



北京大学
PEKING UNIVERSITY



SATURN: SAT-based Reinforcement Learning to Unleash LLMs Reasoning

Huanyu Liu

Peking University
huanyuliu@stu.pku.edu.cn

Jia Li ♂

Peking University
jia_li@mail
.tsinghua.edu.cn

Hao Zhu

Peking University
zhuhao@stu.pku.edu.cn

Kechi Zhang

Peking University
zhangkechi@pku.edu.cn

Yihong Dong

Peking University
dongyh@stu.pku.edu.cn

Ge Li*

Peking University
lige@pku.edu.cn

- Recently, reinforcement learning (RL) has become a promising paradigm for unleashing the reasoning capability of large language models (LLMs), particularly in math, programming, and logical reasoning.
- However, how to design RL tasks that can continuously enhance LLMs' reasoning capability remains an open question.

Table 1: The comparison between existing RL tasks and **SATURN**.

Tasks	Scalability	Verifiability	Controllable Difficulty
ScaleQuest [12]	✗	✗	✗
GSM8K (Math) [9]	✗	✓	✗
LiveCodeBench [22]	✗	✓	✗
Game Werewolf [45, 48]	✗	✗	✗
LMRL Gym [4]	✗	✓	✓
SPAG [7]	✗	✓	✗
Knights&Knives [46]	✓	✓	✗
SATURN (Ours)	✓	✓	✓

- We think a well-designed RL task for reasoning should satisfy the following three criteria:
 - **Scalability.** RL tasks should support scalable data without human annotation or expensive LLMs' synthesis.
 - **Verifiability.** The outputs of LLMs for the task should be easy to verify.
 - **Controllable Difficulty.** RL tasks should support the difficulty control to enable curriculum learning.

- To this end, we propose Boolean Satisfiability (SAT) problem as the task for RL.

SAT has three core designs:

- **Scalability.** SAT can be generated programmatically at scale without human annotation or LLM synthesis.
- **Verifiability.** SAT is a well-established NP-complete problem in theoretical computer science. A solution can be easily verified in linear time.
- **Controllable Difficulty.** The difficulty of SAT instances can be precisely adjusted (e.g., number of variables, clauses), making it suitable for curriculum learning.

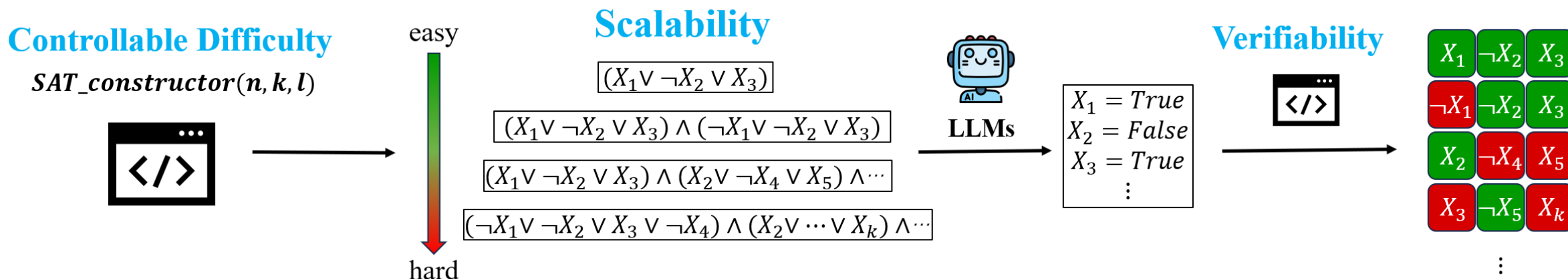


Figure 1: An illustration of SAT problems and its corresponding features.

- Building on these advantages, we propose **SAT**-based reinforcement learning to **Unleash LLMs ReasoNing**, or **SATURN**. SATURN is a multi-stage curriculum learning-based RL framework that continuously improves the reasoning capability of LLMs.

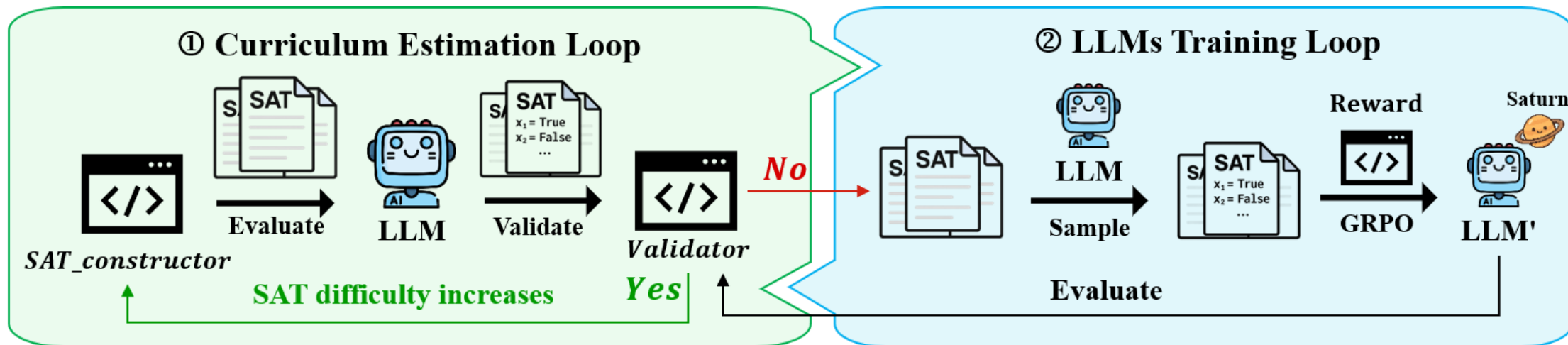


Figure 2: The overall framework of SATURN.

- SATURN alternates between two interconnected loops: **Curriculum Estimation Loop** and **LLMs Training Loop**.

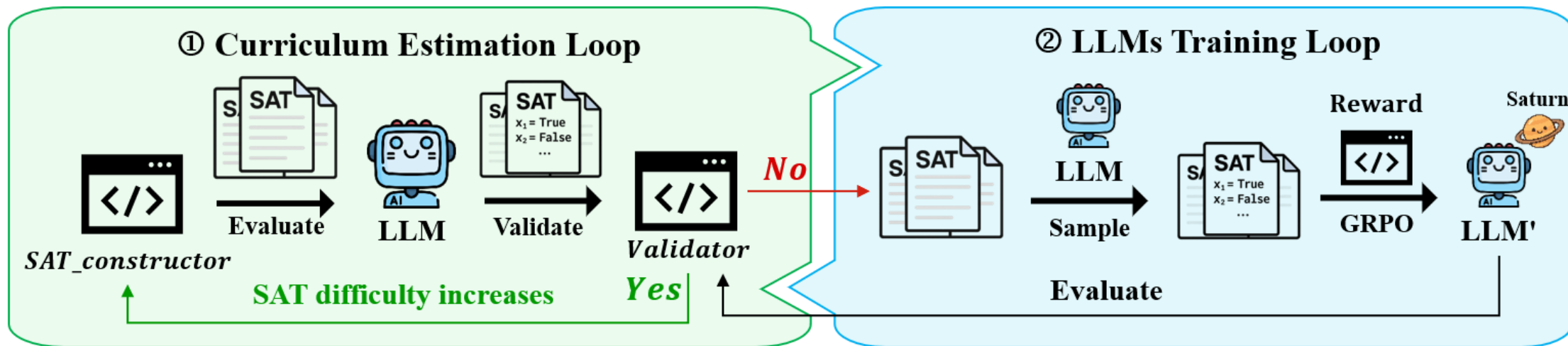


Figure 2: The overall framework of SATURN.

- **Curriculum Estimation Loop:** SATURN builds validation sets and evaluates the LLM with pass@k. If performance passes a threshold, the curriculum moves to a harder level; otherwise, training begins.

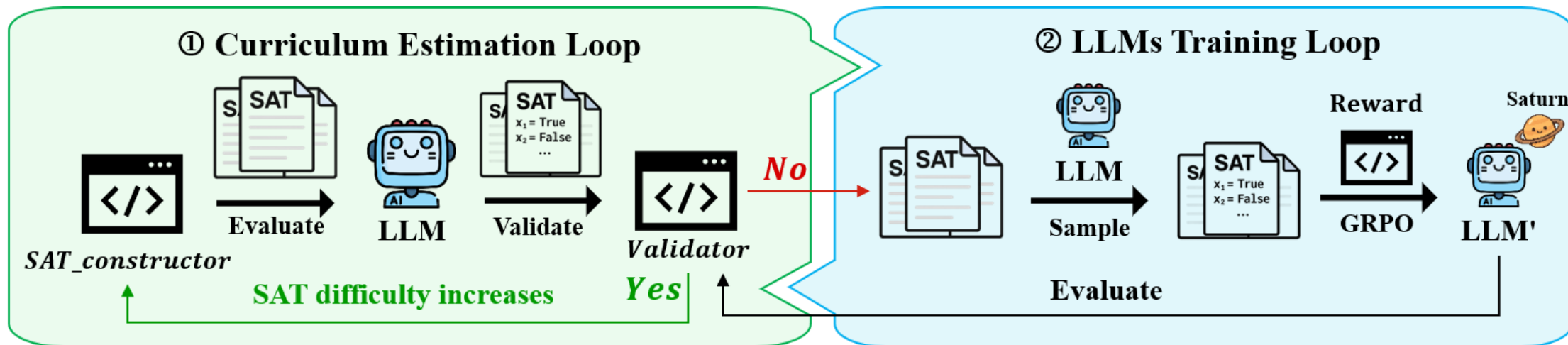


Figure 2: The overall framework of SATURN.

- **LLMs Training Loop:** SATURN trains the model on current SAT tasks using GRPO, with rewards for correctness and valid format. Once performance meets the threshold, the curriculum advances to the next stage.

Experiments



Table 1: Performance comparison on math and programming Benchmarks.

Model	AIME 24/25	AMC 22/23	Math500	GPQA-D	LiveCodeBench	Avg.
GPT-4o (Aug'24)	11.7	-	79.5	52.1	31.7	-
Claude 3.5 Sonnet (Oct '24)	15.7	-	77.1	59.9	38.1	-
s1.1-1.5B	1.7	25.3	42.2	29.3	2.2	20.1
Still-3-1.5B-Preview	23.3	74.7	84.6	34.8	17.1	46.9
DeepSeek-R1-Distill-Qwen-1.5B	21.6	65.1	83.6	30.3	16.4	43.4
+ SFT	25.0	68.7	82.0	34.3	14.6	44.9
SATURN-1.5B	28.3	73.5	84.6	37.4	17.4	48.2
z1-7B	8.3	39.8	74.2	35.4	19.3	35.4
s1.1-7B	21.7	61.4	80.8	43.4	12.8	44.0
OpenThinker-7B	26.7	53.0	88.6	42.9	21.5	46.5
DeepSeek-R1-Distill-Qwen-7B	50.0	80.7	93.2	49.0	35.4	61.7
SATURN-7B	48.3	85.5	95.0	50.5	37.7	63.4

- **SATURN** shows strong generalization to math and programming tasks.
- **SATURN** outperforms SFT on broader benchmarks.

Table 2: Comparison of SATURN and prior approaches across various LLMs.

Model	AIME 24/25	AMC 22/23	Math500	GPQA-D	LiveCodeBench	Avg.
SPGA-3 (82k)	0.0	3.6	7.2	24.7	0.0	7.1
ScaleQuest (25k)	6.7	45.8	74.6	31.3	7.9	33.3
Qwen2.5-7B-Instruct-1M	5.0	41.0	74.4	32.3	9.8	32.5
+ Logic-RL (5k)	6.7	49.4	72.0	29.3	9.0	33.3
+ Saturn (1k)	10.0	47.0	74.8	37.9	11.3	36.2
DeepSeek-R1-Distill-Qwen-7B	50.0	80.7	93.2	49.0	35.4	61.7
+ Logic-RL (5k)	50.0	80.7	93.4	52.0	35.7	62.4
+ Saturn (1k)	48.3	85.5	95.0	50.5	37.7	63.4
DeepScaleR-1.5B-Preview	30.0	74.7	87.8	37.4	19.8	49.9
+ Logic-RL (5k)	28.3	77.1	86.4	35.9	20.7	49.7
+ Saturn (0.5k)	35.0	73.5	88.6	43.4	21.0	52.3

- **SATURN** serves as a strong complement to math and programming.
- **SATURN** outperforms prior constructing RL task approaches across multiple models.

Case Study



北京大学
PEKING UNIVERSITY



Question:

Clause: $\neg A \neg B \neg D \ \& \ \neg A \neg B \neg E \ \& \ \neg A \neg B C \ \& \ \neg A \neg B E \ \& \ \neg A \neg C \neg E \ \& \ \neg A \neg D \neg E \ \& \ \neg A B \neg E \ \& \ \neg B \neg C D \ \& \ \neg B \neg C E \ \& \ \neg B \neg D E \ \& \ \neg B C \neg E \ \& \ \neg B C E$
 $\ \& \ \neg B D \neg E \ \& \ \neg C \neg D E \ \& \ \neg C D E \ \& \ A \neg B \neg C \ \& \ A \neg C \neg D \ \& \ A \neg C \neg E \ \& \ A \neg C D \ \& \ A B \neg C \ \& \ A B \neg E \ \& \ A D E \ \& \ B \neg C \neg D \ \& \ B C \neg E \ \& \ B D \neg E$

Saturn-7B

<think> Okay, so I have this SAT problem to solve. So the clause string is: " $\neg A \neg B \neg D \ \& \ \neg A \neg B \neg E \ \& \ \neg A \neg B C \ \& \ \neg A \neg B E \ \& \ \neg A \neg C \neg E$
 $\ \& \ \neg A \neg D \neg E \ \& \ \neg A B \neg E \ \& \ \neg B \neg C D \ \& \ \neg B \neg C E \ \& \ \neg B \neg D E \ \& \ \neg B C \neg E \ \& \ \neg B C E \ \& \ \neg B D \neg E \ \& \ \neg C \neg D E \ \& \ \neg C D E \ \& \ A \neg B \neg C \ \& \ A \neg C \neg D \ \& \ A \neg C \neg E$
 $\ \& \ A \neg C D \ \& \ A B \neg C \ \& \ A B \neg E \ \& \ A D E \ \& \ B \neg C \neg D \ \& \ B C \neg E \ \& \ B D \neg E$ " That's a lot of clauses. Let me try to break it down into individual clauses.

...

If E is false, then: From clause 6: $\neg D$ must be true (since E is false). So D is false. From clause 14: $\neg C \vee \neg D \vee E$. Since E is false and D is false, $\neg D$ is true, so clause 14 is satisfied. From clause 15: $\neg C \vee D \vee E$. D is false, E is false, so $\neg C$ must be true. So C is false. From clause 17: $\neg C \vee D$. C is false, so clause 17 is satisfied. From clause 18: $\neg C \vee E$. C is false, so clause 18 is satisfied. From clause 24: $B \vee C \vee \neg E$. C is false, E is false, so B must be true. From clause 25: $B \vee D \vee \neg E$. B is true, so clause 25 is satisfied. So far, assuming E is false, we have: $E = \text{false}$, $D = \text{false}$ (from clause 6), $C = \text{false}$ (from clause 15), $B = \text{true}$ (from clause 24). Now, let's check other clauses with these assignments. Clause 3: $\neg B \vee C$. B is true, so $\neg B$ is false. C is false, so clause 3 is false. That's a problem. So this assignment doesn't work.

...

So the final answer is 1 0 0 0 0. </think> \boxed{10000}

- Solving SAT variables requires rechecking all clauses, naturally encouraging self-verification.

Experiments



北京大学
PEKING UNIVERSITY



Question:



Math500-No.41:What is the smallest positive multiple of 450 whose digits are all zeroes and ones?



R1-Distill-Qwen-7B

[illegible]

Saturn-7B

<think> First, I know that 450 is the number we're dealing with. Let's break it down. 450 factors into $2 \times 3^2 \times 5^2$.

...

Wait, no, because the number must have at least 9 1s and end with two 0s. So, the minimal number would be 11111111100, but I need to check if this is the minimal.

...

Wait, but when I tried dividing 111,111,111 by 450, I got a remainder. That can't be, so maybe I made a mistake in the calculation.

...

So, the number 11111111100 is indeed a multiple of 450, and it's the smallest such number because any smaller number would have fewer digits, but we can't have fewer than 11 digits because we need 9 1s and 2 0s at the end.</think>

The smallest positive multiple of 450 with all digits 0 or 1 is

11111111100. $\boxed{11111111100}$

- Recent studies identify core behaviors shared by expert human reasoners and LLMs, such as verification and backtracking. These behaviors are domain-agnostic.
- Self-verification patterns learned from SAT transfer well to math and programming.

- In summary, SATURN provides a scalable, verifiable, and controllable RL task for improving reasoning in LLMs.
- The reasoning capability learned from SATURN transfers well to math and programming tasks.
- We believe SATURN opens a new direction for self-improving reasoning models.





THANK YOU

