Knowledge-to-Verification: Exploring RLVR for LLMs in Knowledge-Intensive Domains

Zhonghang Yuan[1*], **Zhefan Wang**[1*], **Fang Hu**[2*], **Zihong Chen**[1], **Jinzhe Li**[1], **Gang Li**[1], **Jie Ying**[1], **Huanjun Kong**[1], **Songyang Zhang**[1], **Nanqing Dong**[1,2†]
[1]Shanghai Artificial Intelligence Laboratory
[2]Shanghai Innovation Institute

Abstract

Reinforcement learning with verifiable rewards (RLVR) has demonstrated promising potential to enhance the reasoning capabilities of large language models (LLMs) in domains such as mathematics and coding. However, its applications on knowledge-intensive domains have not been effectively explored due to the scarcity of high-quality verifiable data. Furthermore, current RLVR focuses solely on the correctness of final answers, leading to the limitations of flawed reasoning and sparse reward signals. In this work, we propose **K**nowledge-**t**o-**V**erification (**K2V**), a framework that extends RLVR to knowledge-intensive domains through automated verifiable data synthesis, while enabling verification of the LLM's reasoning process. Extensive experiments demonstrate that K2V enhances the reasoning of LLM in knowledge-intensive domains without significantly compromising the model's general capabilities. This study also suggests that integrating automated data synthesis with reasoning verification is a promising direction to enhance model capabilities in these broader domains.

1 Introduction

Recent large language models (LLMs), such as OpenAI-o1 (OpenAI et al., 2024), DeepSeek-R1 (Guo et al., 2025a), and Qwen3 (Yang et al., 2025a), have demonstrated remarkable progress in reasoning. Central to this progress is reinforcement learning with verifiable rewards (RLVR) (Shen et al., 2025; Peng et al., 2025; Stojanovski et al., 2025), which drives the model to self-explore during training by comparing its outputs against a verifiable ground truth, thereby enhancing its capacity for complex problem-solving.

However, current RLVR methods are confined to mathematical reasoning (Zeng et al., 2025; Liu et al., 2025) and coding tasks (He et al., 2025; Luo et al., 2025a; Cui et al., 2025), lacking the transferability to knowledge-intensive domains (*e.g.* agriculture, law, and medicine), which heavily rely on specialized knowledge. This narrow focus can be attributed to two main reasons: *unverifiable answers* and *limited data*. Firstly, for mathematics, the correctness of an LLM's response can be directly validated by a rule-based verifier (Hu et al., 2025), and for coding, unit tests can be directly executed on model-generated code in a sandboxed environment (Luo et al., 2025a). However, in knowledge-intensive domains, the answers are typically in the form of open-ended text, which cannot be automatically validated. Secondly, in the domains of mathematics and coding, a vast amount of verifiable data can be acquired from the internet and textbooks (Ma et al., 2025), and synthesizing such data is also relatively straightforward (Yang et al., 2025a). In contrast, data collected for knowledge-intensive domains are commonly unverifiable and low quality text. Knowledge-intensive domains also lack effective data synthesis methods, relying on costly, expert-level manual annotation (Dubois et al., 2023).

Moreover, current RLVR methods suffer from two inherent limitations. The first one is *flawed reasoning*. The traditional RLVR focus solely on the correctness of the final answer, ignoring the validity of the reasoning process. This reward mechanism can lead LLM to exhibit issues such as linguistic incoherence (Guo et al., 2025a) and unfaithful reasoning (Chen et al., 2025a). The second one is *sparse rewards*. Awarding a binary reward based only on the final answer creates an overly sparse reward signal, which increases the variance of the policy gradient estimate, introduces noise to the training, and results in unstable learning and slower convergence (Su et al., 2025).

To address these problems, we propose **K**nowledge-**t**o-**V**erification (**K2V**), which automatically synthesizes verifiable question-answering

[*]Equal contribution.
[†]Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 40715–40749
July 2-7, 2026

(QA) pairs in knowledge-intensive domains, while also enabling the verification of the LLMs' reasoning process. The motivation of K2V is that the structured knowledge is easier to verify than unstructured knowledge. To this end, we present a fill-blank style verification, which first organizes knowledge from the corpora into a knowledge graph (KG) (Hogan et al., 2022), and then transforms the conventional knowledge graph completion (KGC) task (Ji et al., 2021; Yao et al., 2025) into fill-blank style QA pairs. This enables the efficient synthesis of large-scale verifiable QA pairs. Furthermore, directly verifying the correctness of an LLM's reasoning process is challenging. However, following the principle of *Divide and Conquer* (Cormen et al., 2022), this task can be decomposed into multiple binary-verifiable subtasks. Specifically, we introduce a checklist-style verification method that generates a checklist for each QA pair. This checklist consists of multiple verifiable subtasks that describe the criteria for a correct reasoning process. Each subtask can be answered with a simple yes or no. Finally, we propose an answer-gated reward mechanism. This design ensures that the reasoning reward is awarded only when the final answer is correct, thereby anchoring the model's logical consistency to factual accuracy and preventing potential reward hacking.

To evaluate the effectiveness and robustness of K2V, extensive experiments were conducted on three representative knowledge-intensive domains: agriculture, law, and medicine. The results based on Qwen2.5 (Yang et al., 2024) and Llama3 (Grattafiori et al., 2024) backbones show that K2V can enhance the reasoning of LLMs without significantly compromising their general abilities, and in general outperforms the existing baselines that can synthesize verifiable data for knowledge-intensive domains. Ablation studies further suggest that the proposed verification and rewarding designs are simple yet effective, which might be of interest to the broad RLVR community.

Our contributions are summarized as follows:

- We present K2V, a scalable framework that explores RLVR in knowledge-intensive domains.
- We introduce fill-blank style verification, which is designed to synthesize large-scale verifiable QA pairs, and checklist-style verification, which aims to verify the model's reasoning process.
- We integrate answer and reasoning rewards through an answer-gated reward mechanism, which ensures factual correctness and prevents potential reward hacking.
- We conduct extensive experiments to demonstrate that K2V enhances model's reasoning capabilities in knowledge-intensive domains without significantly compromising general capabilities.

2 Related Work

Reinforcement Learning with Verifiable Rewards. Unlike conventional reinforcement learning from human feedback (RLHF) that relies on scalar reward models (Ouyang et al., 2022), RLVR aims to enhance the model's reasoning abilities by computing rewards for tasks that can be automatically verified (Guo et al., 2025a), *e.g.* mathematics (Zeng et al., 2025) and coding (He et al., 2025). Thus, for reasoning tasks in the knowledge-intensive domains that can not be automatically verified, existing RLVR studies can not be directly applied. Even though VeriFree (Zhou et al., 2025) attempts to use the model's internal probability distribution as a reward signal, it still rely on the availability of verifiable data. Furthermore, existing RLVR studies overlook the quality of reasoning process (Chen et al., 2025a), which may lead to flawed reasoning, *e.g.* studies such as Light-R1 (Wen et al., 2025) focus primarily on the correctness of final answers. In this study, we aim to bridge the gap between verifiable data and knowledge-intensive tasks.

Data Synthesis. There have been recent studies that leverage data synthesis to improve supervised fine-tuning (SFT) performanc. Liquid (Lee et al., 2023), Synthetic Data RL (Guo et al., 2025b), and BDS (Dedhia et al., 2025) synthesize verifiable QA pairs by extracting key information from the corpus, but fails to establish associations across different tasks. Genie (Yehudai et al., 2024) and Evol-Instruct (Xu et al., 2024) primarily focus on open-ended text, which can not be verified. Open-R1 (Hugging Face, 2025) and DeepScaleR (Luo et al., 2025b) focus on synthesizing verifiable data in mathematics or coding domains, but struggling to integrate specialized expertise from knowledge-intensive domains to generate verifiable data. We perform both quantitative and qualitative comparison between K2V and existing RLVR methods that can be applied to knowledge driven tasks in Section 4.2 and Appendix H.

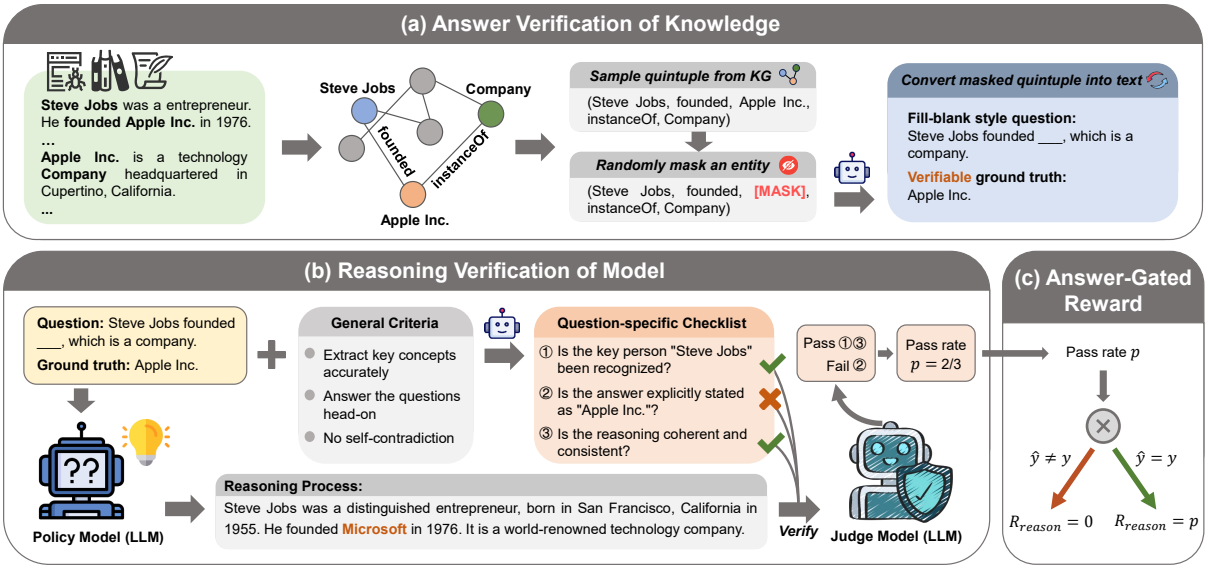

Figure 1: An overview of K2V. **(a)** K2V begins by constructing a KG from unstructured corpora. It then samples quintuples from the KG and randomly masks an entity. This masked quintuple is then converted into a fill-blank style question, where the name of the masked entity serves as the verifiable ground truth. **(b)** Given a QA pair, the Policy Model generates a reasoning process. To verify this reasoning process, K2V first creates a question-specific checklist by instantiating a set of general criteria. The Judge Model then verifies the reasoning process against each item in the checklist. The pass rate on the checklist serves as a dense reward signal. **(c)** Reasoning reward is awarded only when the predicted answer $\hat{y}$ matches the ground truth y. Otherwise, the reasoning reward is 0 regardless of how logical the reasoning process may appear.

3 Knowledge-to-Verification

In this section, we first present fill-blank style verification (Section 3.1), which enables the synthesis of verifiable QA pairs. Next, we introduce a checklist-style verification (Section 3.2), which not only validates the model's reasoning process but also provides dense reward signals. Finally, we discuss the answer-gated reward mechanism (Section 3.3), which ensures factual correctness and prevents potential reward hacking. An overview of K2V is presented in Figure 1.

3.1 Answer Verification of Knowledge

Knowledge Graph Completion. A KG is a structured representation of factual knowledge, formally defined as $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$, where $\mathcal{E}$ is the set of entities, $\mathcal{R}$ is the set of relations, and $\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ represents the set of triples. Each triple $(h, r, t) \in \mathcal{T}$ consists of a head entity h, a relation r, and a tail entity t. The goal of conventional KGC is entity prediction. Given a triple with a missing entity, formally expressed as $(h, r, ?)$ for a missing tail or $(?, r, t)$ for a missing head, the task is to predict the missing entity from the set of entities $\mathcal{E}$.

Fill-blank Style Verification. Following the principles of KGC, we reformulate conventional KGC tasks as fill-blank style QA pairs to synthesize verifiable data in knowledge-intensive domains. Meanwhile, to ensure the synthesized questions possess sufficient context for complex reasoning, K2V operates not on individual triples, but on quintuples. A quintuple σ is defined as:

$$\sigma = (e_1, r_1, e_2, r_2, e_3) \tag{1}$$

Where e_1, e_2, e_3 denote entities and r_1, r_2 denote relations.

To generate a QA pair, K2V first randomly masks one of the three entities (*i.e.*, e_1, e_2, or e_3) to create a masked quintuple σ_{masked}, e.g., $(e_1, r_1, \texttt{[MASK]}, r_2, e_3)$. K2V then employs an LLM, denoted as $\mathcal{F}_{\text{text}}$, to convert the masked quintuple into a fill-blank style question Q_{blank},

$$Q_{\text{blank}} = \mathcal{F}_{\text{text}}(\sigma_{\text{masked}}) \tag{2}$$

where the ground truth for Q_{blank} is the name of the masked entity, which can be automatically verified by a rule-based validator. The prompt of $\mathcal{F}_{\text{text}}$ is shown in Appendix I.1.

QA Pairs Synthesis. To synthesize verifiable QA pairs, K2V first utilizes the GraphGen (Chen et al., 2025b) to construct a KG. Specifically, K2V employs an LLM to perform named entity recognition

(NER) (Keraghel et al., 2024) and relation extraction (RE) (Zhao et al., 2024) on unstructured corpora (see Appendix I.2 for the prompt), then links the entities and relations to construct a KG.

After the KG is constructed, K2V directly samples quintuples from the KG. For each sampled quintuple σ (see Equation 1), K2V randomly selects an entity from $\{e_1, e_2, e_3\}$ to be masked, and then uses $\mathcal{F}_{\text{text}}$ (see Equation 2) to convert the masked quintuple σ_{masked} into a sentence. This sentence is a fill-blank style question, with the ground truth being the name of the masked entity.

K2V operates entirely from scratch, requiring no human annotation or seed datasets (Wang et al., 2023; Yu et al., 2025a). This means K2V can be scalable to large-scale unstructured corpora.

3.2 Reasoning Verification of Model

Given an input question x and a policy model π_θ parameterized by θ, a reasoning process z and a response $\hat{y}$ are sampled from the policy, denoted by $z, \hat{y} \sim \pi_\theta(\cdot|x)$. The reasoning process z is typically a lengthy, open-ended text. Due to the lack of evaluation criteria, directly verifying the correctness of z is challenging. Motivated by this, we propose checklist-style verification.

Checklist-Style Verification. For each question x, we establish a checklist, which is formally represented as a set of k verifiable criteria:

$$C^x = \{c_1, c_2, \ldots, c_k\} \tag{3}$$

where each criterion c_i is a binary-verifiable criterion that assesses a desirable property of a reasoning process. These criteria evaluate the policy's reasoning process from different perspectives, and collectively, they form a checklist C^x for a comprehensive assessment of reasoning quality. Most importantly, **the checklist C^x is question-specific:** a unique checklist is generated for each question x. This enables a tailored evaluation of reasoning quality for different questions.

To perform the verification, K2V employs a generative LLM as the judge model, denoted as J. The judge model assesses the reasoning process z against each criterion c_i in the checklist C^x one by one (see Appendix I.3 for the prompt). For each pair (z, c_i), the judge model outputs a binary score $v_i \in \{0, 1\}$, where $v_i = 1$ indicates that z satisfies the criterion c_i, and $v_i = 0$ otherwise. We can define this verification process as:

$$v_i = J(z, c_i) \tag{4}$$

Once the judge model evaluates all criteria, K2V aggregates these binary scores to compute a pass rate $p \in [0, 1]$, representing the proportion of criteria that the reasoning process z successfully meets:

$$p = \frac{1}{k} \sum_{i=1}^{k} v_i \tag{5}$$

where v_i is computed by Equation 4. This approach decomposes the intractable reasoning verification task into a series of binary-verifiable subtasks. The pass rate p can serve as a dense reward signal.

Checklist Synthesis. We propose a two-stage synthesis pipeline, as introduced below.

First, we define a set of general criteria, formally denoted as:

$$G = \{g_1, g_2, \ldots, g_N\} \tag{6}$$

where each g_i is a universal principle that characterizes a high-quality reasoning process, independent of any specific question. We developed general criteria based on the scoring rubrics from the AP Course and Exam Description [1]. See Appendix J for the examples of general criteria.

Second, for a given QA pair consisting of a question x and a ground truth y, we feed both x, y and the set of general criteria G into an LLM, denoted as S. This LLM is prompted to instantiate the general criteria G into a concrete, question-specific checklist C^x (see Appendix I.4 for the prompt). This synthesis process can be expressed as:

$$C^x = \{c_1, \ldots, c_k\} \sim S(\cdot|x, y, G) \tag{7}$$

We conduct a quality assessment of the synthesized checklist, as shown in Appendix F.

3.3 Answer-Gated Reward Mechanism

In order to anchor logical consistency to factual accuracy and prevent potential reward hacking, K2V employs an answer-gated reward mechanism. The total reward R_{total} is defined as:

$$\begin{aligned} R_{\text{total}} &= R_{\text{format}} + R_{\text{answer}} + R_{\text{reason}} \\ R_{\text{reason}} &= \mathbb{1}(\hat{y} = y) \cdot p \end{aligned} \tag{8}$$

where $\mathbb{1}(\cdot)$ is the indicator function, which equals 1 if the predicted answer $\hat{y}$ matches the ground truth y, and 0 otherwise. p denotes the pass rate calculated in Equation 5.

[1]The AP Course and Exam is a program that provides a college-level introductory course curriculum to high school students.

Table 1: Overall performance on three different knowledge-intensive domains. To evaluate the model's performance more comprehensively in the agricultural domain, we select agriculture-related subsets from CMMLU and MMLU. A similar evaluation strategy is applied to the legal and medical domains. We use the base version for the Qwen backbone and the instruction-tuned version for the Llama backbone. We **bold** the best result and underline the suboptimal one. Avg denotes the average accuracy of a model in a specific domain.

Model	Agriculture				Law				Medicine			
	SeedBench	CMMLU	MMLU	Avg	LawBench	CMMLU	MMLU	Avg	MedQA	CMMLU	MMLU	Avg
Qwen-2.5-3B (Backbone)												
Qwen2.5-3B-Instruct	45.67	61.23	73.94	60.28	42.39	62.82	**63.10**	56.10	73.01	61.22	69.67	67.97
Liquid-3B-Qwen	53.60	60.22	68.07	60.63	32.95	52.98	58.78	48.24	68.89	58.11	63.44	63.48
Genie-3B-Qwen	58.09	59.59	73.34	63.67	36.94	60.45	59.71	52.37	73.02	61.69	68.50	67.74
SDR-3B-Qwen	59.06	63.82	73.99	65.62	37.85	63.37	61.02	54.08	73.67	64.82	**70.89**	69.79
BDS-3B-Qwen	54.71	65.33	70.83	63.62	32.44	68.55	60.82	53.94	72.94	64.54	62.20	66.56
K2V-3B-Qwen	**62.82**	**66.82**	**75.40**	**68.34**	**43.27**	**71.55**	62.01	**58.94**	**78.45**	**67.53**	70.76	**72.24**
Qwen-2.5-7B (Backbone)												
Qwen2.5-7B-Instruct	49.68	72.12	85.45	69.08	54.76	75.04	68.99	66.26	80.71	76.06	80.21	78.99
Liquid-7B-Qwen	61.44	68.42	79.91	69.92	43.98	71.67	65.24	60.30	82.90	74.63	73.39	76.97
Genie-7B-Qwen	64.33	73.73	80.56	72.87	48.07	76.70	66.08	63.62	83.55	77.32	74.68	78.52
SDR-7B-Qwen	65.20	76.73	82.61	74.85	48.20	78.25	69.94	65.46	81.03	79.51	77.96	79.50
BDS-7B-Qwen	60.93	75.72	79.55	72.07	45.63	74.95	68.09	62.89	79.31	74.62	71.05	75.00
K2V-7B-Qwen	**66.81**	**79.16**	**88.30**	**78.09**	**55.20**	**79.53**	**70.69**	**68.47**	**87.16**	**81.36**	**80.46**	**83.00**
Llama-3.2-3B-Instruct (Backbone)												
Llama-3.2-3B-Instruct	41.79	44.41	71.59	52.60	30.36	42.13	56.31	42.93	55.22	44.27	68.19	55.90
Liquid-3B-Llama	51.89	45.52	69.88	55.76	31.85	43.64	59.92	45.13	63.59	43.34	67.71	58.21
Genie-3B-Llama	55.71	42.18	69.39	55.76	31.79	41.46	**61.39**	44.88	63.16	44.85	65.86	57.96
SDR-3B-Llama	56.41	48.07	**73.16**	59.21	33.57	43.57	58.54	45.22	66.16	46.52	70.59	61.09
BDS-3B-Llama	53.75	48.41	65.56	55.91	30.62	43.14	61.04	44.93	64.03	42.47	61.81	56.10
K2V-3B-Llama	**58.60**	**50.53**	71.86	**60.33**	**35.50**	**44.90**	61.29	**47.23**	**68.16**	**49.43**	**71.16**	**62.92**
Llama-3.1-8B-Instruct (Backbone)												
Llama-3.1-8B-Instruct	51.10	55.74	82.37	63.07	37.61	54.49	70.22	54.11	68.62	53.55	**79.39**	67.19
Liquid-8B-Llama	59.84	53.77	81.84	65.15	40.43	53.25	68.54	54.07	69.27	54.01	77.79	67.02
Genie-8B-Llama	62.50	55.49	77.95	65.31	41.07	51.17	70.15	54.13	68.97	52.41	76.39	65.92
SDR-8B-Llama	62.11	54.42	80.22	65.58	41.19	55.22	66.89	54.43	70.26	53.63	77.08	66.99
BDS-8B-Llama	62.53	56.76	78.75	66.01	39.96	51.52	67.73	53.07	66.38	50.99	73.83	63.73
K2V-8B-Llama	**63.90**	**58.43**	**85.14**	**69.16**	**42.35**	**57.96**	**70.79**	**57.03**	**72.55**	**56.32**	78.50	**69.12**

Format Reward. The format reward R_{format} encourages the policy model to generate outputs in a structured format. The desired output template is consistent with that of DeepSeek-R1 (Guo et al., 2025a). A maximum score of 0.75 is awarded if the output perfectly adheres to this template.

Answer Reward. The answer reward R_{answer} is a binary reward, where a score of α is awarded if the predicted answer $\hat{y}$ equals the ground truth y; otherwise, the score is 0. In our main experiments, we set $\alpha = 6$. We also conduct a sensitivity analysis on α, as detailed in Section 4.4.

Reasoning Reward. The reasoning reward R_{reason} is a dense reward signal, which is gated by the answer's correctness. If the predicted answer $\hat{y}$ matches the ground truth y, the model receives a reasoning reward equal to the pass rate p. Otherwise, the reasoning reward is 0, regardless of how logical the reasoning procss appear. We conduct an ablation experiment on the answer-gated mechanism, as detailed in Section 4.3.

4 Experiments

4.1 Experimental Setups

Domains. We conduct experiments in three typical knowledge-intensive domains: agriculture, law, and medicine. While these domains suffer from a scarcity of high-quality data due to the high cost of expert annotation and involve open-ended text that are difficult to verify automatically, reasoning abilities are required.

Corpora. For agriculture, we use RiceCorpus (Yang et al., 2025b), a corpus built by crawling resources from the internet, totaling approximately 37.64 MB. For law, we use DISC-Law-SFT (Yue et al., 2023), a dataset of unverifiable QA pairs. We directly concatenate the questions and answers, then randomly downsample 20.69 MB of the text. This demonstrates that K2V can convert unverifiable SFT data into verifiable data. For medicine, we use shibing624-medical-pretrain [2], a Chinese

[2]https://huggingface.co/datasets/ticoAg/shibing624-medical-pretrain

Table 2: Performance on general benchmarks. K2V-based models apply reinforcement learning directly to the Qwen2.5 base backbones without SFT on any general or mathematical data.

Model	BBH	MATH-500	GSM8K	AIME2024	GPQA-Diamond
Qwen2.5-3B-Instruct	37.21	**66.40**	**85.05**	5.00	31.94
K2V-3B-Qwen	**42.64**	65.20	84.17	**5.83**	**32.45**
Qwen2.5-7B-Instruct	49.63	**76.25**	**91.51**	11.67	35.86
K2V-7B-Qwen	**55.36**	74.60	88.67	**13.33**	**36.62**

corpus sourced from online medical encyclopedias and textbooks. We randomly downsample 23.51 MB of text from this corpus.

LLM Backbones. We adopt widely used open-source models, including Qwen2.5-3B, Qwen2.5-7B, Llama3.2-3B-Instruct, and Llama3.1-8B-Instruct, as backbones. We use them to examine the performance of K2V due to their moderate foundational capabilities in knowledge-intensive domains.

Evaluation. For agriculture, to ensure an fair evaluation, we evaluate on the objective question subset of SeedBench (Ying et al., 2025) and conduct evaluations on agriculture-related subsets of CMMLU(Li et al., 2024) and MMLU (Hendrycks et al., 2020). For law, we evaluate on LawBench (Fei et al., 2024), as well as on law-related subsets of CMMLU and MMLU. For medicine, we evaluate on MedQA-MCMLE (Jin et al., 2021), as well as on medicine-related subsets of CMMLU and MMLU. The details of subset selection for all three domains are provided in the Appendix A. Additionally, we conduct evaluations on general benchmarks, including BBH (Suzgun et al., 2023), MATH-500 (Hendrycks et al., 2021; Lightman et al., 2023), GSM8K (Cobbe et al., 2021), AIME2024 (MAA, 2024), and GPQA-Diamond (Rein et al., 2024). We conduct evaluations in a zero-shot setting with a temperature of 0.6. All models are evaluated on each benchmark four times and the average accuracy is reported.

Implementation Details. All experiments were conducted using the DAPO (Yu et al., 2025b) algorithm for training. We sample 8 responses per prompt for a batch of 64 prompts. The clip threshold is set to (0.2, 0.28) and a learning rate of 1×10^{-6}. The maximum generation length and overlong buffer length are set to 4096 and 512. Both the KL divergence coefficient and the entropy regularization coefficient are set to 0. Unless otherwise specified, $\mathcal{F}_{\text{text}}$, S, and the model used for KG construction are Qwen2.5-72B-Instruct, while the judge model J is Qwen2.5-7B-Instruct. See Appendix B for more implementation details.

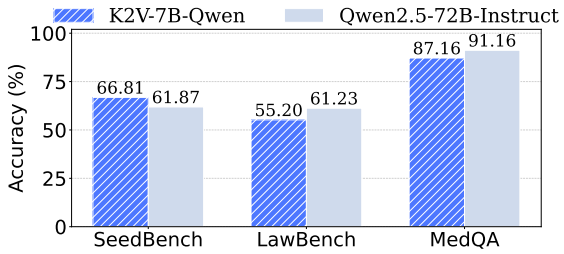

Figure 2: The accuracy of K2V-7B-Qwen and Qwen2.5-72B-Instruct in knowledge-intensive domains. A small LLM trained with K2V and domain corpora can even outperform a much larger LLM.

Baselines. Following the discussion in Section 2, we compare K2V against the following baseline methods. (1) Liquid (Lee et al., 2023) directly extracts questions with multiple verifiable candidate answers from corpora. (2) Genie (Yehudai et al., 2024) synthesizes QA pairs for standard SFT. We perform necessary modifications on its prompt to enhance the verifiability of the synthesized data. (3) SDR (Guo et al., 2025b) uses a task-definition approach to synthesize verifiable QA pairs for RLVR. (4) BDS (Dedhia et al., 2025) synthesizes verifiable multiple-choice data based on a KG. See Appendix C for details of these baseline methods. In addition, we use official LLMs (*i.e.* Qwen and Llama) as the *vanilla* baselines. In contrast to baseline methods, the *vanilla* baselines are not trained on the domain corpora.

4.2 Quantatitive Comparison with Existing RLVR Studies

The main experimental results on three knowledge-intensive domains are reported in Table 1, from which we present the following findings. **Firstly**, K2V significantly enhances reasoning capabilities in knowledge-intensive domains. Across various backbone models, our method overall achieves the best average accuracy in all three domains, outperforming all baseline methods. **Secondly**,

Table 3: Ablation studies on components. Using K2V-3B-Qwen as the baseline, we separately ablate answer verification, reasonging verification, and answer-gated reward mechanism. SFT denotes the model fine-tuned on QA pairs synthesized by our method, using Qwen2.5-3B as the base model.

Model	Knowledge-Intensive Domains			General Domains				
	SeedBench	LawBench	MedQA	BBH	MATH-500	GSM8K	AIME2024	GPQA-Diamond
K2V-3B-Qwen	**62.82**	**43.27**	**78.45**	**42.64**	**65.20**	**84.17**	**5.83**	**32.45**
w/o Answer Verification	43.49	36.88	66.93	20.50	56.65	46.94	0	25.13
w/o Reasoning Verification	61.58	37.97	75.86	39.93	42.40	72.92	0.83	31.69
w/o Answer-Gated	53.35	35.10	72.18	35.11	53.85	58.95	1.67	28.16
SFT	41.38	30.97	54.12	30.49	14.80	20.71	1.67	25.00

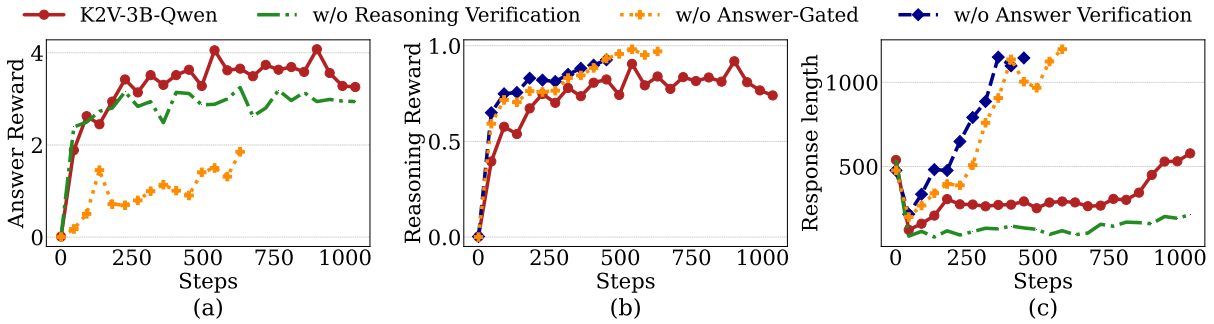

Figure 3: Training dynamics of ablation studies. The **K2V-3B-Qwen** demonstrates stable learning across all metrics. (a) and (c) show that **removing reasoning verification** impairs the model's ability to explore correct answers. (b) and (c) show that **removing answer verification** leads to reward hacking, where the model generates excessively long responses to maximize the reasoning reward, causing training instability. (a), (b) and (c) show that **removing answer-gated reward mechanism** decouples reasoning from factual accuracy. This leads to reward hacking, where the model achieves high reasoning reward through excessive response length and learns reasoning patterns that lead to incorrect answers.

K2V better enables a small LLM to achieve performance comparable to or even exceeding that of a much larger LLM. As shown in Figure 2, K2V-7B-Qwen exceeds Qwen2.5-72B-Instruct on SeedBench. Meanwhile, its performance on MedQA is close to that of Qwen2.5-72B-Instruct. **Finally**, when we aim to investigate the negative effect of domain-specific training on LLM's general reasoning abilities, surprisingly, we found that K2V not only retained these skills but even improved them. As shown in Table 2, without using any general or mathematical data, models trained with K2V do not exhibit a significant decline in general reasoning abilities compared to the *vanilla* baselines, and even shows improvements on certain benchmarks. Specifically, K2V-3B-Qwen and K2V-7B-Qwen, which are trained directly on Qwen2.5 base backbones without prior SFT, consistently surpass the *vanilla* baselines across the BBH, AIME2024, and GPQA-Diamond. This suggests that K2V might have cross-domain advantages: the domain-specific training enables the model's reasoning capabilities to generalize to certain general tasks.

4.3 Ablation Studies

To investigate the individual components of K2V, we conduct comprehensive ablation studies. We compare the K2V-3B-Qwen against four variants under the same data and training parameters.

K2V-3B-Qwen. This is our complete model, trained with both reasoning verification (R_{reason}) and answer verification (R_{answer}). As shown in Table 3, this configuration consistently achieves the highest performance across both knowledge-intensive domains and general domains. The training dynamics in Figure 3 reveal a stable and effective learning process. Both the answer and reasoning reward increase steadily, while the response length follows a healthy increase pattern after an initial decrease.

w/o Answer Verification. This variant removes the answer verification (R_{answer}) and is trained solely with the reasoning verification (R_{reason}). As shown in Figure 3(b), although this model initially achieves higher reasoning reward, it does so through reward hacking. The model discovers that longer outputs more readily meet the criteria

Table 4: Sensitivity analysis of the answer reward α. Using K2V-3B-Qwen as the study object, we test $\alpha \in \{2, 4, 6, 8, 10\}$ while holding all other parameters fixed. The results across both knowledge-intensive and general domains demonstrate that K2V is robust to different magnitudes of the answer reward.

α	Knowledge-Intensive Domains			General Domains				
	SeedBench	LawBench	MedQA	BBH	MATH-500	GSM8K	AIME2024	GPQA-Diamond
2	62.45	42.41	76.30	42.81	64.95	81.80	5.83	29.55
4	62.82	43.27	78.78	42.64	63.60	82.88	5.83	32.07
6	62.82	43.27	78.45	42.64	65.20	84.17	5.83	32.45
8	61.98	42.90	77.29	42.37	65.90	81.37	9.17	30.43
10	63.20	41.93	78.69	41.58	60.95	81.46	8.33	31.06

of checklist, thus it tends to generate excessively long responses to maximize the reasoning reward. As shown in Figure 3(c), its response length grows uncontrollably, leading to out-of-memory (OOM) errors that halted training around 500 steps.

w/o Reasoning Verification. This variant removes the reasoning verification (R_{reason}) and is trained solely with the answer verification (R_{answer}), making it akin to a conventional RLVR. Figure 3(a) shows that without the guidance from reasoning verification, the model achieves a lower answer reward. This demonstrates that the dense signal from R_{reason} is vital for accelerating the model's learning helping it efficiently explore correct answer. Furthermore, as shown in Figure 3(c), its response length stagnates after an initial drop, suggesting a less effective learning.

w/o Answer-Gated. This variant removes the answer-gated reward mechanism, meaning the reasoning reward R_{reason} is no longer gated by the final answer. Regardless of whether the predicted answer is correct, the R_{reason} is set to the pass rate p calculated in Equation 5. As shown in Table 3, this model suffers from a significant performance degradation across all benchmarks. Although Figure 3(b) shows the model achieves high reasoning rewards, its answer reward remains at a low level in Figure 3(a). This indicates that the reasoning process becomes decoupled from factual accuracy, where the model learns reasoning patterns that can lead to incorrect answers. Furthermore, the rapid increase in response length in Figure 3(c) suggests the model suffers reward hacking by generating long text to satisfy the checklist criteria.

SFT. This variant forgoes reinforcement learning entirely and instead uses the QA pairs synthesized by K2V to perform standard SFT on the Qwen2.5-3B. As detailed in Table 3, this model suffers a significant performance collapse on all benchmarks. We hypothesize that this is because the highly structured, short-answer format data synthesized by K2V is ill-suited for SFT, which typically relies on more diverse and conversational data. SFT on verifiable QA pairs merely teaches the model a superficial mapping from questions to answers, rather than enabling it to learn deeper reasoning skills.

4.4 Sensitivity Analysis of Answer Reward α

As shown in Equation 8, the answer reward R_{answer} is set to the variable α when the model predicts the correct final answer. To investigate the impact of the different answer reward magnitude, we conduct a sensitivity analysis. Specifically, we test $\alpha \in \{2, 4, 6, 8, 10\}$ on K2V-3B-Qwen while holding all other parameters fixed.

The results in Table 4 demonstrate that K2V is insensitive to α and maintains stable performance across all configurations. This robustness is primarily attributed to DAPO's group reward normalization, which focuses on the relative ranking of responses for a rollout rather than absolute reward magnitudes. Additionally, we believe that the dense reasoning reward signals provided by checklist-style verification effectively guide the model's exploration, reducing its reliance on answer rewards.

4.5 Results of Smaller Generator

Our approach relies on the in-context understanding ability of LLMs for data synthesis. Specifically, we employ Qwen2.5-72B-Instruct as a generator to convert the masked quintuplets into fill-blank style QA pairs ($\mathcal{F}_{\text{text}}$), instantiate the general criteria (S), and construct the KG. To investigate whether smaller models can serve as generators for data synthesis, We replace the 72B generator with Qwen2.5-14B-Instruct and Qwen2.5-7B-Instruct to perform the entire pipeline of data synthesis. We then use Qwen2.5-3B as the backbone to conduct

Table 5: Results of smaller generator. We use three instruction-tuned versions of Qwen2.5 models (72B, 14B, and 7B) as generators to perform the entire pipeline of data synthesis. The data synthesized by each generator is then used to train Qwen2.5-3B.

Generator size	Knowledge-Intensive Domains			General Domains				
	SeedBench	LawBench	MedQA	BBH	MATH-500	GSM8K	AIME2024	GPQA-Diamond
72B	62.82	43.27	78.45	42.64	65.20	84.17	5.83	32.45
14B	59.71	40.62	76.16	41.36	62.40	83.24	5.83	31.31
7B	60.94	42.79	76.74	41.29	63.00	83.85	5.83	31.44

Table 6: Computational cost of data synthesis. We report the computational cost (GPU hours on one H100) when the generator is Qwen2.5-72B-Instruct.

Stage	GPU hours
KG construction	201.25
QA pairs synthesis	27.00
Checklist synthesis	53.34
Overall	281.59

experiments on the data synthesized by these different generators. For a fair comparison, all training parameters are kept identical to those in the main experiment, and the judge model J remains Qwen2.5-7B-Instruct.

As shown in Table 5, using a 7B or 14B model as the generator results in only a slight performance drop compared to the 72B model. This demonstrates that K2V is robust to the size of generator and can synthesize high-quality data without strictly relying on large generators.

4.6 Computational Cost of Data Synthesis

Our method relies on LLMs throughout the entire pipeline of data synthesis, including KG construction, QA pairs synthesis, and checklist synthesis. To ensure the cost is manageable, we report the computational cost (GPU hours on one H100) of the LLM-related stages when using Qwen2.5-72B-Instruct as the generator. As shown in Table 6, even with a relatively large model as the generator, the computational cost of K2V for data synthesis remains within a generally acceptable range.

5 Conclusion

In this paper, we present K2V, a novel framework that explores RLVR in knowledge-intensive domains. First, we develop fill-blank style verification to automatically synthesize verifiable QA pairs. Moreover, we introduce a checklist-style verification method that not only verifies the model's reasoning process but also provides a dense reward signal. Finally, we propose an answer-gated reward mechanism to integrate answer reward and reasoning reward, which ensures the factual correctness and prevents potential reward hacking. Comprehensive experimental results show that K2V improves reasoning capabilities in knowledge-intensive domains without significantly compromising general capabilities. In the future, we aim to study the multimodal verifiable data synthesis, which plays a critical role in knowledge-intensive reasoning.

Limitations

Despite its effectiveness, this work has a few limitations. First, due to computational resource constraints, our experiments were primarily focused on small to medium-sized models, such as the 3B and 8B versions of Qwen and Llama; therefore, the scalability of K2V on ultra-large-scale models (e.g., 70B parameters) remains to be further explored. Second, the current framework is specifically designed to address the lack of verifiable data in knowledge-intensive domains like agriculture, law, and medicine. Consequently, we did not extend our synthesis and verification methods to the fields of mathematics and coding, which already have relatively mature verification mechanisms. Future research could focus on integrating these diverse domains into a more unified RLVR framework and evaluating performance across a broader range of model scales.

Acknowledgments

This work was supported by the New Generation Artificial Intelligence-National Science and Technology Major Project (2025ZD0121801).

References

Sultan Alneyadi, Elankayer Sithirasenan, and Vallipuram Muthukkumarasamy. 2016. A survey on data leakage prevention systems. *Journal of Network and Computer Applications*, 62:137–152.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vladimir Mikulik, Samuel R. Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. 2025a. Reasoning models don't always say what they think. *CoRR*, abs/2505.05410.

Zihong Chen, Wanli Jiang, Jinzhe Li, Zhonghang Yuan, Huanjun Kong, Wanli Ouyang, and Nanqing Dong. 2025b. Graphgen: Enhancing supervised fine-tuning for llms with knowledge-driven synthetic data generation. *arXiv preprint arXiv:2505.20416*.

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.

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2022. *Introduction to algorithms*. MIT press.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, and 4 others. 2025. Process reinforcement through implicit rewards. *CoRR*, abs/2502.01456.

Bhishma Dedhia, Yuval Kansal, and Niraj K Jha. 2025. Bottom-up domain-specific superintelligence: A reliable knowledge graph is what we need. *arXiv preprint arXiv:2507.13966*.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069.

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Alan Huang, Songyang Zhang, Kai Chen, Zhixin Yin, Zongwen Shen, Jidong Ge, and Vincent Ng. 2024. LawBench: Benchmarking legal knowledge of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7933–7962, Miami, Florida, USA. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025a. Deepseek-r1: incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Yiduo Guo, Zhen Guo, Chuanwei Huang, Zi-Ang Wang, Zekai Zhang, Haofei Yu, Huishuai Zhang, and Yikang Shen. 2025b. Synthetic data rl: Task definition is all you need. *arXiv preprint arXiv:2505.17063*.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. 2025. Skywork open reasoner 1 technical report. *CoRR*, abs/2505.22312.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

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. *arXiv preprint arXiv:2103.03874*.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. 2022. Knowledge graphs. *ACM Comput. Surv.*, 54(4):71:1–71:37.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. 2021. A survey on knowledge graphs: Representation, acquisition, and applications.

IEEE transactions on neural networks and learning systems, 33(2):494–514.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Imed Keraghel, Stanislas Morbieu, and Mohamed Nadif. 2024. A survey on recent advances in named entity recognition. *arXiv preprint arXiv:2401.10825*.

Seongyun Lee, Hyunjae Kim, and Jaewoo Kang. 2023. Liquid: A framework for list question answering dataset generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13014–13024.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024. Cmmlu: Measuring massive multitask language understanding in chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11260–11285.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Yuxing Lu, Wei Wu, Xukai Zhao, Rui Peng, and Jinzhuo Wang. Karma: Leveraging multi-agent llms for automated knowledge graph enrichment. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, and Maurice Weber. 2025a. Deepcoder: A fully open-source 14b coder at o3-mini level. *Notion Blog*.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025b. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. Notion Blog.

Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. 2025. General-reasoner: Advancing llm reasoning across all domains. *arXiv preprint arXiv:2505.14652*.

MAA. 2024. American invitational mathematics examination - aime. Accessed on: 2025-09-21.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 244 others. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.

Miao Peng, Nuo Chen, Zongrui Suo, and Jia Li. 2025. Rewarding graph reasoning process makes llms more generalized reasoners. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 2257–2268.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Maohao Shen, Guangtao Zeng, Zhenting Qi, Zhang-Wei Hong, Zhenfang Chen, Wei Lu, Gregory Wornell, Subhro Das, David Cox, and Chuang Gan. 2025. Satori: Reinforcement learning with chain-of-action-thought enhances llm reasoning via autoregressive search. In *International Conference on Machine Learning*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297.

Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, Abdulhakeem Adefioye, Jean Kaddour, and Andreas Köpf. 2025. Reasoning gym: Reasoning environments for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2505.24760*.

Zhenpeng Su, Leiyu Pan, Xue Bai, Dening Liu, Guanting Dong, Jiaming Huang, Wenping Hu, and Guorui Zhou. 2025. Klear-reasoner: Advancing reasoning capability via gradient-preserving clipping policy optimization. *arXiv preprint arXiv:2508.07629*.

Mujeen Sung, Minbyul Jeong, Yonghwa Choi, Donghyeon Kim, Jinhyuk Lee, and Jaewoo Kang. 2022. Bern2: an advanced neural biomedical named

entity recognition and normalization tool. *Bioinformatics*, 38(20):4837–4839.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Zengzhi Wang, Fan Zhou, Xuefeng Li, and Pengfei Liu. 2025. Octothinker: Mid-training incentivizes reinforcement learning scaling. *arXiv preprint arXiv:2506.20512*.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. 2025. Light-r1: Curriculum SFT, DPO and RL for long COT from scratch and beyond. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 318–327, Vienna, Austria. Association for Computational Linguistics.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Fan Yang, Huanjun Kong, Jie Ying, Zihong Chen, Tao Luo, Wanli Jiang, Zhonghang Yuan, Zhefan Wang, Zhaona Ma, Shikuan Wang, Wanfeng Ma, Xiaoyi Wang, Xiaoying Li, Zhengyin Hu, Xiaodong Ma, Minguo Liu, Xiqing Wang, Fan Chen, and Nanqing Dong. 2025b. Seedllm·rice: A large language model integrated with rice biological knowledge graph. *Molecular Plant*, 18(7):1118–1129.

Liang Yao, Jiazhen Peng, Chengsheng Mao, and Yuan Luo. 2025. Exploring large language models for knowledge graph completion. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Asaf Yehudai, Boaz Carmeli, Yosi Mass, Ofir Arviv, Nathaniel Mills, Assaf Toledo, Eyal Shnarch, and Leshem Choshen. 2024. Genie: Achieving human parity in content-grounded datasets generation. *arXiv preprint arXiv:2401.14367*.

Jie Ying, Zihong Chen, Zhefan Wang, Wanli Jiang, Chenyang Wang, Zhonghang Yuan, Haoyang Su, Huanjun Kong, Fan Yang, and Nanqing Dong. 2025. SeedBench: A multi-task benchmark for evaluating large language models in seed science. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31395–31449, Vienna, Austria. Association for Computational Linguistics.

Ping Yu, Jack Lanchantin, Tianlu Wang, Weizhe Yuan, Olga Golovneva, Ilia Kulikov, Sainbayar Sukhbaatar, Jason Weston, and Jing Xu. 2025a. Cot-self-instruct: Building high-quality synthetic prompts for reasoning and non-reasoning tasks. *arXiv preprint arXiv:2507.23751*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025b. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, and Zhongyu Wei. 2023. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *Preprint*, arXiv:2309.11325.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.

Xiaoyan Zhao, Yang Deng, Min Yang, Lingzhi Wang, Rui Zhang, Hong Cheng, Wai Lam, Ying Shen, and Ruifeng Xu. 2024. A comprehensive survey on relation extraction: Recent advances and new frontiers. *ACM Computing Surveys*, 56(11):1–39.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*

36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.

Xiangxin Zhou, Zichen Liu, Anya Sims, Haonan Wang, Tianyu Pang, Chongxuan Li, Liang Wang, Min Lin, and Chao Du. 2025. Reinforcing general reasoning without verifiers. *arXiv preprint arXiv:2505.21493*.

A Details of Model Evaluation

As shown in Section 4, to ensure an objective and fair evaluation, we evaluate the model on the objective subset of SeedBench. Specifically, we used six subsets including 1-1, 1-2, 1-3, 3-1, 3-2, and 3-3. These subsets have unique ground-truth answers, ensuring the fairness of the evaluation.

Furthermore, to conduct a more comprehensive evaluation in the domains of agriculture, law, and medicine, we also select corresponding subsets from CMMLU and MMLU. The specific subsets are as follows:

- CMMLU in the agricultural: virology, high school biology, food science, agronomy.
- CMMLU in the law: legal & moral basis, professional law, jurisprudence, international law, college law.
- CMMLU in the medicine: anatomy, high school biology, medical statistics, virology, college medicine, Clinical knowledge, professional medicine, traditional, chinese medicine.
- MMLU in the agricultural: high school biology, college biology.
- MMLU in the law: international law, jurisprudence, professional law.
- MMLU in the medicine: anatomy, clinical knowledge, high school biology, college medicine, medical genetics, professional medicine, college biology.

We use OpenCompass (Contributors, 2023) as the evaluation framework. All evaluations were conducted in a zero-shot setting. We perform four inference runs for each benchmark and reported the Avg@4 results. We employ a temperature of 0.6.

B Impltementation Details of K2V

In this section, we describe the implementation details of our work, including data synthesis and model training.

B.1 Data Synthesis

K2V first constructs a knowledge graph (KG) using the Graphgen (Chen et al., 2025b) framework (parameters are listed in Table 8). Graphgen is a framework designed for synthesizing SFT data that utilizes LLMs for Entity Recognition (ER) and Named Entity Recognition (NER); we modified its code to suit the requirements of K2V. Next, K2V samples quintuplets from the knowledge graph and randomly masks an entity. An LLM then converts these masked quintuplets into verifiable question-answer (QA) pairs in a fill–blank format. Finally, K2V generates a checklist for each QA pair using an LLM. All steps involving LLMs—from KG construction to QA pairs synthesis and checklist generation—employ Qwen2.5-72B-Instruct.

B.2 Model Training

All experiments were implemented using the DAPO algorithm (Yu et al., 2025b) based on the verl framework (Sheng et al., 2025). For each batch of 64 prompts, we sampled 8 responses per prompt. The clipping threshold and learning rate were set to (0.2, 0.28) and 1×10^{-6}. The number of training epochs is set to 2. The maximum generation and overlong buffer lengths were fixed at 4,096 and 512, respectively. To ensure high-quality training data, we employ the dynamic sampling strategy with group filtering. Specifically, we generate candidates from a prompt batch that is three times larger than the training batch (i.e., a generation batch size of 192) and filter them based on sequence rewards. For the optimization objective, both the KL divergence coefficient and the entropy regularization coefficient are set to 0. To accelerate the rollout phase, we utilize the vLLM engine. Furthermore, to optimize GPU memory usage, we enable Fully Sharded Data Parallel (FSDP) with parameter and optimizer offloading, as well as gradient checkpointing. The judge model J is Qwen2.5-7B-Instruct. All experiments were conducted on 8 Nvidia H100 GPUs. A summary of the key hyperparameters can be found in Table 7.

C Impltementation Details of Baselines

Liquid. We reproduced the Liquid (Lee et al., 2023) method as our baseline, which consists of four stages: answer extraction, question generation, iterative filtering, and answer expansion. For entity recognition, we used spaCy (Honnibal et al., 2020) and BERN2 (Sung et al., 2022) to identify general-domain and biomedical entities respectively, employing the same corpus as our K2V method. While default models were applied in all model-dependent stages, we replaced them with structurally similar Chinese-adapted models for

Table 7: Key hyperparameters for model training in verl.

Hyperparameter	Value
Optimization & Algorithm	
Algorithm	DAPO
Learning rate	1×10^{-6}
Clip ratio range (asymmetric)	$[0.2, 0.28]$
KL coefficient	0.0
Entropy coefficient	0.0
Overlong buffer length	512
Overlong penalty factor	1.0
Data & Generation	
Global batch size	64
Mini-batch size	32
Micro-batch size (per GPU)	16
Generation batch size	192
Rollout samples per prompt (N)	8
Max prompt length	1024
Max response length	4096
System & Infrastructure	
Inference engine	vLLM
FSDP offload (Param & Optim)	True
Gradient checkpointing	True
GPU memory utilization	0.8

Chinese corpus processing, as the original models do not support Chinese. All training configurations align with those used in K2V.

Genie. We re-implemented Genie (Yehudai et al., 2024) as a baseline, following its three-stage pipeline of content preparation, generation, and filtering. To ensure a fair comparison with our K2V method, we employ Qwen2.5-72B-Instruct for the generation stage. We slightly modify the original prompts to improve the verifiability of the generated data, making it better suited for RLVR. Since the official source code is not publicly available, we independently implemented the filtering mechanism based on the methodology described in the paper, including checks for format, faithfulness, and quality. All training settings follow those of K2V.

Synthetic Data RL. We adopted the data generation pipeline from the Synthetic Data RL (SDR) (Guo et al., 2025b) framework as a comparative baseline. Utilizing the official implementation , we employed Qwen2.5-72B-Instruct as the instructor model for data synthesis and rewriting while Qwen2.5-7B-Base served as the reference model to assess sample difficulty. We customized the task description and format instructions to suit the agriculture, legal, and medical domains and grounded the synthesis process by retrieving relevant context from our proprietary raw corpora. We strictly adhered to the original protocol by performing initial generation, difficulty-adaptive rewriting, and consistency-based filtering to isolate high-potential samples. All training settings follow those of K2V.

BDS. We re-implemented Bottom-up Domain-specific Superintelligence (BDS) (Dedhia et al., 2025) as a baseline, which consists of three stages: (1) Content Preparation: We construct a knowledge graph and systematically extract 4-node simple paths (3-hop relations) as logical chains for question generation. To ensure computational feasibility, we limit the maximum number of paths to 20,000 and restrict source nodes to those with outgoing edges, considering the first 1,000 such nodes. (2) Generation: We employ Qwen2.5-72B-Instruct for the generation stage, consistent with our K2V method settings. For each extracted path, we construct a prompt that requests a question in multiple-choice format (4 options). (3) Filtering: Following the description in the BDS paper, We redesign a rule-based post-processing function to enforce strict quality control. This parser validates the output structure, extracts question stems, options, and correct answers, and automatically discards malformed generations. All training configurations align with those used in K2V.

D Results Across Diverse Backbones

In Section 4.2, we conduct experiments in three knowledge-intensive domains using the Qwen base models (Qwen2.5-3B and Qwen2.5-7B) and the Llama instruction-tuned models (Llama-3.2-3B-Instruct and Llama-3.1-8B-Instruct).

We also conduct experiments using the Qwen instruction-tuned model (Qwen2.5-3B-Instruct) and the Llama base model (Llama-3.2-3B). As shown in Table 9, K2V-3B-Llama-Base exhibits a severe performance collapse. It is important to emphasize that this failure is not attributable to flaws within the K2V framework itself, but is a direct consequence of the RL-unfriendly nature of the Llama-3.2-3B. As analyzed by OctoThinker (Wang et al., 2025), Llama base models demonstrate instability during RL due to a significant gap between their pretraining data distribution and the requirements of reasoning tasks. This disparity causes the model to produce anomalous behaviors, such as premature answer generation or falling into infinite loops. The robustness of K2V has been demonstrated by the success of Qwen2.5-3B and Llama-3.2-3B-Instruct. The primary bottleneck lies in the Llama base model's lack of necessary cold-start.

Table 8: The hyperparameters of GraphGen, which is used to construct the KG.

Parameter	Our Value	Description
qa_form	aggregated	Type of QA form desired.
expand_method	max_width	Method for controlling graph expansion.
bidirectional	True	Expanding the graph in both directions (True) or one direction (False).
max_extra_edges	2	Maximum number of edges to expand.
max_tokens	256	Maximum number of tokens.
max_depth	1	Maximum depth for traversal in each direction.
edge_sampling	max_loss	Strategy for edge selection at the same layer.
isolated_node_strategy	ignore	Handling strategy for isolated nodes.

Table 9: Results across diverse models. K2V-3B-Qwen and K2V-3B-Qwen-Ins use Qwen2.5-3B and Qwen2.5-3B-Instruct as their backbone models, respectively. Similarly, K2V-3B-Llama and K2V-3B-Llama-Base are built upon the Llama-3.2-3B-Instruct and Llama-3.2-3B, respectively. The performance collapse of K2V-3B-Llama-Base is not attributable to K2V, but rather to the RL-unfriendly characteristics of Llama-3.2-3B. This base model has significant distributional gap between pretraining data and reasoning tasks, which causes the model to exhibit anomalous behaviors, such as prematurely generating answers or falling into infinite loops. In fact, this performance collapse is not unique to K2V; several other baseline methods encounter similar issues on the Llama-3.2-3B.

Model	Agriculture				Law				Medicine			
	SeedBench	**CMMLU**	**MMLU**	**Avg**	**LawBench**	**CMMLU**	**MMLU**	**Avg**	**MedQA**	**CMMLU**	**MMLU**	**Avg**
Qwen models												
Qwen2.5-3B-Instruct	45.67	61.23	73.94	60.28	42.39	62.82	63.10	56.10	73.01	61.22	69.67	67.97
K2V-3B-Qwen	**62.82**	66.82	75.40	68.34	**43.27**	**71.55**	62.01	58.94	**78.45**	**67.53**	70.76	72.24
K2V-3B-Qwen-Ins	61.38	**67.82**	**76.90**	**68.70**	45.57	65.28	**63.59**	58.14	76.39	65.94	**71.51**	71.28
Llama models												
Llama-3.2-3B-Instruct	41.79	44.41	71.59	52.60	30.36	42.13	56.31	42.93	55.22	44.27	68.19	55.90
K2V-3B-Llama	**58.60**	**50.53**	**71.86**	**60.33**	**35.50**	**44.90**	**61.29**	**47.23**	**68.16**	**49.43**	**71.16**	**62.92**
K2V-3B-Llama-Base	28.83	19.91	29.56	26.10	10.66	13.22	21.66	15.18	25.29	16.44	26.35	22.69

This performance gap suggests that an RL-friendly base model is essential for scaling reasoning capabilities. Although K2V provides high-quality verifiable data and fine-grained rewards, these advantages cannot compensate for a base model's inability to maintain coherent reasoning. The improvement of Llama-3.2-3B-Instruct and Qwen2.5-3B demonstrates K2V's strong generalization across different architectures. The failure of the Llama-3.2-3B base model likely stems from its inability to support direct reinforcement learning without a cold-start.

E Analysis of Data Leakage

To ensure that the performance gains of K2V are not caused by memorizing test data, We checked for overlaps between K2V-synthesized questions and the SeedBench, LawBench, and MedQA benchmarks. We calculate n-gram similarity (Alneyadi et al., 2016) using the Qwen2.5 tokenizer, testing n values from 22 to 30. A test sample is considered leaked if any of its n-token sequences appear in the training set. Table 10 shows the number of leaked samples detected, which indicate a low risk of data leakage.

F Quality Assessment of Checklists

To assess the quality of checklist, we employ Qwen3-235B-A22B-Instruct-2507 as the judge model to assess the checklists. Specifically, we randomly sample 10000 instances from the synthesized checklist across the agriculture, legal, and medical domains. The judge model evaluates each checklist on a scale of 1 to 5 based on three specific dimensions:

- Relevance: Are the items in the checklist directly relevant to the specific question and the ground truth ? Do they check for information that actually matters for this problem ?
- Verifiability: Are the criteria objective and verifiable? Can a third-party evaluator easily determine "yes" or "no" without ambiguity?
- Necessity: Does the checklist cover the neces-

Table 10: Data leakage detection results across different n-gram lengths. The results indicate a low risk of data leakage

Benchmark	Total Samples	n		
		22	26	30
SeedBench	1,975	5	0	0
LawBench	9,500	30	20	1
MedQA	3,426	0	0	0

Table 11: Average quality scores of synthesized checklists. We randomly sample 10000 instances from the synthetic checklists and assess their quality using Qwen3-235B-A22B-Instruct-2507 across three dimensions: relevance, verifiability, and necessity. Score denotes the average value across these 10000 instances, with a range of [1, 5].

Evaluation Dimension	Score
Relevance	4.37
Verifiability	4.29
Necessity	4.60

sary steps or facts required to reach the correct conclusion? Are there missing critical steps or redundant unnecessary steps?

The assessment results in Table 11 demonstrate that the K2V is able to produce high-quality checklists across all dimensions. Specifically, the high necessity score (4.60) indicates that the checklists effectively cover the essential logical chains and knowledge points required for accurate reasoning, while the strong scores in relevance (4.37) and verifiability (4.29) confirm that the criteria are both closely aligned with the questions and sufficiently objective to provide stable reward signals. The results confirm that K2V can automatically synthesize checklists that provide a robust foundation for verifying the LLM's reasoning process.

G Quality Assessment of Knowledge Graph

To comprehensively validate the quality of the KG, we conduct both automated and manual evaluation of the constructed KG.

G.1 Automated Evaluation

Following the previous work of quality evaluation on KG (Lu et al.; Pan et al., 2024), we conduct evaluation on three dimensions: extraction accuracy, semantic consistency, and structural robustness.

G.1.1 Extraction Accuracy

Extraction accuracy measures the quality of named entity recognition (NER) and relation extraction (RE). Based on the LLM-as-judge method (Zheng et al., 2023), we employ Qwen2.5-72B-Instruct to compute the three metrics: (1) Accuracy: Measures the proportion of correct entities among all extracted entities. (2) Completeness: Measures the proportion of ground-truth entities successfully extracted from each text chunk. (3) Precision: Measures the exactness of entity names and descriptions.

As shown in Table 12, our KG construction pipeline demonstrates high extraction quality. Due to time and resource constraints, we randomly sampled 1000 text chunks to evaluate the extraction accuracy.

G.1.2 Semantic Consistency

Semantic consistency measures the degree of semantic conflict among multi-source entities, which refer to entities appearing in multiple text chunks simultaneously. We calculate two types of conflicts: (1) Entity Type Consistency: An entity is considered to be in conflict if its type differs across different chunks. (2) Description Consistency: An entity is considered to be in conflict if its description across different chunks exhibit discrepancies. We employ an LLM-as-Judge approach to detect such conflicts. Evaluation results show that our method achieves a low conflict rate(1.52%).

G.1.3 Structural Robustness

Structural robustness measures the structural integrity of KG. We compute two metrics that represent characteristics of the KG: (1) Noise Ratio: Measures the proportion of isolated nodes (entities without any relationships). Lower values indicate better quality. (2) Connectivity (LCC-Ratio): Measures the proportion of nodes contained within the largest connected component. Higher values indicate better quality. Evaluation results show that our method exhibits low noise ratio (Noise Ratio=0.088) and robust connectivity (LCC-Ratio=0.859).

G.2 Manual Evaluation

To further validate the factual accuracy of constructed KG, we conduct a manual evaluation. Specifically, we first randomly sample 200 entities and 200 relations from the KG, where each entity

Table 12: The result of the extraction accuracy of constructed KG.

Metric	NER	RE
Accuracy	0.726	0.810
Completeness	0.669	0.725
Precision	0.721	0.782

or relation corresponds to a raw text chunk. Secondly, two PhD students manually verify whether the description of sampled entities and relations are consistent with the factual knowledge contained in the corresponding text chunks. Finally, we calculate the consistency rate CR for the sampled entities and relations using the following formula:

$$CR = \frac{M}{N} \tag{9}$$

Where N is the total number of entities or relations. M is the number of entities or relations that are consistent with the raw text chunk.

Evaluation results show that our method yields a consistency rate of 97.00% for entities and 95.50% for relations. These high consistency rates ensure the factual accuracy of constructed KG.

H Qualitative Comparison with Existing RLVR Methods

In this section, we present the QA pairs synthesized by K2V, Liquid, Genie, SDR, and BDS, and provide an analysis of these cases.

H.1 Synthesized Data of K2V

The first case is shown in Figure 4. The question of this case provides a rich clinical and molecular context that necessitates complex medical reasoning. Rather than a simple factual query, the question integrates specific symptoms of CMT2C, such as vocal cord and diaphragm paralysis, with precise genomic data like chromosome 12q23-24 and the protein's role in calcium signaling. The checklist of this case effectively validates whether the model understands the underlying pathophysiology of CMT2C by requiring not only the correct identification of the TRPV4 gene but also its specific chromosomal location (12q23-24) and its biological role in calcium signaling and mechanosensation. This checklist prevents the model from relying on simple keyword matching, as it must explain the causal link between genetic mutations and protein dysfunction to satisfy the criteria.

The second case is shown in Figure 5. The question of this case is more than a simple definition lookup. By describing the interactions between multiple subprocesses, such as mRNA turnover, translational control, and mRNA-binding proteins, it constructs a complex knowledge context. The model must understand both the temporal dimension (occurring after transcription) and the functional dimension (regulation of stability and translation) to successfully infer and reconstruct the core concept of post-transcriptional control. The checklist of this case ensures reasoning completeness by requiring not only a standard definition but also a clear description of essential steps such as mRNA processing, splicing, and export. In terms of technical depth, the checklist specifically mandates explanations for mRNA turnover and translational control, effectively verifying whether the model understands how these mechanisms co-regulate final protein levels.

The third case is shown in Figure 6. The question of this case is not a simple definition of terms. Instead, it establishes a dense contextual framework by detailing the anatomy of the lower medulla, the sensory pathways of the lower body, and the synapsing and decussation of second-order neurons. This narrative requires the model to accurately distinguish the "gracile nucleus" (for the lower body) from the "cuneate nucleus," thereby validating its deep understanding of hierarchical neural pathways. The checklist of this case does not merely repeat the answer; instead, it decomposes the complex physiological process of the Dorsal Column-Medial Lemniscus (DCML) pathway into key logical nodes. First, it accurately captures anatomical specificity by verifying whether the model correctly associates the gracile nucleus with lower-body sensation, as opposed to the cuneate nucleus of the upper body. Second, it covers the dynamics of sensory transmission in detail, from the ipsilateral ascent of first-order neurons in the spinal cord to the synapse in the medulla and the subsequent decussation to the contralateral thalamus.

H.2 Synthesized Data of Liquid

As shown in Figure 7, the data synthesized by Liquid primarily consists of simple, direct fact-retrieval questions that lack the rich contextual constraints necessary to stimulate complex reasoning chains. While these QA pairs cover domain-specific terms, their brevity limits the model's need to perform multi-step logic or integrate diverse

knowledge points, which are essential for developing sophisticated reasoning capabilities. Furthermore, the ground truths often provide a list of synonymous or categorical terms. In knowledge-intensive fields like law or medicine, such relatively shallow tasks may lead the model to prioritize surface-level memorization over logical deduction.

H.3 Synthesized Data of Genie

As shown in Figure 8, the data Synthesized by Genie exhibits limitations in both structural complexity and objective verifiability. While the questions address domain-specific topics, they are primarily formatted as simple factual queries that do not provide enough context to support long-chain reasoning. A key drawback is the production of ground truths that are difficult to verify automatically, as exemplified by Question 2. Instead of a concise term or entity, the answer is a descriptive sentence. This conversational and open-ended format poses a challenge for rule-based verifiers, as it is hard to determine correctness through exact matching or simple parsing.

H.4 Synthesized Data of SDR

As shown in Figure 9, the data format synthesized by SDR can be specified to match the fill-in blank style of K2V. By embedding specific terms within detailed background descriptions, such as biological mechanisms or legal procedures, SDR provides the model with clear logical clues. This setup requires the model to derive the correct answer by integrating the provided context, which ensures the uniqueness of the ground truth. While these questions primarily focus on single-step logical matching and are relatively straightforward compared to complex multi-step reasoning tasks, their rigorous descriptions and accurate knowledge mapping provide a reliable foundation for training models in specialized domains.

H.5 Synthesized Data of BDS

As shown in Figure 10, the BDS method adopts a multiple-choice format, which ensures high verifiability for the model. Since answers are restricted to four options, the evaluation system can precisely judge the correctness of outputs, providing clear and unambiguous feedback signals for reinforcement learning. Regarding reasoning depth, this method utilizes knowledge graphs to link fragmented knowledge points into situational scenarios, such as liability determination in maritime law or complication inference for livestock diseases. This requires models to move beyond simple definition retrieval and perform logical integration across multiple knowledge points to make judgments based on complex stems.

Question:
CMT2C is a specific subtype of Charcot-Marie-Tooth disease type 2 (CMT2), characterized by vocal cord and diaphragm paralysis. This condition is primarily caused by mutations in the { } gene, which is located on chromosome 12q23-24. The { } gene encodes the transient receptor potential cation channel, subfamily V, member 4, a protein that plays a crucial role in calcium signaling and mechanosensation. Mutations in this gene not only lead to CMT2C but are also associated with various other neurological and musculoskeletal disorders. Therefore, the dysfunction of the { } protein due to genetic mutations is a key factor in the development of CMT2C and its related symptoms.

Ground truth:
TRPV4

Checklist:
1. Accurately identifies the gene responsible for CMT2C as TRPV4.
2. Correctly states that the TRPV4 gene is located on chromosome 12q23-24.
3. Describes the role of the TRPV4 protein in calcium signaling and mechanosensation.
4. States that mutations in the TRPV4 gene lead to CMT2C and other neurological and musculoskeletal disorders.
5. Explains that the dysfunction of the TRPV4 protein due to genetic mutations is a key factor in the development of CMT2C and its related symptoms.
6. Avoids over-extrapolation or unfounded speculation beyond the scope of the given evidence.
7. he overall response is well-structured, logically coherent, and clearly written, avoiding self-contradictions and redundant statements.

Figure 4: The first case of K2V.

Question:
{ } is a critical aspect of gene expression regulation that occurs after the RNA has been transcribed. This process encompasses several key steps, including mRNA processing, splicing, export, turnover, and translational control. Firstly, mRNA turnover is a crucial component of { }, involving the degradation of mRNA molecules. This degradation regulates the amount of mRNA available for translation, thereby influencing gene expression. Secondly, translational control, another essential process in { }, involves the regulation of mRNA translation into proteins. This step ensures that the correct amount and type of proteins are produced. Additionally, mRNA-binding proteins play a vital role in { } by binding to mRNA and regulating its stability, transport, and translation. Therefore, the interplay between mRNA turnover, translational control, and mRNA-binding proteins ensures a finely tuned and dynamic regulation of gene expression.

Ground truth:
Post-transcriptional control

Checklist:
1. Accurately defines post-transcriptional control as a critical aspect of gene expression regulation that occurs after RNA has been transcribed.
2. Clearly describes the key steps involved in post-transcriptional control, including mRNA processing, splicing, export, turnover, and translational control.
3. Correctly explains mRNA turnover as a crucial component of post-transcriptional control, involving the degradation of mRNA molecules to regulate the amount of mRNA available for translation.
4. Accurately describes translational control as another essential process in post-transcriptional control, involving the regulation of mRNA translation into proteins to ensure the correct amount and type of proteins are produced.
5. Explains the role of mRNA-binding proteins in post-transcriptional control, including their function in regulating mRNA stability, transport, and translation.
6. Uses appropriate biological terminology and concepts to describe the processes involved in post-transcriptional control.
7. Avoids over-extrapolation or unfounded speculation beyond the scope of the given information.

Figure 5: The second case of K2V.

Question:
The { }, located in the lower medulla oblongata, plays a crucial role in processing sensory information from the lower half of the body, particularly for touch and proprioception. This nucleus receives input from larger fibers that ascend through the posterior and posterolateral columns of the spinal cord on the same side. Once these fibers reach the { }, they synapse, and the axons of the second-order neurons then arise from this nucleus. These axons decussate, or cross over, to the contralateral side and continue their journey to the thalamus, forming part of the lemniscal pathway. Therefore, the sensory information from the lower body is effectively transmitted to higher brain centers for further processing.

Ground truth:
gracile nucleus

Checklist:
1. Correctly identifies the gracile nucleus as the structure located in the lower medulla oblongata.
2. Accurately describes the role of the gracile nucleus in processing sensory information, specifically for touch and proprioception.
3. Correctly states that the gracile nucleus receives input from larger fibers ascending through the posterior and posterolateral columns of the spinal cord on the same side.
4. Clearly explains that once these fibers reach the gracile nucleus, they synapse, and the axons of the second-order neurons arise from this nucleus.
5. Correctly describes the decussation (crossing over) of the axons of the second-order neurons to the contralateral side.
6. Accurately states that the axons of the second-order neurons continue their journey to the thalamus, forming part of the lemniscal pathway.
7. Avoids irrelevant information and focuses on answering the question directly.

Figure 6: The third case of K2V.

Question 1:
What is the name of the gene that controls the sheath purple color ?
Ground truth 1:
P urple Sh eath1, PSH1

Question 2:
Before being arrested, who had the right to hire a lawyer ?
Ground truth 2:
Accused, Defendant, Family member

Question 3:
What are some of the symptoms of subacute myopathy ?
Ground truth 3:
myalgias, muscle weakness

Figure 7: Three cases of Liquid. The data synthesized by Liquid contains multiple candidate answers, which are separated by commas in the figure.

Question 1:
In which tissue did OsIspE show the highest expression ?
Ground truth 1:
leaf blades

Question 2:
What are the criminal subjects of credit card fraud ?
Ground truth 2:
Generally, natural persons can be the criminal subjects of this crime

Question 3:
What are organisms called that derive their energy directly from sunlight ?
Ground truth 3:
phototrophic

Figure 8: Three cases of Genie.

Question 1:
A point mutation in OsSPL14 on the OsmiR156-targeted site leads to OsSPL14 escape from the cleavage by { }. This results in a transgenic rice with increased grain yield.
Ground truth 1:
OsmiR156

Question 2:
A decrease in the ability to smell, known as { }, can occur after a cold or flu. This sensory dysfunction is often linked to changes in the nasal mucous membranes.
Ground truth 2:
anosmia

Question 3:
Xiao Li applied for an extension of the litigation period in the lawsuit, and it needed to be approved by { }.
Ground truth 3:
Court

Figure 9: Three cases of SDR.

Question 1:

A local farmer in the city of Suihua is interested in enhancing the nitrogen fixation process in his rice fields to improve crop yield. He has heard about the role of certain enzymes in plants and animals that can influence this process. Considering the farmer's goal and the local agricultural practices, which of the following enzymes or factors should he investigate further for its potential indirect impact on his rice crops?

A. Rat Neuronal NO Synthase (NNOS)
B. Soybean Nodulation Factor
C. Corn Ethylene Synthase
D. Wheat Glutamine Synthetase

Ground truth 1:

A

Question 2:

A farmer is transporting a group of tourists on a boat to visit a remote island where he has a farm. One of the tourists, Mr. Zhang, has brought his own luggage on board. According to the Maritime Commercial Law, the farmer, as the carrier, has certain responsibilities.

A. The farmer is responsible for ensuring that Mr. Zhang's luggage is securely stored on the boat.
B. The farmer is not responsible for any damage to Mr. Zhang's luggage during the journey.
C. The farmer must provide insurance for Mr. Zhang's luggage.
D. The farmer is only responsible for Mr. Zhang's luggage if it is lost.

Ground truth 2:

A

Question 3:

A man notices that several of his cattle are showing signs of lethargy and reduced feed intake. The veterinarian suspects a liver condition and recommends conducting biochemical tests to confirm the diagnosis. If the tests confirm liver disease, which of the following gastrointestinal (GI) issues might also be observed in the affected cattle?

A. Increased rumination
B. Diarrhea
C. Increased appetite
D. Weight gain

Ground truth 3:

B

Figure 10: Three cases of BDS.

I Prompts

K2V utilizes auxiliary LLMs for data synthesis and model training. In this section, we provide the detailed prompts.

I.1 Prompt of $\mathcal{F}_{\text{text}}$

The prompt of the $\mathcal{F}_{\text{text}}$ is as shown in Table 13. This prompt is used to instructs an LLM to convert the masked quintuple into a fill-blank style QA pair.

I.2 Prompt for NER and RE

The prompt for NER and RE is as shown in Table 14. This prompt is used to instruct an LLM to extract entities and relations from the corpus.

I.3 Prompt of Judge Model

The prompt of the Judge Model is as shown in Table 15. This prompt is used to instruct an LLM to verify the reasoning process based on a checklist.

I.4 Prompt for Synthesizing Checklist

The prompt for synthesizing the checklist is as shown in Table 16.

Prompt of the $\mathcal{F}_{\text{text}}$

−Role−

You are an NLP expert responsible for generating a logically structured and coherent rephrased version of the TEXT based on ENTITIES and RELATIONSHIPS provided below. Use English as output language.

−Goal−

To generate a version of the text that is rephrased and conveys the same meaning as the original entity and relationship descriptions, while:

1. Following a clear logical flow and structure
2. Establishing proper cause-and-effect relationships
3. Ensuring temporal and sequential consistency
4. Creating smooth transitions between ideas using conjunctions and appropriate linking words like 'firstly,' 'however,' 'therefore,' etc.

−Instructions−

1. Analyze the provided ENTITIES and RELATIONSHIPS carefully to identify:
   - Key concepts and their hierarchies
   - Temporal sequences and chronological order
   - Cause-and-effect relationships
   - Dependencies between different elements
2. Organize the information in a logical sequence by:
   - Starting with foundational concepts
   - Building up to more complex relationships
   - Grouping related ideas together
   - Creating clear transitions between sections
3. Rephrase the text while maintaining:
   - Logical flow and progression
   - Clear connections between ideas
   - Proper context and background
   - Coherent narrative structure
4. Review and refine the text to ensure:
   - Logical consistency throughout
   - Clear cause-and-effect relationships

################
-ENTITIES-
################
{entities}

################
-RELATIONSHIPS-
################
{relationships}

Table 13: Prompt of the $\mathcal{F}_{\text{text}}$. This prompt is used to instructs an LLM to convert the masked quintuple into a fill-blank style QA pair.

Prompt for NER and RE

You are an NLP expert, skilled at analyzing text to extract named entities and their relationships.

−Goal−

Given a text document that is potentially relevant to this activity and a list of entity types, identify all entities of those types from the text and all relationships among the identified entities. Use English as output language.

−Steps−

1. Identify all entities. For each identified entity, extract the following information:
   - entity_name: Name of the entity, use same language as input text. If English, capitalized the name.
   - entity_type: One of the following types: concept, date, location, keyword, organization, person, event, work, nature, artificial, science, technology, mission, gene
   - entity_summary: Comprehensive summary of the entity's attributes and activities
   - Format each entity as:
     `("entity"<|><entity_name><|><entity_type><|><entity_summary>)`
2. From the entities identified in step 1, identify all pairs of (source_entity, target_entity) that are *clearly related* to each other. For each pair of related entities, extract the following information:
   - source_entity: name of the source entity, as identified in step 1
   - target_entity: name of the target entity, as identified in step 1
   - relationship_summary: explanation as to why you think the source entity and the target entity are related to each other
   - Format each relationship as:
     `("relationship"<|><source_entity><|><target_entity><|> <relationship_summary>)`
3. Identify high-level key words that summarize the main concepts, themes, or topics of the entire text. These should capture the overarching ideas present in the document.Format the content-level key words as `("content_keywords"<|><high_level_keywords>)`
4. Return output in Englist as a single list of all the entities and relationships identified. Use **##** as the list delimiter.
5. When finished, output `<|COMPLETE|>`

################
-Input Text-
################
{input_text}

Table 14: Prompt for NER and RE. This prompt is used to instruct an LLM to extract entities and relations from the corpus.

Prompt of the Judge Model

You are an impartial and meticulous AI examiner.

Your task is to evaluate a student's [Reasoning Process] for a given [Question-Answer Pair] against a specific, detailed [Criterion].

The [Question-Answer Pair] is a fill-in-the-blank question, with `"{ }"` indicating the content to be filled in. A fill-in-the-blank question may contain multiple `"{ }"`, and the content to be filled in for each `"{ }"` is the same.

Your judgment must be strict, objective, and based solely on the provided information.

NOTE: Your output can only be `"yes"` or `"NO"`

[Question-Answer Pair]
question: *question*
answer: *answer*

[Criterion]
criterion: *criterion*

[Reasoning Process]
reasoning process: *reasoning process*

Table 15: Prompt of the Judge Model. This prompt is used to instruct an LLM to verify the reasoning process based on a checklist.

Prompt for synthesizing the checklist

You are a senior expert in agriculture and biology, specializing in creating and grading exam questions. Your task is to create a set of detailed scoring checklist for a [Specific Question] based on the provided [General Criteria].

[Specific Question]:

A complete question in the field of agriculture and biology, including the question and the corresponding answer.

[General Criteria]:

Concepts and Knowledge:

1. Accurately defines the core biological concepts involved in the question.
2. Clearly describes the involved biological processes in the correct logical sequence.
3. Accurately explains the meaning and relationships represented by abstract biological models in words.
4. Applies abstract biological concepts to the given specific scenario.
5. Correctly explains the connection between a biological concept or process and other related principles.

Scientific Method and Design:

1. Clearly states a relevant null hypothesis or alternative hypothesis.
2. Accurately identifies the independent, dependent, and key control variables of an experiment.
3. Makes a logical and reasonable prediction of the experimental outcome based on a scientific hypothesis.
4. Evaluates the validity or potential flaws of a given experimental design.

Statistics and Evaluation:

1. In appropriate contexts, correctly uses statistical concepts to explain the reliability of data.
2. Based on data analysis, draws a conclusion of "support," "refute," or "inconclusive" for a given scientific hypothesis.
3. Explains outliers or anomalous data points and analyzes their potential causes or impact on the conclusion.

Argumentation and Reasoning:

1. Makes a scientific claim that is specific and supported by concrete evidence.
2. Clearly articulates how the evidence supports the scientific claim, demonstrating a strong logical chain.
3. Predicts the likely consequences of a change (e.g., disturbance, mutation) to a system based on biological principles.
4. Explains the underlying biological reason for an observed phenomenon or experimental result.
5. Avoids over-extrapolation or unfounded speculation beyond the scope of the given evidence.
6. The overall response is well-structured, logically coherent, and clearly written, avoiding self-contradictions and redundant statements.

Based on the [General Criteria] above, design a set of detailed and objectively scorable checklist for the provided [Specific Exam Question]. The checklist will be used to evaluate the student's problem-solving approach (reasoning process). The checklist should consist of multiple independent criteria. Each criteria must be a clear, specific statement describing what an ideal step or thought process should achieve, making it objectively assessable. Please ensure The checklist are closely related to the core knowledge and skill requirements of the [Specific Exam Question]. Only output the checklist, with no other content. Please structure the output in JSON format. For example:
`["criteria 1", "criteria 2",]`

Table 16: Prompt for synthesizing the checklist.

J Examples of General Criteria

K2V synthesizes question-specific checklist based on general criteria, which reflect a high-quality reasoning process and are independent of any specific problem. We develop general criteria for three domains: agriculture (Table 17), medicine (Table 18), and law (Table 19). It is worth noting that although different domains require unique general criteria, adapting them to a new domain usually only requires simple keyword substitution based on an existing one.

General criteria in the agricultural domain

Concepts and Knowledge:

1. Accurately defines the core biological concepts involved in the question.
2. Clearly describes the involved biological processes in the correct logical sequence.
3. Accurately explains the meaning and relationships represented by abstract biological models in words.
4. Applies abstract biological concepts to the given specific scenario.
5. Correctly explains the connection between a biological concept or process and other related principles.

Scientific Method and Design:

1. Clearly states a relevant null hypothesis or alternative hypothesis.
2. Accurately identifies the independent, dependent, and key control variables of an experiment.
3. Makes a logical and reasonable prediction of the experimental outcome based on a scientific hypothesis.
4. Evaluates the validity or potential flaws of a given experimental design.

Data Processing and Analysis:

1. Accurately and correctly extracts key data points.
2. Clearly and comprehensively describes the overall trend or significant patterns in the given data.
3. Accurately describes the relationship between variables (e.g., positive correlation, negative correlation, no correlation).
4. Correctly performs necessary mathematical calculations (e.g., rate, rate of change, percentage) to support the analysis.

Statistics and Evaluation:

1. In appropriate contexts, correctly uses statistical concepts to explain the reliability of data.
2. Based on data analysis, draws a conclusion of "support," "refute," or "inconclusive" for a given scientific hypothesis.
3. Explains outliers or anomalous data points and analyzes their potential causes or impact on the conclusion.

Argumentation and Reasoning:

1. Makes a scientific claim that is specific and supported by concrete evidence.
2. Clearly articulates how the evidence supports the scientific claim, demonstrating a strong logical chain.
3. Predicts the likely consequences of a change (e.g., disturbance, mutation) to a system based on biological principles.
4. Explains the underlying biological reason for an observed phenomenon or experimental result.
5. Avoids over-extrapolation or unfounded speculation beyond the scope of the given evidence.
6. The overall response is well-structured, logically coherent, and clearly written, avoiding self-contradictions and redundant statements.

Table 17: General criteria in the agricultural domain.

General criteria in the medical domain

Concepts and Knowledge:

1. Accurately defines the core medical concepts involved in the question.
2. Clearly describes the involved medical processes in the correct logical sequence.
3. Accurately explains the meaning and relationships represented by abstract biological or medical models in words.
4. Applies abstract biological or medical concepts to the given specific scenario.
5. Correctly explains the connection between a medical concept or process and other related principles.

Scientific Method and Design:

1. Clearly states a relevant null hypothesis or alternative hypothesis.
2. Accurately identifies the independent, dependent, and key control variables of an experiment.
3. Makes a logical and reasonable prediction of the experimental outcome based on a scientific hypothesis.
4. Evaluates the validity or potential flaws of a given experimental design.

Data Processing and Analysis:

1. Accurately and correctly extracts key data points.
2. Clearly and comprehensively describes the overall trend or significant patterns in the given data.
3. Accurately describes the relationship between variables (e.g., positive correlation, negative correlation, no correlation).
4. Correctly performs necessary mathematical calculations (e.g., rate, rate of change, percentage) to support the analysis.

Statistics and Evaluation:

1. In appropriate contexts, correctly uses statistical concepts to explain the reliability of data.
2. Based on data analysis, draws a conclusion of "support," "refute," or "inconclusive" for a given scientific hypothesis.
3. Explains outliers or anomalous data points and analyzes their potential causes or impact on the conclusion.

Argumentation and Reasoning:

1. Makes a scientific claim that is specific and supported by concrete evidence.
2. Clearly articulates how the evidence supports the scientific claim, demonstrating a strong logical chain.
3. Predicts the likely consequences of a change (e.g., disturbance, mutation) to a system based on biological or medical principles.
4. Explains the underlying biological or medical reason for an observed phenomenon or experimental result.
5. Avoids over-extrapolation or unfounded speculation beyond the scope of the given evidence.
6. Based on diagnostic or analytical results, proposes specific and feasible treatment or management recommendations that comply with clinical guidelines and ethical principles.
7. Clearly articulates the rationale for the proposed recommendations and weighs their potential benefits and risks.
8. Be able to ignore irrelevant information and focus on answering the question directly.
9. The overall response is well-structured, logically coherent, and clearly written, avoiding self-contradictions and redundant statements.

Table 18: General criteria in the medical domain

General criteria in the legal domain

Fact and Issue Identification:

1. Accurately identifies and extracts key legally relevant facts from the case material.
2. Clearly and accurately identifies the core legal issues or points of contention presented in the case.
3. Is able to distinguish between legally relevant and irrelevant facts. II. Rule Statement and Interpretation

Rule Statement and Interpretation:

1. Accurately states the legal rules (including statutes, judicial interpretations, fundamental principles, etc.) relevant to the identified issues.
2. Correctly explains the meaning and constituent elements of the legal rules.
3. Where appropriate, is able to articulate the legislative intent, value orientation, or legal theory behind the relevant rules.

Application and Analysis:

1. Effectively connects the identified key facts to the relevant legal rules (i.e., the process of "subsumption").
2. Logically analyzes whether the facts of the case satisfy (or fail to satisfy) the constituent elements of the legal rules.
3. Is able to analyze and argue from the perspectives of all involved parties (e.g., plaintiff/defendant, prosecution/defense).
4. Is able to anticipate and respond to potential and compelling counterarguments or defenses.
5. When dealing with complex problems, is able to conduct a layered, step-by-step analysis of different claims or legal relationships.

Conclusion and Consequences:

1. Based on the preceding analysis, draws a clear, reasonable, and persuasive conclusion for each issue.
2. Is able to articulate the specific legal consequences corresponding to the conclusion (e.g., type and scope of civil liability, determination of criminal responsibility).
3. Proposes solutions or legal advice that are specific, feasible, and in compliance with legal regulations and professional ethics.

Overall Structure and Expression:

1. The overall response is clearly structured and logically coherent (e.g., follows a framework like IRAC: Issue, Rule, Application, Conclusion).
2. Uses legal terminology accurately and appropriately.
3. The reasoning process is rigorous, avoiding over-extrapolation or speculation not supported by the given facts or law.
4. Is able to ignore irrelevant information and focus on answering the question directly.
5. The overall response is well-written, clear, and avoids self-contradictions or unnecessary redundancy.
6. The overall response is clearly written, avoiding self-contradictions and redundant statements.

Table 19: General criteria in the legal domain

K The Use of LLMs

We use OpenAI's GPT-4 as a writing assistant to help improve the clarity, grammar, and style of this manuscript. All scientific ideas, experiments, and conclusions were conceived and executed by the human authors.