Segment Policy Optimization: Effective Segment-Level Credit Assignment in RL for Large Language Models

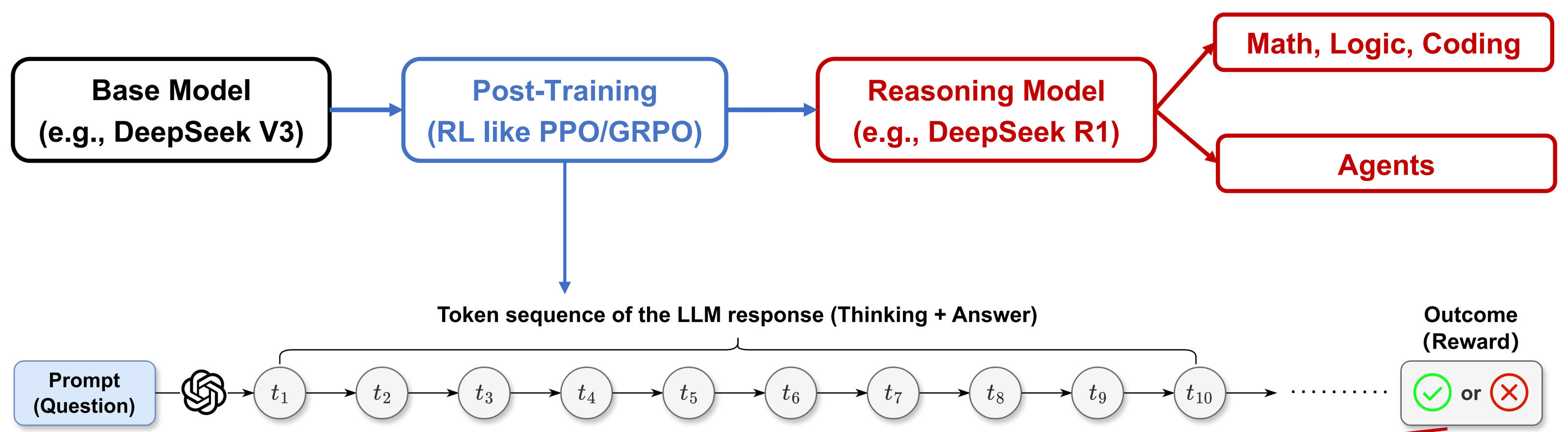
Yiran Guo[1], Lijie Xu[1*], Jie Liu[1*], Dan Ye[1], Shuang Qiu[2]

[1]Institute of Software, Chinese Academy of Sciences [2] City University of Hong Kong

Paper Code Covered by Jiqizhixin

Key problem and challenges of RL for LLM

LLM Training Pipeline



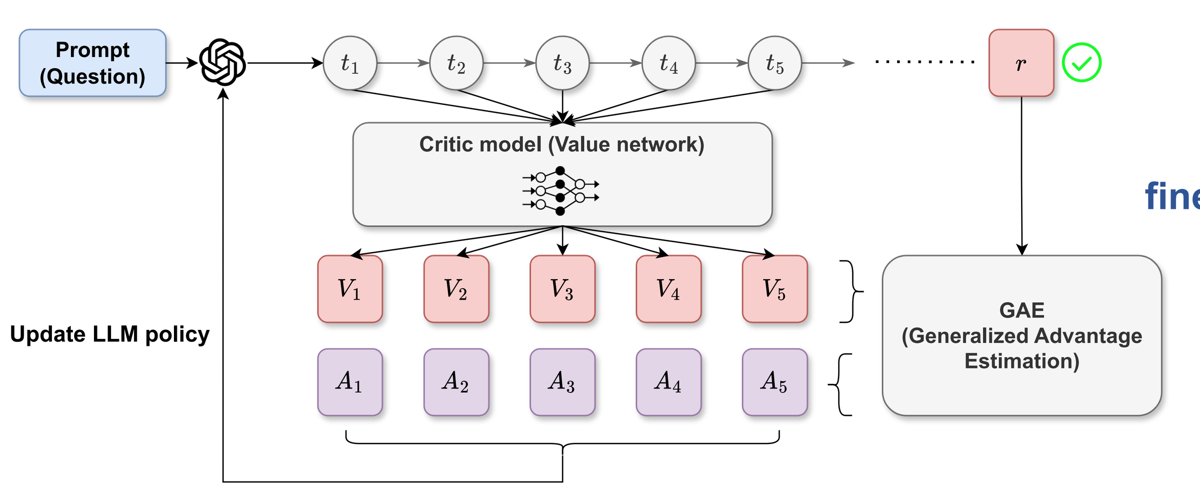
Key Problem: Credit assignment

How to attribute the final evaluation result (reward signal) of the entire sequence (LLM response) to the specific decision-making actions (i.e., tokens) within that sequence?

Challenges

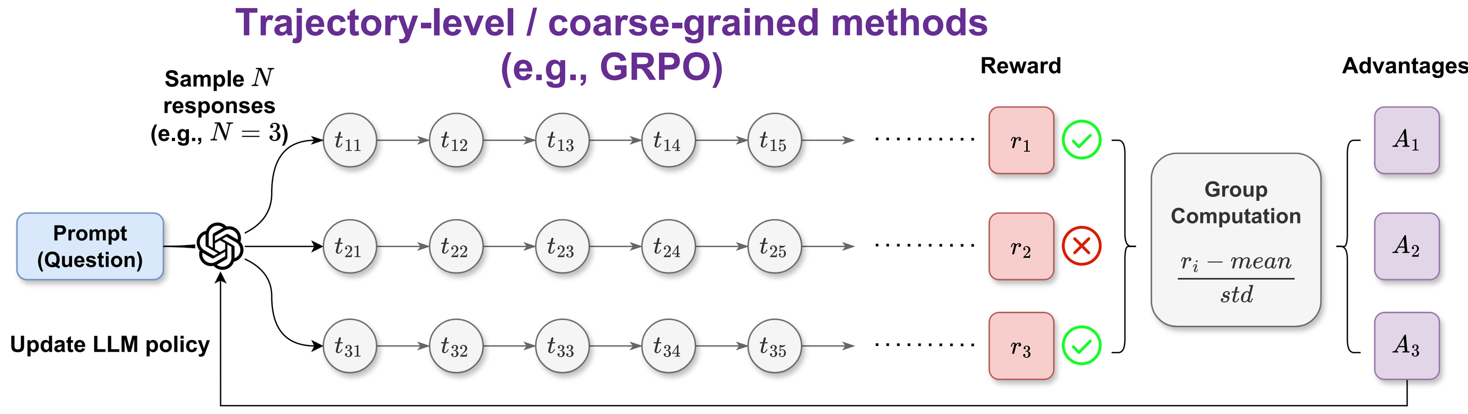
Different from traditional RL, the LLM reward is extremely sparse (clear success or failure feedback is only available at the end of the entire sequence).

Existing methods (Token-level vs. Trajectory-level)



Token-level / fine-grained methods (e.g., PPO)

Trajectory-level / coarse-grained methods (e.g., GRPO)



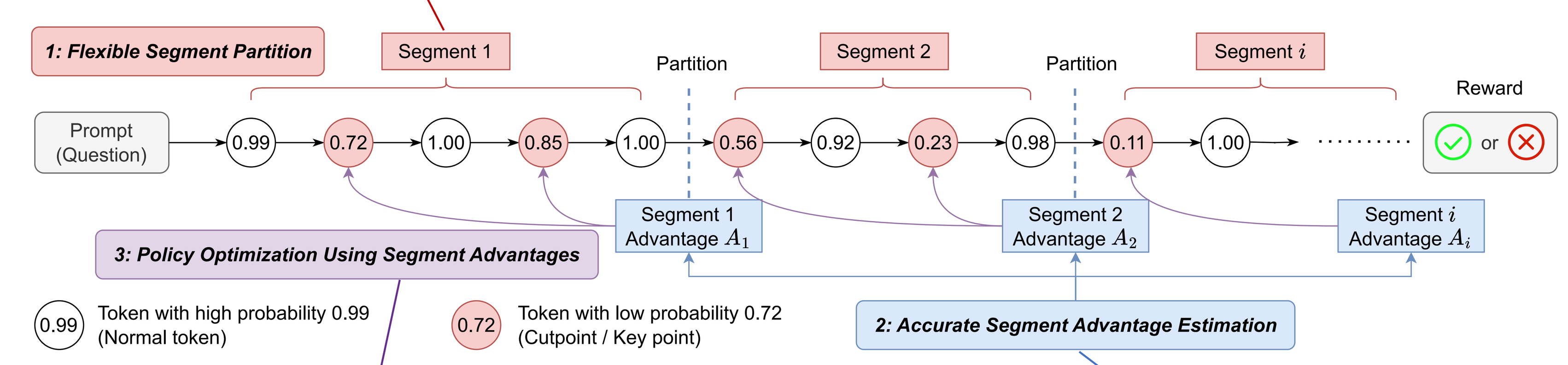
Method Granularity	Example	Mechanism	Drawbacks
Token-Level (Fine-Grained)	PPO	Uses a critic model to estimate advantage (A_i) for every token (t_i).	Inaccurate critic: Hard to train the critic model, leading to unreliable value V predictions. High overhead: Critic model training/prediction.
Trajectory-Level (Coarse-Grained)	GRPO	Uses a single advantage signal (A_i) from the final reward for the entire sequence (each t_{ij} gets the same A_i).	Imprecise credit: Cannot reward partial progress or penalize redundancy. Overfitting: Performance on validation sets saturates early.

Our SPO framework (Segment-level)

Key idea: Mid-grained (segment-level) advantage estimation can unify and overcome the limitations of token-level and trajectory-level methods. (See Feature 1 and 2; Segment → a number of contiguous tokens)

Feature 1: Flexible segment partition (without requiring semantic completeness)
→ Unify token-level and trajectory-level methods (flexible adjustment of granularity from token-level to trajectory-level)

Our SPO framework with three components



Feature 3: Probability-mask optimization strategy (enhance the credit assignment)

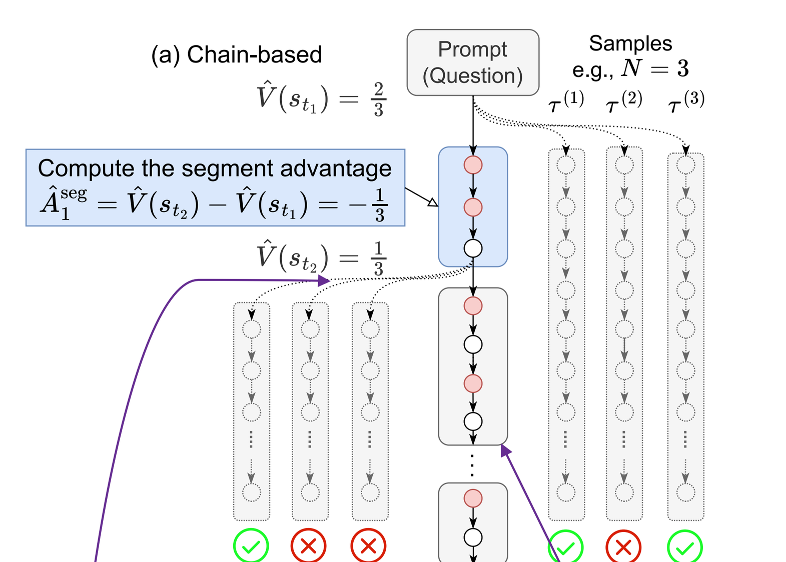
→ Selectively assign the segment advantages to critical (low-probability) tokens instead of all tokens within a segment.
→ Critical tokens (*cutpoints*) represent positions where the model's reasoning trajectory could diverge.

Feature 2: Accurate segment advantage estimation (based on Monte Carlo (MC) sampling)

→ SPO vs. PPO (SPO eliminates the need for an additional, unstable critic model.)
→ SPO vs. GRPO (SPO can reward partial progress for unsuccessful responses and penalize redundancy or unnecessary portions within successful responses.)

Two algorithms based on SPO

SPO-chain for short CoT

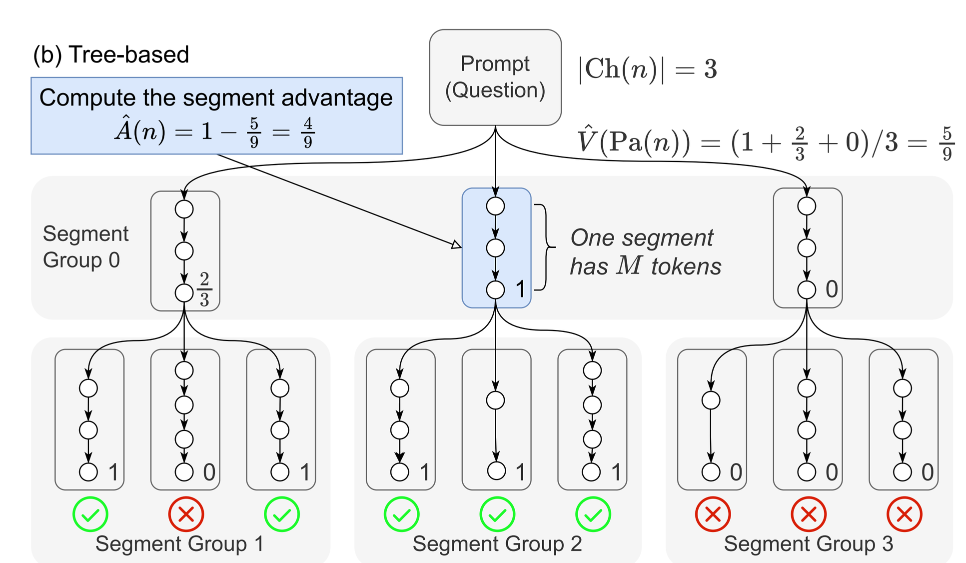


Features of SPO-chain:

(1) Adaptive cutpoint-based segment partition strategy: Let each segment contain a number of cutpoints, avoiding unnecessary segment (where all tokens within the segment have probabilities close to 1).

(2) Chain-based Monte Carlo sampling: For each segment, SPO independently samples N trajectories to estimate the value and segment advantage.

SPO-tree for long CoT



Features of SPO-tree:

(1) Fixed token count segment partition strategy: Each segment contains a fixed number of tokens (to well support tree-based sampling; unlikely all tokens within a long segment have probabilities close to 1).

(2) Tree-based segment advantage estimation: Directly yield segment advantages after tree-based trajectory generation, avoiding costly resampling and significantly improving efficiency for long CoT.

Example

Segment partition example (Each segment contains 5 cutpoints, i.e., red tokens)

Prompt: [MATH_TASK] Problem: Gloria wants to buy the $129,000 mountain cabin that her friend Alfonso is selling. She only has $150 in cash. She intends to raise the remaining amount by selling her mature trees for lumber. She has 20 cypress trees, 600 pine trees, and 24 maple trees. She will get $100 for each cypress tree, $300 for a maple tree, and $200 per pine tree. After paying Alfonso for the cabin, how much money will Gloria have left? Solution:

Segment 1 — Blue vertical line → segment boundary — Red tokens → cutpoints (tokens with probability < 0.9)

The total value of the cypress trees that she will | sell is 20 * $100 = $2000.
The total value of the maple trees that she will sell is | 600 * $300 = $18000.
The total value of the pine trees that she will sell is 24 * $200 = $4800.
The total value of all the trees that she will sell is $2000 + $18000 + $4800 = $2 4800.
| After selling all the trees, she will have $24800 - $150 = $23300 left.
23300

Problem: The total value of the maple trees should be 24 * $300, but the model outputs 600 * $300 here and shows low confidence at the digit 6.

Problem: The calculation should be $24800 - $12900, but the model instead outputs $24800 - $150 and is uncertain at the digit 5.

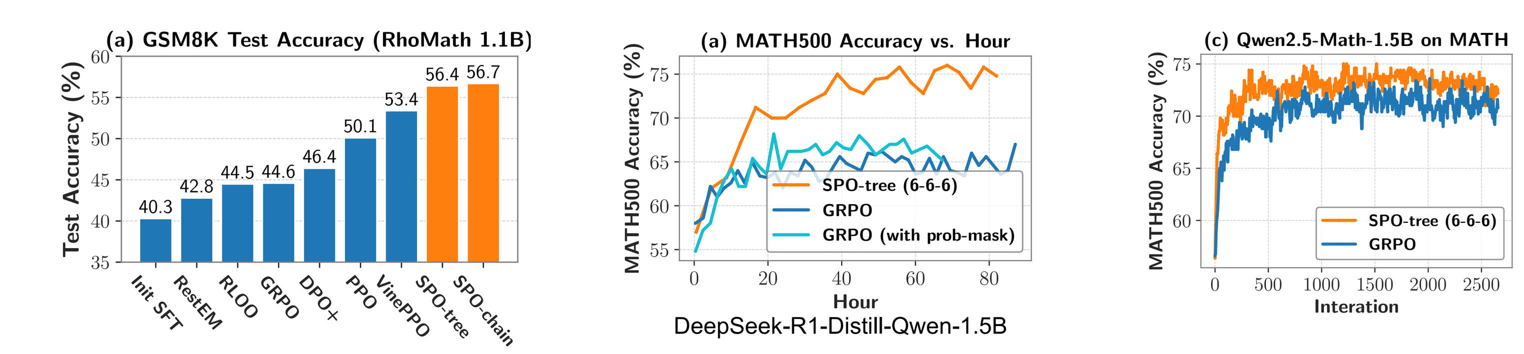
Implication: Cutpoints are the locations where the model's reasoning trajectory can shift, and they are the main drivers behind "segment advantage".

Cutpoint-based segment partition strategy (generate effective segments; See Features of SPO-chain)

Probability-mask optimization strategy (optimize critical tokens to enhance credit assignment; See Feature 3 of SPO)

Experiments

SPO-chain and SPO-tree outperform existing methods on GSM8K and MATH.

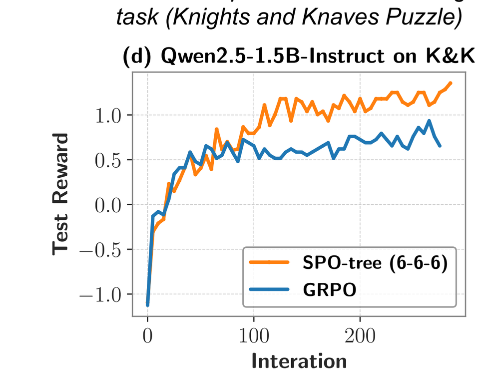


Accuracy comparison with various context sizes with DeepSeek-R1-Distill-Qwen-1.5B

Dataset	Eval Context Size	Base	GRPO	SPO-tree	DeepScaleR*	STILL-3*
MATH500	2K	0.566	0.62	**0.736**	0.538	0.662
	4K	0.74	0.752	**0.828**	0.744	0.794
	32K	0.838	0.84	0.848	**0.878**	0.846
AIME24	2K	0.067	0.033	**0.1**	0	0.067
	4K	0.167	**0.2**	**0.2**	0.167	0.133
	32K	0.267	**0.333**	**0.333**	**0.333**	0.233

* **These models were trained using larger datasets and longer context windows**. Specifically, DeepScaleR uses 8K→16K→24K context and STILL-3 sets 'generate_max_len' to 29000 during training, while our GRPO and SPO-tree use a maximum context of 4K during training.

SPO also outperforms GRPO on logic task (Knights and Knaves Puzzle)



Different partition granularities — *Cutpoint-based vs. Other partition strategies* — *Effectiveness of probability-mask strategy*