KNOWLEDGE DISTILLATION FOR CLOSED-SOURCE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Closed-source language models such as GPT-4 have achieved remarkable performance. Recently, many studies have focused on enhancing the capabilities of smaller models, through knowledge distillation (KD) on those closed-source language models. However, due to the inability to directly access the closed-source language model's output distribution, KD methods can currently only be performed using one-hot labels, which hinders the effectiveness of KD. To address this limitation, we propose a Bayesian estimation-based knowledge distillation method. Specifically, our method comprises prior estimation and posterior estimation. The prior estimation obtains a prior distribution by leveraging the corpus generated by the closed-source language model. The posterior estimation updates the prior distribution to obtain a posterior distribution, based on continued sampling results. Then we utilize the prior and posterior distributions for distillation. Experimental results showcase that, in the context of KD for closed-source language model, our method outperforms the current KD methods that directly fine-tune on the one-hot labels.

1 INTRODUCTION

While closed-source large language models (LLMs) such as GPT-3.5 and GPT-4 have shown great superiority over open-source counterparts like LLaMA(Touvron et al., 2023) and Falcon(Penedo et al., 2023), they can only be accessed via API calls and allow limited customization and transparency. One way to address this problem is to transfer their capabilities to open-source language models, typically smaller in size, by prompting closed-source LLMs to generate samples that reflect their capabilities and fine-tuning open-source language models on the generated one-hot labels.

Knowledge distillation (KD) (Hinton et al., 2015) is an effective technology that aims to obtain a small but strong student model by distilling knowledge from a large teacher model. The objective function in Hinton et al. (2015) involves calculating the Kullback-Leibler (KL) divergence between the output distributions of the teacher model and the student model. By minimizing the KL divergence, the student model is able to mimic the behavior and learn the intrinsic knowledge of the teacher model. However, many current methods (Hsieh et al., 2023; Jiang et al., 2023; Ho et al., 2022) that perform KD on the closed-source LLMs involves solely fine-tuning student model on one-hot labels generated by the teacher model, as illustrated in Figure1. In contrast to using output distribution (soft labels) to compute KL divergence, transferring deeper and more fundamental knowledge from teacher model to student model is constrained when relying solely on fine-tuning with one-hot labels. This represents a limitation in current KD methods for closed-source LLMs.

To address this limitation, we propose Bayesian estimation-based knowledge distillation to perform effective knowledge distillation on closed-source language model (LM). Our method first estimates the inaccessible output distribution (referred as to latent distribution) of closed-source LM, and then performs KD on the estimated distribution. Our approach comprises two main components: prior estimation and posterior estimation. (1) The prior estimation is designed to estimate the latent distribution by leveraging corpus generated by the closed-source LM. Our hypothesis is that within the generated corpus, there are underlying patterns that characterize the latent distribution. Through prior estimation, a prior distribution that approximates the latent distribution can be obtained. (2) By continuously sampling from a proxy of the closed-source LM, posterior estimation derives a posterior distribution to approximate the latent distribution. Then we perform KD on these esti-

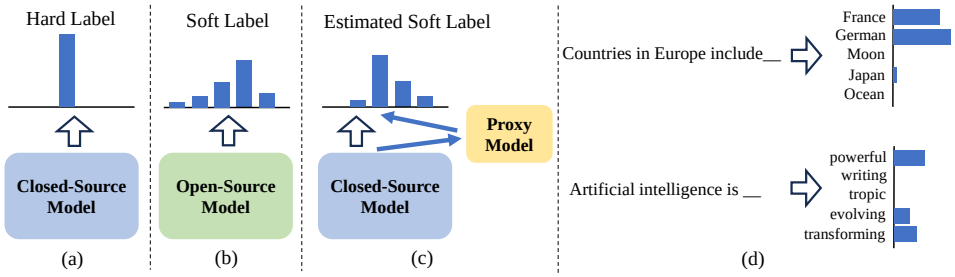

Figure 1: (a) In knowledge distillation of closed-source models, only one-hot labels (hard labels) can be obtained. (b) In knowledge distillation of open-source models, output distributions (soft labels) can be obtained. (c) Our method obtains estimated soft labels from closed-source models by leveraging a proxy model. (d) Compared to hard labels, soft labels allow students to learn more profound knowledge by guiding them to learn from multiple valid targets.

mated distributions. The utilization of the estimated distributions enables student model to tap into more profound and essential aspects of the closed-source teacher model's knowledge during distillation process. It fosters a more comprehensive and insightful learning experience compared to the previous closed-source KD paradigm relying solely on one-hot labels.

We conduct extensive experiments with LLaMA (Touvron et al., 2023) across various representative benchmarks, such as BBH(Suzgun et al., 2022), AGIEval(Zhong et al., 2023), ARCClark et al. (2018), MMLU(Hendrycks et al., 2021), CSQA(Talmor et al., 2019) and GSM8K(Cobbe et al., 2021). In the context of KD for closed-source LM, the empirical results demonstrate the effectiveness of our method over directly fine-tuning on the one-hot labels. For example, our method achieves an average accuracy improvement across the six benchmarks from 36.31% to 39.43% with LLaMA-7B, over methods that solely fine-tune on one-hot labels. These findings provide compelling evidence of the effectiveness of the proposed method.

2 Related Work

The concept of knowledge distillation (KD) was originally introduced by Hinton et al. (2015) with the aim of transferring the knowledge from a teacher model to a smaller student model. This knowledge transfer is achieved by minimizing KL divergence between output distributions of the teacher model and the student model. Current KD methods can be categorized into two primary types: knowledge distillation for open-source models and knowledge distillation for closed-source models.

2.1 Open-Source Knowledge Distillation

Knowledge distillation was first applied to distilling open-source models. For instance, Sanh et al. (2019) applies KD to the pre-training process of BERT (Devlin et al., 2019), yielding smaller models with minor performance drops. Jiao et al. (2020) allows the student model's intermediate features to mimic the teacher model's intermediate features, by minimizing the Mean Squared Error (MSE) loss function. Other approaches, such as the one proposed by Gu et al. (2023), focus on distilling open-source generative language model like LLaMA. Additionally, Park et al. (2019) leverages sample-wise relative information within the teacher model to perform knowledge distillation on ResNet (He et al., 2016). Mirzadeh et al. (2019) introduces an intermediate network to bridge the parameter size gap between the CNN teacher model and the CNN student model. However, it's important to note that in all these methods, the student model needs access to the internal features or parameters of the teacher model, which is not feasible in the context of distilling closed-source LM.

2.2 Closed-Source Knowledge Distillation

Given the outstanding performance of current SOTA closed-source LLMs like GPT-3.5 and GPT-4, many studies have shifted their focus towards transferring knowledge from these closed-source LLMs into smaller models. Some approaches, such as Hsieh et al. (2023); Ho et al. (2022); Mukherjee et al. (2023) utilize rationales generated by closed-source LLMs as training data. They then perform fine-tuning on these generated rationales to transfer the teacher model's reasoning abilities

Notations	Descriptions
$\mathcal{C}$	Corpus generated by the closed-source language model
$\mathbb{V}$	Vocabulary of language model
$\mathcal{M}$	Proxy model
I	Input instruction
w_t	The t^{th} response token, $w_t \in \mathbb{V}$
Q_{w_t}	Probability $\Pr(w_t \mid w_{t-1}, \ldots, w_1, I)$ in the student model
$P^*_{w_t}$	Probability $\Pr(w_t \mid w_{t-1}, \ldots, w_1, I)$ in the closed-source model
P_{w_t}	Random variable associated with the value of $P^*_{w_t}$
Y	Discrete random event, $Y \in \{0, 1\}$
$f_{W_t}(P_{w_t})$	Probability dense function of P_{w_t}
$f_{W_t \mid Y}(P_{w_t} \mid Y)$	Conditional probability dense function of P_{w_t} given event Y
$\mathbb{E}(P_{w_t})$	Prior probability
$\mathbb{E}(P_{w_t} \mid \mathcal{M})$	Posterior probability

Table 1: Notations and descriptions.

into the student model. To enhance the student's capabilities, Jiang et al. (2023), for instance, identifies challenging samples and has the closed-source teacher generate more to fine-tune the student.

However, in the context of knowledge distillation for closed-source LM, most existing methods stop at fine-tuning on the teacher-generated one-hot labels. Our work, on the other hand, focuses on distilling knowledge from the closed-source LM more efficiently by estimating the latent distribution. We achieve this by introducing Bayesian estimation-based methods to soften the one-hot labels provided by the closed-source teacher. We enhance the effectiveness of knowledge transferring from the closed-source teacher model to the student model, by minimizing the KL divergence between the output distribution of the student model and the estimated output distribution.

3 METHOD

We present Bayesian estimation-based knowledge distillation to enhance the efficiency of knowledge distillation for closed-source LM.

3.1 PROBLEM STATEMENT

In this section, we first provide notations in Table 1. We consider a language model with vocabulary $\mathbb{V}$, takes an instruction I as input and generates response tokens $w_1, w_2, w_3 \ldots$ as output. At time t, the probability of generating token w_t can be represented as $\Pr(w_t \mid w_{t-1}, \ldots, w_1, I)$. We refer the distribution as the probabilities $\Pr(w_t \mid w_{t-1}, \ldots, w_1, I)$ encompassing all words within vocabulary $\mathbb{V}$. Let $P^*_{w_t}$ be the probability $\Pr(w_t \mid w_{t-1}, \ldots, w_1, I)$ in closed-source LM, then token-level objective function of KD for the closed-source LM at time t can be derived as follows:

$$\mathcal{L}_t^{\text{kl}} = \sum_{w_t \in \mathbb{V}} P^*_{w_t} \log \frac{P^*_{w_t}}{Q_{w_t}} \tag{1}$$

Where the Q_{w_t} is the probability $\Pr(w_t \mid w_{t-1}, \ldots, w_1, I)$ in student model. Due to the inaccessibility of $P^*_{w_t}$, this objective function degrades to computing cross entropy with one-hot labels, which might limit the performance of KD. To this end, our goal is to estimate a probability to approximate the $P^*_{w_t}$ (referred to as latent probability). Subsequently, we perform KD on the estimated probabilities. The overall architecture of our method is shown in Figure 2.

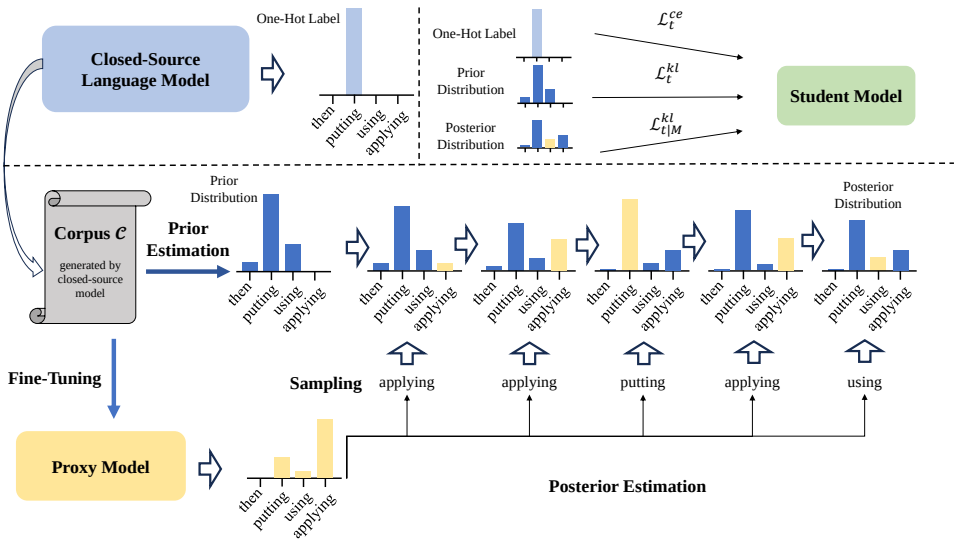

Figure 2: Overview of our method. We first obtain prior distribution through the prior estimation. Then in the posterior estimation, the prior distribution is updated through iterative sampling from a proxy of the closed-source LM. The final objective function involves three targets: one-hot label, prior distribution, and posterior distribution.

3.2 Estimation Methods

3.2.1 Prior Estimation

In this section, we elaborate on the proposed prior estimation method. The prior estimation aims to estimate a probability to approximate the latent probability $P^*_{w_t}$ at each time step t. Given sequence $(w'_t, w'_{t-1}, \ldots, w'_1, I)$, the prior estimation aims to inform, at time step t, a high probabilities for the student model to generate the ground-truth token w'_t while still allowing for some probability of other valid tokens. Given corpus $\mathcal{C}$ generated by the closed-source LM, for a specific sequence $(w'_t, w'_{t-1}, \ldots, w'_1, I) \in \mathcal{C}$, and for $\forall w_t \in \mathbb{V}$, if $w_t = w'_t$, then the value of $\Pr(w_t|w'_{t-1}, \ldots, w'_1, I)$ can be computed as:

$$p_{w_t} = \frac{\#(w_t, w'_{t-1}, \ldots, w'_{t-n})}{\gamma\#(w'_{t-1}, \ldots, w'_{t-n})} + \frac{\gamma - 1}{\gamma} \tag{2}$$

If $w_t \neq w'_t$, then the value of $\Pr(w_t|w'_{t-1}, \ldots, w'_1, I)$ can be computed as:

$$p_{w_t} = \frac{\#(w_t, w'_{t-1}, \ldots, w'_{t-n})}{\gamma\#(w'_{t-1}, \ldots, w'_{t-n})} \tag{3}$$

Where the $\#$ represents the count of a particular response tokens sequence appears in $\mathcal{C}$. The n is the window size. The γ is a hyperparameter, $\gamma \in \mathbb{Z}^+$. The γ is used to adjust the dominant probability contribution of the ground-truth token w'_t. For instance, when $\gamma = 2$, term $\frac{\gamma-1}{\gamma}$ ensures that the probability $\Pr(w'_t|w'_{t-1}, \ldots, w'_1, I)$ of generating ground-truth token w'_t is greater than 50%.

An assumption behind the prior estimation is that language models typically generate the next token with a strong association to the most recent preceding tokens. Through Equation 2 and Equation 3, we obtain a scalar probability value p_{w_t}. We consider the value of $P^*_{w_t}$ as a continuous random variable denoted as P_{w_t}, $P_{w_t} \in [0, 1]$, with probability density function $f_{W_t}(P_{w_t})$. The $f_{W_t}(P_{w_t})$ can be predefined in a way that the expected value of P_{w_t} is equal to the previously computed scalar p_{w_t}. Then a prior probability for approximating the latent probability $P^*_{w_t}$ can be obtained by calculating the expectation of P_{w_t} (replace P_{w_t} with x):

$$\mathbb{E}(P_{w_t}) = \int_0^1 x f_{W_t}(x)dx = p_{w_t} \tag{4}$$

3.2.2 POSTERIOR ESTIMATION

The posterior estimation is based on the prior estimation to estimate $P^*_{w_t}$. Specifically, the posterior estimation involves continued sampling from the closed-source LM. An intuitive idea is that, given a sampled token $\hat{w}_t$ and a target token w_t, if the sampling results in $\hat{w}_t = w_t$, the probability of generating w_t should be increased; on the other hand, if the sampling results in $\hat{w}_t \neq w_t$, then the probability of generating w_t should be decreased. A discrete random event Y is defined as follows: In a sampling round of the closed-source LM, given input sequence $(w_{t-1}, \ldots, w_1, I)$ and a target token w_t, if the sampled token $\hat{w}_t$ = w_t, then $Y = 1$; otherwise, $Y = 0$.

In practice, we achieve this by introducing an open-source language model $\mathcal{M}$ as a proxy of the closed-source model. The $\mathcal{M}$ is first fine-tuned on the corpus $\mathcal{C}$ for preliminary alignment. We feed the sequence $(w_{t-1}, \ldots, w_1, I)$ into $\mathcal{M}$ to sample a generated token $\hat{w}_t$ at time t. In a sampling round, we update the prior probability dense function $f_{W_t}(P_{w_t})$ based on the event Y. If $Y = 1$ occurs, according to Bayes' theorem:

$$f_{W_t|Y}(P_{w_t}|Y = 1) \propto \Pr(Y = 1|P_{w_t}) f_{W_t}(P_{w_t}) = P_{w_t} f_{W_t}(P_{w_t}) \tag{5}$$

Where $f_{W_t|Y}(P_{w_t}|Y = 1)$ is the posterior probability dense function conditioned on event $Y = 1$. Then, we integrating over $P_{w_t} f_{W_t}(P_{w_t})$ to get a normalization factor η:

$$\eta = \int_0^1 x f_{W_t}(x)dx \tag{6}$$

Then the value of $f_{W_t|Y}(P_{w_t}|Y = 1)$ can be calculated as $f_{W_t|Y}(P_{w_t}|Y = 1) = \frac{1}{\eta} P_{w_t} f_{W_t}(P_{w_t})$. In a sampling round, if event $Y = 0$ occurs instead, according to Bayes' theorem:

$$f_{W_t|Y}(P_{w_t}|Y = 0) \propto \Pr(Y = 0|P_{w_t}) f_{W_t}(P_{w_t}) = (1 - P_{w_t}) f_{W_t}(P_{w_t}) \tag{7}$$

Where $f_{W_t|Y}(P_{w_t}|Y = 0)$ is the posterior probability dense function conditioned on event $Y = 0$. Similarly, we integrating over $(1 - P_{w_t}) f_{W_t}(P_{w_t})$ to get the normalization factor η:

$$\eta = \int_0^1 (1 - x) f_{W_t}(x)dx \tag{8}$$

Then the value of $f_{W_t|Y}(P_{w_t}|Y = 0)$ can be calculated as $f_{W_t|Y}(P_{w_t}|Y = 0) = \frac{1}{\eta}(1 - P_{w_t}) f_{W_t}(P_{w_t})$. The sampling process for $\mathcal{M}$ typically involves multiple iterations, where posterior probability density function $f_{W_t|Y}(P_{w_t}|Y)$ of each round will update the prior probability density function $f_{W_t}(P_{w_t})$ for the next round. And we define $f_{W_t}(P_{w_t})$ in the first round as the probability density function obtained through prior estimation. We denote the final posterior probability dense function as $f_{W_t|\mathcal{M}}(P_{w_t}|\mathcal{M})$. Then a posterior probability for approximating the latent probability $P^*_{w_t}$ can be obtained by calculating the conditional expectation:

$$\mathbb{E}(P_{w_t}|\mathcal{M}) = \int_0^1 x f_{W_t|\mathcal{M}}(x|\mathcal{M})dx \tag{9}$$

3.3 OVERALL OBJECTIVE

The overall objective function at time step t comprises three objectives. Let $\mathbb{1}_{w_t}$ be the one-hot label, the first objective at time step t can be derived by calculating the cross entropy as $\mathcal{L}^{\text{ce}}_t = -\sum_{w_t \in \mathbb{V}} \mathbb{1}_{w_t} \log Q_{w_t}$. The second objective at time step t can be derived based on the prior

estimation as $\mathcal{L}_t^{\text{kl}} = \sum_{w_t \in \mathbb{V}} \mathbb{E}(P_{w_t}) \log \frac{\mathbb{E}(P_{w_t})}{Q_{w_t}}$. We first normalize $\mathbb{E}(P_{w_t}|\mathcal{M}) = \frac{\mathbb{E}(P_{w_t}|\mathcal{M})}{\sum_{w_t' \in \mathbb{V}} \mathbb{E}(P_{w_t'}|\mathcal{M})}$, then the third objective at time step t can be derived based on the posterior estimation as $\mathcal{L}_{t|\mathcal{M}}^{\text{kl}} = \sum_{w_t \in \mathbb{V}} \mathbb{E}(P_{w_t}|\mathcal{M}) \log \frac{\mathbb{E}(P_{w_t}|\mathcal{M})}{Q_{w_t}}$. Given a sequence with length T, the overall objective function can be derived as follows:

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^{T} (\mathcal{L}_t^{\text{ce}} + \alpha \mathcal{L}_t^{\text{kl}} + \beta \mathcal{L}_{t|\mathcal{M}}^{\text{kl}}) \tag{10}$$

Where the α and β are hyperparameters used to adjust the contributions of the $\mathcal{L}_t$ and $\mathcal{L}_{t|\mathcal{M}}$ in the total loss. When $\alpha = 0$ and $\beta > 0$, the student model does not learn from the prior distribution. And the student model does not learn from the posterior distribution when $\alpha > 0$ and $\beta = 0$.

4 Experimental Setup

In this section, we setup a series of experiments to test the distilled models' capabilities on various benchmarks. These benchmarks assess the model across wide range of capabilities including reading comprehension, commonsense knowledge, mathematical skills and logical reasoning.

4.1 Datasets

We utilize the OpenOrca(Mukherjee et al., 2023) dataset as our training corpus. The OpenOrca dataset is a collection of FLAN(Longpre et al., 2023) data augmented by closed-source LLMs like GPT-4 and GPT-3.5. Following the settings in OpenOrca-Preview1-13B[1] of paper Mukherjee et al. (2023), and consider time efficiency, we conduct training on a subset of the original corpus containing 200k instances. We also utilize the Alpaca(Taori et al., 2023) dataset as an additional experimental configuration.

We utilize benchmarks including BBH(Suzgun et al., 2022), AGIEval(Zhong et al., 2023), ARC(Challenge)(Clark et al., 2018), MMLU(Hendrycks et al., 2021), CSQA(Talmor et al., 2019) and GSM8K(Cobbe et al., 2021) for evaluation. Following the settings of Mukherjee et al. (2023), we focus on datasets that involve multiple-choice questions. For all datasets, we conduct evaluation under zero-shot setting without any exemplars and without any CoT(Wei et al., 2022).

4.2 Backbone Models

We employ currently state-of-the-art closed-source LLMs GPT-4 as well as text-davinci-003 as the closed-source teacher models. We utilize LLaMA-7B and LLaMA-13B as student models, which are initialized with pre-trained weights obtained from Hugging Face[2]. We choose LLaMA-33B as the proxy model. We employ top-p sampling for decoding. We train our models on 8 32GB V100 GPUs. To accelerate training, we leverage LoRA (Hu et al., 2021). Additional details can be found in Appendix A.

4.3 Baselines

We consider instruction fine-tuning (IFT) approach as our baseline. For baseline models, to ensure a fair comparison, we only consider models that have access to their original fine-tuning datasets. Therefore we select OpenOrca-Perview1-13B from Mukherjee et al. (2023) and Alpaca (Taori et al., 2023) as our baseline models. In addition, we also train our own version of baseline models.

5 Result and Analysis

In this section, we present the main results, ablation studies and additional experiments. All corpus for proxy model fine-tuning, prior estimation, posterior estimation, and student distillation are iden-

[1] https://huggingface.co/Open-Orca/OpenOrca-Preview1-13B

[2] https://huggingface.co/models

Models	#Params	BBH	AGIEval	ARC	MMLU	CSQA	GSM8K	Average
GPT-4	-	67.4	56.4	-	86.4	-	92.0	-
LLaMA-7B (IFT)	7B	36.08	24.14	47.49	38.81	58.71	12.65	36.31
LLaMA-7B (ours)	7B	38.52	26.92	52.40	41.18	62.52	14.97	**39.43**
OpenOrca-Preview1-13B	13B	41.47	30.12	59.77	48.10	69.77	18.22	44.58
LLaMA-13B (IFT)	13B	42.77	26.74	58.2	45.3	66.27	20.93	43.37
LLaMA-13B (ours)	13B	44.83	29.35	61.84	48.17	68.94	23.36	**46.08**

Table 2: The results of the LLaMA models with different sizes on six benchmarks. We compare our approach to methods directly instruction fine-tuning on the one-hot labels. The performance of OpenOrca-Preview1-13B is assessed through our own evaluation. All student models are trained on the OpenOrca dataset.

Models	#Params	BBH	AGIEval	ARC	MMLU	CSQA	GSM8K	Average
text-davinci-003	-	70.7	41.9	-	64.6	-	-	-
Alpaca-7B	7B	34.19	24.16	39.35	33.66	36.16	13.99	30.25
LLaMA-7B (ours)	7B	34.92	24.32	40.3	34.14	38.32	14.33	**31.06**
Alpaca-13B	13B	38.1	26.9	52.57	41.41	55.27	19.27	38.92
LLaMA-13B (ours)	13B	40.82	28.35	53.84	42.17	56.78	19.83	**40.3**

Table 3: The results of the LLaMA models with different sizes on six benchmarks. We compare our method with Alpaca. All student models are trained on the Alpaca dataset.

tical. Unless otherwise specified, "IFT" represents the baseline model that we have implemented ourselves, and the default training corpus we utilize is OpenOrca.

5.1 Main Results

Table 2 shows the performance comparison of our method against baseline models on the six benchmarks. Detailed experimental results can be found in Appendix C. The training corpus we utilized in this table is the OpenOrca dataset. Our method outperforms OpenOrca-Preview1-13B from Mukherjee et al. (2023) as well as our own implemented IFT models under both the 7B and 13B model parameter scales. Table 3 shows the performance comparison between our method and Alpaca. The training corpus we utilized in this table is the Alpaca dataset. It shows a similar trend, with our method outperforming Alpaca. A case study in Table 4 demonstrates that our model exhibits better comprehension and answer generation capabilities in terms of reasoning ability compared to the baseline IFT model. The experimental results demonstrate that in the context of KD for closed-source LM, distilling knowledge using the estimated soft labels through our method yields superior results compared to directly using one-hot labels.

5.2 Ablation Study

This ablation study examines the impact of components within our method. While retaining the standard cross-entropy loss $\mathcal{L}_t^{\text{ce}}$, we evaluate the effect of using only the prior estimation ($\alpha > 0$,

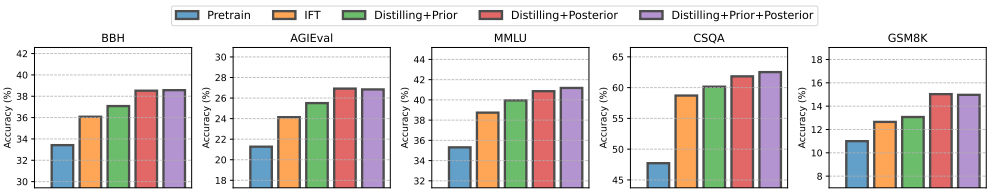

Figure 3: Ablation results of LLaMA-7B across five representative benchmarks. In "Distilling+Prior" we adjust $\alpha = 0.5, \beta = 0$, to investigate the effect of the prior estimation. In "Distilling+Posterior" we adjust $\alpha = 0, \beta = 1$, to investigate the effect of the posterior estimation. In "Distilling+Prior+Posterior" we adjust $\alpha = 0.5, \beta = 1$ to study combined effect of both.

Instruction	The fastest car was 1.25 times faster than the 2nd fastest car. The 2nd fastest car was 5 times faster than the average car. The average car can go 120 Miles per hour. How fast does the faster car go?
Ours	The 2nd fastest car is 5 times faster than the average car, so it can go 5 x 120 = 600 Miles per hour. The faster car is 1.25 times faster than the 2nd fastest car, so it can go 1.25 x 600 = 750 Miles per hour. Therefore, the faster car goes 750 Miles per hour. ✓
IFT	The faster car can go 1.25 x 5 = 60 Miles per hour. Therefore, the faster car goes 60 Miles per hour. ×

Table 4: Examples from GSM8K dataset. Our method enhances the student model's comprehension and leads to more detailed responses compared to the IFT method.

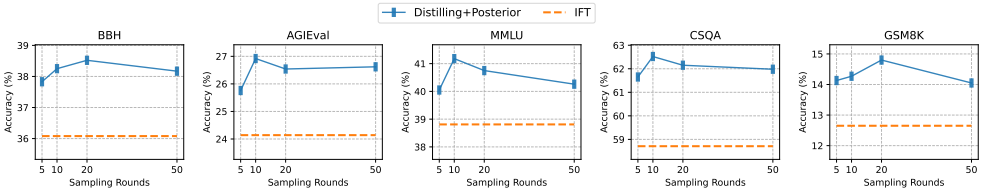

Figure 4: The comparison of knowledge distillation performance using the posterior distribution under different sampling round settings, as well as the comparison with IFT, with the model utilizing LLaMA-7B.

$\beta = 0$), and using only posterior estimation ($\alpha = 0$, $\beta > 0$), and using both ($\alpha > 0$, $\beta > 0$). We select five representative benchmarks. All results are presented in Figure 3.

Effect of the prior estimation Compared to IFT, distilling on the prior distribution (Distilling+Prior) can enhance the model performance. The results indicate that, in addition to guiding the student towards learning from ground-truth token, informing the student model about other valid tokens benefits the distillation. The consistent improvement over IFT suggests that the prior estimation can capture these valid tokens that represent the capabilities of the teacher model.

Effect of the posterior estimation Compared to IFT, distilling on the posterior distribution (Distilling+Posterior) significantly boosts the performance. The improvement over "Distilling+Prior" indicates that, the sampling results from proxy model further refines the prior distribution. The posterior distribution can provide more comprehensive information that is beneficial for distillation.

Combined effect of both As shown in Figure 3, we incorporate the prior distribution and the posterior distribution into the distillation process (Distilling+Prior+Posterior). We observe that the effect is similar to "Distilling+Posterior", with limited improvements seen on only a subset of the benchmarks. We analyze the reason for this phenomenon is the posterior distribution already contains the information from the prior distribution, the improvement gained from incorporating the prior distribution is limited.

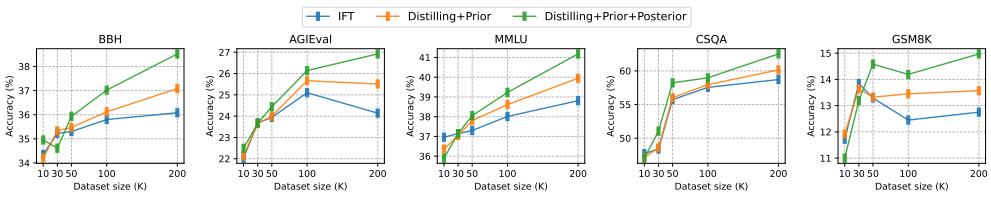

Figure 5: Under different dataset sizes, we investigate the comparison of three methods: IFT, distilling on prior distribution (Distilling+Prior), and distilling on both prior and posterior distributions (Distilling+Prior+Posterior), with the student model utilizing LLaMA-7B.

5.3 IMPACT OF SAMPLING ROUNDS

In this section, we discuss the impact of the number of sampling rounds on posterior estimation. The results are represented in Figure 4. We observe that the best performance is achieved on most benchmarks when the sampling rounds falls within the range of [10,20]. Furthermore, excessive sampling, such as 50 times, leads to a decline in performance. We analyze this phenomenon can be attributed to the distribution discrepancy and prior distribution vanishing.

Distribution discrepancy We observe there exist discrepancies between the ground-truth one-hot labels provided by the closed-source LM and the output distribution of the proxy model. Although the proxy model has been aligned by fine-tuning on corpus $\mathcal{C}$ generated by the closed-source LM, the token with the highest probability given by the proxy model at some positions is different from the ground-truth token (For example, when the ground-truth label at the current position is "\n", the proxy model assigns a high probability (e.g., 0.99) to "<\s>", while the probability of "\n" becomes close to 0), as elaborated in Appendix B.2. In this case, the inconsistency in distributions may negatively impact the performance of the distillation.

Prior Distribution Vanishing In Bayesian estimation, there exists a phenomenon where the prior distribution vanishing as the posterior estimation undergoes excessive iterations. In other words, the impact of the prior distribution weakens with each successive iteration.

We analyze that in Figure 4, excessive sampling (e.g., 50 times) leads to the degeneration of the posterior distribution into the proxy model's output distribution, resulting in negative impact on the performance of knowledge distillation. Therefore, it is important to control the number of samples within a reasonable range. Based on our experimental results, we find that choosing a sampling count between 10 and 20 works fine.

5.4 IMPACT OF CORPUS SIZE

We investigate the effect of training corpus $\mathcal{C}$ size, as shown in Figure 5. We observe that as the size of the training corpus $\mathcal{C}$ increases, the method "Distilling+Prior+Posterior" consistently outperforms the performance of IFT across benchmarks. A similar trend can also be observed in the method "Distilling+Prior". We analyze that our method benefits from a larger corpus. As the corpus size increases, it becomes more advantageous for the prior estimation to estimate a more accurate and information-rich distribution, subsequently influencing the posterior estimation.

6 CONCLUSION

In this work, we address the challenge of knowledge distillation for closed-source language models, where directly access to the teacher's output distribution is not available. We proposed Bayesian estimation-based knowledge distillation to estimate the output distribution of closed-source language models, enabling effective knowledge distillation. Our approach comprises two main components: prior estimation and posterior estimation. The prior estimation involves obtaining a prior distribution by leveraging the corpus generated by the closed-source language model. The posterior estimation updates prior distribution based on continued sampling results from a proxy model. Extensive experiments are conducted based on LLaMA. The results across various benchmarks consistently show that our method outperforms directly fine-tuning on one-hot labels, when it comes to knowledge distillation of closed-source language models.

REFERENCES

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*, 2022.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*, 2023.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. Lion: Adversarial distillation of closed-source large language model. *arXiv preprint arXiv:2305.12870*, 2023.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4163–4174, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.372. URL https://aclanthology.org/2020.findings-emnlp.372.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. The flan collection: Designing data and methods for effective instruction tuning, 2023.

Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant, 2019.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4, 2023.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only, 2023.

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

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

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models, 2023.

Models	Batch Size	Max Length	Lora Rank	#GPUs	Precision	Dimension	#Heads	#Layers
LLaMA-33B	1	512	96	8	float16	6656	52	60
LLaMA-13B	4	512	16	8	float16	5120	40	40
LLaMA-7B	6	512	16	4	float16	4096	32	32

Table 5: Model configurations.

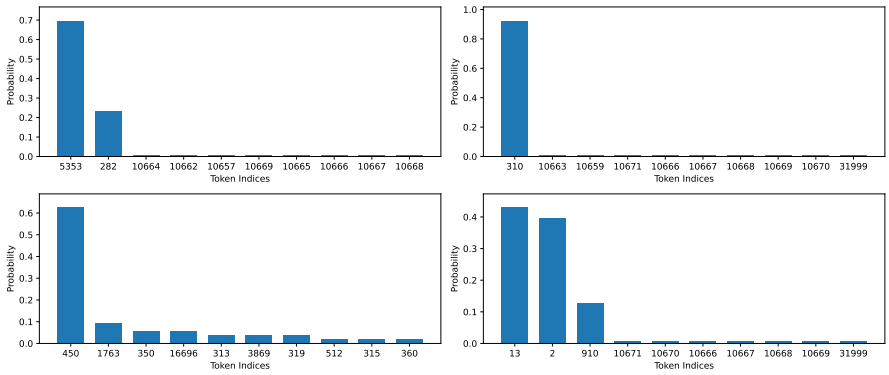

Figure 6: The issue of probability sparsity in the output distribution. A significant portion of probability values concentrates on a few tokens, while the probabilities for other tokens are close to zero.

A Experimental Details

The model configurations are provided in Table 5. We train the student models for three epochs, experimenting with learning rates of 1e-5, 3e-5, and 5e-5 during training. In the knowledge distillation process, we use the following hyperparameters: For the total loss, $\alpha = 0.5$ and $\beta = 1$. For prior estimation, we set $\gamma = 3$ and $n = 5$. For posterior estimation, we conduct 10 rounds of sampling. We evaluate the models on the benchmarks using the final checkpoint.

B Distribution Analysis

B.1 Probability Sparsity

During the distillation process, we observed a phenomenon of probability sparsity in the output distribution of the proxy model. Typically, only a few tokens have high probabilities, while the probabilities of other tokens are close to zero, as shown in Figure 6. In our distillation process, we retained only the probabilities of the top ten tokens with the highest probabilities, setting the probabilities of the remaining tokens to zero. This phenomena indicates that during the sampling process of the proxy model, we don't need to perform a large number of samples to cover all tokens with non-zero probabilities.

B.2 Distribution Discrepancy

We observe that as the number of sampling rounds increased, the model's performance improved on most benchmarks. However, when the number of sampling rounds becomes excessive, such as 50 rounds, the model's performance started to decrease, as shown in Figure 4. We analyze that when the number of sampling rounds becomes excessive, the posterior distribution tends to degenerate into the proxy distribution. When directly using the proxy distribution for knowledge distillation, we observe discrepancies between the proxy distribution and ground-truth labels (For example, when the ground-truth label at the current position is "\n", the proxy distribution assigns a high probability (e.g., 0.99) to "<\s>", while the probability of "\n" becomes close to 0.), which can lead to issues in distillation. More cases are shown in Figure 7.

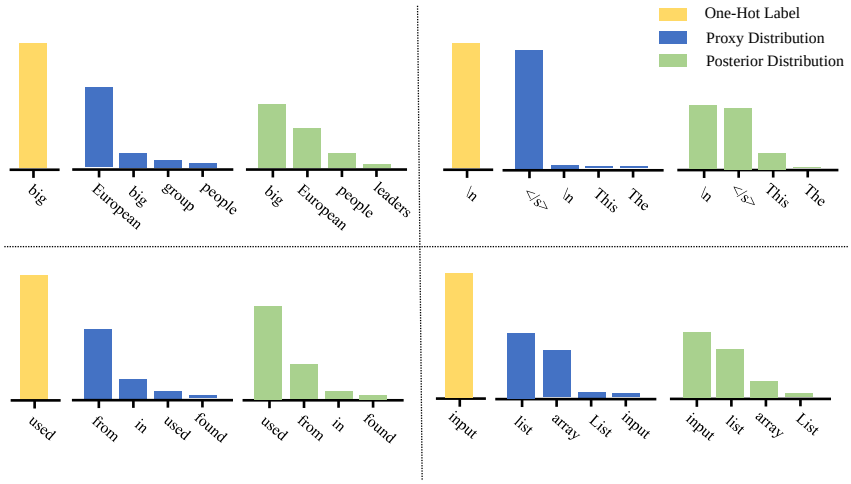

Figure 7: Discrepancies between the the ground-truth distribution and the output distribution of proxy model (proxy distribution) in terms of the top-4 token, while the posterior distribution can stay consistent with the ground-truth distribution.

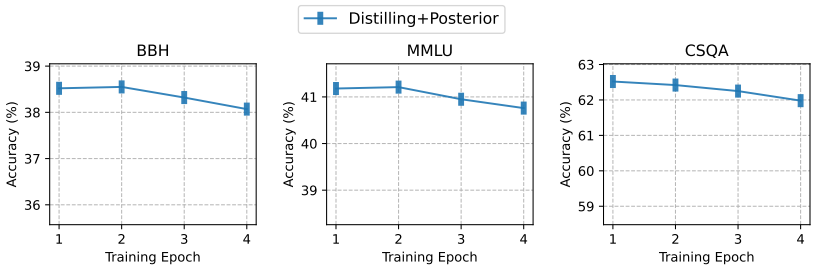

Figure 8: The change in performance of distilling on the posterior distribution (Distilling+Posterior) with the fine-tuning epochs of the proxy model. We utilize LLaMA-7B as the student model, and LLaMA-33B as the proxy model.

C Experimental Results

The detailed experimental results for the LLaMA model on BBH, AGIEval, and MMLU benchmarks are presented in Table 8, Table 7 and Table 9. We also conducted experiments on the FlanT5(Longpre et al., 2023) model using the OpenOrca dataset, and the results are shown in the Table 6. We find that, compared to the IFT method, our approach does lead to some improvement, although the improvement is limited. We speculate that this might be because FlanT5 is a model that has been fine-tuned with instructions, and its original model already had some basic capabilities for these tasks. Therefore, the additional training results in limited improvement.

We also investigated the impact of continuous fine-tuning of the proxy model on the OpenOrca corpus, as shown in the Figure 8. We find that as the number of epochs for fine-tuning the proxy model increases, it leads to a decrease in the performance of posterior estimation. We speculate that this may be due to the proxy model overfitting to the current corpus, resulting in a decrease in the effectiveness. During training, we avoid excessive fine-tuning epochs for the proxy model.

Models	#Params	BBH	AGIEval	ARC	MMLU	CSQA	GSM8K	Average
GPT-4	-	-	56.4	-	86.4	-	92.0	-
FlanT5-large (IFT)	780M	34.63	28.12	46.44	39.41	76.78	4.54	38.32
FlanT5-large (ours)	780M	35.22	28.84	46.61	39.34	76.93	4.71	**38.61**
FlanT5-xl (IFT)	3B	38.47	28.34	59.6	46.91	84.79	6.12	44.04
FlanT5-xl (ours)	3B	39.51	30.1	60.12	46.78	85.38	7.1	**44.83**

Table 6: The results of the FlanT5 models with different parameter sizes on the six benchmarks. We compare our method with IFT.

Models	#Params	AQuA-RAT	LogiQA	LSAT-AR	LSAT-LR	SAT-English (w/o Psg.)	SAT-Math	Average
LLaMA-7B (IFT)	7B	19.71	26.81	18.22	27.44	30.35	22.29	24.14
LLaMA-7B (ours)	7B	22.39	29.68	19.46	33.33	30.46	26.19	26.92
LLaMA-13B (IFT)	13B	18.61	27.59	17.7	34.58	36.27	25.7	26.74
LLaMA-13B (ours)	13B	25.22	29.63	19.65	36.67	33.5	31.43	**29.35**

Table 7: Performance comparison in AGIEval benchmark on the selected multiple-choice English questions. We use OpenOrca dataset as training corpus.

Tasks	LLaMA-13B (IFT)	LLaMA-13B (ours)	LLaMA-7B (IFT)	LLaMA-7B (ours)
Boolean Expressions	58.8	62.4	65.06	66.4
Causal Judgement	61.27	63.01	56.98	61.85
Date Understanding	50.0	54.02	49.3	49.26
Disambiguation QA	56.8	60.0	49.4	54.8
Formal Fallacies	56.4	54.4	54.0	54.0
Geometric Shapes	25.2	23.6	12.42	22.4
Hyperbaton	63.6	66.8	49.2	54.8
Logical Deduction (5 objects)	33.8	36.14	26.51	30.96
Logical Deduction (3 objects)	23.39	30.12	18.7	18.11
Logical Deduction (7 objects)	44.2	51.6	42.17	42.8
Movie Recommendation	77.59	79.32	50.78	53.42
Navigate	51.6	56.8	45.6	55.2
Penguins in a Table	32.61	36.11	30.58	34.91
Reasoning about Colored Objects	39.6	42.8	27.54	30.33
Ruin Names	36.4	33.8	15.2	14.8
Salient Translation Error Detection	31.6	37.2	24.0	28.4
Snarks	48.31	52.25	43.82	45.7
Sports Understanding	60.8	60.4	56.0	55.6
Temporal Sequences	17.28	11.2	13.49	9.68
Tracking Shuffled Objects (5 objects)	19.46	21.1	17.2	17.74
Tracking Shuffled Objects (7 objects)	14.63	17.17	11.98	14.8
Tracking Shuffled Objects (3 objects)	37.5	36.02	33.9	32.52
Average	42.77	**44.83**	36.08	**38.52**

Table 8: Zero-shot performance comparison in Big-Bench Hard benchmark on multiple-choice questions.

Models	#Params	Humanities	Other	Social Sciences	STEM	Average
LLaMA-7B (IFT)	7B	38.49	44.63	40.24	31.87	38.81
LLaMA-7B (ours)	7B	41.4	47.32	42.17	33.82	41.18
LLaMA-13B (IFT)	13B	46.02	53.19	48.24	33.91	45.34
LLaMA-13B (ours)	13B	47.81	56.7	51.36	36.79	**48.17**

Table 9: Performance comparison on the Massive Multitask Language Understanding benchmark.