GKT: A Novel Guidance-Based Knowledge Transfer Framework For Efficient Cloud-edge Collaboration LLM Deployment

Anonymous ACL submission

Abstract

The burgeoning size of Large Language Models (LLMs) has led to enhanced capabilities in generating responses, albeit at the expense of increased inference times and elevated resource demands. Existing methods of acceleration, predominantly hinged on knowledge distillation, generally necessitate fine-tuning of considerably large models, such as Llama-7B, posing a challenge for average users. Furthermore, present techniques for expediting inference and reducing costs operate independently. To address these issues, we introduce a novel and intuitive **G**uidance-based **K**nowledge **T**ransfer (GKT) framework. This approach leverages a larger LLM as a "teacher" to create guidance prompts, paired with a smaller "student" model to finalize responses. Remarkably, GKT requires no fine-tuning and doesn't necessitate the teacher and student models to have the same vocabulary, allowing for extensive batch generation to accelerate the process while ensuring user customization. GKT can be seamlessly integrated into cloud-edge collaboration architectures, and is versatile enough for plug-and-play application across various models. It excels in both efficiency and affordability, epitomizing a "cheap and cheerful" solution. GKT achieves a maximum accuracy improvement of 14.18 %, along with a 10.72× speed-up on GSM8K and an accuracy improvement of 14.00 % along with a 7.73× speed-up in CSQA. When utilizing ChatGPT as teacher model and Llama2-70B as the student model, we can achieve 95.00% of ChatGPT's performance at 52% of the cost. The results highlight substantial enhancements in accuracy and processing speed on the GSM8K and CSQA datasets, surpassing the performance of using either the student or teacher models in isolation.

1 Introduction

The swift advancement of large language models (LLMs) has dramatically pushed the frontiers of AI technology. LLMs, with their vast number of parameters, are exceptionally adept at comprehending human intentions, offering high-quality reasoning, and responses. However, the immense size of these models is a double-edged sword. While it improves model performance, it also leads to slower inference times and higher computational costs. As the demand for LLM usage increases, relying solely on LLMs for auto-regressive inference actually demands an overwhelming amount of computational resources and time, posing a substantial deployment challenges for cloud services and resource-constrained devices.

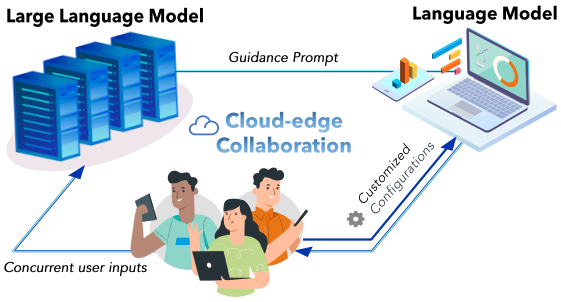


Figure 1: Cloud-edge collaboration: The GKT framework facilitates cloud-edge collaboration by deploying the larger teacher model on remote cloud servers and the smaller student model on lightweight mobile devices. GKT allows for brief guidance prompts to be easily transmitted to mobile devices, significantly reducing data transmission costs. In cloud-edge collaboration, users can also perform simple personalized generation settings on their mobile devices.

Consequently, many recent studies (Leviathan et al., 2023a; Ning et al., 2023; Jiang et al., 2023) have taken steps to improve the inference efficiency of LLMs. One prevalent and widely adopted approach is knowledge distillation (Hinton et al., 2015; Yim et al., 2017; Tunstall et al., 2023; Jiang et al., 2023; Chiang et al., 2023; Li et al., 2023). The majority of knowledge distillation frameworks utilize large language models as "teacher" mod-

els to generate training samples. These samples are then used to train more compact "student" language models, effectively teaching them to mimic the performance and capabilities of teacher model. However, this process still demands a carefully crafted data generation mechanism and the subsequent training of the student model. Despite being smaller in size, to achieve satisfactory results in general tasks, many student models still maintain a considerable parameter size, often around 7B (Tunstall et al., 2023) and 13B (Jiang et al., 2023; Chiang et al., 2023). This is in line with the observations by Wei et al. (2022a), who noted that the emergent abilities of LLMs for language understanding typically become evident when the model size exceeds 10 billion parameters.

Feature	KD	SD	GKT
Accelerates Inference	✓	✓	✓
Preserves Model Architecture	✓	✓	✓
Eliminates Additional Fine-tuning	✗	✓	✓
Allows Custom Settings	✗	✗	✓
Enables Cloud-Edge Collaboration	✗	✗	✓
Allows Different vocabulary	✗	✗	✓

Table 1: Comparative analysis of Knowledge Distillation (KD), Speculative Decoding (SD), and Guidance-based Knowledge Transfer (GKT).

To solve this limitation, another research line focuses on speculative decoding (Leviathan et al., 2023a; He et al., 2023; Leviathan et al., 2023b), which involves using a more efficient, smaller model to generate token predictions, which the larger target model then evaluates. If a token prediction is accepted, it's used; if not, it's discarded, and the target model generates a new token. The method is shown to significantly speed up inference without needing changes to the model's architecture or training procedures. However, users often have diverse generation requirements, such as the desire to adjust generation parameters like temperature and top_p. While speculative decoding can ensure consistency with the output of the final large model, employing batch generation to save time during high concurrent access can impede the ability to meet user-specific generation settings.

Considering the limitations of these methods, we were inspired by the common human experience of "Getting started is the hardest part." and sociological studies (Goldberg et al., 2014) showing that effective prompts provided by teachers in classrooms can significantly improve student performance in exams. Thus, we propose a novel knowledge transfer framework: **G**uidance-based **K**nowledge **T**ransfer (GKT). Our framework involves two steps: firstly, using an LLM as a teacher model to generate guidance prompts from concurrent user inputs through batch generation. Secondly, a smaller LM acts as the student model, which simply completes the answers based on the guidance prompts, allowing for user-customized generation settings.

GKT reduces the burden of LLM inference, thereby speeding up response generation. Unlike knowledge distillation, our framework does not require generating dataset from teacher model and fine-tuning the student model. It also doesn't necessitate the teacher and student models to have the same vocabulary, allowing for extensive batch generation to accelerate the process while ensuring user customization. GKT can also be seamlessly integrated into cloud-edge collaboration architectures as shown in Figure 1. GKT deploys the larger teacher model on remote cloud servers and the smaller student model on lightweight mobile devices, such as smartphones. This setup allows for brief guidance prompts to be easily transmitted to mobile devices, significantly reducing data transmission costs. In cloud-edge collaboration, users can also perform simple personalized generation settings on their mobile devices. The table 1 succinctly delineates the pros and cons between the Guidance-based Knowledge Transfer (GKT) framework, Knowledge Distillation (KD) and Speculative Decoding (SD), providing a clear comparative perspective.

Finally, our study follows the philosophy that "No such thing as bad student. Only bad teacher." We conducted various experiments to explore the optimal guidance strategies for the teacher model, seeking answers to questions such as how much and what type of help a teacher should provide to maximize the student model's accuracy while minimizing the teacher model's inference time. The results demonstrate that GKT not only achieves a significant accuracy improvement of 14.18% on the GSM8K dataset, but also enhances speed by 10.72 $\times$. Furthermore, on the CSQA dataset, it records a noteworthy accuracy increase of 14.00% along with a 7.73 $\times$ acceleration in inference speed.

2 Method

The schematic representation of our Guidance-Based Knowledge Transfer (GKT) framework is

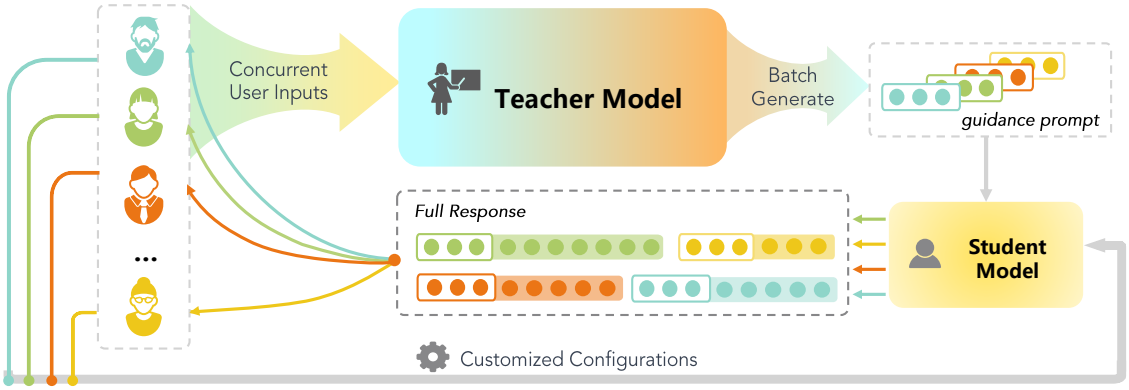


Figure 2: System overview. Our framework consists of two steps: guidance generation and response completion. In guidance generation, teacher model generates guidance prompts using batch generation to process concurrent user inputs. In response completion, student model receives guidance prompt and complete the response. Student model generates output with a batch size of 1 which allows customize generation settings by the user

depicted in Figure 2. GKT encompasses a two-step process: guidance generation and response completion. During the guidance generation phase, a large language model serves as the "teacher" model. This model processes concurrent user inputs and employs batch generation to craft guidance prompts. Subsequently, in the response completion stage, a smaller language model functions as the "student" model. This model offers flexibility in generation settings, allowing for user customization. The guidance prompt created in the first stage is then fed into this smaller model, facilitating the completion of the response with enhanced efficiency. We will now elaborate on each stage:

Guidance Generation Given concurrent user inputs $Q = \{q_1, q_2, ..., q_n\}$, where q_i represents the input question from user i, the teacher model M_t batch generates the guidance prompts:

$$G = \{g_1, g_2, ..., g_n\} = \mathbf{F}(M_t(Q))$$

Here, G represents the batch-generated guidance, g_i is the guidance for user i, and $\mathbf{F}(\cdot)$ denotes the projection operation that generates the guidance prompt from the generated text. In this paper, we explore different projection operations including: (1) Cut-off guidance generation (2) Concise guidance generation (3) Hint guidance generation. We will elaborate on these methods in Section 4.2

Response generation In response generation, we use a smaller language model M_s as student model. For every user i, M_s generates the final response r_i by:

$$r_i = M_s^i(g_i)$$

Where M_s^i stands for M_s under the user i's custom generation setting.

3 Experimental Settings

Datasets In this paper we use two challenge but widely used dataset: GSM8K (Cobbe et al., 2021) for arithmetic reasoning and CSQA (Talmor et al., 2019) for commonsense reasoning. GSM8K focus on arithmetic reasoning which is a collection of grade school math word problems, each requiring 2 to 8 steps to solve. The solutions mainly involve a series of basic arithmetic calculations to arrive at the final answer. CSQA focuses on commonsense question answering which includes multiple-choice questions that require commonsense knowledge for answering. The detailed dataset statistics can be found in Appendix A. In all our experiments, we use the same prompt settings in Manual-CoT (Wei et al., 2022b) otherwise stated. The full prompt can be found in Appendix D

Hyperparameter All experiments were conducted on an NVIDIA A800 GPU. Detailed hyperparameter settings for the experiments are provided in Appendix B. In our experiments, we tested various teacher models, including Flan-t5-xl (Scao et al., 2022), Bloom-7B(Scao et al., 2022), Llama2-70B(Touvron et al., 2023) and Llama2-13B. Correspondingly, the student models used were Flan-t5-large, Bloom-3B, Llama2-13B and Llama2-7B.

4 Results and Exploration

In this section, we delve into the empirical results of our comprehensive analysis using the GKT framework. The essence of this exploration lies in

Dataset	Model	Output Length	$\text{ACC}_{\text{teacher}}(\%)$	**ACC(%)**	$\Delta(ACC)$	Time(s)	Speed Up
GSM8K	*Single Model*						
	Llama2-7B	200	-	13.87	-	6945.70	1.31×
		300	-	14.40	-	10304.18	1.38×/13.98×
	Llama2-13B	200	-	21.23	7.36	9066.17	-
		300	-	23.65	9.25	14215.12	10.13×
	Llama2-70B	300	-	56.63	42.23 / 32.98	144018.55	-
	Bloom-3B	300	-	2.35	-	4376.05	1.00×
	Bloom-7B	300	-	4.40	2.05	4392.74	-
	GKT Framework						
	Llama2-13B→Llama2-7B	30→200	3.33	17.66	3.79	7762.62	1.17×
		30→300	3.33	17.82	3.42	10793.34	1.32×
		40→300	4.62	19.18	4.78	10871.01	1.31×
		40(concise)→300	4.62	19.26	4.86	10707.43	1.33×
	Llama2-70B→Llama2-7B	40→300	7.81	28.58	**14.18**	13440.71	**10.72** ×
	Llama2-70B→Llama2-13B	40→300	7.81	35.41	11.76	16250.34	8.86×
	Bloom-7B→Bloom-3B	40→300	2.05	2.58	0.23	5198.60	0.84×
	Llama2-7B→Bloom-3B	40→300	3.71	6.82	4.47	4154.40	1.06×
	Llama2-13B→Bloom-3B	40→300	4.62	7.73	5.38	4186.22	3.40×
	Llama2-13B→Bloom-7B	40→300	4.62	10.31	5.91	3275.90	3.82×
CSQA	*Single Model*						
	Llama2-7B	100	-	60.69	-	3239.04	1.31 × / 13.36×
		300	-	60.61	-	9321.51	1.30×
	Llama2-13B	100	-	71.17	10.48	4235.46	10.22×
		300	-	71.09	10.48	12128.50	-
	Llama2-70B	100	-	76.58	15.89 / 5.41	43285.73	-
	Bloom-3B	100	-	20.88	-	2157.97	1.05×
	Bloom-7B	100	-	21.79	-	2269.35	-
	GKT Framework						
	Llama2-13B→Llama2-7B	10→100	0.00	61.02	0.33	3472.33	1.22×
		20→100	0.00	64.70	4.01	3656.64	1.16×
		30→100	18.76	69.86	9.17	3579.21	1.18×
		30 → 300	18.76	69.86	9.25	10171.32	1.19×
		40 →300	62.74	71.01	10.40	10068.97	1.20×
		50 →300	70.43	71.09	10.48	10361.78	1.17×
	Llama2-70B→Llama2-7B	30→100	18.84	74.69	14.00	5600.38	**7.73**×
	Llama2-70B→Llama2-13B	30→100	18.84	76.16	4.99	6598.43	6.56×
	Bloom-7B→Bloom-3B	20→100	0.00	20.96	0.08	2395.36	0.95×
		30→100	11.06	20.63	-0.25	2250.04	1.01×
	Llama2-7B→Bloom-3B	30→100	21.86	40.05	19.17	2134.58	1.52×
	Llama2-13B→Bloom-3B	30→100	18.76	41.20	**20.32**	2210.29	1.92 ×
	Llama2-13B→Bloom-7B	30→100	18.76	39.80	18.01	2405.20	1.76×

Table 2: Results for GSM8K and CSQA. "→" signifies the transition from the teacher model to the student model, with settings on the left of the arrow (Model, Output Length) pertaining to the teacher model, and those on the right corresponding to the student model. "$\text{ACC}_{\text{teacher}}(\%)$" and "ACC(%)" denotes the mean accuracy(%) achieved by the teacher model alone and the overall framework, respectively. "Δ(ACC)" denotes the improvement in accuracy (%) achieved by the GKT framework compared to using only the student model (Δ(ACC) for Llama2-70B shows two numbers separated by "/", The number on the left (right) of the "/" is the change in accuracy when using Llama2-7B (Llama2-13B) as the student model). "Speed Up" indicates the acceleration factor of the GKT framework relative to using only the teacher model. ("Speed Up" for Llama2-7B shows two numbers separated by "/", The number on the left (right) of the "/" is the change in accuracy when using Llama2-13B (Llama2-70B) as the student model). "concise" denotes that we use concise guidance generation method and the detailed analysis can be found in section 4.2

quantifying the effectiveness of GKT in enhancing the accuracy of student models while ensuring computational efficiency. We first report the overall results for the GKT on the GSM8K and CSQA datasets Then, we delve into various dimensions of knowledge transfer including the optimal guidance generation methods, the intriguing dynamics between different types of teacher and student models, the influence that a teacher can exert on student and the optimal guidance length.

4.1 Overall Results

The overall results for the Guidance-based Knowledge Transfer (GKT) framework on the GSM8K and CSQA datasets are presented in Table 2. The results indicate that on the GSM8K dataset, we achieved a maximum accuracy improvement of 14.18 % compared to simply using the student

model, along with a 10.72× speed up relative to the teacher model. On the CSQA dataset, we observed a accuracy improvement of 14.00 % compared to the student model, and a maximum 7.73× speed-up in comparison to the teacher model.

4.2 How to facilitate student learning effectively?

To investigate how to generate better guidance to assist student models in answering questions, we experimented with three different guidance generation methods as described in Section 2: (1) Cut-off Guidance Generation, (2) Concise Guidance Generation, and (3) Hint Guidance Generation.

Cut-off Guidance Generation: We employed the simplest method of cutting off, where the teacher model generates only a fixed number of the first m tokens as guidance.

Concise Guidance Generation: In this approach, we added the prompt: "Provide the answer in a brief manner:" to guide the model to generate more concise guidance responses.

Hint Guidance Generation: Here, we introduced the prompt "Provide a brief hint for the question:" to encourage the teacher model not to give direct answers but to offer hints in a guiding manner, aiding the student model in generating responses.

The detailed results of these experiments on GSM8K can be seen in Table 3. To more intuitively understand the acceleration effect of each component in the Guidance-based Knowledge Transfer (GKT) framework, we have created a trace diagram of GKT's performance on the GSM8K dataset, as illustrated in Figure 3. Intriguingly, we found that providing hints leads to poorer outcomes compared to directly giving the answer. We speculate that this may be due to the limited inferential and reasoning capabilities of the smaller models. Instead of giving hints for them to infer, it might be more effective to provide direct answers. We also observed that Concise Guidance Generation was relatively effective, as the brevity of the guidance reduces the inferential workload for the student model. By prompting the teacher model to produce shorter answers, this method not only improved the accuracy of the model's responses but also accelerated the inference speed.

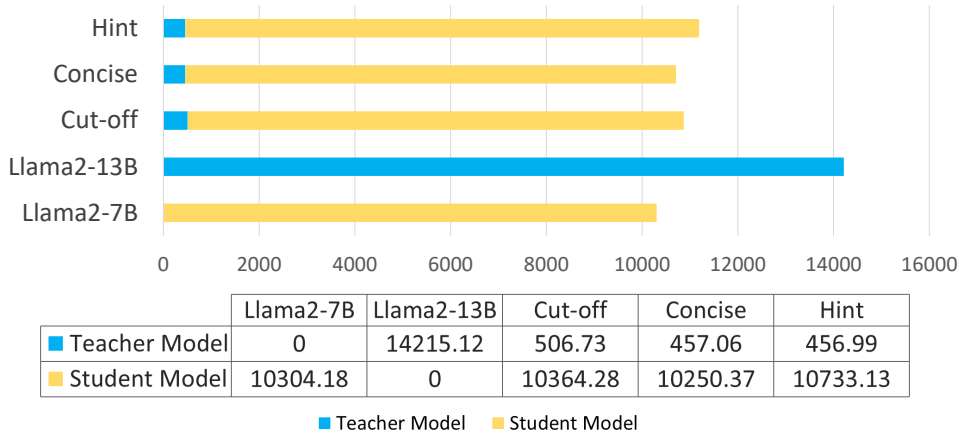
GKT for Cloud-edge Collaboration LLM Deployment Figure 3 demonstrates that utilizing only the Llama2-13B model for the GSM8K dataset, comprising a total of 1319 examples, results in an average response time of 10.78 seconds per example ($14215.12 \div 1319 = 10.78s$). This implies a single-user service capability within this timeframe. Conversely, the deployment of the GKT framework in a Cloud-Edge collaboration environment for LLMs markedly reduces the large model's response time to 0.38 seconds on average ($506.73 \div 1319 = 0.38s$), and the small model processes each example in 7.86 seconds ($10364.28 \div 1319 = 7.86s$), culminating in a total response time of 8.24 seconds. Therefore, theoretically, by employing batch processing, **the GKT framework can facilitate simultaneous service to 24 users within the 8.24-second window.** This efficiency stems from the large model's capacity for batch processing in the cloud, which can concurrently serve multiple users (with a batch size of 24). At the same time, the small models are deployed on distinct edge devices, facilitating parallel operations and enabling personalized user experiences. **In stark contrast, reliance solely on the Llama2-13B model limits service to a single user within the 10.78-second timeframe.** Hence, the GKT framework substantially augments the parallelism in user service provision in Cloud-Edge collaborative LLM deployments.

4.3 How To Identify the Right Teacher for the Right Student ?

To find the right student for the right teacher, we experimented with the decoder-only model Llama and the encoder-decoder model Flan-t5. Our experiments show that both types of models can achieve certain improvements through GKT. However, an intriguing phenomenon we observed was that replacing the teacher model of Flan-t5 with a larger Llama model led to an accuracy decrease, rather than an increase. We speculate that one possible reason could be that for encoder-decoder models, the encoder and decoder use different parameters. Using a large model's answers as input for the smaller model's encoder may disrupt the coherence of thought, leading to reduced inferential ability. In contrast, for decoder-only models, directly concatenating inputs and targets seems to aid in better inference. The significant structural differences between T5 and Llama models could result in inconsistent thinking patterns between models, implying that the teacher and student are not perfectly in sync, thereby diminishing the overall effectiveness.

Model	Prompt	Output Length	**ACC(%)**	$\Delta(ACC)$	Time(s)	Speed Up
Single Model						
Llama2-7B	-	300	14.40	-	10304.18	1.38×
Llama2-13B	-	300	23.65	9.25	14215.12	-
GKT Framework						
	-	40→300	19.18	4.78	10871.01	1.31×
Llama2-13B→Llama2-7B	"Provide the answer in a brief manner: "	40(concise)→300	19.26	4.86	10707.43	1.33×
	"Provide a brief hint for the question: "	40(hint)→300	19.11	4.71	11190.12	1.27×

Table 3: Results of different guidance generation methods on GSM8K. "concise" and "hint" denotes Concise Guidance Generation and Hint Guidance Generation respectively



	Llama2-7B	Llama2-13B	Cut-off	Concise	Hint
Teacher Model	0	14215.12	506.73	457.06	456.99
Student Model	10304.18	0	10364.28	10250.37	10733.13

Figure 3: The trace diagram for GKT framework on GSM8K. The vertical axis represents the guidance generation method used (where "Llama2-13B" indicates the exclusive use of the Llama2-13B model for response generation, and "Llama2-7B" denotes the sole use of the Llama2-7B model). The horizontal axis represents the runtime (s) of the GKT across the entire dataset.

4.4 How Can a Teacher Influence His Student?

Based on the above findings, we conducted further experiments to explore the impact of a teacher's guidance on student models, specifically focusing on decoder-only models. In these experiments, we used Llama2 and Bloom, both decoder-only models. The overall results can be seen in Table 5

Table 5 presents a detailed overview of the impact of teacher model choice on the performance and efficiency of student models, as tested on the GSM8K and CSQA datasets. The results highlight the differential effects on accuracy and processing speed, depending on the combination of teacher and student models used. The results also revealed that the Llama model possesses stronger inferential capabilities and stores more common-sense knowledge, performing better on both datasets. When we provided the same Bloom student model with the more experienced teacher model Llama, under similar-sized teacher model conditions, we observed that the Llama model, as a teacher, could enhance the student model's accuracy by 20 %. In contrast, using Bloom-7B as the teacher resulted in a decrease in accuracy. This outcome underscores the critical influence of the teacher on the student.

Additionally, by comparing results on the GSM8K and CSQA datasets, we found that this approach of using larger models to guide smaller ones can better transfer the common-sense knowledge stored in the teacher model. However, it had a less pronounced effect on mathematical reasoning abilities in the GSM8K dataset. On the CSQA dataset, when employing a Llama2-7B teacher model, Bloom-3B model's accuracy is improved by nearly 20 %, with a 1.52 times speed increase. These results convincingly demonstrate that our method can effectively transfer the knowledge and reasoning abilities stored in larger models to smaller ones.

To delve deeper into the extent of influence a teacher model can have on a student model, we utilized the superior-performing ChatGPT (OpenAI, 2023) as the teacher model, with Llama2-7B, 13B, and 70B serving as the student model. The Table 6 clearly demonstrates that using a more powerful teacher model significantly enhances performance. When the number of guidance tokens is kept constant, employing ChatGPT as the teacher

Model	Output Length	**ACC(%)**	$\Delta(ACC)$	Time(s)	Speed Up
Single Model					
Flan-t5-large (0.8B)	300	6.90	-	2343.06	1.09×
Flan-t5-xl (3B)	300	11.30	4.40	2557.76	-
Llama2-7B	300	14.40	-	10304.18	1.38×
Llama2-13B	300	23.65	12.35	14215.12	-
GKT Framework					
Flan-t5-xl→Flan-t5-large	40→300	7.88	0.98	2190.05	1.17×
Llama2-13B→Llama2-7B	40→300	19.18	4.78	10871.01	1.31×
Llama2-13B→Flan-t5-large	40→300	6.52	-0.38	2330.48	6.10×
Llama2-7B→Flan-t5-large	40→300	6.21	-0.69	2284.67	4.51×

Table 4: Decoder-only Model VS Encoder-Decoder Model. We experimented with the decoder-only model Llama and the encoder-decoder model Flan-t5.

Model	Output Length	**ACC(%)**	$\Delta(ACC)$	Time(s)	Speed Up
GSM8K					
Llama2-13B→Llama2-7B	40→300	19.18	4.78	10871.01	1.31×
Bloom-7B→Bloom-3B	40→300	2.58	0.23	5198.60	0.84×
Llama2-7B→Bloom-3B	40→300	6.82	4.47	4154.40	1.06×
Llama2-13B→Bloom-3B	40→300	7.73	5.38	4186.22	3.40×
Llama2-13B→Bloom-7B	40→300	10.31	5.91	3725.90	3.82×
CSQA					
Llama2-13B→Llama2-7B	30→100	69.86	9.17	3579.21	1.18×
Bloom-7B→Bloom-3B	30→100	20.63	-0.25	2250.04	1.01×
Llama2-7B→Bloom-3B	30→100	40.05	19.17	2134.58	1.52×
Llama2-13B→Bloom-3B	30→100	41.2	20.32	2210.29	1.92×
Llama2-13B→Bloom-7B	30→100	39.8	18.01	2405.2	1.76×

Table 5: Comparative analysis of teacher influence on student models using decoder-only models, Llama and Bloom.

	Average Output Length		**ACC**	$\Delta(ACC)$
	ChatGPT	Ours		
Full ChatGPT	76.68	-	68.16	-
Llama2-7B	-	10 → 300	19.41	0.23
	-	20 → 300	27.52	8.34
	-	30 → 300	32.83	13.65
	-	40 → 300	40.79	21.61
Llama2-13B	-	40 → 300	48.14	24.49
Llama2-70B	-	40 → 300	64.75	8.12

Table 6: Result on GSM8K when using ChatGPT as teacher model and Llama2-7B, 13B, 70B as student model. "→" signifies the output length transition from ChatGPT to the Llama2 model. "Δ(ACC)" denotes the improvement in accuracy (%) achieved by GKT framework compared to using only the corresponding Llama2 model. "Full ChatGPT" denotes the ChatGPT performance on GSM8K without GKT

model results in a substantial improvement over using Llama2-13B as the teacher model. Specifically, on the GSM8K dataset, there's an increase of 21.53 % compared to Llama2-13B teacher (from 19.26% to 40.79%). We can also see from the table that stronger student model can preserve more teacher model abilities. From another perspective, this approach allows for more cost-effective outcomes. When utilizing ChatGPT's API interface and employing the best performance Llama2-70B as the student model, we can achieve 95.00% of ChatGPT's performance at 52% of the cost, effectively showcasing the GKT framework's "cheap and cheerful" charm.

4.5 Further Exploration

In exploring the ideal amount of guidance a teacher model should offer to student models, we plotted accuracy against varying guidance lengths for the GSM8K and CSQA datasets (from 10 to 40 tokens). The detailed experiment can be found in Appendix C.1. Our results indicate that the teacher model should output the first 40 tokens for the GSM8K dataset and the first 30 tokens for the CSQA dataset, as these lengths optimize performance. We also investigated the influence of few-shot exemplars on student models' performance in Appendix C.2. The overall performance improved with an increase of exemplar number. Consequently, we chose 8-shot exemplars for GSM8K and 7-shot for CSQA,

aligning with the Manual-CoT (Wei et al., 2022b).

5 Related Work

The aspect most relevant to our work is knowledge distillation(Tunstall et al., 2023; Jiang et al., 2023; Chiang et al., 2023; Li et al., 2023; Chen et al., 2023). Knowledge distillation involves condensing the expertise from larger models into more compact versions, thereby ensuring these smaller models retain high efficiency while still achieving impressive performance. Li et al. (2023) introduces Symbolic Chain-of-Thought Distillation (SCoTD), which trains a smaller "student" language model using the outputs (reasoning chains) of a larger "teacher" model. The teacher model first generates multiple reasoning chains for a given task, showcasing step-by-step problem-solving. The student model then are fine-tuned on these examples, essentially mimicking the teacher's reasoning process. This training enables the student model to perform complex reasoning tasks more effectively, despite its smaller size. Tunstall et al. (2023) presents a distillation framework named Distilled Direct Preference Optimization (dDPO) result in a 7B model named ZEPHYR. The method comprises three steps: (1) distilled supervised fine-Tuning (2) AI feedback through preferences and (3) distilled direct preference optimization. Different from knowledge distillation, our work focuses on leveraging the knowledge of larger models to enhance the overall efficiency and performance. The proposed GKT framework circumvents the usual requirements of producing a distillation dataset from a teacher model or fine-tuning the student model, streamlining the knowledge transfer process.

Another line of study that is related to our work is speculative decoding (Leviathan et al., 2023a). Speculative decoding uses two models: the original target model and a much smaller approximate model. The smaller model handles autoregressive sampling, while the larger assesses the output. Simple tokens are generated by the smaller model, with complex tokens handled by the larger. Based on speculative decoding, (He et al., 2023) proposed Retrieval-Based Speculative Decoding (REST) which combines speculative decoding with retrieval techniques. Instead of using a smaller LM for draft generation, REST bypasses the need for an additional small LM by retrieving draft tokens from a pre-built datastore containing context-continuation pairs. These drafts are then verified by a large LM.

Recently, Ning et al. (2023) proposed skeleton-of-thought (SoT). SoT first guides LLMs to first create a concise "skeleton" of an answer and then fill in each point of the skeleton in parallel, speeding up the response process. Xu et al. (2023) introduces Super In-Context Learning (SuperICL), a method enhancing the performance of large language models (LLMs) by integrating them with smaller, locally fine-tuned models. These smaller models, acting as plug-ins, provide task-specific knowledge and predictions. The process involves fine-tuning a small model on task-specific data, using it to generate predictions and confidence scores for in-context examples, and combining these with the LLM's general language understanding. This approach aims to overcome limitations of In-Context Learning (ICL) in handling larger datasets, improving both performance and stability on supervised tasks. While SuperICL also integrates LLMs and LMs, it primarily focuses on enhancing model performance without considering the efficiency of the framework. This focus on performance enhancement can even negatively impact overall efficiency. Whereas, GKT focuses on finding the optimal balance between efficiency and effectiveness

6 Conclusion

We introduce the innovative Guidance-based Knowledge Transfer (GKT) framework, designed to leverage the knowledge of larger models to enhance the efficiency and performance of smaller models, while maintaining the flexibility of personalized generation settings, allowing users to freely adjust the generation parameters. The unique collaborative framework of GKT seamlessly integrates into cloud-edge architectures, deploying smaller models on edge devices to minimize data transmission delays and expedite response generation. Our results demonstrate remarkable improvements: a maximum accuracy increase of 14.18% and a 10.72× speed-up on the GSM8K dataset, and a 14.00% accuracy enhancement with a 7.73× speed increase on the CSQA dataset. Moreover, when utilizing ChatGPT as teacher model and Llama2-70B as the student model, we can achieve 95.00% of ChatGPT's performance at 52% of the cost. GKT signify major strides in performance metrics, combining accuracy with computational speed-ups - all wrapped up in a "cheap and cheerful" package.

Limitation

The Guidance Knowledge Transfer (GKT) framework exhibits certain limitations compared to traditional knowledge distillation methods. Knowledge distillation primarily relies on a pre-trained smaller model, often leading to faster inference speeds. However, the GKT framework still depends on a large model to generate guidance prompts during inference. This approach aims to reduce the limitations brought about by fine-tuning through performance loss during inference while supporting customized configurations of the framework.

Moreover, the length of guidance prompts generated by the GKT framework for different datasets is specific. Although the generation method is universally applicable, finding a universally appropriate length for guidance prompts that suits various scenarios remains a challenge. This means that additional adjustments and optimizations may be necessary for different datasets and application contexts to ensure optimal performance. Therefore, one of the future research directions is to explore how to more effectively determine the appropriate lengths of guidance prompts to enhance the universality and flexibility of the GKT framework.

References

Zeming Chen, Qiyue Gao, Antoine Bosselut, Ashish Sabharwal, and Kyle Richardson. 2023. DISCO: distilling counterfactuals with large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5514–5528. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Saryn R Goldberg, Jennifer Andrea Rich, and Amy Masnick. 2014. The use of metacognitive writing-to-learn prompts in an engineering statics class to improve student understanding and performance. In *2014 ASEE Annual Conference & Exposition*, pages 24–1251.

Zhenyu He, Zexuan Zhong, Tianle Cai, Jason D. Lee, and Di He. 2023. REST: retrieval-based speculative decoding. *CoRR*, abs/2311.08252.

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

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

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023a. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023b. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2665–2679, Toronto, Canada. Association for Computational Linguistics.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Huazhong Yang, and Yu Wang. 2023. Skeleton-of-thought: Large language models can do parallel decoding. *CoRR*, abs/2307.15337.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. 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)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of LM alignment. *CoRR*, abs/2310.16944.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Canwen Xu, Yichong Xu, Shuohang Wang, Yang Liu, Chenguang Zhu, and Julian J. McAuley. 2023. Small models are valuable plug-ins for large language models. *CoRR*, abs/2305.08848.

Junho Yim, Donggyu Joo, Ji-Hoon Bae, and Junmo Kim. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 7130–7138. IEEE Computer Society.

Appendix

A Dataset Statistics

The detailed dataset statistics can be found in Table 7

Dataset	GSM8K	CSQA
#Instance	1319	1221
Average input length (Words)	47	28
Answer Format	int	str
Question Format	grade school math problems	single-choice question
Few-shot	8	7

Table 7: Dataset statistics

B Hyperparameters

The detailed hyperparameter settings can be found in Table 8

Parameters	Value
batch size (Bloom 7B)	32
batch size (Llama2 13B)	24
batch size (Llama2 70B)	10
top_p	0.9
temperature	0.8
max_seq_len	1024

Table 8: Hyperparameters

C Further Exploration

C.1 How much guidance should a teacher offer to students ?

To investigate the optimal amount of assistance a teacher should provide to a student, or in other words, to determine the most suitable guidance length for maximizing student benefits, we conducted experiments. Intuitively, one might assume that the more a teacher model outputs, the higher the accuracy of the model's response. However, longer guidance tends to weaken the model's acceleration effect. Therefore, we plotted a line graph showing the changes in accuracy as the guidance length varied from 10 to 40, in intervals of 10, as depicted in Figure 5. Based on our experiments, we established that for the GSM8K dataset, the teacher model should output the first 40 tokens of the answer. For the CSQA dataset, which generally requires shorter responses, we set the teacher model to output the first 30 tokens.

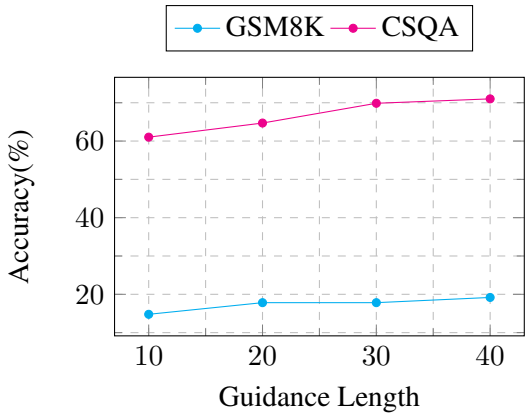


Figure 4: Performance of different length ranges. We use Llama2-13B as teacher model and Llama2-7B as student model.

C.2 How few-shot exemplars affect students?

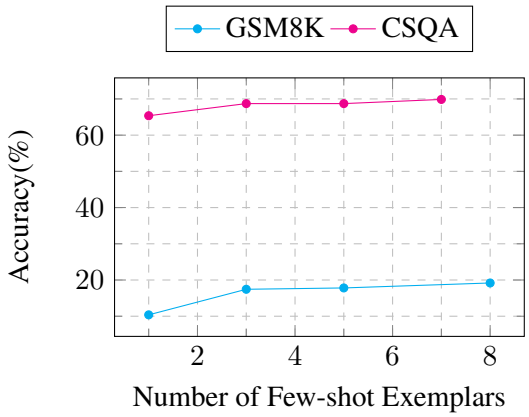


Figure 5: Performance of different few-shot exemplar number

In this investigation, we focused on the role of few-shot exemplars in influencing the performance of student models. We used a range of few-shot exemplars, provided through Manual-CoT (Wei et al., 2022b), acros GSM8K and CSQA. The variation in accuracy with the change in the number of few-shot exemplars is illustrated in Figure 5. As depicted in Figure 5, it can be observed that the performance of the student models improves with the increase in the number of exemplars. Thus, in this experiment, we select 8-shot for GSM8K and 7-shot for CSQA in line with the Manual-CoT.

D Prompt used for GSM8K and CSQA

The prompt used for GSM8K and CSQA can be found in Figure 6 and Figure 7 respectively.

Prompt for GSM8K

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?
A: "There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been 21 - 15 = 6.
The answer is 6.

Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?
A: There are originally 3 cars. 2 more cars arrive. 3 + 2 = 5.
The answer is 5.

Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?
A: Originally, Leah had 32 chocolates. Her sister had 42. So in total they had 32 + 42 = 74. After eating 35, they had 74 - 35 = 39.
The answer is 39.

Q: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny?
A: Jason started with 20 lollipops. Then he had 12 after giving some to Denny. So he gave Denny 20 - 12 = 8.
The answer is 8.

Q: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now?
A: Shawn started with 5 toys. If he got 2 toys each from his mom and dad, then that is 4 more toys. 5 + 4 = 9.
The answer is 9.

Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?
A: There were originally 9 computers. For each of 4 days, 5 more computers were added. So 5 * 4 = 20 computers were added. 9 + 20 is 29.
The answer is 29.

Q: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday?
A: Michael started with 58 golf balls. After losing 23 on tuesday, he had 58 - 23 = 35. After losing 2 more, he had 35 - 2 = 33 golf balls.
The answer is 33.

Q: Olivia has $23. She bought five bagels for $3 each. How much money does she have left?
A: Olivia had 23 dollars. 5 bagels for 3 dollars each will be 5 x 3 = 15 dollars. So she has 23 - 15 dollars left. 23 - 15 is 8.
The answer is 8.

Figure 6: The prompt used for GSM8K

Prompt for CSQA

Q: What do people use to absorb extra ink from a fountain pen? Answer Choices: (a) shirt pocket (b) calligrapher's hand (c) inkwell (d) desk drawer (e) blotter
A: The answer must be an item that can absorb ink. Of the above choices, only blotters are used to absorb ink.
The answer is (e).

Q: What home entertainment equipment requires cable?
Answer Choices: (a) radio shack (b) substation (c) television (d) cabinet
A: The answer must require cable. Of the above choices, only television requires cable.
The answer is (c).

Q: The fox walked from the city into the forest, what was it looking for? Answer Choices: (a) pretty flowers (b) hen house (c) natural habitat (d) storybook
A: The answer must be something in the forest. Of the above choices, only natural habitat is in the forest.
The answer is (b).

Q: Sammy wanted to go to where the people were. Where might he go? Answer Choices: (a) populated areas (b) race track (c) desert (d) apartment (e) roadblock
A: The answer must be a place with a lot of people. Of the above choices, only populated areas have a lot of people.
The answer is (a).

Q: Where do you put your grapes just before checking out? Answer Choices: (a) mouth (b) grocery cart (c)supermarket (d) fruit basket (e) fruit market
A: The answer should be the place where grocery items are placed before checking out. Of the above choices, grocery cart makes the most sense for holding grocery items.
The answer is (b).

Q: Google Maps and other highway and street GPS services have replaced what? Answer Choices: (a) united states (b) mexico (c) countryside (d) atlas
A: The answer must be something that used to do what Google Maps and GPS services do, which is to give directions. Of the above choices, only atlases are used to give directions.
The answer is (d).

Q: Before getting a divorce, what did the wife feel who was doing all the work? Answer Choices: (a) harder (b) anguish (c) bitterness (d) tears (e) sadness
A: The answer should be the feeling of someone getting divorced who was doing all the work. Of the above choices, the closest feeling is bitterness.
The answer is (c).

Figure 7: The prompt used for CSQA