Revisiting Knowledge Distillation for Autoregressive Language Models

Qihuang Zhong[1], **Liang Ding**[2], **Li Shen**[3], **Juhua Liu**[1*], **Bo Du**[1*], **Dacheng Tao**[4]

[1] School of Computer Science, National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence and Hubei Key Laboratory of Multimedia and Network Communication Engineering, Wuhan University, China

[2] The University of Sydney, Australia [3] Sun Yat-sen University, China [4] Nanyang Technological University, Singapore

{zhongqihuang, liujuhua, dubo}@whu.edu.cn

{mathshenli, liangding.liam, dacheng.tao}@gmail.com

Abstract

Knowledge distillation (KD) is a common approach to compress a teacher model to reduce its inference cost and memory footprint, by training a smaller student model. However, in the context of autoregressive language models (LMs), we empirically find that larger teachers might dramatically result in a poorer student. In response to this problem, we conduct a series of analyses and reveal that *different tokens have different teaching modes*, neglecting which will lead to performance degradation. Motivated by this, we propose a simple yet effective **adaptive teaching** approach (ATKD) to improve the KD. The core of ATKD is to reduce rote learning and make teaching more diverse and flexible. Extensive experiments on 8 LM tasks show that, with the help of ATKD, various baseline KD methods can achieve consistent and significant performance gains (up to +3.04% average score) across all model types and sizes. More encouragingly, ATKD can improve the student model generalization effectively.

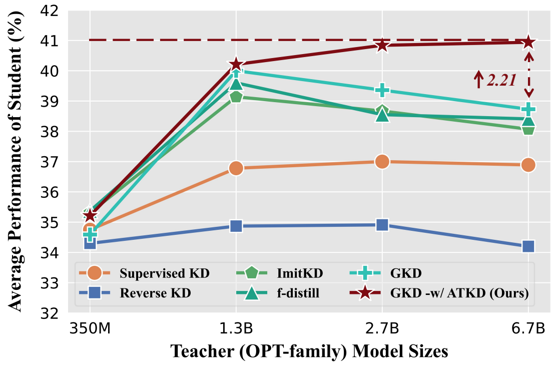

Figure 1: **Comparisons of different KD methods** for distilling the student (OPT-125M). The x-axis denotes the OPT-based teacher sizes, while the y-axis denotes the average performance of students on $\mathcal{S}_{\text{NLG}}$ and $\mathcal{S}_{\text{NLU}}$. The evaluation details are in §4. Notably, ATKD can be combined with various KD methods, and we only report the results of "GKD + ATKD" for ease of illustration.

1 Introduction

Autoregressive language models (LMs), such as GPT-4 (OpenAI, 2023), PaLM (Chowdhery et al., 2023) and LLaMA2 (Touvron et al., 2023), have achieved great success in a numerous tasks (Zhong et al., 2023; Peng et al., 2023b; Lu et al., 2023). However, with the scaling of model size, the inference and deployment of these LMs become more computationally expensive and memory intensive, hindering the development of industrial applications. Hence, it is crucial and green to compress these LMs and accelerate the inference, while not losing much performance (Schwartz et al., 2020).

To achieve this goal, a common approach is knowledge distillation (KD), which aims to compress a large teacher model by distilling its knowledge into a small student model (Hinton et al., 2015; Kim and Rush, 2016). Recently, in the context of autoregressive LMs, various novel learning algorithms have been proposed to achieve better distillation performance (Wen et al., 2023; Agarwal et al., 2024). Despite their remarkable performance, we empirically find a **counter-intuitive** phenomenon, where *larger teachers might dramatically result in a poorer student, especially when the model capability gap is large*. As illustrated in Figure 1, the performance of student degrades when the teachers are too large, which is similar to the findings of Mirzadeh et al. (2020); Cho and Hariharan (2019); Zhang et al. (2023).

Although a few works aim to investigate this problem and propose to fill the gap, they are mostly studied for vision models (Mirzadeh et al., 2020; Cho and Hariharan, 2019) or discriminative language understanding models (Zhang et al., 2023), while the autoregressive KD for generative LMs is yet to be explored. In this work, we investigate this problem from the perspective of the distillation objective, which is at the core of autoregressive KD.

* Corresponding Authors: Juhua Liu (e-mail: liujuhua@whu.edu.cn), Bo Du (e-mail: dubo@whu.edu.cn)

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10900–10913
August 11-16, 2024

Specifically, taking the classical token-level KD objective, *i.e., forward KL-Divergence*, as an example, we first reformulate it as two parts: 1) **target-oriented knowledge distillation** (TKD), which enforces the student model to learn the target-related information ; 2) **diversity-oriented knowledge distillation** (DKD), which encourages the student to learn more diverse knowledge from the teacher in the non-target classes. These two parts are tied by a token-wise factor, which reflects the teacher's uncertainty and we denote it as **uncertainty coefficient** (UNC). After reformulating the distillation objective, we conduct a series of preliminary analyses on the popular OPT-family (Zhang et al., 2022) models, and find that:

> ❶ UNC measures the learning difficulties of tokens, where the hard-to-learn ones are more important for KD.
>
> ❷ DKD contributes more but is greatly suppressed, especially for the larger teachers.
>
> ❸ TKD plays different roles in tokens with different learning difficulties.

Based on these observations, we can conclude that **different tokens have different teaching modes**, and (one of) the limitations of KD comes from the neglect of this principle. To address this limitation, we propose a simple yet effective adaptive teaching method (referred to as **ATKD**) to improve the KD. The core of ATKD is to reduce rote learning and make teaching more diverse and flexible. Specifically, ATKD skips the target-oriented teaching for the (less-informative) easy-to-learn tokens and pays more attention to the diverse learning of hard-to-learn tokens.

We evaluate ATKD on a variety of LM benchmarks, including 5 language generation tasks and 3 language understanding tasks, upon 3 types of autoregressive LMs: OPT (Zhang et al., 2022), Pythia (Biderman et al., 2023) and LLaMA (Touvron et al., 2023). Results show that ATKD can not only alleviate the problem of performance degradation in larger teachers, but also bring consistent and significant improvements (up to +3.04% average score) into various baseline KD methods among all model types and sizes. Moreover, compared to the standard KD, ATKD can effectively improve the generalization of distilled students.

Contributions. To summarize, our contributions are three-fold: (1) Our study reveals that *different tokens have different teaching modes*, neglecting which will cause the sub-optimal distillation performance, especially in larger teachers. (2) We propose a simple yet effective, plug-and-play approach (ATKD) to alleviate this problem and improve the quality of teaching. (3) Extensive experiments show that ATKD outperforms the standard KD with up to +3.04% average gains and improves the student's model generalization effectively.

2 Rethinking Knowledge Distillation for Autoregressive LMs

In this section, we first delve into the mechanism of classic knowledge distillation and then present the empirical analyses of this strategy in detail.

2.1 Recap of Knowledge Distillation

Notations. For autoregressive LMs, the classic KD aims to approximately minimize Kullback-Leibler (KL) divergence between the teacher and student output distribution at each token (Hinton et al., 2015). Let $\mathbf{y} = \{\mathrm{y}_1, ..., \mathrm{y}_T\}$ denote the target sequence and V denote the vocabulary, we refer to $\mathbf{y}_{<t}$ as $\{\mathrm{y}_1, ..., \mathrm{y}_{t-1}\}$, where $t \in \{1, ..., T\}$ and $\mathrm{y}_t \in V$. Specifically, the loss function can be formulated as:

$$\begin{aligned}\mathcal{L}_{\mathrm{KL}}(\mathbf{p}||\mathbf{q}) &= -\sum_{t=1}^{T} \mathrm{KL}\left(\mathbf{p}(\mathrm{y}_t|\mathbf{y}_{<t})||\mathbf{q}(\mathrm{y}_t|\mathbf{y}_{<t})\right) \\ &= -\sum_{t=1}^{T} \mathbf{p}(\mathrm{y}_t|\mathbf{y}_{<t}) \log\left(\frac{\mathbf{p}(\mathrm{y}_t|\mathbf{y}_{<t})}{\mathbf{q}(\mathrm{y}_t|\mathbf{y}_{<t})}\right),\end{aligned}$$

where $\mathbf{p} = [p_1, ..., p_C]$ and $\mathbf{q} = [q_1, ..., q_C]$[1] are the predicted distributions of the teacher and student, respectively; p_i is the probability of the i-th class and C is the number of vocabulary V, KL refers to the KL divergence. For simplicity, we denote $\mathbf{p}(\mathrm{y}_t|\mathbf{y}_{<t})$ as $\mathbf{p}^t$, and p_i^t as the probability of the i-th class at t-th step. Here, p_i^t is determined using a softmax function:

$$p_i^t = \frac{\exp(z_i^t)}{\sum_{j=1}^{C} \exp(z_j^t)}, \tag{1}$$

where z_i^t represents the logit of the i-th class in V. Let g_t denote the target token/class at t-th step, we can obtain the binary probabilities $\mathbf{p}_{\mathbf{b}}^t = [p_{g_t}^t, p_{\backslash g_t}^t]$,

[1]For simplicity, we only consider the formulation of $\mathbf{p}$ in the following context. Note that the $\mathbf{q}$ is similar to $\mathbf{p}$.

where probability of the target class $p^t_{g_t}$ and non-target classes $p^t_{\backslash g_t}$ can be calculated as:

$$p^t_{g_t} = \frac{\exp(z^t_{g_t})}{\sum_{j=1}^{C}\exp(z^t_j)}, p^t_{\backslash g_t} = \frac{\sum_{k=1,k\neq g_t}^{C}\exp(z^t_k)}{\sum_{j=1}^{C}\exp(z^t_j)}.$$

Moreover, for independently analyzing the probabilities among non-target classes, we declare $\hat{\mathbf{p}}^t = [\hat{p}^t_1, ..., \hat{p}^t_{g_t-1}, \hat{p}^t_{g_t+1}, ..., \hat{p}^t_C]$, where $\hat{p}^t_i$ is:

$$\hat{p}^t_i = \frac{\exp(z^t_i)}{\sum_{j=1,j\neq g_t}^{C}\exp(z^t_j)}. \tag{2}$$

Reformulation of $\mathcal{L}_{\text{KL}}$. Here, we are inspired by Zhao et al. (2022)[2], and attempt to reformulate $\mathcal{L}_{\text{KL}}$ with the binary probabilities $\mathbf{p}^t_{\mathbf{b}}$ and the probabilities among non-target classes $\hat{\mathbf{p}}^t$, which can be reformulated as:

$$\mathcal{L}_{\text{KL}} = -\sum_{t=1}^{T}(p^t_{g_t}\log(\frac{p^t_{g_t}}{q^t_{g_t}}) + \sum_{j=1,j\neq g_t}^{C} p^t_j \log(\frac{p^t_j}{q^t_j})). \tag{3}$$

According to Eq. 1 and 2, we have $p^t_i = \hat{p}^t_i * p^t_{\backslash g_t}$, and can further rewrite Eq. 3 as:

$$\begin{aligned}
\mathcal{L}_{\text{KL}} = &-\sum_{t=1}^{T}\Bigg(p^t_{g_t}\log(\frac{p^t_{g_t}}{q^t_{g_t}}) \\
&+p^t_{\backslash g_t}\sum_{j=1,j\neq g_t}^{C}\hat{p}^t_i\left(\log(\frac{\hat{p}^t_j}{\hat{q}^t_j}) + \log(\frac{p^t_{\backslash g_t}}{q^t_{\backslash g_t}})\right)\Bigg) \\
= &-\sum_{t=1}^{T}\Bigg(p^t_{g_t}\log(\frac{p^t_{g_t}}{q^t_{g_t}}) + p^t_{\backslash g_t}\log(\frac{p^t_{\backslash g_t}}{q^t_{\backslash g_t}}) \\
&+p^t_{\backslash g_t}\sum_{j=1,j\neq g_t}^{C}\hat{p}^t_i\log(\frac{\hat{p}^t_j}{\hat{q}^t_j})\Bigg) \\
= &-\sum_{t=1}^{T}\left(\text{KL}(\mathbf{p}^t_{\mathbf{b}}||\mathbf{q}^t_{\mathbf{b}}) + p^t_{\backslash g_t}\text{KL}(\hat{\mathbf{p}}^{\mathbf{t}}||\hat{\mathbf{q}}^{\mathbf{t}})\right).
\end{aligned} \tag{4}$$

As seen, we can reformulate the classic KD objective as a combination of binary classification loss on the target class, and KL loss on the non-target classes. The former forces the student to learn the target-related information, and we thus denote it as **target-oriented knowledge distillation** (TKD). Conversely, the latter encourages the student to distill the diverse knowledge among non-target classes, and we denote it as **diversity-oriented knowledge distillation** (DKD). Moreover, we find that TKD and DKD are tied by a token-wise factor $p^t_{\backslash g_t}$, which could reflect the teacher's uncertainty on the tokens, *i.e.*, the larger $p^t_{\backslash g_t}$ denotes the more uncertainty[3] in the teacher output distribution. Hence, we refer to $p^t_{\backslash g_t}$ as **uncertainty coefficient** (UNC).

[2]Although the reformulation of $\mathcal{L}_{\text{KL}}$ is inspired by the previous work (Zhao et al., 2022), we take a further step by exploring the potential mechanism of autoregressive KD from the perspective of teaching modes among different tokens, which are our main contributions.

2.2 Empirical Analyses

Setting. We conduct experiments by first fine-tuning larger LMs on the instruction-response dataset $\mathcal{D}$ as teachers. Then, we use different KD methods to distill a smaller student on $\mathcal{D}$ with the teacher's guidance. Here, we use the original OPT-125M as the student and use the other OPT-family models (*i.e.*, OPT-350M/-1.3B/-2.7B/-6.7B) as teachers. Alpaca-GPT4 (Peng et al., 2023a) is used as training data, and the models are evaluated on three instruction-following datasets, *i.e.*, DollyEval (Gu et al., 2023), VicunaEval (Chiang et al., 2023) and SelfInst (Wang et al., 2022). We follow (Gu et al., 2023) and use the LLM-based metric, *i.e.*, **LLM-as-a-Judge**, to quantify the model responses. Specifically, we ask GPT-3.5-Turbo-1106[4] to compare model responses with the ground-truth answers and raise 1-10 scores for both responses and report the ratio of the total score of model responses and ground-truth answers.

Findings. To reveal the drawbacks of $\mathcal{L}_{\text{KL}}$ and explore the reasons for performance degradation in large teachers, we conduct systematic analyses to investigate the different effects of UNC, TKD and DKD, respectively. Through the extensive analyses, we empirically observe that:

❶ UNC measures the learning difficulties of tokens, where the hard-to-learn ones are more important for KD. Motivated by the token imbalance nature and the truth that different tokens in a sequence contribute differently to the sentence meaning (Church and Hanks, 1990; Chen et al., 2020), we conjecture that different tokens play different roles in autoregressive KD. Intuitively, the tokens with less uncertainty have simple learning patterns and *easy-to-learn*, while the more uncertain

[3]For example, the token with $p_{\backslash g_t} = 0.7$ is more uncertain than the one with $p_{\backslash g_t} = 0.1$.

[4]The analysis of this evaluator is shown in Appendix A.2.

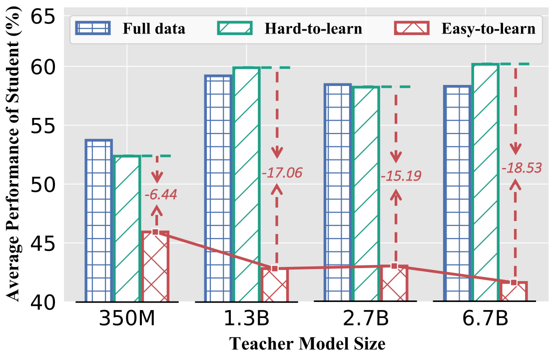

Figure 2: **Comparisons of different training tokens.** The y-axis denotes the average performance of students (OPT-125M) on the evaluated tasks, while the x-axis denotes the sizes of OPT-based teachers.

Method	350M	1.3B	2.7B	6.7B
1) Full data are used.				
TKD-only	49.19	48.01	47.21	48.29
DKD-only	**54.00**	**57.78**	**59.43**	**60.42**
TKD+DKD	52.97	57.01	58.66	58.70
2) Easy-to-learn tokens are used.				
TKD-only	39.21	43.82	42.37	41.43
DKD-only	**48.68**	**54.43**	**58.26**	**60.02**
TKD+DKD	45.59	44.97	45.09	44.66
3) Hard-to-learn tokens are used.				
TKD-only	47.40	45.15	44.63	48.32
DKD-only	51.42	58.51	55.47	59.88
TKD+DKD	**53.26**	**60.49**	**60.60**	**61.47**

Table 1: **Comparisons of different teaching objectives.** The best results within the same training set are in **bold**.

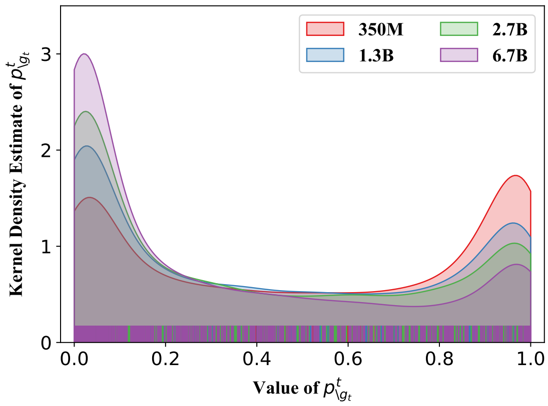

Figure 3: **Illustration of distributions of UNC ($p^t_{\backslash g_t}$)** among different OPT-based teachers on 100 training samples (about 10K tokens). In particular, we use the kernel density estimate for visualizing, where the larger density refers to more tokens.

tokens are more informative and are *hard-to-learn*. To verify our conjecture, we rank the training tokens according to the UNC for each mini-batch and evenly split them into two subsets. For clarity, one subset (denoted as "hard-to-learn") includes samples with top-50% uncertainty, while the remaining samples are in the other subset (denoted as "easy-to-learn"). We train the student model with vanilla $\mathcal{L}_{\text{KL}}$ on different training sets, and illustrate the results in Figure 2.

Obviously, training on the "hard-to-learn" tokens achieves much better performance than on the "easy-to-learn" tokens, and even outperforms the full-data training. This indicates that *tokens with more uncertainty contain more "dark knowledge" and are more important for KD*. Conversely, due to the shallow patterns of easy-to-learn tokens, forcing the student to learn from them might suffer from over-fitting, leading to poorer performance. More interestingly, *this phenomenon seems to be more significant in larger teachers*.

❷ DKD contributes more (than TKD) but is greatly suppressed, especially for the larger teachers. Here, we delve into the individual effect of TKD and DKD by comparing the performance of (1) "TKD-only", (2) "DKD-only" and (3) "TKD+DKD" (where both are decoupled and simply added, *i.e.*, ignoring the effect of UNC). The contrastive results among different training sets (as mentioned in ❶) are listed in Table 1.

As seen,"DKD-only" outperforms the "TKD-only" among all model sizes and training sets by a large margin, indicating that the diversity-oriented knowledge is of vital importance to autoregressive KD. However, in Eq. 4, we can find that the effect of DKD is suppressed by the UNC (ranging from 0 to 1), which might lead to the sub-optimal performance. To verify it, we further analyze the distributions of UNC across different model sizes. In practice, we randomly sample 100 instances from the training dataset and illustrate the distributions of UNC in Figure 3. It can be seen that UNC is generally smaller (tends to be 0) in large models than in small models, *i.e.*, the larger models, the more suppressed the effect of DKD. This is also indicated by the results of "TKD+DKD", as removing the UNC seems to alleviate the performance degradation problem in the large models (except training on easy-to-learn tokens, where the further analyses are shown in ❸). In general, these analyses prove that *DKD is more important but is greatly suppressed by the* UNC *in the larger models*, which could be the main reason why a larger teacher leads to a poorer student.

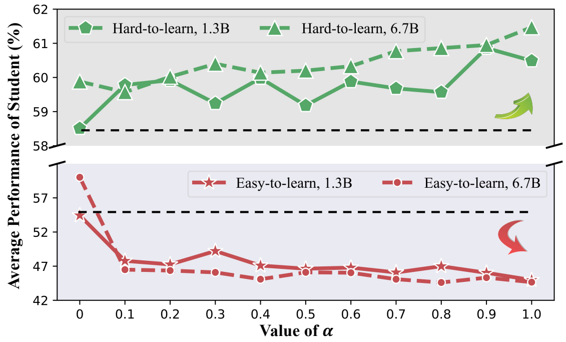

Figure 4: **Effect of TKD in different training tokens.** Here, we report the performance of students distilled with "$\alpha\times$TKD+DKD", where α is varied from 0 to 1. For ease of illustration, we only illustrate the results of using OPT-1.3B and OPT-6.7B as teachers.

❸ TKD plays different roles in tokens with different learning difficulties. We can observe an interesting phenomenon in Table 1, where adding TKD upon DKD ("TKD+DKD") seems to dramatically result in performance degrades when training on the easy-to-learn set, compared to the singly DKD (*e.g.*, decreasing from 60.02% to 44.66%). Conversely, in the case of hard-to-learn tokens, adding TKD brings remarkable performance gains. These results motivate us to investigate the special effect of TKD on different tokens, by comparing the performance of different combinations of TKD and DKD in the setting of "$\alpha\times$TKD+DKD". The contrastive performance of varied α is illustrated in Figure 4. It can be seen that TKD indeed behaves differently in different training sets. TKD hurts the knowledge transfer of easy-to-learn tokens, but is beneficial to the learning of hard-to-learn tokens. We attribute it to the different learning difficulties of tokens, as *the target-oriented learning on easy-to-learn tokens might damage the diversity of students* (Tan et al., 2008). On the other hand, *adding target-related supervision signals could reduce the learning difficulties on the hard-to-learn tokens*, thus leading to better performance.

3 Improving Knowledge Distillation with Adaptive Teaching Modes

Based on the observations in §2, we recognize that ***different tokens have different teaching modes***, and the side effect (*i.e.*, problem degrades in larger teachers) of KD mainly comes from the neglect of this principle. To this end, we propose to improve the autoregressive KD with adaptive teaching modes (ATKD). In this section, we introduce the ATKD approach in detail.

Motivation and Overview of ATKD. In addition to the empirical findings in §2, our ATKD is also inspired by a famous education initiative (Tan et al., 2008), "Teach Less, Learn More", which highlights that *reducing rote learning and making education more diverse and flexible can improve the quality of teaching and enhance student learning*. Intuitively, due to the large capability gap between teacher and student models, target-oriented learning of easy-to-learn tokens may encourage the student to simply mimic the teacher's shallow style but not to learn its dark knowledge (Gudibande et al., 2023). That is, the student might fall short in generalizing to more tasks, leading to sub-optimal performance. Motivated by this, our ATKD aims to encourage the students to learn from different perspectives for different tokens. In short, ATKD skips the target-oriented teaching for the easy-to-learn tokens, and pays more attention to the learning of diverse knowledge in the hard-to-learn tokens. By doing so, our ATKD forces the student to learn more flexible and diverse knowledge, and thus improve overall performance.

To achieve this goal, we should first obtain the easy-/hard-to-learn tokens. As mentioned in ❶ of §2.2, UNC can effectively measure the learning difficulties of tokens, and we thus use it as a metric to select the easy-/hard-to-learn tokens. Specifically, for each mini-batch, we rank the training tokens according to UNC and select the top-k[5] tokens as hard-to-learn tokens, while the others are easy-to-learn. Then, ATKD performs the KD processes with adaptive teaching modes as follows:

Adaptive Teaching Modes of ATKD. As aforementioned, TKD and DKD contribute differently in easy-/hard-to-learn tokens. Thus, instead of using a unified teaching mode for all tokens, we use adaptive teaching modes for easy-to-learn and hard-to-learn tokens, respectively. Specifically, we decouple the TKD and DKD (*i.e.*, DKD will not be suppressed by the UNC) to enhance the diverse learning of students. Moreover, for the easy-to-learn tokens, considering that the student can easily learn the target-class information, we skip the target-oriented teaching, *i.e.*, removing TKD. On the other hand, both TKD and DKD are used for hard-to-learn tokens, as we empirically found that target-oriented teaching is essential to the learning of hard-to-learn tokens. The learning objectives of

[5] k ranges from 0% to 100%, and is set as 50% by default. The analysis of k can be found in §4.3.

different tokens can be formulated as:

$$\mathcal{L}_{\mathrm{KL}}^{e} = -\sum_{t \in \mathcal{D}_e} \mathrm{KL}(\hat{\mathbf{p}}^{\mathbf{t}} || \hat{\mathbf{q}}^{\mathbf{t}}),$$

$$\mathcal{L}_{\mathrm{KL}}^{h} = -\sum_{t \in \mathcal{D}_h} \mathrm{KL}(\mathbf{p}_{\mathbf{b}}^{t} || \mathbf{q}_{\mathbf{b}}^{t}) + \mathrm{KL}(\hat{\mathbf{p}}^{\mathbf{t}} || \hat{\mathbf{q}}^{\mathbf{t}}),$$

where $\mathcal{D}_e$ and $\mathcal{D}_h$ denote the sets of easy-to-learn and hard-to-learn tokens, respectively.

Additionally, since the hard-to-learn tokens contain more informative knowledge and are more important, we adaptively combine the easy-to-learn $\mathcal{L}_{\mathrm{KL}}^{e}$ and hard-to-learn $\mathcal{L}_{\mathrm{KL}}^{h}$ objectives and formulate the overall learning objective of ATKD as:

$$\mathcal{L}_{\mathrm{KL}}^{all} = \lambda * \mathcal{L}_{\mathrm{KL}}^{e} + (1-\lambda) * \mathcal{L}_{\mathrm{KL}}^{h}, \tag{5}$$

where λ is a weight factor to balance the different objectives, which is empirically[6] set as 0.2.

4 Evaluation

4.1 Setup

Tasks and Datasets. We conduct extensive experiments on various LM benchmarks, covering a diversity of language generation tasks (denoted as $\mathcal{S}_{\mathrm{NLG}}$) and language understanding tasks (denoted as $\mathcal{S}_{\mathrm{NLU}}$). Specifically, $\mathcal{S}_{\mathrm{NLG}}$ consists of 5 widely-used generation tasks, *i.e.*, DollyEval (Gu et al., 2023), VicunaEval (Chiang et al., 2023), SelfInst (Wang et al., 2022), Koala (Geng et al., 2023), and WizardLM (Xu et al., 2023) benchmarks. $\mathcal{S}_{\mathrm{NLU}}$ includes 3 popular classification tasks, *i.e.*, MMLU (Hendrycks et al., 2020), Drop (Dua et al., 2019) and BBH (Suzgun et al., 2022). The details of all tasks are shown in Appendix A.1.

For evaluation on $\mathcal{S}_{\mathrm{NLG}}$, we report the zero-shot performance by directly evaluating the instruction-following responses using the **LLM-as-judge** metric[7]. We use the same evaluation prompt in Gu et al. (2023) to instruct the ChatGPT to judge the usefulness of model responses. Notably, for each query in $\mathcal{S}_{\mathrm{NLG}}$, we set the maximum number of output tokens as 256. As for $\mathcal{S}_{\mathrm{NLU}}$, we follow Chen et al. (2023) and use the code provided by Chia et al. (2023) to conduct benchmark evaluation. Specifically, we use 5-shot direct prompting and measure the exact-match score for MMLU (Hendrycks et al., 2020). Regarding the Drop (Dua et al., 2019) and BBH (Suzgun et al., 2022), 3-shot direct prompting is used and exact-match scores are reported.

Models. We evaluate ATKD on three types of LMs with various sizes: OPT (Zhang et al., 2022) (*student*: 125M, *teachers*: 350M, 1.3B, 2.7B, 6.7B), Pythia (Biderman et al., 2023) (*student*: 410M, *teachers*: 1.4B, 2.8B), and LLaMA (*student*: 68M (Miao et al., 2023), *teachers*: 1.1B (Zhang et al., 2024), 7B (Touvron et al., 2023)). Alpaca-GPT4 (Peng et al., 2023a) consisting of 52K GPT4-generated instruction-response pairs is used as training data. For teachers, we train each model with a batch size of 128 and a peak learning rate of 2e-5. For distilling students, the learning rate is selected in {2e-4, 2e-5} depending on model sizes, while the batch size is 256 and the maximum tokenizer length is 512. All models are trained for 3 epochs, and all experiments are conducted on 8 NVIDIA A800 (80GB) GPUs.

Baselines. We consider 5 cutting-edge KD baselines in our main experiment: Supervised KD (Hinton et al., 2015), Reverse KD (Gu et al., 2023), ImitKD (Lin et al., 2020), f-distill (Wen et al., 2023) and GKD (Agarwal et al., 2024). For reference, we also report the performance of teachers as the upper bound. We use the codebase of Liu et al. (2023) to implement these baselines and distill students.

4.2 Compared Results

Results of distilled models are shown in Table 2 and 3. For ease of illustration, we only report the overall performance of $\mathcal{S}_{\mathrm{NLG}}$ and $\mathcal{S}_{\mathrm{NLU}}$, respectively, where the detailed results are listed in Table 7 and 10. From these results, we can find that:

ATKD effectively alleviates the problem of performance degrades in larger teachers. As seen, various baseline KD methods suffer from this problem, *e.g.*, distilling OPT using GKD (*1.3B: 40.00% v.s. 6.7B: 38.73%*). However, with the help of our ATKD, the students can generally achieve better performance in larger teachers among various baseline KD methods, *i.e.*, alleviating the problem. These results can prove the effectiveness of ATKD in improving the quality of teaching.

ATKD brings consistent and significant performance gains among all model sizes and types. From Table 2, we can see that, compared with the baseline methods, our ATKD consistently achieves

[6] It should be noted that we do not finely adjust it for different datasets and tasks, but we still achieve good performance consistently. The analysis of λ is shown in §4.3.

[7] Although some studies show that LLM-as-Judge may exhibit a certain degree of bias (Zhao et al., 2023; Sottana et al., 2023), powerful LLMs, *e.g.*, ChatGPT and GPT-4, are capable of making preference determinations that are highly consistent with those of human annotators (Dubois et al., 2023).

Method	OPT-350M			OPT-1.3B			OPT-2.7B			OPT-6.7B		
	$\mathcal{S}_{\text{NLG}}$	$\mathcal{S}_{\text{NLU}}$	Avg.	$\mathcal{S}_{\text{NLG}}$	$\mathcal{S}_{\text{NLU}}$	Avg.	$\mathcal{S}_{\text{NLG}}$	$\mathcal{S}_{\text{NLU}}$	Avg.	$\mathcal{S}_{\text{NLG}}$	$\mathcal{S}_{\text{NLU}}$	Avg.
Teacher	58.33	20.36	39.35	68.90	22.60	45.75	74.21	22.28	48.25	78.71	23.43	51.07
Supervised KD	50.62	18.88	34.75	55.57	17.99	36.78	55.30	18.69	37.00	55.45	18.33	36.89
+ATKD	52.16	19.58	35.87	56.76	19.73	38.25	57.26	19.48	38.37	57.56	19.31	38.43
Δ (↑)	+1.54	+0.69	**+1.12**	+1.20	+1.74	**+1.47**	+1.96	+0.78	**+1.37**	+2.11	+0.98	**+1.54**
Reverse KD	50.54	18.05	34.30	51.60	18.15	34.87	51.26	18.56	34.91	50.08	18.33	34.20
+ATKD	50.86	19.13	34.99	54.40	19.40	36.90	54.34	19.27	36.80	54.37	19.16	36.76
Δ (↑)	+0.32	+1.08	**+0.70**	+2.80	+1.25	**+2.03**	+3.08	+0.70	**+1.89**	+4.29	+0.83	**+2.56**
ImitKD	52.27	18.35	35.31	59.87	18.41	39.14	59.88	17.46	38.67	58.86	17.28	38.07
+ATKD	52.36	18.66	35.51	60.76	19.29	40.02	60.77	19.18	39.97	62.66	19.56	41.11
Δ (↑)	+0.09	+0.31	**+0.20**	+0.89	+0.88	**+0.88**	+0.89	+1.71	**+1.30**	+3.80	+2.28	**+3.04**
f-distill	52.18	18.57	35.37	59.74	19.46	39.60	60.01	17.08	38.55	59.02	17.80	38.41
+ATKD	52.69	18.80	35.75	61.30	19.54	40.42	60.70	19.02	39.86	61.25	19.18	40.22
Δ (↑)	+0.51	+0.23	**+0.37**	+1.55	+0.08	**+0.82**	+0.68	+1.94	**+1.31**	+2.23	+1.38	**+1.80**
GKD	51.87	17.32	34.59	61.23	18.77	40.00	61.24	17.48	39.36	60.59	16.87	38.73
+ATKD	51.90	18.52	35.21	61.36	19.07	40.21	62.46	19.21	40.84	62.62	19.26	40.94
Δ (↑)	+0.04	+1.20	**+0.62**	+0.13	+0.30	**+0.21**	+1.22	+1.73	**+1.48**	+2.03	+2.39	**+2.21**

Table 2: **Results (%) of students (OPT-125M) distilling with different teachers and KD methods**. "Avg." means the average performance of $\mathcal{S}_{\text{NLG}}$ and $\mathcal{S}_{\text{NLU}}$. "Δ (↑)" denotes the performance gains of ATKD against the baselines. We see that our ATKD 1) brings consistent and significant performance gains and 2) effectively alleviates the problem of performance degrades in larger teachers.

better performance (up to **+3.04%** average gains) across various model sizes. Moreover, as seen in Table 3, in addition to OPT, ATKD also works well in Pythia-family and LLaMA-family models. These results demonstrate the universality of our ATKD and indicate that ATKD has great potential to expand to more LMs.

ATKD is beneficial to various baseline KD methods. In the preliminary analyses, we only conducted experiments on the typical Supervised KD. Here, we additionally investigate the combinability of ATKD and other baseline KD methods. As observed in Table 2, ATKD can bring consistent performance gains among all baseline KD methods. For example, with the help of ATKD, Revere KD and ImitKD achieve +1.80% and +1.36% average performance gains, respectively.

Method	Pythia-410M		LLaMA-68M	
	1.4B	2.8B	1.1B	7B
Teacher	67.86	73.50	75.23	84.17
Supervised KD	60.66	59.91	30.06	27.94
+ATKD	61.81	61.22	31.19	30.19
Δ (↑)	**+1.15**	**+1.31**	**+1.13**	**+2.25**
Reverse KD	55.92	54.67	26.15	25.94
+ATKD	57.05	57.94	26.73	26.99
Δ (↑)	**+1.14**	**+3.27**	**+0.58**	**+1.05**

Table 3: **Results (%) of students (Pythia-410M and LLaMA-68M)**. Due to the space limitation, we only report the results upon two typical KD baselines.

4.3 Ablation Study

Here, we 1) first evaluate the impact of ratio k, and 2) then investigate the effect of coefficient λ. Notably, we use the Supervised KD as the baseline and report the performance of OPT-125M on $\mathcal{S}_{\text{NLG}}$ tasks in this part.

Impact of ratio k. The ratio k that is used to select the hard-to-learn tokens, is an important hyper-parameter in ATKD. In this study, we analyze its influence by evaluating the performance with different k spanning from 0% to 100% at 10% intervals on $\mathcal{S}_{\text{NLG}}$ tasks. Figure 5 **(a)** illustrates the average results, in which we can find that: 1) Too large k values (*e.g.*, 70%) lead to performance degradation, as many of the selected tokens are "false" hard-to-learn and might distort the adaptive teaching. 2) The model's performance stably increases between 10% and 50%, and ATKD performs best with $k = 50\%$, thus leaving as our default settings.

Impact of coefficient λ. The factor λ in Eq. 5, which is used to balance different objectives, is also needed to be investigated. Figure 5 **(b)** illustrates the results of varied λ ranging from 0 to 1. As seen, compared to the single learning of hard-to-learn to-

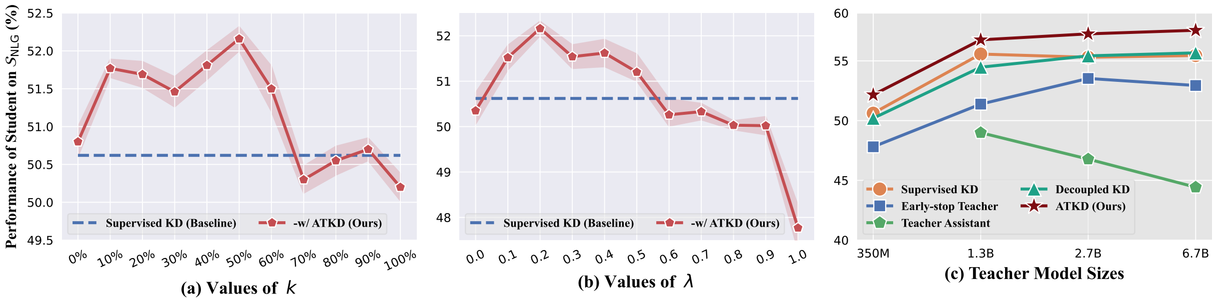

Figure 5: **(a) Effect of different ratios** (top-k) for selecting hard-to-learn tokens, **(b) Parameter analysis of** α in Eq. 5, and **(c) Comparison of different KD methods** that aim to alleviate the problem of performance degrades in larger teachers. We use the Supervised KD as the baseline and report the performance of OPT-125M on $\mathcal{S}_{\text{NLG}}$.

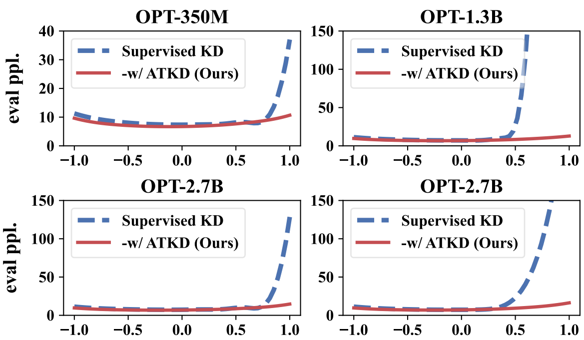

Figure 6: **1D visualization of loss landscapes of OPT-125M** distilled by different methods and teachers. The y-axis denotes the model perplexity on VicunaEval. We see that ATKD effectively smooths the loss landscape.

kens, incorporating some supervision signals from easy-to-learn tokens results in better performance. However, too large λ values (*e.g.*, 0.9) would be harmful to the effectiveness of ATKD, as paying much attention to the learning of easy-to-learn tokens might lead to overfitting. More specifically, the case of $\lambda = 0.2$ performs best, and we thereby use this setting in our experiments.

4.4 Discussion

Here, we conduct further analyses to discuss: 1) whether ATKD outperforms the other counterparts, and 2) whether it gains better model generalization.

Comparison with other counterparts. To the best of our knowledge, there are no existing KD methods that involve solving the problem of performance degradation for autoregressive LLMs. Thus, we compare ATKD with the related methods in the vision community: "Early-stop Teacher" (Cho and Hariharan, 2019), "Teacher Assistant"[8] (Mirzadeh et al., 2020) and "Decoupled KD" (Zhao et al., 2022). The contrastive results are illustrated in Figure 5 **(c)**, from which we can find that: 1) Suppressing the teacher's performance via early stopping or leveraging a smaller assistant might not be effective and even lead to worse performance, 2) Although "Decoupled KD" could alleviate this problem, it achieves sub-optimal performance, as it equally adopts the same teaching modes for all tokens. Takeaway: ***among all methods, our ATKD can not only alleviate this problem but also bring further performance gains in a simple manner, proving its superiority.***

Model Generalization. Enforcing the student to learn more diverse knowledge could improve its generalization. To verify this conjecture, we visualize the loss landscapes of different distilled OPT-125M models on the VicunaEval task. In practice, we follow He et al. (2021); Zhong et al. (2022) to plot the 1D loss curve by linear interpolation between the model weights before (denoted as θ_0) and after (denoted as θ_1) distilling, *i.e.*, "$\theta_1 + \beta \cdot (\theta_1 - \theta_0)$", where β is a scalar parameter that is ranged from -1 to 1. The 1D visualization results are illustrated in Figure 6, and we find that "-w/ ATKD (Ours)" shows a flatter and optimal property against the baseline Supervised KD. Takeaway: ***These results prove that ATKD can smooth the loss landscape and improve the model generalization effectively.***

5 Related Works

Recently, autoregressive LMs (OpenAI, 2023; Chowdhery et al., 2023; Touvron et al., 2023) have shown their superior performance by solving various NLP tasks in a generative manner. Despite their success, they usually suffer from unbearable inference latency (Leviathan et al., 2023). To this end, several model compression approaches are

[8]We use the OPT-350M as the assistant model and only report the results distilling from teachers larger than 350M.

proposed to reduce the model size and accelerate the inference (Hinton et al., 2015; Jaszczur et al., 2021; Zhu et al., 2023; Chen et al., 2024). Among these efforts, KD strategy (Hinton et al., 2015), which aims at training a smaller student model with the guidance of a teacher model, has attracted great attention recently (Ding et al., 2021; Wen et al., 2023; Gu et al., 2023; Agarwal et al., 2024). Although these KD methods realize promising performance when distilling (relatively) smaller LMs, they might fall short in distilling larger LMs (*e.g.*, OPT-6.7B) especially when the student is of a small scale. In fact, this phenomenon has been observed in the vision community (Mirzadeh et al., 2020; Cho and Hariharan, 2019) and language understanding models (Zhang et al., 2023). To alleviate this problem, a few studies including teacher assistant-based (Mirzadeh et al., 2020) and student-friendly (Cho and Hariharan, 2019; Zhao et al., 2022; Zhang et al., 2023) distillation have been recently explored.

The above efforts are generally used for vision models or discriminative LMs, while the autoregressive KD for generative LMs is yet to be explored. To the best of our knowledge, we are the (nearly) first to alleviate the problem of performance degradation in larger autoregressive teacher LMs. Different from the previous methods that aim to directly bridge the performance gap between teacher and student, we attempt to improve the quality of teaching by exploring and addressing the limitations of existing KD objectives.

6 Conclusion

In this paper, we reveal and address the limitations of KD in compressing the larger autoregressive teachers. Based on a series of preliminary analyses, we find that equally adopting the same teaching modes for all tokens is sub-optimal, as learning more target-oriented knowledge of the easy-to-learn tokens might lead to overfitting and result in poor performance. To address these limitations, we improve KD with a novel adaptive teaching algorithm. It skips the target-oriented teaching for easy-to-learn tokens and pays more attention to the diverse learning of hard-to-learn tokens. Experiments show that our approach consistently and significantly improves distillation performance across all model architectures. In-depth analyses prove that our approach indeed alleviates the problem, and further improves the model generalization.

Limitations

Our work has several potential limitations. First, given the limited computational budget, we only validate our ATKD on up to 7B autoregressive LMs in the main experiments. Although the extra analysis in Appendx A.3 shows that ATKD has the great potential to work well in distilling larger teachers, it will be more convincing if scaling up to super-large model size (*e.g.*, 70B) and applying ATKD to more cutting-edge model architectures. On the other hand, besides the distillation performance, we believe that there are still other properties, *e.g.*, training efficiency and model robustness, of LMs that can be improved by our ATKD approach, which are not fully explored in this work.

Ethics and Reproducibility Statements

Ethics We take ethical considerations very seriously and strictly adhere to the ACL Ethics Policy. This paper proposes an adaptive teaching algorithm to improve existing KD strategies. It aims to compress the existing larger LMs into smaller students, instead of encouraging them to learn privacy knowledge that may cause the ethical problem. Moreover, all training and evaluation datasets used in this paper are publicly available and have been widely adopted by researchers. Thus, we believe that this research will not pose ethical issues.

Reproducibility In this paper, we discuss the detailed experimental setup, such as hyper-parameters and statistic descriptions. More importantly, we will publicly release our code in `https://github.com/WHU-ZQH/ATKD` to help reproduce the experimental results of this paper.

Acknowledgements

We are grateful to the anonymous reviewers and the area chair for their insightful comments and suggestions. This work was supported in part by the National Natural Science Foundation of China under Grant 623B2076, U23B2048, 62076186 and 62225113, in part by the National Key Research and Development Program of China under Grant 2023YFC2705700, and in part by the Innovative Research Group Project of Hubei Province under Grant 2024AFA017. The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

References

Rishabh Agarwal, Nino Vieillard, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillaiton of language models: Learning from self-generated mistakes. In *ICLR*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *ICML*.

Hong Chen, Chengtao Lv, Liang Ding, Haotong Qin, Xiabin Zhou, Yifu Ding, Xuebo Liu, Min Zhang, Jinyang Guo, Xianglong Liu, et al. 2024. Db-llm: Accurate dual-binarization for efficient llms. *arXiv preprint*.

Kehai Chen, Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2020. Content word aware neural machine translation. In *ACL*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint*.

Yew Ken Chia, Pengfei Hong, Lidong Bing, and Soujanya Poria. 2023. Instructeval: Towards holistic evaluation of instruction-tuned large language models. *arXiv preprint*.

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.

Jang Hyun Cho and Bharath Hariharan. 2019. On the efficacy of knowledge distillation. In *ICCV*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*.

Kenneth Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F Wong, Dacheng Tao, and Zhaopeng Tu. 2021. Understanding and improving lexical choice in non-autoregressive translation. In *ICLR*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *NAACL-HLT*.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint*.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. Koala: A dialogue model for academic research. *Blog post, April*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Knowledge distillation of large language models. *arXiv preprint*.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *arXiv preprint*.

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. 2021. On the effectiveness of adapter-based tuning for pretrained language model adaptation. In *ACL*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *ICLR*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint*.

Sebastian Jaszczur, Aakanksha Chowdhery, Afroz Mohiuddin, Lukasz Kaiser, Wojciech Gajewski, Henryk Michalewski, and Jonni Kanerva. 2021. Sparse is enough in scaling transformers. *NeurIPS*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *EMNLP*.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *ICML*.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. 2020. Autoregressive knowledge distillation through imitation learning. In *EMNLP*.

Xiaoxuan Liu, Lanxiang Hu, Peter Bailis, Ion Stoica, Zhijie Deng, Alvin Cheung, and Hao Zhang. 2023. Online speculative decoding. *arXiv preprint*.

Qingyu Lu, Baopu Qiu, Liang Ding, Kanjian Zhang, Tom Kocmi, and Dacheng Tao. 2023. Error analysis prompting enables human-like translation evaluation in large language models: A case study on chatgpt. *arXiv preprint*.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Rae Ying Yee Wong, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. 2023. Specinfer: Accelerating generative llm serving with speculative inference and token tree verification. *arXiv preprint*.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. In *AAAI*.

OpenAI. 2023. Gpt-4 technical report.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023a. Instruction tuning with gpt-4. *arXiv preprint*.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023b. Towards making the most of chatgpt for machine translation. In *Findings of EMNLP*.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2020. Green ai. *Communications of the ACM*.

Andrea Sottana, Bin Liang, Kai Zou, and Zheng Yuan. 2023. Evaluation metrics in the era of GPT-4: reliably evaluating large language models on sequence to sequence tasks. *arXiv preprint*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint*.

Kelvin HK Tan, Charlene Tan, and Jude SM Chua. 2008. Innovation in education: The" teach less, learn more" initiative in singapore schools. *Innovation in education*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint*.

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

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint*.

Chen Zhang, Yang Yang, Jiahao Liu, Jingang Wang, Yunsen Xian, Benyou Wang, and Dawei Song. 2023. Lifting the curse of capacity gap in distilling language models. In *ACL*.

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

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*.

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. 2022. Decoupled knowledge distillation. In *CVPR*.

Jiaxu Zhao, Meng Fang, Shirui Pan, Wenpeng Yin, and Mykola Pechenizkiy. 2023. GPTBIAS: A comprehensive framework for evaluating bias in large language models. *arXiv preprint*.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. *arXiv preprint*.

Qihuang Zhong, Liang Ding, Li Shen, Peng Mi, Juhua Liu, Bo Du, and Dacheng Tao. 2022. Improving sharpness-aware minimization with fisher mask for better generalization on language models. In *Findings of EMNLP*.

Miaoxi Zhu, Qihuang Zhong, Li Shen, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Zero-shot sharpness-aware quantization for pre-trained language models. In *EMNLP*.

A Appendix

A.1 Details of Tasks and Datasets

In this work, we conduct extensive experiments on several language generation and understanding tasks. Here, we introduce the descriptions of these tasks and datasets in detail. Firstly, we present the statistics of all evaluated datasets in Table 4. Then, each task is described as:

DollyEval. DollyEval (Gu et al., 2023) is a 500-sample test set that is splitted from the databricks-dolly-15k[9] dataset.

VicunaEval. VicunaEval (Chiang et al., 2023) contains 80 challenging questions used in the Vicuna evaluation.

SelfInst. SelfInst (Wang et al., 2022) is a user-oriented instruction-following test set with 252 samples.

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

Test set	Task	# Types	# Samples
$\mathcal{S}_{\text{NLG}}$	DollyEval	Generation	500
	VicunaEval	Generation	80
	SelfInst	Generation	242
	Koala	Generation	180
	WizardLM	Generation	218
$\mathcal{S}_{\text{NLU}}$	MMLU	Classification	14,079
	Drop	Classification	9,540
	BBH	Classification	6,511

Table 4: **Statistics of all test sets** used in this paper.

Evaluator	Method	350M	1.3B	2.7B	6.7B
ChatGPT	Supervised KD	46.93	51.92	53.02	53.78
	+ATKD	**52.75**	**52.99**	**53.69**	**54.74**
GPT-4	Supervised KD	30.09	32.28	32.45	33.28
	+ATKD	**32.48**	**33.21**	**33.53**	**34.49**

Table 5: **Comparison between ChatGPT-based and GPT-4-based automatic evaluators**. Here, we report the evaluation results of students (OPT-125M) on the Koala benchmark, and we can see that ChatGPT makes similar judgments to GPT-4.

Koala. This test set consists of 180 queries that Geng et al. (2023) source from publicly available user-written language model prompts.

WizardLM. WizardLM (Xu et al., 2023) consists of 218 instances, each of which is an instruction for a specific skill, such as Math, Reasoning, Complex Formats, and so on.

MMLU. Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2020) is a popular benchmark designed to measure the multitask accuracy of LLMs, covering 57 tasks.

Drop. Discrete Reasoning Over Paragraphs (DROP) (Dua et al., 2019) is a math-based reading comprehension task that requires a system to perform discrete reasoning over passages extracted from Wikipedia articles.

BBH. BIG-Bench Hard (BBH) (Suzgun et al., 2022) is a subset of 23 challenging tasks from the BIG-Bench benchmark (Srivastava et al., 2023), which focuses on tasks believed to be beyond the capabilities of current language models.

A.2 ChatGPT *v.s.* GPT-4

Although the GPT-4 is more commonly used as the automatic evaluator for the "LLM-as-Judge" metric (Chen et al., 2023; Chiang et al., 2023), it requires a much higher cost, especially for our extensive experiments. As an alternative, we use the cheaper ChatGPT as the automatic evaluator to evaluate the model responses. Here, to verify whether ChatGPT is enough to reflect the behavior of LMs, we conduct a comparative study on ChatGPT and GPT-4. Specifically, taking the responses of OPT-125M on Koala as an example, we use the ChatGPT and GPT-4 to measure the score, respectively. As listed in Table 5, GPT-4 seems to be more strict in evaluating the model responses, as the evaluated scores of GPT-4 are generally lower than those of ChatGPT. Nevertheless, both automatic evaluators make similar judgments, *i.e.*, our ATKD performs better than baselines among all model sizes. Thus, we believe that *ChatGPT is enough to reflect whether the model generates a useful response, and it is credible to use ChatGPT as the automatic evaluator in this study.*

Method	Dolly	Vicuna	SelfInst	WizardLM
Distilling from larger teacher, OPT-13B				
Supervised KD	58.59	48.88	55.63	42.44
+ATKD	**62.02**	**56.07**	**60.16**	**48.37**
Δ ($\uparrow$)	+3.43	+7.19	+4.53	+5.93

Table 6: **Results of student (OPT-125M) distilling from larger teacher (OPT-13B)**. Here, we report several tasks from the $\mathcal{S}_{\text{NLG}}$ set. We can find that our ATKD works well in distilling larger models.

A.3 Whether Our Method Works Well in Distilling Larger Models.

To verify whether our method works well in the larger model settings, we conduct additional experiments using the OPT-13B teacher model. We apply the Supervised KD and our ATKD methods to distill the OPT-13B into the OPT-125M student model. Evaluation results on several $\mathcal{S}_{\text{NLG}}$ tasks are listed in Table 6, where we use the LLM-as-a-Judge as the metric. As seen, when using the OPT-13B model, the Supervised KD still suffers from the problem of performance degradation, as the distilled student model performs much worse than those of smaller teacher models in Table 7. Conversely, our ATKD can effectively alleviate this problem and achieve much better performance (*i.e.*, up to +7.19 on the VicunaEval dataset) than the baseline Supervised KD. These results indicate that our ATKD has the great potential to expand to super-large-scale model scenarios.

Method	$\mathcal{S}_{\text{NLG}}$					$\mathcal{S}_{\text{NLU}}$			Average	
	DollyEval	VicunaEval	SelfInst	Koala	WizardLM	MMLU	Drop	BBH	$\mathcal{S}_{\text{NLG}}$	$\mathcal{S}_{\text{NLU}}$
SFT -w/o KD	55.05	38.45	52.52	45.27	42.35	21.66	3.8	27.5	49.75	17.65
Teacher-OPT-350M	64.96	51.09	61.98	52.13	46.84	26.03	6.98	28.08	58.33	20.36
Supervised KD	54.90	45.93	53.86	46.93	41.98	24.44	4.85	27.36	50.62	18.88
+**ATKD**	56.14	43.35	52.98	52.75	44.88	24.52	6.94	27.27	**52.16**	**19.58**
Reverse KD	55.53	44.30	50.25	50.14	42.04	22.54	4.88	26.73	50.54	18.05
+**ATKD**	54.81	43.68	52.66	51.05	42.26	23.74	6.95	26.70	**50.86**	**19.13**
ImitKD	55.68	41.30	54.73	52.51	45.55	24.06	4.25	26.73	52.27	18.35
+**ATKD**	55.10	43.64	55.37	52.49	45.84	25.07	4.39	26.51	**52.36**	**18.66**
f-distill	56.31	43.52	52.67	52.69	44.93	24.71	4.50	26.49	52.18	18.57
+**ATKD**	54.86	42.61	57.22	53.00	46.15	24.60	5.04	26.76	**52.69**	**18.80**
GKD	53.76	44.41	53.82	54.62	45.83	23.93	1.42	26.61	51.87	17.32
+**ATKD**	54.43	44.35	53.88	54.79	44.31	25.40	2.29	27.88	**51.90**	**18.52**
Teacher-OPT-1.3B	72.29	68.86	74.35	65.02	58.30	24.78	14.00	29.01	68.90	22.60
Supervised KD	60.89	52.35	57.95	51.92	44.92	22.27	4.57	27.13	55.57	17.99
+**ATKD**	62.35	51.52	59.59	52.99	45.86	25.08	6.43	27.67	**56.76**	**19.73**
Reverse KD	57.16	46.36	50.75	50.10	42.94	23.02	4.22	27.21	51.60	18.15
+**ATKD**	59.08	48.41	57.17	52.04	44.71	26.06	5.44	26.71	**54.40**	**19.40**
ImitKD	64.55	50.74	61.99	59.15	50.73	23.45	4.31	27.47	59.87	18.41
+**ATKD**	65.27	53.70	63.41	60.00	50.70	25.76	4.90	27.20	**60.76**	**19.29**
f-distill	64.80	51.45	61.57	59.00	49.78	26.59	4.71	27.08	59.74	19.46
+**ATKD**	65.72	51.56	62.96	60.72	53.35	26.58	4.84	27.21	**61.30**	**19.54**
GKD	63.48	56.08	64.73	61.54	53.83	25.99	4.42	25.89	61.23	18.77
+**ATKD**	64.84	56.75	64.43	60.66	52.25	25.69	4.69	26.82	**61.36**	**19.07**
Teacher-OPT-2.7B	75.64	74.43	80.99	74.12	63.39	24.74	12.86	29.25	74.21	22.28
Supervised KD	59.16	52.89	58.31	53.02	45.88	22.89	5.63	27.56	55.30	18.69
+**ATKD**	62.47	54.47	60.22	53.69	46.01	23.83	6.48	28.12	**57.26**	**19.48**
Reverse KD	56.09	48.58	49.46	51.07	43.34	24.08	4.23	27.38	51.26	18.56
+**ATKD**	59.79	50.96	55.73	50.70	44.54	24.65	5.76	27.39	**54.34**	**19.27**
ImitKD	63.30	57.55	62.98	59.23	50.01	22.82	4.50	25.07	59.88	17.46
+**ATKD**	65.04	57.27	63.11	59.93	50.37	25.11	6.17	26.25	**60.77**	**19.18**
f-distill	63.78	58.58	62.79	58.57	50.00	22.21	4.40	24.63	60.01	17.08
+**ATKD**	64.45	57.00	63.03	59.92	51.49	24.57	5.33	27.17	**60.70**	**19.02**
GKD	64.13	57.42	64.41	63.59	50.56	22.78	3.42	26.24	61.24	17.48
+**ATKD**	66.84	60.73	63.23	63.02	51.72	25.42	4.57	27.65	**62.46**	**19.21**
Teacher-OPT-6.7B	81.03	77.38	84.92	78.65	67.01	24.67	15.16	30.45	78.71	23.43
Supervised KD	60.01	49.41	58.22	53.78	45.51	23.46	5.43	26.10	55.45	18.33
+**ATKD**	63.08	53.75	60.05	54.74	45.84	24.23	5.95	27.74	**57.56**	**19.31**
Reverse KD	53.73	47.33	49.70	49.50	43.61	23.95	4.30	26.73	50.08	18.33
+**ATKD**	59.13	52.24	57.63	52.21	42.38	25.62	4.80	27.05	**54.37**	**19.16**
ImitKD	62.32	57.64	63.02	57.08	48.24	22.59	4.02	25.23	58.86	17.28
+**ATKD**	65.07	58.07	65.93	63.76	54.29	25.89	6.68	26.11	**62.66**	**19.56**
f-distill	63.25	55.97	62.06	57.23	48.56	24.25	4.03	25.12	59.02	17.80
+**ATKD**	64.51	59.48	64.04	62.28	50.48	25.15	5.57	26.82	**61.25**	**19.18**
GKD	64.37	58.47	61.63	62.19	50.23	22.03	3.53	25.04	60.59	16.87
+**ATKD**	66.68	60.87	65.29	63.19	50.51	25.84	4.36	27.58	**62.62**	**19.26**

Table 7: **Full results of Table 2, *i.e.*, performance of student (OPT-125M) on $\mathcal{S}_{\text{NLG}}$ and $\mathcal{S}_{\text{NLU}}$ across different teachers and KD methods.** "Average" denotes the average results of $\mathcal{S}_{\text{NLG}}$ and $\mathcal{S}_{\text{NLU}}$, and "SFT -w/o KD" refers to the results of the vanilla student that is tuned on the ground-truth data. Better results among baseline KD methods and ours are in **bold**.

Method	$\mathcal{S}_{\text{NLG}}$, **Pythia-410M**				
	Dolly	Vicuna	SelfInst	Koala	WizardLM
SFT-w/o KD	61.81	57.38	60.62	50.67	50.06
Teacher-1.4B/-1.1B	69.51	73.13	69.59	65.17	62.44
Supervised KD	62.62	64.61	63.47	56.36	55.15
+**ATKD** (Ours)	63.87	62.70	65.64	59.13	54.72
Reverse KD	58.82	56.17	58.87	53.17	48.15
+**ATKD** (Ours)	61.14	57.80	57.30	53.85	49.77
Teacher-2.8B/-7B	75.84	76.63	72.99	70.95	69.63
Supervised KD	61.10	60.38	63.51	56.99	55.43
+**ATKD** (Ours)	63.37	64.31	63.06	59.22	54.79
Reverse KD	58.80	54.99	53.74	52.61	47.79
+**ATKD** (Ours)	61.21	63.23	56.38	57.23	50.81

Table 8: Results of Pythia-410M.

$\mathcal{S}_{\text{NLG}}$, **LLaMA-68M**				
Dolly	Vicuna	SelfInst	Koala	WizardLM
26.37	26.67	28.37	27.27	23.72
78.82	77.50	75.02	72.01	69.08
29.63	28.74	31.51	31.84	28.45
30.29	29.50	34.65	33.25	28.34
25.69	25.74	27.98	29.23	22.78
26.02	25.31	28.95	29.23	24.35
86.50	83.25	83.70	84.18	79.68
27.27	28.31	29.26	30.02	26.15
30.08	28.95	30.97	32.09	28.47
25.65	25.11	27.85	28.03	23.05
26.70	27.38	28.35	29.53	23.91

Table 9: Results of LLaMA-68M.

Table 10: **Full results of Table 3, *i.e.*, performance of students (Pythia-410M, Table 8 and LLaMA-68M, Table 9) on $\mathcal{S}_{\text{NLG}}$.** Notably, for Pythia-410M, we use the Pythia-1.4B/2.8B as teachers, while LLaMA-1.1B/7B are used as teachers for LLaMA-68M.