
DDK: Distilling Domain Knowledge for Efficient Large Language Models

Jiaheng Liu^{*†,1}, Chenchen Zhang^{*2}, Jinyang Guo³, Yuanxing Zhang², Haoran Que¹, Ken Deng¹, Zhiqi Bai¹, Jie Liu⁴, Ge Zhang⁵, Jiakai Wang², Yanan Wu¹, Congnan Liu¹, Jiamang Wang², Lin Qu², Wenbo Su¹, Bo Zheng¹

¹Taobao & Tmall Group of Alibaba, ²Alibaba Group, ³The University of Sydney,

⁴The Chinese University of Hong Kong, ⁵University of Waterloo
{ljh411989}@alibaba-inc.com

Abstract

Despite the advanced intelligence abilities of large language models (LLMs) in various applications, they still face significant computational and storage demands. Knowledge Distillation (KD) has emerged as an effective strategy to improve the performance of a smaller LLM (i.e., the student model) by transferring knowledge from a high-performing LLM (i.e., the teacher model). Prevailing techniques in LLM distillation typically use a black-box model API to generate high-quality pretrained and aligned datasets, or utilize white-box distillation by altering the loss function to better transfer knowledge from the teacher LLM. However, these methods ignore the knowledge differences between the student and teacher LLMs across domains. This results in excessive focus on domains with minimal performance gaps and insufficient attention to domains with large gaps, reducing overall performance. In this paper, we introduce a new LLM distillation framework called DDK, which dynamically adjusts the composition of the distillation dataset in a smooth manner according to the domain performance differences between the teacher and student models, making the distillation process more stable and effective. Extensive evaluations show that DDK significantly improves the performance of student models, outperforming both continuously pretrained baselines and existing knowledge distillation methods by a large margin.

1 Introduction

Recent advancements in Large Language Models (LLMs) such as LLaMA [7, 9, 58, 59] have garnered significant attention due to their strong intelligence. However, these models also impose considerable computational and storage demands, particularly in practical deployments such as instant chat, copilot, and query rewriting. Consequently, the development of lightweight yet efficacious LLMs suitable for real-world applications has become an area of increasing research interest. Several small-scale LLMs, e.g., Phi [39] and MiniCPM [32], have been designed to facilitate rapid inference on devices with limited resources. These models are generally trained from scratch using a large volume of selectively curated high-quality datasets, which could be prohibitive for the broader research community. Meanwhile, there has been a surge in the exploration of model compression techniques [40] to reduce the resource footprint of LLMs. Apart from these techniques, knowledge distillation (KD) emerges as a prominent method for creating effective neural networks, which transfer knowledge from a high-performing teacher model to a compact student model.

* First two authors contributed equally.

† Corresponding Author: Jiaheng Liu.

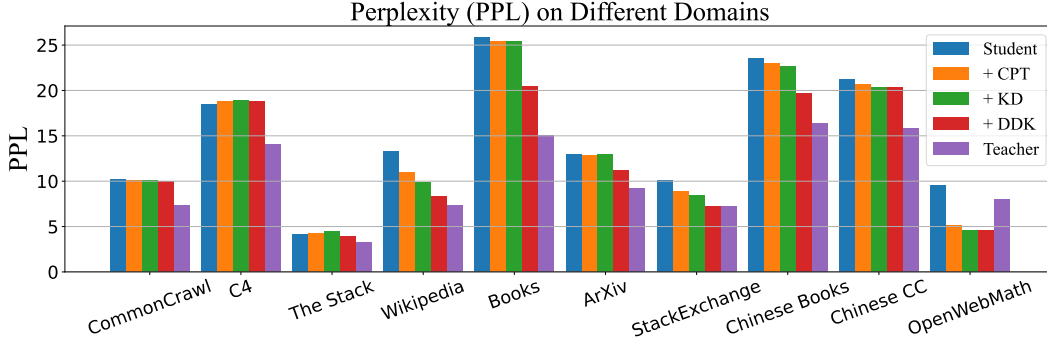


Figure 1: The perplexity scores of different methods across different domains for different methods (See Section 4 for more details.). Note that “Chinese CC” denotes “Chinese CommonCrawl”.

The primary challenges in enhancing the performance of KD approaches on LLMs stem from two main aspects: i) appropriately utilizing the data [3, 67]; ii) stabilize the distillation process [68]. Recently, it has become increasingly acknowledged that the mixture ratios of various domains within the training dataset substantially affect the performance [20, 67, 69]. Regarding the issue of data composition, the influence of domain-specific mixtures for KD remains underexplored. As shown in Fig. 1, the performance between Qwen-1.5 1.8B [7] (student) and Qwen-1.5 14B [7] (teacher) reveals that the performance gap varies significantly across domains. For instance, in the “Books” domain, the student model significantly underperforms the teacher model, while in “The Stack” domain, the difference is minimal, which indicates that the “Books” domain is relatively not optimized well for the student model compared to the teacher model, and more data from the “Books” domain should be included. Therefore, we aim to design a knowledge distillation framework that can dynamically adjust the data composition during distillation to reallocate more computation to domains, where the student and teacher models have larger performance gaps.

In this paper, we introduce a novel methodology, termed **Distill Domain Knowledge** for LLMs (DDK), which effectively optimizes domain-specific mixtures to address the performance discrepancy between teacher and student models across different domains. Specifically, DDK begins by quantifying the performance deviations between the teacher and student LLMs using an offline-collected validation dataset across various domains. Next, it periodically re-calculates the domain discrepancy factor based on the performance gap between the teacher and student models. Finally, DDK employs a domain knowledge-guided sampling strategy to sample data from different domains with varying probabilities based on the calculated domain discrepancy factor. Additionally, inspired by the optimization algorithms [36], we propose a factor smooth updating mechanism to augment the stability and robustness of the DDK approach. For the supervision loss, we minimize the differences in the output logits between the teacher and student models. As demonstrated in Fig. 1, the performance gap across domains is significantly reduced by DDK.

Our main contributions are summarized as follows:

- To the best of our knowledge, we are the first to study the influence of domain-specific data mixtures for distilling LLMs, and efficiently transfer the domain knowledge of the teacher network upon the domain weights.
- DDK proposes a factor smooth updating strategy to strategically enhance the appropriate focus of the distillation process on targeted domains, which effectively stabilizes the domain knowledge guided sampling process for smoother distillation.
- Extensive experiments on multiple benchmark datasets demonstrate the effectiveness and generalization ability of our proposed DDK.

2 Related Works

Large Language Models. The emergence of LLMs [62, 72, 24, 41, 19, 65, 49, 53, 27, 6, 66] marks a significant milestone in the domain of natural language processing, with notable examples including GPT3, Lamda, Palm, and several others [2, 4, 10, 44, 57]. For example, Radford and Narasimhan

[50] introduced the GPT model, leveraging multiple layers of transformer decoder blocks, while Meta later developed LLaMA [58] employing an enhanced transformer architecture, subsequently evolved into LLaMA2 [59]. Recent advancements have also seen the application of instruction tuning [13, 63] and learning through human feedback [8, 45, 74] to better align LLMs with human understanding and foster the creation of versatile AI assistants [21, 43]. Despite their potential, LLMs’ extensive capabilities are often accompanied by vast sizes [35, 64], demanding significant computational resources. In this work, we aim to focus on how to produce small LLMs based on the knowledge distillation approach.

Knowledge Distillation. Knowledge distillation is a pivotal technique in model compression and acceleration [61, 42, 26, 70, 25], primarily employed to transfer knowledge from a robust, well-trained teacher model to a compact student model [29, 1, 37]. Recently, several approaches to knowledge distillation tailored for LLMs have been proposed. These approaches can be broadly classified into two categories: *White-box KD* leverages either the internal parameters or the logits of the teacher LLM during the distillation process [22, 46, 56, 71]. For example, Gu et al. [23] propose that traditional Kullback-Leibler divergence (KLD) objective is inappropriate for open text generation tasks and propose MiniLLM to minimize reverse KLD through policy gradient techniques [54]. Conversely, *black-box KD* relies solely on the outputs from the teacher model [12, 30, 34, 48, 60]. For example, “Distilling Step-by-Step” strategy [31] employs Chain of Thought (CoT) prompting to provide sophisticated guidance during distillation. These two types of KD approaches mainly focus on aligning the generative behaviors of the teacher and student models. DDK delves into the efficacies of domain-specific distillation, aiming to mitigate the discrepancies in performance between the teacher and student model across different domains. Hence, DDK is fundamentally orthogonal to these methods.

3 Methodology

3.1 Overview

Figure 2 illustrates the comprehensive architecture of the DDK framework. DDK employs a large-scale teacher LLM and a comparatively smaller student LLM, with the objective of transferring knowledge from the former to the latter to enhance performance utilizing a specially curated distillation dataset. Initially, the distillation dataset is constructed by randomly sampling from the training corpus. Throughout the distillation process, we continuously assess the domain-specific performance of both the teacher and student LLMs, and use domain knowledge guided sampling to dynamically update the data mixture on the student’s abilities within specific domains. As the domain proficiency of the student LLM evolves during distillation, we introduce a factor smooth updating strategy to ensure the robustness of the domain knowledge-guided sampling approach. Finally, DDK provides of a better student LLM, optimized for enhanced performance across targeted domains.

3.2 Domain Knowledge Guided Sampling

The distilled student LLMs are anticipated to exhibit robust competence across various preset domains. Nevertheless, prevailing knowledge distillation techniques tailored for LLMs tend to homogeneously optimize performance across these domains, leading to potential performance degradation. To address this issue, we design the domain knowledge guided sampling strategy to enhance distillation efficacy by prioritizing domain-specific complexities.

Domain discrepancy factor construction. We consider a dataset \mathcal{D} that has been partitioned into N distinct domains. We denote the pre-trained teacher LLM as \mathcal{M}_T and the student model, which is currently under training, as \mathcal{M}_S . To efficiently identify and prioritize data that may yield the most learning benefit, particularly from domains where the student model underperforms, we introduce a *domain discrepancy factor* denoted as $\mathbf{r} \in \mathbb{R}^N$. Each component $\mathbf{r}[i]$ of this vector quantitatively represents the discrepancy in performance between the teacher and student models within the i -th domain. As we assume a good student should exhibit close approximation to the teacher across all domains, \mathbf{r} is calibrated to reflect differential performance indices as follows:

$$\mathbf{r}[i] = \exp(\ell_S[i]/\ell_T[i]) / \sum_{i' \in \{1, \dots, N\}} \exp(\ell_S[i']/\ell_T[i']) \tag{1}$$

where $\ell_S[i] = \exp(\text{CE}(\mathcal{M}_S(V_i), Y_i))$ and $\ell_T[i] = \exp(\text{CE}(\mathcal{M}_T(V_i), Y_i))$.

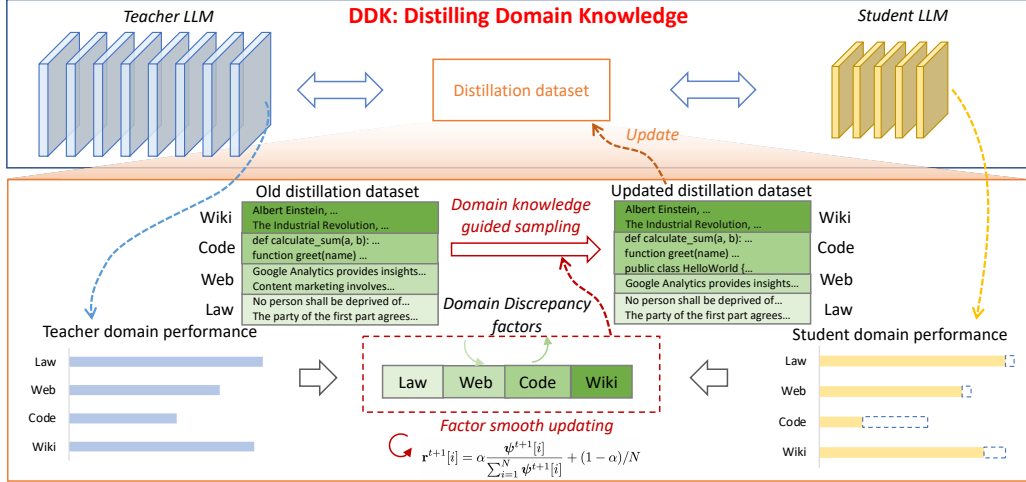


Figure 2: Overview of the distillation process of DDK. First, the training dataset is divided into distinct domains based on predefined criteria. Then, DDK dynamically modulates the distribution of domain-specific data, augmenting the amount allocated to domains where the student model struggles the most. The proportions attributed to each domain are recalculated at distillation intervals by employing a factor smooth updating approach.

Here, V_i and Y_i are the inputs and the ground-truth labels of the validation dataset of the i th domain. $\text{CE}(\cdot)$ represents the cross-entropy loss. $\ell_S \in \mathbb{R}^N$ and $\ell_T \in \mathbb{R}^N$ are the perplexity scores over the validation sets of all domains for student and teacher respectively, indexed by the domain index i . In this case, a higher value of $r[i]$ signifies a pronounced disparity in domain-specific proficiency between the student model and the teacher model. Accordingly, it is imperative to allocate more relevant data to enhance the domain expertise.

Domain knowledge guided sampling. We employ a domain knowledge-informed sampling strategy to refine the composition of the distillation dataset, which utilizes a probabilistic mechanism defined by vector \mathbf{r} to iteratively select samples from the training corpus. The process continues cyclically once a domain data has been exhausted. Finally, DDK strategically increases the data allocation towards underperforming domains, thereby mitigating the performance discrepancies between the teacher and student models across all domains.

3.3 Factor Smooth Updating

With the domain knowledge guided sampling strategy, we can dynamically focus on more challenging domains during the distillation process. Nonetheless, we observe that the domain discrepancy factor exhibits significant fluctuations throughout this procedure. Such rapid alterations may precipitate exceedingly unbalanced data sampling, potentially compromising the stability of the distillation.

Factor smooth updating. To enhance the stability of the distillation process, we periodically adjust the domain discrepancy factor every K iterations throughout the distillation process, thereby partitioning it into discrete intervals. The parameter K is pivotal as it governs the system’s capacity to address immediate discrepancies and influences the stability of the data mixture. We denote the domain discrepancy factor for the i -th domain at the t -th interval of distillation as $r^t[i]$. Similarly, let $\ell_S^t[i]$ and $\ell_T^t[i]$ denote the perplexity scores at the beginning of the t -th distillation interval. In DDK, the domain discrepancy factor at the $(t + 1)$ -th interval is defined as:

$$\mathbf{r}^{t+1}[i] = \alpha \frac{\psi^{t+1}[i]}{\sum_{i=1}^N \psi^{t+1}[i]} + (1 - \alpha)/N, \quad (2)$$

where $\psi^{t+1}[i] = \mathbf{r}^t[i] \exp(\ell_S^{t+1}[i]/\ell_T^{t+1}[i])$.

Note that a constant term is incorporated in $r^t[i]$ to preclude the occurrence of excessively small values, thereby guaranteeing a baseline probability for data sampling across various domains. The parameter α , designated as the smoothing coefficient, is fixed at a value of 0.5 in our experimental

Algorithm 1 Distillation procedure of the DDK framework.

Input: Distillation dataset D ; The steps per distillation interval K ;

- 1: Initialize domain discrepancy factor \mathbf{r}^0 based on Eq. 1;
- 2: Randomly sample $D^0 \subset D$ that supports K steps distillation;
- 3: Initialize student training iteration $c = 0$, distillation interval $t = 0$;
- 4: **for** each iteration in the training process **do**
- 5: // Update student LLM parameters
- 6: Read a batch of samples and use Eq. 3 to update the parameters of student LLM;
- 7: $c = c + 1$
- 8: **if** $c \bmod K == 0$ **then**
- 9: // Update distillation data mixture
- 10: $t = t + 1$;
- 11: Use Eq. 2 to update domain discrepancy factor \mathbf{r}^t ;
- 12: Sample a dataset, $D^t \subset D$, that supports K steps distillation according to \mathbf{r}^t ;
- 13: Shuffle D^t ;
- 14: **if** t reaches a preset maximal number of intervals **then**
- 15: Stop the distillation loop;

Output: The distilled student LLM;

setup. In addition, the inclusion of ψ^t imparts a history mixture information on the modification of the domain discrepancy factor. This mechanism facilitates a gradual modification of $\mathbf{r}^t[i]$, thereby minimizing fluctuations and ensuring a stable, domain knowledge-driven distillation process for fetching informative data.

3.4 Overall Optimization

As we jointly update the student LLM parameters and the domain discrepancy factor in the distillation process, the optimization object can be written as follows:

$$\min_{\theta_S} \sum_{i \in \{1, \dots, N\}} \text{CE}(\mathcal{M}_S(V_i), Y_i) + \gamma \text{KL}(\text{Softmax}(z_S(V_i), T), \text{Softmax}(z_T(V_i), T)), \quad (3)$$

where θ_S is the parameters of the student model. $z_S(\cdot)$ and $z_T(\cdot)$ are the output hidden states from student and teacher LLMs, respectively. We leverage KL-divergence to approximate the student model’s output to the teacher model’s output, over a distillation temperature T . γ is the factor to balance these two terms. Algorithm 1 summarizes the pseudo-code of the DDK process. In practice, the distillation process is typically concluded either when all available data has been fully utilized or when the domain discrepancy factor approaches a threshold indicative of minimal disparity between the teacher and student models.

4 Experiments

In this section, we make comprehensive evaluations to answer two research questions: **RQ1**: To what extent does the DDK process improve the performance of a small-scale LLM? **RQ2**: How does the dynamic domain-specific guidance contribute to the overall improvement?

4.1 Experimental Setup

Model configuration details. We use the Qwen-1.5 [7] and LLaMA2 [59] to demonstrate the effectiveness of DDK. Regarding the Qwen-1.5 series, we use Qwen-1.5 14B and Qwen-1.5 1.8B as the teacher and student models, respectively. For LLaMA2 series, we use LLaMA2 13B and TinyLLaMA 1.1B [73] as the teacher and student models, respectively.

Training details. Due to the unavailability of training data for LLaMA2 and Qwen-1.5 models, we mainly utilize RedPajama [16] for distillation, which consists of training data derived from seven distinct domains: CommonCrawl, C4, The Stack, Wikipedia, Books, ArXiv, and StackExchange.

Table 1: Results of different methods on the Qwen-1.5 models. Note that we use Qwen-1.5 14B and Qwen-1.5 1.8B as teacher and student models, respectively. “W.G.,” “C.QA” and “H.E.” denote Winogrande, CommonsenseQA and Humeneval datasets, respectively.

Methods	CEval	MMLU	RACE	C3	W.G.	GSM8K	C.QA	Arc-E	Arc-C	H.E.	MBPP	Avg.
Teacher (14B)	78.68	64.34	89.95	77.38	68.74	67.63	82.06	87.58	80.59	37.80	44.00	70.80
Student (1.8B)	59.66	44.48	69.57	58.27	57.85	38.4	64.70	70.23	50.31	11.87	18.00	49.39
+ CPT	60.13	45.01	69.00	60.30	56.98	42.50	64.78	72.00	51.03	13.12	20.45	50.48
+ KD [29]	61.29	43.63	70.12	63.92	58.01	49.58	66.26	73.41	54.56	15.63	25.15	52.87
+ TED [40]	62.04	45.21	69.95	63.18	57.38	49.28	65.27	74.74	55.00	13.75	22.69	52.59
+ MiniLLM [23]	61.66	45.07	68.92	63.37	57.14	48.90	64.46	74.52	53.92	16.88	23.55	52.58
+ DDK (Ours)	63.75	46.01	71.56	65.53	59.10	53.54	66.75	75.01	55.03	27.13	26.10	55.41

Table 2: Results of different methods on the LLaMA models. Note that we use LLaMA2 13B and TinyLLaMA 1.1B as teacher and student models, respectively.

Methods	CEval	MMLU	RACE	C3	W.G.	GSM8K	COSE-QA	Arc-E	Arc-C	H.E.	MBPP	Avg.
Teacher (13B)	34.32	49.31	62.85	46.03	63.77	24.10	52.17	73.30	49.40	18.30	28.10	45.60
Student (1.1B)	23.92	24.89	22.92	35.24	55.49	14.19	19.08	24.18	24.12	5.62	16.58	24.20
+ CPT	26.79	26.26	24.24	38.91	56.20	15.03	20.39	28.06	26.03	6.88	17.35	26.01
+ KD	27.12	26.13	23.84	37.43	53.91	15.92	22.52	29.40	26.27	7.50	17.97	26.18
+ TED	27.49	26.43	24.18	37.61	55.72	14.74	22.93	28.61	25.40	8.13	17.45	26.24
+ MiniLLM	26.74	26.45	24.32	37.18	54.46	16.30	22.93	29.46	25.84	8.13	18.28	26.37
+ DDK (Ours)	27.86	28.74	27.76	42.41	57.62	17.44	25.39	36.29	30.15	9.36	19.51	29.32

Moreover, to enhance the model’s proficiency in Chinese and Mathematics, we also incorporate three cleaned open-source datasets (i.e., Chinese Books [19], Chinese CommonCrawl [19], and OpenWebMath [47]). Therefore, there are ten domain datasets for the distillation. In addition, to assess the disparity in performance between teacher and student models across the ten domains, we have constructed a domain-specific validation set for each domain, where each domain includes 500 samples. During the distillation phase, the student models are trained on approximately 15B tokens. For the training framework, we employ the DeepSpeed-Chat code¹ as our codebase, and conduct all experiments using 16 NVIDIA A100 GPUs (80G), where FlashAttention V2 [17] is used to accelerate training. For the training schedule, we first apply the warm-up strategy to increase the learning rate from 0 to $3e^{-5}$ in 1,000 steps. Then, we use the cosine learning rate schedule, where the final learning rate is $3e^{-6}$ and the whole training step is about 30,000 steps. Empirically, we set the distillation interval K as 1,000 and the temperature T as 1.0.

Evaluation details. As we do not conduct instruction tuning on the student models, we mainly report the zero-shot, close-ended results across commonly used datasets including C-Eval [33] (val), MMLU [28] (test), RACE [38] (high, test), C3 [52] (test), WinoGrande [51] (val), GSM8K [15] (test), CommonsenseQA [55] (val), Arc-E [14] (test), Arc-C [14] (test) and HumanEval [11] (test). We also report the 3-shot performance on MBPP [5] (test).

Baseline details. We compare DDK with four baseline methods:

- **CPT** denotes that we continue to pre-train the student model by using the same number of training tokens without considering domains.
- **KD** [29] denotes the standard knowledge distillation by computing the KLD between the teacher and student logits without considering domains.
- **TED** [40] denotes to use task-aware filters to align the hidden representations of the student and the teacher at each transformer layer.
- **MiniLLM** [23] denotes to replace the forward KL divergence with reverse KL divergence, which prevents the student model from overestimating the low-probability regions of the teacher distribution.

¹<https://github.com/microsoft/DeepSpeedExamples/tree/master/applications/DeepSpeed-Chat>

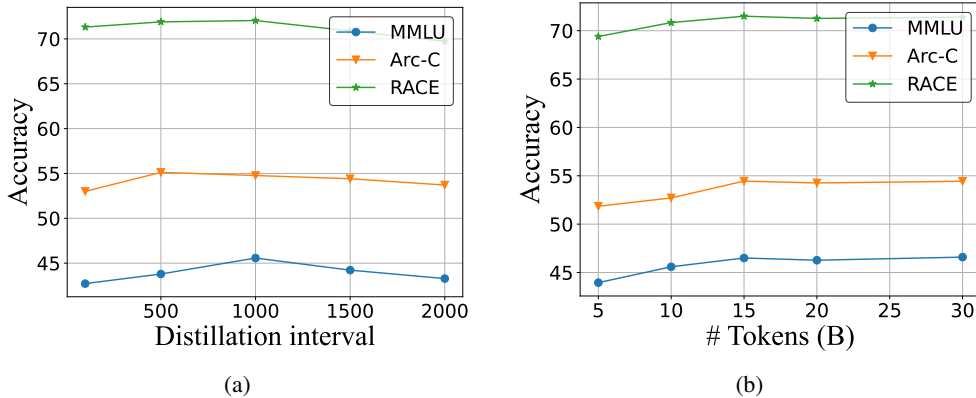


Figure 3: (a). Effect of distillation interval. (b). Effect of the number of training tokens.

4.2 Main Results

As shown in Table 1-2, we report the performance results of different baseline methods. The following observations provide a comprehensive response to RQ1: (1) We see that the absence of knowledge transfer from the teacher model significantly impedes the student model’s capabilities in intricate tasks such as coding (e.g., HumanEval) and Chinese comprehension (e.g., C3). (2) DDK outperforms other baseline methods when using different types of teacher and student models, which demonstrates the effectiveness of DDK for training small student LLMs. (3) The baseline methods KD, TED, and MiniLLM exhibit similar performance. For instance, the average accuracy of these three approaches hovers around 52% when distilling onto the Qwen student model. We hypothesize that in the context of LLM distillation, domain data mixture may emerge as a key performance bottleneck, and the existing baseline techniques fail to adequately address this challenge. (4) The performance gains vary across different domains. Notably, when distilling the Qwen model, we achieve significant improvements on the reasoning tasks (e.g., Code on Humaneval and MBPP, Math on GSM8K), which indicates that the student model can improve a lot on the reasoning tasks under the guidance of the teacher model. This empirical observation suggests that DDK is successful in directing additional attention toward the more challenging problem domains.

4.3 Ablation Study

In this section, we perform ablation studies to assess the robustness of the DDK model and its sensitivity to key hyperparameters. We collected data using Qwen 1.5 and reported its performance on the validation sets of MMLU, RACE, and ARC-C, which differ from those discussed in the previous subsection. Initially, we concentrate on addressing RQ1 through fine-grained analyses.

Effect of data sampling strategies. We propose two variants of data sampling strategies on DDK. For DDK (w/o FS), we just remove the factor smooth updating mechanism and directly take r^t as the probability of each domain. For DDK (ES), we sample data from each domain equally. The results are shown in Fig. 4, and we can suppose that both factor smooth updating and domain knowledge guided sampling contribute to the distillation owing to the existence of domain-specific discrepancy.

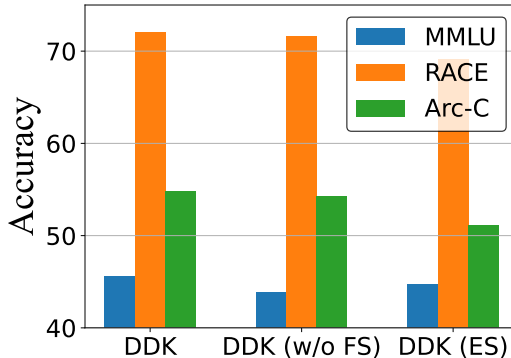


Figure 4: Effect of data sampling strategies.

Effect of distillation interval. Fig. 3 (a) shows the evaluation results on the effect of the distillation interval hyperparameter (i.e., K) in Alg. 1. We observe that increasing K from 100 to 1,000 leads to better performance, indicating that a rapid updating frequency may destabilize the distillation process. However, further increasing K leads to inferior results. We conclude that when the updating

frequency is small, the domain weights update quickly and the student LLM weights can not be sufficiently optimized for the current distillation interval. Meanwhile, when the updating frequency is large, there is insufficient alignment between the LLM weights and the optimal domain weights.

Effect of training tokens. As shown in Fig. 3 (b), we investigate the relationship between the results on three representative datasets and the number of training steps for Qwen-1.5 1.8B model when using Qwen-1.5 14B model as teacher. At the first 10B tokens, the results improve quickly, which indicates that the student models can benefit a lot with the supervision of the teacher model. When further increasing the training iterations, we observe that the performance tends to plateau, which indicates a fast convergence of distillation by DDK.

4.4 Further Analysis

We provide further investigation to show the applicability of DDK across more scenarios.

Generalization ability of using different teacher / student models. To show the generalization ability of DDK on different student models, we use Qwen-1.5 14B as the teacher model and use Qwen-1.5 4B as the student model. As shown in Table 3, DDK surpasses the baseline methods by a large margin. Additionally, comparative analysis with the enhancements observed when employing Qwen-1.5 1.8B as the student model, as presented in Table 1, verifies that a more capable student model tends to yield superior performance improvements. We then apply another teacher model to show the generalization ability of DDK. Specifically, we take Qwen-1.5 7B and Qwen-1.5 1.8B as teacher and student models, respectively. As documented in Table 4, DDK consistently facilitates the most substantial enhancement. These results demonstrate the efficacy and robustness of DDK in leveraging diverse teacher-student model configurations.

Generalization ability on Code LLMs. We implement DDK on LLMs, selecting the Code LLM StarCoder as a case study for empirical evaluation. Within the StarCoder series, we deploy StarCoder 15.5B as the teacher model and StarCoder 3B as the student model. The training corpus is primarily derived from four programming language domains—Python, Java, TypeScript, and C#—sampled from The Stack V2 dataset², with each language representing a distinct domain. We report the performance on the repository-level code completion dataset (i.e., CrossCodeEval [18]). The results in Table 5 affirm that DDK brings notable enhancements in the performance of StarCoder 3B, thereby highlighting DDK’s efficacy in tackling the vertical distillation task.

Table 3: Results of different methods on the Qwen-1.5 models. Note that we use Qwen-1.5 14B and Qwen-1.5 4B as teacher and student models, respectively.

Methods	CEval	MMLU	RACE	C3	W.G.	GSM8K	C.QA	ARC-E	ARC-C	H.E.	MBPP	Avg.
Teacher (14B)	78.68	64.34	89.95	77.38	68.74	67.63	82.06	87.58	80.59	37.80	44.00	70.80
Student (4B)	67.60	53.23	80.17	65.26	64.08	52.24	74.24	79.30	66.20	25.60	29.20	59.74
+ CPT	68.05	52.78	79.56	67.72	63.61	54.00	74.32	80.20	66.67	26.30	31.00	60.38
+ KD	68.35	52.90	80.13	70.31	63.53	56.00	75.51	82.19	67.18	27.50	32.85	61.50
+ MiniLLM	68.20	51.93	79.22	68.78	62.27	55.72	73.87	83.92	67.37	28.13	33.05	61.13
+ DDK (Ours)	68.57	53.17	82.53	70.25	64.85	62.09	75.14	84.10	68.95	30.63	39.12	63.58

Table 4: Results of different methods on the Qwen-1.5 models. Note that we use Qwen-1.5 7B and Qwen-1.5 1.8B as teacher and student models, respectively.

Methods	CEval	MMLU	RACE	C3	W.G.	GSM8K	C.QA	ARC-E	ARC-C	H.E.	MBPP	Avg.
Teacher (7B)	74.10	58.39	85.78	76.03	65.59	54.53	79.28	85.78	72.30	35.63	37.40	65.89
Student (1.8B)	59.66	44.48	69.57	58.27	57.85	38.4	64.70	70.23	50.31	11.87	18.00	49.39
+ CPT	60.13	45.01	69.00	60.30	56.98	42.50	64.78	72.00	51.03	13.12	20.45	50.48
+ KD	62.63	45.07	69.86	61.18	57.08	48.14	65.27	73.74	52.50	13.75	22.69	51.99
+ MiniLLM	62.40	45.20	69.10	61.45	57.46	47.56	65.11	73.86	52.97	14.38	23.31	52.07
+ DDK (Ours)	64.41	46.44	70.98	63.37	57.54	54.06	66.83	74.43	55.09	11.88	24.98	53.64

Analysis on the in-context learning abilities. We evaluate in-context learning capabilities utilizing DDK and the other baselines through several few-shot benchmarks in Table 6. As shown in Table 6, we observe that our DDK consistently manifests considerable enhancements in performance, affirming that DDK transcends mere static knowledge transfer to the student model and augments the in-context learning capacity greatly.

²<https://huggingface.co/datasets/bigcode/the-stack-v2>

Table 5: Results of different methods on the StarCoder models. Note that we use StarCoder 15.5B and StarCoder 3B as teacher and student models, respectively.

Methods	Python		JAVA		TypeScript		C#		Avg.	
	EM	ES	EM	ES	EM	ES	EM	ES	EM	ES
Teacher (15.5B)	35.9	66.1	41.5	72.9	38.7	73.7	56.3	79.3	43.1	73.0
Student (3B)	20.8	41.5	25.3	51.4	25.7	56.2	40.5	60.5	28.1	52.4
+ CPT	24.8	49.3	31.6	61.5	30.5	63.7	47.1	68.4	33.5	60.7
+ KD	26.5	53.2	32.4	61.1	31.6	64.5	48.0	69.8	34.6	61.2
+ DDK (Ours)	31.7	62.2	34.6	69.8	33.2	69.3	50.9	76.2	37.6	69.4

Table 6: Few-shot (**5-shot**) performance results of different methods on the Qwen-1.5 models. Note that we use Qwen-1.5 14B and Qwen-1.5 1.8B as teacher and student models, respectively.

Methods	CEval	MMLU	GSM8K	Arc-E	Arc-C	Avg.
Qwen-14B	79.86	66.30	69.14	89.24	82.25	77.36
Student (1.8B)	61.96	45.59	38.4	72.16	52.11	54.04
+ CPT	60.92	45.60	43.36	73.10	52.28	55.05
+ KD	61.66	44.28	50.26	73.87	54.69	56.95
+ DDK (Ours)	65.38	47.59	55.19	76.64	57.01	60.36

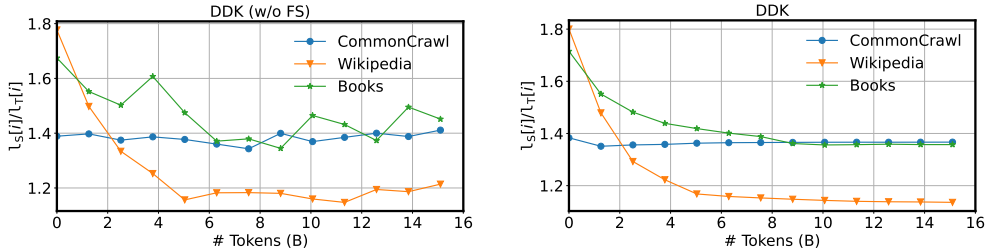


Figure 5: Visualization on the domain discrepancy among three domains.

Visualization. To better show the effectiveness of the factor smooth strategy in DDK, we compare the DDK (w/o FS) with our DDK by showing the domain discrepancy in the training process, where DDK (w/o FS) means that we remove the factor smooth updating strategy. Specifically, in Fig. 5, we compute the $(\ell_S[i]/\ell_T[i])$ as the ratio to represent the domain discrepancy for i -th domain, where a large ratio means a large discrepancy. As shown in Fig. 5, we observe that the ratio updates smoothly in DDK. Besides, in Table 4, the DDK is better than DDK (w/o FS), which means DDK can benefit a lot when using the factor smooth updating strategy.

Moreover, we refer readers to see Appendix B.2 and Appendix C for more details on the training costs and inference examples.

5 Conclusion

In this study, we introduce DDK, a novel framework for knowledge distillation tailored for LLMs. Our initial investigations underscore the criticality of optimizing domain data mixtures in the context of LLM distillation. To address this, we propose a domain knowledge-guided sampling approach that dynamically modulates the sampling probabilities across various domains. Furthermore, we put forward a factor smooth update strategy aimed at enhancing both the stability and the efficacy of the distillation process. Comprehensive evaluations of several benchmark datasets with diverse teacher-student model configurations demonstrate the effectiveness of the DDK framework.

References

- [1] R. Agarwal, N. Vieillard, Y. Zhou, P. Stanczyk, S. R. Garea, M. Geist, and O. Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024.
- [2] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, E. Goffinet, D. Heslow, J. Launay, Q. Malartic, B. Noune, B. Pannier, and G. Penedo. Falcon-40B: an open large language model with state-of-the-art performance. 2023.
- [3] I. Amos, J. Berant, and A. Gupta. Never train from scratch: Fair comparison of long-sequence models requires data-driven priors. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=PdaPky8MUn>.
- [4] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023. URL <https://arxiv.org/abs/2305.10403>.
- [5] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le, and C. Sutton. Program synthesis with large language models, 2021.
- [6] G. Bai, J. Liu, X. Bu, Y. He, J. Liu, Z. Zhou, Z. Lin, W. Su, T. Ge, B. Zheng, and W. Ouyang. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. *arXiv*, 2024.
- [7] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, B. Hui, L. Ji, M. Li, J. Lin, R. Lin, D. Liu, G. Liu, C. Lu, K. Lu, J. Ma, R. Men, X. Ren, X. Ren, C. Tan, S. Tan, J. Tu, P. Wang, S. Wang, W. Wang, S. Wu, B. Xu, J. Xu, A. Yang, H. Yang, J. Yang, S. Yang, Y. Yao, B. Yu, H. Yuan, Z. Yuan, J. Zhang, X. Zhang, Y. Zhang, Z. Zhang, C. Zhou, J. Zhou, X. Zhou, and T. Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.
- [8] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022. URL <https://arxiv.org/abs/2204.05862>.
- [9] Baichuan. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*, 2023. URL <https://arxiv.org/abs/2309.10305>.
- [10] T. Brown, B. Mann, N. Ryder, M. Subbiah, et al. Language models are few-shot learners. In *Proceedings of NeurIPS*, 2020. URL <https://papers.nips.cc/paper/2020/hash/1457c0d6bfbcb4967418bfb8ac142f64a-Abstract.html>.
- [11] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba. Evaluating large language models trained on code. 2021.
- [12] Z. Chen, Q. Gao, A. Bosselut, A. Sabharwal, and K. Richardson. DISCO: Distilling counterfactuals with large language models. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5514–5528, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.302. URL <https://aclanthology.org/2023.acl-long.302>.
- [13] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. URL <https://arxiv.org/abs/2210.11416>.

- [14] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.
- [15] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- [16] T. Computer. Redpajama: an open dataset for training large language models, October 2023. URL <https://github.com/togethercomputer/RedPajama-Data>.
- [17] T. Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. 2023.
- [18] Y. Ding, Z. Wang, W. U. Ahmad, H. Ding, M. Tan, N. Jain, M. K. Ramanathan, R. Nallapati, P. Bhatia, D. Roth, and B. Xiang. Crosscodeeval: A diverse and multilingual benchmark for cross-file code completion. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.
- [19] X. Du, Z. Yu, S. Gao, D. Pan, Y. Cheng, Z. Ma, R. Yuan, X. Qu, J. Liu, T. Zheng, X. Luo, G. Zhou, B. Yuan, W. Chen, J. Fu, and G. Zhang. Chinese tiny llm: Pretraining a chinese-centric large language model, 2024.
- [20] S. Fan, M. Pagliardini, and M. Jaggi. DOGE: Domain reweighting with generalization estimation. In *Second Agent Learning in Open-Endedness Workshop*, 2023. URL <https://openreview.net/forum?id=qiKqsqwYXm>.
- [21] Google. Bard, 2023. URL <https://blog.google/technology/ai/bard-google-ai-search-updates/>.
- [22] J. Gou, B. Yu, S. J. Maybank, and D. Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021.
- [23] Y. Gu, L. Dong, F. Wei, and M. Huang. Knowledge distillation of large language models, 2024.
- [24] H. Guo, J. Yang, J. Liu, L. Yang, L. Chai, J. Bai, J. Peng, X. Hu, C. Chen, D. Zhang, et al. Owl: A large language model for it operations. *arXiv preprint arXiv:2309.09298*, 2023.
- [25] J. Guo, J. Liu, and D. Xu. Jointpruning: Pruning networks along multiple dimensions for efficient point cloud processing. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.
- [26] J. Guo, J. Liu, and D. Xu. 3d-pruning: A model compression framework for efficient 3d action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(12): 8717–8729, 2022.
- [27] J. Guo, J. Wu, Z. Wang, J. Liu, G. Yang, Y. Ding, R. Gong, H. Qin, and X. Liu. Compressing large language models by joint sparsification and quantization. In *Proceedings of the 41st International Conference on Machine Learning*, pages 16945–16957, 2024.
- [28] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.
- [29] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [30] N. Ho, L. Schmid, and S.-Y. Yun. Large language models are reasoning teachers. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.830. URL <https://aclanthology.org/2023.acl-long.830>.

- [31] C.-Y. Hsieh, C.-L. Li, C.-k. Yeh, H. Nakhost, Y. Fujii, A. Ratner, R. Krishna, C.-Y. Lee, and T. Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.507. URL <https://aclanthology.org/2023.findings-acl.507>.
- [32] S. Hu, Y. Tu, X. Han, C. He, G. Cui, X. Long, Z. Zheng, Y. Fang, Y. Huang, W. Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.
- [33] Y. Huang, Y. Bai, Z. Zhu, J. Zhang, J. Zhang, T. Su, J. Liu, C. Lv, Y. Zhang, J. Lei, Y. Fu, M. Sun, and J. He. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems*, 2023.
- [34] Y. Jiang, C. Chan, M. Chen, and W. Wang. Lion: Adversarial distillation of proprietary large language models, 2023.
- [35] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. URL <https://arxiv.org/abs/2001.08361>.
- [36] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [37] J. Ko, S. Kim, T. Chen, and S. Yun. Distillm: Towards streamlined distillation for large language models. *ArXiv*, abs/2402.03898, 2024.
- [38] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.
- [39] Y. Li, S. Bubeck, R. Eldan, A. Del Giorno, S. Gunasekar, and Y. T. Lee. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023.
- [40] C. Liang, S. Zuo, Q. Zhang, P. He, W. Chen, and T. Zhao. Less is more: Task-aware layer-wise distillation for language model compression. *ICML*, 2023.
- [41] J. Liu, Z. Bai, Y. Zhang, C. Zhang, Y. Zhang, G. Zhang, J. Wang, H. Que, Y. Chen, W. Su, et al. E2-llm: Efficient and extreme length extension of large language models. *arXiv preprint arXiv:2401.06951*, 2024.
- [42] J. Liu, J. Li, K. Wang, H. Guo, J. Yang, J. Peng, K. Xu, X. Liu, and J. Guo. Lta-pcs: Learnable task-agnostic point cloud sampling. In *CVPR*, 2024.
- [43] OpenAI. OpenAI: Introducing ChatGPT, 2022. URL <https://openai.com/blog/chatgpt>.
- [44] OpenAI. GPT-4 technical report, 2023. URL <https://cdn.openai.com/papers/gpt-4.pdf>.
- [45] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. In *Proceedings of NeurIPS*, 2022. URL <https://arxiv.org/abs/2203.02155>.
- [46] S. Padmanabhan, Y. Onoe, M. J. Zhang, G. Durrett, and E. Choi. Propagating knowledge updates to lms through distillation. *arXiv preprint arXiv:2306.09306*, 2023.
- [47] K. Paster, M. D. Santos, Z. Azerbayev, and J. Ba. Openwebmath: An open dataset of high-quality mathematical web text, 2023.
- [48] B. Peng, C. Li, P. He, M. Galley, and J. Gao. Instruction tuning with gpt-4. *ArXiv*, abs/2304.03277, 2023.
- [49] H. Que, J. Liu, G. Zhang, C. Zhang, X. Qu, Y. Ma, F. Duan, Z. Bai, J. Wang, Y. Zhang, X. Tan, J. Fu, W. Su, J. Wang, L. Qu, and B. Zheng. D-cpt law: Domain-specific continual pre-training scaling law for large language models. *ArXiv*, abs/2406.01375, 2024.

- [50] A. Radford and K. Narasimhan. Improving language understanding by generative pre-training. 2018. URL <https://api.semanticscholar.org/CorpusID:49313245>.
- [51] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.
- [52] K. Sun, D. Yu, D. Yu, and C. Cardie. Investigating prior knowledge for challenging chinese machine reading comprehension. *Transactions of the Association for Computational Linguistics*, 2020. URL <https://arxiv.org/abs/1904.09679v3>.
- [53] T. Sun, L. Chai, J. Yang, Y. Yin, H. Guo, J. Liu, B. Wang, L. Yang, and Z. Li. Uni-Coder: Scaling code large language model via universal code. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1812–1824, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.100. URL <https://aclanthology.org/2024.acl-long.100>.
- [54] R. S. Sutton, D. A. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, 1999.
- [55] A. Talmor, J. Herzig, N. Lourie, and J. Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.
- [56] I. Timiryasov and J.-L. Tastet. Baby llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. *arXiv preprint arXiv:2308.02019*, 2023.
- [57] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. URL <https://arxiv.org/pdf/2302.13971.pdf>.
- [58] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023.
- [59] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [60] P. Wang, Z. Wang, Z. Li, Y. Gao, B. Yin, and X. Ren. Scott: Self-consistent chain-of-thought distillation. In *Annual Meeting of the Association for Computational Linguistics*, 2023.
- [61] Z. Wang, J. Guo, R. Gong, Y. Yong, A. Liu, Y. Huang, J. Liu, and X. Liu. Ptsbench: A comprehensive post-training sparsity benchmark towards algorithms and models. In *ACM Multimedia 2024*.
- [62] Z. M. Wang, Z. Peng, H. Que, J. Liu, W. Zhou, Y. Wu, H. Guo, R. Gan, Z. Ni, M. Zhang, Z. Zhang, W. Ouyang, K. Xu, W. Chen, J. Fu, and J. Peng. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. *arXiv preprint arXiv: 2310.00746*, 2023.
- [63] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned language models are zero-shot learners. In *Proceedings of ICLR, 2022*. URL <https://openreview.net/forum?id=gEZrGCozdqR>.
- [64] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022. URL <https://openreview.net/pdf?id=yzkSU5zdWd>.
- [65] Y. Wu, J. Liu, X. Bu, J. Liu, Z. Zhou, Y. Zhang, C. Zhang, Z. Bai, H. Chen, T. Ge, et al. Conceptmath: A bilingual concept-wise benchmark for measuring mathematical reasoning of large language models. *arXiv preprint arXiv:2402.14660*, 2024.

- [66] M. Xia, T. Gao, Z. Zeng, and D. Chen. Sheared llama: Accelerating language model pre-training via structured pruning. *arXiv preprint arXiv:2310.06694*, 2023.
- [67] S. M. Xie, H. Pham, X. Dong, N. Du, H. Liu, Y. Lu, P. S. Liang, Q. V. Le, T. Ma, and A. W. Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36, 2024.
- [68] Z. Yang, A. Zeng, Z. Li, T. Zhang, C. Yuan, and Y. Li. From knowledge distillation to self-knowledge distillation: A unified approach with normalized loss and customized soft labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17185–17194, 2023.
- [69] J. Ye, P. Liu, T. Sun, Y. Zhou, J. Zhan, and X. Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *arXiv preprint arXiv:2403.16952*, 2024.
- [70] Z. Yuan, J. Guo, D. An, J. Wu, H. Zhu, J. Li, X. Chen, K. Xu, and J. Liu. VRDistill: Vote refinement distillation for efficient indoor 3d object detection. In *ACM Multimedia 2024*, 2024.
- [71] C. Zhang, D. Song, Z. Ye, and Y. Gao. Towards the law of capacity gap in distilling language models. *arXiv preprint arXiv:2311.07052*, 2023.
- [72] G. Zhang, S. Qu, J. Liu, C. Zhang, C. Lin, C. L. Yu, D. Pan, E. Cheng, J. Liu, Q. Lin, et al. Map-neo: Highly capable and transparent bilingual large language model series. *arXiv preprint arXiv:2405.19327*, 2024.
- [73] P. Zhang, G. Zeng, T. Wang, and W. Lu. Tinyllama: An open-source small language model, 2024.
- [74] Z. Zhou, J. Liu, C. Yang, J. Shao, Y. Liu, X. Yue, W. Ouyang, and Y. Qiao. Beyond one-preference-for-all: Multi-objective direct preference optimization. *arXiv preprint arXiv:2310.03708*, 2023.

A Broader Impacts and Limitations

Broader Impacts. Based on our DDK, we can boost the performance of small LLMs a lot under the guidance of the teacher LLMs. In real-world scenes, we hope our enhanced small LLMs can be deployed on low-resource devices (e.g., PCs or Mobiles) and make future life better with better small LLMs.

Limitations. First, we have to tune several hyperparameters to achieve better performance in our DDK. Second, due to the GPU resource limitations, we only distill the 1.1B, 1.8B and 4B sizes under the guidance of teachers with 7B and 10B+ sizes. In the future, we will investigate the performance of knowledge distillation using larger student and larger teacher models.

B More Details

B.1 Discussion on the differences between DDK and Sheared LLaMA

First, for the solved tasks, the Sheared LLaMA [66] is proposed for structure pruning, while our DDK is proposed for knowledge distillation on the LLMs. Therefore, our DDK is orthogonal to the Sheared LLaMA and we can further improve the performance of small models pruned by Sheared LLaMA. Second, for the technical details, the sheared LLaMA needs to first fit a scaling function using three open-sourced LLaMA models (i.e., LLaMA2-7B/13B/70B) when pruning the LLaMA model series, where the fitting data points are very limited. In Sheared LLaMA, the authors also claim the estimated reference losses for different domains are also biased as the limited data points for estimating the scaling law parameters. Besides, in many real-world pruning scenarios, we cannot usually obtain a series of models under the same training setting. Thus, we cannot predict the losses across different domains of the small-size model when a series of models are not provided. In contrast, in DDK, we directly use the domain loss predicted by the teacher model as the reference loss without fitting the scaling function, where the teacher-predicted losses are used as the accurate guidance to improve the student model. Besides, the Sheared LLaMA is proposed to enhance the training efficiency of the continued pre-training setting of the pruned model, the guidance of the teacher model is not included. In addition, our DDK also introduces the factor smooth updating strategy, which is not used in Sheared LLaMA and can make the change of the domain mixture more stable. Moreover, in the following table, we also provide the results of replacing our proposed factor smooth updating strategy with the strategy of Sheared LLaMA, and observe that DDK is better a lot, which further shows the effectiveness of our factor smooth updating strategy.

B.2 Details on the training costs

As shown in Table 7, we compare the TFLOPs of three representative baseline methods, and observe that the training costs of our DDK are acceptable when compared with the baseline KD method.

Table 7: Training TFLOPs on all data of different methods for Qwen-1.5. For KD and DDK, we use the Qwen-1.5 14B to distill the Qwen-1.5 1.8B.

Models	CPT	KD	DDK
TFLOPs	1.456e8	5.364e8	5.401e8

C More Sentence Examples

In the following, we provide more examples generated by the original and distilled models. We find that the sentences generated by the distilled model are superior to those generated by the original model in terms of fluency, relevance, and informativeness regarding the given topic.

GSM8K

Prompt:

Question: Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for \$2 per fresh duck egg. How much in dollars does she make every day at the farmers' market?

Answer: 18

Question: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?

Answer: 3

Question: Josh decides to try flipping a house. He buys a house for \$80,000 and then puts in \$50,000 in repairs. This increased the value of the house by 150%. How much profit did he make?

Answer: 70000

Question: James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total meters does he run a week?

Answer:

Assistant (Qwen-1.5 1.8B):

1800

Assistant (Qwen-1.5 1.8B DDK):

540

Prompt:

Gunter is trying to count the jelly beans in a jar. He asks his friends how many they think are in the jar. One says 80. Another says 20 more than half the first one. A third says 25% more than the first one. What is their average guess?

Answer:

Assistant (Qwen-1.5 1.8B):

First, we need to find half of the first number, which is $80 / 2 = 40$.

Then, we need to add 20 more to that number, which is $40 + 20 = 60$.

Next, we need to find 25% of the first number, which is $80 * 0.25 = 20$.

Finally, we add that number to the first number, which is $60 + 20 = 80$.

Now, we can find the average by adding up all the guesses and dividing by the number of guesses, which is $80 + 60 + 20 = 160$.

Finally, we divide that number by the number of guesses, which is 3, to find the average, which is $160 / 3 = 53.33$.

Therefore, the answer is 53.

Assistant (Qwen-1.5 1.8B DDK):

The first friend's guess is 80 jelly beans.

The second friend's guess is 20 more than half of 80, which is $(80/2) + 20 = 40 + 20 = 60$ jelly beans.

The third friend's guess is 25% more than 80, which is $80 + (0.25 * 80) = 80 + 20 = 100$ jelly beans.

To find the average guess, we add up all the guesses and divide by the number of friends, which is 3.

So the average guess is $(80 + 60 + 100) / 3 = 240 / 3 = 80$ jelly beans.

80

The answer is: 80

Prompt:

问题: 冬瓜、黄瓜、西瓜、南瓜都能吃, 什么瓜不能吃?

答案:

Assistant (Qwen-1.5 1.8B):

冬瓜、黄瓜、西瓜、南瓜都能吃, 南瓜不能吃。

Assistant (Qwen-1.5 1.8B DDK):

傻瓜

解析: 冬瓜、黄瓜、西瓜、南瓜都能吃, 傻瓜不能吃。

Prompt:

问题: 什么东西天气越热, 它爬的越高?

答案: 温度计

问题: 什么动物, 你打死了它却流了你的血?

答案: 蚊子

问题: 什么布剪不断?

答案:

Assistant (Qwen-1.5 1.8B):

蚊帐

Assistant (Qwen-1.5 1.8B DDK):

瀑布

Prompt:

如何写一篇关于环保的激励人心的演讲稿? :

Assistant (Qwen-1.5 1.8B):

环保是当今社会面临的一个重要问题, 它不仅关系到我们的生存环境, 也关系到我们的未来。因此, 我们需要采取行动来保护我们的地球。以下是一篇关于环保的激励人心的演讲稿:

尊敬的各位领导、各位来宾、亲爱的朋友们:

大家好! 今天, 我非常荣幸能够在这里发表关于环保的演讲。环保是我们每个人都应该关注的问题, 它不仅关系到我们的生存环境, 也关系到我们的未来。

首先, 让我们来看看环保的重要性。我们的地球是我们的家园, 它是我们生存的基础。然而, 由于人类的过度开发和污染, 我们的地球正面临着严重的环境问题。例如, 全球变暖、气候变化、水资源短缺、土地退化等问题, 这些问题不仅影响着我们的健康和生活质量, 也威胁着我们的生存环境。

因此, 我们必须采取行动来保护我们的地球。我们可以从以下几个方面入手:

首先, 我们应该减少对环境的污染。我们应该尽可能地减少使用一次性塑料制品, 如塑料袋、塑料瓶等。我们应该尽量使用可重复使用的物品, 如购物袋、水瓶等。

此外, 我们还可以选择使用环保产品, 如环保清洁剂、环保洗衣液等。

其次, 我们应该节约能源。我们应该尽可能地减少能源的消耗, 如

Assistant (Qwen-1.5 1.8B DDK):

好的, 以下是一篇关于环保的激励人心的演讲稿: 尊敬的各位来宾, 大家好!

今天, 我想和大家分享一些关于环保的话题。我们都知道, 环保是我们每个人都应该关注的问题。我们的地球是我们共同的家园, 我们应该尽我们所能来保护它。

首先, 我们需要认识到环保的重要性。我们的地球正面临着许多环境问题, 如气候变化、空气污染、水污染等等。这些问题不仅影响着我们的健康和生活质量, 也影响着我们的未来。因此, 我们必须采取行动来保护我们的地球。

其次, 我们需要采取行动来保护我们的环境。我们可以从小事做起, 如节约用水、节约用电、减少使用塑料袋等等。我们还可以参加环保活动, 如植树造林、垃圾分类等等。这些行动虽然看似微不足道, 但它们可以积少成多, 对环境产生积极的影响。

最后, 我们需要鼓励更多的人加入到环保行动中来。我们可以向身边的人宣传环保知识, 鼓励他们采取环保行动。我们还可以加入环保组织, 参与环保活动, 为环保事业贡献自己的力量。

尊敬的各位来宾, 让我们一起行动起来, 保护我们的地球, 让我们的家园更加美好! 谢谢大家!

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: We have highlighted the main contribution of this paper in the abstract and instruction sections - DKK focuses on the domain discrepancies to adaptively improve the efficacy of knowledge distillation tailored to LLMs.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: As mentioned in Appendix A.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Equations 2 and 3 concludes the main theory assumptions and proofs of this paper. We also provide sufficient explanations attached to them.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The full details of the experiments are provided in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All data used in this paper is open-sourced. The codes for the baseline methods are also collected from Github.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: This information is provided in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We have made sufficient ablation studies to show the confidence of the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Mentioned in Section 4 and Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [Yes]

Justification: No ethic issues are related to this topics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss the broader impact in Appendix A.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: No such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have cited all relevant papers.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We have not yet release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: All evaluations are conducted by rules.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Unrelated.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.