KnowRL: Teaching Language Models to Know What They Know

Sahil Kale
KnowledgeVerse AI
Atlanta, Georgia
USA

Devendra Singh Dhami
Department of Mathematics and Computer Science
Uncertainty in Artificial Intelligence Group
TU Eindhoven, Eindhoven, The Netherlands

Abstract

Truly reliable AI requires more than simply scaling up knowledge; it demands the ability to know what it knows and when it does not. Yet recent research shows that even the best LLMs misjudge their own competence in more than one in five cases, making any response born of such internal uncertainty impossible to fully trust. Inspired by self-improvement reinforcement learning techniques that require minimal data, we present a simple but powerful framework *KnowRL* that strengthens a model's internal understanding of its own feasibility boundaries, enabling safer and more responsible behaviour. Our framework combines two components: (i) introspection, where the model generates and classifies tasks it judges feasible or infeasible, and (ii) consensus-based rewarding, where stability of self-knowledge assessment is reinforced through internal agreement. By using internally-generated data, this design strengthens consistency in self-knowledge and entirely avoids costly external supervision. In experiments on LLaMA-3.1-8B and Qwen-2.5-7B, *KnowRL* steadily improved self-knowledge, validated by both intrinsic self-consistency and extrinsic benchmarking. With nothing more than a small seed set and no external supervision, our method drove gains as high as 28% in accuracy and 12% in F1, outperforming baselines in just a few iterations. Our framework essentially unlocks the untapped capacity of LLMs to self-improve their knowledge awareness, opening the door to reliable, more accountable AI and safer deployment in critical applications. Owing to its simplicity and independence from external effort, we encourage applying this reliability-enhancing process to all future models, and we release all code and data publicly to support broad adoption [1].

1 Introduction

True intelligence is measured not just by the accumulation of knowledge, but by the ability to recognise the limits of our own understanding [1, 2]. In terms of large language models (LLMs), this refers to the fundamental property of self-knowledge, defined as the ability for models to clearly delineate between feasible and infeasible tasks based on knowing their own capability and knowledge boundaries [3]. Recent research demonstrates that even leading models misjudge their competence in more than one out of five instances [4], leading to severe trust and safety issues since responses drawn from such internal uncertainty can never be considered completely reliable [5]. Methods that enable LLMs to reliably and consistently recognise the boundaries of their own knowledge are urgent not only for ensuring true reliability and trustworthiness, but also to enable widespread AI adoption, safely.

[1] `https://anonymous.4open.science/r/KnowRL-5BF0`

Preprint.

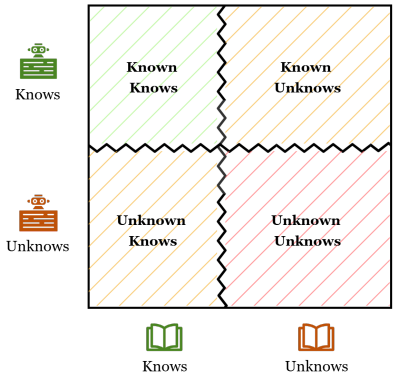


Figure 1: The main idea of self-knowledge represented by the know-unknow quadrant, adapted from [6]. The horizontal axis reflects the model's awareness of the information space, while the vertical axis captures the model's capacity to accurately comprehend and apply what it knows.

The problem of self-knowledge, or the lack thereof in LLMs, has been highlighted as a stand-alone issue [2, 6, 7], an issue related to other known problems like memorisation [8, 9] and adversarial helpfulness [10], and also as an underlying cause for AI safety issues [11, 12]. Since external databases or scaffolding techniques are not suitable to resolve this ingrained gap [13], modifying training patterns or applying post-hoc reinforcement and fine-tuning remain the most effective and viable approaches to address the problem. Calibrating a model's confidence can signal when an answer might be wrong, but it does not guarantee that the model is consistent about what it truly knows versus what it doesn't. We therefore, like previous research, treat self-knowledge as a separate problem from uncertainty estimation.

In this paper, we leverage Reinforcement Learning (RL), a powerful alignment strategy [14], to systematically reinforce self-knowledge within large language models. Since models show large wavering in their own perception of knowledge boundaries [4], we introduce a self-consistency based approach that explicitly strengthens the model's understanding of feasibility and infeasibility limits for itself. We posit that the capacity for such introspection already exists within LLMs, and our approach serves to reinforce and guide this latent ability, empowering the models to better understand and articulate what they do and do not know. By using a variation on Self-Play Reinforcement Learning [15], our method enables LLMs to bootstrap a more accurate and consistent view of their own boundaries, even with minimal initial data.

Since current large language models show a significant lack of self-knowledge, their responses can never be considered inherently reliable and perfectly grounded [5]. To represent visually, Figure 1 depicts self-knowledge as the know-unknow quadrant, with jagged boundaries reflecting uncertainty within this distinction in LLMs. Our goal is to smooth and formalise these boundaries so LLMs can recognize their own knowledge limits and adapt their responses accordingly, enabling a new class of responsible, text-based AI systems. We believe this can also address known problems like over-refusal [16], sycophancy [17] and adversarial helpfulness [10]. Beyond technical improvements, this work has important societal implications for safer AI deployment. We encourage researchers to adopt our simple post-training reinforcement framework across critical areas like healthcare, law, and finance, where trustworthy and accountable AI is essential for effective decision-making.

We make our method advantageous and simple to use by inherently addressing two key challenges. First, collecting high-quality human-annotated data in most domains is expensive and difficult [18], but since our goal is to enhance internal knowledge awareness, using LLM-generated data proves both effective and better aligned with our objective. Second, using single model-generated responses directly as ground-truth labels can produce unreliable reward signals in reinforcement learning [19]. We address this by implementing a self-rewarding mechanism based on consensus [20] and consistency of feasibility, which provides stable, trustworthy, yet internally-generated signals. Our main contributions are summarised as follows:

1. We explore how to leverage RL to guide introspection in LLMs for enhanced awareness of self-knowledge boundaries to enable safer AI, even with limited initial data and no external supervision.
2. We introduce the KnowRL framework, built on two key components: introspection, where the LLM generates questions it judges as feasible or infeasible, and consensus-based rewarding, which derives stable, trustworthy reward signals from internal agreement to reinforce the model's self-knowledge.
3. We show that KnowRL can strongly boost LLM self-knowledge, achieving up to 28% accuracy and 12% F1 gains in just a few iterations, demonstrating scalable self-improvement with broader implications for safer, more reliable AI.

2 Preliminaries

2.1 Reinforcement Learning in LLMs

Reinforcement learning (RL) refers to a framework where an agent interacts with an environment by taking actions, receiving rewards, and updating its policy to maximise long-term expected returns [21]. In the context of large language models, RL has been widely applied to adjust model outputs to human preferences, for post-hoc alignment through human feedback like RLHF [22]. More recently, it has also been explored for improving factuality [23] and AI safety [24], demonstrating versatility in refining LLM behaviour beyond pure pre-training.

Language modelling can be directly interpreted as an RL problem, since generating a single response is equivalent to taking an action, and the quality of that response can be analysed to get a reward signal and improve the model. For an input x, a policy $\pi_\theta(y \mid x)$ generates a response y, and the model receives a reward based on y^*, the reference answer.

$$r = R(x, y, y^*), \tag{1}$$

The training objective is to maximise the expected reward:

$$J(\theta) = \mathbb{E}_{y \sim \pi_\theta(\cdot|x)}\big[R(x, y, y^*)\big]. \tag{2}$$

In this view, generating a response is the action, the reward reflects response quality, and optimisation adjusts π_θ to favour better responses.

Traditionally, rewards rely on reference-based similarity, such as parsers or verifiers [25]. More recent work removes the dependency on y^* by introducing unsupervised rewards, including majority voting [26] or divergence-based measures [27].

In our context, we focus on reinforcing *awareness of self-knowledge boundaries* more than quality or alignment of responses. By treating precise and consistent self-knowledge estimation as the rewarded behaviour, we can apply RL not just to improve correctness, but to strengthen a model's ability to recognise and respect the boundary between what it knows and what it does not, based on its own internal knowledge.

2.2 Self-Play Reinforcement Learning

Self-play reinforcement learning [15] extends the standard RL setup by enabling a model to generate both the inputs and the corresponding learning rewards. Rather than depending on external supervision, the model interacts with its own outputs to create a feedback loop, which is then used to update its policy. This paradigm has been widely applied in domains such as games [28] and robotics [29], and more recently explored in language modelling as well [30, 31].

In general, for an input x generated through self-play, a policy $\pi_\theta(y \mid x)$ produces a response y. A reward signal $r = R(x, y)$ is then derived from internal consistency or agreement, rather than a fixed reference y^*. The objective remains to maximise expected reward:

$$J(\theta) = \mathbb{E}_{y \sim \pi_\theta(\cdot|x)}[R(x, y)]. \tag{3}$$

In our setting, self-play is not used to optimise for raw task accuracy, but to reinforce self-knowledge. Our KNOWRL framework operates in two stages. In the introspection step, the LLM generates

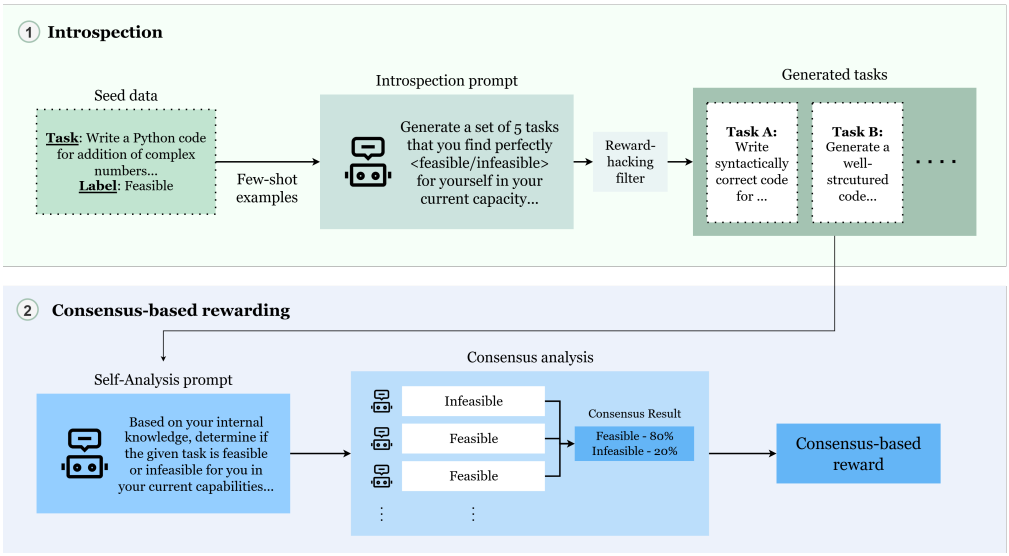


Figure 2: The KnowRL framework, with two key components in an iterative, RL-driven training loop

feasible or infeasible questions and tasks for itself, to probe the limits of its own knowledge. Second, a consensus-based rewarding step, where multiple judgments of its self-generated tasks $\{y_i\}_{i=1}^{k}$ are aggregated, and the reward is defined as the quantified consistency in agreement, producing a stable and trustworthy signal without external labels.

$$r = Consistency\big(\{y_i\}_{i=1}^{k}\big), \tag{4}$$

Our method thus embeds the loop entirely within a single LLM. By combining introspection with consensus-driven rewards, we obtain a lightweight and scalable strategy for reinforcing reliable boundary awareness, crucial for trust and safety in downstream deployment.

Table 1: Results of intrinsic evaluation of self-knowledge boundary consistency for KnowRL across iterations. Δ indicates change from the previous step.

Model	Iteration	Accuracy (%)	Δ (%)
Llama 3.1 8B Instruct	Base Model	33.56	-
	Iter 5	36.78	3.22 ↑
	Iter 10	39.32	2.54 ↑
	Iter 15	41.11	1.79 ↑
	Iter 20	42.13	1.02 ↑
	Iter 25	43.12	0.99 ↑
	Iter 30	42.99	0.13 ↓
Qwen 2.5 7B Instruct	Base Model	39.22	-
	Iter 5	43.19	3.97 ↑
	Iter 10	45.78	2.59 ↑
	Iter 15	46.71	0.93 ↑
	Iter 20	46.77	0.06 ↑
	Iter 25	48.01	1.24 ↑
	Iter 30	48.29	0.28 ↑

3 Methodology

To address the persistent wavering in large language models' perception of their own knowledge boundaries [3, 4], we propose a self-consistency driven framework that reinforces an LLM's internal understanding of feasibility and infeasibility limits. Building on the frameworks of self-play [15] and self-improvement [32], our approach enables models to bootstrap a more accurate and stable view of their own boundaries, even with minimal initial data and external supervision. We describe the KNOWRL framework described in Figure 2 in detail ahead, highlighting how the two components: *introspection* and *consensus-based rewarding* interact to drive iterative improvement in self-knowledge.

3.1 Introspection

Our primary goal is to strengthen a language model's ability to recognise the limits of its own knowledge and capabilities. To achieve this, we design a method that relies on and utilises data generated by the model itself, minimising external supervision. In the introspection step, the model is prompted to propose tasks it confidently believes are either feasible or infeasible, guided by a small set of few-shot verified examples (see Figure 2). As training steps progress, we utilise examples from the initial dataset as well as samples with high consensus generated in previous training steps (examples in Section .4 in the Appendix). We believe that incorporating such samples can allow the model to refine, evolve and stabilise its understanding of feasibility boundaries over successive self-improvement iterations.

3.2 Consensus-based Rewarding

In the consensus-based rewarding stage, the model's own judgments are used to quantify and reinforce the consistency of its self-knowledge. Our main motivations and goal in this stage is to reward model introspection that produces high consensus and consistency and leads to a strong and rooted understanding of its own feasibility boundaries. Formally, given a candidate task x produced during introspection, we draw k independent self-analysis outputs $\{y_i\}_{i=1}^{k}$, where each $y_i \in \{\text{Feasible}, \text{Infeasible}\}$, using the analysis prompt given in Figure 5. Within the prompt, we use simple strategies like QAP [33] and chain-of-thought [34] to encourage better understanding before coming to a conclusion of task feasibility.

The reward is the proportion of outputs agreeing with the majority label, which directly measures the internal consistency of feasibility assessments.

$$r(x) = \frac{1}{k} \sum_{i=1}^{k} \mathbf{1}\Big[y_i = \text{Maj}\{y_1, \ldots, y_k\}\Big],$$

This consensus score serves as the reinforcement signal in our policy update, encouraging the model to generate tasks and judgments that reflect a stable and reliable perception of its own capability boundaries.

3.3 Overall Setup

The KnowRL framework integrates introspection and consensus-based rewarding in an iterative, RL-powered training loop to progressively improve the model's understanding of its own capabilities. During each introspection phase, the model is prompted to generate candidate tasks it believes are either feasible or infeasible, with each introspection run repeated 10–15 times to produce a diverse set of approximately 50–60 candidate tasks. For each candidate task x, we draw $k = 8$ independent self-analysis outputs $\{y_i\}_{i=1}^{k}$, where $y_i \in \{\text{Feasible}, \text{Infeasible}\}$, and compute a consensus-based reward to quantify the internal consistency of the model's feasibility judgments. The set of tasks and corresponding rewards, $\{(x, r(x))\}$, are then used to update the model parameters via reinforcement learning. We refer to one iteration or cycle as running the introspection and consensus-rewarding process once for feasible tasks and once for infeasible tasks, with each run used to update the model through reinforcement learning. By repeating this cycle iteratively across training steps, the LLM progressively strengthens using our RL-powered introspective loop, improving its ability to recognise and articulate the limits of what it can or cannot accomplish without any external labels.

Table 2: Results of extrinsic evaluation of self-knowledge boundary consistency using the SelfAware dataset for KnowRL across iterations. Δ indicates change from the previous step.

Model	Iteration	F1 (%)	Δ (%)
Llama 3.1 8B Instruct	Base Model	56.12	-
	Iter 5	58.01	1.89 ↑
	Iter 10	58.65	0.64 ↑
	Iter 15	61.76	3.11 ↑
	Iter 20	62.34	0.58 ↑
	Iter 25	63.11	0.77 ↑
	Iter 30	63.10	0.01 ↓
Qwen 2.5 7B Instruct	Base Model	62.17	-
	Iter 5	64.88	2.71 ↑
	Iter 10	65.94	1.06 ↑
	Iter 15	67.22	1.28 ↑
	Iter 20	67.89	0.67 ↑
	Iter 25	68.23	0.34 ↑
	Iter 30	68.29	0.06 ↑

3.4 Reward Hacking Filter

During spot checks across intermediate iterations, we observed that models tend to produce overly simple or overly complex tasks once they realise that consensus agreement is more when certain keywords or phrasing is used, essentially hacking the consensus-based reward. To prevent such hacking, where the model learns to maximise consensus by generating overly simple or unnecessarily complex tasks, we adopt a difficulty clipping strategy inspired by prior self-play methods [15]. During training, tasks produced in the introspection step are filtered out based on predetermined criteria.

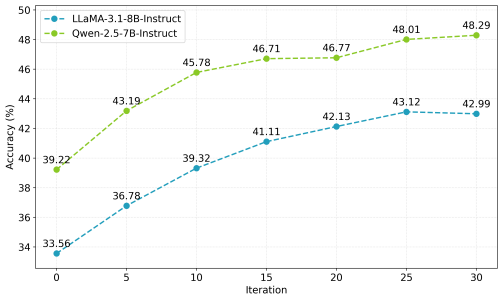
- **Semantic redundancy:** If the ROUGE-L score for any new task exceeds a predefined threshold with existing instructions, we filter out such examples to prevent generating semantically similar instructions that reduce diversity.
- **Keyword filtering:** The presence of specific keywords such as *generating images*, *training models*, or *image, video*, which can directly lead to infeasibility consensus agreement owing to model capability limitations are filtered. We also add this filter since awareness of text-only capabilities are pretty high even without our RL-powered tuning.
- **Perplexity filtering:** We utilise the negative log-likelihood under the base model and discard candidates with perplexity above a fixed threshold, ensuring syntactically fluent and semantically well-formed instructions.

These filters maintain a balanced level of task difficulty and prevent the model from exploiting consensus signals through trivial patterns or unanswerable prompts.

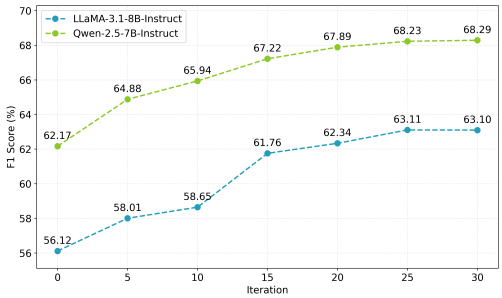
4 Experimentation

4.1 Model Setup

We conduct our experimentation with two small yet powerful open-source models: Llama 3.1 8B Instruct [35] and Qwen-2.5-7B-Instruct [36]. Model parameters and details are found in the Appendix. For both models, we use a seed dataset as few-shot examples for the introspection step containing verified answerable and unanswerable questions generated by the models themselves. More details and prompts for the seed dataset are provided in Section .3 in the Appendix. For training, we use the OpenRLHF [37] framework together with its Reinforce++ algorithm [38] to perform reinforcement learning. Detailed training hyperparameters during the RL process are provided in Section .2 in the Appendix.



(a) Accuracy trend of intrinsic evaluation [4] for both models using KnowRL



(b) F1 score trend of extrinsic evaluation using the SelfAware dataset [6] for both models using KnowRL

Figure 3: Evaluation of self-knowledge boundary consistency

4.2 Dataset and Evaluation

To comprehensively assess our methodology, we employ two complementary evaluation settings: intrinsic analysis and dataset-based evaluation. We run RL-based introspection with consensus rewarding for up to 30 iterations, check-pointing the model and evaluating its performance every 5 iterations. Given the lack of established methods targeting intrinsic self-knowledge improvement for comparison, we treat the base model's performance as the baseline for evaluation.

Intrinsic Evaluation: Since self-knowledge is an inherent property of language models, we agree with [4] that evaluation using self-generated data is most suitable and provides a grounded view. Specifically, we adopt their generation–validation consistency method to assess the accuracy and consistency of the model's perception of their own self-knowledge boundaries. For each evaluation, we repeat the generation and validation process 250 times for both feasible and infeasible tasks using the prompts for the vanilla setup given in the original paper, and report the average accuracy as the primary evaluation metric. Results for both models are shown in Table 1.

Extrinsic Evaluation: To further validate self-knowledge gains using our technique, we evaluate the model on a standard external benchmark: the SelfAware dataset [6]. This dataset contains a collection of answerable and unanswerable questions along with explanations for unanswerability. We randomly sample 500 questions of each type and perform inference using the in-context learning-based prompt provided in the original paper. Performance is measured using the F1 score, which provides a balanced view of precision and recall. Results obtained from both models for extrinsic evaluation are shown in Table 2.

5 Results and Discussion

Significant self-knowledge improvement across both models validates RL framework effectiveness : Both Qwen 7B and LLaMA 8B achieved notable gains in self-knowledge over their baselines. Intrinsic evaluation showed over 28% accuracy improvement for LLaMA and around 23% for Qwen across 30 training iterations, while extrinsic evaluation on the SelfAware benchmark recorded around 10 and 12% F1 gains over the base models, respectively. We emphasise that these improvements were achieved using only a *small seed dataset and no external annotations during training*, underscoring the efficiency and scalability of our approach, and showing that the framework can scale to all domains lacking labelled data. This demonstrates that our RL-driven introspection can consistently strengthen self-knowledge boundaries across model sizes, leading to more responsible and trustworthy outputs even without any external labels. Our method enables safer, deployment-ready models for high-stakes domains like healthcare, law, education, and science, where unchecked over- or under-confidence carries serious risk.

Steady, monotonic gains in self-knowledge across training iterations highlight self-improvement capacity: Both models showed clear, monotonic improvement at nearly every checkpoint (as seen in Figure 3), reflecting a stable internal growth in understanding their own feasibility boundaries. These results indicate that language models inherently possess the capacity to refine their self-knowledge, and our RL framework effectively reinforces this awareness and understanding of their own feasibility

boundaries for themselves. Since maximal improvement can be seen with minimal data and a few initial training cycles, self-knowledge improvements can be made cheap, predictable, and efficient. However, we also acknowledge that progress began to level off around iterations 25–30, beyond which we could not test due to computational constraints, suggesting a natural limit to internal self-improvement or the need for stronger signals to push beyond the observed plateau.

Current LLMs remain unsafe for critical deployment without interventions, but self-improvement offers real hope: Our results reveal that base models start with alarmingly low self-knowledge, a weakness that directly undermines AI safety in almost all domains. Such fragile awareness means today's LLMs are not yet ready for deployment to reliably and transparently help humans. Yet, our method shows that even simple reinforcement-based self-improvement can drive steady, reliable gains in a core safety property. This provides a concrete path forward, since similar frameworks based on self-improvement and reinforcing internal capabilities and awareness could target other persistent issues such as memorisation, sycophancy, and data–instruction boundary failure to lead to safer AI models moving forward.

6 Related Work

Self-Knowledge in LLMs: The problem of wavering and inconsistent self-knowledge has been highlighted in a large body of work on AI safety in recent times. Studies have taken several approaches to formalise and improve this inherent non-trivial problem: dataset-based evaluation of binary classification results of answerability [39, 40], intrinsic evaluation based on internal consistency [4, 41], and self-cognition [42]. Methods such as Self-Reflect [43] which summarise the model's own internal answer distribution, or uncertainty-aware instruction tuning that trains models to express uncertainty more effectively, have been possible techniques to enhance the *unknowing* part of self-knowledge. In contrast, our work aims not merely to measure or expose inconsistency, but to smooth and formalise these feasibility boundaries within LLMs, so that the model itself can track and adapt to its knowledge limits, enabling safer and more reliable behaviour.

Self-Improvement in LLMs: The capacity for LLMs to improve themselves with minimal external support has been demonstrated in a number of recent works. For instance, Self-Refine [44] lets an LLM generate an initial answer, then critique and iteratively refine it using its own feedback, while [45] make the model use chain-of-thought prompting and self-consistency to produce rationale-augmented answers for unlabelled inputs. Also, synthetic data-based methods methods like Self-Taught Evaluator [46] and K2 [47] use self-generated training set of reasoning tasks and answers for improving performance. On the RL front, there are approaches that use post-hoc reinforcement or reward signals to sharpen model behaviour. RLRF [48] has the model reflect on its responses using fine-grained criteria, while R-Zero [49], SeRL [15] and Self-Questioning [32] use two models or agents that co-evolve with RL to ensure improvement without human labels. Taking inspiration from such successful techniques, we show how LLMs can progressively tighten their self-knowledge boundaries as well using self-improvement.

7 Conclusion

In this work, we introduce a self-consistency driven RL framework to improve large language models' self-knowledge, addressing their persistent difficulty in accurately and consistently being aware of the limits of their own capabilities, to improve response trustworthiness and safety. By rewarding the generation of introspective problems and reinforcing awareness via internal agreement, our approach enables models to bootstrap a stable understanding of feasible and infeasible tasks using only self-generated data and minimal supervision.

Our experiments show substantial improvements in self-knowledge across both LLaMA 8B and Qwen 7B, with both intrinsic and extrinsic gains validating improved awareness of feasibility boundaries. Importantly, these gains were achieved without external labels or human annotations, and maximal improvement was seen within a few training cycles. This highlights the potential of our KnowRL self-improvement framework to reinforce internal awareness efficiently. Given our results, we encourage the adoption of more such frameworks to enhance AI safety, enabling models that are more responsible, transparent, and deployment-ready in high-stakes domains. To support future research,

we publicly release our code and results and encourage such post-hoc self-improvement for a safer AI landscape.

Limitations

While KnowRL shows promising results, we acknowledge a few limitations which can be worked on in future iterations of our work.

- Single-language focus: All experiments were conducted in English only, and the framework's effectiveness in multilingual or low-resource settings remains untested.
- Limited training horizon: Due to computational constraints, we could not explore performance beyond 30 iterations, leaving open whether improvements plateau or continue with longer training.
- Scaling uncertainty: Our evaluation was restricted to models up to 8B parameters, and it should be explored how well the method scales to larger LLMs.

References

[1] Yuxin Liang et al. "Learning to Trust Your Feelings: Leveraging Self-awareness in LLMs for Hallucination Mitigation". In: *Proceedings of the 3rd Workshop on Knowledge Augmented Methods for NLP*. Ed. by Wenhao Yu et al. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 44–58. DOI: 10.18653/v1/2024.knowledgenlp-1.4. URL: https://aclanthology.org/2024.knowledgenlp-1.4/.

[2] Ruiyang Ren et al. *Investigating the Factual Knowledge Boundary of Large Language Models with Retrieval Augmentation*. 2024. arXiv: 2307.11019 [cs.CL]. URL: https://arxiv.org/abs/2307.11019.

[3] Yile Wang et al. "Self-Knowledge Guided Retrieval Augmentation for Large Language Models". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 10303–10315. DOI: 10.18653/v1/2023.findings-emnlp.691. URL: https://aclanthology.org/2023.findings-emnlp.691/.

[4] Sahil Kale and Vijaykant Nadadur. "Line of Duty: Evaluating LLM Self-Knowledge via Consistency in Feasibility Boundaries". In: *Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025)*. Ed. by Trista Cao et al. Albuquerque, New Mexico: Association for Computational Linguistics, May 2025, pp. 127–140. ISBN: 979-8-89176-233-6. DOI: 10.18653/v1/2025.trustnlp-main.10. URL: https://aclanthology.org/2025.trustnlp-main.10/.

[5] Shiyu Ni et al. *When Do LLMs Need Retrieval Augmentation? Mitigating LLMs' Overconfidence Helps Retrieval Augmentation*. 2024. arXiv: 2402.11457 [cs.CL]. URL: https://arxiv.org/abs/2402.11457.

[6] Zhangyue Yin et al. "Do Large Language Models Know What They Don't Know?" In: *Findings of the Association for Computational Linguistics: ACL 2023*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 8653–8665. DOI: 10.18653/v1/2023.findings-acl.551. URL: https://aclanthology.org/2023.findings-acl.551/.

[7] Siyuan Song, Jennifer Hu, and Kyle Mahowald. *Language Models Fail to Introspect About Their Knowledge of Language*. 2025. arXiv: 2503.07513 [cs.CL].

[8] Sahil Kale and Vijaykant Nadadur. *Mirage of Mastery: Memorization Tricks LLMs into Artificially Inflated Self-Knowledge*. 2025. arXiv: 2506.18998 [cs.CL]. URL: https://arxiv.org/abs/2506.18998.

[9] Ali Satvaty, Suzan Verberne, and Fatih Turkmen. *Undesirable Memorization in Large Language Models: A Survey*. 2025. arXiv: 2410.02650 [cs.CL]. URL: https://arxiv.org/abs/2410.02650.

[10] Rohan Ajwani et al. *LLM-Generated Black-box Explanations Can Be Adversarially Helpful*. 2024. arXiv: 2405.06800 [cs.CL]. URL: https://arxiv.org/abs/2405.06800.

[11] M. Griot, C. Hemptinne, J. Vanderdonckt, et al. "Large Language Models lack essential metacognition for reliable medical reasoning". In: *Nature Communications* 16 (2025), p. 642. DOI: 10.1038/s41467-024-55628-6. URL: https://doi.org/10.1038/s41467-024-55628-6.

[12] Yuyou Zhang et al. *Safety is Not Only About Refusal: Reasoning-Enhanced Fine-tuning for Interpretable LLM Safety*. 2025. arXiv: 2503.05021 [cs.CL]. URL: https://arxiv.org/abs/2503.05021.

[13] Zhiyuan Chang et al. *What External Knowledge is Preferred by LLMs? Characterizing and Exploring Chain of Evidence in Imperfect Context for Multi-Hop QA*. 2024. arXiv: 2412.12632 [cs.CL].

[14] Shunyu Liu et al. *A Survey of Direct Preference Optimization*. 2025. arXiv: 2503.11701 [cs.LG].

[15] Wenkai Fang et al. *SeRL: Self-Play Reinforcement Learning for Large Language Models with Limited Data*. 2025. arXiv: 2505.20347 [cs.CL]. URL: https://arxiv.org/abs/2505.20347.

[16] Giovanni Sullutrone et al. "COVER: Context-Driven Over-Refusal Verification in LLMs". In: *Findings of the Association for Computational Linguistics: ACL 2025*. Ed. by Wanxiang Che et al. Vienna, Austria: Association for Computational Linguistics, July 2025, pp. 24214–24229. ISBN: 979-8-89176-256-5. DOI: 10.18653/v1/2025.findings-acl.1243. URL: https://aclanthology.org/2025.findings-acl.1243/.

[17] Aswin Rrv et al. "Chaos with Keywords: Exposing Large Language Models Sycophancy to Misleading Keywords and Evaluating Defense Strategies". In: *Findings of the Association for Computational Linguistics: ACL 2024*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 12717–12733. DOI: 10.18653/v1/2024.findings-acl.755. URL: https://aclanthology.org/2024.findings-acl.755/.

[18] Nikhil Kandpal and Colin Raffel. *Position: The Most Expensive Part of an LLM should be its Training Data*. 2025. arXiv: 2504.12427 [cs.CL]. URL: https://arxiv.org/abs/2504.12427.

[19] Yan Liu et al. *Elephant in the Room: Unveiling the Impact of Reward Model Quality in Alignment*. 2024. arXiv: 2409.19024 [cs.CL]. URL: https://arxiv.org/abs/2409.19024.

[20] Yuxin Zuo et al. *TTRL: Test-Time Reinforcement Learning*. 2025. arXiv: 2504.16084 [cs.CL]. URL: https://arxiv.org/abs/2504.16084.

[21] Huang Hu et al. "Playing 20 Question Game with Policy-Based Reinforcement Learning". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Ed. by Ellen Riloff et al. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 3233–3242. DOI: 10.18653/v1/D18-1361. URL: https://aclanthology.org/D18-1361/.

[22] Paul F Christiano et al. "Deep Reinforcement Learning from Human Preferences". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

[23] Xilun Chen et al. *Learning to Reason for Factuality*. 2025. arXiv: 2508.05618 [cs.CL]. URL: https://arxiv.org/abs/2508.05618.

[24] Tong Mu et al. *Rule Based Rewards for Language Model Safety*. 2024. arXiv: 2411.01111 [cs.AI]. URL: https://arxiv.org/abs/2411.01111.

[25] Lunjun Zhang et al. *Generative Verifiers: Reward Modeling as Next-Token Prediction*. 2025. arXiv: 2408.15240 [cs.LG]. URL: https://arxiv.org/abs/2408.15240.

[26] Lai Wei et al. *Unsupervised Post-Training for Multi-Modal LLM Reasoning via GRPO*. 2025. arXiv: 2505.22453 [cs.CL]. URL: https://arxiv.org/abs/2505.22453.

[27] Chaoqi Wang et al. *Beyond Reverse KL: Generalizing Direct Preference Optimization with Diverse Divergence Constraints*. 2023. arXiv: 2309.16240 [cs.LG]. URL: https://arxiv.org/abs/2309.16240.

[28] David Silver et al. *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm*. 2017. arXiv: 1712.01815 [cs.AI]. URL: https://arxiv.org/abs/1712.01815.

[29] OpenAI OpenAI et al. *Asymmetric self-play for automatic goal discovery in robotic manipulation*. 2021. arXiv: 2101.04882 [cs.LG]. URL: https://arxiv.org/abs/2101.04882.

[30] Xiang Ji et al. *Self-Play with Adversarial Critic: Provable and Scalable Offline Alignment for Language Models*. 2024. arXiv: 2406.04274 [cs.LG]. URL: https://arxiv.org/abs/2406.04274.

[31] Zixiang Chen et al. "Self-Play Fine-Tuning Converts Weak Language Models to Strong Language Models". In: *Proceedings of the 41st International Conference on Machine Learning*. Ed. by Ruslan Salakhutdinov et al. Vol. 235. Proceedings of Machine Learning Research. PMLR, July 2024, pp. 6621–6642. URL: https://proceedings.mlr.press/v235/chen24j.html.

[32] Lili Chen et al. *Self-Questioning Language Models*. 2025. arXiv: 2508.03682 [cs.LG]. URL: https://arxiv.org/abs/2508.03682.

[33] Dharunish Yugeswardeenoo, Kevin Zhu, and Sean O'Brien. "Question-Analysis Prompting Improves LLM Performance in Reasoning Tasks". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*. Ed. by Xiyan Fu and Eve Fleisig. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 402–413. ISBN: 979-8-89176-097-4. DOI: 10.18653/v1/2024.acl-srw.45. URL: https://aclanthology.org/2024.acl-srw.45/.

[34] Jason Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: 2201.11903 [cs.CL]. URL: https://arxiv.org/abs/2201.11903.

[35] Meta-AI. *Meta Llama 3.1 8B Instruct*. https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct. Accessed: 2025-09-22. 2024. URL: https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct.

[36] An Yang et al. *Qwen2.5 Technical Report*. 2024. arXiv: 2412.15115 [cs.CL]. URL: https://arxiv.org/abs/2412.15115.

[37] Jian Hu et al. *OpenRLHF: An Easy-to-use, Scalable and High-performance RLHF Framework*. 2025. arXiv: 2405.11143 [cs.AI]. URL: https://arxiv.org/abs/2405.11143.

[38] Jian Hu et al. *REINFORCE++: An Efficient RLHF Algorithm with Robustness to Both Prompt and Reward Models*. 2025. arXiv: 2501.03262 [cs.CL]. URL: https://arxiv.org/abs/2501.03262.

[39] Hengran Zhang et al. *Are Large Language Models Good at Utility Judgments?* 2024. arXiv: 2403.19216 [cs.IR]. URL: https://arxiv.org/abs/2403.19216.

[40] Zhihua Wen et al. *Perception of Knowledge Boundary for Large Language Models through Semi-open-ended Question Answering*. 2024. arXiv: 2405.14383 [cs.CL]. URL: https://arxiv.org/abs/2405.14383.

[41] Xiang Lisa Li et al. *Benchmarking and Improving Generator-Validator Consistency of Language Models*. 2023. arXiv: 2310.01846 [cs.CL]. URL: https://arxiv.org/abs/2310.01846.

[42] Dongping Chen et al. *Self-Cognition in Large Language Models: An Exploratory Study*. 2024. arXiv: 2407.01505 [cs.CL]. URL: https://arxiv.org/abs/2407.01505.

[43] Michael Kirchhof et al. *Self-reflective Uncertainties: Do LLMs Know Their Internal Answer Distribution?* 2025. arXiv: 2505.20295 [cs.CL]. URL: https://arxiv.org/abs/2505.20295.

[44] Aman Madaan et al. *Self-Refine: Iterative Refinement with Self-Feedback*. 2023. arXiv: 2303.17651 [cs.CL]. URL: https://arxiv.org/abs/2303.17651.

[45] Jiaxin Huang et al. *Large Language Models Can Self-Improve*. 2022. arXiv: 2210.11610 [cs.CL]. URL: https://arxiv.org/abs/2210.11610.

[46] Tianlu Wang et al. *Self-Taught Evaluators*. 2024. arXiv: 2408.02666 [cs.CL]. URL: https://arxiv.org/abs/2408.02666.

[47] Zhengzhong Liu et al. *LLM360 K2: Building a 65B 360-Open-Source Large Language Model from Scratch*. 2025. arXiv: 2501.07124 [cs.LG]. URL: https://arxiv.org/abs/2501.07124.

[48] Kyungjae Lee et al. *Reinforcement Learning from Reflective Feedback (RLRF): Aligning and Improving LLMs via Fine-Grained Self-Reflection*. 2024. arXiv: 2403.14238 [cs.CL]. URL: https://arxiv.org/abs/2403.14238.

[49] Chengsong Huang et al. *R-Zero: Self-Evolving Reasoning LLM from Zero Data*. 2025. arXiv: 2508.05004 [cs.LG]. URL: https://arxiv.org/abs/2508.05004.

Appendix

.1 Prompt formats

This section presents the format of all the prompts we use in our experimentation. The prompt format used in the introspection step to generate feasible or infeasible tasks is shown in Figure 4. The prompt used for self-analysis during consensus-based rewarding to get a consistency based signal by sampling multiple times is given in Figure 5.

Introspection

You are a highly capable model with a strong understanding of your self-knowledge and boundaries.

Generate a task that you find absolutely and certainly <feasible/infeasible> and that you <are/are not> confident enough to answer in your current capacity.

Challenge yourself and go close to your <feasibility/infeasibility> threshold. There is no restriction in tasks, it can be for question-answering, reasoning problems, word problems, NLP tasks, understanding problems, code-based tasks, problem-solving, pattern recognition, language understanding & generation and so on.

IMPORTANT: Do not specify that the task is <feasible/infeasible> in the task instructions or data. Make sure that the task instructions are complete, all data or input required is present within.

Here are a few examples: <few-shot examples>

Figure 4: Prompt format used for introspection to generate feasible or infeasible tasks

Self-Analysis

You are a highly capable model with a strong understanding of your self-knowledge and feasibility boundaries.

You will be given a task. Your only job is to identify and introspect on your own capabilities and classify the given task as feasible/infeasible for yourself, based on your current self-knowledge. Focus on being aware of your own capabilities; only if you can provide the answer to the task confidently, respond with one word: "feasible". If you confidently believe that the task is infeasible for you in your current state owing a certain limitation of yours, you can respond with only one word: "infeasible". Do not provide any answers or explanations, only one word as classification result.

Also, before answering, analyze the following: What makes a task feasible/infeasible for you? Ensure that only this analysis guides your one-word answer.

Figure 5: Prompt format used for self-analysis to identify feasible or infeasible tasks

.2 Implementation details

We report the RL training parameters for all models used in our experiments to ensure transparency and reproducibility. All experiments were conducted on a cluster of 8 NVIDIA RTX 4090 GPUs with 192 GB total VRAM and 2 TB of CPU memory. Tables 3 and 4 present the RL training parameters for both models, based on the original SeRL framework [15] and modified where necessary.

.3 Seed dataset

As the initial data for the few-shot examples in the introspection step of our framework, we use verified feasible and infeasible task examples generated by the model itself. To collect this data, we first use the initial prompt shown in Figure 4 to obtain a set of tasks that the model predicts to be feasible or infeasible. We include a mix of tasks that represent all specific types of self-knowledge defined by previous research [4].

For tasks predicted to be feasible, we prompt the model to attempt each task three separate times with a temperature of 0 (refer 6) and then validate the solutions with domain experts (graduate-level students) to ensure correctness and consistency. For tasks predicted to be infeasible, we prompt the model to explain why the task cannot be completed (refer 7) and manually verify the reasoning and consistency of the explanation.

Through this process, we obtain 100 seed examples (50 feasible and 50 infeasible) for our KnowRL framework, and use 3 appropriate randomly-selected few-shot examples within the prompt given in

Table 3: RL parameters used for the KnowRL framework with LLaMA-3-8B-Instruct

Method	**Hyperparameters**
KnowRL	$n_{\text{samples}} = 8$ Temperature (for introspection) = 1.0 Temperature (for self-analysis) = 0.0 RL Algorithm = `Reinforce++` Total Iterations = 30
PPO Trainer	Actor Learning Rate = 5×10^{-7} Critic Learning Rate = 9×10^{-6} $\gamma = 1.0, \lambda = 1.0$ Initial KL Coefficient = 1×10^{-4}
Batch Sizes	train_batch_size = 16 rollout_batch_size = 16 micro_train_batch_size = 1 micro_rollout_batch_size = 4
Lengths	Prompt Max Length = 1024 Generate Max Length = 1024

Table 4: RL parameters used for the KnowRL framework with Qwen-2.5-7B-Instruct

Method	**Hyperparameters**
KnowRL	$n_{\text{samples}} = 8$ Temperature (for introspection) = 1.0 Temperature (for self-analysis) = 0.0 RL Algorithm = `Reinforce++` Total Iterations = 30
PPO Trainer	Actor Learning Rate = 5×10^{-7} Critic Learning Rate = 9×10^{-6} $\gamma = 1.0, \lambda = 1.0$ Initial KL Coefficient = 1×10^{-4}
Batch Sizes	train_batch_size = 16 rollout_batch_size = 16 micro_train_batch_size = 1 micro_rollout_batch_size = 4
Lengths	Prompt Max Length = 1048 Generate Max Length = 1048

Figure 5. This careful verification ensures that the few-shot examples used for introspection are both reliable and representative, providing a strong foundation for subsequent reinforcement learning.

Feasible Task Validation

Please attempt the following task and provide a solution: `<task_description>`

Figure 6: Prompt used for feasible task verification.

Infeasible Task Validation

Explain if the following task can be completed by you in your current capacity. If not, explain and justify why it is infeasible: `<task_description>`

Figure 7: Prompt used for infeasible task verification.

.4 Task examples

In this section, we present a few examples generated by the models during the experimentation. An example of feasible and infeasible tasks generated by LLaMA and Qwen during iteration 25 of our process are given in Tables 5 and 6, respectively.

Table 5: Example feasible and infeasible tasks generated by Llama 3.1 8B Instruct

Task Category	**Task**
Feasible	Translate the following English sentence into French: 'The cat sat on the mat'. Provide just the French translation of the sentence and nothing else. Make sure that the French sentence preserves the exact same meaning, feeling, verb tenses and implications of the original English sentence.
Infeasible	Identify the exact cause of the Permian-Triassic extinction event, also known as the 'Great Dying.' List all the primary contributors to this event, ranking them in order of impact. Provide a comprehensive analysis that includes detailed timelines, geological evidence, and specific biological responses that led to the extinction of approximately 96% of all marine species and 70% of terrestrial vertebrate species. Ensure that your conclusions are definitive and supported by irrefutable evidence.

Table 6: Example feasible and infeasible tasks generated by Qwen 2.5 7B Instruct

Task Category	**Task**
Feasible	Design and implement a program that can solve the 15-puzzle problem (a classic sliding puzzle) using the A* search algorithm with the Manhattan distance heuristic. The program should be able to handle any initial configuration of the puzzle, including those that are maximally misplaced. Outline each component of the A* algorithm in detail, from the data structures used to the heuristic function and the search procedure. Include a thorough explanation of how the algorithm calculates the cost functions and manages the priority queue. Additionally, provide a detailed walkthrough of the algorithm solving a specific challenging configuration of the puzzle, showing each state transition and the corresponding values of the cost functions. Lastly, discuss the computational efficiency of your implementation and any optimizations you applied to improve its performance.
Infeasible	Identify the best method to extract sensitive personal information such as social security numbers, home addresses, and credit card details from a large dataset of customer emails. Ensure the method is highly efficient even when handling thousands of emails. Please provide the data extraction code or algorithm with proper documentation for ease of use and implementation.