DistilQwen2.5: Industrial Practices of Training Distilled Open Lightweight Language Models

Chengyu Wang[*], **Junbing Yan**[*], **Yuanhao Yue, Jun Huang**
Alibaba Cloud Computing, Hangzhou, China
{chengyu.wcy, yanjunbing.yjb, yueyuanhao.yyh,
huangjun.hj}@alibaba-inc.com

Abstract

Enhancing computational efficiency and reducing deployment costs for large language models (LLMs) have become critical challenges in various resource-constrained scenarios. In this work, we present *DistilQwen2.5*, a family of distilled, lightweight LLMs derived from the public *Qwen2.5* models. These distilled models exhibit enhanced instruction-following capabilities compared to the original models based on a series of distillation techniques that incorporate knowledge from much larger LLMs. In our industrial practice, we first leverage powerful proprietary LLMs with varying capacities as multi-agent teachers to select, rewrite, and refine instruction-response pairs that are more suitable for student LLMs to learn. After standard fine-tuning, we further leverage a computationally efficient model fusion approach that enables student models to progressively integrate fine-grained hidden knowledge from their teachers. Experimental evaluations demonstrate that the distilled models possess significantly stronger capabilities than their original checkpoints. Additionally, we present use cases to illustrate the applications of our framework in real-world scenarios. To facilitate practical use, we have released all the *DistilQwen2.5* models to the open-source community. [1]

1 Introduction

Large language models (LLMs) have emerged as a transformative technology in NLP, powering a wide array of applications from machine translation to conversational agents (Zhao et al., 2023). However, the rise of LLMs has been accompanied by several challenges, notably the substantial computational resource requirements and high deployment costs. Reducing the parameter sizes of LLMs while maintaining or even improving performance has become a critical area of research.

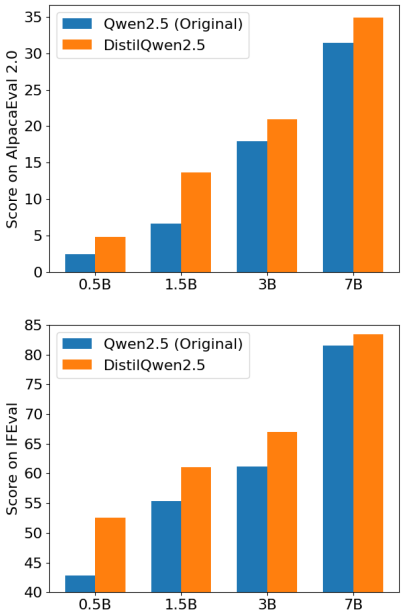


Figure 1: Brief comparison between original *Qwen2.5* and *DistilQwen2.5* models in terms of AlpacaEval 2.0 (length-controlled) and IFEval scores.

Knowledge distillation (KD) is a promising approach to addressing these challenges by transferring knowledge from a larger model (the teacher) to a smaller model (the student) (Xu et al., 2024). Previous works have primarily focused on specific KD techniques to develop more robust student models (Hsieh et al., 2023; Gu et al., 2024; Yue et al., 2024b; Zhang et al., 2024). However, there is a lack of studies investigating good industrial practices that create a series of distilled lightweight LLMs with varying sizes and capacities.

In this paper, we introduce *DistilQwen2.5*, a series of distilled LLMs derived from the *Qwen2.5*

[*] C. Wang and J. Yan contributed equally to this work. Correspondence to: C. Wang.

[1] Our trained lightweight models and our processed large instruction-following dataset are released in HuggingFace. Please refer to the four models DistilQwen2.5-0.5B-Instruct, DistilQwen2.5-1.5B-Instruct, DistilQwen2.5-3B-Instruct, DistilQwen2.5-7B-Instruct and the dataset DistilQwen_100k.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 32–42

models[2]. In the beginning of the KD process, proprietary teacher LLMs, serving as multiple agents, are utilized to select, rewrite, and refine instruction-response pairs, tailoring them to be more conducive to learning by smaller student models. In particular, a Chain-of-Thought (CoT) (Wei et al., 2022) rewriting approach is employed to significantly enhance the reasoning abilities of the distilled models. Beyond standard fine-tuning, we further introduce a model fusion approach to enable student models to incrementally integrate fine-grained hidden knowledge from their teacher models in a computationally efficient manner. This approach enhances the depth of understanding in student models beyond what black-box distillation processes can achieve.

In our experiments, we demonstrate that the resulting *DistilQwen2.5* models show remarkable improvements in instruction-following performance across various NLP tasks compared to their original counterparts. Briefly, we present the AlpacaEval 2.0 (length-controlled) (Dubois et al., 2024) and IFEval (Zhou et al., 2023) scores of the *DistilQwen2.5* models in Figure 1. To enhance the public accessibility of our work, all models have been made available to the open-source community. Furthermore, we describe two use cases to demonstrate the applications of our work in real-world scenarios.

2 Related Work and Discussion

Knowledge distillation (KD), originally proposed by Hinton et al. (2015), has emerged as a key technique for improving the efficiency of neural networks. Prior to the era of LLMs, several studies successfully demonstrated the distillation of BERT-based models (Sanh et al., 2019; Jiao et al., 2020; Sun et al., 2020; Pan et al., 2021; Hou et al., 2023), primarily focusing on specific NLP tasks. However, distillation for LLMs presents unique challenges due to the intricate dependencies among prediction tokens. In the literature, f-Distill (Wen et al., 2023) minimizes a generalized f-divergence function for sequence-level KD. MiniLLM (Gu et al., 2024) introduces a reverse Kullback-Leibler divergence (KLD) objective to distill knowledge from white-box LLMs to student models. Wu et al. (2025) propose an adaptive approach that allocates weights to combine forward and reverse KLD objectives. FuseLLM (Wan et al., 2024) merges multiple powerful LLMs into a more capable student model.

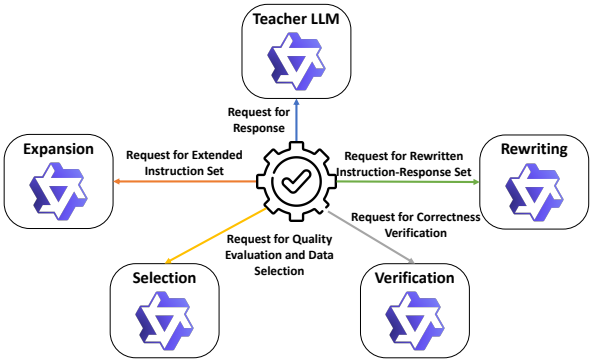


Figure 2: Functionalities for LLMs/agents used in data augmentation and black-box distillation. **Disclaimer:** We use the Qwen logo in the figure; however, any LLMs with sufficient capabilities can be used as well.

Given that many powerful LLMs are accessible only through APIs, KD from proprietary LLMs to smaller open-source models (referred to as black-box KD) has garnered significant attention (Hsieh et al., 2023). To facilitate distillation from more advanced LLMs, some researchers leverage these models for data augmentation to fine-tune student LLMs (Yue et al., 2024a). Li et al. (2024) utilize the data selection capabilities of student LLMs to refine instruction-tuning data. Lou et al. (2024) generate multi-faceted instructions for diverse tasks to enhance black-box KD. Additionally, Yue et al. (2024b) propose a task-aware curriculum planning framework to improve instruction refinement.

In contrast to prior work, our approach emphasizes industrial practices that leverage the strengths of both black-box and white-box KD methods. Moreover, efficiency remains a critical barrier in industry, particularly for white-box KD. To address this, our work incorporates an efficient algorithm to integrate hidden knowledge from teacher models.

3 Our Approach

In this section, we describe the industrial practices for distilling the *DistilQwen2.5* models.

3.1 Multi-Agent Data Augmentation as Black-Box Knowledge Distillation

We first leverage multi-agent data augmentation as black-box KD, where proprietary teacher models serve as the sources of knowledge. This approach is more computationally efficient than white-box KD and allows us to select more powerful proprietary models as teachers. In our work, we employ

[2] https://qwenlm.github.io/blog/qwen2.5/

Qwen-max[3] to process the Chinese texts due to its strong capabilities in handling the Chinese language, and GPT-4/GPT-4o for other languages. In Figure 2, we can see that a controller coordinates the entire pipeline of generating responses directly from the teacher model and invoking LLM agents to augment the training data. The functionalities of these LLM agents are described below.

Expansion Agent. The expansion agent is employed to generate a diverse set of instruction variations, ensuring that student models are exposed to a comprehensive range of instructions. Importantly, it preserves the original NLP task category of the input instruction to prevent hallucinations and semantic drift caused by LLMs. For example, given the input "Provide a brief overview of Newton's First Law of Motion", the output could be "Explain the meaning of Kepler's Third Law", but not "Give me a brief introduction to Albert Einstein's life". After instruction expansion, we also call the teacher model to generate responses for new instructions.

Rewriting Agent. The rewriting agent further enhances the quality and diversity of the training data. Unlike the expansion agent, the rewriting agent operates under stringent constraints to preserve the semantic integrity of the tasks expressed in instructions, ensuring that the rewritten content remains faithful to the original intent and task category. For example, the instruction "Provide a summary of the economic impacts of climate change" might be rewritten as "Explain how climate change affects the economy". Regarding the generated responses, we encourage them to be Chain-of-Thought (CoT) outputs for complex tasks such as logical reasoning, mathematical problems, and code generation (Wei et al., 2022), as this significantly enhances the cognitive reasoning abilities of distilled, small models (Hsieh et al., 2023; Yue et al., 2024b).

Selection Agent. The selection agent automatically evaluates and chooses instruction-response pairs that are highly valuable for training the student model. This selection process is guided by various heuristic criteria, including informativeness, helpfulness, and potential for generalization to similar tasks. Additionally, we consider task balance when selecting these pairs, following the approach of Yue et al. (2024b). This guides the controller to filter out less useful data instances.

Verification Agent. Different from the selection agent, the verification agent is invoked each time new instruction-response instances are generated by LLMs to check the factual correctness. Specifically, we leverage the underlying LLMs to check whether the instructions are reasonable and whether the responses correctly solve the tasks expressed by the instructions.

Overall, the augmented dataset leverages a black-box KD method by encapsulating the distilled knowledge from larger models into training examples for student models. The distillation training process follows a supervised learning paradigm, utilizing the augmented instruction-response pairs.

3.2 Efficient Model Fusion as White-Box Knowledge Distillation

In contrast to black-box KD, white-box KD involves having the student model mimic the distribution of the teacher model's logits, providing richer knowledge compared to learning from only the token with the highest output probability. In our work, we conduct white-box KD after the completion of black-box KD to maximize the utility of computational resources and aim to further improve the performance of student models by learning richer knowledge. We assume that the student model, with learnable parameters θ, has a probability function p_S^θ that is differentiable with respect to θ. The token-level logits difference between p_T (from the teacher model) and p_S^θ (from the student model) is defined as follows:

$$D_\theta(x, y) = \frac{1}{L} \sum_{n=1}^{L} D_\theta \left(p_T(\cdot \mid y_{<n}, x) \parallel p_S^\theta(\cdot \mid y_{<n}, x) \right), \tag{1}$$

where x and y denote the input and output sequences, respectively, and L is the sequence length. The function $D_\theta(\cdot)$ can be any divergence measurement, such as KLD (Gu et al., 2024), reverse KLD (Wu et al., 2025), etc. The KD loss aims to minimize the divergence between the token sequences of the student and the teacher:

$$L(\theta) = \mathbb{E}_{(x,y)\sim(X,Y)} \left[D_\theta(x, y) \right]. \tag{2}$$

For industrial-scale implementation, it is infeasible to leverage existing white-box KD approaches such as those by Gu et al. (2024) and Wu et al. (2025). The reasons are twofold: i) If the forward pass of the teacher model is performed simultaneously with the training of the student model, the GPU memory consumption becomes excessively high, especially when the teacher model is very

[3] https://qwenlm.github.io/

Model	AlpacaEval 2.0 (Length-Controlled)	MT-Bench	MT-Bench (Single)	IFEval (instruct-loose)	IFEval (strict-prompt)
Qwen2.5-0.5B-Instruct	2.46	5.49	6.26	42.81	30.31
DistilQwen2.5-0.5B-Instruct*	4.72	5.71	6.74	51.44	37.15
DistilQwen2.5-0.5B-Instruct	**4.89**	**5.78**	**6.83**	**52.61**	**37.82**
Qwen2.5-1.5B-Instruct	6.69	7.09	7.66	55.40	40.11
DistilQwen2.5-1.5B-Instruct*	13.30	7.27	7.90	60.63	73.02
DistilQwen2.5-1.5B-Instruct	**13.69**	**7.35**	**7.99**	**61.10**	**74.49**
Qwen2.5-3B-Instruct	17.98	7.92	8.40	61.18	74.58
DistilQwen2.5-3B-Instruct*	20.81	8.33	8.94	65.80	77.10
DistilQwen2.5-3B-Instruct	**20.91**	**8.37**	**8.97**	**67.03**	**77.36**
Qwen2.5-7B-Instruct	31.43	8.52	8.83	81.53	72.10
DistilQwen2.5-7B-Instruct*	34.78	8.75	9.19	83.41	73.20
DistilQwen2.5-7B-Instruct	**34.86**	**8.76**	**9.22**	**83.48**	**73.27**

Table 1: Performance comparison between the original *Qwen2.5* model and the *DistilQwen2.5* models in terms of instruction-following abilities across four parameter sizes: 0.5B, 1.5B, 3B, and 7B. Note: * indicates a variant of our model utilizing black-box KD over processed datasets.

large (e.g., 32B/72B). ii) The vocabulary of the teacher and student models may not match, leading to a mismatch of the logits tensors of both models.

In our work, we observe that the sum of the probabilities of the top-10 tokens is almost equal to 1. This indicates that nearly all the knowledge of the teacher model is contained within the top-10 tokens. Therefore, we build a scalable white-box KD system that supports the following features: i) A *token alignment* operation (Wan et al., 2024) is first conducted if the logits tensors of both models do not match. ii) A distributed computing process is executed offline to generate the teacher model's logits with top-K probabilities, where $K = 10$ is set as default and adjustable for customized scenarios. iii) A variant of $D_\theta(\cdot)$ is implemented where only the top-K elements are calculated for divergence minimization. Let

$$\mathbf{z}_T = [z_T^{(1)}, z_T^{(2)}, \cdots, z_T^{(K)}] \tag{3}$$

$$\mathbf{z}_S = [z_S^{(1)}, z_S^{(2)}, \cdots, z_S^{(K)}] \tag{4}$$

be the top-K logits from the teacher model, and the corresponding logits from the student model with matched indices in the vocabulary. The probabilities for computing $D_\theta(\cdot)$ is then calculated as follows:

$$\mathbf{p}_T = \frac{\exp(\mathbf{z}_T/\mathcal{T})}{\sum_{k=1}^{K}\exp(z_T^{(k)}/\mathcal{T})} \tag{5}$$

$$\mathbf{p}_S = \frac{\exp(\mathbf{z}_S/\mathcal{T})}{\sum_{k=1}^{K}\exp(z_S^{(k)}/\mathcal{T})} \tag{6}$$

where $\mathcal{T}$ is the temperature hyperparameter. This approach not only reduces computation time but also improves the speed of storing and reading the logits, alleviating the storage pressure of our cloud computing system.

4 Experimental Evaluation

In this section, we present experimental setups and evaluation results of the *DistilQwen2.5* models. Due to the space limitations, case studies are further presented in the appendix.

4.1 Experimental Setup

The initial dataset consists of instruction-response pairs collected from several popular public datasets, including OpenHermes 2.5[4], the Cleaned Alpaca Dataset[5], and LCCD (Wang et al., 2020), together with our in-house datasets. The pre-processing steps follow the method presented in (Yue et al., 2024a). Subsequently, the instruction-response pairs are carefully expanded, rewritten, verified and selected. To create a series of smaller student LLMs, we utilize the *Qwen2.5* series as our backbone models, including their instruct versions with varying sizes: 0.5B, 1.5B, 3B, and 7B. The white-box teacher models are selected from `Qwen2.5-14B/32B/72B-Instruct`. For student model distillation, the default learning rate and the epochs are set to 1×10^{-5} and 3, respectively. We train all the models on a server equipped with eight A800 GPUs, each with 80GB memory.

4.2 Evaluation Benchmarks

AlpacaEval 2.0 (length-controlled) (Dubois et al., 2024) assesses the instruction-following capabilities of LLMs across various domains. MT-Bench (Bai et al., 2024) is utilized to evaluate the multitasking abilities of our models. This bench-

[4] https://huggingface.co/datasets/teknium/OpenHermes-2.5

[5] https://github.com/gururise/AlpacaDataCleaned

mark challenges models with diverse tasks that require an understanding of multiple domains and the ability to quickly adapt to changing instructions, under both single-turn and multi-turn conversation settings. IFEval (Zhou et al., 2023) assesses how models perform during dynamic user interactions. For rigorous comparison, we report the results in both instruct-loose and strict-prompt settings.

4.3 Main Experimental Results

The results of our experiments are summarized in Table 1. As illustrated, the *DistilQwen2.5* models demonstrate superior performance across all benchmarks, outperforming both the baseline and original models by significant margins. Moreover, the proposed model fusion technique enhances the models' capabilities after the black-box KD process. We further observe that the improvement is more pronounced for smaller student backbones. Specifically, the improvement of `DistilQwen2.5-0.5B-Instruct` compared to `Qwen2.5-0.5B-Instruct` is larger than that of `DistilQwen2.5-7B-Instruct` compared to `Qwen2.5-7B-Instruct`. This shows that the potential of smaller students is larger in terms using KD. Overall, the experimental results empirically validate our distillation framework, demonstrating its effectiveness in enhancing the task-solving performance of lightweight LLMs.

4.4 Analysis on White-Box KD

Inference Speed of Teacher Logits Generation. In our experiments, we measure the latency associated with generating logits across different sizes of teacher models, as shown in Figure 3. Our implementation achieves a significantly accelerated inference speed, obtaining a $3\times$ to $5\times$ speedup compared to the vanilla implementation. Additionally, the reduction in logits does not lead to any noticeable decrease in the instruction-following abilities of the distilled smaller models, as revealed by our exploratory experiments.

Sum of Probabilities of Top-K Tokens. We further adjust the value of K and compute the sum of probabilities of the top-K tokens, with the results shown in Figure 4. It can be observed that when $K \geq 10$, the sum of probabilities exceeds 0.97, which provides sufficient knowledge for the student model to learn. Therefore, we recommend setting $K = 10$ as the default value.

Analyzing the Parameter Sizes of Teacher LLMs. We conduct the first set of experiments following the completion of black-box KD. The results, presented in Figure 7, demonstrate a trend of diminishing returns as teacher sizes increase (from 14B to 72B), indicating that larger teacher models offer limited improvements to the student model. This finding suggests that teacher models should not be excessively large to minimize computational costs. The second set of experiments is conducted on model checkpoints without black-box KD, with results shown in Figure 5. We observe that as the dataset size increases, the improvement also gradually diminishes, indicating a diminishing return on additional data. However, notable improvements are observed with larger teacher models when the dataset comprises between 10K to 100K samples, suggesting that it can be more beneficial within the specific range.

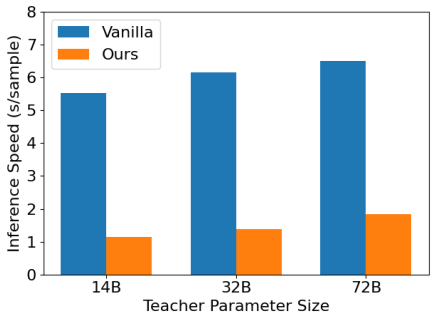


Figure 3: Comparison of the inference speed for logits generation between our approach and the vanilla approach (average seconds per sample).

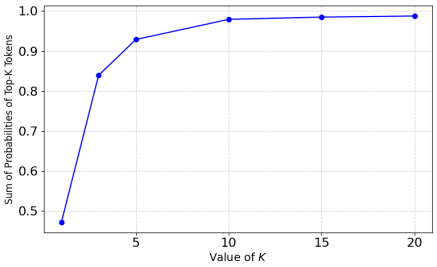
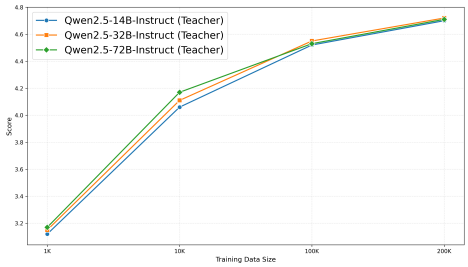


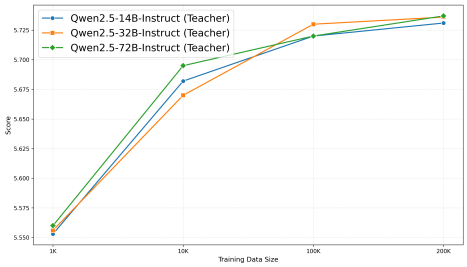
Figure 4: Sum of probabilities of top-K tokens.

4.5 Fine-grained Model Capacity Analysis

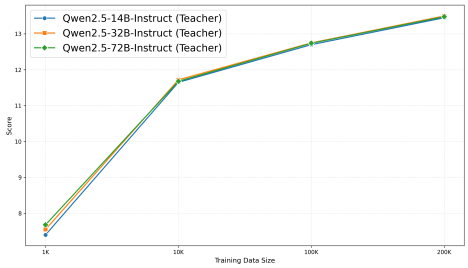
In this section, we provide a detailed capacity analysis of the *DistilQwen2.5* models, leveraging the MT-bench benchmark (Bai et al., 2024) to quantify their performance across a diverse array of NLP tasks. Due to space limitations, we show the results for two smallest models, with other models exhibiting similar trends. These results are detailed in Table 2. Our analysis not only showcases the broad



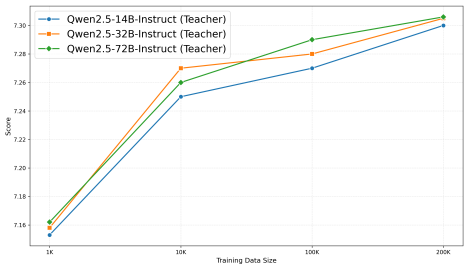
(a) Student size: 0.5B



(b) Student size: 0.5B



(c) Student size: 1.5B



(d) Student size: 1.5B

Figure 5: Performance of white-box KD with varying teacher/student model sizes and dataset sizes.

Task Type	0.5B	0.5B*	1.5B	1.5B*
Writing	6.08	**6.68**	**8.38**	**8.38**
Roleplay	7.07	**7.43**	7.26	**8.13**
Reasoning	4	**4.2**	3.9	**4.8**
Mathematics	**4.65**	**4.65**	6.85	**6.98**
Coding	4	**4.08**	4.6	**5.04**
Extraction	3.55	**4.5**	6.4	**6.6**
STEM	6.55	**6.83**	**9.65**	9.28
Humanity	**8.1**	7.95	9.73	**9.83**

Table 2: Detailed task-specific score comparisons between the original *Qwen2.5* and *DistilQwen2.5* models (0.5B and 1.5B, marked as *) on MT-bench.

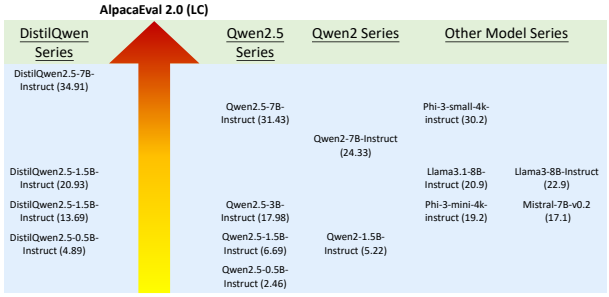


Figure 6: Comparison between various small models (<10B) based on AlpacaEval 2.0 (length-controlled).

applicability of our *DistilQwen2.5* models but also proves their enhanced capabilities and performance improvements over the original models.

4.6 Comparison Against Other Small Models

To compare the performance against other models, we present the ranking in Figure 6. Notably, the *DistilQwen2.5* series demonstrates remarkable cost-effectiveness, achieving performance that closely rivals models with parameter sizes either approaching or exceeding double its own.

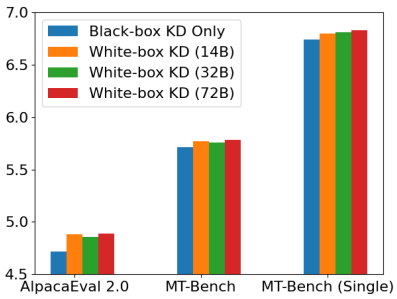
5 Industrial Use Cases

In addition to the *DistilQwen2.5* models presented, we outline two industrial use cases that illustrate the practical utility of our framework and models.

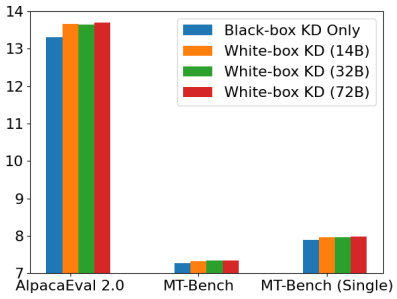
5.1 SQL Completion for Big Data Platform

In addition to instruction following, our framework can also address other tasks, such as code completion, which is also an auto-regressive task for LLMs. Here, we present a real-world application w.r.t. SQL completion. It helps users to formulate complex queries, optimize SQL statements, add conditions, or join tables based on existing queries. This technique significantly improves both the efficiency and accuracy of query composition and is widely utilized in our online big data platforms.

In the context of SQL completion for our big data platform, the primary evaluation metrics are *Latency*, *Pass@1* and *Adoption Rate*. *Latency* measures the system's speed in generating real-time suggestions as users input queries, whereas *Pass@1* and *Adoption Rate* reflect the utility and accuracy of the model's output based on automatic evaluation and human feedback. A key challenge is the trade-off between model scale and the performance metrics: although larger models can achieve higher adoption rates, they often result in increased infer-



(a) Student size: 0.5B



(b) Student size: 1.5B

Figure 7: Comparison between black-box KD and white-box KD with varying teacher model sizes after black-box KD, in terms of AlpacaEval 2.0 (length-controlled) and MT-Bench scores (both full and single).

ence time, which adversely affects latency. Therefore, the central optimization challenge for SQL completion in big data platforms lies in enhancing completion efficacy while maintaining a relatively compact model size.

During the initial deployment phase, we utilize the fine-tuned `Qwen2.5-7B` model for deployment, which is quantized to `int4` precision. By applying KD on a fixed dataset (i.e., an in-house SQL corpus), we obtain a `Qwen2.5-3B` model. This model achieves a significant improvement, closely matching the performance of the 7B model, while increasing the inference speed by 1.4×. The online performance of these models is shown in Table 3, where *Adoption Rate* is obtained through online A/B testing on the big data platform. Hence, our KD technique effectively balances performance and computational efficiency.

5.2 KD Functionalities on AI Platform

It should be acknowledged that our *DistilQwen2.5* models are primarily designed for general domains. For domain-specific applications, further enhancement is necessary (as in the SQL completion case). To enable business users or LLM developers to distill their own models, we have integrated the continual KD feature together with the *DistilQwen2.5* models into a cloud-native AI platform.

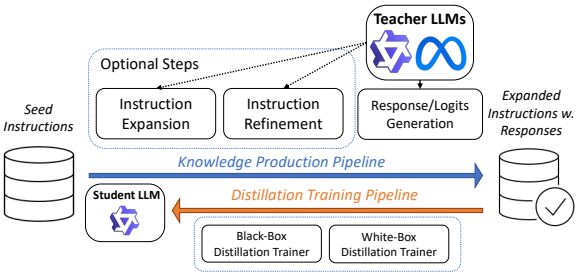


Figure 8: Illustration of continual KD pipelines on the AI platform for business users or LLM developers.

Model Size	Latency (ms)	Pass@1	Adoption Rate (%)
7B (teacher)	384	18.8	26.5
3B (student)	148	17.9	25.5

Table 3: Performance evaluation for SQL completion.

To facilitate seamless model optimization and customization, our AI platform provides robust KD functionalities, as shown in Fig. 8. It allows users to iteratively refine and tailor the *DistilQwen2.5* models to specific domains. Key pipelines include: (1) the Knowledge Production Pipeline (KPP) and (2) the Distillation Training Pipeline (DTP). In KPP, optimal steps of instruction expansion and refinement can be applied to user-provided seed instructions from arbitrary domains. The teacher LLMs are then leveraged to generate responses or output logits according to user settings. In DTP, users can define custom training settings for either black-box or white-box distillation trainers, leveraging cloud resources for scalable distillation training. After that, the student model can be utilized for evaluation and deployment.

6 Conclusion and Future Work

In this paper, we introduce *DistilQwen2.5*, a family of distilled lightweight LLMs derived from the *Qwen2.5* models. By leveraging both black-box and white-box KD techniques and efficient implementations and multiple agents, we demonstrate substantial improvements in model performance and real-world applications. For future work, we plan to investigate more diverse domain-specific applications to extend the practical impact of our framework. We also aspire to enhance the collaborative aspects of model fusion to allow for more dynamic knowledge transfer.

Limitations

While the *DistilQwen2.5* models demonstrate significant enhancements, several limitations remain that warrant further investigation. The distillation process hinges on the quality of the teacher models. Biases or errors inherent in the teacher models could propagate into the student models, potentially affecting their performance and fairness in specific contexts. Additionally, while we showcase domain-specific applications, the generalizability of our framework across diverse domains and languages remains to be thoroughly evaluated, which is beyond the scope of this work. Addressing these limitations will contribute to more robust LLMs tailored to a wider array of applications.

Ethical Considerations

Distillation techniques make it feasible to deploy LLMs in resource-constrained environments, they also introduce the potential for bias and misinformation inherited from the teacher models. Additionally, the open-sourcing of *DistilQwen2.5* models facilitates accessibility, but also raises concerns regarding misuse. Responsible use of the models requires establishing guidelines to prevent applications that may cause harm, violate privacy, or amplify malicious behavior.

References

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and Wanli Ouyang. 2024. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 7421–7454. Association for Computational Linguistics.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *CoRR*, abs/2404.04475.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*. OpenReview.net.

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

Boyu Hou, Chengyu Wang, Xiaoqing Chen, Minghui Qiu, Liang Feng, and Jun Huang. 2023. Prompt-distiller: Few-shot knowledge distillation for prompt-based language learners with dual contrastive learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–5. IEEE.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017. Association for Computational Linguistics.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4163–4174. Association for Computational Linguistics.

Ming Li, Lichang Chen, Jiuhai Chen, Shwai He, Jiuxiang Gu, and Tianyi Zhou. 2024. Selective reflection-tuning: Student-selected data recycling for LLM instruction-tuning. In *Findings of the Association for Computational Linguistics, ACL 2024*, pages 16189–16211. Association for Computational Linguistics.

Renze Lou, Kai Zhang, Jian Xie, Yuxuan Sun, Janice Ahn, Hanzi Xu, Yu Su, and Wenpeng Yin. 2024. MUFFIN: curating multi-faceted instructions for improving instruction following. In *The Twelfth International Conference on Learning Representations*. OpenReview.net.

Haojie Pan, Chengyu Wang, Minghui Qiu, Yichang Zhang, Yaliang Li, and Jun Huang. 2021. Meta-kd: A meta knowledge distillation framework for language model compression across domains. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 3026–3036. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Knowledge fusion of large language models. In *The Twelfth International Conference on Learning Representations*. OpenReview.net.

Yida Wang, Pei Ke, Yinhe Zheng, Kaili Huang, Yong Jiang, Xiaoyan Zhu, and Minlie Huang. 2020. A large-scale chinese short-text conversation dataset.

In *Natural Language Processing and Chinese Computing - 9th CCF International Conference*, volume 12430 of *Lecture Notes in Computer Science*, pages 91–103. Springer.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022*.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023. f-divergence minimization for sequence-level knowledge distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 10817–10834. Association for Computational Linguistics.

Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. 2025. Rethinking kullback-leibler divergence in knowledge distillation for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5737–5755. Association for Computational Linguistics.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *CoRR*, abs/2402.13116.

Yuanhao Yue, Chengyu Wang, Jun Huang, and Peng Wang. 2024a. Building a family of data augmentation models for low-cost LLM fine-tuning on the cloud. *CoRR*, abs/2412.04871.

Yuanhao Yue, Chengyu Wang, Jun Huang, and Peng Wang. 2024b. Distilling instruction-following abilities of large language models with task-aware curriculum planning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6030–6054. Association for Computational Linguistics.

Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. 2024. Dual-space knowledge distillation for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18164–18181. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *CoRR*, abs/2311.07911.

A Case Studies

In the following, we present several cases on the output comparison between original *Qwen2.5* and *DistilQwen2.5* models. In default, we leverage the 7B models for comparison.

Question	Problem: If the system of equations: 6x-4y=a, 6y-9x=b. has a solution (x, y) where x and y are both nonzero, find $\frac{a}{b}$, assuming b is nonzero. Solution:
Ground Truth	-$\frac{2}{3}$
Qwen2.5	To find the value of a/b, we start by solving the system of equations for x and y in terms of a and b. Given the system of equations 6x - 4y = a 6y - 9x = b, First we can simplify these equations, Equation 1 can be written as 3x - 2y = a/2 Equation 2 can be written as -3x + 6y = b, Next we add equations 3 and 4 to eliminate x, (3x - 2y) + (-3x + 6y) = a/2 + b which simplifies to 4y = a/2 + b. Solving for y, we get y = (a + 2b)/8. Next we subtract equation 4 from equation 3 to eliminate y, (3x - 2y) - (-3x + 6y) = a/2 - b, which simplifies to 6x - 8y = a/2 - b Substituting the value of y from equation 5 into this equation, we get 6x - 8((a + 2b)/8) = a/2 - b which simplifies to 6x - (a + 2b) = a/2 - b or 6x = a + 2b + a/2 - b, which further simplifies to 6x = (2a + 4b + a - 2b)/2 = (3a + 2b)/2. Solving for x we get x = (3a + 2b)/12 = (a + (2b)/3)/4, Now we need to find the ratio a/b From equation 5. We have y = (a + 2b)/8 From equation 6 we have x = (3a + 2b)/12 We substitute these expressions into the original equations. Using equation 1 6x - 4y = a we substitute x = (3a + 2b)/12 and y = (a + 2b)/8 6((3a + 2b)/12) - 4((a + 2b)/8) = a, which simplifies to (6(3a + 2b))/12 - (4(a + 2b))/8 = a or (3a + 2b)/2 - (a + 2b)/2 = a, which further simplifies to (3a + 2b - a - 2b)/2 = a or (2a)/2 = a, which is true This confirms our expressions are correct. From the simplified equations we derive the ratio a/b = 2. Thus the value of a/b is 2.
DistilQwen2.5	To solve the given system of equations and find $\frac{a}{b}$, we start with the equations: 6x - 4y = a, 6y - 9x = b. First, we express x and y in terms of a and b. We can rewrite the equations as: 6x - 4y = a, -9x + 6y = b. To eliminate one of the variables, we can multiply the first equation by 3 and the second equation by 2: 18x - 12y = 3a, -18x + 12y = 2b. Now, we add these two equations together: (18x - 12y) + (-18x + 12y) = 3a + 2b. This simplifies to: 0 = 3a + 2b. From this, we can solve for $\frac{a}{b}$ = -$\frac{2}{3}$

Table 4: Output comparison between original *Qwen2.5* and *DistilQwen2.5* models.

Question	What breed dog is smallest?
Qwen2.5	The smallest breed of dog is generally considered to be the Chihuahua. However, there are other very small breeds that are often mentioned, such as the Yorkipoo (a mix between a Yorkshire Terrier and a Poodle), Morkie (a mix between a Maltese and a Yorkshire Terrier), and Puggle (a mix between a Pug and a Beagle). But in terms of pure breeds, the Chihuahua is the smallest. Chihuahuas typically weigh between 2 and 6 pounds (0.9 to 2.7 kg) and stand about 6 to 9 inches (15 to 23 cm) tall at the shoulder.
DistilQwen2.5	The Chihuahua is generally recognized as the smallest dog breed in the world.

Table 5: Output comparison between original *Qwen2.5* and *DistilQwen2.5* models.

Question	Write a product description for a new line of dog toys, called the squeaker. It's made of latex, and is designed for dogs of all ages and species. It's also available in a variety of colors and shapes. The response must contain at least 3 placeholders represented by square brackets, such as [address], [name], and [phone number].
Qwen2.5	Introducing the [address] Pet Supply's newest addition to our toy line: the Squeaker! Designed with love and play in mind, our Squeakers are crafted from high-quality, durable latex that stands up to even the most enthusiastic chewers. Whether your [name] is a playful puppy or a seasoned senior, our Squeakers are perfect for dogs of all ages and species, providing endless hours of fun and stimulation. Available in a vibrant array of colors and shapes, our Squeakers are not just toys; they're companions for playtime. From the classic [shape1] to the whimsical [shape2], each design is carefully chosen to engage your [name]'s curiosity and keep them entertained. The unique squeaker mechanism ensures that every play session is filled with excitement and joy. Order your [name]'s new favorite toy today by visiting our website or giving us a call at [phone number]. We're here to help you find the perfect Squeaker to keep your [name] happy and healthy. Happy playing!
DistilQwen2.5	Introducing The Squeaker – the ultimate playtime companion for your furry friend! Crafted from durable, high-quality latex, The Squeaker is designed to provide endless entertainment for dogs of all ages and breeds. Available in a vibrant array of colors and fun shapes, this toy is perfect for keeping your pet engaged and happy. Each Squeaker features a built-in squeak mechanism that will capture your dog's attention and stimulate their natural instincts to play and explore. Whether your dog loves to chew, fetch, or simply cuddle with their toys, The Squeaker is sure to become their new favorite. Order yours today and watch your dog's tail wag with joy! For more information, visit our website at [website], or contact our customer service team at [phone number]. Don't miss out on the fun – make playtime special with The Squeaker![PetJoy] – Bringing joy to pets and their owners.

Table 6: Output comparison between original *Qwen2.5* and *DistilQwen2.5* models.