figure 5, as outlined in (Zheng et al., 2023; Ko et al., 2024).

B Qualitative Results

We present responses generated by models distilled using different methods based on GPT-2 Base (120M) models distilled from GPT-2 XL (1.5B) in Table 5. The prompts are sampled from `databricks-dolly-15k`. Our results show that SWITCH consistently produces more accurate responses compared to other knowledge distillation baselines.

C Full Table Results

The full results for the OPT family Zhang et al. (2022) are provided in Table 6, and for the OpenLLaMA-2 family Geng and Liu (2023) in Table 7.

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

Instruction:
{instruction}

Input:
{input}

Response:

Figure 4: The prompt template for training and evaluation of instruction-following task experiments, following Gu et al. (2024).

[System]
Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of the response. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, please rate the response on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

[Question]
{question}

[The Start of Assistant's Answer]
{answer}
[The End of Assistant's Answer]

Figure 5: The prompt template for evaluation from GPT-4 feedback following Zheng et al. (2023) and Ko et al. (2024).

	Case # 1
Instruction	What is a dispersive prism?
Input	In optics, a dispersive prism is an optical prism that is used to disperse light, that is, to separate light into its spectral components (the colors of the rainbow). Different wavelengths (colors) of light will be deflected by the prism at different angles. [1] This is a result of the prism material's index of refraction varying with wavelength (dispersion). Generally, longer wavelengths (red) undergo a smaller deviation than shorter wavelengths (blue). The dispersion of white light into colors by a prism led Sir Isaac Newton to conclude that white light consisted of a mixture of different colors.
Ground Truth	A dispersive prism is an optical prism that disperses the light's different wavelengths at different angles. When white light is shined through a dispersive prism it will separate into the different colors of the rainbow.
SFT	A dispersive prism is a glass prism used toatters sunlight, that is, to reduce the blinding sunlight.
KD	A dispersive prism is a prism used to bring light into a single component. The dispersion of red light into colors by a lens produces a spectrum of colors similar to that of a prism and thus it can be described as a prism.
SeqKD	A dispersive prism is a optical prism that is used in the cinema to film light.
GKD	A dispersive prism is a light that occurs when the light from a red light source is scattered. It is also known as a rainbow
MiniLLM	A dispersive prism is used to provide effective infrared light scattering, which reduces incoming light as light travels out higher from tectonic plates in space.
DistiLLM	A dispersive prism is a prism that is used to distribute light, that is, against the background.
SWITCH	A dispersive prism is an optical prism that is used to distribute light, that is,to separate light into its spectral components (the colors of the Rainbow).
	Case # 2
Instruction	How long does it take for a mustard seed to germinate?
Input	Mustard seeds are the small round seeds of various mustard plants. The seeds are usually about 1 to 2 millimetres (0.039 to 0.079 in) in diameter and may be colored from yellowish white to black. They are an important spice in many regional foods and may come from one of three different plants: black mustard (Brassica nigra), brown mustard (B. juncea), or white mustard (Sinapis alba).Grinding and mixing the seeds with water, vinegar or other liquids creates the yellow condiment known as prepared mustard. Mustard seeds generally take eight to ten days to germinate if placed under the proper conditions, which include a cold atmosphere and relatively moist soil.
Ground Truth	Mustard seeds generally take eight to ten days to germinate if placed under the proper conditions, which include a cold atmosphere and relatively moist soil
SFT	The yellowish-white to black color of a mustard seed usually starts about 1 to 2 millimeters (0.079 to 0.079) in diameter and may vary color to black or white depending on the plant.
KD	It takes thousands of days to germinate a mustard seed.
SeqKD	The length of mustard seeds depends on the type of mustard they are grown in. A common type of mustard is yellow mustard, which is grown on the outer side of a mustard plant. A yellow mustard seed can germinate into a white mustard. A brown mustard seed can germinately germinate into a red mustard.
GKD	It takes about two to three days for a mustard seed to grow. The seeds germinate from 1 to 2 millimetrons diameter each. They can be colored from yellowish to black. They are a key spice in many regional foods.
MiniLLM	The term mustard seed germinates about 1 to 2 months
DistiLLM	The time between germination and the onset of yellowish-white-to-black is about 1-2 months.
SWITCH	Mustard seeds take about 8 to 10 days to germinate.

Table 5: Instruction-following cases from the `databricks-dolly-15k` (Conover et al., 2023) dataset.

Model	Parameters	Method	Dolly		SelfInst		Vicuna		S-NI	Unnatural
			GPT4	R-L	GPT4	R-L	GPT4	R-L	R-L	R-L
OPT	2.7B	Teacher (SFT)	48.7	26.2	32.1	13.3	36.9	16.6	23.4	30.4
	125M	SFT	29.1	20.5	21.9	8.7	19.7	13.5	15.7	17.4
		KD	28.7	22.1	22.3	8.9	20.1	14.7	16.5	18.4
		SeqKD	29.4	21.4	22.4	8.4	19.6	14.6	17.4	18.1
		GKD	30.1	22.5	23.1	9.5	22.3	15.1	19.4	21.4
		MiniLLM	31.2	22.7	23.5	10.1	24.1	15.3	16.5	20.3
		DistiLLM	31.6	24.9	24.4	10.7	24.7	16.1	21.4	23.2
		Ours	**32.0**	**25.1**	**24.9**	**11.2**	**25.6**	**16.5**	**23.1**	**24.5**
	350M	SFT	33.7	23.1	24.3	11.9	25.1	15.1	18.8	21.6
		KD	33.6	23.5	24.8	12.3	27.0	16.1	18.9	21.5
		SeqKD	35.3	23.8	24.1	12.8	27.2	15.3	19.5	22.4
		GKD	35.9	23.9	26.3	13.6	27.3	16.3	19.6	24.4
		MiniLLM	36.1	24.8	26.5	13.2	28.2	15.9	20.4	24.0
		DistiLLM	36.9	25.1	**27.1**	**14.2**	29.5	16.8	22.0	25.7
		Ours	**37.0**	**25.4**	26.9	13.7	**29.7**	**17.0**	**22.3**	**26.1**
	1.3B	SFT	40.1	24.8	25.1	13.1	26.4	15.5	20.8	27.0
		KD	40.7	25.1	26.1	13.4	26.8	15.4	21.1	27.2
		SeqKD	41.5	26.1	26.2	12.8	26.1	15.7	21.1	26.6
		GKD	43.1	25.8	27.6	14.1	27.2	16.1	21.5	27.4
		MiniLLM	43.1	25.9	28.1	14.3	27.1	16.6	21.7	27.3
		DistiLLM	43.9	26.8	29.4	15.1	28.1	16.4	24.5	**30.4**
		Ours	**44.1**	**27.0**	**29.5**	**15.5**	**28.5**	**16.9**	**24.9**	30.2

Table 6: Evaluation results on 5 instruction-following datasets with OPT model family (Zhang et al., 2022). Each GPT4 and ROUGE-L score is averaged over 5 random seeds. The best score for each model size is highlighted in **boldface**.

Model	Parameters	Method	Dolly		SelfInst		Vicuna		S-NI	Unnatural
			GPT4	R-L	GPT4	R-L	GPT4	R-L	R-L	R-L
OpenLLaMA2	7B	Teacher (SFT)	63.2	28.8	60.9	20.5	53.1	17.1	34.8	34.5
	3B	SFT	51.3	25.3	50.2	16.5	41.5	15.7	29.7	30.1
		KD	52.5	26.1	51.7	16.6	43.1	16.1	30.1	32.1
		SeqKD	53.1	25.8	53.1	16.3	46.1	16.5	30.5	31.1
		GKD	57.3	27.0	57.2	19.2	49.4	19.5	34.3	35.4
		MiniLLM	58.2	27.4	57.1	19.8	50.9	19.4	35.4	36.2
		DistiLLM	58.9	28.3	58.4	19.7	52.0	19.5	36.1	35.8
		Ours	**59.4**	**28.6**	**58.5**	**20.0**	**52.1**	**19.6**	**36.5**	**36.3**

Table 7: Evaluation results on 5 instruction-following datasets with OpenLLaMA2 family (Geng and Liu, 2023). Each GPT4 and ROUGE-L score is averaged over 5 random seeds. The best score for each model size is highlighted in **boldface**.