DistiLLM: Towards Streamlined Distillation for Large Language Models

Jongwoo Ko[1] **Sungnyun Kim**[1] **Tianyi Chen**[2] **Se-Young Yun**[1]
https://github.com/jongwooko/distillm

Abstract

Knowledge distillation (KD) is widely used for compressing a teacher model to a smaller student model, reducing its inference cost and memory footprint while preserving model capabilities. However, current KD methods for auto-regressive sequence models (*e.g.*, large language models) suffer from missing a standardized objective function. Moreover, the recent use of student-generated outputs to address training-inference mismatches has significantly escalated computational costs. To tackle these issues, we introduce **DistiLLM**, a more effective and efficient KD framework for auto-regressive language models. **DistiLLM** comprises two components: (1) a novel skew Kullback-Leibler divergence loss, where we unveil and leverage its theoretical properties, and (2) an adaptive off-policy approach designed to enhance the efficiency in utilizing student-generated outputs. Extensive experiments, including instruction-following tasks, demonstrate the effectiveness of **DistiLLM** in building high-performing student models while achieving up to 4.3× speedup compared to recent KD methods.

1. Introduction

Recent advancements in auto-regressive language models (LMs, OpenAI 2023; Touvron et al. 2023) such as large language models (LLMs) have significantly improved the quality of text generation in a variety of generative tasks such as task-agnostic instruction-following tasks (Wang et al., 2023b). This success is often attributed to the increased scale of training data and model parameters (*e.g.*, 175B parameters for GPT-3; Brown et al. 2020). However, expanding the parameter count brings associated costs, limiting the deployment of these models due to either high inference costs or large memory footprints. Therefore, a crucial objective for the practical use of these high-capacity models is to compress them by reducing the number of parameters while preserving their performance to the greatest extent possible.

As the necessity of reducing the demands on computational resources becomes important, KD (Hinton et al., 2015) emerges as a promising method. It involves the transfer of knowledge from a teacher model with large capacity (*i.e.,* LLMs) to a smaller student LM. Most approaches of KD have employed the Kullback-Leibler divergence (KLD) loss, enforcing the student model to generate outputs that mirror the teacher model's outputs on a fixed dataset (Kim & Rush, 2016; Sanh et al., 2019). Such methods of KD have significantly enhanced the performance of student models, making them competitive with teacher models while increasing efficiency, especially for various classification tasks (Sun et al., 2019; Mirzadeh et al., 2020).

These approaches using KLD on fixed datasets, however, have two primary shortcomings in applying to auto-regressive LMs. First, using the KLD can lead to sub-optimal results. Given the complexity of generative tasks compared to classification tasks, this can result in the student distribution becoming overly smooth and consequently failing to fully capture the teacher distribution, or becoming overly concentrated in high-probability distributions. This issue—referred to as *mode averaging* or *mode collapse*—arises due to the asymmetric nature of the KLD (Wen et al., 2023; Gu et al., 2024). Second, the use of a fixed dataset in the training phase can cause a distribution mismatch between the sequences observed during training and those generated by the student in auto-regressive inference, leading to exposure bias problems (Arora et al., 2022).

Recent studies have explored various divergence losses (Wen et al., 2023; Agarwal et al., 2024) or the incorporation of student-generated output (SGO, Lin et al. 2020; Agarwal et al. 2024) to address the existing problems. However, these methods often lack standardized objective functions and are less efficient due to the continuous SGO generation. For instance, Agarwal et al. (2024) employed on-policy distillation with SGOs, but it faces low sample efficiency and high generation time as it constantly prompts the student to produce new training sequences. Also, their experiments in-

[1]KAIST AI, Seoul, Republic of Korea [2]Microsoft, Redmond, Washington, USA. Correspondence to: Se-Young Yun <yunseyoung@kaist.ac.kr>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

dicated the optimal divergence seems to be task-dependent, which requires additional efforts to inconveniently select a proper loss function. Gu et al. (2024) introduced a policy optimization method to minimize reverse KLD between student and teacher distributions, yet this also compromises training efficiency by requiring generation from both models in every iteration.

Contributions. In this paper, we introduce **DistiLLM**[1], featuring a novel *skew KLD loss* and an *adaptive off-policy approach*, focusing on both distillation effectiveness and training efficiency. We provide both theoretical and empirical evidence that the components of **DistiLLM** work well individually and synergistically with each other. Our detailed contributions include:

- **Skew KLD:** We focus on the two key issues of existing objective functions for auto-regressive LMs: instability from potential gradient explosions in optimizing the KLD loss, and lack of emphasis on generalizability and convergence. To address these limitations, we introduce skew KLD, a new objective function with a strong theoretical foundation, optimized for stable gradients and minimal approximation errors, empirically leading to faster convergence and superior performance.
- **Adaptive off-policy approach:** While using SGOs in KD is generally effective in improving performance, this approach significantly increases training time (Fig. 2) and makes it challenging to find the optimal proportion for using SGOs. To this end, we propose an *adaptive off-policy* approach module for adaptively and efficiently leveraging SGOs to consider the data perspective of KD.
- **Advanced performance and efficiency: DistiLLM** accomplishes state-of-the-art performances for the student LMs on various generative tasks (*e.g.,* instruction-following or text summarization), while achieving the $2.5 \sim 4.3\times$ training speedup compared to recent KD techniques (Gu et al., 2024; Agarwal et al., 2024).

2. Background

2.1. KD for Auto-regressive Generative LMs

We provide preliminary information on the KD for auto-regressive generative LMs. Given a source and target sequence pair, denoted as $(\mathbf{x}, \mathbf{y})$, KD minimizes divergence D between the distributions of a fixed teacher model $p(\mathbf{y}|\mathbf{x})$ and a parameterized student model $q_\theta(\mathbf{y}|\mathbf{x})$. The training data pairs $(\mathbf{x}, \mathbf{y})$ are either sampled from a fixed ground-truth dataset (Hinton et al., 2015) or from teacher-generated outputs (Kim & Rush, 2016).

Conventionally, KLD, denoted as D_{KL}, is the most widely used loss in KD due to its simplicity and tractability. The sequence-level distillation using the KLD is accurately decomposed into a sum of token-wise distillation:

$$D_{\text{KL}}(p, q_\theta) = \mathbb{E}_{\mathbf{x}}\mathbb{E}_{\mathbf{y}\sim p(\cdot|\mathbf{x})}\left[\log \frac{p(\mathbf{y}|\mathbf{x})}{q_\theta(\mathbf{y}|\mathbf{x})}\right] \tag{1}$$

$$\approx \frac{1}{|\mathcal{D}|}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} p(\mathbf{y}|\mathbf{x}) \log \frac{p(\mathbf{y}|\mathbf{x})}{q_\theta(\mathbf{y}|\mathbf{x})} \tag{2}$$

$$= \frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}_X}\sum_{t}^{|\mathbf{y}|}\sum_{y_t\in V} p(y_t|\mathbf{y}_{<t},\mathbf{x}) \log \frac{p(y_t|\mathbf{y}_{<t},\mathbf{x})}{q_\theta(y_t|\mathbf{y}_{<t},\mathbf{x})} \tag{3}$$

where V is the vocabulary token set and $\mathbf{y}_{<t} := (y_1, y_2, \ldots, y_{t-1})$ represents the sequence of tokens up to index $t-1$. We focus solely on tractable KLD, as other divergences like total variation distance (TVD, Wen et al. 2023) do not effectively decompose sequence-level distillation into token-level components. While the explicit definition of KLD is given in Eq. 1 (Kim & Rush, 2016; Wen et al., 2023), most recent studies, such as Agarwal et al. (2024) and Gu et al. (2024), approximate the distribution matching by minimizing Eq. 2, under the assumption that the teacher's distribution is similar to its training dataset $\mathcal{D}$. For the sake of training efficiency, our method utilizes the definition provided in Eq. 2, while Eq. 1 is used for theoretical analysis of our proposed distillation objective in Thm. 1.

2.2. Pitfalls of Existing Distillation

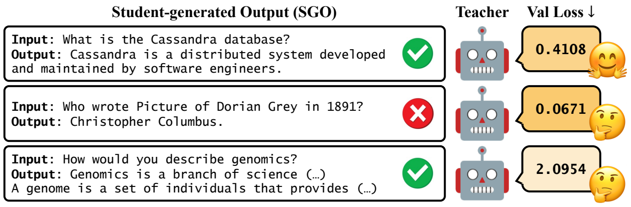
Limitation of objective functions. The KLD objective in KD, primarily due to its asymmetric nature (Wen et al., 2023), often forces the student distribution to cover the entire support set of the teacher distribution, leading to significant limitations. This becomes evident when a sampled data point is included in the teacher distribution's support but falls outside the student distribution, *i.e.*, $\exists(\mathbf{x}, \mathbf{y})$ such that $p(\mathbf{y}|\mathbf{x}) \gg 0$ and $q_\theta(\mathbf{y}|\mathbf{x}) \approx 0$. The limitation becomes pronounced if the student model lacks the capacity to match all support sets of the teacher distribution accurately. Consequently, this results in the student model exhibiting a mode-averaging problem, where it learns an overly smooth distribution in an attempt to cover the teacher's entire support set, as highlighted by recent studies (Wen et al., 2023; Gu et al., 2024).

Such recent studies have partially addressed this issue by applying the reverse KLD (RKLD, Gu et al. 2024; Agarwal et al. 2024), defined as $D_{\text{RKL}}(p, q_\theta) := D_{\text{KL}}(q_\theta, p)$, or generalized JSD (Agarwal et al., 2024) by introducing an interpolation parameter $\beta \in [0, 1]$, defined as

$$\begin{aligned} D^{(\beta)}_{\text{JSD}}(p, q_\theta) := {} & \beta\, D_{\text{KL}}(p, \beta p + (1-\beta)q_\theta) \\ & + (1-\beta)\, D_{\text{KL}}(q_\theta, \beta p + (1-\beta)q_\theta). \end{aligned} \tag{4}$$

These approaches have shown empirical success in auto-regressive LMs, but there is a need for systematic study to provide a standard distillation objective grounded in comprehensive theoretical and experimental analyses. The lack

[1] **DistiLLM** is pronounced as *distill-LLM*, merging the word "distill" with "LLM".



```
Input: What is the Cassandra database?
Output: Cassandra is a distributed system developed
and maintained by software engineers.

Input: Who wrote Picture of Dorian Grey in 1891?
Output: Christopher Columbus.

Input: How would you describe genomics?
Output: Genomics is a branch of science (…)
A genome is a set of individuals that provides (…)
```

Figure 1. Examples of SGOs from GPT-2 (student) and their corresponding validation loss by GPT-2 XL (teacher). Since the teacher model may not be familiar with the SGO, using $p(\mathbf{y}|\mathbf{x})$ as a target distribution can misguide the student model, as shown in Tab. 2.

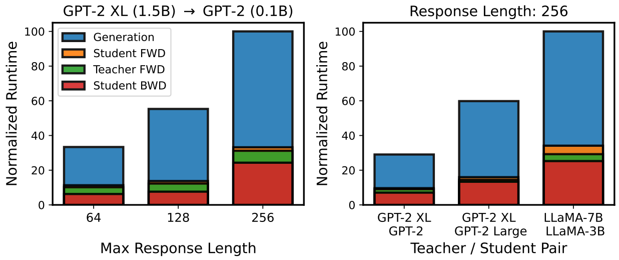


Figure 2. (**Left**): Normalized runtime according to the maximum response length of SGOs with GPT-2 XL teacher and GPT-2 student. (**Right**): Normalized runtime for various sizes of teacher and student models with a response length of 256. FWD and BWD denote forward and backward propagation, respectively.

of such backing for these recently proposed objective functions leads to sub-optimal performance and task-dependent variability (Agarwal et al., 2024).

Limitations of utilizing SGO. Previous KD methods for auto-regressive LMs have encountered a training-inference mismatch between the samples from fixed datasets that are used during training and those produced by the student model during inference. Recent studies (Lin et al., 2020; Agarwal et al., 2024) explored addressing this challenge by prompting the student model to generate SGOs and then training from the feedback of the teacher model on these sequences. This approach addresses the mismatch by training the student model on its familiar, self-generated sequences. These efforts have significantly improved the performance of distilling LLMs (Lin et al., 2020; Agarwal et al., 2024).

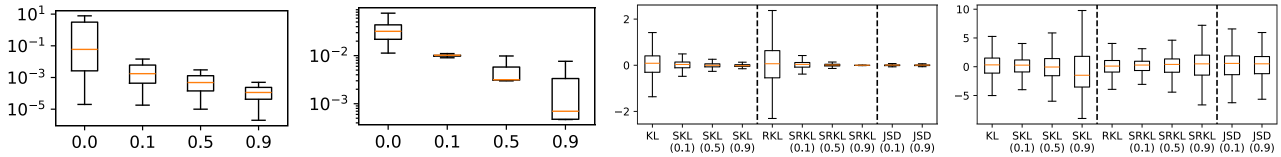
Despite its effectiveness, we identify two main issues with the current utilization of SGO. *First*, teacher models may experience a distribution mismatch between their training data and unfamiliar or inaccurate SGOs, potentially leading to misguidance on q_θ. As depicted in Fig. 1, such a mismatch can result in the teacher model assigning low validation loss to incorrect but shorter generations and high validation loss to longer but correct ones. *Second*, as shown in Fig. 2, generating SGOs for every iteration proves computationally inefficient. Across all experiments, irrespective of the maximum sequence length (ranging from 64 to 256) of SGOs or the model size (from GPT-2 to OpenLLaMA2-3B), the SGO generation accounts for a considerable portion of the total training time, reaching up to 80%.

Algorithm 1 Training pipeline of DistiLLM

```
Input: initial prob. $\phi$, student $q_{\theta_0}$ with parameters $\theta_0$, teacher $p$, total training iterations $T$, training & validation dataset $\mathcal{D}$, $\mathcal{D}_{val}$, empty replay buffer $\mathcal{D}_R$
Output: Student model $q_{\theta_T}$ with trained parameters $\theta_T$
while $t \leq T$ do
  Randomly sample $u \sim \text{Unif}(0,1)$
  /* Linearly Decreasing Replay Ratio */
  if $u < \lambda_R := \phi(1 - \frac{t}{T})$ then
    /* Generate SGO & Update $\mathcal{D}_R$ */
    Generate SGO $\{\tilde{\mathbf{y}}_i\}_{i=1}^B$ from $\{q_{\theta_t}(\cdot|\mathbf{x}_i)\}_{i=1}^B$
    Store SGO into $\mathcal{D}_R$; $\mathcal{D}_R \leftarrow \mathcal{D}_R \cup \{(\mathbf{x}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^B$
  end if
  if $u < \phi$ then
    /* Use SGO in Off-policy Approach (Fig. 4(c)) */
    Sample mini-batch $\{(\mathbf{x}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^B$ from $\mathcal{D}_R$
  else
    /* Use Sample from Fixed Dataset (Fig. 4(a)) */
    Sample mini-batch $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^B$ from $\mathcal{D}$
  end if
  /* Use S(R)KL */
  Update $\theta_t$ by S(R)KL $D_{\text{SKL}}^{(\alpha)}(\cdot, \cdot)$
  if do validation then
    $\mathcal{L}_{prev}, \phi \leftarrow$ SGO_Scheduler($\mathcal{L}_{prev}, \mathcal{D}_{val}, q_{\theta_t}$)
  end if
end while

/* Adaptive SGO Scheduler */
def SGO_Scheduler($\mathcal{L}_{\tilde{t}-1}, \mathcal{D}_{val}, q_\theta$):
  /* Compute Loss for Validation Set */
  $\mathcal{L}_{\tilde{t}} \leftarrow \frac{1}{|D_{val}|} \sum_{\mathbf{x}_{val}, \mathbf{y}_{val}} \texttt{Loss}(q_\theta, \mathbf{x}_{val}, \mathbf{y}_{val})$
  if $\mathcal{L}_{\tilde{t}} > \mathcal{L}_{\tilde{t}-1} + \varepsilon$ then
    Update $\phi_{\tilde{t}} \leftarrow \min(\phi_{\tilde{t}-1} + 1/N_\phi, 1.0)$
  else
    $\mathcal{L}_{\tilde{t}}, \phi_{\tilde{t}} \leftarrow \mathcal{L}_{\tilde{t}-1}, \phi_{\tilde{t}-1}$
  end if
  return $\mathcal{L}_{\tilde{t}}, \phi_{\tilde{t}}$
```

However, to the best of our knowledge, there has been limited comprehensive effort to address these challenges simultaneously. For instance, MiniLLM (Gu et al., 2024) suggests a method that mixes the distributions of teacher and student to alleviate the first challenge. However, this method notably increases training computation due to the requirement of a large teacher model. These challenges motivate us to develop an approach that adaptively balances the positive effect of reduction of training-inference mismatch (Agarwal et al., 2024) and the negative effect of performance degradation from noisy feedback (as shown in Tab. 2). Meanwhile, we also aim to improve the sample efficiency of SGO, thereby enhancing computational efficiency.



(a) Grad. coef. (SKL) (b) Grad. coef. (SRKL) (c) L2 norm (w/o normalization) (d) L2 norm (w/ normalization)

Figure 3. **(a)**-**(b)**: Gradient coefficient distribution for SKL and SRKL across different skew values α, as shown in Eq. 6–7. **(c)**: Distribution of differences between divergence values and their (exponential) moving average of α-S(R)KL, as shown in Thm. 1, and those of β-JSD by substituting SKL into JSD across different α and β, respectively. **(d)**: Normalized L2 norm distribution, dividing the L2 norm in **(c)** by corresponding gradient coefficient values.

3. DistiLLM

In this section, we present the technical details of **DistiLLM**, addressing the limitations of previous methods. Our proposed method includes: (1) ***Skew KLD*** (Sec. 3.1), which significantly improves optimization stability and generalizability. The skew KLD loss addresses the pitfalls of previous objective functions that may lead the student model to sub-optimal, lacking analytical grounding. (2) ***Adaptive off-policy approach*** (Sec. 3.2), which comprises a novel adaptive SGO scheduler to balance the trade-off between noisy feedback and training-inference mismatch by minimally utilizing SGO, and off-policy strategy to improve the sample efficiency of SGO with maintaining the performance. We present the overall pipeline of **DistiLLM** in Algorithm 1.

3.1. Skew (Reverse) KLD

We mathematically present our motivation that skewing such KLD is highly effective in improving the performance of student models with a more favorable optimization process. The definition of skew KLD (SKL, Lee 2001) employs the parameter α that controls the mixing ratio of two distributions. The α-SKL between p and q_θ is defined as the KLD between p and the mixture of distributions $\alpha p + (1-\alpha)q_\theta$:

$$D_{\text{SKL}}^{(\alpha)}(p, q_\theta) = D_{\text{KL}}\left(p, \alpha p + (1-\alpha)q_\theta\right).$$

We similarly define the α-SRKL by $D_{\text{SRKL}}^{(\alpha)}(p, q_\theta) = D_{\text{KL}}(q_\theta, (1-\alpha)p + \alpha q_\theta)$. Here, following our thorough analysis, we present a comprehensive insight suggesting that S(R)KL is superior to other loss functions, owing to its more stable gradient and smaller approximation error.

Stable gradient. To provide stable optimization of SKL, we first analyze the gradients of KLD and SKL to parameter θ. Given a context-target sequence pair $(\mathbf{x}, \mathbf{y})$, we define the gradient of KLD w.r.t. θ (Ji et al., 2023):

$$\nabla_\theta D_{\text{KL}}(p, q_\theta) = -\mathbf{r}_{p,q_\theta} \nabla_\theta q_\theta(\mathbf{y}|\mathbf{x}), \tag{5}$$

where $\mathbf{r}_{p_1,p_2}$ is the ratio between arbitrary distribution p_1 and p_2. The result is the model probability's negative gradient, weighted inversely by its value. If $q_\theta(\mathbf{y}|\mathbf{x}) \approx 0$, the gradient norm grows large, causing a significant, potentially noisy parameter update step. These ingredients can adversely affect gradient updates, impacting the optimization process. We now compute the gradient of SKL w.r.t. θ:

$$\nabla_\theta D_{\text{SKL}}^{(\alpha)}(p, q_\theta) = -\underbrace{(1-\alpha)\mathbf{r}_{p,\tilde{q}_\theta}}_{\text{coefficient}} \nabla_\theta q_\theta(\mathbf{y}|\mathbf{x}), \tag{6}$$

where $\tilde{q}_\theta(\mathbf{y}|\mathbf{x}) = \alpha p(\mathbf{y}|\mathbf{x}) + (1-\alpha) q_\theta(\mathbf{y}|\mathbf{x})$. SKL offers a reduced gradient norm compared to KLD, due to p and q_θ interpolation preventing the denominator of $\mathbf{r}_{p,\tilde{q}_\theta}$ from reaching zero. **This results in a more stable gradient for SKL.** The gradient analysis for RKLD and SRKL reveals similar trends.

$$\nabla_\theta D_{\text{KL}}(q_\theta, p) = -\left(\log \mathbf{r}_{q_\theta,p} + 1\right) \nabla_\theta q_\theta(\mathbf{y}|\mathbf{x}), \tag{7}$$

$$\nabla_\theta D_{\text{SKL}}^{(\alpha)}(q_\theta, p) = -\underbrace{\left(\log \mathbf{r}_{q_\theta,\tilde{p}} + 1 - \alpha \mathbf{r}_{q_\theta,\tilde{p}}\right)}_{\text{coefficient}} \nabla_\theta q_\theta(\mathbf{y}|\mathbf{x}),$$

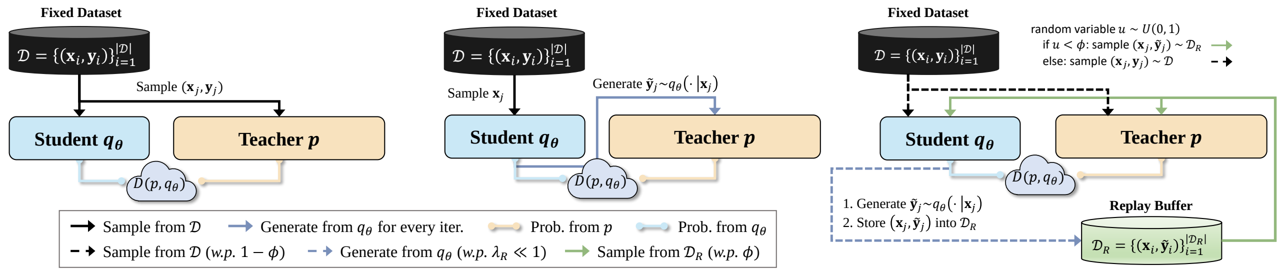
where $\tilde{p}(\mathbf{y}|\mathbf{x}) = (1-\alpha)p(\mathbf{y}|\mathbf{x}) + \alpha q_\theta(\mathbf{y}|\mathbf{x})$. All derivations for the gradients are in Appendix B.1. We visualize the gradient coefficient distribution in Fig. 3(a) and Fig. 3(b) which verify our analysis for gradient. As α becomes large, the coefficient is effectively small for both SKL and SRKL.

Small approximation error. We show that the empirical estimator of SKL from mini-batch training has a bounded L2 norm. This bounded norm ensures that rapid convergence, with minimal error between the estimator and true divergence, yields high generalizability by accurately reflecting the full distribution from the empirical estimator.

Theorem 1. *Let p_n^1 and p_n^2 be empirical distributions of n i.i.d. samples from p^1 and p^2, respectively. Under mild assumptions, we have an upper bound for the L2 norm of α-SKL estimator $D_{SKL}^{(\alpha)}(p_n^1, p_n^2)$ for $D_{SKL}^{(\alpha)}(p^1, p^2)$:*

$$\mathbb{E}[|D_{SKL}^{(\alpha)}(p_n^1, p_n^2) - D_{SKL}^{(\alpha)}(p^1, p^2)|^2] \le \frac{c_1(\alpha)}{n^2} + \frac{c_2 \log^2(\alpha n)}{n} + \frac{c_3 \log^2(c_4 n)}{\alpha^2 n},$$

for $c_1(\alpha) = \min\left\{\frac{1}{\alpha^2}, \frac{\chi^2(p^1,p^2)^2}{(1-\alpha)^2}\right\}$ and positive constants c_2, c_3, c_4 that are independent of n, α, and $D_{KL}(p^1, p^2)$, where $\chi^2(p^1, p^2)$ is chi-square divergence between p^1, p^2.


Fixed Dataset
$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{|\mathcal{D}|}$
Sample $(\mathbf{x}_j, \mathbf{y}_j)$
Student q_θ
Teacher p
$\mathcal{D}(p, q_\theta)$
Sample $\mathbf{x}_j$
Generate $\tilde{\mathbf{y}}_j \sim q_\theta(\cdot|\mathbf{x}_j)$
random variable $u \sim U(0,1)$
if $u < \phi$: sample $(\mathbf{x}_j, \tilde{\mathbf{y}}_j) \sim \mathcal{D}_R$
else: sample $(\mathbf{x}_j, \mathbf{y}_j) \sim \mathcal{D}$
1. Generate $\tilde{\mathbf{y}}_j \sim q_\theta(\cdot|\mathbf{x}_j)$
2. Store $(\mathbf{x}_j, \tilde{\mathbf{y}}_j)$ into $\mathcal{D}_R$
Replay Buffer
$\mathcal{D}_R = \{(\mathbf{x}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^{|\mathcal{D}_R|}$
Sample from $\mathcal{D}$
Generate from q_θ for every iter.
Prob. from p
Prob. from q_θ
Sample from $\mathcal{D}$ (w.p. $1-\phi$)
Generate from q_θ (w.p. $\lambda_R \ll 1$)
Sample from $\mathcal{D}_R$ (w.p. ϕ)

(a) KD from Fixed Dataset **(b) On-policy Approach** **(c) Adaptive Off-policy Approach (Ours)**

Figure 4. (a) KD from fixed dataset (Hinton et al., 2015) shows higher efficiency but lower performance. (b) On-policy approach (Agarwal et al., 2024; Gu et al., 2024) shows higher performance but lower efficiency. (c) Our adaptive off-policy approach shows both higher performance and efficiency. This advantage is attributed to introducing a replay buffer and progressively decreasing a replay ratio $\zeta := (1 - \frac{t}{T})$, which consequently maintains small SGO generation frequency $\lambda_R := \phi(1 - \frac{t}{T})$ during the entire training phase.

The proof is in Appendix B.2. Thm. 1 states that a large α lowers the L2 norm between empirical and true objectives. We show in Fig. 3(c) that α-S(R)KL reduces the error between the value for each mini-batch and their moving average more effectively than (R)KLD. However, considering the gradient scale reduction in Eq. 6–7, the benefit of a reduced L2 norm from Thm.1 is negated by compensating the reduced gradient scale of modern optimizers (Loshchilov & Hutter, 2017). We further provide a statement considering the reverse of approximated gradient coefficient, $\frac{1}{(1-\alpha)}$, especially when $\mathbf{r}_{\cdot,\cdot}$ averages near 1:

Remark 1. *By considering the reverse of approximated gradient scale, we have:*

$$\mathbb{E}[|\tfrac{1}{(1-\alpha)}(D_{SKL}^{(\alpha)}(p_n^1, p_n^2) - D_{SKL}^{(\alpha)}(p^1, p^2))|^2] \leq \frac{c_1^*(\alpha)}{n^2} + \frac{c_2 \log^2(\alpha n)}{(1-\alpha)^2 n} + \frac{c_3 \log^2(c_4 n)}{\alpha^2(1-\alpha)^2 n},$$

for $c_1^*(\alpha) = \min\left\{\frac{1}{\alpha^2(1-\alpha)^2}, \frac{\chi^2(p^1,p^2)^2}{(1-\alpha)^4}\right\}$.

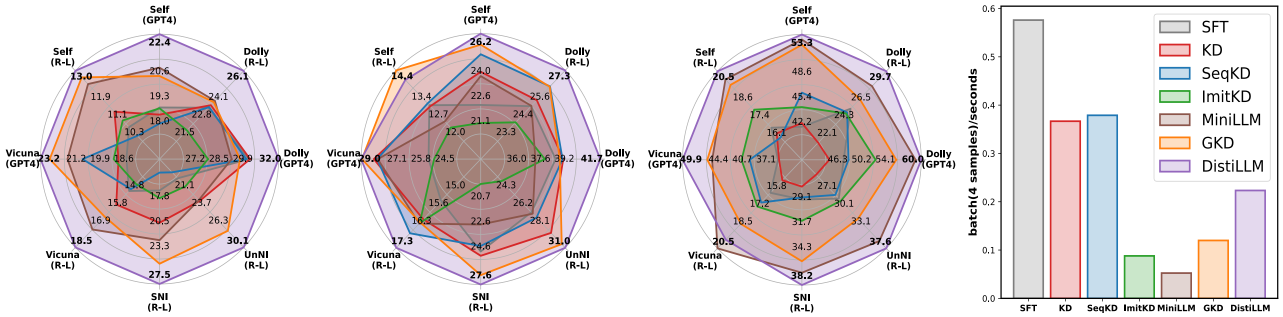
Overall, selecting α involves a trade-off: the relationship between the upper bound of the normalized L2 norm and $\alpha \in [0,1]$ appears to be convex, underscoring the importance of balancing gradient and L2 norm scales, as shown in Fig. 3(d). ***From these results, we also discern a fundamental difference between S(R)KL and JSD: S(R)KL with a mild α achieves an appropriate L2 norm value, whereas $D_{JSD}^{(\beta)}(p, q_\theta) := \beta D_{SKL}^{(\beta)}(p, q_\theta) + (1-\beta) D_{SKL}^{(1-\beta)}(q_\theta, p)$ cannot simultaneously moderate skew values for both terms.*** Our experiments indicate that α-SKL and α-SRKL are most effective with $\alpha = 0.1$, surpassing KLD, RKLD, and JSD in performance, as demonstrated in Tab. 1 and Fig. 8.

3.2. Adaptive Off-policy Approach

In Sec. 2.2, we have discussed two main issues of naïvely using SGO: (1) *the risk of noisy feedback due to the teacher model's unfamiliarity with the SGO* and (2) *the significant increase in training time*. For instance, employing SGO at every training iteration (*on-policy*, Agarwal et al. 2024) can lead to a substantial increase in runtime, up to 5.5× (refer to Fig. 7), and may also result in performance degradation (refer to Fig. 5). To tackle these issues, we propose (1) an adaptive SGO scheduler to conservatively utilize SGO in KD, guided by the validation loss of student models, thus mitigating the risk of noisy feedback; and (2) an efficient off-policy strategy to improve the sample efficiency of SGO.

Adaptive SGO scheduler. We define the probability of using SGOs, denoted as ϕ. We apply SGOs with a probability of ϕ, *i.e.*, using samples from a fixed dataset with a probability of $1-\phi$ (refer to Fig. 4(a)). Unlike previous methods that maintain a consistently high ϕ (Lin et al., 2020; Agarwal et al., 2024), our scheduler starts with low ϕ value, gradually increasing during training. This strategy prevents student models from being overwhelmed by noisy feedback (as shown in Fig. 1). To manage the increase of ϕ, we primarily rely on validation loss as a metric. Our observations indicate that training on a diverse range of SGOs, rather than solely on a fixed dataset, mitigates training-inference mismatch and consequently lowers validation loss. We adjust ϕ by comparing the current and previous validation losses; an increase in validation loss leads to an increase in ϕ. This method effectively improves student model performance by striking a balance between managing noisy feedback and minimizing training-inference mismatch issues. For further details, please refer to Appendix C and Algorithm 1.

Off-policy approach for sample efficiency. To enhance efficiency, we replace the recently adopted on-policy approach (Gu et al., 2024; Agarwal et al., 2024) with an off-policy approach, employing a replay buffer (Mnih et al., 2015; Fedus et al., 2020), **as illustrated in Fig. 4(c)**. In this buffer, we store SGOs from student models at a probability of λ_R, as indicated by **the frequency of the dashed blue arrows**. Subsequently, we randomly draw samples from this pool. We also replace the oldest samples of $\mathcal{D}_R$ with new ones once it reaches its maximum capacity. This off-policy strategy significantly improves the sample efficiency



(a) GPT-2 XL (1.5B) → GPT-2 (0.1B) (b) OPT-2.7B → OPT-1.3B (c) OLLaMA2-7B → OLLaMA2-3B (d) Training Speed (OLLaMA2-3B)

Figure 5. Instruction-following tasks, distilling GPT-2 (Radford et al., 2019), OPT (Zhang et al., 2022), and OpenLLaMA (OLLaMA; Geng & Liu 2023) model families on `databricks-dolly-15k`. GPT4 and R-L indicate GPT-4 feedback and ROUGE-L, respectively. To evaluate training speed, results are obtained using four A100 GPUs. Further details and results are in Fig. 14.

of KD with SGO, saving more resources than on-policy which constantly requires new data.

Off-policy reinforcement learning is prone to high bias error (Kumar et al., 2019; Lee et al., 2023), particularly when there is a significant divergence between past and current policies, using samples from the past policy becomes suboptimal for updating the current policy. To address this, we set $\lambda_R := \phi(1 - \frac{t}{T})$, where t represents the current training iteration and T is the total number of iterations. To explain the philosophy of our design, we define $\zeta := (1 - \frac{t}{T})$ as the replay ratio:

- In the ***early training phase*** (*i.e.*, when adaptive probability ϕ is small), where student model parameters rapidly evolve, we focus on using current SGOs with a high replay ratio to minimize bias error.
- In the ***late stages of training*** (*i.e.*, when ϕ is larger), as the student model nears convergence, we predominantly reuse stored SGOs from $\mathcal{D}_R$ with a small replay ratio.

Hence, we can consistently maintain a small λ_R by applying a large ζ for relatively smaller ϕ values and vice versa, effectively balancing bias error reduction with sample efficiency. Our design of ζ shows higher efficiency and comparable performance with its alternatives as shown in Tab. 9.

Synergy with SKL. Off-policy approach's success stems from the fast convergence speed of S(R)KL while other loss functions cannot be achieved. As Fig. 6 shows, both SKL and SRKL have a significant early-stage improvement, effectively leveraging the off-policy approach without high bias issues. This advantage is also evident in Tab. 4, where, unlike other baselines (Lin et al., 2020; Agarwal et al., 2024) that suffer performance drops when switching from on-policy to off-policy, our method maintains its efficacy. As a result, we verify that our off-policy approach significantly improves the training efficiency with a negligible performance drop as depicted in Tab. 2 and Fig. 7.

4. Experiments

We evaluate **DISTILLM** on instruction-following, text summarization, and machine translation tasks. We apply **DISTILLM** with SRKL and the off-policy approach with initial probability as zero and replay buffer size of 1000 as SRKL with α of 0.1 as determined through our ablation studies in Sec. 5. We compare our approach with previous KD: (1) supervised fine-tuning (SFT) directly fine-tunes the student on fixed datasets; (2) KD (Hinton et al., 2015) uses KLD on fixed datasets; (3) SeqKD (Kim & Rush, 2016) applies SFT to teacher-generated output; (4) ImitKD (Lin et al., 2020) employs KLD on SGO; (5) MiniLLM (Gu et al., 2024) utilizes a policy gradient approach on SGO; and (6) GKD (Agarwal et al., 2024) uses JSD on a mixture of SGOs and a fixed dataset. Further details on the experimental setup are found in Appendix D.

4.1. Task-Agnostic Instruction-Following

Implementation details. Our implementation follows the experiment setup of Gu et al. (2024). We first construct the training data from `databricks-dolly-15k` (Conover et al., 2023), wherein we randomly select 14K samples for training and equally leave 500 samples for validation and testing, respectively. We evaluate the trained models on five instruction-following benchmarks: Dolly evaluation, Self-Instruct (Wang et al., 2023b), Vicuna evaluation, Super-Natural Instructions (Wang et al., 2022), and Unnatural Instruction (Honovich et al., 2023). Similar to Ouyang et al. (2022) and Gu et al. (2024), we add a language modeling (Radford et al., 2018) loss to the OpenWebText (Gokaslan et al., 2019) corpus for all experiments. Employing such an additional loss function on the pre-training corpus has been shown to effectively improve instruction-tuning performance, as demonstrated in the work of Gu et al. (2024). For evaluation, we adopt two metrics: ROUGE-L (Lin, 2004) and GPT-4 feedback (Zheng et al., 2023).

Table 1. Evaluation of the effect of SKL and SRKL loss functions. **Bold** and underline indicate the best and second-best results, respectively, among those from the same evaluation dataset. We report the average and standard deviation of ROUGE-L scores across five random seeds. Green (🟢) and red (🔴) arrows indicate improvement and deterioration over the corresponding baselines.

Loss Function	Dolly Eval (↑)	Self-Instruct (↑)	Vicuna Eval (↑)	Super-Natural (↑)	Unnatural (↑)
KLD	23.52 (0.22)	11.23 (0.46)	15.92 (0.41)	20.68 (0.16)	23.38 (0.13)
RKLD	23.82 (0.34)	10.90 (0.58)	16.11 (0.46)	22.47 (0.21)	23.03 (0.11)
Generalized JSD	24.34 (0.35)	12.01 (0.54)	15.21 (0.61)	25.08 (0.36)	27.54 (0.07)
SKL	24.80 (0.12) 🟢	12.86 (0.34) 🟢	**16.20 (0.57)** 🟢	**26.26 (0.41)** 🟢	28.06 (0.08) 🟢
SRKL	**25.21 (0.27)** 🟢	**12.98 (0.24)** 🟢	15.77 (0.39) 🔴	25.83 (0.15) 🟢	**28.62 (0.10)** 🟢

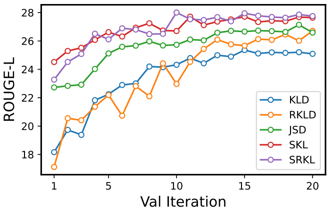


Figure 6. ROUGE-L scores for the validation set across the different loss functions.

Table 2. Evaluation of the adaptive off-policy approach. We report the average and standard deviation of ROUGE-L across five random seeds. The best and second best performances are highlighted **bold** and underline. Green (🟢) and red (🔴) arrows indicate improvement and deduction over the baselines.

Generation	Dolly Eval (↑)	Self-Instruct (↑)	Vicuna Eval (↑)	Super-Natural (↑)	Unnatural (↑)
Skew KLD	24.80 (0.12)	12.86 (0.34)	16.20 (0.57)	26.26 (0.41)	28.06 (0.08)
∟ On-policy	24.27 (0.46) 🔴	13.13 (0.44) 🟢	16.39 (0.21) 🟢	25.87 (0.18) 🔴	26.49 (0.09) 🔴
∟ Mixed	25.27 (0.35) 🟢	12.24 (0.69) 🔴	17.19 (0.29) 🟢	25.30 (0.33) 🔴	26.51 (0.11) 🔴
∟ Adaptive (ours)	**25.90 (0.20)** 🟢	**13.24 (0.30)** 🟢	**17.59 (0.44)** 🟢	**27.62 (0.05)** 🟢	**28.30 (0.11)** 🟢
+ Off-policy (ours)	25.79 (0.28) 🟢	13.03 (0.29) 🟢	17.41 (0.15) 🟢	27.32 (0.09) 🟢	28.13 (0.21) 🟢
Skew RKLD	25.21 (0.27)	12.98 (0.24)	15.77 (0.39)	25.83 (0.15)	28.62 (0.10)
∟ On-policy	26.04 (0.33) 🟢	12.93 (0.54) 🔴	17.45 (0.37) 🟢	27.29 (0.12) 🟢	28.72 (0.10) 🟢
∟ Mixed	26.01 (0.61) 🟢	12.24 (0.69) 🔴	17.19 (0.29) 🟢	26.40 (0.34) 🟢	29.02 (0.14) 🟢
∟ Adaptive (ours)	**26.37 (0.21)** 🟢	13.14 (0.37) 🟢	18.32 (0.17) 🟢	**28.24 (0.22)** 🟢	**30.11 (0.04)** 🟢
+ Off-policy (ours)	26.11 (0.68) 🟢	**13.14 (0.69)** 🟢	**18.46 (0.53)** 🟢	27.51 (0.03) 🟢	29.35 (0.07) 🟢

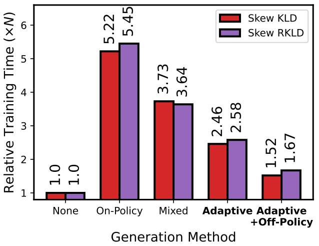


Figure 7. Relative training time for different generation methods for S(R)KL. The adaptive off-policy approach shows significant efficiency.

Main results. Fig. 5 illustrates the instruction-following performances, demonstrating **DistiLLM**'s superiority to state-of-the-art methods, across diverse teacher-student combinations and evaluation metrics. Notably, GKD and MiniLLM, which recently incorporated SGO in distillation, are less effective or even lead to a performance decline in smaller-sized student models, particularly in GPT-4 feedback and some ROUGE-L scores. Further details and results can be found in Appendix E.1 and Tab. 11–13.

For evaluating **DistiLLM** on larger-sized LLMs, we utilize low-rank adaptation (LoRA; Hu et al. 2022) for training. Specifically, we employ OpenLLaMA2-7B (Geng & Liu, 2023) as the teacher model and OpenLLaMA2-3B as the student model. The results, as depicted in Fig. 5(c), reveal that **DistiLLM** significantly surpasses other baseline methods in performance. Notably, while other supervised KD techniques are less effective than MiniLLM in LLM applications, **DistiLLM** uniquely achieves superior performance compared to MiniLLM. This outcome, particularly on task-agnostic instruction-following datasets, underscores **DistiLLM**'s effectiveness in general-purpose LLMs. Moreover, as we can observe in Fig. 5(d), **DistiLLM** requires only 1.6× the training time of naïve KD, whereas other methods take $3 \sim 7\times$. This demonstrates the efficiency of the proposed **DistiLLM** and its broad applicability to expensive LLMs.

Table 3. Text summarization and machine translation results, evaluated with ROUGE-L and BLEU scores, respectively. Using the T5-XL (3B) teacher model, we fine-tune the student models, T5-Base (0.2B) and T5-Small (0.06B).

Dataset	SAMSum		IWSLT 2017 En-De	
T5-XL →	***T5-Base***	***T5-Small***	***T5-Base***	***T5-Small***
KD (Hinton et al., 2015)	46.23	39.52	29.36	21.15
SeqKD (Kim & Rush, 2016)	46.89	40.24	29.07	21.42
ImitKD (Lin et al., 2020)	48.57	41.44	29.87	21.52
GKD (Agarwal et al., 2024)	48.49	41.92	30.24	22.04
DistiLLM (ours)	**49.11**	**42.37**	**30.32**	**22.53**

4.2. Text Summarization and Machine Translation

We evaluate the effectiveness of task-specific LMs on summarization and translation tasks using two datasets, SAMSum (Gliwa et al., 2019) and IWSLT 2017 (Cettolo et al., 2017). For the SAMSum dataset, we use T5-XL v1.1 (Raffel et al., 2020) as the teacher model and T5-Base/-Small v1.1 as the student models. For the IWSLT dataset, we employed mT5-XL (Xue et al., 2021) as the teacher model and mT5-Base/-Small v1.1 as the student models.

Tab. 3 displays ROUGE-L and BLEU (Papineni et al., 2002) scores for student models. We observe that **DistiLLM** outperforms other baselines, but has a smaller performance margin in single-task scenarios than in general-purpose instruction-following tasks. In the SAMSum, students trained with ImitKD outperform those trained with GKD, while in the IWSLT, GKD outperforms ImitKD. These find-

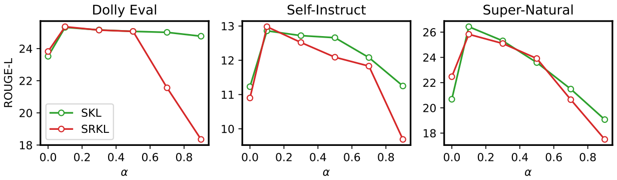


Figure 8. Comparison of the ROUGE-L score using different α.

ings align with the results in Agarwal et al. (2024), which indicate that the effectiveness of objective functions and the use of SGO are task-dependent. Despite these variations, **DistiLLM** consistently shows superiority across different tasks, attributed to its adaptive use of the SGO scheduler and skew divergence. Additional details on the performance in the XSum (Narayan et al., 2018) and CNN/DM (See et al., 2017) datasets are provided in Appendix E.2, further emphasizing the superior performance of **DistiLLM**.

5. Analysis and Discussion

We conduct experimental analyses to verify the effectiveness of each component of **DistiLLM**, distilling GPT-2 XL → GPT-2 in instruction-following datasets.

5.1. Effect of Skew Divergence

In Tab. 1, we compare the performance of various models trained with different objective functions: conventional KLD, RKLD, and JSD with a β of 0.9 (Agarwal et al., 2024), as well as our SKL and SRKL with a α of 0.1. The results show that our proposed objective functions generally outperform the others. Notably, as Fig. 6 illustrates, both SKL and SRKL achieve remarkably high validation ROUGE-L scores during the entire training phase, consistently demonstrating the rapid convergence and strong generalization capabilities of our proposed loss functions. These empirical results verify our theoretical analysis in Sec. 3.1 and indicate that even our simple modification leads to significant performance enhancements.

5.2. Effect of Adaptive Off-policy Approach

Table 4. Application of our off-policy method to the existing KD methods. Off-policy significantly reduces the performance of ImitKD and GKD, as opposed to our proposed **DistiLLM**.

Dataset	Dolly Eval		Self-Instruct		Super-Natural	
Sampling	on-	off-	on-	off-	on-	off-
ImitKD (Lin et al., 2020)	21.63	20.62	10.85	10.09	19.94	18.04
GKD (Agarwal et al., 2024)	23.75	22.89	12.73	12.78	26.05	24.97
DistiLLM (ours)	26.37	26.12	13.14	13.16	28.24	28.20

Table 5. Comparison of the performance from the adaptive SGO scheduler and its terminal probability (0.4 for SKL and 0.3 for SRKL) and the best performance from manually tuned probability and corresponding value (*i.e.*, results in parenthesis).

Loss	Method	Dolly	Self-Inst	Vicuna	SNI	UNI
SKL	Adapt.	25.90	13.24	17.59	27.62	28.30
	Best	25.15 (0.3)	13.17 (0.5)	17.04 (0.3)	27.18 (0.4)	28.33 (0.6)
SRKL	Adapt.	26.37	13.14	18.32	28.24	30.11
	Best	26.38 (0.4)	12.98 (0.3)	17.88 (0.3)	28.24 (0.3)	30.02 (0.3)

In Tab. 2 and Fig. 7, we confirm the effectiveness and efficiency of our adaptive off-policy approach by comparing it with an on-policy (Lin et al., 2020) and a mixed strategy (Agarwal et al., 2024). In the mixed strategy, we use the on-policy approach with a probability of 0.5; otherwise, we sample from the fixed dataset, following the original work. The results indicate that our adaptive SGO scheduler effectively balances the trade-off between the risk of noisy feedback and training-inference mismatch. Notably, while the baselines suffer from performance degradation when applying SKL, our proposed adaptive strategy consistently improves the performance for all datasets. Moreover, the off-policy approach leads to a minimal performance drop while significantly improving computational efficiency. It achieves 2.2× to 3.4× faster training speed compared to the on-policy or mixed strategy.

5.3. Additional Ablation Studies on DistiLLM

Skew values α. As we highlight the importance of proper choice α in Sec. 3.1, we empirically evaluate the performance of α-SKL under the wide range of α. Fig. 8 illustrates that both SKL and SRKL achieve the best performance on the α value of 0.1. These results are highly consistent with the result in Fig. 3, where both normalized L2 norms have the smallest values in $\alpha = 0.1$ and prove the validity of our theoretical analyses in Sec. 3.1. We observed that SKL shows a mild performance reduction as α increases above 0.1, the performance reduction of the SRKL is comparably severe which is discussed in detail in Appendix E.4.

Combining off-policy with existing KD methods. To verify the synergy between our proposed objectives and the off-policy approach, we replaced the on-policy approach in GKD and ImitKD with an off-policy method. Given that the JSD and KLD exhibit slower training speeds compared to our SRKL, as shown in Fig. 6, we observe that the effectiveness of the off-policy approach is significantly lower than that of our proposed **DistiLLM** as reported in Tab. 4. These results validate the substantial effectiveness of combining our two proposed components.

Appropriateness of adaptive probability. Tab. 5 compares the performance of probability values from the adaptive SGO scheduler with the best performance among those obtained from manually defined probabilities. Our results show that, in most cases, the performance using our method exceeds that of the manually selected probabilities. Furthermore, the probability values derived from our adaptive SGO scheduler are close to the optimal manually defined probabilities, with the differences being no more than 0.2

Table 6. Comparison of the performance of GPT-2 student across different KD methods and initialization (with or without fine-tuning before KD).

Dataset	Dolly Eval		Self-Instruct		Super-Natural	
Fine-tuned	✓	✗	✓	✗	✓	✗
MiniLLM (Gu et al., 2024)	23.84	22.56	12.44	10.47	22.62	21.15
GKD (Agarwal et al., 2024)	23.75	23.15	12.73	11.34	26.05	24.48
DistiLLM (SKL)	25.79	25.75	13.03	12.34	27.32	27.24
DistiLLM (SRKL)	26.11	26.14	13.14	12.79	27.51	26.85

for SKL and 0.1 for SRKL.

One-stage Distillation One significant issue with previous KD methods using SGO is their reliance on beginning with fine-tuned student models. Without fine-tuning, student models may produce degenerated SGOs, leading to substantial noisy feedback (Gu et al., 2024). A key advantage of our proposed **DistiLLM** is its rapid convergence speed and innovative adaptive SGO scheduler, which eliminates the need for such fine-tuned student models. Consequently, we evaluated the performance of students distilled from pre-trained parameters without any prior fine-tuning. Tab. 6 demonstrates **DistiLLM**'s relative robustness to the initial state of the student model (*i.e.*, regardless of whether the student was fine-tuned before KD). These findings confirm that **DistiLLM** can maintain efficiency with only a minor reduction in performance, a feat not matched by other methods.

6. Related Work

KD (Hinton et al., 2015) effectively compresses neural networks, allowing smaller student models to match the performance of larger teacher models. Recently, KD has been extended to compressing auto-regressive LMs, such as GPT-3 (Brown et al., 2020), to address the challenges posed by the large scale of current LLMs (Touvron et al., 2023; Anil et al., 2023), making them more viable in compute-intensive frameworks. One popular direction of KD for auto-regressive LMs is to harness LLMs as supervising data generators where only the teacher predictions are accessible like ChatGPT (OpenAI, 2023) APIs. This line of research employed LLMs for guided annotations of unlabeled data (Taori et al., 2023; Peng et al., 2023) or for imparting reasoning capabilities (Wang et al., 2023a; Hsieh et al., 2023), where the resulting generated data are used for fine-tuning smaller LMs.

Another noteworthy approach, when the teacher model is accessible, entails matching the student model's generation distribution with that of the teacher model through divergence loss functions. Recent studies (Lin et al., 2020; Wen et al., 2023; Gu et al., 2024; Agarwal et al., 2024) have focused on finding the proper objectives or using datasets during the distillation for auto-regressive LMs. ImitKD (Lin et al., 2020) demonstrated the effectiveness of SGO in distillation, leading to Agarwal et al. (2024) propose on-policy approach of SGO with diverse objectives like RKLD and JSD. Wen et al. (2023) examined various f-divergences, including total variation distance and JSD, in auto-regressive LMs, while Gu et al. (2024) proposed a policy gradient-based method addressing the high variance issues in RL-based methods. Building on this research, we introduce an effective, efficient KD method, **DistiLLM**, with comprehensive analysis in objective function and data utilization.

7. Conclusion

We have proposed **DistiLLM** to address the challenges of KD frameworks for auto-regressive LMs. Our approach incorporates two key components: (1) SKL which is based on mathematically in-depth analyses and empirical evidence; (2) an adaptive off-policy approach that enhances the utility of SGO by reducing the noisy feedback introduced by SGO and improving the sample efficiency with a replay buffer. Through extensive experiments on various generation tasks, we have demonstrated the superior performance of **DistiLLM**, achieving significant training efficiency and performance improvement.

Impact Statement

This paper aims to improve the efficiency and effectiveness of KD for auto-regressive LMs, such as open-source LLMs (*e.g.,* LLaMA-2 (Touvron et al., 2023) and Falcon (Almazrouei et al., 2023)). A notable feature of our methodology is its dual efficacy in improving performance and computational efficiency. This approach is significant, particularly as it reduces reliance on high-end computational resources for training such small-sized LLMs. We believe that this work does not present any direct ethical concerns. Therefore, a detailed ethical discussion is not considered necessary at this stage. However, we recognize the importance of ethical considerations and are open to further discussion should our work undergo an ethics review.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST), 10%) and the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration, 90%).

References

Agarwal, R., Vieillard, N., Stanczyk, P., Ramos, S., Geist, M., and Bachem, O. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Alhammadi, M., Daniele, M., Heslow, D., Launay, J., Malartic, Q., Noune, B., Pannier, B., and Penedo, G. The falcon series of language models: Towards open frontier models. *https://arxiv.org/abs/2311.16867*, 2023.

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Arora, K., El Asri, L., Bahuleyan, H., and Cheung, J. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 700–710, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.58. URL https://aclanthology.org/2022.findings-acl.58.

Boizard, N., El-Haddad, K., Hudelot, C., and Colombo, P. Towards cross-tokenizer distillation: the universal logit distillation loss for llms. *arXiv preprint arXiv:2402.12030*, 2024.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Cettolo, M., Federico, M., Bentivogli, L., Niehues, J., Stüker, S., Sudoh, K., Yoshino, K., and Federmann, C. Overview of the IWSLT 2017 evaluation campaign. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pp. 2–14, Tokyo, Japan, December 14-15 2017. International Workshop on Spoken Language Translation. URL https://aclanthology.org/2017.iwslt-1.1.

Chen, Z., Deng, Y., Yuan, H., Ji, K., and Gu, Q. Self-play fine-tuning converts weak language models to strong language models, 2024.

Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Conover, M., Hayes, M., Mathur, A., Xie, J., Wan, J., Shah, S., Ghodsi, A., Wendell, P., Zaharia, M., and Xin, R. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-\instruction-tuned-llm.

Fedus, W., Ramachandran, P., Agarwal, R., Bengio, Y., Larochelle, H., Rowland, M., and Dabney, W. Revisiting fundamentals of experience replay. In *International Conference on Machine Learning*, pp. 3061–3071. PMLR, 2020.

Geng, X. and Liu, H. Openllama: An open reproduction of llama, May 2023. URL https://github.com/openlm-research/open_llama.

Gliwa, B., Mochol, I., Biesek, M., and Wawer, A. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 70–79, 2019.

Gokaslan, A., Cohen, V., Pavlick, E., and Tellex, S. Openwebtext corpus, 2019.

Gu, Y., Dong, L., Wei, F., and Huang, M. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Hinton, G. E., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015. URL https://api.semanticscholar.org/CorpusID:7200347.

Honovich, O., Scialom, T., Levy, O., and Schick, T. Unnatural instructions: Tuning language models with (almost) no human labor. In *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. URL https://aclanthology.org/2023.acl-long.806.

Hsieh, C.-Y., Li, C.-L., Yeh, C.-K., Nakhost, H., Fujii, Y., Ratner, A., Krishna, R., Lee, C.-Y., and Pfister, T. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*, 2023.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Ji, H., Ke, P., Hu, Z., Zhang, R., and Huang, M. Tailoring language generation models under total variation distance. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VELL0PlWfc.

Kim, Y. and Rush, A. M. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

Lee, H., Cho, H., Kim, H., Gwak, D., Kim, J., Choo, J., Yun, S.-Y., and Yun, C. Plastic: Improving input and label plasticity for sample efficient reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Lee, K. and Shin, J. Rényicl: Contrastive representation learning with skew rényi divergence. *Advances in Neural Information Processing Systems*, 35:6463–6477, 2022.

Lee, L. On the effectiveness of the skew divergence for statistical language analysis. In *International Workshop on Artificial Intelligence and Statistics*, pp. 176–183. PMLR, 2001.

Lin, A., Wohlwend, J., Chen, H., and Lei, T. Autoregressive knowledge distillation through imitation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6121–6133, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.494. URL https://aclanthology.org/2020.emnlp-main.494.

Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

Liu, L., Pillutla, K., Welleck, S., Oh, S., Choi, Y., and Harchaoui, Z. Divergence frontiers for generative models: Sample complexity, quantization effects, and frontier integrals. *Advances in Neural Information Processing Systems*, 34:12930–12942, 2021.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Mirzadeh, S. I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., and Ghasemzadeh, H. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5191–5198, 2020.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533, 2015.

Narayan, S., Cohen, S. B., and Lapata, M. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, 2018.

OpenAI. Gpt-4 technical report, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In Isabelle, P., Charniak, E., and Lin, D. (eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

Pedro, D. A unified bias-variance decomposition and its applications. In *17th International Conference on Machine Learning*, pp. 231–238, 2000.

Peng, B., Li, C., He, P., Galley, M., and Gao, J. Instruction tuning with gpt-4, 2023.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. *OpenAI blog*, 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2023.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Ren, S., Wu, Z., and Zhu, K. Q. EMO: EARTH MOVER DISTANCE OPTIMIZATION FOR AUTO-REGRESSIVE LANGUAGE MODELING. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4bLXfRd0CX.

Rubenstein, P., Bousquet, O., Djolonga, J., Riquelme, C., and Tolstikhin, I. O. Practical and consistent estimation of f-divergences. *Advances in Neural Information Processing Systems*, 32, 2019.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019. URL https://api.semanticscholar.org/CorpusID:203626972.

See, A., Liu, P. J., and Manning, C. D. Get to the point: Summarization with pointer-generator networks. In Barzilay, R. and Kan, M.-Y. (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL https://aclanthology.org/P17-1099.

Sun, S., Cheng, Y., Gan, Z., and Liu, J. Patient knowledge distillation for BERT model compression. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4323–4332, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1441. URL https://aclanthology.org/D19-1441.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Wan, F., Huang, X., Cai, D., Quan, X., Bi, W., and Shi, S. Knowledge fusion of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jiDsk12qcz.

Wang, P., Wang, Z., Li, Z., Gao, Y., Yin, B., and Ren, X. Scott: Self-consistent chain-of-thought distillation. *arXiv preprint arXiv:2305.01879*, 2023a.

Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Naik, A., Ashok, A., Dhanasekaran, A. S., Arunkumar, A., Stap, D., Pathak, E., Karamanolakis, G., Lai, H., Purohit, I., Mondal, I., Anderson, J., Kuznia, K., Doshi, K., Pal, K. K., Patel, M., Moradshahi, M., Parmar, M., Purohit, M., Varshney, N., Kaza, P. R., Verma, P., Puri, R. S., Karia, R., Doshi, S., Sampat, S. K., Mishra, S., Reddy A, S., Patro, S., Dixit, T., and Shen, X. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *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. URL https://aclanthology.org/2022.emnlp-main.340.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.754. URL https://aclanthology.org/2023.acl-long.754.

Wen, Y., Li, Z., Du, W., and Mou, L. f-divergence minimization for sequence-level knowledge distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10817–10834, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.605. URL https://aclanthology.org/2023.acl-long.605.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. mT5: A massively multilingual pre-trained text-to-text transformer. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y. (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 483–498, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.41. URL https://aclanthology.org/2021.naacl-main.41.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=uccHPGDlao.

DistiLLM: Towards Streamlined Distillation for Large Language Models

Supplementary Material

A. Limitation

While **DistiLLM** shows effectiveness in terms of computational efficiency of training and students' performances compared to recent baselines (Gu et al., 2024; Agarwal et al., 2024), we acknowledge a couple of limitations:

- **KLD-based Objective**: Our focus is mainly on (R)KLD because of its tractability that allows decomposing sequence-level distillation into token-wise distillation (as shown in Eq. (2)-(3)). However, (R)KLD also entails limitations previously identified in Ji et al. (2023); Ren et al. (2024): mode averaging (or mode collapse) and train-inference mismatch. Although DistiLLM, especially S(R)KL, effectively mitigated these issues, additionally combining with objective functions based on TVD (Ji et al., 2023) or EMD (Ren et al., 2024) could further enhance performance. We expect that by linearly interpolating between SKL and EMD (or TVD), we can leverage the strengths of both: the rapid convergence of SKL during the early stages of training and the superior performance of EMD or TVD towards the late training compared to KLD.

- **Supervised Fine-tuning Approach**: **DistiLLM** is is primarily designed for a supervised fine-tuning setup, which has recently shown effectiveness (Chen et al., 2024), Meanwhile, many contemporary chat LLMs utilize preference optimization (Ouyang et al., 2022; Rafailov et al., 2023); thus, an extension of **DistiLLM** to accommodate human preference optimization setups might be considered as future work.

- **Same Tokenizer between Teacher & Student**: Our method focuses on the scenario where the teacher and student models share the same tokenizer, which is also a common setup for white-box KD methods. However, with recent techniques (Wan et al., 2024; Boizard et al., 2024) that facilitate the transfer of knowledge between models with different tokenizers, we can designate this for future work to be explored in conjunction with these new methods.

B. Further Discussion on Skew KLD

Here, we provide the derivation of our theoretical results described in Sec. 3.1 and further (empirical) discussion on our proposed SKL.

B.1. Details for Gradient Analysis

Derivation of Sec. 3.1. We derive the sample-wise gradient of KLD, SKL, RKLD, and SRKL to support our argument that simple skew operation on KLD improves the stability of the optimization.

- *Firstly*, we compute the gradient of KLD for a single sample $(\mathbf{x}, \mathbf{y})$:

$$\begin{aligned}\nabla_\theta D_{\text{KL}}(p, q_\theta) &= -\nabla_\theta p(\mathbf{y}|\mathbf{x}) \log q_\theta(\mathbf{y}|\mathbf{x}) \\ &= -\frac{p(\mathbf{y}|\mathbf{x})}{q_\theta(\mathbf{y}|\mathbf{x})}\nabla_\theta q_\theta(\mathbf{y}|\mathbf{x}).\end{aligned}$$

- *Secondly*, we compute the gradient of SKL for $(\mathbf{x}, \mathbf{y})$:

$$\begin{aligned}&\nabla_\theta D^{(\alpha)}_{\text{SKL}}(p, q_\theta) \\ &\quad = -\nabla_\theta p(\mathbf{y}|\mathbf{x}) \log\left(\alpha p(\mathbf{y}|\mathbf{x}) + (1-\alpha) q_\theta(\mathbf{y}|\mathbf{x})\right) \\ &\quad = -\nabla_\theta p(\mathbf{y}|\mathbf{x}) \log \tilde{q}_\theta(\mathbf{y}|\mathbf{x}) \\ &\quad = -\frac{p(\mathbf{y}|\mathbf{x})}{\tilde{q}_\theta(\mathbf{y}|\mathbf{x})}\nabla_\theta \tilde{q}_\theta(\mathbf{y}|\mathbf{x}) \\ &\quad = -\frac{p(\mathbf{y}|\mathbf{x})}{\tilde{q}_\theta(\mathbf{y}|\mathbf{x})} \cdot (1-\alpha) \cdot \nabla_\theta q_\theta(\mathbf{y}|\mathbf{x}).\end{aligned}$$

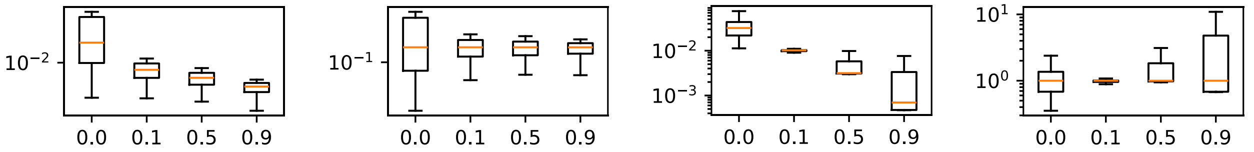
- *Thirdly*, we compute the gradient of RKLD for $(\mathbf{x}, \mathbf{y})$:

$$\begin{aligned}&\nabla_\theta D_{\text{KL}}(q_\theta, p) \\ &\quad = \log p(\mathbf{y}|\mathbf{x}) \nabla_\theta q_\theta(\mathbf{y}|\mathbf{x}) - \nabla_\theta (q_\theta(\mathbf{y}|\mathbf{x}) \log q_\theta(\mathbf{y}|\mathbf{x})) \\ &\quad = \nabla_\theta q_\theta(\mathbf{y}|\mathbf{x}) \cdot (\log p(\mathbf{y}|\mathbf{x}) q(\mathbf{y}|\mathbf{x}) - 1).\end{aligned}$$

- *Lastly*, with a definition of $\tilde{p}(\mathbf{y}|\mathbf{x}) = (1-\alpha)p(\mathbf{y}|\mathbf{x}) + \alpha q_\theta(\mathbf{y}|\mathbf{x})$, we compute the gradient of SRKL for $(\mathbf{x}, \mathbf{y})$:

$$\begin{aligned}&\nabla_\theta D^{(\alpha)}_{\text{SKL}}(q_\theta, p) \\ &= \nabla_\theta (q_\theta(\mathbf{y}|\mathbf{x}) \log \tilde{p}(\mathbf{y}|\mathbf{x})) - \nabla_\theta (q_\theta(\mathbf{y}|\mathbf{x}) \log q_\theta(\mathbf{y}|\mathbf{x})) \\ &= \log \tilde{p}(\mathbf{y}|\mathbf{x}) \nabla_\theta q_\theta(\mathbf{y}|\mathbf{x}) + \frac{q_\theta(\mathbf{y}|\mathbf{x})}{\tilde{p}(\mathbf{y}|\mathbf{x})}\nabla_\theta \tilde{p}(\mathbf{y}|\mathbf{x}) \\ &- \nabla_\theta q_\theta(\mathbf{y}|\mathbf{x}) \log q_\theta(\mathbf{y}|\mathbf{x}) - \nabla_\theta q_\theta(\mathbf{y}|\mathbf{x}) \\ &= -\left(\log \frac{q_\theta(\mathbf{y}|\mathbf{x})}{\tilde{p}(\mathbf{y}|\mathbf{x})} + 1 - \alpha \frac{q_\theta(\mathbf{y}|\mathbf{x})}{\tilde{p}(\mathbf{y}|\mathbf{x})}\right) \nabla_\theta q_\theta(\mathbf{y}|\mathbf{x}),\end{aligned}$$

As we described in Section 3.1, we can prevent the undesired gradient norm explosion due to smoothed distributions $\tilde{q}_\theta(\mathbf{y}|\mathbf{x})$ and $\tilde{p}(\mathbf{y}|\mathbf{x})$ for SKL and SRKL, respectively.



(a) SKL (w/o normalization) (b) SKL (w/ normalization) (c) SRKL (w/o normalization) (d) SRKL (w/ normalization)

Figure 9. Gradient coefficient distribution for SKL and SRKL across different skew values, α. Skewing KLD and RKLD effectively smooth the gradient norm, as seen in (a) and (c). For coefficients normalized by their median value, SKL shows a similar distribution when $\alpha > 0$ while SRKL exhibits explosion, as depicted in (b) and (d).

Further empirical discussion. Recent optimizers, such as Adam (Kingma & Ba, 2014) and AdamW (Loshchilov & Hutter, 2017), adaptively adjust learning rates based on the gradient scale. Hence, it is also important to observe the distribution of the normalized gradient divided by their scale. In Fig. 9, we additionally plot the normalized version of the gradient coefficient, obtained by dividing the corresponding median values of the coefficient. SKL shows consistently reduced variance as α increases not only for original values but also normalized values. However, SRKL shows the smallest variance at α of 0.1 for both original and normalized values, but when α is larger than 0.1, the variance becomes larger as α grows. These results are also related to those in Fig. 8, where SKL shows higher robustness than SRKL across different α values. Still, our mathematical analysis is valid given that the gradient scale becomes smaller as α increases.

B.2. Proof of Theorem 1

Theorem 1 (Restated). *Assume p^1, p^2 be two probability distributions such that $\mathbb{V}_{p^2}[dp^1/dp^2] < \infty$ and p^1 is absolutely continuous with respect to p^2 (i.e., the Radon-Nikodym derivative dp^1/dp^2 exists). Then, for $\forall\alpha < 1/8$, the variance of α-SKL estimator satisfies*

$$\mathbb{E}[|D^{(\alpha)}_{SKL}(p^1_n, p^2_n) - D^{(\alpha)}_{SKL}(p^1, p^2)|^2] \leq \frac{c_1(\alpha)}{n^2} + \frac{c_2 \log^2(\alpha n)}{n} + \frac{c_3 \log^2(c_4 n)}{\alpha^2 n},$$

for $c_1(\alpha) = \min\{\frac{1}{\alpha^2}, \frac{\chi^2(p^1,p^2)^2}{(1-\alpha)^2}\}$ and constants $c_2, c_3, c_4 > 0$ that are independent of n, α, and $D(p^1, p^2)$, where $\chi^2(p^1, p^2) := \mathbb{E}_{p^2}[(dp^1/dp^2)^2]$. Here, we denote $p^1 \ll p^2$ as p^1 is absolutely continuous with respect to p^2.

In this section, we state and define the regularity assumptions to derive the asymptotic upper bounds for the variance of α-skew KL divergence. Formally, the f-divergence of two distributions is defined as

$$D_f(p^1, p^2) = \mathbb{E}_{\mathbf{y}\sim p^1}\left[f\left(\frac{p^1(\mathbf{y}|\mathbf{x})}{p^2(\mathbf{y}|\mathbf{x})}\right)\right] := \mathbb{E}_{p^1}\left[f\left(\frac{dp^1}{dp^2}\right)\right],$$

where dp^1 and dp^2 are the probability densities of probability p^1 and p^2.

The KL divergence is a f-divergence generated by $f(t) = t\log t - t + 1$, and the α-skew KL divergence is a f-divergence generated by

$$f^{(\alpha)}(t) = t\log\left(\frac{t}{\alpha t + 1 - \alpha}\right) - (1-\alpha)(t-1).$$

By following Liu et al. (2021) and Lee & Shin (2022), we state the following regularity assumptions on the functions f and f^*.

Assumption B.1. *The generator f is twice continuously differentiable with $f'(1) = 0$. Moreover*

(A1): *We have $C_0 := f(0) < \infty$ and $C^*_0 := f^*(0) < \infty$.*

(A2): *There exist constants $C_1, C^*_1 < \infty$ such that for any $t \in (0,1)$, we have $|f'(t)| \leq C_1 \max\{1, \log(1/t)\}$ and $|(f^*)'(t)| \leq C^*_1 \max\{1, \log(1/t)\}$.*

(A3): *There exist constants $C_2, C^*_2 < \infty$ such that for every $t \in (0,\infty)$, we have $\frac{t}{2}f''(t) \leq C_2$ and $\frac{t}{2}(f^*)''(t) \leq C^*_2$.*

One can observe that both KLD and RKLD do not satisfy Assumption B.1 because KLD is unbounded. On the other hand, the α-skew KL divergence satisfies Assumption B.1 from the following proposition.

Lemma B.1 (Liu et al. 2021)**.** *The α-skew KL divergence generated by $f^{(\alpha)}$ satisfies Assumption B.1 with*

$$C_0 = 1-\alpha,\ \ C^*_0 = \log\frac{1}{\alpha} - 1 + \alpha,\ \ C_1 = 1,$$
$$C^*_1 = \frac{(1-\alpha)^2}{\alpha},\ \ C_2 = \frac{1}{2},\ \ C^*_2 = \frac{1-\alpha}{8\alpha}.$$

For general f-divergences which satisfy Assumption B.1, the following concentration bound holds.

Lemma B.2 (Liu et al. 2021)**.** *Assume f satisfies Assumption B.1, and let p^1 and p^2 be two distributions with $p^1 \ll p^2$. Let p^1_m be m i.i.d samples from p^1 and p^2_n be n i.i.d samples from p^2. Then the f-divergence D_f satisfies* *following:*

$$\mathbb{P}[|D_f(p_m^1, p_n^2) - \mathbb{E}[D_f(p_m^1, p_n^2)]| > \epsilon] \le$$
$$2\exp\left(-\frac{\epsilon^2}{\frac{2}{m}(C_1 \log m + c_1)^2 + \frac{2}{n}(C_1^* \log n + c_2)^2}\right)$$

where $c_1 = \max\{C_0^*, C_2\}$ *and* $c_2 = \max\{C_0, C_2^*\}$.

Thus, the following lemma derives a concentration bound for the α-skew KL divergence by plugging the constants in Lemma B.1 to Lemma B.2.

Lemma B.3 (Lee & Shin 2022). *For* $\alpha < \frac{1}{8}$, *the following holds:*

$$\mathbb{P}[|D_{SKL}^{(\alpha)}(p_m^1, p_n^2) - \mathbb{E}[D_{SKL}^{(\alpha)}(p_m^1, p_n^2)]| > \epsilon]$$
$$\le 2\exp\left(-\frac{\epsilon^2}{\frac{2}{m}\log^2(\alpha m) + \frac{2}{\alpha^2 n}\log^2(e^{1/8}n)}\right)$$

Proof. Note that $C_0^* = \log(1/\alpha) - 1 + \alpha \ge C_2 = 1/2$, and $C_0 = 1 - \alpha \le C_2^* = \frac{1-\alpha}{8\alpha}$ for $\alpha < \frac{1}{8}$. Then the concentration bound follows from Lemma B.3. □

Lastly, we present the following upper bound on the bias of the empirical estimator of KLD.

Lemma B.4 (Rubenstein et al. 2019). *Suppose* $p^1 \ll p^2$, *and* $\mathbb{V}_{p^1}[dp^1/dp^2] < \infty$. *Then we have*

$$|\mathbb{E}[D_{KL}(p_m^1, p_n^2)] - D_{KL}(p^1, p^2)| \le \frac{\chi^2(p^1, p^2)}{\min\{n, m\}}.$$

Lemma B.5 (Lee & Shin 2022). *For* $\alpha \in (0, 1)$, *the following holds:*

$$|\mathbb{E}[D_{SKL}^{(\alpha)}(p_m^1, p_n^2)] - D_{SKL}^{(\alpha)}(p^1, p^2)| \le \frac{c(\alpha)}{\min\{n, m\}},$$

where $c(\alpha) := \min\left\{\frac{1}{\alpha}, \frac{\chi^2(p_m^1, p_n^2)}{1-\alpha}\right\}$.

Proof. From Lemma B.4, we have

$$|\mathbb{E}[D_{\mathrm{SKL}}^{(\alpha)}(p_m^1, p_n^2)] - D_{\mathrm{SKL}}^{(\alpha)}(p^1, p^2)| \le \frac{\chi^2(p^1, \alpha p^1 + (1-\alpha)p^2}{\min\{n, m\}},$$

where $\chi^2(p^1, \alpha p^1 + (1-\alpha)p^2) = \int \frac{d^2p^1}{\alpha dp^1 + (1-\alpha)dp^2} \le \int \frac{1}{\alpha} dp^1 = \frac{1}{\alpha}$, or $\int \frac{d^2p^1}{\alpha dp^1 + (1-\alpha)dp^2} \le \frac{1}{1-\alpha}\int \frac{d^2p^1}{dp^2} = \frac{\chi^2(p^1, p^2)}{1-\alpha}$. Therefore, we have

$$|\mathbb{E}[D_{\mathrm{SKL}}^{(\alpha)}(p_m^1, p_n^2)] - D_{\mathrm{SKL}}^{(\alpha)}(p^1, p^2)| \le \frac{c(\alpha)}{\min\{n, m\}}$$

for $c(\alpha) := \min\left\{\frac{1}{\alpha}, \frac{\chi^2(p^1, p^2)}{1-\alpha}\right\}$ □

Now we present the proof of Theorem 1 in the main paper.

Proof. Define

$$B_1 := D_{\mathrm{SKL}}^{(\alpha)}(p_n^1, p_n^2) - \mathbb{E}[D_{\mathrm{SKL}}^{(\alpha)}(p_n^1, p_n^2)]$$
$$B_2 := \mathbb{E}[D_{\mathrm{SKL}}^{(\alpha)}(p_n^1 \| p_n^2)] - D_{\mathrm{SKL}}^{(\alpha)}(p^1, p^2).$$

Then, by using bias-variance decomposition (Pedro, 2000), we have

$$\mathbb{E}_{p^1,p^2}[|D_{\mathrm{SKL}}^{(\alpha)}(p_n^1, p_n^2) - D_{\mathrm{SKL}}^{(\alpha)}(p^1, p^2)|^2]$$
$$= \underbrace{\mathbb{E}_{p^1,p^2}[|B_1|^2]}_{\text{Variance}} + \underbrace{\mathbb{E}_{p^1,p^2}[|B_2|^2]}_{\text{Bias}^2}.$$

Since the following holds for any random variable X,

$$\mathbb{V}(X) = \mathbf{E}[(X - \mathbb{E}X)^2]$$
$$= \int_0^\infty \mathbb{P}[|X - \mathbb{E}X|^2 > t]dt$$
$$= \int_0^\infty \mathbb{P}[|X - \mathbb{E}X| > \sqrt{t}]dt,$$

by Lemma B.2, we have a variance for estimator $D_{\mathrm{SKL}}^{(\alpha)}(p_n^1, p_n^2)$ as follows:

$$\mathbb{V}_{p^1,p^2}[D_{\mathrm{SKL}}^{(\alpha)}(p_n^1, p_n^2)] \le$$
$$\int_0^\infty 2\exp\left(-\frac{t}{\frac{2}{n}\left(\log^2(\alpha n) + \frac{1}{\alpha^2}\log^2(e^{1/8}n)\right)}\right)dt. \tag{8}$$

As we can directly compute the bias term through Lemma B.5, we have

$$\mathbb{E}_{p^1,p^2}[|D_{\mathrm{SKL}}^{(\alpha)}(p_n^1, p_n^2)] - D_{\mathrm{SKL}}^{(\alpha)}(p^1, p^2)|^2]$$
$$\le \underbrace{\frac{c_1(\alpha)}{n^2}}_{\text{from Lemma B.5}} + \underbrace{\frac{c_2\log^2(\alpha n)}{n} + \frac{c_3\log^2(c_4 n)}{\alpha^2 n}}_{\text{from Eq. (8)}}, \tag{9}$$

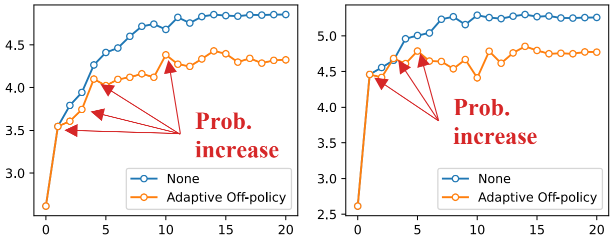
where $c_1(\alpha) = \min\left\{\frac{1}{\alpha^2}, \frac{\chi^2(p^1, p^2)^2}{(1-\alpha)^2}\right\}$ and constants $c_2, c_3, c_4 > 0$ that are independent of n, α, and $D_{\mathrm{KL}}(p^1, p^2)$, where $\chi^2(p^1, p^2) := \mathbb{E}_{p^2}\left[(dp^1/dp^2)^2\right]$. □

C. Details of DistiLLM Algorithm

We describe **DistiLLM** in detail, especially for the adaptive off-policy approach, which could not be fully explained in Sec. 3.2 due to lack of margin.

Instead of using ϕ defined in Sec. 3.2, we further define the probability of using SGOs determined at each validation iteration $\tilde{t}$, denoted as $\phi_{\tilde{t}}$. We adjust the probability by using the following rule with validation loss $\mathcal{L}_{\tilde{t}}$ and $\mathcal{L}_{\tilde{t}-1}$ for iteration $\tilde{t}$ and $\tilde{t}-1$, respectively:

$$\phi_{\tilde{t}} = \begin{cases} \phi_{\tilde{t}-1} & \text{if } \mathcal{L}_{\tilde{t}} \le \mathcal{L}_{\tilde{t}-1} + \varepsilon \\ \min(\phi_{\tilde{t}-1} + 1/N_\phi, 1.0) & \text{otherwise} \end{cases},$$



(a) SKL (b) SRKL

Figure 10. Plot of validation loss values (y-axis) across each validation iteration (x-axis). Although validation loss tends to increase as training progresses, employing SGO effectively prevents this increase. This is the core philosophy of our adaptive SGO scheduler (orange line).

where ε is the loss tolerance, a hyperparameter introduced to mitigate unexpected fluctuations in loss values and enhance the stability of the process. We set this value as 0.1 for all experiments. N_ϕ denotes the total number of stages for adjusting the probability, allowing the probability to adopt any value in the set $\left\{\frac{i}{N_\phi} \mid i = 0, 1, \ldots, N_\phi\right\}$. In our experiments, we set an initial validation loss $\mathcal{L}_0$ as validation loss of initialized student models, ϕ_0 as 0.0, and N_ϕ as 10. If a newly computed validation loss exceeds the preceding one, we increment $\phi_{\tilde{t}}$ by one stage.

Here, we present empirical evidence to justify using validation loss as the primary metric. Fig. 10 shows that the validation loss for both SKL and SRKL consistently increases over iterations, as depicted by the blue lines. This trend indicates that the student models may overly fit to the training dataset and overlook other general sequences, leading to a training-inference mismatch. To create a more versatile student model that performs well on both training and validation datasets, we introduce additional SGOs whenever an increase in validation loss is observed. This approach effectively mitigates the mismatch between training and inference, as well as reduces the risk of noisy feedback.

D. Experimental Setup

We elaborate the detailed experimental setup regarding the datasets used (Sec. D.1), training details (Sec. D.2), and evaluation details (Sec. D.3).

D.1. Dataset Description

We apply **DISTILLM** on various generation tasks including task-agnostic instruction-following, text summarization, and machine translation. We provide detailed descriptions of the datasets used.

- **`databricks-dolly-15k`** (instruction-following, Conover et al. 2023): `databricks-dolly-15k` is an open-source dataset of instruction-following records generated by thousands of Databricks employees in several behavioral categories that are outlined in Ouyang et al. (2022), including brainstorming, classification, closed QA, generation, information extraction, open QA, and summarization.

- **Self-Instruct** (instruction-following, Wang et al. 2022): Self-Instruct is a framework designed to enhance the language model's instruction-following capabilities by leveraging the model's own outputs to generate a vast set of instructional data. It includes 52K instructions and 82K inputs and outputs for tuning, along with 252 expert-written tasks aimed at practical applications and additional 50K examples from public datasets for benchmarking.

- **Vicuna** (instruction-following, Chiang et al. 2023): We also use 80 challenging questions that were used for evaluating Vicuna, following Gu et al. (2024).

- **Super-Natural Instruction** (instruction-following, Wang et al. 2022): Super-Natural Instruction is introduced as a benchmark of 1,616 diverse NLP tasks and their expert-written instructions. The collection covers 76 distinct task types. Its test set consists of 9K samples ranging from 119 tasks.

- **Unnatural Instruction** (instruction-following, Honovich et al. 2023): Unnatural Instruction uses AI to create 240K instructions with little human help, showing that AI-made data can be as good as human-made data for training language models. The core set of this dataset contains 60K samples.

- **SAMSum** (text summarization, Gliwa et al. 2019): SAMSum consists of 16K messenger-like conversations, annotated with a summary for providing a concise overview of the conversation's content by the third person.

- **XSum** (text summarization, Narayan et al. 2018): XSum comprises over 200K news articles, each accompanied by a one-sentence summary designed for the evaluation of abstractive single-document summarization systems, focusing on extreme summarization to capture the essence of articles in a single sentence.

- **CNN/DM** (text summarization, See et al. 2017): CNN/ DM consists of over 300K English news articles that were originally designed for machine-reading and comprehension as well as abstractive question answering, but it now also supports extractive and abstractive summarization.

- **IWSLT 2017** (machine translation, Cettolo et al. 2017): IWSLT 2017 addresses text translation, using a single machine translation system for multiple language directions such as English and German. Here, we specifically focus on an English-to-German (En-De) translation task.

D.2. Training Details

For training the teacher and student models, we used four A100 40GB GPUs for the instruction-following task and four RTX 3090 GPUs for the text summarization and machine translation tasks.

Instruction-following experiments. Our experimental setup for training LMs on `databricks-dolly-15k` primarily follows the experimental setup for Gu et al. (2024). For models within 1B parameters, we search for the learning rates in {5e-4, 1e-4, 5e-5}, the batch sizes in {8, 16, 32} within the possible maximum batch size for A100 40GB GPUs, and train these models for 20 epochs. For models that have more than 1B parameters, we search for the learning rate in {5e-5, 1e-5, 5e-6}, the batch sizes of 8, and train these models for 10 epochs. We fully use the distillation loss for the instruction-following dataset and language modeling loss for OpenWebText (Gokaslan et al., 2019) corpus. The checkpoints of each student are selected by the ROUGE-L scores on the validation set.

For MiniLLM (Gu et al., 2024), we follow the original setup except for the number of GPUs. For ImitKD (Lin et al., 2020), MiniLLM, and GKD (Agarwal et al., 2024), we initialize the student models with the fine-tuned ones according to their original methods, ensuring a fair comparison in our method as well. However, **DistiLLM** demonstrates effectiveness even without the need for such fine-tuned student models, unlike other methods that utilize SGOs. The corresponding results are available in Appendix **??**. We conduct validation at the end of every training epoch. For MiniLLM, we have used the original code [2] (Gu et al., 2024), while for other baselines, we have re-implemented them. To train the OpenLLaMA2 (Geng & Liu, 2023), we use LoRA for query and value weights with a rank of 16.

Task-specific experiments. For text summarization tasks (SAMSum, XSum, and CNN/DM), we train all teacher and student models for 10 epochs. In contrast, for the IWSLT 2017 En-De dataset, we train them for 2 epochs. Since the official code for MiniLLM (Gu et al., 2024) is not available on such tasks, we do not consider conducting experiments with MiniLLM on these tasks. Moreover, as the other methods such as SeqKD (Kim & Rush, 2016), ImitKD (Lin et al., 2020), and GKD (Agarwal et al., 2024) highly increase the training time from obtaining the SGOs or teacher-generated outputs, we only use the 20K of random samples for XSum and CNN/DM, as we described in Sec. 4.2. However, due to the difficulty of machine translation, we use the full dataset of IWSLT 2017. We use a fixed learning rate of 1e-4 and use possible maximum batch size within {8, 32, 64} for RTX 3090 GPUs. We also conducted ten validations for all experiments. For the training teacher model, we utilize LoRA (Hu et al., 2022) for all weights for query, key, value, and output with rank of 16.

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

### Instruction:
{instruction}

### Input:
{input}

### Response:
```

Figure 11. The prompt template for training and evaluation of instruction-following task experiments from Gu et al. (2024).

D.3. Evaluation

For evaluating the teacher and student models, we applied a single A100 40GB GPU for the instruction-following task and a single RTX 3090 GPU for the text summarization and machine translation tasks.

Instruction-following. Our evaluation setup for the instruction-following task also follows the Gu et al. (2024). During the evaluation, we sample the responses from each model using a temperature of 1.0, a max-length limit of 512, and five random seeds (*i.e.,* $\{10, 20, 30, 40, 50\}$). We adopt a prompt wrapper as shown in Fig. 11. However, for GPT-4 feedback, instead of using the prompt introduced in Gu et al. (2024), we use a more popular prompt introduced in Zheng et al. (2023) which is illustrated in Fig. 12 with setting the temperature of 0.7. We also report the ratio of the total score of model responses and ground truth answers by following Gu et al. (2024).

Task-specific experiments. During the evaluation, we sample the responses from each model using a greedy sampling, and a max-length limit of 128. We use ROUGE-L (Lin, 2004) and BLEU (Papineni et al., 2002) for text summarization and machine translation, respectively.

E. Additional Results

In this section, we provide additional experimental results to demonstrate the effectiveness of our proposed method and its components.

E.1. Full Results of Instruction-Following (Fig. 5)

In Fig. 14, we describe the full version of the main result in Sec. 4.1 and Fig. 14. Our proposed **DistiLLM** consistently outperform the baselines such as MiniLLM (Gu et al., 2024) and GKD (Agarwal et al., 2024) in most of the datasets (Dolly Evaluation, Self-Instruct, Vicuna Evaluation, Super-Natural, and Unnatural) and metrics (ROUGE-L and

[2] https://github.com/microsoft/LMOps/tree/main/minillm

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

[Question]
{question}

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

Figure 12. The prompt template for single-answer grading of GPT-4 feedback from Zheng et al. (2023).

Table 7. The full results of Tab. 3, which is the performance comparison of KD methods trained on text summarization and machine translation datasets. We report the ROUGE-L and BLEU scores for the distilled student.

Dataset	SAMSum	XSum	CNN/DM	IWSLT
T5-XL (Teacher)	52.52	30.86	40.84	34.56
T5-XL (3B) → T5-Small (0.06B)				
KD (Hinton et al., 2015)	39.52	21.47	36.43	21.15
SeqKD (Kim & Rush, 2016)	40.24	21.40	36.85	21.42
ImitKD (Lin et al., 2020)	41.44	21.96	37.68	21.52
GKD (Agarwal et al., 2024)	41.92	22.26	37.65	22.04
DistiLLM (ours)	**42.37**	**22.43**	**38.01**	**22.53**
T5-XL (3B) → T5-Base (0.2B)				
KD (Hinton et al., 2015)	46.23	27.15	38.49	29.36
SeqKD (Kim & Rush, 2016)	46.89	27.35	38.65	29.07
ImitKD (Lin et al., 2020)	48.57	28.08	39.16	29.87
GKD (Agarwal et al., 2024)	48.49	28.15	38.98	30.24
DistiLLM (ours)	**49.11**	**28.57**	**39.47**	**30.32**

GPT-4 feedback) regardless of model sizes. The numerical results are reported in Tab. 11, Tab. 12, and Tab. 13. These results demonstrate that **DistiLLM** achieves the best performance (**bold number**) in most cases, except for a few second best performances (<u>underlined number</u>).

E.2. Full Results of Task-Specific KD (Tab. 3)

In addition to the results on SAMSum and IWSLT 2017 datasets in Sec. 4.2 and Tab. 3, we further conduct experiments on XSum and CNN/DM datasets. As the existing KD methods using SGO require up to 5× training time compared to naïve KD, we randomly sample 20K from the training dataset. Tab. 7 further shows the performance of XSum and CNN/DM for **DistiLLM** and other KD baselines. Our **DistiLLM** consistently outperforms other baselines in terms of ROUGE-L and BLEU scores.

Table 8. Comparison of the ROUGE-L score of GPT-2 student using different α. All results show the concave form in terms of ROUGE-L scores.

Loss	α	Dolly	Self	Vicuna	SNI	UNI
SKL	0.0	23.52	11.23	15.92	20.68	23.38
	0.1	24.80	12.86	16.20	26.43	28.06
	0.3	24.54	12.72	16.25	25.31	27.92
	0.5	24.55	12.66	16.05	23.59	27.48
	0.7	24.49	12.08	15.86	21.49	27.15
	0.9	24.27	11.25	14.84	19.07	26.51
SRKL	0.0	23.82	10.90	16.11	22.47	23.03
	0.1	25.21	12.98	15.77	25.83	28.62
	0.3	25.00	12.52	15.53	25.10	27.87
	0.5	25.92	12.09	15.39	23.91	26.98
	0.7	21.41	11.73	14.69	20.65	24.16
	0.9	18.20	9.69	13.71	17.50	19.35

E.3. Terminal Probability by Adaptive SGO Scheduler

In the following list, we report the terminal probability values determined by our adaptive SGO scheduler. We observe that the final probability values are varying across the different tasks. This highlights the importance of using our novel SGO scheduler which can adaptively balance the SGOs and a fixed dataset.

- **GPT-2 family** (`databricks-dolly-15k`): 0.4 (Base), 0.4 (Medium), 0.5 (Large)
- **OPT family** (`databricks-dolly-15k`): 0.2 (125M), 0.3 (350M), 0.5 (1.3B)
- **OpenLLaMA2** (`databricks-dolly-15k`): 0.6 (3B)
- **T5** (SAMSum): 0.7 (Small), 0.8 (Base)
- **T5** (IWSLT 2017): 0.5 (Small), 0.7 (Base)
- **T5** (XSum): 0.3 (Small), 0.6 (Base)
- **T5** (CNN/DM): 0.4 (Small), 0.6 (Base)

E.4. Sensitivity Study for α

As shown in Tab. 8, we also report the numerical values of the results in Fig. 6 for all datasets. The results for Vicuna and Unnatural are similar to the results for Dolly, Self-Instruct, and Super-Natural Instruction datasets (*i.e.,* making concave form in terms of ROUGE-L scores as α increases). These results highly relate to our theoretical analysis in Sec. 3.1 that the normalized approximation error dividing by gradient showed the convexity. This also provides the importance of the skew value α which has an important role in balancing the approximation error and gradient scale.

Table 9. Comparison of diverse scheduling strategy of replay ratio.

Replay ratio ζ	Time	Dolly	Self	Vicuna	SNI	UNI
Constant (0.5)	×2.13	26.32	12.99	17.31	27.45	28.64
Increasing (t/T)	×2.39	**26.44**	12.42	17.60	26.80	**29.90**
Decreasing ($1-t/T$)	×**1.67**	26.11	**13.14**	**18.46**	**27.51**	29.35

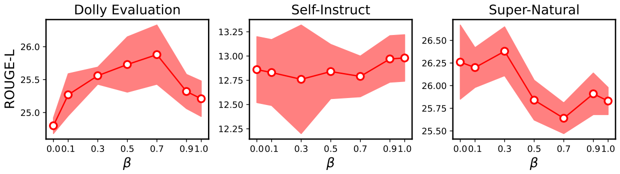


Figure 13. Comparison of the ROUGE-L scores using different β for ISKL (*i.e.,* interpolations between SKL and SRKL).

E.5. Design of Replay Ratio

Tab. 9 summarizes the training time and performance according to the different scheduling strategies for replay ratio in the off-policy training. Despite the high bias error (Lee et al., 2023) of off-policy training, the performance of the constant or increasing ζ is similarly compared to our proposed decreasing manner thanks to our adaptive probability of using SGO. Furthermore, our scheduling strategy not only demonstrates the highest training efficiency but also consistently delivers the best performance overall, achieving the best results in three out of five cases.

E.6. Synergy of SKL and SRKL

Similar to generalized JSD (Agarwal et al., 2024), we consider the interpolation between the SKL and SRKL using the coefficient $\beta \in [0, 1]$. Here, we define interpolated SKL (ISKL) as follows:

$$D^{(\beta)}_{\text{ISKL}}(p, q_\theta) = \beta D^{(0.1)}_{\text{SKL}}(p, q_\theta) + (1-\beta) D^{(0.1)}_{\text{SRKL}}(p, q_\theta).$$

We set the α for 0.1 based on previously described results. Fig. 13 shows the results of ISKL across different β values. Note that we do not use SGO for these results. We observe that there is no consistent tendency between the β and corresponding performance. However, the performance for ISKL for all β still outperforms the other loss functions previously used in auto-regressive KD (*i.e.,* JSD and RKLD).

E.7. Qualitative Evaluation

We provide some responses generated by the models distilled by different methods based on OpenLLaMA2-3B in Tab. 14 and Tab. 15. The prompts are sampled from the `databricks-dolly-15k`, Self-Inst, and Vicuna. Our results demonstrate that **DistiLLM** produces more detailed and accurate responses compared to other KD baselines. Specifically, **DistiLLM** excels at comprehending the precise instructions given (for example, following alphabetical order in Case #1 from Tab. 14, and identifying categorization candidates in Case #2 from Tab. 15). Additionally, we observe that **DistiLLM** achieves high-quality output across various types of tasks, including code generation (as in Case #1 in Tab. 15) and mathematical reasoning (as in Case #3 in Tab. 15).

Table 10. Comparison of the performance of GPT-2 student across different capacities of replay buffer for off-policy approach.

$\mathcal{D}_R$ Capacity	Dolly	Self	Vicuna	SNI	UNI
250	24.71	11.95	16.79	24.88	26.73
500	25.32	12.46	17.64	25.69	27.00
1000	26.11	**13.14**	**18.46**	**27.51**	**29.35**
2000	**26.48**	12.88	17.49	26.51	28.61
4000	25.58	12.65	17.22	25.65	27.55

E.8. Replay Buffer Capacity

We also conduct experiments to confirm the effect of the capacity of the replay buffer on our off-policy approach. Table 10 summarizes the performance associated with different capacities of the replay buffer, $\mathcal{D}_R$. We observe that determining the appropriate capacity involves a trade-off. A capacity that is too small may lead to overfitting on a limited number of samples within $\mathcal{D}_R$. Conversely, a capacity that is too large results in the inclusion of outdated SGO in the replay buffers, potentially introducing a high bias issue as noted by Lee (2023) (Lee et al., 2023). In our experiments, a capacity value of 1000 demonstrates the most balanced performance overall.

Table 11. Comparison with state-of-the-art KD methods, fine-tuned GPT-2 model families (Radford et al., 2019) on `databricks-dolly-15k` dataset. All results are based on our re-implementation. The **bold** and underline markings indicate the best and second-best results, respectively, among those from the same evaluation dataset and student model.

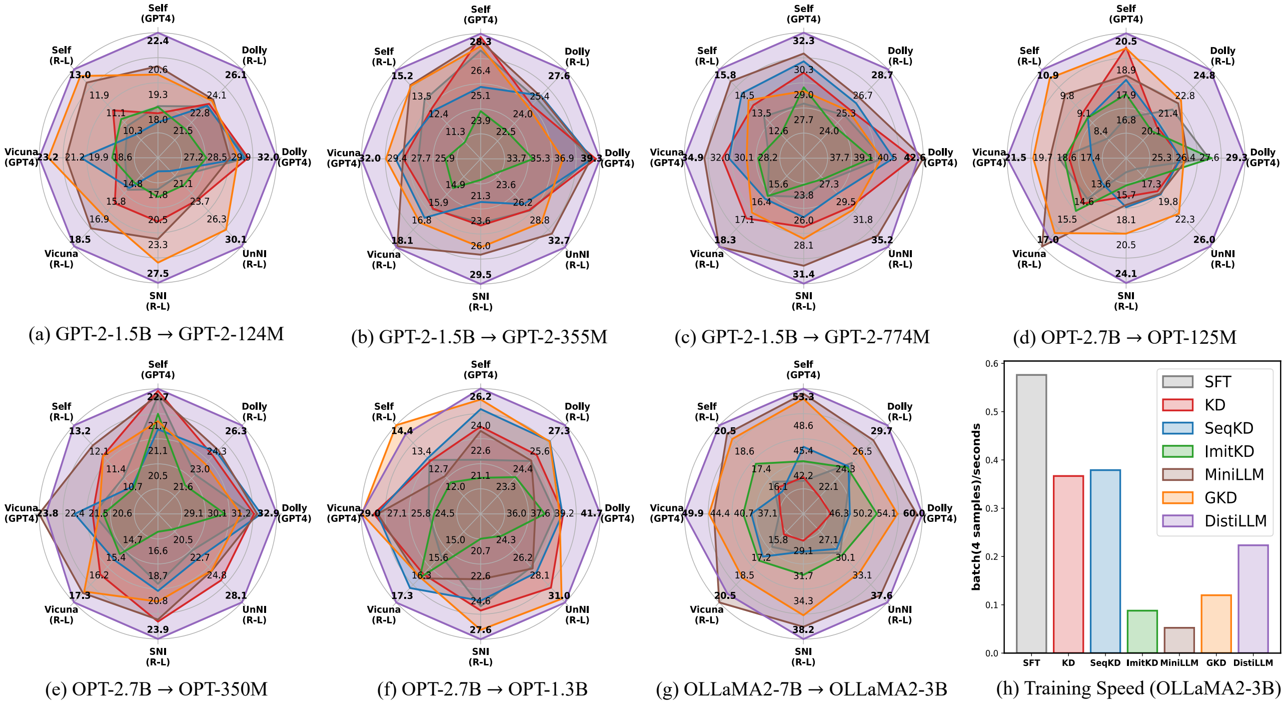
KD	Dolly Evaluation		Self-Instruct		Vicuna Evaluation		Super-Natural	Unnatural
Method	GPT-4 Eval (↑)	ROUGE-L (↑)	GPT-4 Eval (↑)	ROUGE-L (↑)	GPT-4 Eval (↑)	ROUGE-L (↑)	ROUGE-L (↑)	ROUGE-L (↑)
GPT-2 XL (Teacher)	44.91 (0.65)	26.53 (0.35)	34.98 (0.92)	14.41 (0.34)	33.19 (1.13)	16.30 (0.40)	27.76 (0.36)	32.06 (0.12)
GPT-2 XL (1.5B) → GPT-2 (0.1B)								
SFT	30.26 (0.76)	23.33 (0.22)	18.75 (0.55)	10.56 (0.54)	18.43 (0.10)	15.12 (0.47)	17.08 (0.29)	20.07 (0.13)
KD (Hinton et al., 2015)	30.33 (0.59)	23.52 (0.19)	18.39 (0.70)	11.23 (0.41)	18.87 (0.28)	15.92 (0.37)	20.68 (0.14)	23.38 (0.12)
SeqKD (Kim & Rush, 2016)	30.11 (0.35)	23.38 (0.37)	17.94 (0.55)	10.18 (0.18)	18.71 (0.67)	15.01 (0.28)	15.08 (0.12)	19.21 (0.06)
ImitKD (Lin et al., 2020)	27.91 (0.48)	21.63 (0.51)	18.71 (0.42)	10.85 (0.38)	19.14 (0.14)	14.70 (0.29)	17.94 (0.10)	21.24 (0.07)
MiniLLM (Gu et al., 2024)	29.27 (0.41)	23.84 (0.26)	20.76 (0.07)	12.44 (0.28)	20.82 (0.57)	18.29 (0.36)	22.62 (0.26)	23.26 (0.09)
GKD (Agarwal et al., 2024)	29.75 (0.06)	23.75 (0.15)	20.34 (0.47)	12.73 (0.24)	20.48 (0.43)	16.64 (0.24)	26.05 (0.23)	27.70 (0.10)
DistiLLM	**31.97 (0.06)**	**26.11 (0.68)**	**22.45 (1.29)**	**13.14 (0.69)**	**23.21 (1.67)**	**18.46 (0.53)**	**27.51 (0.03)**	**29.35 (0.07)**
GPT-2 XL (1.5B) → GPT-2 Medium (0.3B)								
SFT	38.56 (0.43)	25.23 (0.36)	27.48 (0.51)	13.35 (0.36)	27.88 (1.21)	16.17 (0.50)	23.77 (0.23)	27.27 (0.08)
KD (Hinton et al., 2015)	39.00 (0.77)	24.75 (0.54)	28.17 (0.93)	12.84 (0.48)	28.97 (0.72)	16.14 (0.37)	24.00 (0.23)	27.20 (0.12)
SeqKD (Kim & Rush, 2016)	38.78 (0.52)	25.54 (0.34)	25.62 (0.76)	12.69 (0.46)	29.54 (0.58)	16.61 (0.44)	21.79 (0.22)	26.33 (0.15)
ImitKD (Lin et al., 2020)	34.84 (0.11)	22.60 (0.27)	24.38 (0.07)	10.65 (0.48)	25.90 (0.39)	15.07 (0.39)	19.71 (0.20)	21.81 (0.07)
MiniLLM (Gu et al., 2024)	38.68 (0.13)	25.49 (0.28)	27.96 (0.57)	14.30 (0.49)	29.00 (1.16)	18.09 (0.32)	26.74 (0.22)	30.68 (0.13)
GKD (Agarwal et al., 2024)	36.82 (0.79)	24.51 (0.32)	27.70 (0.25)	14.26 (0.29)	30.25 (1.16)	16.86 (0.36)	26.05 (0.23)	29.05 (0.11)
DistiLLM	**39.28 (0.39)**	**27.62 (0.28)**	**28.34 (0.29)**	**15.24 (0.43)**	**32.05 (1.01)**	**18.14 (0.28)**	**29.50 (0.24)**	**32.71 (0.09)**
GPT-2 XL (1.5B) → GPT-2 Large (0.8B)								
SFT	40.27 (0.36)	25.19 (0.17)	28.58 (0.05)	13.40 (0.35)	28.07 (0.14)	16.10 (0.41)	23.89 (0.23)	26.82 (0.12)
KD (Hinton et al., 2015)	42.03 (1.26)	26.51 (0.47)	30.20 (0.90)	13.99 (0.58)	31.82 (0.28)	17.07 (0.20)	26.48 (0.31)	30.12 (0.09)
SeqKD (Kim & Rush, 2016)	40.88 (0.58)	26.11 (0.31)	30.79 (1.42)	14.61 (0.36)	31.29 (0.32)	16.23 (0.39)	25.64 (0.18)	28.40 (0.06)
ImitKD (Lin et al., 2020)	39.60 (0.39)	24.36 (0.58)	29.43 (0.76)	12.48 (0.29)	28.74 (0.49)	16.11 (0.39)	22.84 (0.31)	26.51 (0.16)
MiniLLM (Gu et al., 2024)	**42.62 (0.26)**	26.45 (0.48)	31.20 (0.23)	15.23 (0.22)	33.27 (0.09)	**18.29 (0.32)**	29.81 (0.28)	33.80 (0.13)
GKD (Agarwal et al., 2024)	40.18 (0.77)	25.52 (0.40)	29.23 (0.27)	14.25 (0.25)	29.86 (0.46)	16.82 (0.39)	27.50 (0.16)	30.48 (0.10)
DistiLLM	42.34 (1.09)	**28.68 (0.33)**	**32.29 (0.25)**	**15.85 (0.44)**	**34.89 (1.29)**	18.28 (0.33)	**31.35 (0.12)**	**35.23 (0.10)**

Table 12. Comparison with state-of-the-art KD methods, fine-tuned OPT model families (Zhang et al., 2022) on `databricks-dolly-15k` dataset. All results are based on our re-implementation. The **bold** and underline markings indicate the best and second-best results, respectively, among those from the same evaluation dataset and student model.

KD	Dolly Evaluation		Self-Instruct		Vicuna Evaluation		Super-Natural	Unnatural
Method	GPT-4 Eval (↑)	ROUGE-L (↑)	GPT-4 Eval (↑)	ROUGE-L (↑)	GPT-4 Eval (↑)	ROUGE-L (↑)	ROUGE-L (↑)	ROUGE-L (↑)
OPT-2.7B (Teacher)	43.15 (0.57)	26.18 (0.21)	25.62 (0.96)	11.19 (0.29)	30.48 (0.61)	15.48 (0.37)	19.19 (0.11)	22.65 (0.14)
OPT-2.7B (2.7B) → OPT-125M (0.1B)								
SFT	27.56 (0.06)	21.78 (0.19)	16.97 (0.54)	8.09 (0.39)	18.93 (0.12)	14.40 (0.17)	13.45 (0.20)	15.08 (0.05)
KD (Hinton et al., 2015)	26.26 (0.11)	20.54 (0.38)	19.91 (0.02)	9.16 (0.29)	18.42 (0.28)	14.65 (0.47)	15.79 (0.26)	18.23 (0.09)
SeqKD (Kim & Rush, 2016)	26.22 (0.09)	20.72 (0.59)	18.55 (0.25)	8.94 (0.37)	17.29 (0.62)	13.56 (0.33)	16.80 (0.36)	18.68 (0.16)
ImitKD (Lin et al., 2020)	27.69 (0.21)	20.16 (0.19)	17.95 (0.25)	8.95 (0.50)	18.66 (0.56)	15.05 (0.52)	14.72 (0.21)	16.55 (0.10)
MiniLLM (Gu et al., 2024)	26.56 (0.26)	22.24 (0.32)	18.73 (0.22)	9.92 (0.47)	19.39 (0.08)	**16.97 (0.49)**	16.58 (0.19)	18.59 (0.09)
GKD (Agarwal et al., 2024)	26.37 (0.24)	22.46 (0.27)	19.87 (0.16)	10.59 (0.36)	20.23 (0.74)	16.25 (0.65)	19.33 (0.20)	21.40 (0.16)
DistiLLM	**29.30 (0.02)**	**24.75 (0.31)**	**20.50 (0.18)**	**10.93 (0.49)**	**21.54 (0.12)**	16.55 (0.38)	**24.06 (0.26)**	**26.02 (0.11)**
OPT-2.7B (2.7B) → OPT-350M (0.3B)								
SFT	31.68 (0.05)	22.58 (0.38)	22.52 (0.36)	11.07 (0.31)	21.68 (0.43)	15.10 (0.31)	19.33 (0.43)	21.69 (0.04)
KD (Hinton et al., 2015)	31.81 (0.10)	24.01 (0.27)	22.65 (0.05)	11.97 (0.29)	21.72 (0.19)	16.12 (0.45)	22.50 (0.16)	25.39 (0.12)
SeqKD (Kim & Rush, 2016)	32.13 (0.17)	24.30 (0.20)	21.66 (1.58)	10.69 (0.45)	22.42 (0.48)	15.51 (0.18)	19.93 (0.31)	22.58 (0.15)
ImitKD (Lin et al., 2020)	30.48 (0.31)	21.77 (0.72)	22.06 (0.64)	10.62 (0.28)	21.30 (0.18)	15.27 (0.51)	14.97 (0.20)	19.12 (0.16)
MiniLLM (Gu et al., 2024)	31.95 (0.18)	24.44 (0.20)	22.54 (0.23)	12.41 (0.40)	**23.81 (0.32)**	16.89 (0.34)	22.33 (0.17)	24.20 (0.08)
GKD (Agarwal et al., 2024)	31.21 (0.63)	23.39 (0.34)	21.87 (0.40)	11.96 (0.54)	21.46 (0.10)	16.83 (0.49)	20.82 (0.24)	24.18 (0.19)
DistiLLM	**32.85 (0.81)**	**26.33 (0.26)**	**22.70 (0.57)**	**13.24 (0.29)**	23.54 (0.69)	**17.28 (0.23)**	**23.95 (0.17)**	**28.10 (0.11)**
OPT-2.7B (2.7B) → OPT-1.3B (1.3B)								
SFT	38.65 (0.18)	24.97 (0.33)	22.06 (0.07)	13.08 (0.29)	25.41 (0.10)	15.52 (0.48)	24.99 (0.17)	27.18 (0.16)
KD (Hinton et al., 2015)	39.17 (0.28)	25.36 (0.35)	23.99 (0.48)	13.04 (0.62)	28.36 (0.88)	16.21 (0.50)	25.33 (0.21)	29.41 (0.13)
SeqKD (Kim & Rush, 2016)	39.12 (0.46)	26.26 (0.28)	25.01 (0.26)	13.15 (0.19)	28.18 (1.45)	16.73 (0.52)	24.56 (0.17)	27.76 (0.06)
ImitKD (Lin et al., 2020)	37.91 (0.57)	23.92 (0.47)	21.01 (0.52)	12.21 (0.59)	25.11 (0.61)	16.35 (0.38)	22.86 (0.28)	27.33 (0.16)
MiniLLM (Gu et al., 2024)	37.27 (0.74)	24.99 (0.41)	23.74 (0.91)	12.45 (0.96)	28.15 (0.65)	16.37 (0.17)	22.86 (0.28)	27.33 (0.16)
GKD (Agarwal et al., 2024)	38.93 (1.12)	26.24 (0.20)	25.57 (0.33)	**14.43 (0.75)**	29.00 (0.50)	16.33 (0.37)	26.86 (0.20)	30.62 (0.26)
DistiLLM	**41.69 (0.35)**	**27.30 (0.25)**	**26.21 (1.09)**	14.06 (0.64)	**29.04 (1.14)**	**17.27 (0.29)**	**27.60 (0.16)**	**31.02 (0.09)**

Table 13. Comparison of state-of-the-art KD methods using OpenLLaMA2-7B (Geng & Liu, 2023) and OpenLLaMA2-3B as teacher and student models, respectively. We fine-tune the models on `databricks-dolly-15k` dataset. All results are based on our re-implementation. The **bold** and underline markings indicate the best and second-best results, respectively, among those from the same evaluation dataset and student model.

KD	Dolly Evaluation		Self-Instruct		Vicuna Evaluation		Super-Natural	Unnatural
Method	GPT-4 Eval (↑)	ROUGE-L (↑)	GPT-4 Eval (↑)	ROUGE-L (↑)	GPT-4 Eval (↑)	ROUGE-L (↑)	ROUGE-L (↑)	ROUGE-L (↑)
OpenLLaMA2-7B (Teacher)	56.86 (0.63)	27.59 (0.32)	54.11 (1.38)	18.99 (0.61)	46.65 (0.25)	17.42 (0.57)	31.26 (0.17)	31.14 (0.07)
SFT	47.27 (0.17)	25.11 (0.58)	41.66 (0.28)	16.52 (0.56)	35.23 (0.33)	16.33 (0.39)	29.28 (0.45)	29.17 (0.12)
KD (Hinton et al., 2015)	44.92 (0.56)	20.95 (0.49)	42.13 (0.06)	16.12 (0.80)	35.06 (0.17)	15.39 (0.39)	27.93 (0.26)	25.16 (0.22)
SeqKD (Kim & Rush, 2016)	48.05 (0.64)	24.67 (0.47)	45.96 (0.13)	15.83 (0.62)	39.57 (1.83)	17.06 (0.47)	29.06 (0.28)	28.56 (0.05)
ImitKD (Lin et al., 2020)	52.48 (1.51)	24.53 (0.26)	44.14 (1.12)	17.80 (0.80)	40.82 (1.08)	17.36 (0.22)	31.50 (0.12)	29.54 (0.13)
MiniLLM (Gu et al., 2024)	59.01 (1.64)	27.88 (0.28)	52.66 (1.66)	19.94 (0.58)	45.33 (0.92)	**20.50 (0.41)**	36.91 (0.28)	36.33 (0.14)
GKD (Agarwal et al., 2024)	55.86 (0.94)	26.30 (0.31)	52.01 (1.03)	19.56 (1.03)	46.08 (0.17)	18.66 (0.34)	35.71 (0.20)	32.57 (0.11)
DISTILLM	**59.94 (1.19)**	**29.73 (0.52)**	**53.29 (0.75)**	**20.39 (0.56)**	**49.88 (0.46)**	19.62 (0.41)	**37.64 (0.18)**	**37.56 (0.08)**



(a) GPT-2-1.5B → GPT-2-124M

(b) GPT-2-1.5B → GPT-2-355M

(c) GPT-2-1.5B → GPT-2-774M

(d) OPT-2.7B → OPT-125M

(e) OPT-2.7B → OPT-350M

(f) OPT-2.7B → OPT-1.3B

(g) OLLaMA2-7B → OLLaMA2-3B

(h) Training Speed (OLLaMA2-3B)

Figure 14. Instruction-following tasks, distilling GPT-2 (Radford et al., 2019), OPT (Zhang et al., 2022), and OpenLLaMA (OLLaMA; Geng & Liu 2023) model families on `databricks-dolly-15k` dataset. GPT4 and R-L indicate GPT-4 feedback (Zheng et al., 2023) and ROUGE-L (Lin, 2004), respectively. To evaluate training speed, results are obtained using four A100 GPUs. In SeqKD (Kim & Rush, 2016), we also consider the generation time for teacher-generated data.

Table 14. Instruction-following cases from the `databricks-dolly-15k` (Conover et al., 2023) dataset.

	Case # 1
Instruction	Extract the names of the cities and rank them in alphabetical order?
Input	The United Kingdom is a constitutional monarchy and parliamentary democracy.[m][25] Its capital and largest city is London, the capital of England, a global city and financial centre with a population of over 14 million people. Edinburgh, Cardiff and Belfast are the national capitals of Scotland, Wales and Northern Ireland respectively.
Ground-truth	The cities listed in alphabetical order are Belfast, Cardiff, Edinburgh and London.
SFT ✗	Alphabetical order of cities in the United Kingdom: \n\n London, Edinburgh, Cardiff, Belfast, Glasgow, Southampton, Birmingham, Coventry, Leicester, Norwich, Swindon, Cambridge
KD ▲	London, Edinburgh, Cardiff, Belfast
SeqKD ✗	Edinburgh, Cardiff, Belfast are the national capital of Scotland, Wales and North Irelandrespectively.
ImitKD ▲	London.Edinburgh.Cardiff.Belfast.
MiniLLM ▲	London, Edinburgh, Cardiff, Belfast
GKD ▲	London, Edinburgh, Cardiff, Belfast
DistiLLM ✓	The following are the names of the cities in alphabetical order: \n \n Belfast\n Cardiff\n Edinburgh\n London
	Case # 2
Instruction	Extract all of the dates mentioned in this paragraph and list them using bullets in the format Date - Description
Input	Daniel Leavitt was born November 16, 1813, at Rye, New Hampshire, the son of Benning Leavitt, influential businessman, state senator, county commissioner and later Chicopee Selectman,[5] and his wife Olive (Jenness) Leavitt. Daniel Leavitt married in 1838 at West Springfield, Massachusetts, Ruth Jeannette Ball. They had three children.
Ground-truth	• November 16, 1813 - This is when Daniel Leavitt was born \n • 1838 - This is when Daniel Leavitt married his wife Ruth Jeannette Ball.
SFT ✗	• November 16,1813 - Daniel Leavitt was born\n• November 17, 1814 - His mother Olive Leavitt died\n • November 1, 1815 - His father Benning Leavitt died\n * December 24, 1818 - Daniel and Ruth Jeannette Ball were married\n * March 31, 1838 - They got married in March 31,1838\n * May 21, 1840 - Their third child was born\n * May 31, 2018 - Daniel and his children restored their home Rye\n * November 1, 2021 - Daniel Leavitt died
KD ✗	- November 16,
SeqKD ✗	• November 16, 2013 - Daniel Leavitt was born on this date.\n • September 27, 1838 - Daniel married at West Springfield, Massachusetts.
ImitKD ▲	- November 16,1813 - Day when he was born\n - West Springfield, Massachusetts - Marriage ceremony\n - September 5,1857 - Year he died
MiniLLM ▲	• November 16,1813 - Daniel Leavitt was born at Rye, New Haven, the son of Bening Leavitt, influential Businessman, state senator and County Commissioner, and his wife Olive (jenness) Leavit.\n • November 19,1838 - Daniel Leavit married at West Springfield, Massachusetts.\n • November 23,1913 - Daniel Leavy died at Chicopee, Massachusetts.
GKD ▲	• November 16,1813 - at Rye, New hampshire\n • July 2,1838 - at West Springfield, Massachusetts
DistiLLM ✓	• November 16,1813 - Daniel Leavitt was born at Rye, New Hampsire\n • November 1838 - Daniel Leavitt married Ruth Jeannette Ball
	Case # 3
Instruction	Which Dutch artist painted “Girl with a Pearl Earring”?
Input	
Ground-truth	Vermeer
SFT ✗	Dimitri Heidelbauer
KD ✗	Malahema
SeqKD ✗	Dutch artist, Mondrian, painted the Girl with a Pearl Earring.
ImitKD ✗	Malahema Vanpolder
MiniLLM ✗	Vincent Van Gogh
GKD ▲	Grachtengordel, which is commonly known as “Girl with a pearl earring” is an oil painting by Johannes Vermeer. It was first exhibited in June 1665 in the Hofje van Ca’ra van Oostende, which was in the city of 5de Oostende.
DistiLLM ✓	Johannes Vermeer

Table 15. Instruction-following cases from the Self-Instruct (Wang et al., 2023b) and Vicuna (Chiang et al., 2023). Note that the ground-truth for case #1 is for Python, not for the bash command. However, only **DistiLLM** generates the simple correct responses.

Case # 1 (Self-Instruct)	
Instruction	Convert the given description to a bash command.
Input	read the first 10 lines from f
Ground-truth	f.readlines(10)
SFT ▲	tail -n 10 f
KD ✗	f -n 10
SeqKD ✗	cat f
ImitKD ✗	cat f — nl 10 \n \n \n Writing data into the file \n \n \t Christina writes the first 10 documents to back t \n \n 10 documents written \n \n Christina reads the first 10 files from t \n \n Christina writes the first ten ABC documents to t rw ec \n \n 10 documents read
MiniLLM ✓	cat f — head -10
GKD ▲	echo -e '\n read – we read from stdin\n f – the file\n ' — read -n 10 f
DistiLLM ✓	head -n 10 f
Case # 2 (Self-Instruct)	
Instruction	Categorize the Business into one of the Restaurants, Home Services, Auto Services and Miscellaneous based on its Specialties.
Input	Call us at 650-636-4884 or visit our website to receive a quote. This shop specializes in New Tires and General Auto Repair. We carry all tires in-house and have a wide range to fit any budget or car specifics. If you are unsure what tires you need there are experts on hand and available to help you pick the best tire for your needs. We also carry commercial vehicle tires and can serve a wide array of fleets.
Ground-truth	Auto Services
SFT ✗	This shop specializes in New Tases and General Auto Repair.
KD ✗	Home Services, Auto Services and restaurant will be Restaurant because it focuses on Restaurants while automotive store and general auto repair are both services and restaurant would therefore not belong in either category.
SeqKD ▲	The types of businesses listed here are Auto Services, Home Services and Miscellaneous. Because this business has multiple types it is considered a Mischangeluz.
ImitKD ✗	Restaurants - Home Services - Auto Services - Miscellaneous
MiniLLM ✗	Restaurants, Home Services, AutoServices and Miscellaneous
GKD ✗	Options: Restaurants, Home Services, Car Services, Miscellaneous
DistiLLM ✓	Specialties: Auto Services
Case # 3 (Vicuna)	
Instruction	Given that f(x) = 5x^3 - 2x + 3, find the value of f(2).
Input	
Ground-truth	To find the value of f(2), we simply substitute 2 in place of x in the given equation of f(x): \n \n f(2) = 5(2)^3 - 2(2) + 3 \n \n f(2) = 5(8) - 4 + 3 \n \n f(2) = 40 - 1 \n \n Therefore, the value of f(2) is 39.
SFT ✗	We know that \(f(2) = 5(2)^3 - 2(2) + 3 = 1028 - 2 = 1021, \hospital doors 5 meters apart. The width of the hospital doors should be \(\frac{1021}{5}\) = 202.8 inches. At least 202 hospital doors should be narrow enough to fit the estimated population of f(2)iashti2.
KD ✗	Given f(x) = (5x^3)^–3 – 2x^3 – 3, when f(2) is given, substitute the value of x to derive f(2) back. In the given expression,\n f(2) is obtained by saying multiplication of 5 and 2 over x^2 so that\n f(x) = (x^2 * 5)^–3 – 3 (adding significance over the parenthesis)\n So, f(2) is obtained as\n \t f(2) = (2 * 5)^(–3) – 3 \n So, final answer for f(2) is \n \t f(2)-eq (2 * 5)^(-3) – 3 \n Being little bit educated, it can be sequenced solution chain also. So, \n \t f(2)\{= 5\} ×(-3) \+ 2 ×[-3]3 * 2 α/2 \n or \n \t f(2)= 6 -2 + 3 \n Finding the value of f(x) is a linear inversion case.
SeqKD ✗	We can substitute for x in f(x) to get f(2) = 5(2 squared) + 2 = 10 and then f(10) = 5^3 + 2 = 25, which is not an integer. Thus, f(2) cannot be an integer, and therefore there isn't a single positive integer that corresponds to f(2).
ImitKD ✗	The polynomial f(x) can be rewritten as:\n f(x) = 2x^4 + 5x^2 + 3.\n \n By straightforward algebraic manipulation, we can see that f(2) = -1 \n therefore f(2) = 2 + 5 * 2 + 3 = -1
MiniLLM ✗	f(2) = 2(5^3 - 2) + 3
GKD ✗	The polynomial f(x) can be rewritten as:\n 5x^3 -2x + 3 = (5x)^2 - 2(5x) + 3 - (2x)^2 + 3\n =¿5(x^2 - 2) + 3 = 0\n =¿x^2 - 6 = 0
DistiLLM ✓	f(2) = 5(2)^3 - 2(2) + 3 = 40 - 4 + 3 = 39