

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 KNOWLEDGE-TO-VERIFICATION: UNLOCKING REINFORCEMENT LEARNING WITH VERIFIABLE REWARDS FOR LLMS IN KNOWLEDGE-INTENSIVE DOMAINS

Anonymous authors

Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) has demonstrated promising potential to enhance the reasoning capabilities of large language models in domains such as mathematics and coding. However, its application has not been effectively extended to knowledge-intensive domains due to the issue of unverifiable answers and the Limited of high-quality verifiable data. Furthermore, existing RLVR methods suffer from two inherent drawbacks. First, they focus solely on the final answer’s correctness while ignoring the verification of reasoning process, which can lead to flawed reasoning. Second, this reliance on a final answer creates a sparse binary reward signal that destabilizes the training process. To address these challenges, we propose **Knowledge-to-Verification (K2V)**, a framework that extends RLVR to knowledge-intensive domains and enabling verification of the model’s reasoning process, without any human supervision. K2V is built on two key observations. First, structured knowledge is easier to verify than unstructured knowledge. Second, a complex reasoning process can be decomposed into a series of verifiable sub-tasks. Specifically, K2V first constructs a knowledge graph from text and then models the knowledge verification as a text-based knowledge graph completion task, thereby automatically synthesizing large-scale verifiable question-answering (QA) pairs. Then, K2V generates a checklist of sub-tasks for each QA pair. The model’s reasoning process is verified by evaluating these sub-tasks, which in turn provides dense rewards. Extensive experiments demonstrate that K2V enhances the model’s fundamental reasoning skills, which improves its reasoning capabilities in knowledge intensive domains while keeping its capabilities in general domains stable, or even slightly improved. Our work suggests that extending RLVR to knowledge-intensive domains through automated data synthesis is a promising direction. Meanwhile, verifying the reasoning process proves to be an effective method for overcoming the inherent drawbacks of RLVR. The code is available at <https://anonymous.4open.science/r/k2v-C123>.

1 INTRODUCTION

Recent large language models (LLMs), such as OpenAI-o1 (Jaech 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 (Hu et al., 2025; Zeng et al., 2025; Liu et al., 2025) and coding tasks (He et al., 2025; Luo et al., 2025; Cui et al., 2025), lack the transferability to knowledge-intensive domains (*e.g.* agriculture, medicine, *etc.*). This narrow focus can be attributed to two main reasons (1) Unverifiable Answers: In mathematics, the correctness of a model’s response to certain problems, such as computation or multiple-choice questions, can be directly validated by a rule-based verifier. (Hu et al., 2025). As for the domain of code, unit tests can be directly executed on model-generated code within a sandboxed environment (Luo et al., 2025). However, in knowledge-intensive domains, answers are typically available only in

054 a free-form format. These answers exhibit high diversity and complexity, which means they cannot
 055 be directly validated by rule-based verifiers. (2) Limited Data: From a data source perspective, the
 056 internet contains large-scale verifiable math problems and executable open-source code (Ma et al.,
 057 2025). In contrast, data scraped for knowledge-intensive domains is predominantly unverifiable text
 058 or of low quality. From a data synthesis perspective, because problems in mathematics and code
 059 are easier to verify, synthesizing new verifiable problems is also more straightforward (Yang et al.,
 060 2025a). However, knowledge-intensive domains lack effective data synthesis methods and rely on
 061 costly, expert-level manual annotation (Dubois et al., 2024).

062 Moreover, current RLVR methods suffer from two inherent drawbacks across various domains (1)
 063 Flawed Reasoning: These methods focus solely on the correctness of the final answer, ignoring the
 064 validity of the reasoning process. This reward mechanism can lead the LLMs' reasoning to exhibit
 065 issues like linguistic incoherence (Guo et al., 2025a) and unfaithful reasoning (Chen et al., 2025a).
 066 (2) Sparse Rewards: Awarding a binary reward based only on the final answer creates an overly
 067 sparse reward signal, which increases the variance of the policy gradient estimate, introduces noise
 068 to the training, and results in unstable learning and slower convergence (Su et al., 2025).

069 In this paper, we investigate how to advance current RLVR methods by addressing their fundamental
 070 limitations. We formulate our research around two central questions:

- 072 • **RQ1:** How can we address the issue of unverifiable answers, thereby resolving the limited
 073 data and ultimately extending the RLVR to knowledge-intensive domains ?
- 074 • **RQ2:** How can we mitigate the issue of flawed reasoning, thereby resolving the sparse
 075 rewards and ultimately overcome the inherent drawbacks of current RLVR ?

077 To answer these questions, we propose **Knowledge-to-Verification (K2V)**, which synthesizes veri-
 078 fiable question-answering (QA) pairs in knowledge-intensive domains without requiring any human
 079 annotation or seed datasets (Wang et al., 2022; Yu et al., 2025a), while also enabling the verification
 080 of the LLMs' reasoning process. *The core observation of K2V is that knowledge in a structured rep-
 081 resentation is more easily verifiable than unstructured knowledge.* (e.g. knowledge within a dense,
 082 unstructured text becomes more easily verifiable when organized into a table). To this end, K2V
 083 begins by structuring knowledge from the corpus into a Knowledge Graph (KG). It then models
 084 knowledge verification as a text-based Knowledge Graph Completion (KGC) task (Hogan et al.,
 085 2021; Yao et al., 2025), effectively converting knowledge from large-scale unstructured texts into
 086 verifiable QA pairs. Furthermore, *we posit that directly verifying the correctness of an LLM's rea-
 087 soning process is challenging. However, this task can be decomposed into multiple, binary-verifiable
 088 sub-tasks.* (e.g. when solving a physics problem, the reasoning process can be verified by checking
 089 sub-tasks such as whether the correct formula was chosen and whether the correct values were sub-
 090 stituted). Specifically, we introduce a checklist-style verification method that generates a checklist
 091 for each QA pair. This checklist consists of several independent, human-interpretable, and binary-
 092 verifiable sub-tasks that describe the criteria for a correct reasoning process.

093 Experiments across three distinct domains demonstrate that K2V enhances reasoning capabilities
 094 in knowledge-intensive domains without compromising, and even improving, the model's general
 095 capabilities. In the agriculture domain, leveraging Qwen2.5-3B (Qwen et al., 2025) as backbone,
 096 K2V achieves 63.04 on SeedBench (Ying et al., 2025), outperforming the much larger Qwen2.5-
 097 72B-Instruct (62.06). Notably, without using any general data, K2V's general performance is on
 098 par with the official instruction-tuned model Qwen2.5-3B-Instruct, even achieving a increase of
 099 23.99% on BBH (Suzgun et al., 2022) and 5.24% on GPQA-Diamond (Rein et al., 2024). Further
 100 experiments based on Llama3.1 (Dubey et al., 2024) and Llama3.2 (Meta, 2024) confirm the robust
 101 performance of K2V.

102 The contributions of this work can be summarized as below. We propose K2V, a framework that ef-
 103 ficiently extends RLVR to knowledge-intensive domains. In K2V, we introduce two novel designs.
 104 First, we model knowledge verification as a text-based KGC task, which enables the cost-effective
 105 synthesis of verifiable QA pairs. Second, we employ a checklist-style verification method to vali-
 106 date the model's reasoning process, which provides a dense reward signal. Extensive experiments
 107 demonstrate that K2V enhances the model's fundamental reasoning skills, which in turn signifi-
 108 cantly improves its capabilities in knowledge-intensive domains while keeping general capabilities
 109 stable or even slightly improved.

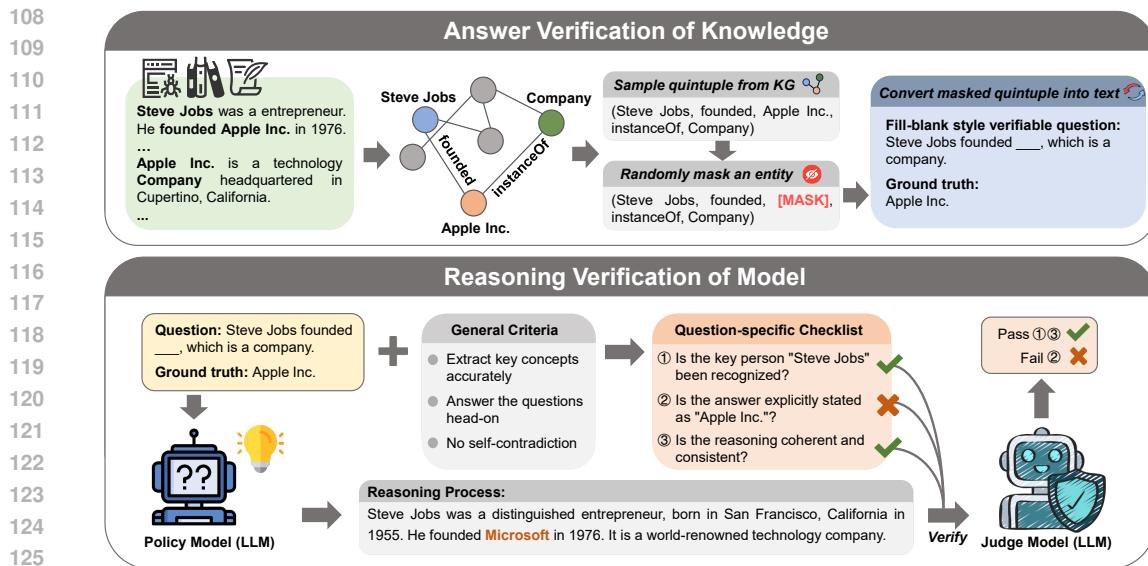


Figure 1: Overview of K2V. **(Top)** K2V begins by constructing a KG from unstructured text. It then samples quintuples from the KG and randomly masks an entity. This masked quintuple is then converted into a fill-blank style verifiable question. **(Bottom)** Given a question, Policy Model generates a reasoning process. To verify this reasoning process, K2V generates a Question-Specific Checklist by instantiating a set of General Criteria based on the given question and its ground truth. A Judge Model then verifies the reasoning process against each item in the checklist one by one.

2 KNOWLEDGE-TO-VERIFICATION

In this section, we first introduce how to synthesize QA pairs with verifiable answers in knowledge-intensive domains (Section 2.1). We then present checklist-style verification, which not only validates the correctness of the model’s reasoning process but also provides dense rewards (Section 2.2). Finally, we discuss how to combine the two distinct rewards derived from answer verification and reasoning verification (Section 2.3). An overview of K2V is shown in Figure 1.

2.1 ANSWER VERIFICATION OF KNOWLEDGE

Knowledge-intensive domains have large-scale knowledge-rich, unstructured corpus that can be easily and inexpensively crawled from the internet in a scalable manner. The goal of K2V is to convert the knowledge within unstructured corpus into verifiable QA pairs. We will begin by explaining the fundamental concepts of text-based Knowledge Graph Completion (KGC), as introduced below.

Text-based KGC. 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 factual triples. Each triple $(h, r, t) \subseteq \mathcal{T}$ 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 \mathcal{E} (Ji et al., 2021).

Different from the conventional KGC, **K2V reformulates the task as text-based KGC, where knowledge verification is represented as fill-blank style verifiable QA pairs.** Meanwhile, to ensure the generated questions possess sufficient context for complex reasoning, K2V operates not on individual triples, but on quintuples. A quintuple τ is defined as:

$$\tau = (e_1, r_1, e_2, r_2, e_3) \quad (1)$$

which represents two consecutive triples, (e_1, r_1, e_2) and (e_2, r_2, e_3) , forming a path of length two in the KG. This structure provides a richer context than a single triple, making the generated questions more diverse and the reasoning task more challenging and realistic.

To generate a QA pair, we first randomly mask one of the three entities (e_1 , e_2 , or e_3) to create a masked quintuple τ_{masked} , e.g., $(e_1, r_1, [\text{MASK}], r_2, e_3)$. We then employ a textualization function, $\mathcal{F}_{\text{text}}$, powered by a generative LLM, to convert the masked quintuple into a fill-blank question.

$$Q_{\text{blank}} = \mathcal{F}_{\text{text}}(\tau_{\text{masked}}) \quad (2)$$

The ground truth for this fill-blank question Q_{blank} is the name of the masked entity. The prompt of $\mathcal{F}_{\text{text}}$ is shown in Table 7 in Appendix.

QA Pairs Synthesis. Text-based KGC provides a pathway to convert quintuples into fill-blank style verifiable QA pairs. Consequently, once a KG is constructed in the knowledge-intensive domains (Chen et al., 2025b), we can synthesize numerous verifiable QA pairs at a low cost by sampling quintuples from the KG.

First, K2V constructs a KG from a diverse, unstructured corpus crawled from the internet, which includes web pages, scientific literature, and textbooks. To convert this raw text into KG, K2V leverages a generative LLM to perform Named Entity Recognition (NER) (Keraghel et al., 2024) and Relation Extraction (RE) (Zhao et al., 2024) on the chunks from the corpus (see Table 8 in Appendix for the prompt). Entities from different chunks are then linked to construct the final KG.

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

K2V operates entirely from scratch, requiring no human annotation or seed datasets (Wang et al., 2022; Yu et al., 2025a). This approach is both efficient and scalable: the more unstructured corpus are available, the more verifiable QA pairs can be synthesized (**RQ1**).

2.2 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 goal of K2V is to verify the correctness of the reasoning process, thereby providing a dense reward. In particular, we first discuss how checklist-style verification decomposes a complex reasoning verification task into multiple, binary-verifiable sub-tasks, as introduced below.

Checklist-Style Verification. The reasoning process z of a policy model is typically a lengthy, open-ended text that describes the policy’s thinking details for a given question x . Due to the lack of evaluation criteria, directly verifying the correctness of z is challenging. Motivated by this, we propose checklist-style verification. For each question x , we establish a checklist, which is formally represented as a set of k verifiable criteria:

$$CL_x = \{c_1, c_2, \dots, c_k\} \quad (3)$$

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 CL_x for a comprehensive assessment of reasoning quality. Most importantly, **the checklist CL_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 CL_x one by one (see Table 10 in Appendix 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:

$$v_i = J(z, c_i) \quad (4)$$

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:

$$p = \frac{1}{k} \sum_{i=1}^k v_i \quad (5)$$

216 where v_i is computed by Equation 4. This approach decomposes the intractable reasoning verification
 217 task into a series of binary-verifiable sub-tasks. Additionally, the resulting pass rate p can serve
 218 as a dense reward signal (**RQ2**).
 219

220 **Checklist Synthesis.** Checklist-style verification enables us to validate the model’s reasoning process
 221 while providing a dense reward signal. The subsequent challenge is to synthesize a problem-
 222 specific checklist CL_x (see Equation 3) for any given prompt x . We propose a two-stage synthesis
 223 pipeline, as introduced below.

224 First, we define a set of general criteria, formally denoted as:
 225

$$G = \{g_1, g_2, \dots, g_N\} \quad (6)$$

226 where each g_i is a high-level, universal principle that characterizes a high-quality reasoning process,
 227 independent of any specific question. We developed a set of general criteria based on the scoring
 228 rubrics from the AP Biology Course and Exam Description¹. An example is shown in Table 5.
 229

230 Second, for a given QA pair consisting of a question x and a ground truth y , we feed both x, y
 231 and the set of general criteria G into a generative LLM, denoted as S . This LLM is prompted to
 232 instantiate the abstract principles in G into a concrete, question-specific checklist CL_x suitable for
 233 verifying reasoning related to (see Table 9 in Appendix for the prompt). This synthesis process can
 234 be expressed as:
 235

$$CL_x = \{c_1, \dots, c_k\} \sim S(\cdot|x, y, G) \quad (7)$$

236 This automated pipeline enables K2V to generate tailored checklists for a wide variety of questions,
 237 ensuring that our verification is both scalable and contextually appropriate.
 238

240 2.3 REWARD COMPOSITION

241 In the preceding sections, we introduced two distinct verification mechanisms: answer verification
 242 for validating the correctness of knowledge, and reasoning verification for assessing the quality of
 243 the reasoning process. Following DeepSeek-R1 (Guo et al., 2025a), we form the training signal by
 244 directly summing the individual reward components.
 245

$$R_{\text{total}} = R_{\text{format}} + R_{\text{answer}} + R_{\text{reasoning}} \quad (8)$$

246 **Format Reward (R_{format}).** The format reward, R_{format} , encourages the policy model to generate
 247 outputs in a structured and parsable format. The desired output template is consistent with that of
 248 DeepSeek-R1 (Guo et al., 2025a). A maximum score of 0.75 is awarded if the output perfectly
 249 adheres to this template.
 250

251 **Answer Reward (R_{answer}).** The answer reward, R_{answer} , provides a binary reward based on the final
 252 answer’s correctness. If the model’s predicted answer \hat{y} exactly matches the ground truth y , a reward
 253 of α is granted; otherwise, the reward is 0. In our main experiments, we set $\alpha = 6$. We also conduct a
 254 sensitivity analysis on the value of α to investigate how different answer reward values affect model
 255 performance, as detailed in Section 3.4.
 256

257 **Reasoning Reward ($R_{\text{reasoning}}$).** The reasoning reward, $R_{\text{reasoning}}$, is a dense signal derived from our
 258 checklist-style verification (Section 2.2). This reward encourages the model to produce logically
 259 coherent reasoning processes, and is set directly to the pass rate p calculated in Equation 5.
 260

262 3 EXPERIMENTS

264 In this section, we empirically investigate the effectiveness of K2V. Our analysis includes not only
 265 an evaluation of the overall model performance but also ablation studies of its components and a
 266 sensitivity analysis to varying reward values.
 267

268 ¹The AP Biology Course and Exam is a program run by the U.S. College Board that provides a college-level
 269 introductory biology curriculum to high school students. The scoring rubrics for this exam can be found in:
<https://apcentral.collegeboard.org/media/pdf/ap-biology-course-and-exam-description.pdf>

270 3.1 EXPERIMENTAL SETUPS
271272 **Backbone Models.** We use Qwen2.5 (Qwen et al., 2025), Llama3.1 (Dubey et al., 2024), and
273 Llama3.2 (Meta, 2024) series models as our backbone. For the Qwen series, our K2V-3B and K2V-
274 7B models are built upon Qwen2.5-3B-Base and Qwen2.5-7B-Base, respectively, while all other
275 baselines are implemented on Qwen2.5-3B-Base. For the Llama series, K2V-3B and K2V-8B are
276 based on Llama-3.2-3B-Instruct and Llama-3.1-8B-Instruct, respectively.277 **Training Corpus.** K2V’s training begins with an unstructured text corpus. We trained separate
278 models in the domains of agriculture, medicine and law. For agriculture, we use RiceCorpus (Yang
279 et al., 2025b), a corpus built by crawling extensive resources from the internet, including 1.4 mil-
280 lion papers published over the past 40 years and 1,207 related books, totaling 3,397.49 GB. This
281 data covers multiple disciplines such as molecular biology, plant breeding, and agronomic prac-
282 tices. The raw corpus was processed through a meticulously designed cleaning pipeline, yielding
283 a high-quality corpus of approximately 1.1B tokens. For medicine, we use shibing624-medical-
284 pretrain (shibing624, 2024), a Chinese corpus sourced from online medical encyclopedias and text-
285 books. We randomly downsample 23.51 MB of text from this corpus for training. For law, we use
286 DISC-Law-SFT (Yue et al., 2023), a dataset of QA pairs for supervised fine-tuning (SFT). We di-
287 rectly concatenate the questions and answers, then randomly downsample 20.69 MB of the text as
288 our training corpus. This indirectly demonstrates K2V’s robustness to varied corpus sources.289 **Evaluation.** In addition to evaluations in agriculture, medicine, and law, we also tested the model’s
290 performance on general benchmarks. All evaluations are conducted in a zero-shot setting, and we
291 report the pass@1 results under greedy sampling. For agriculture, to ensure an objective and fair
292 evaluation, we evaluate on the multiple-choice subset of SeedBench (Ying et al., 2025) and conduct
293 experiments on agriculture-related subsets of CMMLU(Li et al., 2023) and MMLU (Hendrycks
294 et al., 2020). For medicine, we evaluate on MedQA-MCMLE (Jin et al., 2021), as well as on
295 medicine-related subsets of CMMLU and MMLU. For law, we evaluate on LawBench (Fei et al.,
296 2023), as well as on law-related subsets of CMMLU and MMLU. The details of subset selection
297 for all three domains are provided in the Appendix C.3.Finally, for general benchmarks, we eval-
298 uate on BBH (Suzgun et al., 2022), MATH-500 (Hendrycks et al., 2021; Lightman et al., 2023b),
299 GSM8K (Cobbe et al., 2021), AIME2024 (MAA, 2024), and GPQA-Diamond (Rein et al., 2024).300 **Baselines.** We compare our method against the following robust baselines: (1) Base and Instruction-
301 Tuned Models: We include models from the Qwen2.5, as well as several models from the Llama
302 series. (2) LIQUID (Lee et al., 2023): Extracts questions with multiple candidate answers from
303 unstructured corpus. (3) GENIE (Yehudai et al., 2024): Synthesizes QA pairs from unstructured
304 corpus. We slightly modify its prompt to enhance the verifiability of the synthesized data. (4)
305 Synthetic Data RL (Guo et al., 2025b): Uses a task-definition approach to synthesize verifiable
306 questions from unstructured corpus for RLVR. (5) BDS (Dedhia et al., 2025): Synthesizes verifiable
307 multiple-choice questions based on KG.308 3.2 MAIN RESULTS
309310 The main experimental results on knowledge-intensive domains are reported in Table 1 and Figure 2,
311 from which we observe several key advantages of K2V: **First**, K2V significantly enhances reason-
312 ing capabilities in knowledge-intensive domains, consistently outperforming strong baselines. K2V
313 successfully extends RLVR to specialized fields without requiring any human annotation. Our mod-
314 els surpass their respective base models and all four baselines across the agriculture, medical, and
315 legal domains. **Second**, K2V enables smaller models to achieve performance comparable to or even
316 exceeding that of much larger models. As shown in Figure 2, K2V-7B, built on the Qwen2.5-7B,
317 scores 65.90 on SeedBench, surpassing the Qwen2.5-72B-Instruct (62.06). Meanwhile, its per-
318 formance on the agriculture subsets of CMMLU and MMLU is very close to that of Qwen2.5-72B.
319 Most strikingly, the K2V-3B, built on the Qwen2.5-3B, also surpasses the Qwen2.5-72B-Instruct on
320 SeedBench. This demonstrates the remarkable data and training efficiency of our framework. **Third**,
321 K2V demonstrates robust generalizability across different model series. The consistent and signifi-
322 cant performance gains observed on both the Qwen and Llama model series confirm that our frame-
323 work is not limited to a specific model architecture. **Finally**, K2V’s effectiveness is proven across
324 multiple knowledge-intensive domains. The performance improvements are not confined to a single
325 area, K2V achieves strong results across all three tested domains: agriculture, medicine, and law.

Table 1: Overall performance on three different knowledge-intensive domains.

| Model | SeedBench | CMLLU | MMLU | MedQA-MCMLE | CMLLU | MMLU | LawBench | CMLLU | MMLU |
|-----------------------|--------------------|--------------|--------------|----------------|--------------|--------------|--------------|--------------|--------------|
| | Agriculture Domain | | | Medical Domain | | Legal Domain | | | |
| Qwen Models | | | | | | | | | |
| Qwen2.5-3B-Instruct | 46.04 | 63.22 | 75.54 | 71.10 | 62.45 | 70.69 | 42.60 | 62.34 | 64.03 |
| Qwen2.5-7B-Instruct | 49.14 | 75.13 | 85.43 | 76.67 | 76.76 | 79.30 | 54.86 | 75.21 | 69.93 |
| LIQUID | 53.23 | 54.02 | 69.14 | 70.15 | 58.10 | 59.20 | 38.49 | 60.57 | 64.96 |
| GENIE | 57.51 | 51.06 | 70.29 | 73.21 | 56.27 | 61.24 | 40.20 | 61.82 | 64.49 |
| Synthetic Data RL | 58.62 | 53.39 | 66.31 | 72.58 | 58.06 | 60.73 | 39.42 | 60.39 | 62.65 |
| BDS | 59.37 | 62.41 | 69.00 | 76.36 | 62.17 | 63.50 | 42.40 | 62.79 | 63.26 |
| K2V-3B (Ours) | 63.21 | 63.97 | 76.56 | 79.58 | 68.13 | 71.43 | 43.15 | 62.59 | 63.83 |
| K2V-7B (Ours) | 65.90 | 80.11 | 88.13 | 83.60 | 80.36 | 78.16 | 55.90 | 76.39 | 67.12 |
| Llama Models | | | | | | | | | |
| Llama-3.2-3B-Instruct | 47.03 | 29.56 | 62.06 | 61.00 | 30.76 | 59.56 | 30.94 | 33.79 | 46.25 |
| Llama-3.1-8B-Instruct | 58.25 | 51.72 | 74.05 | 70.15 | 49.16 | 71.21 | 38.81 | 50.82 | 64.44 |
| K2V-3B (Ours) | 57.15 | 47.59 | 72.10 | 65.20 | 39.52 | 60.54 | 36.24 | 35.72 | 47.62 |
| K2V-8B (Ours) | 62.85 | 55.20 | 83.30 | 73.13 | 55.28 | 70.19 | 42.13 | 54.08 | 65.80 |

Table 2: Performance on general domain. In this table, the backbones for K2V-3B and K2V-7B are Qwen2.5-3B and Qwen2.5-7B, respectively. Their training corpus consists exclusively of agricultural data, containing no general or mathematical data.

| Model | BBH | MATH-500 | GSM8K | AIME2024 | GPQA-Diamond |
|----------------------|--------------|--------------|--------------|--------------|--------------|
| Qwen2.5-3B-Instruct | 37.26 | 65.00 | 85.44 | 3.33 | 28.79 |
| K2V-3B (Ours) | 46.20 | 65.00 | 81.27 | 6.67 | 30.30 |
| Qwen2.5-7B-Instruct | 49.94 | 76.60 | 91.66 | 16.67 | 34.34 |
| K2V-7B (Ours) | 60.10 | 73.80 | 87.11 | 13.33 | 36.36 |

We also investigated whether domain-specific training with K2V would negatively impact general reasoning abilities. Surprisingly, we found that K2V not only preserved these skills but actually enhanced them. As shown in Table 2, Trained exclusively on the agriculture corpus without any general or mathematical data, our K2V models show marked improvements on general benchmarks. Specifically, K2V-3B substantially improves upon the official instruction-tuned Qwen2.5-3B-Instruct by 23.99% on BBH (46.20 vs. 37.26) and 5.24% on GPQA-Diamond (30.30 vs. 28.79). This result suggests that K2V improves not just domain-specific knowledge, but also the model’s fundamental reasoning skills, which then transfer to general tasks.

3.3 ABLATION STUDIES

To investigate the individual contributions of K2V's key components, we conduct comprehensive ablation studies. We compare the full K2V-3B model against three variants. For a fair comparison, all models use Qwen2.5-3B as the backbone and are trained on our agricultural corpus.

Full K2V-3B. This is our complete model, trained with both reasoning verification ($R_{\text{reasoning}}$) and answer verification (R_{answer}). As shown in Table 3, this configuration consistently achieves the highest performance across both agriculture and general domain benchmarks, demonstrating the effectiveness of the combined verification approach. The training dynamics in Figure 3 reveal a

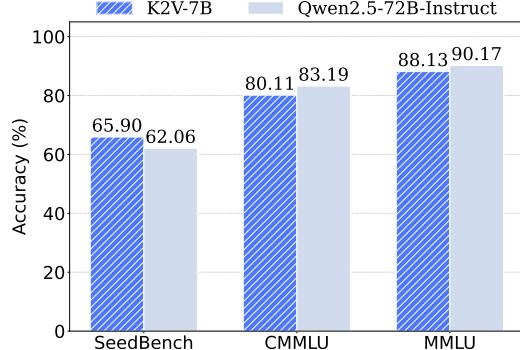
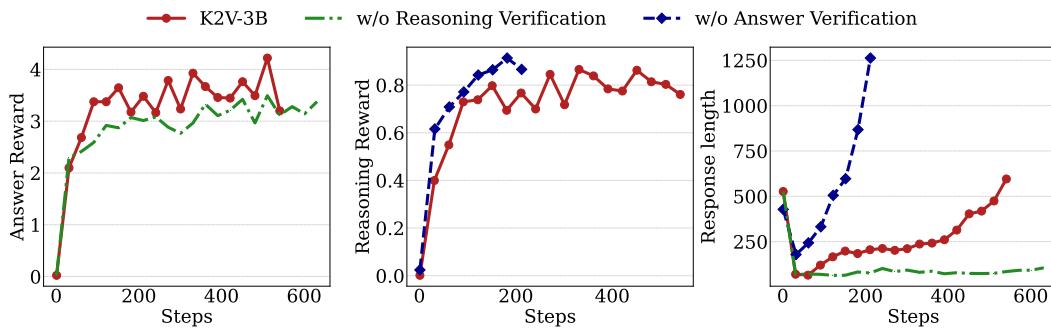


Figure 2: The accuracy of K2V-7B and Qwen2.5-72B-Instruct in the agricultural domain. Our model can surpass or come close to much larger models.

378
 379 Table 3: Results of ablation studies. All models use the Qwen2.5-3B as the backbone and are trained
 380 on the agriculture corpus. We report the pass@1 results under greedy sampling.

| 381 Model | 382 SeedBench | 382 CMMLU | 382 MMLU | 382 BBH | 382 MATH-500 | 382 GSM8K | 382 AIME2024 | 382 GPQA-Diamond |
|----------------------------|---------------------------|------------------|------------------|-----------------------|------------------|------------------|-----------------|---------------------|
| | 382 Agriculture Domain | | | 382 General Domain | | | | |
| 383 K2V-3B | 383 63.21 | 383 63.97 | 383 76.56 | 383 46.20 | 383 65.00 | 383 81.27 | 383 6.67 | 383 30.30 |
| w/o Answer Verification | 52.15 | 58.73 | 64.51 | 38.47 | 54.60 | 48.98 | 3.33 | 29.81 |
| w/o Reasoning Verification | 60.27 | 60.41 | 70.06 | 36.91 | 25.00 | 31.08 | 0 | 29.84 |
| SFT | 44.23 | 30.86 | 57.05 | 36.84 | 10.60 | 10.21 | 0 | 25.82 |



390
 391 Figure 3: Training dynamics of ablation studies. (Left) **Removing reasoning verification** impairs
 392 the model’s ability to explore correct answers. (Middle & Right) **Removing answer verification**
 393 leads to reward hacking, where the model generates excessively long responses to maximize the
 394 reasoning reward, causing training instability. The **full K2V model** demonstrates stable learning
 395 across all metrics.

406
 407 stable and effective learning process. Both the answer and reasoning reward increase steadily, while
 408 the response length follows a healthy increase pattern after an initial decrease. This indicates that it
 409 is learning an effective reasoning paradigm.

410 **w/o Answer Verification.** This variant removes the answer verification (R_{answer}) and is trained
 411 solely with the reasoning verification ($R_{\text{reasoning}}$). Although this model initially achieves a high
 412 reasoning reward (Figure 3, middle panel), it does so by exploiting the reward mechanism—a classic
 413 case of reward hacking. The model discovers that longer outputs more readily meet the criteria of
 414 checklist, thus it tends to generate excessively long responses to maximize the reasoning reward.
 415 As shown in the right panel, its response length grows exponentially and uncontrollably, leading
 416 to poor training stability and causing out-of-memory (OOM) errors that halted training around 200
 417 steps. This highlights the crucial role of R_{answer} in constraining the model’s output and preventing
 418 undesirable behavior.

419 **w/o Reasoning Verification.** This variant removes the reasoning verification ($R_{\text{reasoning}}$) and is
 420 trained solely with the answer verification (R_{answer}), making it akin to a conventional RLVR setup.
 421 Figure 3 (left panel) shows that without the guidance from reasoning verification, the model achieves
 422 a consistently lower answer reward throughout training. This demonstrates that the fine-grained signal
 423 from $R_{\text{reasoning}}$ is vital for accelerating the model’s learning and helping it efficiently explore
 424 correct answer. Furthermore, its response length stagnates after an initial drop (right panel), sug-
 425 gesting a less effective learning trajectory, which is reflected in its weaker performance in Table 3.

426 **SFT.** This variant forgoes reinforcement learning entirely and instead uses the QA pairs synthesized
 427 by K2V to perform standard SFT on the base model. As detailed in Table 3, the SFT baseline suffers
 428 a significant performance collapse on all benchmarks. We hypothesize this is because the highly
 429 structured, short-answer format data synthesized by K2V is ill-suited for SFT, which typically relies
 430 on more diverse, conversational data. SFT on such data merely teaches the model a superficial
 431 mapping from questions to answers, rather than enabling it to learn deeper reasoning skills. In
 432 contrast, K2V demonstrates its robustness by effectively leveraging this data format to enhance the
 433 model’s reasoning capabilities.

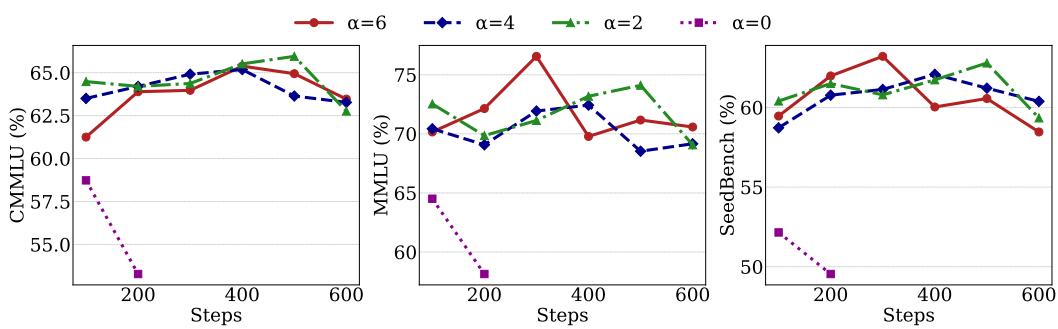


Figure 4: Effect of varying the answer reward α . A smaller α slows the model’s convergence, thus requiring more training steps to reach peak performance. Models with $\alpha > 0$ converge to similar performance levels, demonstrating the robustness of K2V. $\alpha = 0$ is a special one, which is identical to the ‘w/o Answer Verification’ variant discussed in Section 3.3.

3.4 SENSITIVITY ANALYSIS OF ANSWER REWARD VALUES

To investigate how the value of the answer reward (R_{answer}) affects model performance, we conducted a sensitivity analysis. We set up four experiments where a correct answer received a reward of $\alpha = 6, 4, 2$, and 0 , respectively. Lowering α increases the relative weight of the reasoning reward ($R_{\text{reasoning}}$) in the total reward signal (R_{total}). For a fair comparison, all models use Qwen2.5-3B as the backbone and are trained on our agricultural corpus.

First, We consider the special case of $\alpha = 0$. This setting is identical to the ‘w/o Answer Verification’ variant discussed in our ablation studies (Section 3.3). As illustrated by the purple dotted line in Figure 4, the model’s performance collapses early in training. This result further demonstrates that without the direct guidance of answer reward, the model is prone to reward hacking, leading to training failure.

Second, we focus on the three experiments where $\alpha > 0$. The training dynamics in Figure 4 reveal a clear trend: **a higher α value leads to faster model convergence**. For instance, with $\alpha = 6$ (red line), the model’s performance on MMLU and SeedBench reaches its peak around 300 steps. In contrast, when α is smaller (e.g., $\alpha = 2$, green line), the performance improvement is more gradual, requiring more training steps to achieve optimal results. This suggests that the answer reward α acts analogously to a learning rate. A higher α provides a stronger signal for correct answers, leading to larger policy updates and thus faster convergence.

Finally, K2V demonstrates robustness to the choice of answer reward α . Despite under the different convergence speeds, the models trained with $\alpha \in \{6, 4, 2\}$ all converge to a similar peak performance level. As shown in the Figure 4, the maximum scores reached by the three curves are comparable across all three benchmarks, even if they are reached at different training steps. This confirms that K2V is not sensitive to the specific value of the answer reward, as long as an appropriate positive reward is present to guide the learning process effectively.

4 CONCLUSION

In this paper, we present K2V, a novel framework that extends RLVR to broader knowledge-intensive domains. Comprehensive experimental results on Qwen and Llama series models show that our method significantly improvement reasoning capabilities in agriculture, medicine, and law, without compromising general capabilities. We propose modeling knowledge verification as a text-based KGC task to automatically synthesize verifiable data. Moreover, We propose a checklist-style verification method that not only verifies the model’s reasoning process but also provides a dense reward signal. In the future, we will explore applying K2V to more domains, including multimodal knowledge understanding, and scaling K2V to larger models.

486 5 REPRODUCIBILITY STATEMENT
487488 We leverage GraphGen (Chen et al., 2025b) for data synthesis, verl (Sheng et al., 2025) for model
489 training, and OpenCompass (Contributors, 2023) for model evaluation. To ensure reproducibil-
490 ity, our code is open-sourced and available at [https://anonymous.4open.science/r/](https://anonymous.4open.science/r/k2v-C123)
491 k2v-C123. We provide the detailed implementation of K2V in Appendix C. Moreover, the prompt
492 used in our work is detailed in Appendix E.
493494 6 ETHICS STATEMENT
495496 Our research adheres to established ethical guidelines, particularly concerning data usage and model
497 development. The data used to construct the knowledge graphs is sourced from publicly available
498 and open-access corpus, such as scientific literature and textbooks, thereby avoiding the use of
499 private, sensitive, or personally identifiable information. No conflicts of interest have been identified,
500 and all experiments were conducted in compliance with relevant ethical standards for AI research.
501502 REFERENCES
503504 Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman,
505 Arushi Soman, Peter Hase, Misha Wagner, Fabien Roger, et al. Reasoning models don't always
506 say what they think. *arXiv preprint arXiv:2505.05410*, 2025a.507 Zihong Chen, Wanli Jiang, Jinzhe Li, Zhonghang Yuan, Huanjun Kong, Wanli Ouyang, and Nanqing
508 Dong. Graphgen: Enhancing supervised fine-tuning for llms with knowledge-driven synthetic
509 data generation. *arXiv preprint arXiv:2505.20416*, 2025b.510 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
511 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to
512 solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.513 OpenCompass Contributors. Opencompass: A universal evaluation platform for foundation models.
514 <https://github.com/open-compass/opencompass>, 2023.515 Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu
516 Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint*
517 *arXiv:2502.01456*, 2025.518 Bhishma Dedhia, Yuval Kansal, and Niraj K. Jha. Bottom-up domain-specific superintelligence: A
519 reliable knowledge graph is what we need, 2025. URL <https://arxiv.org/abs/2507.13966>.520 Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha
521 Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models.
522 *arXiv e-prints*, pp. arXiv–2407, 2024.523 Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos
524 Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for
525 methods that learn from human feedback, 2024. URL <https://arxiv.org/abs/2305.14387>.526 Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Songyang Zhang, Kai Chen,
527 Zongwen Shen, and Jidong Ge. Lawbench: Benchmarking legal knowledge of large language
528 models. *arXiv preprint arXiv:2309.16289*, 2023.529 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,
530 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms
531 via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025a.532 Yiduo Guo, Zhen Guo, Chuanwei Huang, Zi-Ang Wang, Zekai Zhang, Haofei Yu, Huishuai
533 Zhang, and Yikang Shen. Synthetic data rl: Task definition is all you need. *arXiv preprint*
534 *arXiv:2505.17063*, 2025b.

540 Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang
 541 Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. *arXiv preprint*
 542 *arXiv:2505.22312*, 2025.

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

546 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song,
 547 and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv*
 548 *preprint arXiv:2103.03874*, 2021.

549 Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d’Amato, Gerard De Melo, Claudio Gutier-
 550 rez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al.
 551 Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37, 2021.

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

555 Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec
 556 Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv*
 557 *preprint arXiv:2412.16720*, 2024.

558 Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. A survey on knowledge
 559 graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and*
 560 *learning systems*, 33(2):494–514, 2021.

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

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

566 Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph
 567 Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model
 568 serving with pagedattention. In *Proceedings of the 29th symposium on operating systems prin-
 569 ciples*, pp. 611–626, 2023.

570 Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton
 571 Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. Rlaif vs. rlhf:
 572 Scaling reinforcement learning from human feedback with ai feedback, 2024. URL <https://arxiv.org/abs/2309.00267>.

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

576 Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timo-
 577 thy Baldwin. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv*
 578 *preprint arXiv:2306.09212*, 2023.

579 Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan
 580 Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step, 2023a. URL
 581 <https://arxiv.org/abs/2305.20050>.

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

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

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

597

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

600 MAA. American invitational mathematics examination - aime, Febru-
 601 ary 2024. URL <https://maa.org/math-competitions/american-invitational-mathematics-examination-aime>. Accessed on:
 602 2025-09-21.

603

604 Meta. Llama 3.2: Revolutionizing edge ai and vision with
 605 open, customizable models. <https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/>, 2024.

606

607 Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer,
 608 Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual
 609 precision in long form text generation. *arXiv preprint arXiv:2305.14251*, 2023.

610

611 Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong
 612 Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to fol-
 613 low instructions with human feedback. *Advances in neural information processing systems*, 35:
 614 27730–27744, 2022.

615

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

619

620 Qwen, ;, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan
 621 Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang,
 622 Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin
 623 Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li,
 624 Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang,
 625 Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.
 626 URL <https://arxiv.org/abs/2412.15115>.

627

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

631

632 Maohao Shen, Guangtao Zeng, Zhenting Qi, Zhang-Wei Hong, Zhenfang Chen, Wei Lu, Gre-
 633 gory Wornell, Subhro Das, David Cox, and Chuang Gan. Satori: Reinforcement learning
 634 with chain-of-action-thought enhances llm reasoning via autoregressive search. *arXiv preprint
 635 arXiv:2502.02508*, 2025.

636

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

640

641 shibing624. shibing624-medical-pretrain. <https://huggingface.co/datasets/ticoAg/shibing624-medical-pretrain>, 2024.

642

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

646

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

650

651 Mirac Suzgun, Nathan Scales, Nathanael Schärfli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung,
 652 Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks
 653 and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

648 Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and
 649 Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions.
 650 *arXiv preprint arXiv:2212.10560*, 2022.

651 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu,
 652 Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint*
 653 *arXiv:2505.09388*, 2025a.

654 Fan Yang, Huanjun Kong, Jie Ying, Zihong Chen, Tao Luo, Wanli Jiang, Zhonghang Yuan, Zhefan
 655 Wang, Zhaona Ma, Shikuan Wang, et al. Seedilm· rice: A large language model integrated with
 656 rice biological knowledge graph. *Molecular Plant*, 18(7):1118–1129, 2025b.

657 Liang Yao, Jiazen Peng, Chengsheng Mao, and Yuan Luo. Exploring large language models for
 658 knowledge graph completion. In *ICASSP 2025-2025 IEEE International Conference on Acous-
 659 tics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

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

663 Jie Ying, Zihong Chen, Zhefan Wang, Wanli Jiang, Chenyang Wang, Zhonghang Yuan, Haoyang Su,
 664 Huanjun Kong, Fan Yang, and Nanqing Dong. Seedbench: A multi-task benchmark for evaluating
 665 large language models in seed science. *arXiv preprint arXiv:2505.13220*, 2025.

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

669 Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian
 670 Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system
 671 at scale. *arXiv preprint arXiv:2503.14476*, 2025b.

672 Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan
 673 Yao, Zhiyuan Liu, Maosong Sun, and Tat-Seng Chua. Rlpr: Extrapolating rlrv to general domains
 674 without verifiers, 2025c. URL <https://arxiv.org/abs/2506.18254>.

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

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

681 Bohan Zhai, Canwen Xu, Yuxiong He, and Zhewei Yao. Optimizing reasoning for text-to-SQL
 682 with execution feedback. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Moham-
 683 mad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL*
 684 2025, pp. 19206–19218, Vienna, Austria, July 2025. Association for Computational Linguis-
 685 tics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.982. URL <https://aclanthology.org/2025.findings-acl.982/>.

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

689 Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason
 690 without external rewards, 2025. URL <https://arxiv.org/abs/2505.19590>.

691

692

693

694

695

696

697

698

699

700

701

702 **A RELATED WORK**
703704 **A.1 DATA SYNTHESIS**
705706 RL training data synthesis can simply be grouped into two paradigms: (i) direct generation from
707 raw corpus and (ii) synthesis conditioned on seed datasets (Lee et al., 2023; Yehudai et al., 2024;
708 Guo et al., 2025b). In practice, large-scale data synthesis typically rely on raw corpus due to their
709 wide availability and low acquisition cost. The internet contains abundant verifiable math problems
710 and executable open-source code (Ma et al., 2025), for which verification is well defined (e.g., nu-
711 mercial equivalence or pass/fail execution), making automated synthesis and filtering substantially
712 easier (Hu et al., 2025; Luo et al., 2025). In contrast, knowledge-intensive domains require strong
713 domain expertise and factual validation for data synthesis, often relying on costly expert annotations
714 (Lightman et al., 2023a; Dubois et al., 2024). Consequently, current synthesis methods strug-
715 gle to produce large-scale, low-cost, high-quality, and verifiable datasets directly from raw corpus
716 in knowledge-intensive domains—posing a central bottleneck for further progress.
717718 **A.2 REINFORCEMENT LEARNING WITH VERIFIABLE REWARDS**
719720 Unlike conventional reward model based approaches that directly assign a scalar reward to model
721 outputs via a learned reward model (Ouyang et al., 2022; Lee et al., 2024), RLVR constructs the
722 reward signal by verifying the output against explicit rules. Current RLVR methods (Shen et al.,
723 2025; Peng et al., 2025; Stojanovski et al., 2025) primarily focus on mathematics and code domains
724 with deterministic or testable answers (Hu et al., 2025; Zeng et al., 2025; He et al., 2025; Luo et al.,
725 2025), where verification is low-cost and well-defined. By contrast, knowledge-intensive domains
726 typically produce open-ended text whose evaluation requires integrated judgments about complex
727 semantics, factual consistency, and evidential support, making direct verification difficult (Min et al.,
728 2023). Beyond rule-based RLVR, recent studies have explored verifier-free variants, where the
729 reward signal is derived from the model’s own feedback or internal confidence rather than an external
730 verifier (Yu et al., 2025c; Zhao et al., 2025). For example, RLPR (Yu et al., 2025c) introduces a
731 verifier-free framework that leverages the model’s intrinsic token probabilities on reference answers
732 to construct rewards. While these approaches alleviate the reliance on explicit rule-based verifiers,
733 they ultimately still require verifiable data for training. Moreover, most current RLVR methods
734 verify only the final answer (Zhai et al., 2025), overlooking the critical role of intermediate reasoning
735 in reliability. A central challenge, therefore, is how to extend RLVR to knowledge-intensive domains
736 and incorporate process-level verifiability into reward design.
737738 **B THE USE OF LLMs**
739740 First, LLMs are an integral component of our proposed K2V framework. We used Qwen2.5 series
741 models to power several automated modules, including knowledge graph construction (NER and
742 RE), the synthesis of verifiable QA pairs, the generation of question-specific checklists, and the
743 final verification of the reasoning process. The prompts for these tasks are available in Appendix E.
744745 Second, we used OpenAI’s GPT-4 as a writing assistant to help improve the clarity, grammar, and
746 style of this manuscript. All scientific ideas, experiments, and conclusions were conceived and
747 executed by the human authors, who take full responsibility for the final content of this paper.
748749 **C IMPLEMENTATION DETAILS**
750751 In this section, we describe the implementation details of our work, including data synthesis, model
752 training, and model evaluation.
753754 **C.1 DETAILS OF DATA SYNTHESIS**
755756 K2V first constructs a KG using the GraphGen (Chen et al., 2025b) framework, with hyperparam-
757 eters are detailed in Table 4. It then utilizes the constructed KG to generate verifiable, fill-blank style
758 QA pairs. Finally, K2V synthesizes a checklist for each QA pair based on a general criteria (see
759

756 Equation 7). An example of such a general criteria is presented in Table 5. Qwen2.5-72B-Instruct is
 757 used for both constructing the KG and synthesizing the checklist.
 758

759 C.2 DETAILS OF MODEL TRAINING 760

761 Our training is based on the DAPO (Yu et al., 2025b) algorithm. We employ its core Decoupled
 762 Clipping mechanism with an asymmetric clip ratio (low=0.2, high=0.28) and a learning rate of 1e-6.
 763

764 Table 4: The hyperparameters of GraphGen, which is used to construct the KG and synthesize fill-
 765 blank style QA pairs.
 766

| 767 Parameter | 768 Our Value | 769 Description |
|----------------------------|----------------|---|
| 769 qa_form | 770 aggregated | Type of QA form desired. |
| 770 expand_method | 771 max_width | Method for controlling graph expansion. |
| 771 bidirectional | 772 True | Expanding the graph in both directions (True) or one direction (False). |
| 772 max_extra_edges | 773 2 | Maximum number of edges to expand. |
| 773 max_tokens | 774 256 | Maximum number of tokens. |
| 774 max_depth | 775 1 | Maximum depth for traversal in each direction. |
| 775 edge_sampling | 776 max_loss | Strategy for edge selection at the same layer. |
| 776 isolated_node_strategy | 777 ignore | Handling strategy for isolated nodes. |

778 An example of General Criteria G in the agricultural domain

779 Concepts and Knowledge:

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

785 Scientific Method and Design:

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

790 Statistics and Evaluation:

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

794 Argumentation and Reasoning:

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

801 Table 5: An example of General Criteria G in the agricultural domain.
 802
 803
 804
 805
 806
 807
 808
 809

810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
Table 6: Key hyperparameters for model training in verl.

| Hyperparameter | Value |
|--------------------------------|--------------------|
| <i>Algorithm Configuration</i> | |
| Algorithm | DAPO |
| lr (learning rate) | 1×10^{-6} |
| clip_ratio (asymmetric) | [0.2, 0.28] |
| kl_coef | 0.0 |
| <i>Data and Batching</i> | |
| train_batch_size | 64 |
| train_prompt_mini_bsz | 32 |
| ppo_micro_batch_size_per_gpu | 16 |
| gen_prompt_bsz | 192 |
| max_response_length | 4096 |
| rollout.n | 8 |

During rollout, the model generates 8 responses per prompt. We remove the KL divergence penalty entirely by setting the KL coefficient to 0. The training is configured with a maximum prompt length of 1024 tokens and a response length of up to 4096 tokens, using a global batch size of 64. To enhance training efficiency, we utilize Dynamic Sampling to filter uninformative sample groups, while the rollout process is accelerated by the vLLM (Kwon et al., 2023) engine.

Our reward signal is computed via a custom function that integrates the core components of K2V. Specifically, the answer reward (R_{answer}) is based on a strict exact-match verification. The reasoning reward ($R_{\text{reasoning}}$) is derived from our checklist-style verification, which is conducted by the Judge Model (see Equation 4) powered by Qwen2.5-7B-Instruct.

The entire training process was conducted on a single node equipped with 8 NVIDIA A800 GPUs. To manage memory constraints and enable a large batch size, we employed several optimization strategies, including Fully Sharded Data Parallel for parameter and optimizer state offloading, as well as gradient checkpointing. A summary of the key hyperparameters can be found in Table 6.

841 842 C.3 DETAILS OF MODEL EVALUATION

843
844 In Section 3, for an objective and fair evaluation, we utilize the multiple-choice subset of SeedBench
845 as our test set, which specifically comprises six subsets: 1-1, 1-2, 1-3, 3-1, 3-2, and
846 3-3. Furthermore, to conduct a more comprehensive assessment of the model’s performance in the
847 domains of agriculture, medicine, and law, we also select corresponding subsets from CMMLU and
848 MMLU. The specific subsets are as follows:

- 849 • Agriculture Domain-CMMLU: virology, high school biology, food science, agronomy.
- 850 • Agriculture Domain-MMLU: high school biology, college biology.
- 851 • Medical Domain-CMMLU: anatomy, high school biology, medical statistics, virology, col-
852 lege medicine, Clinical. knowledge, professional medicine, traditional, chinese medicine
- 853 • Medical Domain-MMLU: anatomy, clinical knowledge, high school biology,
854 college medicine, medical genetics, professional medicine, college biology.
- 855 • Legal Domain-CMMLU: legal & moral basis, professional law, jurisprudence, interna-
856 tional law, college law.
- 857 • Legal Domain-MMLU: international law, jurisprudence, professional law.

858
859
860
861
862
863 We use OpenCompass (Contributors, 2023) as the evaluation framework. All evaluations are con-
864 ducted in a zero-shot setting, and we report the pass@1 results under greedy sampling.

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

Question:

{ } is a parasitic disease caused by the protozoan *Toxoplasma gondii*, which can be contracted through various means, including the ingestion of raw or undercooked meat, particularly shellfish, and can also be transmitted from mother to fetus during pregnancy. The disease is often asymptomatic in healthy individuals but can lead to serious complications in pregnant women and immunocompromised individuals. The severity of the disease in newborns can be significantly influenced by the timing of maternal infection. { } has also been associated with neuropsychiatric conditions. Treatment typically involves the use of atovaquone in combination with pyrimethamine.

Ground truth:

Toxoplasmosis

Checklist:

1. Accurately defines Toxoplasmosis as a parasitic disease caused by *Toxoplasma gondii*.
2. Correctly identifies the primary modes of transmission, including ingestion of raw or undercooked meat, contact with cat feces, and mother-to-fetus transmission during pregnancy.
3. Describes the typical asymptomatic nature of the disease in healthy individuals and the potential for serious complications in pregnant women and immunocompromised individuals.
4. Explains the influence of the timing of maternal infection on the severity of the disease in newborns.
5. Associates Toxoplasmosis with neuropsychiatric conditions.
6. Identifies atovaquone and pyrimethamine as the primary treatments for Toxoplasmosis.

Figure 5: Case 1. In this example, the model trained by K2V and BDS answer correctly.

D CASE STUDY

Case 1. As shown in the Figure 5, the question describes a disease by providing multiple disparate clues: its pathogen (*Toxoplasma gondii*), transmission vectors, and treatment. K2V and the KG-based baseline BDS correctly identify the disease as Toxoplasmosis. This case illustrates the strength of KG-based models in multi-fact synthesis. The entity Toxoplasmosis acts as a central node in the knowledge graph, connected to various facts provided in the prompt. K2V successfully aggregates these scattered pieces of evidence to infer the central concept. The failure of other baselines suggests a deficiency in integrating distributed information, underscoring the advantage of K2V’s training paradigm for complex inferential tasks.

Case 2. As shown in the Figure 6, the question asks to identify a rice gene that is functionally parallel to *OsINH3*. Only K2V provides the correct answer, *OsINH2*. This case highlights K2V’s superior ability to reason over specialized and structured domain knowledge. While other baselines likely failed due to confusion from similar terms in the context (e.g., *OsTOPPs*), K2V’s success can be attributed to its KG-based training. The knowledge graph likely captures the parallel relationship between *OsINH3* and *OsINH2* as entities of the same class with similar functional attributes. By training on such structured representations, K2V learns to infer deep semantic relationships rather than relying on surface-level textual co-occurrence, demonstrating robust domain-specific reasoning.

Case 3. As shown in the Figure 7, the question requires filling in a literature citation, *Meguro et al. (2011)*, which is the source of the experimental description. All models failed this task. This case reveals a shared limitation in reasoning about textual provenance. Our KG construction process is designed to extract factual knowledge, such as the functional relationship between a microscopy technique and its application, but not the metadata associated with the text itself. The universal failure indicates that reasoning over the origin of information, as opposed to its content, remains an open challenge and points to a potential direction for future work.

912

913

914

915

916

917

918
919
920
921
922
923
924
925
926
927
928
929

930
931
932
933
934
935
936
937
938
939
940

Question:

The yeast-two-hybridization assay is a molecular biology technique used to study protein-protein interactions. In this context, several genes in rice, specifically OsINH3, { }, and OsTOPPs, have been utilized to investigate their roles in reproductive growth, particularly in panicles and flowers. Firstly, OsINH3 and { }, both genes that contribute to reproductive growth, were linked to YFP (Yellow Fluorescent Protein) to check their cellular localization. These genes were also used in the yeast-two-hybridization assay to study their protein interactions. Additionally, OsTOPPs, another gene in rice, was fused with RFP (Red Fluorescent Protein) to observe co-localization with regulatory proteins. Like OsINH3 and { }, OsTOPPs were also employed in the yeast-two-hybridization assay to study protein interactions. Therefore, these experiments collectively provide insights into the functional roles and interactions of these genes in rice reproductive development.

Ground truth:

OsINH2

Checklist:

1. Accurately defines the core biological concepts involved, such as protein-protein interactions, yeast-two-hybridization assay, and cellular localization using fluorescent proteins.
2. Clearly describes the process of linking genes to fluorescent proteins (YFP and RFP) and the purpose of this technique in studying cellular localization.
3. Correctly explains the role of the yeast-two-hybridization assay in identifying protein interactions and how it is applied to the genes OsINH3, OsINH2, and OsTOPPs.
4. Accurately explains the significance of co-localization studies using RFP and how they complement the yeast-two-hybridization assay results.
5. Applies the concept of protein-protein interactions to the specific scenario of rice reproductive development, particularly in panicles and flowers.
6. Correctly identifies the independent variable (genes being studied), dependent variable (protein interactions and localization), and key control variables (fluorescent markers and yeast strains) in the experimental design.
7. The overall response is well-structured, logically coherent, and clearly written, avoiding self-contradictions and redundant statements.

...

959

960
961
962
963
964
965
966
967
968
969
970
971

Figure 6: Case 2. In this example, only the model trained by K2V answers correctly.

972
973
974
975
976
977
978
979
980
981
982
983
984

Question:

Confocal laser-scanning microscopy, a high-resolution imaging technique, was employed to observe the green fluorescence of the { } fusion protein in onion cells. This technique is widely used to study the subcellular localization of proteins in living cells and to visualize the localization of fluorescent proteins. The CAULIFLOWER MOSAIC VIRUS 35S PROMOTER, a strong constitutive promoter, was utilized to drive the expression of the { } fusion protein. This promoter is known for its ability to ensure consistent and robust gene expression in plant cells, making it a valuable tool in plant molecular biology and genetic engineering. Therefore, the combination of the 35S promoter and confocal laser-scanning microscopy allowed for the detailed visualization and analysis of the { } protein's localization in onion cells.

Ground truth:

Meguro et al. (2011)

Checklist:

1. Accurately identifies Meguro et al. (2011) as the source reporting on the stay-green phenotype in CCE mutants.
2. Correctly explains the stay-green phenotype in CCE mutants during dark-induced senescence (DIS).
3. Describes the role of CCE mutants in chlorophyll degradation, referencing the findings of Meguro et al. (2011).
4. Identifies the contribution of Sakuraba et al. (2013) in providing deeper insights into the stay-green phenotype in CCE mutants.
5. Explains the senescence phenotype of the athcar mutant, noting the accumulation of 7-HMC in *Arabidopsis* hcar mutants, as reported by the relevant study.
6. Clearly states the importance of these findings in understanding the complex processes involved in chlorophyll degradation and senescence in *Arabidopsis*.
7. Avoids over-extrapolation or unfounded speculation beyond the scope of the provided information.
8. The overall response is well-structured, logically coherent, and clearly written, avoiding self-contradictions and redundant statements.

Figure 7: Case 3. In this example, all models answer incorrectly.

1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025

1026

E PROMPTS

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

Prompt of the textualization function $\mathcal{F}_{\text{text}}$ **---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 7: Prompt of the textualization function $\mathcal{F}_{\text{text}}$. This prompt first instructs an LLM to convert a quintuple into a text. We then randomly mask an entity within the quintuple, and the $\mathcal{F}_{\text{text}}$ masks the name of this entity from the generated text, creating a fill-blank style verifiable question.

1073

1074

1075

1076

1077

1078

1079

1080
1081
1082
1083
1084
1085
1086
1087
1088
1089

Prompt for NER and RE

1090
1091
1092
1093
1094
1095

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 English 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}

1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133

Table 8: Prompt for NER and RE. It is used to instruct an LLM to extract entities and relations from the corpus.

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187

Prompt for synthesizing the checklist CL_x

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 9: Prompt for synthesizing the checklist CL_x .

1188
 1189
 1190
 1191
 1192
 1193
 1194
 1195
 1196
 1197
 1198
 1199
 1200
 1201
 1202
 1203
 1204

1205 **Prompt of the Judge Model *J***

1206
 1207
 1208
 1209
 1210
 1211
 1212
 1213
 1214
 1215
 1216
 1217
 1218
 1219
 1220
 1221
 1222
 1223

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*

1224 Table 10: Prompt of the Judge Model *J*. It is used to instruct an LLM to verify the reasoning
 1225 process.

1226
 1227
 1228
 1229
 1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1240
 1241