

# Thinking with DistilQwen: A Tale of Four Distilled Reasoning and Reward Model Series

Wenrui Cai<sup>1,2,\*</sup>, Chengyu Wang<sup>2,†</sup>, Junbing Yan<sup>2</sup>, Jun Huang<sup>2</sup>, Xiangzhong Fang<sup>1</sup>

<sup>1</sup> Shanghai Jiao Tong University, Shanghai, China

<sup>2</sup> Alibaba Cloud Computing, Hangzhou, China

{cwrcwr, xzfang}@sjtu.edu.cn

{chengyu.wcy, yanjunbing.yjb, huangjun.hj}@alibaba-inc.com

## Abstract

Recently, the demand for small and efficient reasoning models to support real-world applications has driven the development of knowledge distillation techniques that balance reasoning performance and inference speed. In this paper, we further extend the DistilQwen model family, initialized from the Qwen models, by introducing four model series specifically designed to meet industrial requirements. The distilled model collection comprises: (1) slow-thinking models, optimized for reasoning tasks that require high accuracy; (2) two series of adaptive-thinking models, which dynamically adjust reasoning strategies based on input tasks to maximize efficiency across diverse scenarios; and (3) distilled reward models, which enable further reinforcement learning of reasoning models using distilled knowledge. Comprehensive evaluations across multiple benchmarks demonstrate both high inference efficiency and strong reasoning performance for these models, as well as the practical utility of distilled reward models. We further show that these models support industry practitioners by providing scalable training and inference functionalities on the Alibaba Cloud PAI (Platform for Artificial Intelligence) platform.<sup>1</sup>

## 1 Introduction

In the rapidly evolving landscape of large language models (LLMs), the need for efficient reasoning models that can seamlessly integrate into real-world applications has become increasingly urgent. Industries worldwide increasingly rely on advanced LLMs, which require not only high reasoning performance but also fast inference speeds to support timely decision-making (Vrdoljak et al.,

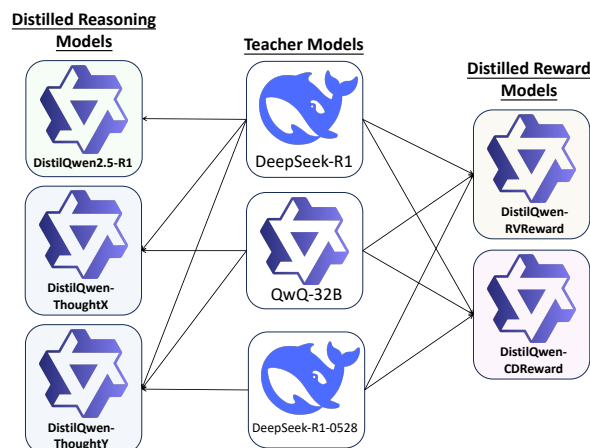


Figure 1: Roadmap for training DistilQwen reasoning and reward models.

2025; Zhong et al., 2025; Cha et al., 2025). These dual requirements have fueled growing interest in knowledge distillation (KD) methods, which aim to balance accuracy with computational efficiency (Xu et al., 2024).

In response to these industrial needs, we present a comprehensive extension of the DistilQwen model family (Wang et al., 2025b), introducing four model series tailored for a wide array of reasoning scenarios:

- **Slow-thinking models:** Optimized for tasks where accuracy is paramount, these models ensure consistently high precision.
- **Adaptive-thinking models (two series):** Motivated by recent insights that reasoning length and depth should vary by task (Sui et al., 2025; Cai et al., 2025b), these models dynamically adjust their reasoning strategies to specific requirements and outperform previous ones.
- **Adaptive-thinking-based reward models:** Derived from our training strategies for above series, these models support further reinforcement learning using distilled knowledge.

\*The work was conducted during the internship at Alibaba Cloud Computing.

<sup>†</sup> Corresponding author.

<sup>1</sup>Resources are released in the EasyDistill toolkit (Wang et al., 2025a). URL: <https://github.com/modelscope/easydistill>

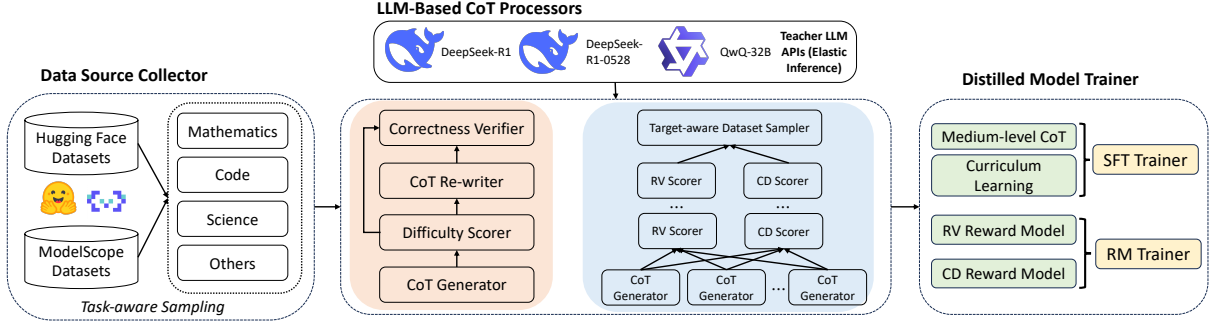


Figure 2: High-level process for obtaining DistilQwen reasoning and reward models.

The development roadmap for these model series is shown in Figure 1. DistilQwen reasoning and reward models are evaluated on diverse benchmarks, demonstrating robust performance. Furthermore, practicality is demonstrated by their integration into industrial AI platforms for fine-tuning and on-line deployment.

## 2 Related Work

Knowledge distillation (KD), first introduced by Hinton et al. (2015), has been key to reducing parameter counts in language models. Before LLMs, several studies distilled BERT-based models (Sanh et al., 2019; Jiao et al., 2020; Sun et al., 2020; Pan et al., 2021), mainly for natural language understanding. Distillation for LLMs poses additional challenges due to complex token dependencies.

In the literature, MiniLLM (Gu et al., 2024) uses a reverse Kullback-Leibler divergence (KLD) objective to transfer knowledge from white-box LLMs to student models. An adaptive strategy (Wu et al., 2025) combines forward and reverse KLD via dynamic weighting.  $f$ -Distill (Wen et al., 2023) minimizes a generalized  $f$ -divergence function at the sequence level. Li et al. (2025) propose a bi-directional logits difference loss to improve KD performance. For black-box KD (where only APIs are available), knowledge distillation signals like output logits are absent; researchers leverage data augmentation with instruction-response pairs (Hsieh et al., 2023; Li et al., 2024; Lou et al., 2024; Ranaldi and Freitas, 2024; Kim et al., 2024; Yue et al., 2024; Ma et al., 2025).

With the rise of large reasoning models such as DeepSeek-R1 (DeepSeek-AI, 2025), distilling Chain-of-Thought (CoT) reasoning ability has attracted attention (Wang et al., 2025c). Chen et al. (2025) study the effect of granularity, format, and teacher choice on CoT distillation. Cai et al.

(2025b) improve reasoning in small models by considering cognitive gaps. Multi-teacher KD (Tian et al., 2025) uses teacher forcing to transmit diverse reasoning skills. Self-training methods (Zhang et al., 2025) activate latent reasoning capacity in small models.

In our work, we open-source a range of small models for challenging reasoning tasks to facilitate further research and industrial applications.

## 3 Algorithm Implementation

We elaborate on our industrial practice for training the DistilQwen reasoning and reward models. The overall pipeline is illustrated in Figure 2.

### 3.1 Data Source Collector

The foundation of our approach is the Data Source Collector, which aggregates CoT datasets from platforms such as Hugging Face<sup>2</sup> and ModelScope<sup>3</sup>. These datasets span domains including mathematics, code, science, and more, providing rich and diverse sources for training reasoning models such as OpenThoughts<sup>4</sup>, DeepMath-103K (He et al., 2025), OpenCodeReasoning<sup>5</sup>, etc. We subsequently perform task-aware re-sampling to balance the distributions across different types of tasks.

### 3.2 LLM-Based CoT Processors

Directly performing vanilla SFT training on raw CoT datasets does not necessarily yield strong student models. Below, we describe our LLM-based CoT processors, which effectively and efficiently generate and refine CoT datasets for knowledge distillation (KD).

<sup>2</sup><https://huggingface.co/datasets>

<sup>3</sup><https://modelscope.cn/datasets>

<sup>4</sup><https://huggingface.co/datasets/open-thoughts/OpenThoughts2-1M>

<sup>5</sup><https://huggingface.co/datasets/nvidia/OpenCodeReasoning>

### 3.2.1 Elastic Teacher LLM Inference

In our implementation, directly invoking third-party APIs for teacher LLM inference is not feasible due to the requirements for robust, scalable, and elastic inference. Instead, we deploy inference services for DeepSeek-R1 (DeepSeek-AI, 2025), DeepSeek-R1-0528<sup>6</sup>, and QwQ-32B<sup>7</sup> on our computing clusters, where each server is equipped with eight NVIDIA H20 GPUs (96GB each). Because the queries per second (QPS) requirements for these models vary across subsequent steps, the number of inference nodes per model can be elastically adjusted to maximize computational resource utilization. These inference APIs form the foundation of our system; thus, we do not need to manage hardware for CoT generation and processing later.

### 3.2.2 Slow-Thinking CoT Processor

The slow-thinking CoT processor is employed to optimize CoT training sets for **slow-thinking models**, i.e., the **DistilQwen2.5-R1** series (using DeepSeek-R1 (DeepSeek-AI, 2025) as the teacher model and the Qwen2.5 series as students).

**CoT Generator.** The core module is the CoT Generator, which leverages DeepSeek-R1 to generate structured CoTs by exploring complex solution spaces. However, the output reflects how this ultra-large model (with 671B parameters) solves problems, which may not be entirely suitable for smaller models to learn from. To create diverse solution paths, multiple inference outcomes are generated using varying temperatures.

**CoT Difficulty Scorer.** A key challenge in improving KD effectiveness for smaller student models is addressing the capacity gap between teacher and student models. Several concurrent works (Cai et al., 2025b; Yu et al., 2025) propose assessing and rewriting CoTs to better suit student learning; however, these methods often require iterative processing and incur high computational costs. In our approach, the CoT Difficulty Scorer evaluates the complexity of each generated CoT using the same teacher LLM (DeepSeek-R1). By assigning difficulty levels (easy, medium, hard), the scorer distinguishes intricate CoTs from simpler or excessively challenging ones, enabling students to prioritize learning on medium-level CoTs. This systematic scoring helps models develop a deeper understanding of complex reasoning scenarios.

<sup>6</sup><https://huggingface.co/deepseek-ai/DeepSeek-R1-0528>

<sup>7</sup><https://huggingface.co/Qwen/QwQ-32B>

**CoT Re-writer and Verifier.** As reported by Cai et al. (2025b), rewritten and verified versions of CoT datasets often enable smaller LLMs to achieve stronger reasoning abilities. However, the complexity of these processing steps limits parallelization over millions of CoTs in our dataset. Therefore, we adopt a simple yet effective strategy by rewriting and verifying CoTs only at the easy and hard difficulty levels in a one-pass process. Incorrect CoTs are discarded based on the verifier, which serves as a safeguard to preserve reasoning correctness. Since multiple CoTs are generated per problem, in most cases, at least one suitable CoT is obtained. Overall, integrating these steps results in large, higher-quality CoT training sets that constitute the foundation for training our **slow-thinking models**.

### 3.2.3 Adaptive-Thinking CoT Processor

Beyond the slow-thinking processor, the adaptive-thinking CoT processor further optimizes CoT training sets for **adaptive-thinking models**, namely the **DistilQwen-ThoughtX** series (using DeepSeek-R1 and QwQ-32B as teacher models and the Qwen2.5 series as students), and its updated version following the release of DeepSeek-R1-0528: the **DistilQwen-ThoughtY** series (using DeepSeek-R1, DeepSeek-R1-0528, and QwQ-32B as teacher models and the Qwen3 series<sup>8</sup> as students). These models dynamically adjust the lengths of CoTs according to problem complexity, thereby further improving reasoning abilities and avoiding “over-thinking.”

**CoT Generator.** The implementation of the CoT Generator here is largely similar to that of the slow-thinking processor. The key difference is that we generate multiple CoTs per problem using different teacher models. Additionally, the inference temperatures are varied to increase diversity.

**RV and CD Scorers.** During the upgrade of DistilQwen reasoning models, we discovered that rewriting alone is insufficient, as it does not address situations where models tend to “over-think” simpler problems (Sui et al., 2025). Our Reasoning Verbosity (RV) and Cognitive Difficulty (CD) Scorers, derived from the work in (Cai et al., 2025a), further assess the quality of CoTs by ensuring they are appropriately verbose for challenging problems and match the cognitive capacity of the student models. This dual scoring mechanism ensures that models are exposed to a broad spectrum of CoT

<sup>8</sup><https://qwenlm.github.io/blog/qwen3/>

processes that are better aligned with their capabilities and the problem difficulty. Consequently, the models can learn to adaptively think based on the input problems, leading to higher accuracy and faster inference.

**Target-aware Dataset Sampler.** Finally, given a target student model, we sample an optimal subset of CoTs for training. Note that our CoT-based system is not static; as more powerful LLMs become publicly available, we can continually collect higher-quality CoTs to train stronger small models. This is further demonstrated by the significant improvement of **DistilQwen-ThoughtY** over **DistilQwen-ThoughtX** shown in our experiments.

### 3.3 Distilled Model Trainer

**SFT Trainer with Curriculum Learning.** To effectively train our **slow-thinking** and **adaptive-thinking** models, we adopt supervised fine-tuning (SFT) enhanced with curriculum learning principles (Soviany et al., 2022). The training pipeline begins with medium-level CoT examples to ensure stable convergence and prevent overfitting. As training progresses, the curriculum gradually incorporates more challenging samples, promoting generalization and robustness across diverse scenarios. Please refer to the experimental results for additional details.

**RM Trainer with RV and CD Score Estimation.** Beyond evaluating CoT quality, RV and CD scores are also leveraged to train lightweight models as reward predictors, which can subsequently enhance the model’s reasoning ability via reinforcement learning (RL). To this end, our reward model (RM) trainer produces two dedicated reward models, each initialized from Qwen2.5-7B-Instruct: one for RV and one for CD. This constitutes a special case of knowledge distillation (KD), as the prediction outcomes are derived from very large teacher models. This approach circumvents the need to directly employ the original teacher models as RMs during the more resource-demanding RL training process.

Although RL training is not the primary focus of this work, we briefly describe how **DistilQwen-Reward** models are integrated into RL training. We use Group Relative Policy Optimization (GRPO) (DeepSeek-AI, 2025) as a representative RL algorithm. Let  $f_{RV}(x)$  and  $f_{CD}(x)$  denote the predicted RV and CD scores by our models, given input  $x$ . Note that the predicted scores are normalized to  $[0, 1]$ . The corresponding RV and CD

rewards are defined as follows:

$$R_{RV}(x) = -|f_{RV}(x) - \text{Clip}(f_{RV}(x), L_{RV}, H_{RV})| \quad (1)$$

$$R_{CD}(x) = -|f_{CD}(x) - \text{Clip}(f_{CD}(x), L_{CD}, H_{CD})| \quad (2)$$

where  $(L_{RV}, H_{RV})$  and  $(L_{CD}, H_{CD})$  are the respective score intervals for the output CoTs. If a score lies within its designated interval, the penalty term is zero; outside the interval, the penalty increases linearly with the distance to the nearest boundary. Combined with the conventional *accuracy* and *format* rewards (denoted as  $R_{fnt}$  and  $R_{acc}$ , respectively) used in standard GRPO, the revised overall reward function is:

$$R = R_{fnt} + R_{acc} + \lambda_{RV} R_{RV}(x) + \lambda_{CD} R_{CD}(x) \quad (3)$$

where  $\lambda_{RV}$  and  $\lambda_{CD}$  are tunable hyperparameters that determine the weighting. In our experiments, we further validate the effectiveness of the **DistilQwen-Reward** models relative to vanilla GRPO. We plan to continue developing RL-enhanced lightweight reasoning models in future work, which may extend beyond the scope of the KD techniques presented in this paper.

## 4 Evaluation

In this section, we present the evaluation results for all DistilQwen reasoning and reward models.

### 4.1 Evaluation Benchmarks

To comprehensively evaluate the reasoning capabilities of our models, we conduct experiments on challenging benchmarks spanning mathematics, programming, and question answering. Among these, AIME2024<sup>9</sup> features problems that require multi-step reasoning and intricate mathematical understanding. MATH500 (Hendrycks et al., 2021) comprises 500 difficult mathematical problems across various domains, including algebra, geometry, number theory, and calculus. GPQA Diamond (Rein et al., 2023) is a complex question-answering benchmark designed to assess general-purpose reasoning and comprehension abilities. Finally, LiveCodeBench V2 (Jain et al., 2025) evaluates models’ programming and code generation skills, comprising coding tasks that range from algorithmic challenges to practical problems.

<sup>9</sup>[https://artofproblemsolving.com/wiki/index.php/2024\\_AIME\\_I](https://artofproblemsolving.com/wiki/index.php/2024_AIME_I)



Model	Training Set Size	AIME2024	MATH-500	GPQA Diamond	LiveCodeBench V2
Qwen2.5-3B-Instruct	-	6.7	62.6	32.8	11.3
<b>DistilQwen2.5-3B-R1</b>	105K	<b>16.7</b>	<b>70.0</b>	<b>34.3</b>	<b>18.0</b>
Qwen2.5-7B-Instruct	-	10.0	73.6	33.3	30.7
OpenThinker-7B	114K	31.3	83.0	42.4	39.9
<b>DistilQwen2.5-7B-R1</b>	105K	<b>43.3</b>	<b>88.4</b>	<b>42.9</b>	<b>46.4</b>
Qwen2.5-14B-Instruct	-	16.7	78.2	43.4	37.4
<b>DistilQwen2.5-14B-R1</b>	105K	<b>46.7</b>	<b>90.8</b>	<b>51.5</b>	<b>54.4</b>
Qwen2.5-32B-Instruct	-	16.7	81.4	45.5	47.3
OpenThinker-32B	114K	66.0	90.6	61.6	<b>68.9</b>
<b>DistilQwen2.5-32B-R1</b>	105K	<b>70.0</b>	<b>93.8</b>	<b>62.1</b>	66.0

Table 1: Performance comparison among **slow-thinking models** in terms of deep reasoning abilities. Note that **DistilQwen2.5-R1** and OpenThinker models leverage the same set of source reasoning problems (with a few filtered out by our verifier) and the same teacher model for training.

Model	AIME2024	MATH500	GPQA Diamond	LiveCodeBench V2
<i>Experiments on DistilQwen-ThoughtX</i>				
Qwen2.5-7B-Instruct	10.0	73.6	33.3	30.7
<b>DistilQwen-ThoughtX-7B</b>	<b>56.7</b>	<b>90.2</b>	<b>50.0</b>	<b>56.8</b>
Qwen2.5-32B-Instruct	16.67	81.4	45.5	47.3
<b>DistilQwen-ThoughtX-32B</b>	<b>80.0</b>	<b>92.6</b>	<b>64.0</b>	<b>73.4</b>
<i>Experiments on DistilQwen-ThoughtY</i>				
Qwen3-4B (thinking mode)	73.3	93.2	54.0	75.7
<b>DistilQwen-ThoughtY-4B</b>	<b>76.7</b>	<b>95.2</b>	<b>56.1</b>	<b>75.8</b>
Qwen3-8B (thinking mode)	<b>76.7</b>	94.0	62.0	62.8
<b>DistilQwen-ThoughtY-8B</b>	<b>76.7</b>	<b>94.6</b>	<b>62.1</b>	<b>78.1</b>
Qwen3-32B (thinking mode)	76.7	94.8	<b>65.7</b>	72.2
<b>DistilQwen-ThoughtY-32B</b>	<b>90.0</b>	<b>95.2</b>	63.6	<b>76.3</b>

Table 2: Performance of **adaptive-thinking models**. Note that the Qwen2.5 models are non-reasoning models, while the Qwen3 models can act as reasoning models with their thinking modes enabled.

## 4.2 Evaluation of Slow-Thinking Models

**Slow-thinking models**, namely the **DistilQwen2.5-R1** series, encompass model scales of 3B, 7B, 14B, and 32B parameters. The source reasoning problems are taken from OpenThoughts<sup>10</sup>, with CoTs generated, rewritten, and verified using DeepSeek-R1. We perform SFT training on medium-level CoTs for three epochs, followed by additional training on harder examples. After SFT, direct preference optimization (DPO) (Rafailov et al., 2023) is applied, yielding modest further improvements, though this is not our main focus here.

As presented in Table 1, our proposed approach markedly enhances the reasoning abilities of existing LLMs, delivering consistent and substantial gains across multiple benchmarks compared to the original Qwen2.5 models and the OpenThinker models<sup>11</sup>, which are trained on the same set of reasoning problems with the same teacher model.

Additionally, we evaluate **DistilQwen2.5-R1** models using inference-time scaling, where the

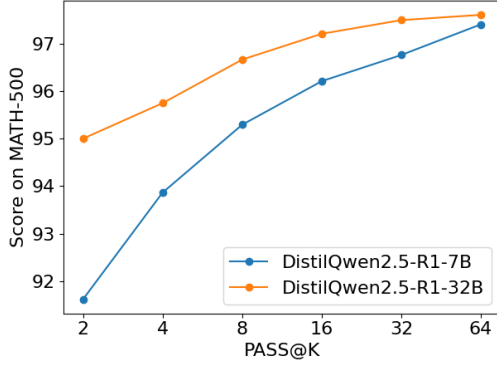
models generate  $k$  answers for the same question and are measured using the  $Pass@K$  metric. The findings indicate that increasing the number of reasoning attempts  $K$  leads to significant accuracy improvements for both models. Notably, the 7B model exhibits a steep upward trend on MATH500 and GPQA Diamond, gradually approaching the performance level of the 32B model while reducing inference computation requirements.

## 4.3 Evaluation of Adaptive-Thinking Models

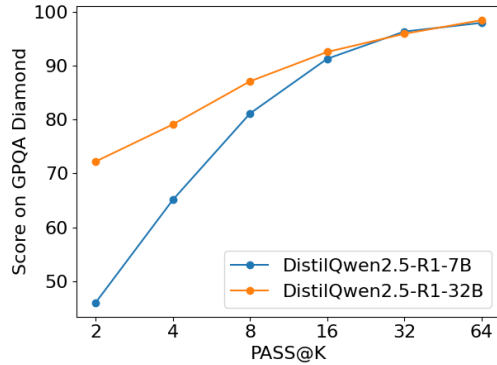
In Table 2, we present the evaluation results for our **adaptive-thinking models**. The **DistilQwen-ThoughtX** series, built upon Qwen2.5 models and trained with the dataset from (Cai et al., 2025a), encompasses model scales of 7B and 32B. The results demonstrate substantial improvements at all scales, surpassing the **slow-thinking models**. The **DistilQwen-ThoughtY** series, initialized from Qwen3 models with reasoning modes enabled, achieves further advances. These models span three scales (4B, 8B, and 32B) and are trained on the previous series’ dataset and a subset of 365K CoTs generated from DeepSeek-R1-0528 (to be released). The experiments confirm that

<sup>10</sup><https://huggingface.co/datasets/open-thoughts/OpenThoughts-114k>

<sup>11</sup><https://huggingface.co/open-thoughts/OpenThinker-7B>



(a) Task: MATH500.



(b) Task: GPQA Diamond.

Figure 3: Performance of **DistilQwen2.5-R1** models in terms of  $Pass@K$  under multiple inference attempts.

Model	GSM8K	MATH500	AIME2024
7B-R1	1223.61	6586.36	12856.23
ThoughtX-7B	834.36	6031.11	14597.96
ThoughtY-8B	844.23	5932.95	15632.85
32B-R1	1178.92	6434.50	13583.19
ThoughtX-32B	742.04	5927.32	16387.53
ThoughtY-32B	723.18	5723.08	17231.84

Table 3: Analysis of averaged output CoT lengths of **adaptive-thinking models**. “DistilQwen” and “DistilQwen2.5” prefixes are omitted from model names.

our adaptive-thinking CoT processor and training strategies effectively enhance reasoning capabilities across model scales and tasks.

In addition, Table 3 analyzes the output CoT lengths according to the difficulty of reasoning problems. The results clearly demonstrate that our **adaptive-thinking models** generate more optimal CoTs based on their understanding of the input. For example, the **adaptive-thinking models** produce shorter CoTs for simpler problems in GSM8K<sup>12</sup>, and generate longer CoTs for more challenging problems in MATH500 and AIME2024.

<sup>12</sup><https://huggingface.co/datasets/openai/gsm8k>

**Further Discussion.** The reason why the results obtained with adaptive-thinking models are significantly better than those with slow-thinking models is that adaptive-thinking models utilize substantially more training data. Adaptive-thinking models sample from over 2 million CoTs, resulting in training sets of at least 500K, while slow-thinking models train on only approximately 100K data points.

Although the slow-thinking recipe can effectively enhance a model’s reasoning capabilities, it is unsuitable for customized, diverse training requirements due to its lack of quantitative CoT evaluation information. Therefore, we proposed the adaptive-thinking recipe, and we did not further increase the training data for slow-thinking models.

However, this does not imply that the slow-thinking recipe is obsolete. In scenarios where training data is inherently limited, the sampling process of the adaptive-thinking recipe would further reduce available training data. In such cases, we recommend using the slow-thinking recipe to ensure satisfactory training results.

#### 4.4 Evaluation of Reward Models

Table 4 presents comparison results between the standard GRPO algorithm and GRPO augmented with our RV- and/or CD-based **distilled reward models**. We use Qwen2.5-7B-Instruct as the base model and randomly sample 10K mathematical problems from the previous training set for RL training. Specifically, we conduct RL training on the model directly, with no CoT-based SFT on mathematical problems, ensuring that models learn mathematical reasoning purely via RL. As shown, GRPO enhanced with RV/CD-based reward models consistently outperforms vanilla GRPO, corroborating our hypothesis that RV/CD scores distilled from teacher models can effectively benefit RL training, which is a promising direction for future work.<sup>13</sup>

## 5 Industrial Solutions

In addition to releasing all our DistilQwen reasoning models to the open-source community for use in various local environments, we have integrated these models into the Alibaba Cloud PAI (Platform for Artificial Intelligence) platform that supports the entire lifecycle of LLM usage, including

<sup>13</sup>In SFT training, we utilize over 500K training samples, whereas in RL training, we use only 10K samples. Consequently, the results in Table 4 are lower than those in Tables 1 and Table 2.

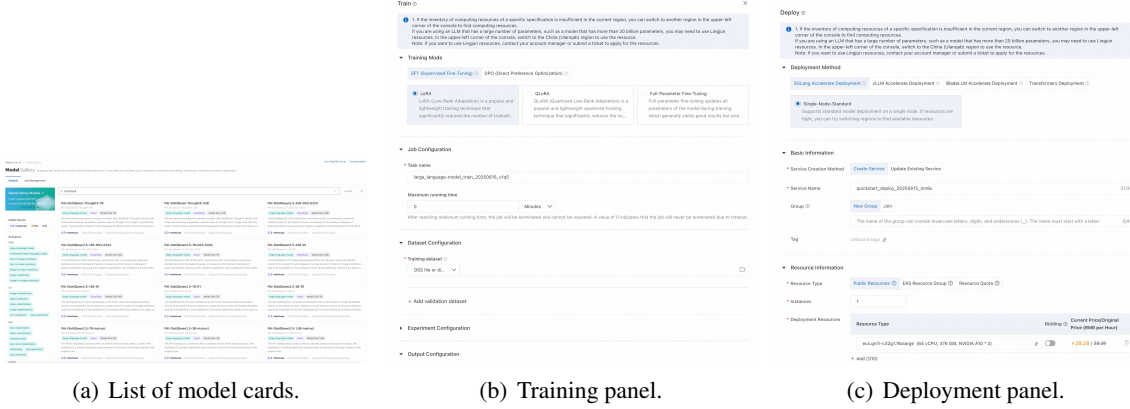


Figure 4: Snapshots of the integration of DistilQwen reasoning models with the AI platform.

Reward Setting	MATH500	AIME2024
Qwen2.5-7B-Instruct (Raw)	73.6	10.0
Vanilla GRPO	78.8	13.3
GRPO+RV	79.0	13.3
GRPO+CD	80.8	16.7
GRPO+RV+CD	<b>81.4</b>	<b>20.0</b>

Table 4: Performance of **distilled reward models**. RL performance is measured on a subset of mathematical problems using various reward settings. None of the models have undergone CoT-based SFT for mathematical reasoning tasks.

training, evaluation, compression, and deployment. Snapshots of model cards as well as the training and deployment panels are shown in Figure 4.

By embedding the DistilQwen reasoning models into AI platforms, businesses can leverage and further adapt these specialized capabilities to support real-world applications, including but not limited to decision making, code generation, problem solving, and multi-agent systems. The platform’s deployment functionalities enable seamless integration of these models into existing systems via RESTful APIs compatible with the OpenAI format, thereby facilitating easier and more efficient usage.

## 6 Conclusion and Future Work

In this work, we have expanded the DistilQwen model collection by introducing four model series tailored to address diverse reasoning requirements. Our **slow-thinking models** prioritize accuracy for demanding tasks, while the **adaptive-thinking models** dynamically optimize reasoning strategies to balance efficiency and performance. In addition, the **distilled reward models** facilitate further enhancement through reinforcement learning based on distilled knowledge. Extensive eval-

uations demonstrate that our models achieve a favorable trade-off between inference efficiency and reasoning capability. Furthermore, by integrating these models into a scalable AI platform, we provide practical tools that effectively support industry practitioners in model training and deployment.

## Limitations

Although the proposed DistilQwen reasoning models perform well across several benchmarks, their effectiveness may vary in highly specific or dynamic real-world contexts where benchmarks do not fully capture operational complexity. Furthermore, the reinforcement learning framework supported by the distilled reward models depends on the quality of the distilled knowledge, which may propagate biases or errors inherent in the teacher models. We suggest that future research focus on addressing these limitations and further improving model adaptability and precision in real-world applications.

## Ethical Considerations

The development and deployment of the DistilQwen reasoning models requires careful consideration of ethical implications. It is essential to address potential biases introduced by the training data, as such biases may influence reasoning outcomes. Since our models are intended for research and industrial use, issues of data privacy, security, and compliance with relevant regulations should be rigorously addressed during practical implementation.

## Acknowledgments

This work was supported by Alibaba Research Intern Program.

## References

- Wenrui Cai, Chengyu Wang, Junbing Yan, Jun Huang, and Xiangzhong Fang. 2025a. [Reasoning with omnithought: A large cot dataset with verbosity and cognitive difficulty annotations](#). *CoRR*, abs/2505.10937.
- Wenrui Cai, Chengyu Wang, Junbing Yan, Jun Huang, and Xiangzhong Fang. 2025b. [Training small reasoning llms with cognitive preference alignment](#). *CoRR*, abs/2504.09802.
- Seungeon Cha, Jinseok Park, Hojin Choi, Hokyoung Ryu, and Kyoungwon Seo. 2025. [CLONE: synthetic guideline-based clinical reasoning with large language models for early diagnosis of mild cognitive impairment](#). In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 122:1–122:14. ACM.
- Xinghao Chen, Zhijing Sun, Wenjin Guo, Miaoran Zhang, Yanjun Chen, Yirong Sun, Hui Su, Yijie Pan, Dietrich Klakow, Wenjie Li, and Xiaoyu Shen. 2025. [Unveiling the key factors for distilling chain-of-thought reasoning](#). *CoRR*, abs/2502.18001.
- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *CoRR*, abs/2501.12948.
- 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.
- Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. [Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning](#). *CoRR*, abs/2504.11456.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the MATH dataset](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.
- Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. [Distilling the knowledge in a neural network](#). *CoRR*, abs/1503.02531.
- 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.
- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2025. [Livecodebench: Holistic and contamination free evaluation of large language models for code](#). In *The Thirteenth International Conference on Learning Representations*. OpenReview.net.
- 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.
- Minju Kim, Haein Jung, and Myoung-Wan Koo. 2024. [SELF-EXPERTISE: knowledge-based instruction dataset augmentation for a legal expert language model](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1098–1112. Association for Computational Linguistics.
- Minchong Li, Feng Zhou, and Xiaohui Song. 2025. [Bild: Bi-directional logits difference loss for large language model distillation](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1168–1182. 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, Hanzhi 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.
- Weijian Ma, Ruoxin Chen, Ke-Yue Zhang, Shuang Wu, and Shouhong Ding. 2025. [Instruct where the model fails: Generative data augmentation via guided self-contrastive fine-tuning](#). In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence*, pages 5991–5999. AAAI Press.
- 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.



- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*.
- Leonardo Ranaldi and André Freitas. 2024. [Self-refine instruction-tuning for aligning reasoning in language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2325–2347. Association for Computational Linguistics.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Di-rani, Julian Michael, and Samuel R. Bowman. 2023. [GPQA: A graduate-level google-proof q&a benchmark](#). *CoRR*, abs/2311.12022.
- 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.
- Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. [Curriculum learning: A survey](#). *Int. J. Comput. Vis.*, 130(6):1526–1565.
- Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and Xia Ben Hu. 2025. [Stop overthinking: A survey on efficient reasoning for large language models](#). *CoRR*, abs/2503.16419.
- 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.
- Yijun Tian, Yikun Han, Xiushi Chen, Wei Wang, and Nitesh V. Chawla. 2025. [Beyond answers: Transferring reasoning capabilities to smaller llms using multi-teacher knowledge distillation](#). In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 251–260. ACM.
- Josip Vrdoljak, Zvonimir Boban, Ivan Males, Roko Skrabic, Marko Kumric, Anna Ottosen, Alexander Clemenceau, Josko Bozic, and Sebastian Völker. 2025. [Evaluating large language and large reasoning models as decision support tools in emergency internal medicine](#). *Comput. Biol. Medicine*, 192:110351.
- Chengyu Wang, Junbing Yan, Wenrui Cai, Yuanhao Yue, and Jun Huang. 2025a. [Easydistill: A comprehensive toolkit for effective knowledge distillation of large language models](#). *CoRR*, abs/2505.20888.
- Chengyu Wang, Junbing Yan, Yuanhao Yue, and Jun Huang. 2025b. [Distilqwen2.5: Industrial practices of training distilled open lightweight language models](#). *CoRR*, abs/2504.15027.
- Chengyu Wang, Taolin Zhang, Richang Hong, and Jun Huang. 2025c. [A short survey on small reasoning models: Training, inference, applications and research directions](#). *CoRR*, abs/2504.09100.
- 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.
- Qianjin Yu, Keyu Wu, Zihan Chen, Chushu Zhang, Manlin Mei, Lingjun Huang, Fang Tan, Yongsheng Du, Kunlin Liu, and Yurui Zhu. 2025. [Rethinking the generation of high-quality cot data from the perspective of llm-adaptive question difficulty grading](#). *CoRR*, abs/2504.11919.
- Yuanhao Yue, Chengyu Wang, Jun Huang, and Peng Wang. 2024. [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.
- Yong Zhang, Bingyuan Zhang, Zhitao Li, Ming Li, Ning Cheng, Minchuan Chen, Tao Wei, Jun Ma, Shaojun Wang, and Jing Xiao. 2025. [Self-enhanced reasoning training: Activating latent reasoning in small models for enhanced reasoning distillation](#). *CoRR*, abs/2502.12744.
- Wuchang Zhong, Jinglin Huang, Maoqiang Wu, Weinan Luo, and Rong Yu. 2025. [Large language model based system with causal inference and chain-of-thoughts reasoning for traffic scene risk assessment](#). *Knowl. Based Syst.*, 319:113630.