大道至简实现大语言模型思维链无损压缩
Occam’s Razor: Lossless CoT Compression for LLMs

Reporter: 吴太强 Taiqiang Wu

Supervisor: 黄毅 Dr. Ngai Wong

HKU EEE Ngai Lab

Mar. 2026

takiwu@connect.hku.hk

The Art of Efficient Reasoning: Data, Reward, and Optimization

Taiqiang Wu[♣♠*], **Zenan Xu**[♠*], **Bo Zhou**[♠†], **Ngai Wong**[♣†]
[♣]The University of Hong Kong, [♠]LLM Department, Tencent
chaysezhou@tencent.com, nwong@eee.hku.hk

https://wutaiqiang.github.io/project/Art

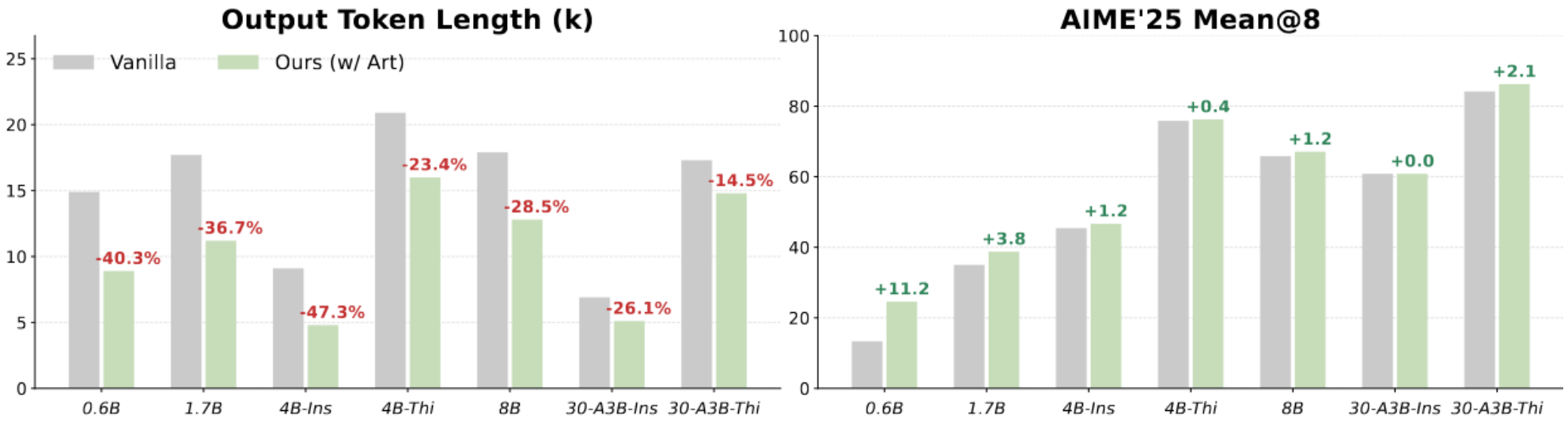


Figure 2: Performance comparison on AIME'25 for Qwen3 models ranging from 0.6B to 30B.

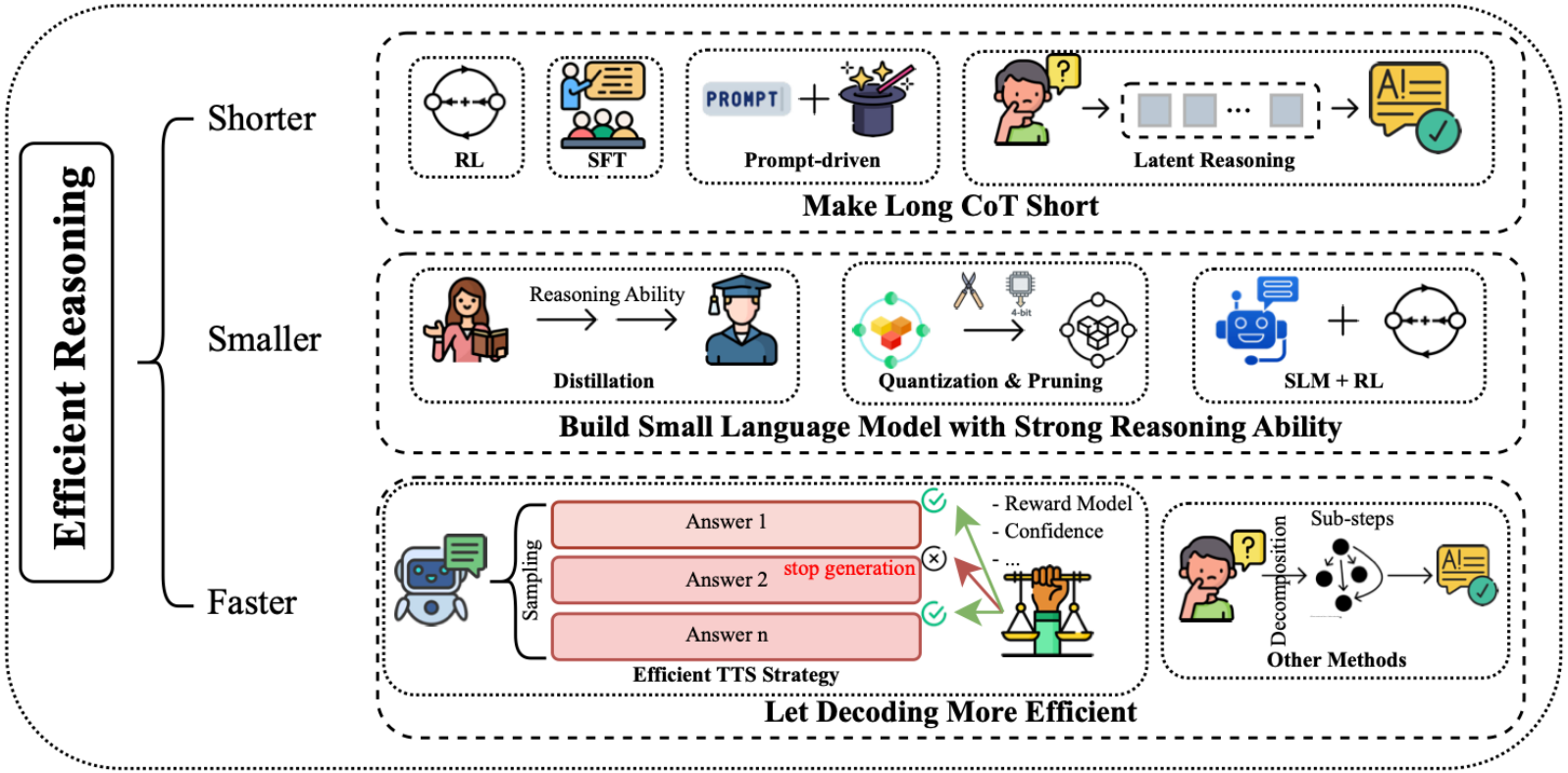
Agenda

- **背景介绍 Introduction**
  - 大语言模型思维链压缩 CoT Compression for LLMs
  - 奖励重塑方法 Reward Shaping Methods
- **数据, 奖励与优化 Data & Reward & Optimization**
  - 强化学习的数据 Data for RL
  - 负样本奖励分配 Reward Assignment for Negative Rollouts
  - 异策略优化 Off-policy Optimization via Staleness
- **Qwen3 实战 Qwen3 CoT Compression**
  - Qwen3系列压缩 Compression for Qwen3 0.6~30B
  - 调参建议 Suggestion for Hyper-parameters
- **总结与未来方向 Insights & Future Work**

Agenda

- **背景介绍 Introduction**
  - 大语言模型思维链压缩 CoT Compression for LLMs
  - 奖励重塑方法 Reward Shaping Methods
- **数据, 奖励与优化 Data & Reward & Optimization**
  - 强化学习的数据 Data for RL
  - 负样本奖励分配 Reward Assignment for Negative Rollouts
  - 异策略优化 Off-policy Optimization via Staleness
- **Qwen3 实战 Qwen3 CoT Compression**
  - Qwen3系列压缩 Compression for Qwen3 0.6~30B
  - 调参建议 Suggestion for Hyper-parameters
- **总结与未来方向 Insights & Future Work**

LLM 高效思考 Efficient reasoning for LLM



- 思维链压缩 CoT compression
- 人类从经验中找到捷径 We human do so

https://arxiv.org/pdf/2504.10903

奖励重塑实现高效思考

Reward shaping for efficient reasoning

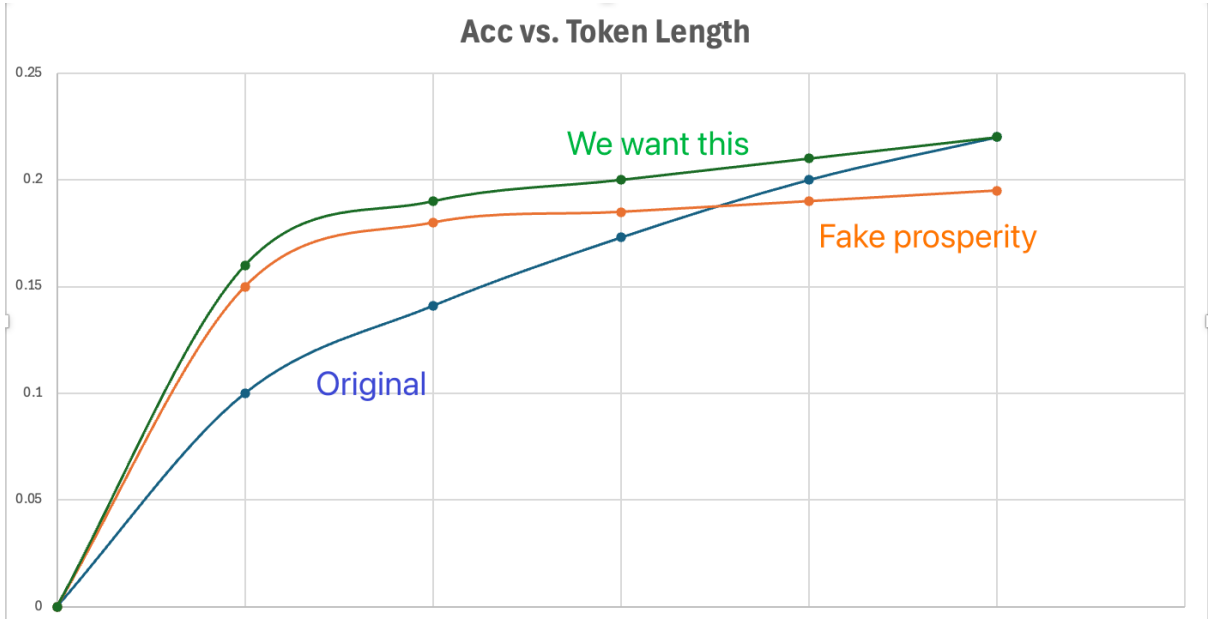
设计哲学 Philosophy

- 正确 CoT 奖励比错误的高 Correct CoTs receive higher rewards than wrong CoTs.
- 短+正确的思维链 奖励高于 长+正确的思维链 Shorter correct CoTs receive higher rewards than longer correct CoTs

https://lilianweng.github.io/posts/2025-05-01-thinking/

如何观测？How to observe?

给定budget 的性能 Performance under a fix budget 🤱♀



低 token下更好，但是上限拉低 ❌

Low budget better, but lower upper-bound

低 token下更好，上限不掉 ✅

Low budget better, and upper-bound keeps

-> 多 token 限制下的表现 2k,4k,8k,...,32k

Performance under multi-level budgets

如何观测？How to observe?

简单的数学题的性能 Performance on easy math benchmarks 🤦♀️

- 挑战性数学题 Challenge math benchmark: AIME’24 AIME’25
- 代码测试集 Code benchmark: LiveCodeBench/private benchmark
- 私有数据集 Private OOD benchmark: science/logic/multi-turn

模型 Models: Deepseek-distilled-qwen-1.5B & Qwen3 series

Two-Stage Paradigm

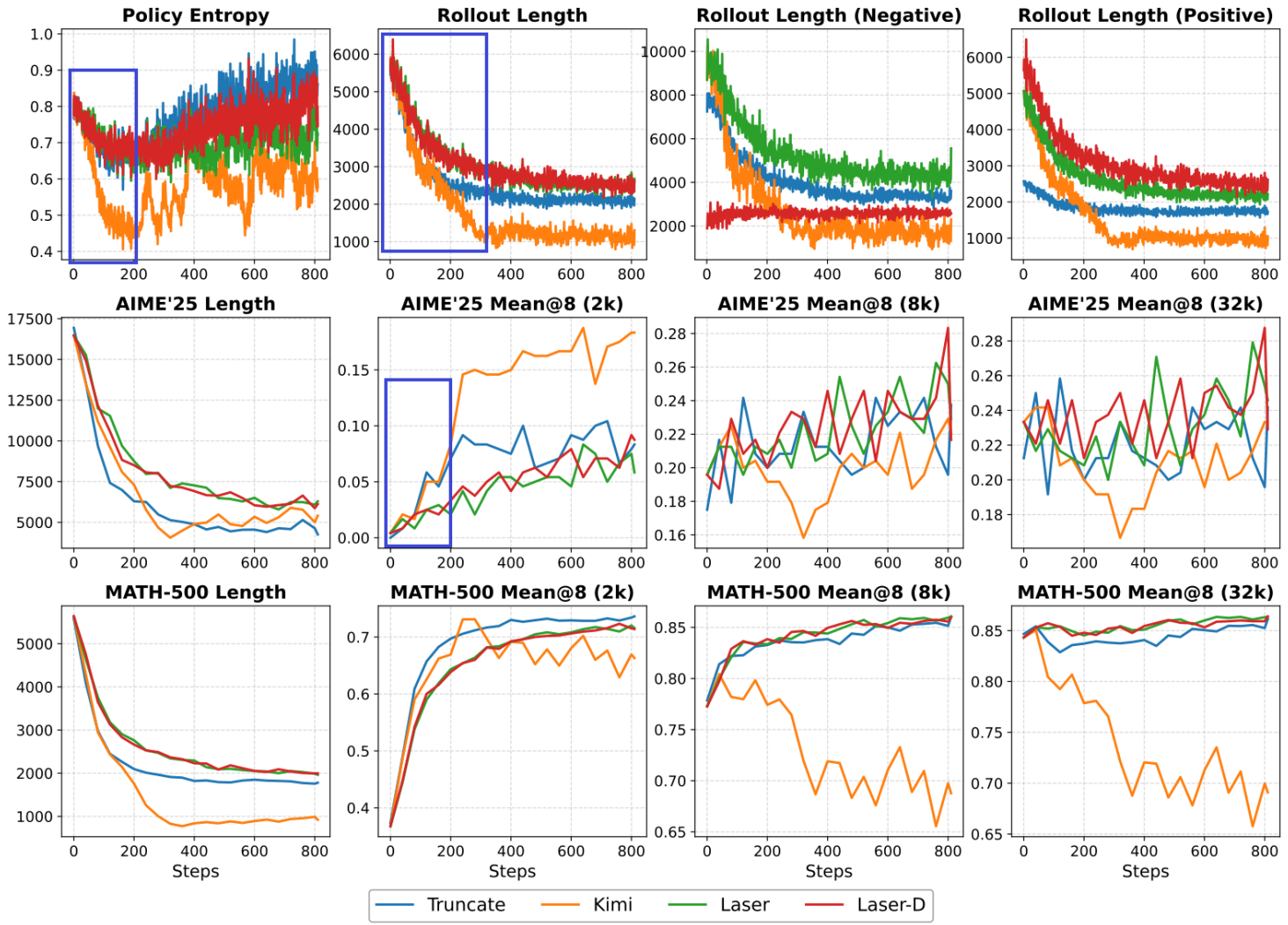


Figure 2: Training dynamics of various reward shaping methods on `DeepSeek-R1-Distill-Qwen-1.5B`. All of them follow the two-stage paradigm. The behaviors are distinct when evaluated under different token budgets.

Stage I: Length Adaptation

Policy Entropy/ Rollout Length 下降 decrease， 适应长度要求 fit length constrain first

$$reward_i := f(c_i) \rightarrow reward := g(c_i, l_i)$$

Stage II: Reasoning Refinement

Policy Entropy 上升 increase

长度开始震荡，效果拉升 stationary phase regarding length increase the information density of each token

Two-Stage Paradigm

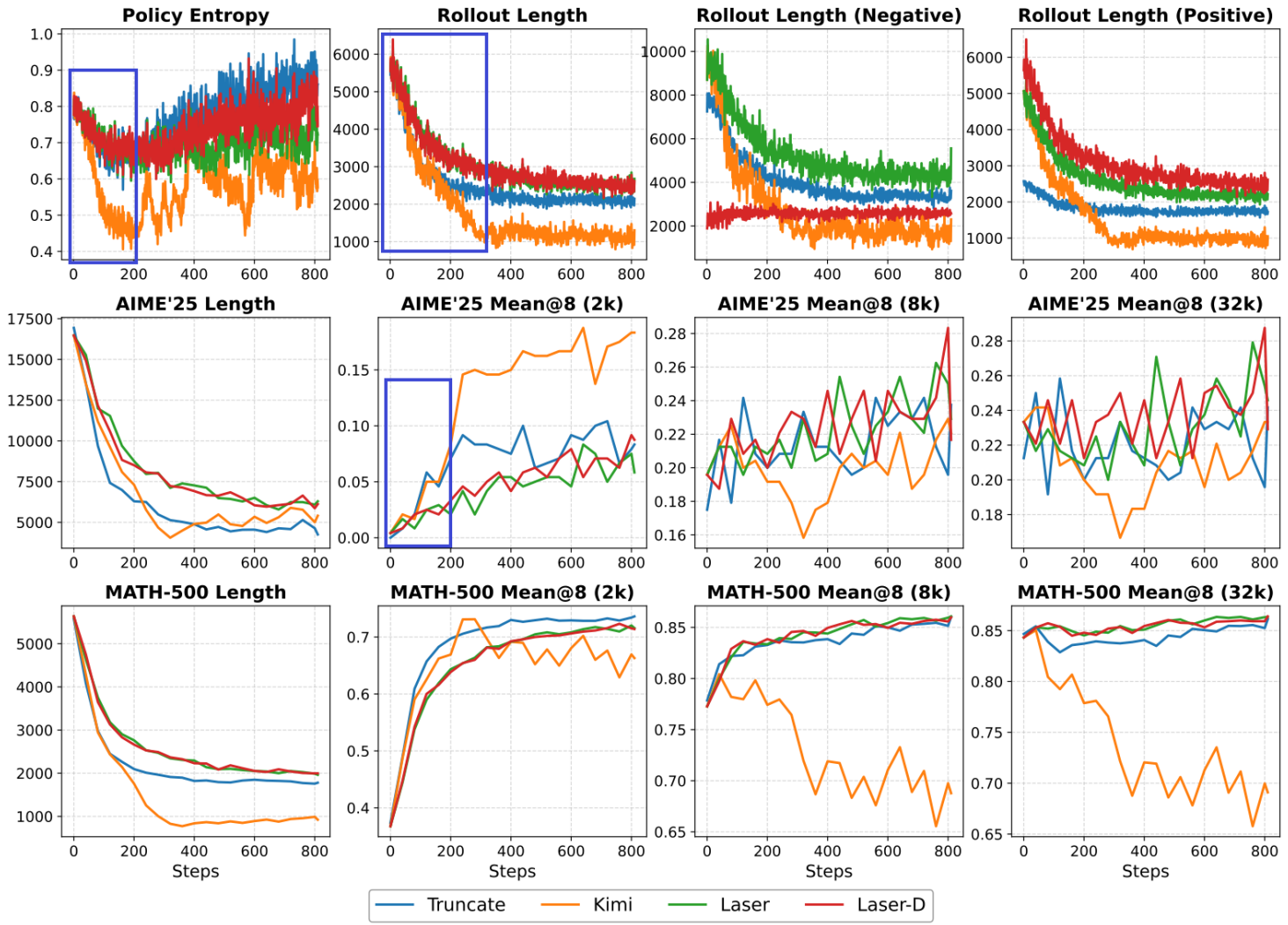


Figure 2: Training dynamics of various reward shaping methods on `DeepSeek-R1-Distill-Qwen-1.5B`. All of them follow the two-stage paradigm. The behaviors are distinct when evaluated under different token budgets.

$$R_T(x, y_i) = \mathbb{I}(y_i \text{ is correct}) \cdot \mathbb{I}(L(y_i) \le L_T), \tag{2}$$

$$R_{\text{Kimi}}(x, y_i) = \mathbb{I}(y_i \text{ is correct}) \cdot (1 + \alpha(0.5 - \tilde{L}(y_i))) + \mathbb{I}(y_i \text{ is incorrect}) \cdot \min\left(0, \alpha(0.5 - \tilde{L}(y_i))\right). \tag{4}$$

$$R_{\text{Laser}}(x, y_i) = \mathbb{I}(y_i \text{ is correct}) \cdot (1 + \alpha \cdot \mathbb{I}(L(y_i) < L_T)). \tag{5}$$

$$R_{\text{Laser-D}}(x, y_i) = \mathbb{I}(y_i \text{ is correct}) \cdot (1 + \alpha \cdot \mathbb{I}(L(y_i) < L_T)) + \mathbb{I}(y_i \text{ is incorrect}) \cdot (\alpha \cdot \mathbb{I}(L(y_i) \ge L_T)). \tag{6}$$

简单的截断策略表现不错

Simple truncation performs well

Agenda

- 背景介绍 **Introduction**
  - 大语言模型思维链压缩 CoT Compression for LLMs
  - 奖励重塑方法 Reward Shaping Methods
- **数据, 奖励与优化 Data & Reward & Optimization**
  - 强化学习的数据 Data for RL
  - 负样本奖励分配 Reward Assignment for Negative Rollouts
  - 异策略优化 Off-policy Optimization via Staleness
- **Qwen3 实战 Qwen3 CoT Compression**
  - Qwen3系列压缩 Compression for Qwen3 0.6~30B
  - 调参建议 Suggestion for Hyper-parameters
- 总结与未来方向 **Insights & Future Work**

Data

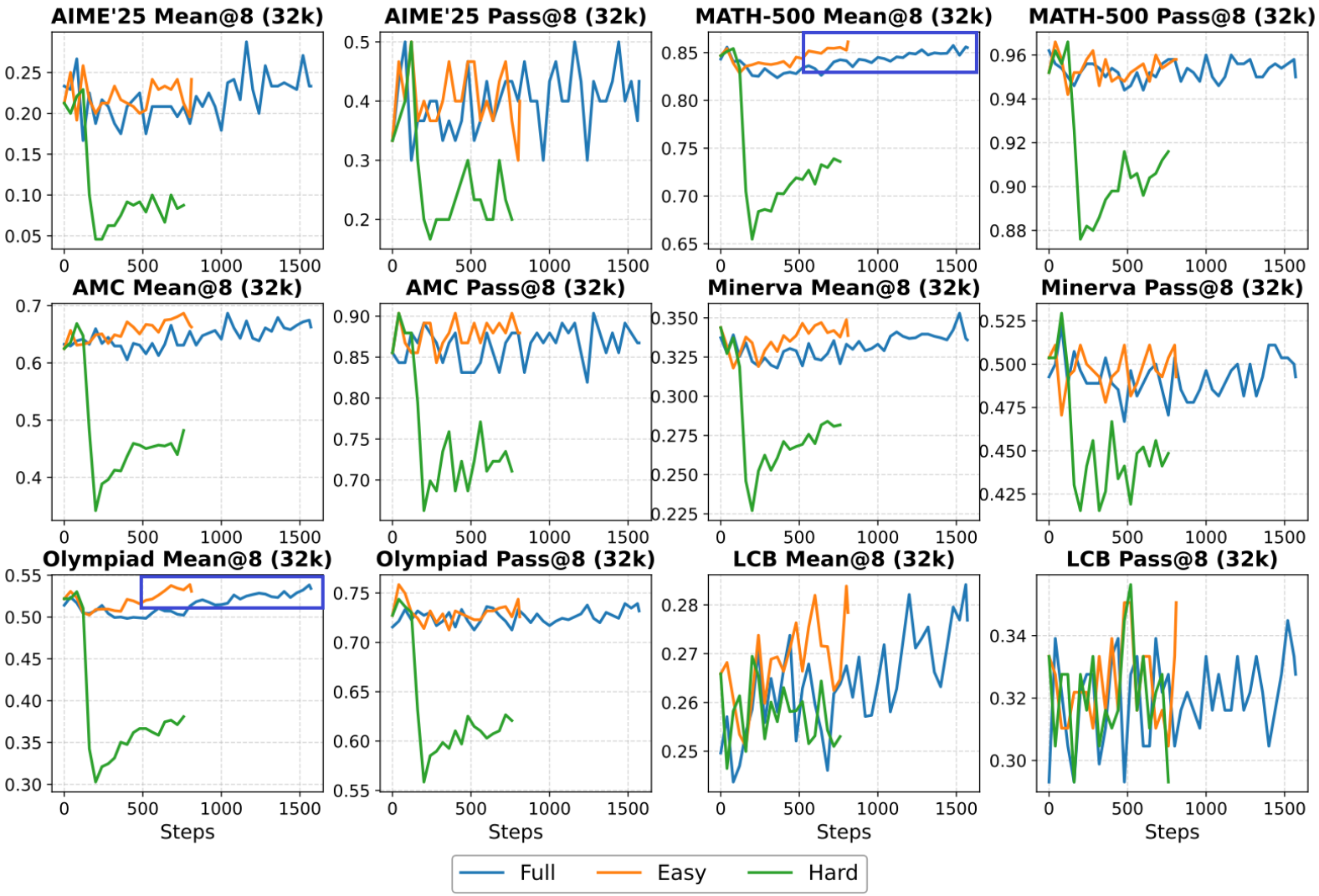


Figure 3: Performance training on all prompts and easy/hard counterparts (rollout $L_R = 16k$, target $L_T = 4k$).

把数据按照通过率分为两组：

简单 & 难

Split the data into two parts:

easy (pass rate >4/8) and hard (pass rate <=4/8)

- 难题会导致崩溃
- 与全量相比，简单题上限类似甚至更高
- Hard part lead to collapse
- Simple is comparable even better than full set

Data

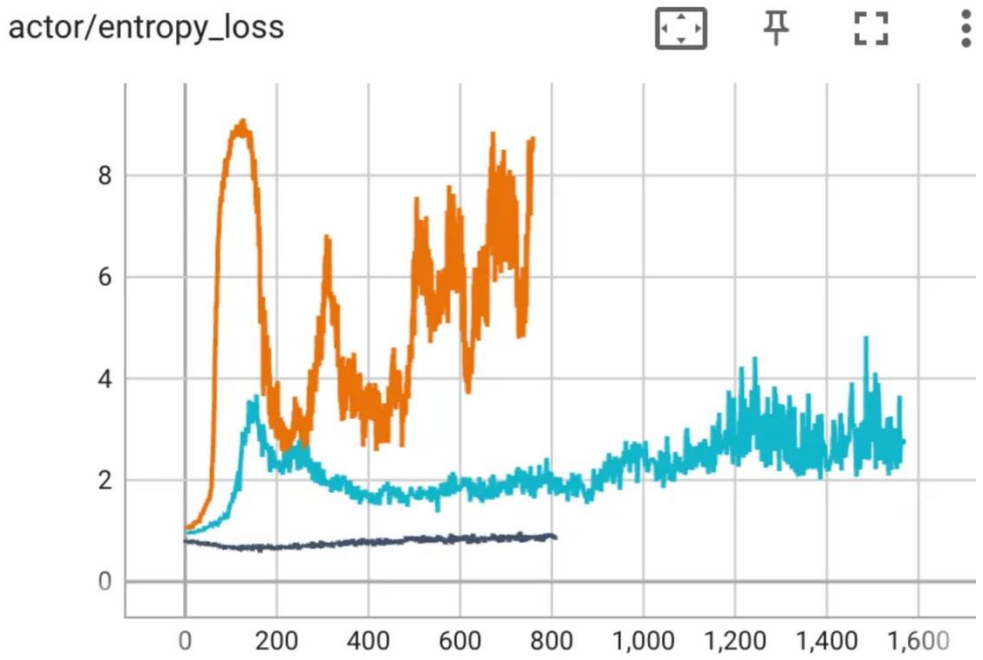
在简单题上，准确性有一定保障，专心优化长度

$$reward_i := f(c_i) \rightarrow reward := g(c_i, l_i)$$

On easy parts, accuracy (c_i) is somehow guaranteed. -> Partial optimization

在困难题上，短且正确的优势值过大, 来自于GRPO 的特点

Normalize ([1,0,0,0]) vs. Normalize ([1,1,1,0]), the advantage of 1 in the former group is larger



简单题的 Entropy 更平滑, 有效奖励密度更大

Training dynamic on easy part is much more smooth,

as the **positive reward density** is larger

Data

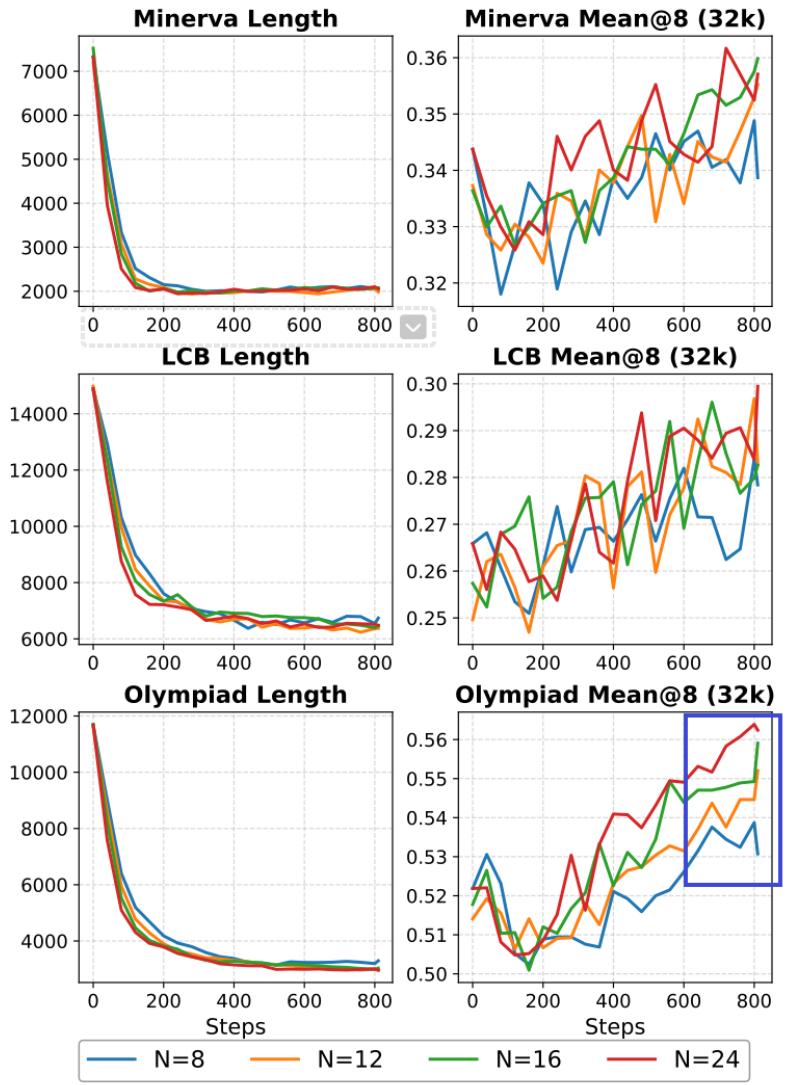


Figure 4: Performance with various rollouts N using DeepScaleR-Easy.

更大的 rollout N 上限更高 -> 有效奖励密度更高

Larger rollout N better -> more positive reward

Insights towards Training Data: *The key is to ensure sufficient and effective rewards. Training on easier prompts allows LLMs to focus on length reduction without compromising performance. Larger rollout N would be better if computational resource allows.*

Reward on Negative Rollouts

Strategy	Correct		Incorrect	
	Short	Long	Short	Long
Vanilla	1	0	0	0
-I	1	0	-	-
-L&C	1	-	0	0
-L&C-S&I	1	-	-	0
-L&C-L&I	1	-	0	-

Table 1: Reward for different strategies on negative rollouts. − denotes masking out.

强化学习的本质是负样本的艺术

RL is the art of assigning negative signal

在原始的 Truncate 中，短&正确奖励为 1，其余为 0

In vanilla truncate strategy, only S&C (short and correct) is assigned as reward 1, otherwise 0

可以对部分负样本 Mask 处理

Mask some negative rollouts rather than zero-rewarding

Reward on Negative Rollouts

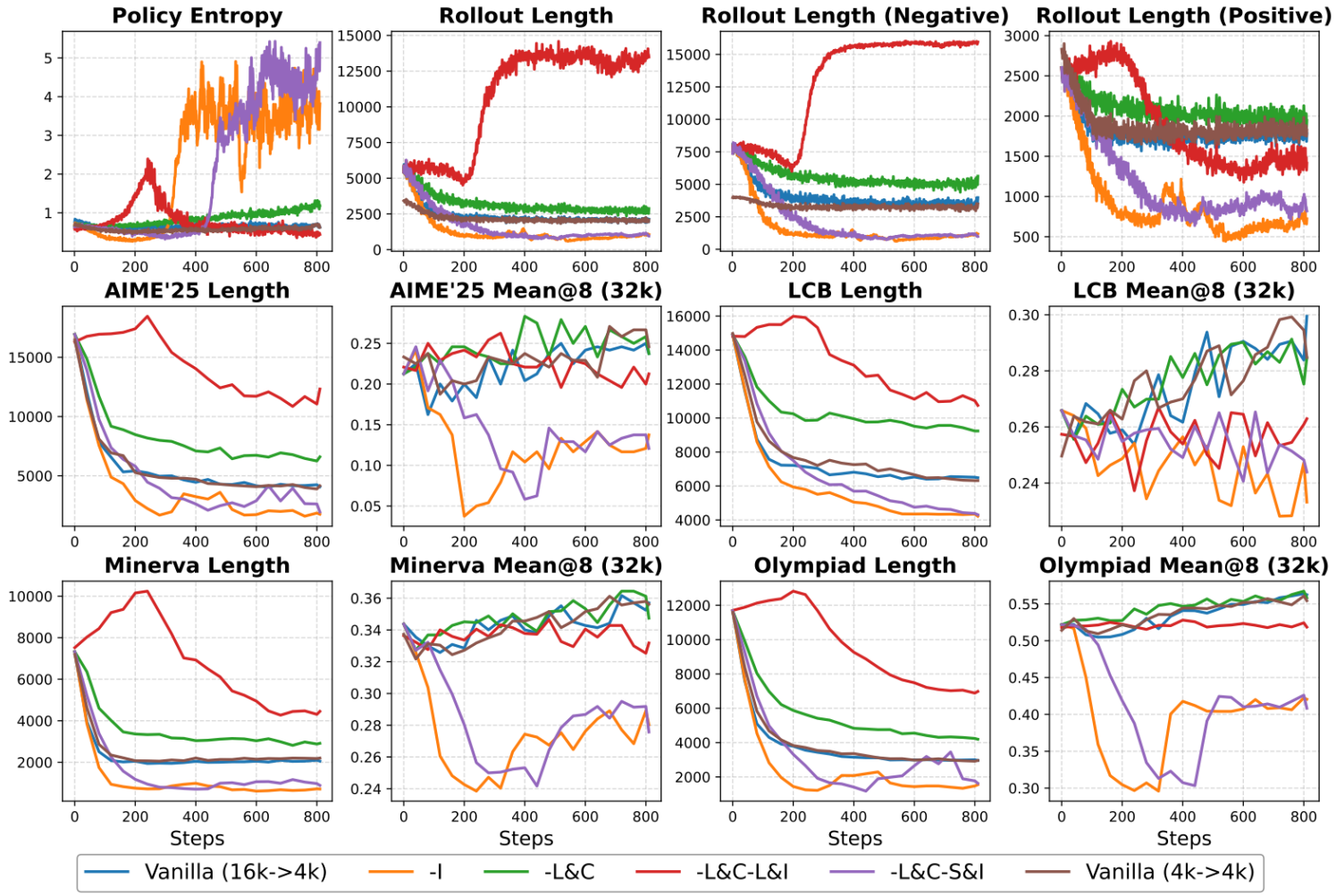


Figure 5: Performance for various reward strategies on negative rollouts (rollout $L_R = 16k$, target $L_T = 4k$, $N = 24$). We also visualize $L_R = 4k$, $L_T = 4k$ for comparison.

-I, -L&C-S&I 短即是对陷阱 The trap that short is correct

所有短的都是正样本，所有长的都是负样本，模型掉入陷阱，输出超短，性能暴跌

The LLMs suffer from the trap that short equals correct while long equals incorrect, leading to performance drop with extremely short outputs

Reward on Negative Rollouts

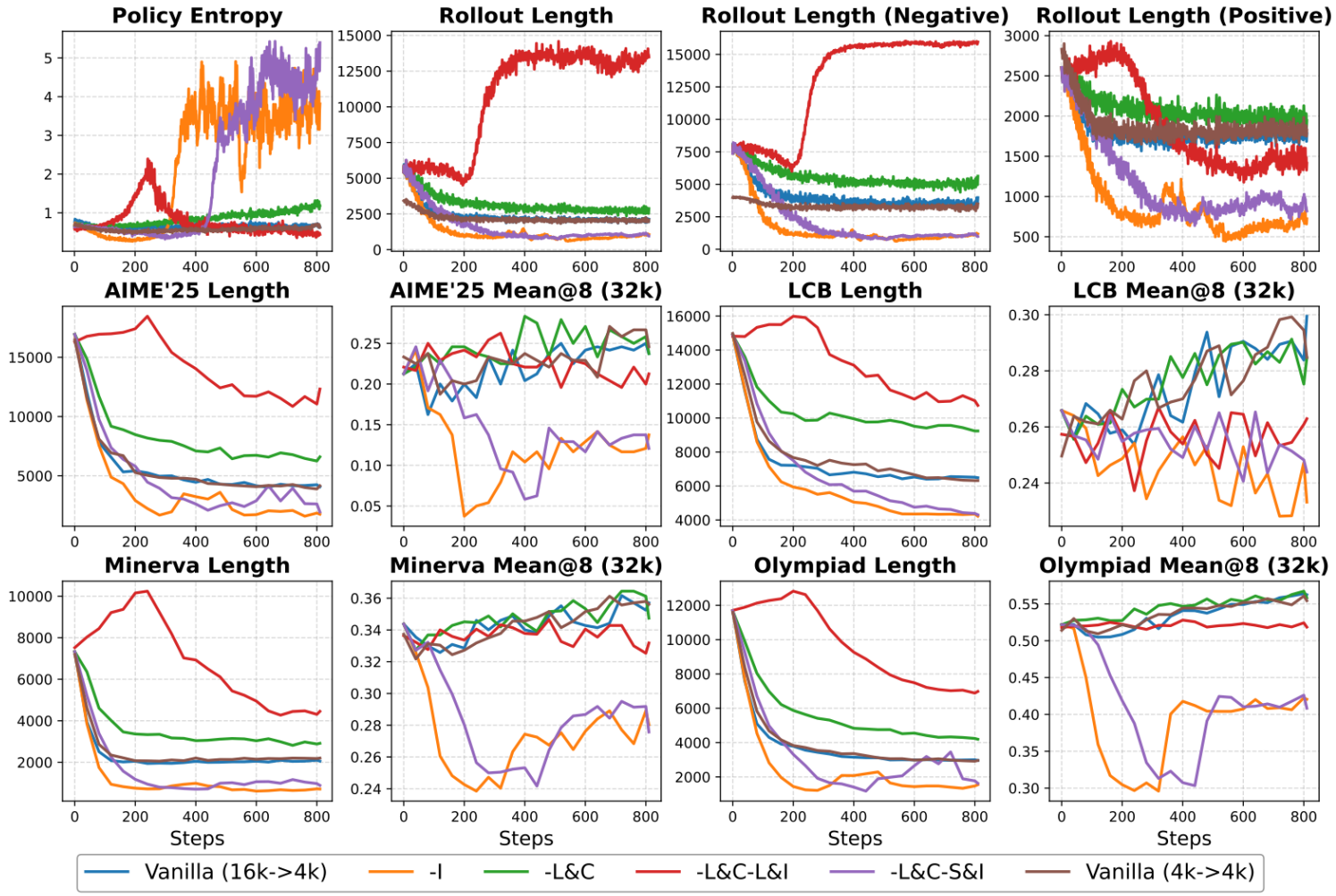


Figure 5: Performance for various reward strategies on negative rollouts (rollout $L_R = 16k$, target $L_T = 4k$, $N = 24$). We also visualize $L_R = 4k$, $L_T = 4k$ for comparison.

-L&C-L&I 只有短输出, Short output only

模型开始 hack， 输出不受惩罚的长文本

The LLMs hack this by generating overlong outputs which are masked out and thus no penalty

Reward on Negative Rollouts

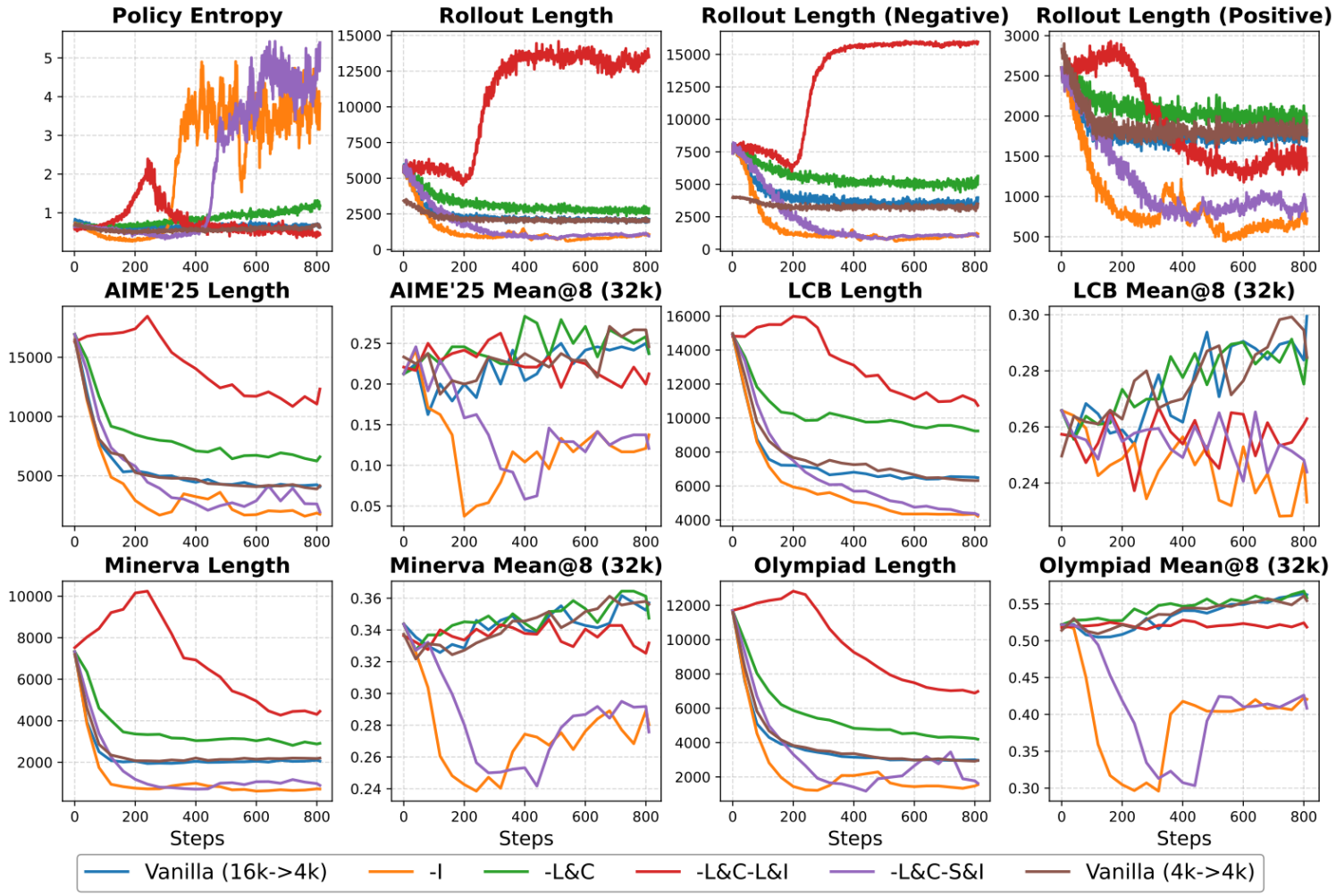


Figure 5: Performance for various reward strategies on negative rollouts (rollout $L_R = 16k$, target $L_T = 4k$, $N = 24$). We also visualize $L_R = 4k$, $L_T = 4k$ for comparison.

-L&C Mask掉超长&正确的样本，而不是赋0 Mask the overlong correct rather than zero-rewarding

模型输出变长，性能微涨

LLMs perform better with longer output

Reward on Negative Rollouts

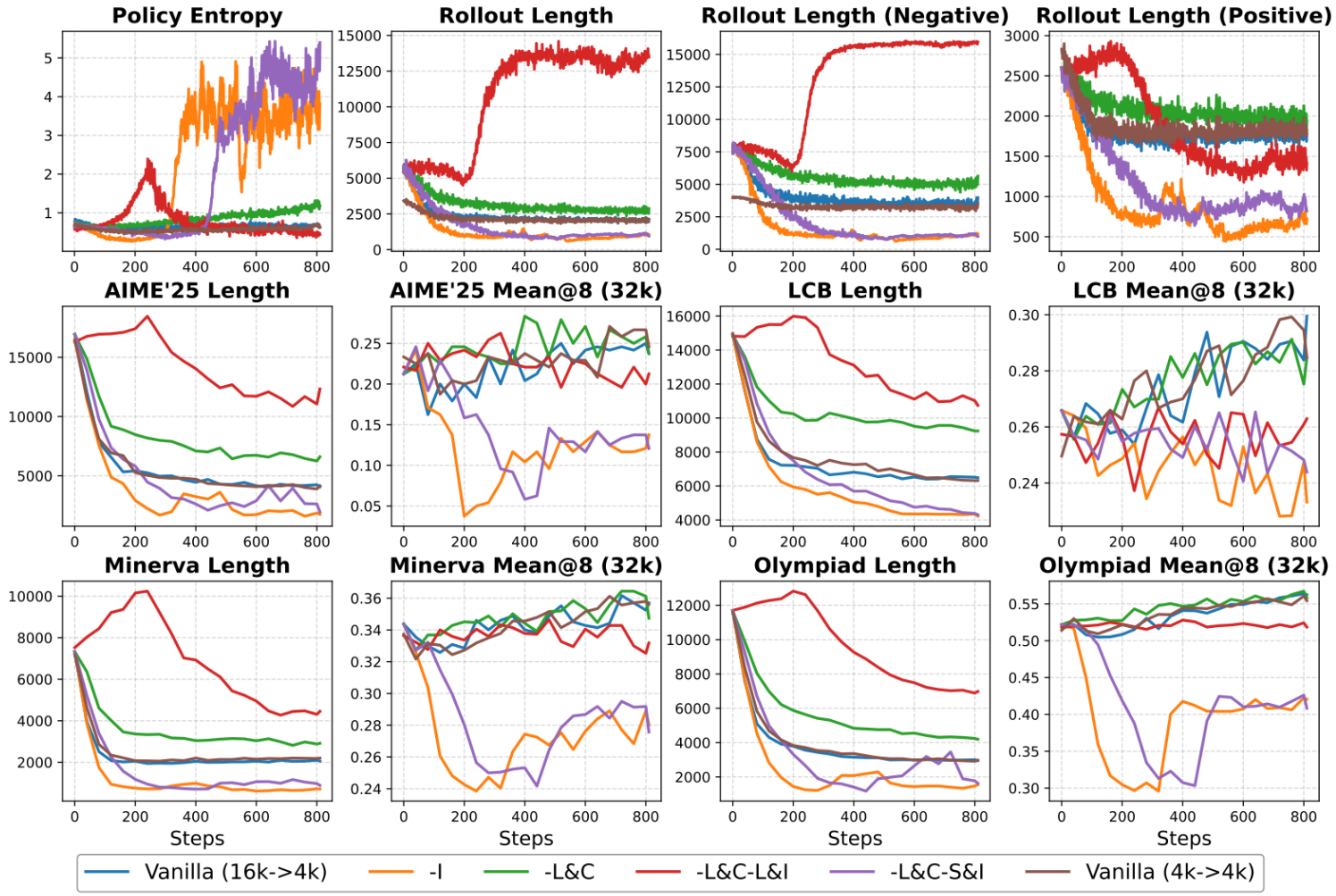


Figure 5: Performance for various reward strategies on negative rollouts (rollout $L_R = 16k$, target $L_T = 4k$, $N = 24$). We also visualize $L_R = 4k$, $L_T = 4k$ for comparison.

Roll 4k -> target 4k vs.

Roll 16k -> target 4k

正样本都是<4k 的正确答案

负样本相当于截断到 4k

Positive samples: both are correct answers within 4k

Negative samples: the former strategy equals to cut off the outputs to 4k

Roll 4k -> target 4k 更优 Better!

Reward on Negative Rollouts

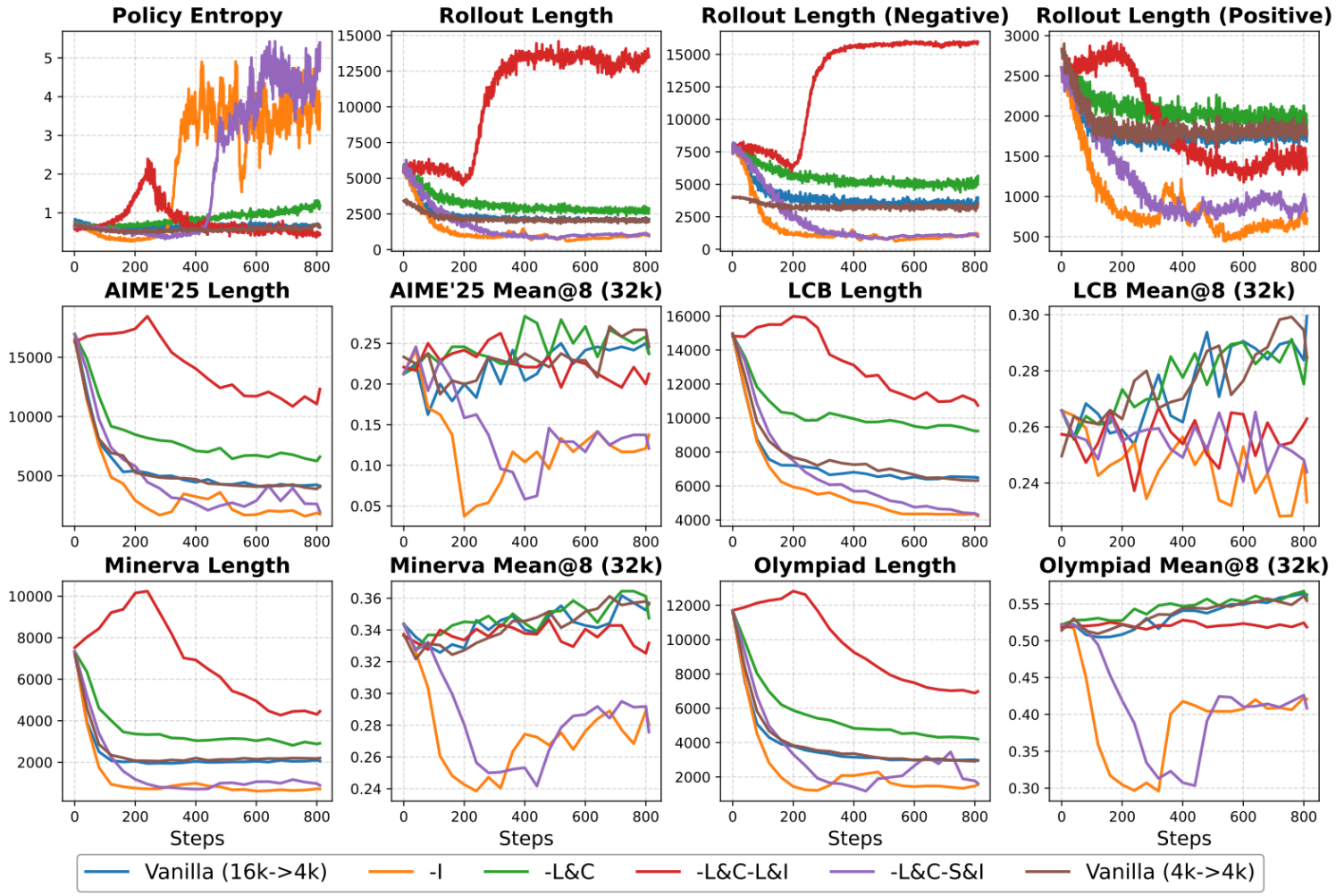


Figure 5: Performance for various reward strategies on negative rollouts (rollout $L_R = 16k$, target $L_T = 4k$, $N = 24$). We also visualize $L_R = 4k$, $L_T = 4k$ for comparison.

Roll 4k -> target 4k 模型是如何变短的？

How can the CoT be short since rollout length equals target length?

通常来说，正样本短于负样本，隐式推动模型变短

Typically, the positive samples are shorter than negative samples, which implicitly push the LLMs be short

Off-Policy Staleness

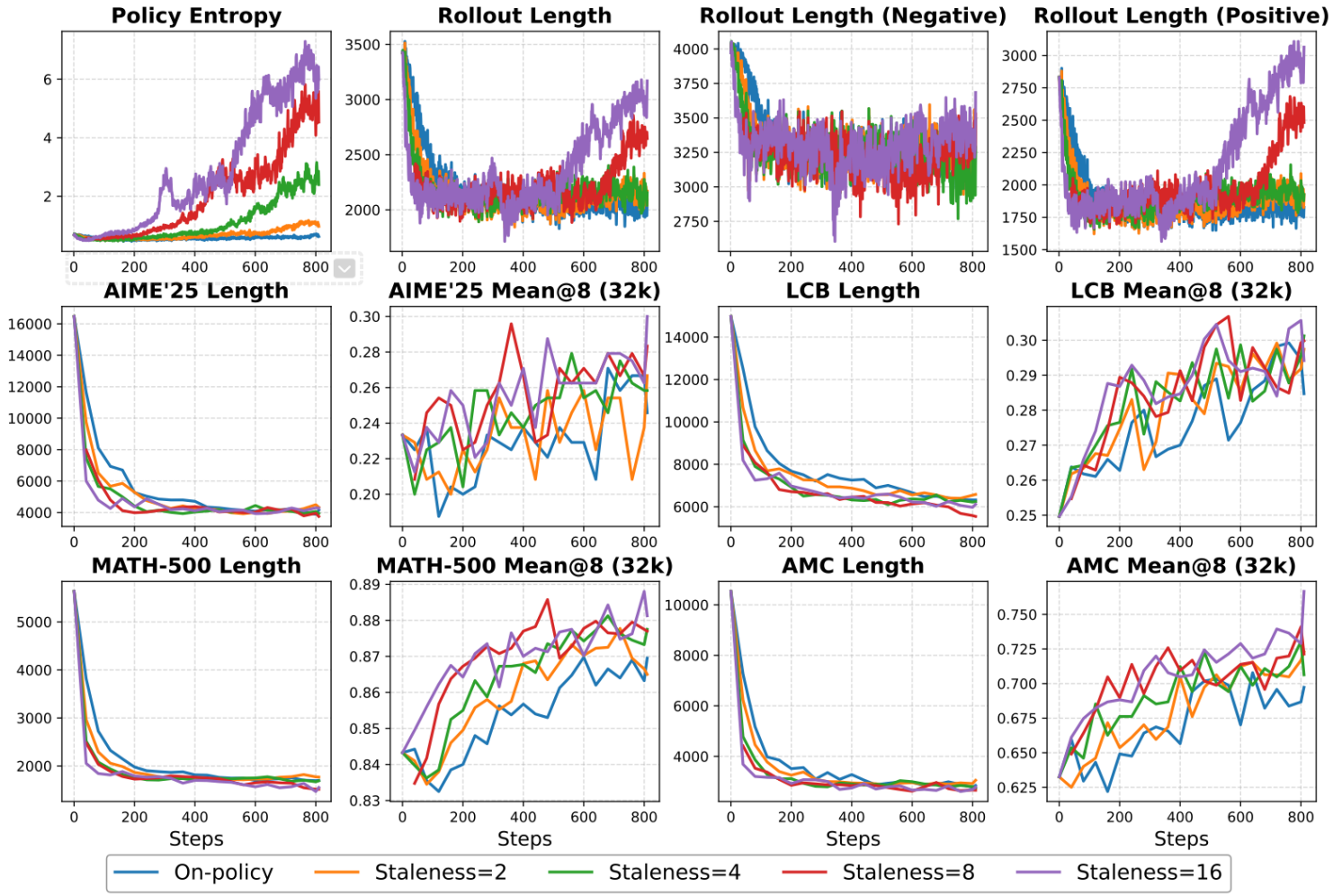


Figure 6: Performance for off-policy strategy with various staleness (i.e., 2,4,8,16).

通过 rollout/update比例引入 off-policy

Introducing off-policy staleness via batch size ratio of rollout and update

更大的staleness 收敛更快，但是引入了不稳定因素， 如熵快速增大

Larger staleness speeds up the process, while introducing potential unstable

-> 对更大的模型慎用, carefully for larger LLMs

Agenda

- 背景介绍 **Introduction**
  - 大语言模型思维链压缩 CoT Compression for LLMs
  - 奖励重塑方法 Reward Shaping Methods
- 数据, 奖励与优化 **Data & Reward & Optimization**
  - 强化学习的数据 Data for RL
  - 负样本奖励分配 Reward Assignment for Negative Rollouts
  - 异策略优化 Off-policy Optimization via Staleness
- **Qwen3 实战 Qwen3 CoT Compression**
  - Qwen3系列压缩 Compression for Qwen3 0.6~30B
  - 调参建议 Suggestion for Hyper-parameters
- 总结与未来方向 **Insights & Future Work**

Qwen3 Results

Method	Mean@8↑	Pass@8↑	Length↓
	Qwen3-0.6B		
Vanilla	13.33	26.67	14.9k
Ours (step 640)	24.58 (+11.25)	36.67 (+10.00)	8.9k (↓40.3%)
	Qwen3-1.7B		
Vanilla	35.00	60.00	17.7k
Ours (step 560)	38.75 (+3.75)	60.00	11.2k (↓36.7%)
	Qwen3-4B-Instruct-2507		
Vanilla	45.42	66.67	9.1k
Ours (step 1440)	46.67 (+1.25)	70.00 (+3.33)	4.8k (↓47.3%)
	Qwen3-4B-Thinking-2507		
Vanilla	75.83	90.00	20.9k
Ours (step 200)	76.25 (+0.42)	86.67	16.0k (↓23.4%)
	Qwen3-8B		
Vanilla	65.83	86.67	17.9k
Ours (step 100)	67.08 (+1.25)	83.33	12.8k (↓28.5%)
	Qwen3-30B-A3B-Instruct-2507		
Vanilla	60.83	83.33	6.9k
Ours (step 600)	60.83	76.67	5.1k (↓26.1%)
	Qwen3-30B-A3B-Thinking-2507		
Vanilla	84.17	96.67	17.3k
Ours (step 120)	86.25 (+2.08)	96.67	14.8k (↓14.5%)

Table: Performance on AIME'25 for Qwen3 models.

参数配方:

1) 使用简单题子集
2) Rollout N 尽可能大
3) 合理设置 Rollout 与目标长度
4) 使用同策略更新策略

Receipt:

1) train on Easy prompts
2) Larger rollout N
3) Setting Rollout & Target Length
4) On-policy optimization

长度下降 15%-50%, 性能不掉 Comparable or better performance with 15-50% short CoTs

Qwen3 Results

Benchmark	Qwen3-30B-A3B-Instruct-2507			Qwen3-30B-A3B-Thinking-2507		
	Mean@4↑	Pass@4↑	Length↓	Mean@4↑	Pass@4↑	Length↓
Domain #1	27.5 / 28.0	40.9 / 40.9	7145 / 4881	34.2 / 34.1	50.0 / 49.8	11536 / 10237
Domain #2	35.1 / 36.0	51.7 / 50.7	7258 / 5182	42.2 / 41.6	68.8 / 68.4	11744 / 10847
Domain #3	20.5 / 21.0	31.9 / 29.8	3985 / 3102	27.0 / 26.2	34.0 / 34.7	8826 / 8278
Domain #4	15.4 / 15.8	24.0 / 25.0	7695 / 3176	16.4 / 14.7	28.9 / 26.2	4726 / 4292
Domain #5	40.6 / 40.6	53.3 / 55.4	6487 / 4500	57.6 / 58.6	78.2 / 78.2	2598 / 2351
Domain #6	6.9 / 6.8	10.8 / 10.8	12322 / 9481	18.3 / 16.7	33.8 / 30.4	21346 / 18115
Domain #7	25.4 / 24.4	43.8 / 40.3	897 / 799	25.5 / 25.2	21.2 / 20.8	4962 / 4312
Domain #8	41.5 / 40.6	52.5 / 51.2	1429 / 1309	29.6 / 29.5	45.6 / 44.8	13912 / 11726
Domain #9	49.4 / 48.2	66.6 / 67.5	1100 / 991	37.2 / 37.5	51.5 / 52.5	3302 / 2943
Domain #10	14.1 / 13.0	27.4 / 24.9	6835 / 5069	47.3 / 45.6	63.3 / 60.0	2492 / 2311
Average	25.2 / 24.9	37.6 / 36.9	5622 / 3735	30.7 / 30.0	44.6 / 43.1	9285 / 8077

Table 3: Performance on private out-of-distribution benchmarks covering various domains. Each cell reports *Vanilla / Ours*. Our proposed model thinks shorter while maintaining comparable performance.

在覆盖 10 个领域的私有 OOD 数据集上测试，思维链显著变短，效果基本不变

On the private OOD benchmarks containing 10 domains, the CoTs are shorter while comparable performance

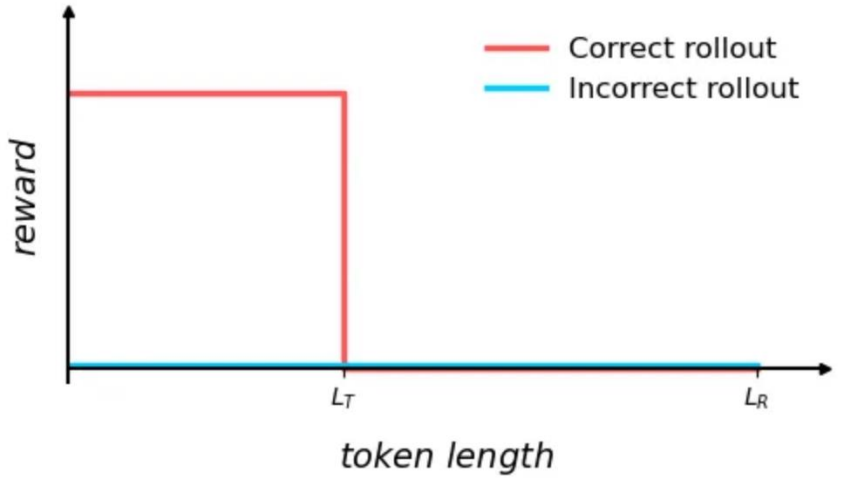
Suggestion for Hyper-parameters

对于 rollout长度，选取模型输出截断率比较低的值，典型值如 8k 12k 16k

对于 target长度，初始设置为 rollout长度，模型长度不下降则轻微下调，下降太快则两个长度都扩大

For rollout length, set as a number with lower initial truncation rate, such as 8k, 12k, 16k

For target length, one typical value is equal to rollout length. If the CoTs do not shorten, try decrease the value of target length. Otherwise, increasing two lengths at the same time.



Agenda

- 背景介绍 **Introduction**
  - 大语言模型思维链压缩 CoT Compression for LLMs
  - 奖励重塑方法 Reward Shaping Methods
- 数据, 奖励与优化 **Data & Reward & Optimization**
  - 强化学习的数据 Data for RL
  - 负样本奖励分配 Reward Assignment for Negative Rollouts
  - 异策略优化 Off-policy Optimization via Staleness
- **Qwen3 实战 Qwen3 CoT Compression**
  - Qwen3系列压缩 Compression for Qwen3 0.6~30B
  - 调参建议 Suggestion for Hyper-parameters
- **总结与未来方向 Insights & Future Work**

Insights

- 模型学习长度偏置相对容易 It is easier for LLMs to fit length constrains than getting right answer considering *reward=f(c, l)*

- 核心在于避免显式引入长度陷阱 The key is to avoid the trap that short equals correct. The LLMs should learn how to be concise rather than just be short.

- 基于内在的先验可隐式推动模型变简洁 We can implicitly push the LLMs be efficient via priors that correct outputs are typically shorter than incorrect ones.

Future Work

- 更多元的训练数据 More diverse training prompts
- 动态设置采样长度与截断长度 Setting the rollout and target length adaptively.
- 引入工具来简化思维过程 Tool call for efficient thinking like human