Rethinking Kullback-Leibler Divergence in Knowledge Distillation for Large Language Models

Anonymous authors
Paper under double-blind review

Abstract

Kullback-Leiber divergence has been widely used in Knowledge Distillation (KD) to compress Large Language Models (LLMs). Contrary to prior assertions that reverse Kullback-Leibler (RKL) divergence is mode-seeking and thus preferable over the mean-seeking forward Kullback-Leibler (FKL) divergence, this study empirically and theoretically demonstrates that neither mode-seeking nor mean-seeking properties manifest in KD for LLMs. Instead, RKL and FKL are found to share the same optimization objective and both converge after a sufficient number of epochs. However, due to practical constraints, LLMs are seldom trained for such an extensive number of epochs. Meanwhile, we further find that RKL focuses on the tail part of the distributions, while FKL focuses on the head part at the beginning epochs. Consequently, we propose a simple yet effective Adaptive Kullback-Leiber (AKL) divergence method, which adaptively allocates weights to combine FKL and RKL. Metric-based and GPT-4-based evaluations demonstrate that the proposed AKL outperforms the baselines across various tasks and improves the diversity and quality of generated responses.

1 Introduction

Large Language Models (LLMs), such as GPT-4 (OpenAI, 2023), OPT (Zhang et al., 2022), and LLaMA (Touvron et al., 2023), have achieved great success in various Natural Language Process (NLP) tasks. Such strong capabilities are often attributed to the increased scale of training data and model size, such as the 13 billion parameters in LLaMA (Touvron et al., 2023). However, large model size also brings expensive computing costs and high latency, which limit the deployment for real-world applications. Therefore, it is crucial to compress the model size while keeping the performance as much as possible. Knowledge Distillation (KD, (Hinton et al., 2015)), which trains a compact student model by mimicking the behaviors of the teacher model, has been a promising method to compress the Language Models (LMs) (Wu et al., 2023). Hence, how to apply KD methods appropriately to compress LLMs has been a hot topic recently.

The vanilla KD methods (Hinton et al., 2015) try to train a better student model by aligning the logits q_θ with logits p from the teacher model via forward KL (FKL) divergence $FKL(p, q_\theta) = \sum_z p(z) \log(p(z)/q_\theta(z))$. Meanwhile, the reverse KL (RKL) divergence is defined as $RKL(p, q_\theta) = \sum_z q_\theta(z) \log(q_\theta(z)/p(z))$ and typically $RKL \neq FKL$ due to the asymmetry of KL divergence.[1] For continuous p and learnable q_θ, FKL causes the mean-seeking behavior where q_θ tends to cover the main mass of p simultaneously, while RKL causes the mode-seeking behavior where q_θ tends to select one mode (Koller & Friedman, 2009; Chan et al., 2022; Wang et al., 2023a). Therefore, recent methods (Gu et al., 2023; Agarwal12 et al., 2024; Kim et al., 2024; Ko et al., 2024) argue that RKL is more suitable than FKL to align the logits from LLMs since RKL avoids learning too many long-tail variants.

However, such mean-seeking and mode-seeking behaviors rely on **two assumptions**: 1) the student distribution q is unimodal such as the Gaussian distribution, and 2) teacher

---

[1]In this paper, we denote FKL divergence and RKL divergence as FKL and RKL, respectively.

distribution p and student distribution q are continuous. Intuitively, these two conditions **do not hold** in the KD for LLMs, where p and q are discrete and q is not guaranteed to be unimodal. In this paper, we prove the aforementioned conclusion empirically and theoretically, and further show that both FKL and RKL converge to the same objective after a sufficient number of epochs (more than 50 epochs in our demonstration experiments). In practice, we do not train for such an extensive number of epochs, such as 10 epochs in Gu et al. (2023). Meanwhile, we find that RKL focuses on the tails part of teacher distribution p while FKL focuses on the head part at the beginning epochs. Based on this observation, we propose a novel Adaptive Kullback-Leiber (AKL) divergence, aiming to align the distributions better. The key insight is to adaptively allocate the weights to combine the FKL and RKL based on p and q distributions.

We evaluate AKL on various popular benchmarks. Experimental results demonstrate that AKL outperforms all the baseline methods. Besides the Rouge-L scores, we also employ the GPT-4 (OpenAI, 2023) to score the generated responses on diversity and quality under different seeds. The GPT-4 scores show that AKL can improve the diversity and quality of generated responses. Our contributions can be concluded as follows:

- We demonstrate that the mean-seeking behavior of FKL and mode-seeking behavior of RKL do not hold in the KD for LLMs. Instead, both FKL and RKL converge to the same objective after a sufficient number of epochs.

- We find that FKL focuses on the head part and RKL focuses on the tail part at the beginning epochs, and thus propose a novel AKL to align the distributions better under typically limited epochs.

- We demonstrate the effectiveness of AKL via results on various benchmarks. Moreover, GPT-4 scores indicate that AKL improves the diversity and quality of the generated responses.

2 Related Work

2.1 KD for LLM Compression

Transformer-based Autogressive Large Language Models can be compressed via Quantization (Tao et al., 2022; Huang et al., 2024), Knowledge Distillation (Gu et al., 2023; Agarwal12 et al., 2024), Pruning (Tao et al., 2023; Xia et al., 2023), and Efficient Architecture (Wang et al., 2023b). We refer the reader to Tang et al. (2024) for a comprehensive survey. In this paper, we focus on the knowledge distillation methods for LLM compression.

Typically, KD methods for LLM can be categorized into 1) black-box KD, where only the teacher predictions are available, and 2) white-box KD, where more output from the teacher model can be employed. The Black-Box KD methods (Kim & Rush, 2016; Taori et al., 2023; Chiang et al., 2023) employ the responses of teacher models to improve the training data for student models. Meanwhile, white-box KD methods align the logits from teacher models and student models during the token generation process, through token-scale forward KL (Kim et al., 2023) or reverse KL (Gu et al., 2023; Agarwal12 et al., 2024; Kim et al., 2024; Ko et al., 2024). In this paper, we first revisit the forward KL and reverse KL in the KD for LLMs and then propose a novel AKL.

2.2 Forward KL and Reverse KL

Based on the analysis in Minka et al. (2005), reverse KL is zero forcing because $p = 0$ forces $q = 0$, and the forward KL tends to model the main mass as requiring $q > 0$ whenever $p > 0$. Due to the zero forcing behavior, reverse KL is widely used in domain adaption tasks (Nguyen et al., 2021) and reinforcement learning (Czarnecki et al., 2019). Meanwhile, forward KL is firstly employed in knowledge distillation by Hinton et al. (2015). After that, Lee et al. (2023) simply adds the forward KL and reverse KL for KD. Amara et al. (2022) proposes BD-KD, a weighted sum of forward KL and reverse KL for online KD tasks, where the weights are either 1 or λ (hyper-parameter, $\lambda > 1$) based on the entropy of student and
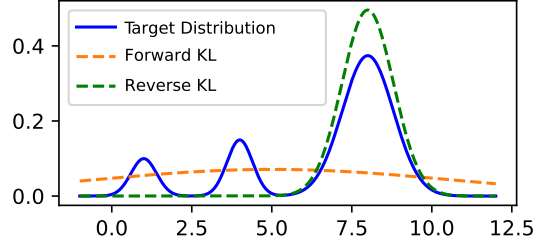
Figure 1: The toy example in Gu et al. (2023), where they fit a Gaussian mixture (distribution of teacher) with a Gaussian distribution (distribution of student) using FKL and RKL.

teacher. Compared to BD-KD, our proposed AKL assigns adaptive weights based on the head and tail parts rather than the entropy of the whole distribution.

3 Preliminary and Rethinking

3.1 Background

Conditional language generation tasks aim to generate a response $\mathbf{y} = \{y_t\}_{t=1}^T$ based on the input $\mathbf{x}$, where $y_t \in \{Y_1, Y_2, ..., Y_V\}$ and V is the vocabulary size. Traditional FKL aims to align the token generation distributions, which can be decomposed into a step-wise KL divergence (Wen et al., 2023) as follows:

$$\mathcal{J}_{FKL} = \sum_{t=1}^T FKL(p(y_t|\mathbf{y}_{<t}), q_\theta(y_t|\mathbf{y}_{<t})) = \sum_{t=1}^T \sum_{j=1}^V p(Y_j|\mathbf{y}_{<t}) \log(\frac{p(Y_j|\mathbf{y}_{<t})}{q_\theta(Y_j|\mathbf{y}_{<t})}). \tag{1}$$

Meanwhile, we can replace the FKL with RKL, and thus get:

$$\mathcal{J}_{RKL} = \sum_{t=1}^T RKL(p(y_t|\mathbf{y}_{<t}), q_\theta(y_t|\mathbf{y}_{<t})) = \sum_{t=1}^T \sum_{j=1}^V q_\theta(Y_j|\mathbf{y}_{<t}) \log(\frac{q_\theta(Y_j|\mathbf{y}_{<t})}{p(Y_j|\mathbf{y}_{<t})}). \tag{2}$$

Recent methods (Gu et al., 2023; Agarwal12 et al., 2024; Kim et al., 2024; Ko et al., 2024) argue that RKL is more suitable than FKL because FKL leads to the mean-seeking behavior and RKL leads to the mode-seeking behavior. As shown in Figure 1, mean-seeking means the q_θ tries to cover the main mass (the mean of three peaks) of p simultaneously. Meanwhile, mode-seeking means q_θ assigns high probabilities to the main mass (largest peak).

3.2 Rethinking FKL and RKL

Empirical Analysis   However, the mean-seeking behavior of FKL and the mode-seeking behavior of RKL relies on two assumptions: 1) student distribution q_θ follows the Gaussian distribution, and 2) both teacher distribution p and student distribution q_θ are continuous. Intuitively, since q_θ follows the Gaussian distribution, then the learned distribution needs to select where to cover most, either the main mass (FKL) or one peak (RKL).

In the KD for LLMs, these two assumptions do not hold anymore. On the one side, student distribution q_θ is calculated by the softmax function and does not follow the standard Gaussian distribution. On the other side, both teacher distribution p and student distribution q_θ are discrete and defined on vocabulary set $\{Y_1, Y_2, ..., Y_V\}$. Figure 2 shows the results on toy examples under the setting of KD for LLMs. For FKL and RKL, we employ the same initialized q_θ and train for 200 epochs. Both FKL and RKL converge to p perfectly after optimizing for 200 epochs regardless of the shape of p.

Theoretical Analysis   We also demonstrate the phenomena theoretically. Since the $\mathcal{J}_{FKL}$ and $\mathcal{J}_{RKL}$ are additive considering generated tokens, we can analyze the convergences
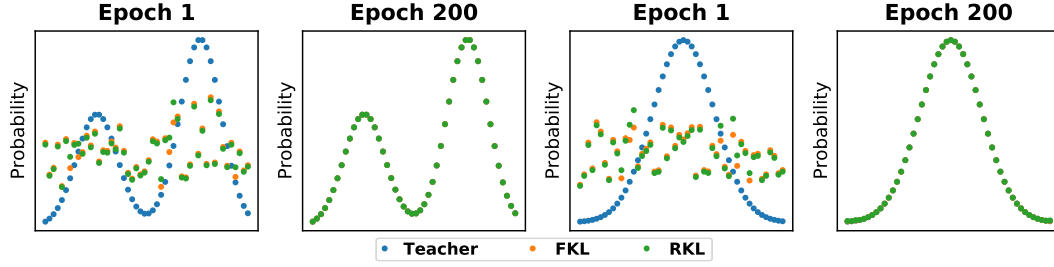
Figure 2: The convergence of FKL and RKL on toy data under epoch 1 and epoch 200. The initial distribution q is the same for FKL and RKL. After 200 epochs, both FKL and RKL can converge to the target distribution well regardless of the shape of p.

at each token separately. Let $(z_1^p, z_2^p, z_3^p, ..., z_V^p)$ and $(z_1^q, z_2^q, z_3^q, ..., z_V^q)$ denote the output at step t before the final softmax operation from the teacher model and the student model, respectively. We have

$$p(Y_j|\mathbf{y}_{<t}) = \frac{\exp(z_j^p)}{\sum\limits_{k=1}^{V} \exp(z_k^p)}; \quad q_\theta(Y_j|\mathbf{y}_{<t}) = \frac{\exp(z_j^q)}{\sum\limits_{k=1}^{V} \exp(z_k^q)}. \tag{3}$$

Then the gradient for z_j^q under FKL and RKL can be calculated by the chain rule as follows:

$$\frac{\partial FKL(p, q_\theta)}{\partial z_j^q} = q_\theta(Y_j|\mathbf{y}_{<t}) - p(Y_j|\mathbf{y}_{<t}) \tag{4}$$

$$\frac{\partial RKL(p, q_\theta)}{\partial z_j^q} = q_\theta(Y_j|\mathbf{y}_{<t})(\log(\frac{q_\theta(Y_j|\mathbf{y}_{<t})}{p(Y_j|\mathbf{y}_{<t})}) - RKL(p, q_\theta)). \tag{5}$$

Considering the converge condition:

$$\frac{\partial F(R)KL(p, q_\theta)}{\partial z_j^q} = 0 \ \forall j \in \{1, 2, 3, ..., V\}, \tag{6}$$

we can infer that for both FKL and RKL, the sufficient and necessary condition for converge is

$$q_\theta(Y_j|\mathbf{y}_{<t}) = p(Y_j|\mathbf{y}_{<t}) \ \forall j \in \{1, 2, 3, ..., V\}. \tag{7}$$

Please refer to Appendix A for the proof. In summary, RKL and FKL share the same optimization objective, forcing the student model to generate the same logits as the teacher model.

Deeper Insights FKL and RKL are the special cases of f-divergence (Sason & Verdú, 2016). The f-divergence is defined as:

$$D_f(P||Q) = \int f(\frac{dP}{dQ})dQ, \tag{8}$$

where f is convex and $f(1) = 0$. Following the Jensen's inequality:

$$D_f(P||Q) = \int f(\frac{dP}{dQ})dQ \geq f(\int \frac{dP}{dQ}dQ) = f(1) = 0. \tag{9}$$

For FKL $f(x) = x\log(x)$, and for RKL $f(x) = -\log(x)$. Then we can revisit the mode-seeking and mean-seeking behaviors in the view of f-divergence:
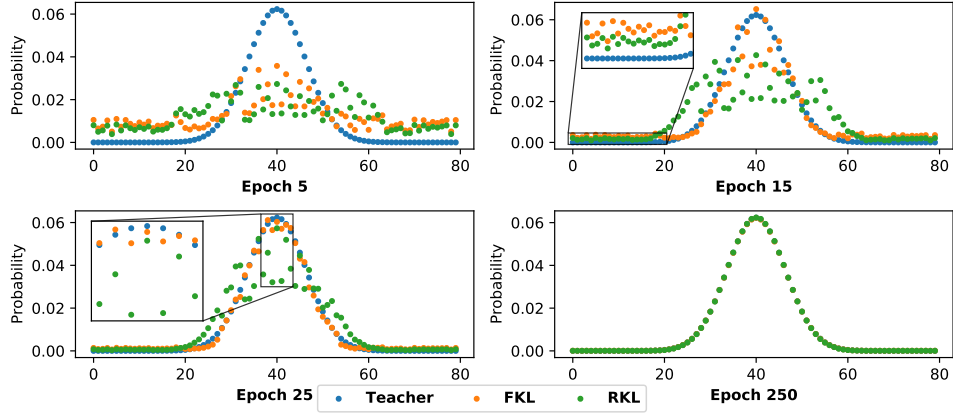
Figure 3: The distributions at various epochs for FKL and RKL on toy data, where the teacher distribution and initial student distribution are the same. We can find that FKL focuses on the head part and RKL on the tail part at the beginning epochs, and both converge finally.

- Taking $f(x) = x \log(x)$, we have $\lim_{x \to +\infty} f(x) = +\infty$, so the FKL avoids $\frac{p(z)}{q(z)}$ goes to $+\infty$. Since $0 < p(z) < 1$ and $0 < q(z) < 1$, $q(z)$ should not be too small especially when $p(z)$ is large. Therefore, q would try to cover (assign larger value) as many peaks of p as possible, leading to the mean-seeking behavior of FKL.

- Taking $f(x) = -\log(x)$, we have $\lim_{x \to +\infty} f(x) = -\infty$ and $\lim_{x \to 0^+} f(x) = +\infty$. According to Equation 9, RKL is nonnegative and thus only avoids $\frac{p(z)}{q(z)}$ go to 0^+, which means $q(z)$ should not be too large ($q(z) \to 1$) when $p(z)$ is small ($p(z) \to 0$). Therefore, q would cover one mode of p when q follows the Gaussian distribution, which is also known as the mode-seeking behavior of RKL.

Difference between FKL and RKL   In this paper, we demonstrate that the mean-seeking behavior of FKL and the mode-seeking behavior of RKL do not hold in the KD for LLMs empirically and theoretically. Instead, FKL and RKL share the same optimizing objective, that is, q_θ from the student is the same as p from the teacher. We thus ask: *What is the difference between FKL and RKL in the KD for LLMs?*

The answer is the optimization process. Considering $FKL(p, q_\theta) = \sum p(z) \log \frac{p(z)}{q_\theta(z)}$, larger $p(z)$ means a larger weight in total loss and also more likely to generate a larger $\log \frac{p(z)}{q_\theta(z)}$. Hence, fitting the area with larger $p(z)$ (a.k.a. the head part) is the priority of FKL. Similarly, $RKL(p, q_\theta)$ is defined as $\sum q_\theta(z) \log \frac{q_\theta(z)}{p(z)}$. When $p(z)$ is smaller, $\frac{q_\theta(z)}{p(z)}$ is easier to be $+\infty$, leading to a larger loss. Therefore, fitting the area with smaller $p(\cdot)$ (a.k.a. the tail part) is the priority of RKL.

Figure 3 shows the results on toy data. For RKL and FKL, we employ the same teacher and initial student distribution. After training for 250 epochs, both FKL and RKL converge as mentioned before. However, we can find that RKL fits better at the tail part and FKL fits better at the head part at beginning epochs.

Summary   Deeper insights on FKL and RKL in KD for LLMs can be concluded as follows.

In the KD for LLMs, the mean-seeking and mode-seeking behaviors do not hold for forward KL (FKL) and reverse KL (RKL), respectively. Instead, they share the same optimization objective. Meanwhile, FKL focuses on the head part and RKL focuses on the tail part at the beginning.

4 Methodology

4.1 Motivation

In Figure 3, we can find that both FKL and RKL converge to the teacher distribution after 250 epochs. However, when performing experiments on LLMs, we **do not train for such an extensive number of epochs**, especially on the large corpus. For example, Gu et al. (2023) trains only 20 epochs for models with less than 1.3B parameters and 10 epochs for others. Meanwhile, under such a small epoch, FKL and RKL focus on the head and tail parts, respectively. Both FKL and RKL fail to align the logit distributions well. To tackle it, one intuition is to combine both the FKL and the RKL. However, how to assign the loss weights to balance FKL and RKL becomes the next challenge.

4.2 Adaptive Kullback-Leiber Divergence

In this paper, we propose a novel Adaptive Kullback-Leiber (AKL) divergence method, which adaptively allocates the weights to combine FKL and RKL. The key idea is to assign larger weights to FKL if the gap on the head part is larger than the tail part, and vice versa. Specifically, for the logit distribution $p(y_t|\mathbf{y}_{<t}) \in \mathbb{R}^V$ from the teacher model and $q_\theta(y_t|\mathbf{y}_{<t}) \in \mathbb{R}^V$ from the student model at token t, we first define the indicate mask $M \in \mathbb{R}^V$ as follows:

$$M[i] = \begin{cases} 1 & \text{when } Y_i \text{ belongs to head part} \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

In AKL, we can get one feasible M by solving:

$$min \sum_{i=0}^{V} M[i] \ \ s.t. \ \sum_{j=0}^{V} M[j]p(Y_j|\mathbf{y}_{<t}) \geq \mu, \tag{11}$$

where the hyper-parameter μ is the threshold such as 0.5. We can easily solve this optimizing problem by sorting the $p(Y_j|\mathbf{y}_{<t})$. After that, the gap g_{head} on the head part and g_{tail} on the tail part are calculated by:

$$g_{head} = \sum_{i=0}^{V} M[i]\epsilon(p(Y_i|\mathbf{y}_{<t}), q_\theta(Y_i|\mathbf{y}_{<t})); \tag{12}$$

$$g_{tail} = \sum_{i=0}^{V} (1 - M[i])\epsilon(p(Y_i|\mathbf{y}_{<t}), q_\theta(Y_i|\mathbf{y}_{<t})) \tag{13}$$

where $\epsilon(\cdot)$ is the function to define the point-wise gap. Then AKL is defined as

$$AKL(p, q_\theta) = \frac{g_{head}}{g_{head} + g_{tail}}FKL(p, q_\theta) + \frac{g_{tail}}{g_{head} + g_{tail}}RKL(p, q_\theta). \tag{14}$$

In this way, a larger gap g_{head} would bring a larger weight to FKL, and vice versa.

5 Experiments

5.1 Implement

Training For training data, we employ the instruction response dataset following Gu et al. (2023), which is built from databricks-dolly-15k[2] and contains 14k samples for training, 500 samples for valid, and 500 samples for testing. We first fine-tune the teacher model and then distill the teacher model. For teacher models, we select GPT-2 (Radford et al., 2019) with 1.5B parameters and Llama (Touvron et al., 2023) with 7B parameters. Meanwhile, the

---

[2]https://github.com/databrickslabs/dolly/tree/master

Method	GPT2 1.5B → GPT2 120M			LLaMA 6.7B → TinyLLaMA 1.1B		
	Dolly (500)	S-NI (1694)	UnNI (10000)	Dolly (500)	S-NI (1694)	UnNI (10000)
Teacher	26.98±0.27	27.25±0.16	31.61±0.16	26.73±0.65	32.75±0.24	34.61±0.08
SFT	23.01±0.18	16.48±0.28	18.43±0.09	22.05±0.38	27.79±0.20	25.96±0.13
SeqKD	23.30±0.38	16.35±0.17	18.51±0.04	22.67±0.57	26.97±0.32	27.35±0.06
FKL	23.46±0.56	16.63±0.48	19.27±0.06	22.24±0.38	28.07±0.41	26.93±0.09
RKL	22.62±0.34	17.88±0.17	19.35±0.06	23.95±0.53	28.90±0.41	27.89±0.08
FKL+RKL	23.36±0.53	17.83±0.23	20.37±0.08	24.08±0.51	30.98±0.31	30.48±0.10
AKL (Ours)	**23.88±0.46**	**19.15±0.21**	**21.97±0.13**	**24.40±0.42**	**31.37±0.23**	**31.05±0.17**

Table 1: The results of Rouge-L scores under different distillation settings. Our proposed AKL outperforms all baselines without any extra parameters, especially in the larger UnNI dataset.

students are GPT-2 with 120M parameters and TinyLlama (Zhang et al., 2024) with 1.1B parameters[3], respectively.

In AKL, we set μ as 0.5 and the gap function $\epsilon(p(z), q(z)) = |p(z) - q(z)|$. For GPT-2, we set the batch size as 32 and train for 20 epochs. The learning rate is 5e-4. For TinyLlama, we set the batch size as 60 and train for 10 epochs. The learning rate is 1e-5. For all students, the maximum input length is 512. It takes around 1h to train on 4 Nvidia A100 GPUs.

Evaluation After distillation, we evaluate the student models on Dolly (contains 500 samples), S-NI (contains 1694 samples), and UnNI (contains 10000 samples). We report the results of the Rouge-L scores for five different seeds. More details can be found in Appendix B. For the baselines, we employ the following methods:

- **SFT**: directly train the student model on the dataset without distillation;
- **SeqKD** (Kim & Rush, 2016): train the student model on the data generated from the teacher model;
- **FKL**: employ forward KL during knowledge distillation;
- **RKL**: employ reverse KL during knowledge distillation;
- **FKL+RKL**: simply combine forward KL and reverse KL via $0.5FKL + 0.5RKL$.

In particular, we **do not** compare with MiniLLM since it pre-trains on extra corpus including OpenWebText (Gokaslan et al., 2019) and training data for RoBERTa (Liu et al., 2019).

5.2 Main Results

Table 1 reports the performance of the teacher and student models under different settings. Some findings are summarized as follows:

- Previous methods (Gu et al., 2023; Agarwal12 et al., 2024) argue that RKL is more suitable than FKL. However, the gap between FKL and RKL is minor as shown in Table 1. In particular, FKL gets a score of 23.46 for GPT2 on Dolly, which is larger than the 22.62 of RKL.
- Simply adding FKL and RKL (a.k.a FKL+RKL) brings a better performance than FKL and RKL, which is consistent with the conclusion in previous work (Lee et al., 2023). Since FKL focuses on the head part and RKL focuses on the tail part, adding them would help the student learn more information.
- Our proposed AKL, which adaptively assigns weights based on the gaps between logits, learns a better student than all baselines **without any extra parameters**. In particular, on the larger dataset UnNI with 10000 samples, AKL outperforms FKL+AKL with a p-value less than 0.001 for both GPT 2 and TinyLLama.

[3]https://huggingface.co/TinyLlama/TinyLlama-1.1B-intermediate-step-1195k-token-2.5T
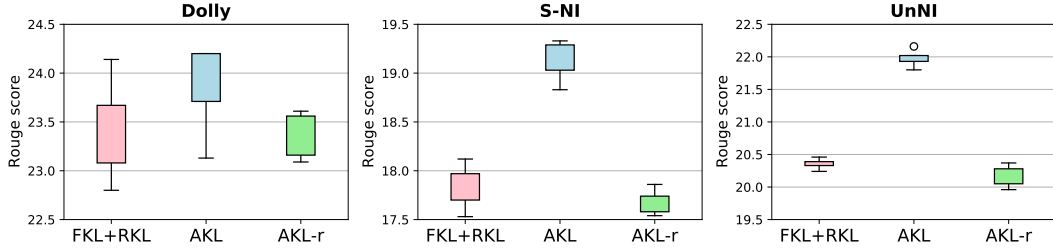
Figure 4: The results of FKL+RKL, proposed AKL, and AKL-r on GPT 2 120M. After flipping the loss weight, AKL-r performs worse than AKL on all three datasets.

Methods	Closed QA	General QA	Open QA	Information Extraction	Summary	Brain Storming	Creative Writing	Classification
Teacher	31.52	16.95	24.99	24.60	30.73	23.21	14.62	54.12
SFT	22.75	14.43	16.68	22.09	25.21	16.27	13.48	46.64
SeqKD	22.95	15.51	17.47	<u>24.68</u>	27.27	15.79	13.52	<u>51.60</u>
FKL	**27.46**	16.07	17.68	24.30	**28.10**	15.59	14.15	47.01
RKL	23.99	<u>16.32</u>	<u>20.07</u>	19.54	26.27	**19.40**	13.76	51.39
FKL+RKL	24.49	14.94	18.40	21.79	25.33	17.13	<u>14.29</u>	48.00
AKL (Ours)	<u>26.85</u>	**16.79**	**21.17**	**25.43**	<u>27.55</u>	<u>18.69</u>	**15.00**	**53.19**

Table 2: Rouge-L scores for TinyLLama 1.1B on different subtasks from Dolly. **Bold** denotes the best results and <u>underline</u> for the second. FKL and RKL perform well on different sub-tasks (i.e. FKL on Closed QA and RKL on Brain Storming). Meanwhile, our proposed AKL can combine the advantages of FKL and RKL well while FKL+RKL fails.

6 Extensive Analysis

6.1 Ablation: Flipping the Loss Weights

As shown in Equation 14, we assign a larger weight to FKL if the gap on the head part is larger than the tail part. For ablation, we flip the loss weights and then propose AKL-r, which is defined as:

$$AKL\text{-}r(p, q_\theta) = \frac{g_{tail}}{g_{head} + g_{tail}} FKL(p, q_\theta) + \frac{g_{head}}{g_{head} + g_{tail}} RKL(p, q_\theta). \quad (15)$$

In this way, the weight for FKL would be smaller when the gap is larger on the head part. Figure 4 shows the results on GPT 2. After flipping the loss weights, AKL-r performs worse than AKL and FKL+RKL, which proves the effectiveness of weight assignment in AKL.

6.2 Performance under Various Tasks

We study the performance on various subtasks from Dolly, which contains 8 subtasks in the test set. Table 2 shows the performance of TinyLLama 1.1B. FKL performs well on the Closed QA and Summary subtasks, while RKL performs well on the Brain Storming. FKL+RKL tends to perform moderately and the scores typically lie in the range of FKL and RKL, such as 48.00 on the Classification sub-task which is between 47.01 of FKL and 51.39 of RKL. Meanwhile, AKL adaptively assigns weights to add FKL and RKL and thus can combine the advantages well. In particular, AKL even outperforms the teacher model on the Information Extraction and Creative Writing sub-tasks.

Moreover, we evaluate AKL for more benchmarks including Winogrande (Sakaguchi et al., 2021), OpenBookQA (Mihaylov et al., 2018), BoolQ (Clark et al., 2019), and ARC (Clark et al., 2018). Experimental results indicate that AKL outperforms all the baselines, demonstrating its robustness. Please refer to Appendix D for more details.
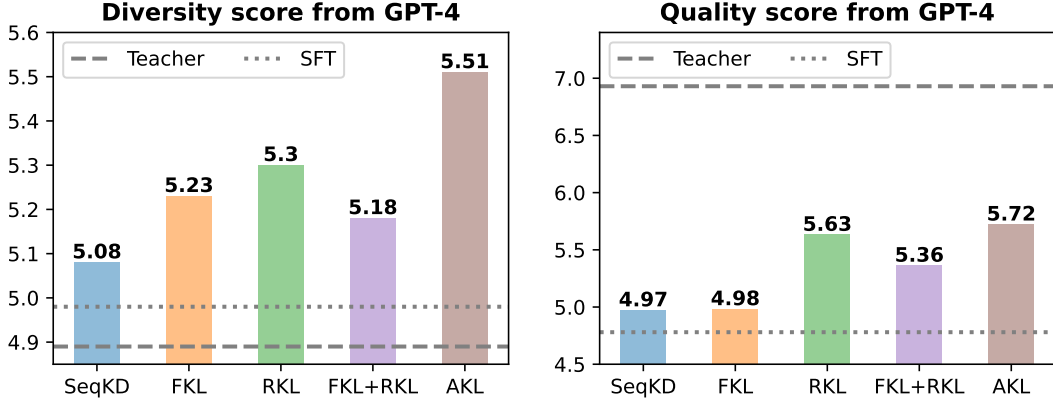
Figure 5: The scores of diversity and quality from GPT-4 (total score: 10) on the responses from TinyLLama. AKL can improve the diversity and quality of generated responses compared to the baselines.

6.3 Evaluation via GPT-4

We report Rouge-L scores in the aforementioned experiments to demonstrate the effectiveness of the proposed AKL. Meanwhile, the scores from powerful LLMs such as GPT-4 are also an important assessment for generated responses. Therefore, we design prompts and request GPT-4 to score the generated responses on **Diversity** and **Quality**. The diversity score evaluates the ability to generate diverse answers, especially on the open question. The quality score evaluates the ability to respond reasonably and accurately. Please refer to Appendix C for more details.

Figure 5 indicates the average diversity and quality scores for the baselines and AKL. Both FKL and RKL can improve the diversity since they mix the information (text generation space) from teacher and student. It is worth noting that diversity scores **do not** evaluate the correctness of the answers. Meanwhile, FKL and RKL also improve the quality of responses by aligning with the teacher model. In conclusion, AKL outperforms all the baselines considering diversity and quality **without** any extra parameters, demonstrating its effectiveness.

6.4 Case Study

We further analyze the cases of generated responses from TinyLLama. Please refer to Appendix E for detailed examples. In short, AKL can effectively distill the knowledge from the teacher model, improving the diversity and quality of generated responses.

7 Conclusion

This work first demonstrates that the mean-seeking behavior of FKL and the mode-seeking behavior of RKL do not hold in the KD for LLMs. Instead, FKL and RKL share the same optimization objective that the student generates the same logits as the teacher, and both FKL and RKL converge after a sufficient number of epochs. However, in practice, we train for far less epochs. Meanwhile, we find that FKL focuses on the head part and RKL focuses on the tail part at the beginning epochs. Therefore, we proposed a novel AKL to add FKL and RKL adaptively. Empirical results on various students demonstrate the effectiveness of AKL. Besides, GPT-4 evaluation shows that AKL can improve the diversity and quality of generated responses. In future work, we would consider scaling AKL to larger models with different structures and more tasks.

References

Rishabh Agarwal12, Nino Vieillard, Yongchao Zhou13, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. 2024.

Ibtihel Amara, Nazanin Sepahvand, Brett H Meyer, Warren J Gross, and James J Clark. Bd-kd: Balancing the divergences for online knowledge distillation. 2022.

Alan Chan, Hugo Silva, Sungsu Lim, Tadashi Kozuno, A. Rupam Mahmood, and Martha White. Greedification operators for policy optimization: Investigating forward and reverse KL divergences. *J. Mach. Learn. Res.*, 23:253:1–253:79, 2022.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. 2(3):6, 2023.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Wojciech M Czarnecki, Razvan Pascanu, Simon Osindero, Siddhant Jayakumar, Grzegorz Swirszcz, and Max Jaderberg. Distilling policy distillation. In *The 22nd international conference on artificial intelligence and statistics*, pp. 1331–1340. PMLR, 2019.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023.

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

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Knowledge distillation of large language models. *CoRR*, abs/2306.08543, 2023. doi: 10.48550/ARXIV.2306.08543.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14409–14428, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.806.

Wei Huang, Yangdong Liu, Haotong Qin, Ying Li, Shiming Zhang, Xianglong Liu, Michele Magno, and Xiaojuan Qi. Billm: Pushing the limit of post-training quantization for llms. 2024.

Gyeongman Kim, Doohyuk Jang, and Eunho Yang. Promptkd: Distilling student-friendly knowledge for generative language models via prompt tuning. *arXiv preprint arXiv:2402.12842*, 2024.

Minsoo Kim, Sihwa Lee, Janghwan Lee, Sukjin Hong, Du-Seong Chang, Wonyong Sung, and Jungwook Choi. Token-scaled logit distillation for ternary weight generative language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, 2016.

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

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009. ISBN 978-0-262-01319-2.

Hyoje Lee, Yeachan Park, Hyun Seo, and Myungjoo Kang. Self-knowledge distillation via dropout. *Computer Vision and Image Understanding*, 233:103720, 2023.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. 2019.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Tom Minka et al. Divergence measures and message passing. Technical report, Technical report, Microsoft Research, 2005.

A Tuan Nguyen, Toan Tran, Yarin Gal, Philip HS Torr, and Atılım Güneş Baydin. Kl guided domain adaptation. *arXiv preprint arXiv:2106.07780*, 2021.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Igal Sason and Sergio Verdú. f-divergence inequalities. *IEEE Transactions on Information Theory*, 62(11):5973–6006, 2016.

Yehui Tang, Yunhe Wang, Jianyuan Guo, Zhijun Tu, Kai Han, Hailin Hu, and Dacheng Tao. A survey on transformer compression. 2024.

Chaofan Tao, Lu Hou, Wei Zhang, Lifeng Shang, Xin Jiang, Qun Liu, Ping Luo, and Ngai Wong. Compression of generative pre-trained language models via quantization. pp. 4821–4836, 2022.

Chaofan Tao, Lu Hou, Haoli Bai, Jiansheng Wei, Xin Jiang, Qun Liu, Ping Luo, and Ngai Wong. Structured pruning for efficient generative pre-trained language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 10880–10895, 2023.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. doi: 10.48550/ARXIV.2302.13971.

Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. Beyond reverse kl: Generalizing direct preference optimization with diverse divergence constraints. 2023a.

Jiahao Wang, Songyang Zhang, Yong Liu, Taiqiang Wu, Yujiu Yang, Xihui Liu, Kai Chen, Ping Luo, and Dahua Lin. Riformer: Keep your vision backbone effective but removing token mixer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14443–14452, 2023b.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5085–5109, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.340.

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

Taiqiang Wu, Cheng Hou, Zhe Zhao, Shanshan Lao, Jiayi Li, Ngai Wong, and Yujiu Yang. Weight-inherited distillation for task-agnostic BERT compression. *CoRR*, abs/2305.09098, 2023. doi: 10.48550/ARXIV.2305.09098.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared llama: Accelerating language model pre-training via structured pruning. *arXiv preprint arXiv:2310.06694*, 2023.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. 2024.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068, 2022. doi: 10.48550/ARXIV.2205.01068.

A   FKL and RKL converge

Proof. Considering the converge condition for FKL:

$$\frac{\partial FKL(p, q_\theta)}{\partial z_j^q} = 0 \; \forall j \in \{1, 2, 3, ..., V\}. \tag{16}$$

We have

$$\frac{\partial FKL(p, q_\theta)}{\partial z_j^q} = q_\theta(Y_j|\mathbf{y}_{<t}) - p(Y_j|\mathbf{y}_{<t}) \; \forall j \in \{1, 2, 3, ..., V\}, \tag{17}$$

and then

$$\frac{\partial FKL(p, q_\theta)}{\partial z_j^q} = 0 \; \forall j \in \{1, 2, 3, ..., V\} \Leftrightarrow q_\theta(Y_j|\mathbf{y}_{<t}) = p(Y_j|\mathbf{y}_{<t}) \; \forall j \in \{1, 2, 3, ..., V\}. \quad \square$$

Proof. Considering the converge condition for RKL:

$$\frac{\partial RKL(p, q_\theta)}{\partial z_j^q} = 0 \; \forall j \in \{1, 2, 3, ..., V\}. \tag{18}$$

We have

$$\frac{\partial RKL(p, q_\theta)}{\partial z_j^q} = q_\theta(Y_j|\mathbf{y}_{<t})(\log(\frac{q_\theta(Y_j|\mathbf{y}_{<t})}{p(Y_j|\mathbf{y}_{<t})}) - RKL(p, q_\theta)). \tag{19}$$

Sufficiency. Put $q_\theta(Y_j|\mathbf{y}_{<t}) = p(Y_j|\mathbf{y}_{<t}) \; \forall j \in \{1, 2, 3, ..., V\}$ into 19:

$$\begin{aligned}
\frac{\partial RKL(p, q_\theta)}{\partial z_j^q} &= q_\theta(Y_j|\mathbf{y}_{<t})(\log(\frac{q_\theta(Y_j|\mathbf{y}_{<t})}{p(Y_j|\mathbf{y}_{<t})}) - RKL(p, q_\theta)) \\
&= q_\theta(Y_j|\mathbf{y}_{<t})RKL(p, q_\theta) \\
&= q_\theta(Y_j|\mathbf{y}_{<t})(\sum_{m=1}^{V} q_\theta(Y_m|\mathbf{y}_{<t}) \log(\frac{q_\theta(Y_m|\mathbf{y}_{<t})}{p(Y_m|\mathbf{y}_{<t})})) \\
&= 0 \; \forall j \in \{1, 2, 3, ..., V\}
\end{aligned} \tag{20}$$

Necessity. When Equation 18 holds, then $\forall i, j \in \{1, 2, 3, ..., V\}$, we can get:

$$q_\theta(Y_i|\mathbf{y}_{<t})\frac{\partial RKL(p, q_\theta)}{\partial z_j^q} - q_\theta(Y_j|\mathbf{y}_{<t})\frac{\partial RKL(p, q_\theta)}{\partial z_i^q} = q_\theta(Y_i|\mathbf{y}_{<t}) * 0 - q_\theta(Y_j|\mathbf{y}_{<t}) * 0 = 0. \tag{21}$$

Put Equation 19 into Equation 21, thus

$$q_\theta(Y_i|\mathbf{y}_{<t})q_\theta(Y_j|\mathbf{y}_{<t}) \log(\frac{q_\theta(Y_j|\mathbf{y}_{<t})p(Y_i|\mathbf{y}_{<t})}{p(Y_j|\mathbf{y}_{<t})q_\theta(Y_i|\mathbf{y}_{<t})}) = 0. \tag{22}$$

Meanwhile, we have $0 < q_\theta(Y_i|\mathbf{y}_{<t}) < 1$ and $0 < q_\theta(Y_j|\mathbf{y}_{<t}) < 1$, therefore

$$\begin{aligned}
&\frac{q_\theta(Y_j|\mathbf{y}_{<t})}{q_\theta(Y_i|\mathbf{y}_{<t})} = \frac{p(Y_j|\mathbf{y}_{<t})}{p(Y_i|\mathbf{y}_{<t})} \; \forall i, j \in \{1, 2, 3, ..., V\} \\
\rightarrow &\sum_{i=1}^{V} \frac{q_\theta(Y_i|\mathbf{y}_{<t})}{q_\theta(Y_m|\mathbf{y}_{<t})} = \sum_{i=1}^{V} \frac{p(Y_i|\mathbf{y}_{<t})}{p(Y_m|\mathbf{y}_{<t})} \; \forall m \in \{1, 2, 3, ..., V\} \\
\rightarrow &\frac{\sum_{i=1}^{V} q_\theta(Y_i|\mathbf{y}_{<t})}{q_\theta(Y_m|\mathbf{y}_{<t})} = \frac{\sum_{i=1}^{V} p(Y_i|\mathbf{y}_{<t})}{p(Y_m|\mathbf{y}_{<t})} \; \forall m \in \{1, 2, 3, ..., V\} \\
\rightarrow &\frac{1}{q_\theta(Y_m|\mathbf{y}_{<t})} = \frac{1}{p(Y_m|\mathbf{y}_{<t})} \; \forall m \in \{1, 2, 3, ..., V\} \\
\rightarrow &q_\theta(Y_m|\mathbf{y}_{<t}) = p(Y_m|\mathbf{y}_{<t}) \; \forall m \in \{1, 2, 3, ..., V\}.
\end{aligned} \tag{23}$$

To summarize, we have:

$$\frac{\partial RKL(p, q_\theta)}{\partial z_j^q} = 0 \ \forall j \in \{1, 2, 3, ..., V\} \Leftrightarrow q_\theta(Y_j | \mathbf{y}_{<t}) = p(Y_j | \mathbf{y}_{<t}) \ \forall j \in \{1, 2, 3, ..., V\}. \quad \square$$

B Evaluation details

The details for the evaluation dataset are as follows:

- Dolly: human-written instruction-response pairs [4]. We divide the data set into 14k samples for training, 500 samples for validation, and 500 samples for testing following Gu et al. (2023).
- S-NI: the test set of SUP-NATINST (Wang et al., 2022), which contains 9K samples from 119 English tasks. In this paper, we employ the samples with ground truth responses longer than 11.
- UnNI: dataset from Honovich et al. (2023). Similarly, we employ the samples with ground-truth responses longer than 11.

We use five random seeds $\{10, 20, 30, 40, 50\}$ for generation.

C Scores from GPT-4

We design prompts to get scores from the GPT-4 model. For the diversity score, the prompt is:

> Give a score (1-10, the higher the better) to judge the diversity of 5 answers to one question. Q: [*Input*] Ans 1: [*Answer 1*] Ans 2: [*Answer 2*] Ans 3: [*Answer 3*] Ans 4: [*Answer 4*] Ans 5: [*Answer 5*]. Just output the score.

where *Input* and *Answer 1-5* denote the corresponding input and responses under five seeds (10, 20, 30, 40, 50). For the quality score, we first get the score for each answer and then calculate the average, aiming to avoid the influence of other answers. The prompt is designed as:

> Give a score (1-10, the higher the better) to judge the quality of the answer to the question. Q: [*Input*] Ans: [*Answer*]. Just output the score.

For the five responses from the same input, we record the individual scores and calculate the average as the final score. Due to the cost, we employ the responses from the Open QA and Closed QA subtasks.

D Results on more benchmarks

We further evaluate the performance of TinyLLaMa on the following benchmarks:

- Winogrande (Sakaguchi et al., 2021): the task of common sense reasoning, we report the accuracy under the setting of zero-shot learning;
- OpenBookQA (Mihaylov et al., 2018): question answering task modeled after open book exams, we report the accuracy under the setting of zero-shot learning;
- BoolQ (Clark et al., 2019): question answering task for yes/no questions, we report the accuracy under the setting of zero-shot learning;

---

[4]https://github.com/databrickslabs/dolly/tree/master

Methods	Winogrande	OpenBookQA	BoolQ	ARC
Teacher	68.70	37.20	78.56	51.71
SFT	59.03	27.60	62.50	35.84
SeqKD	59.11	26.80	62.23	36.43
FKL	59.27	26.80	63.14	36.09
RKL	59.66	27.80	63.45	36.09
FKL+RKL	59.58	28.20	62.99	35.92
AKL (Ours)	**60.77**	**28.60**	**63.48**	**36.52**

Table 3: The scores of TinyLLama. **Bold** denotes the best.

- ARC (Clark et al., 2018): a new dataset of 7787 genuine grade-school level, multiple-choice science questions, we report the normalized accuracy of the challenge part under the 5-shot prompt.

We implement the evaluation based on the LM-Evaluation-Harness (Gao et al., 2023) framework. As shown in Table 3, our proposed AKL outperforms all the baselines, which indicates the robustness of AKL.

E   Cases

We show some cases for the outputs of baselines and proposed AKL. AKL can effectively inherit the knowledge from the teacher model, such as the artistry (see Table 4), grammar information (see Table 5), correct information (see Table 6), and representation skill (see Table 7).

Input	Write a haiku about how much I love pickles.
GPT-4	Crunchy, briny treat,/ Love for pickles knows no bounds, / Savor every bite.
Teacher	Pickles I love, / Like a salty hug.
SFT	Syrup so sweet, pickles my soul.
SeqKD	A slipping pickle,/ With melty mayo drizzled high,/ Aha, divine!
FKL	Salty goodness in a bundle... / I love pickles!
RKL	I love pickles so much/ They're definitely delicious/ You should try them too
FKL+RKL	Love is fickle, / But pickles are eternal.
AKL (Ours)	Bulbs of great deliciousness / Shining in the sun / Like precious jewels.

Table 4: One case from the **Creative Writing** subtask in Dolly. Only AKL and FKL+RKL learn the artistry from the teacher model.

Input	Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. ###Instruction: Given a description of an object, answer the question 'What is it?' ###Input: A thing that has mass and occupies space.
GPT-4	It is matter.
Teacher	A thing that has mass and occupies space is call **an object**.
SFT	It is **a object**!
SeqKD	a glove
FKL	It is exactly what it seems - a statement that is exactly true. It is a mass/space-occupying object, and much more.
RKL	A thing that Has Mass and Occupies Space
FKL+RKL	A thing that has mass and occupies space
AKL (Ours)	**An object** is a thing that has mass and occupies space.

Table 5: One case from the **UnNI** dataset. There is a grammar mistake for the student model (SFT, 'a' should be 'an'). AKL corrects that mistake with the knowledge from the teacher model.

Input	Tell me whether these cities are in Texas: Austin, Houston, New York, Chicago, Miami, Dallas.
GPT-4	In Texas, you will find the following cities from your list: 1. Austin; 2. Houston; 3. Dallas. The other cities you mentioned are located in different states: 1. New York is in New York State; 2. Chicago is in Illinois; 3. Miami is in Florida.
Teacher	Austin, Houston and Dallas are all cities in Texas. New York, Chicago and Miami are cities in other states. However, there is a Miami in Texas as well! Texas Miami is near the Florida one, but is still a completely different city.
SFT	Austin and Dallas are in Texas. Everyone else is not in Texas.
SeqKD	Austin and Houston are in Texas. New York, Chicago, and Dallas are not in Texas. Dallas is in the state of Texas, while Texas is not alone in the United States. Miami is located in Florida, and Houston is part of the state of Texas, but is located outside of the American Samoa territory. American Samoa is the only territory within the 50 United States that is not owned by the United States. The last city, Texas, does not belong to any of those listed.
FKL	Austin and Dallas are in Texas. Houston, New York and Chicago are cities outside of Texas.
RKL	Austin. Houston. Dallas. New York. Chicago. Miami.
FKL+RKL	Austin, Houston, and Dallas are in Texas. New York, Chicago, and Miami are Not in Texas.
AKL (Ours)	Austin, Houston, and Dallas are in Texas. New York and Chicago are not in Texas. Miami is in Florida, and Dallas is in Texas.

Table 6: One case from the **Classification** subtask in Dolly. AKL is the only method that 1) gets the correct answer and 2) figures out that Miami is in Florida.

16

Input	What should I do on a free afternoon in San Francisco?
Teacher	There are many options for someone with an afternoon free in San Francisco. The city has a wide variety of museums, shops, and landmarks. The Golden Gate Bridge is a must-see if you haven't seen it yet. If museums are your thing, then the De Young and the Legion of honor are both fine options as well. One interesting lesser known option is to walk across the market near the Ferry Building and then walk along the Embarcadero until you get to the Oakland Bay Bridge. Then, as you head back, stop into the Rockridge district for some great wine and cheese at shops like Bioulet, and shop for groceries at the Rasputin deli. If none of these strike your fancy, then San Francisco has a number of taxi cabs and Uber/Lyft rides will be very affordable and easy to come by.
SFT	When in San Francisco, you should start with a walk down Market Street. Gridley Gallery and the Legion of Honor are also recommended. If you like shopping, try Union Square. Find a map located at the info booths around the city and figure out which tour boats go down the Embarcadero. If you like taking pictures, then head up to the Fort Mason area, specifically the Canada Gallery. Have some dinner at Japanese forest and then walk Little H influencer. End your evening at the standard Pier 39.
SeqKD	Find a nice restaurant to eat at, take a walk along the Embarcadero, browse at various boutique as you walk along, then win at the casino at the Fisher Fine Arts Museum.
FKL	San Francisco is a very walkable city, so there are many free attractions in San Francisco. Visitors preferably take a trolley tour to see all the main attractions, but walkable as well. Here are some free attractions: Golden Gate Park; DeYoung Museum; Golden Gate Bridge; Museum of Contemporary Irish and International Art; de Young Museum; Golden Gate Park parade; Explorie trail/mile long iterpretive park trail through Golden Gate Park.
RKL	Go to the DeYoung Museum, pick up lunch at K & L Bakery on Polk Street, and walk around Chestnut Street Studio District.
FKL+RKL	One could explore Golden Gate Park and the SF Museum of Modern Art, or simply walk around the city. The Italian Street Art walk is also fun.
AKL (Ours)	There are many things to do and see in San Francisco. You can visit the more famous landmarks such as Lombard Street cycle and the Golden Gate Bridge. Downtown San Francisco is also home to numerous restaurants and bars. If you are interested in history, then the Golden Gate Park is a great place to spend a afternoon exploring the local art scene. You can also walk around the Fisher House, Hilltop Park and the Conservatory of Flowers. For the more adventurous, there are many walking tours of San Francisco, including the trolley tour and one of the Vessel San Francisco.

Table 7: One case from the **Creative Writing** subtask in Dolly. Only AKL maintains the style (i.e. if you are interested in ##, you can ##) and generates rich suggestions.