Test-Time Reinforcement Learning

한성빈

TTRL: Test-Time Reinforcement Learning

Yuxin Zuo[*1,2] **Kaiyan Zhang**[*1] **Li Sheng**[1,2] **Shang Qu**[1,2] **Ganqu Cui**[2]
Xuekai Zhu[1] **Haozhan Li**[1,2] **Yuchen Zhang**[2] **Xinwei Long**[1] **Ermo Hua**[1]
Biqing Qi[2] **Youbang Sun**[1] **Zhiyuan Ma**[1] **Lifan Yuan**[1]
Ning Ding[†1,2] **Bowen Zhou**[†1,2]
[1]Tsinghua University [2]Shanghai AI Lab
zhang-ky22@mails.tsinghua.edu.cn, dingning@mail.tsinghua.edu.cn

NeurIPS 2025

Terminology

Test-Time Inference

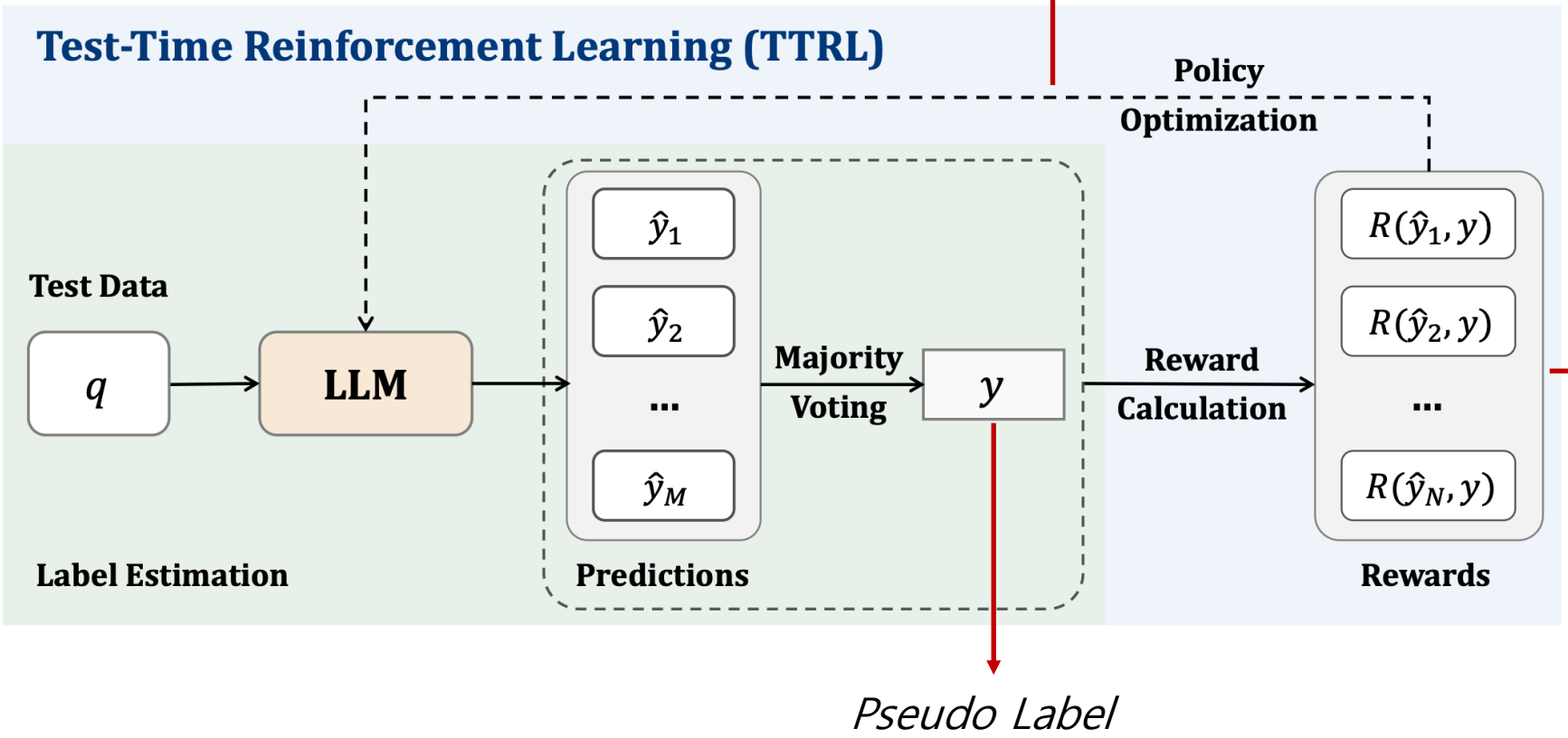
- Inference 시점에 resource 투자해 성능향상
  - Majority-Voting / Best-of-N / MCTS
- Parameter를 업데이트 하지 않음
  - Training 시 보지 않은 OOD data 에 대해선 한계 존재
  - OpenAI o3 (ARC-AGI-1: 75.4% → ARC-AGI-2: 4%)

Test-Time Training

- Test 시점의 Unlabeled Data 통해 학습
  - Test data 의 structure, distribution 특성을 활용하여 model parameter 수정
- RL 을 Test-Time 에 적용시, 가장 큰 문제는?
  - Ground Truth 가 없는 상황에서, 어떻게 Reward 를 정의할 것인가?
- GT 가 없는 setting 에서 RL 을 수행하는 것을 **Test-Time RL** 이라 정의.

TTRL Methodology

$$\theta \leftarrow \theta + \eta \nabla_\theta \mathbb{E}_{y \sim \pi_\theta(\cdot|x)}[r(y, y^*)]$$



$$R(\hat{y}_i, y) = \begin{cases} 1, & \text{if } \hat{y}_i = y, \\ 0, & \text{otherwise.} \end{cases}$$

Experimental Setup

Model

- **Qwen Family:** Qwen2.5-Math-1.5B (Yang et al., 2024a), Qwen2.5-Math-7B (Yang et al., 2024a), Qwen2.5-7B (Yang et al., 2024b), Qwen2.5-32B (Yang et al., 2024b), Qwen3-8B (thinking mode & non-thinking mode) (Yang et al., 2024b);
- **LLaMA Family:** LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024), LLaMA-3.2-3B-Instruct (Grattafiori et al., 2024), LLaMA-3.2-3B-Oat-Zero (Liu et al., 2025b);
- **Mistral Family:** Mistral-Nemo-Instruct-2407 (MistralAI-NeMo, 2024), Ministral-8B-Instruct-2410 (Ministral-8B-Instruct, 2024);
- **DeepSeek Family:** DeepSeek-Math-7B-Instruct (Shao et al., 2024), DeepSeek-R1-LLaMA-8B (Guo et al., 2025);
- **Others:** Skywork-OR1-Math-7B (He et al., 2025);

Training & Evaluation Benchmark

- GPQA
- AIME 2024
- AMC
- MATH-500

Experimental Setup

Implementation Details

- **GRPO**
- **Cosine Scheduler**
  - peak lr: 5e-7
- **Optimizer**
  - AdamW
- **Majority Voting Detail**
  - **Rollouts**
    - 64 rollouts/prompt
    - Use 32 rollouts during training
  - **Temperature**
    - Qwen2.5-Math/LRMs : 1.0
    - Another models : 0.6
  - **Epochs (based on dataset size)**
    - Math-500: 10
    - AMC: 30
    - AIME 2024: 80

Evaluation Setup

- **Pass@1**
  - 16 times/prompt
- **Rollout Setup**
  - Temperature: 0.6
  - Top-p: 0.95
- **Generation length**
  - 3k for non-LRMs
  - 32k for LRMs

Results

Table 1: Main results of **TTRL** on each task. * indicates that Qwen3-8B is evaluated in non-thinking mode within a 3k context. Figure 3 provides results within a 32k context.

Name	AIME 2024	AMC	MATH-500	GPQA	Avg
Math Base Models					
Qwen2.5-Math-1.5B	7.7	28.6	32.7	24.9	23.5
w/ **TTRL**	15.8	48.9	73.0	26.1	41.0
Δ	+8.1	+20.3	+40.3	+1.2	+17.5
	↑ 105.2%	↑ 71.0%	↑ 123.2%	↑ 4.8%	↑ 74.4%
Qwen2.5-Math-7B	12.9	35.6	46.7	29.1	31.1
w/ **TTRL**	40.2	68.1	83.4	27.7	54.9
Δ	+27.3	+32.5	+36.7	−1.4	+23.8
	↑ 211.6%	↑ 91.3%	↑ 78.6%	↓ 4.8%	↑ 76.5%
Vanilla Base Models					
Qwen2.5-7B	7.9	34.8	60.5	31.8	33.8
w/ **TTRL**	23.3	56.6	80.5	33.6	48.5
Δ	+15.4	+21.8	+20.0	+1.8	+14.7
	↑ 194.9%	↑ 62.6%	↑ 33.1%	↑ 5.7%	↑ 43.7%
Qwen2.5-32B	7.9	32.6	55.8	33.2	32.4
w/ **TTRL**	24.0	59.3	83.2	37.7	51.1
Δ	+16.1	+26.7	+27.4	+4.5	+18.7
	↑ 203.8%	↑ 81.9%	↑ 49.1%	↑ 13.6%	↑ 57.7%
Instruct Models					
LLaMA3.1-8B	4.6	23.3	48.6	30.8	26.8
w/ **TTRL**	10.0	32.3	63.7	34.1	35.0
Δ	+5.4	+9.0	+15.1	+3.3	+8.2
	↑ 117.4%	↑ 38.6%	↑ 31.1%	↑ 10.7%	↑ 30.6%
Qwen3-8B*	26.9	57.8	82.3	48.1	53.8
w/ **TTRL**	46.7	69.1	89.3	53.0	64.5
Δ	+19.8	+11.3	+7.0	+4.9	+10.8
	↑ 73.6%	↑ 19.6%	↑ 8.5%	↑ 10.2%	↑ 20.0%

- Post-training 적용된 모델에서도 성능 향상
- 고난도 benchmark (AIME 2024) 에서 더 큰 성능 향상기록
- GPQA 는 general knowledge 벤치마크로, Majority Voting 방식의 보상 추정이 덜 효과적이기에 성능 향상폭이 적음

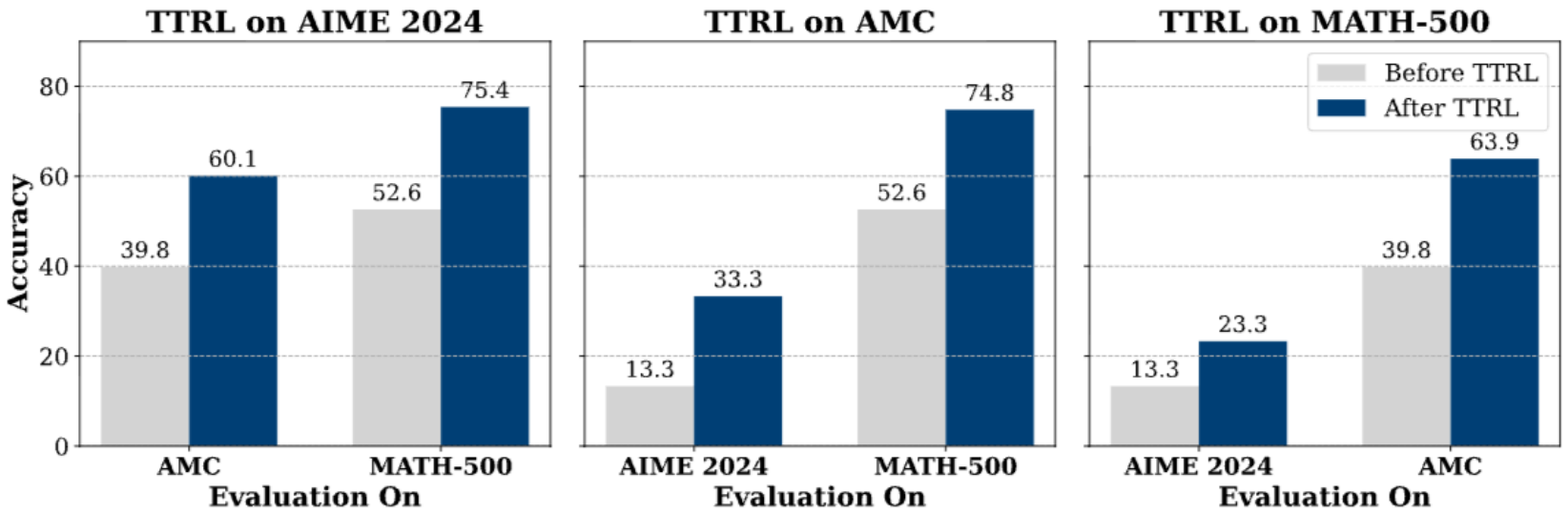
Results

Table 2: Performance of **TTRL** on various models.

Name	AIME	AMC	MATH-500
LLaMA Family			
LLaMA-3.2-3B-Oat-Zero	0.8	15.1	41.9
w/ **TTRL**	3.3	25.3	55.7
Δ	+2.5	+10.2	+13.8
LLaMA-3.2-3B-Instruct	6.0	19.4	43.9
w/ **TTRL**	13.3	31.3	61.6
Δ	+7.3	+11.9	+17.7
Mistral Family			
Mistral-Nemo-Instruct	0.8	15.4	40.8
w/ **TTRL**	0	24.8	51.0
Δ	−0.8	+9.4	+10.2
Ministral-8B-Instruct	1.3	19.7	52.4
w/ **TTRL**	3.3	28.9	57.8
Δ	+2.0	+9.2	+5.4
DeepSeek Family			
DeepSeek-Math-7B-Instruct	1.9	16.3	42.3
w/ **TTRL**	2.5	22.9	52.4
Δ	+0.6	+6.6	+10.1
DeepSeek-R1-LLaMA-8B	51.7	81.6	89.6
w/ **TTRL**	69.2	88.9	90.9
Δ	+17.5	+7.3	+1.3

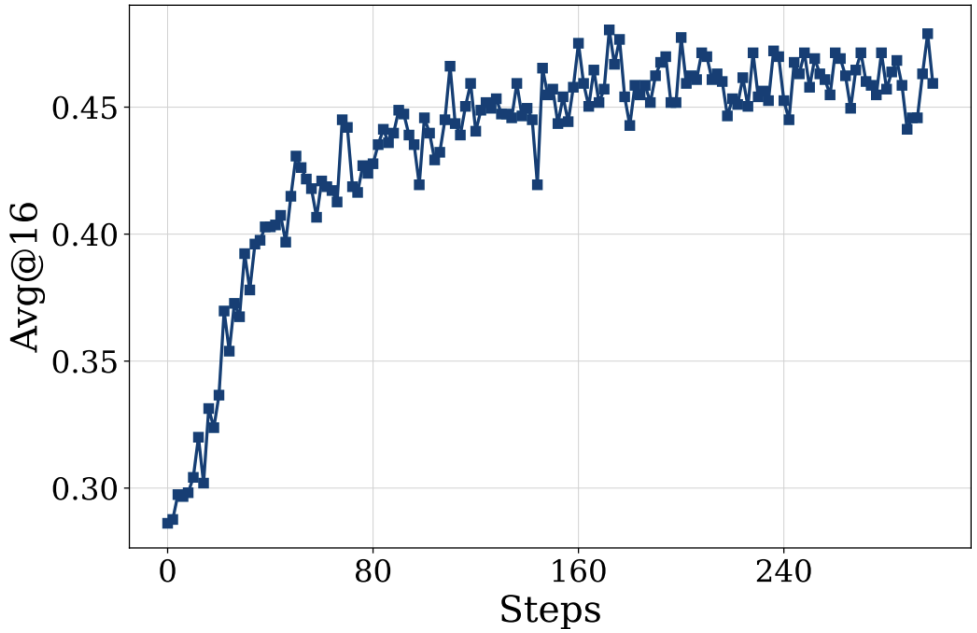
- Qwen 외의 다른 아키텍처 model 에서도 전반적인 성능향상 기록
- Qwen model 들에 비하면 성능향상 폭이 낮음.
- 낮은 초기성능으로 인한 Majority Voting 의 신뢰성, Cold start 문제 등이 원인

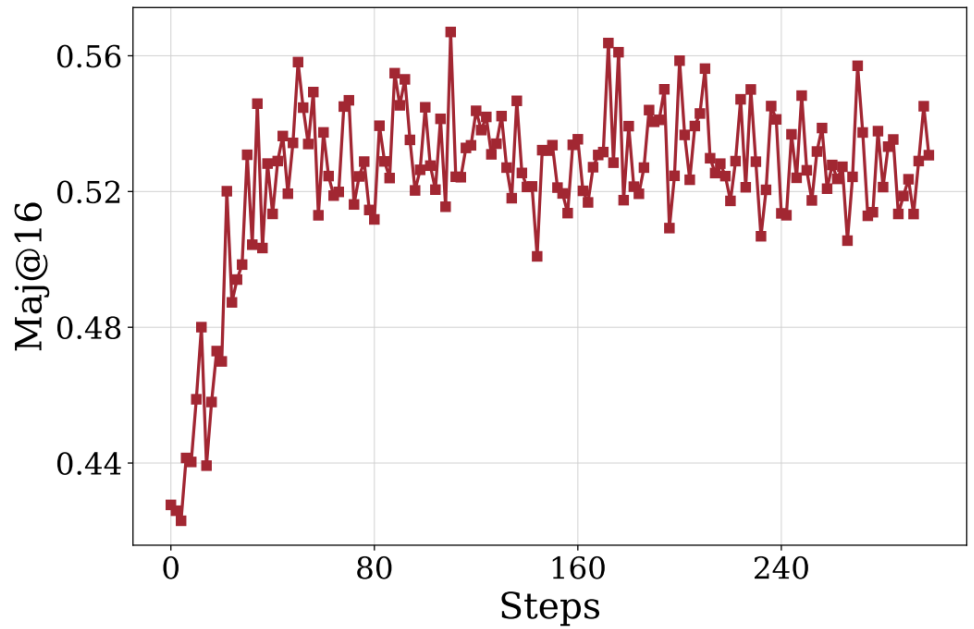
Results



- OOD data 에 대해서도 성능향상
- 특정 Benchmark 에 대한 overfitting 이 아닌, 실제 추론 능력 향상을 유도

Results





- Avg@16과 Maj@16 모두 상승
- 학습 후의 성능이 초기 모델의 Maj@16 을 넘어섬
  - Self-training의 이론적 한계를 넘어섬
- **성공 요인**
  - **선순환 구조**
    - 학습을 통해 모델의 능력이 향상
    - → 모델이 생성하는 답변 정확도 상승
    - → Majority Voting 정확도 상승
    - → 고품질의 reward signal 로 학습
  - **RL의 유연성**
    - Pseudo-label 을 절대적 정답이 reward signal 로 변환하여 학습함으로써, maj@n의 한계를 넘어섬

Discussion

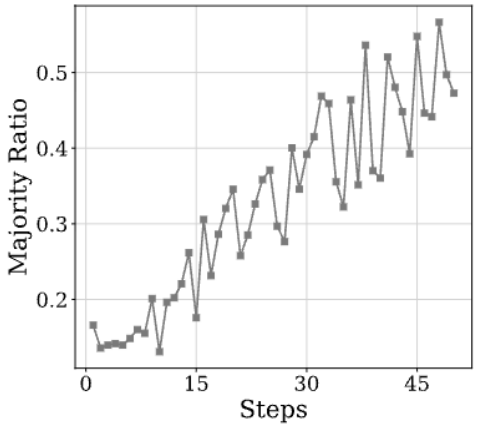
Discussion 1: Why does TTRL Work?

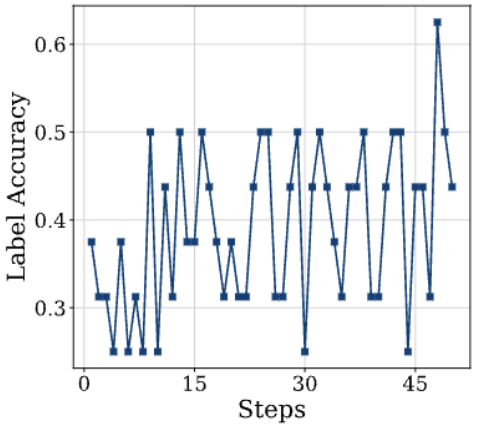
1. **RL's robustness**
   - RL 은 reward 를 절대적인 정답이 아닌, direction signal 역할을 함
   - 따라서, noise 가 일부 섞여 있더라도, model은 상대적으로 옳바른 방향으로 학습할 수 있음

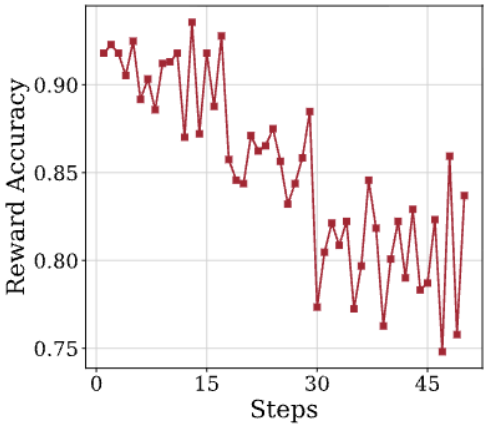
2. **Self-Evolution Loop**
   - TTRL 은 Online-RL method
   - Online-RL 특성을 통해, 선순환 구조 생성
   - [성능 향상 → 더 정확한 라벨 → 더 좋은 Reward Signal]

Discussion

Discussion 1: Why does TTRL Work?





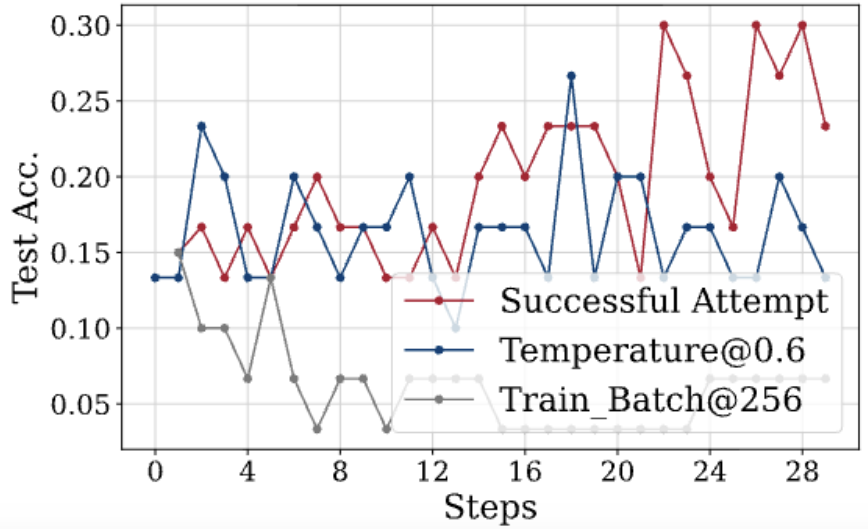


3. Lucky Hit

- Pseudo-Label 이 GT 와 다르더라도, 두 오답이 다르면 0 의 Reward 를 받게 됨
- 즉, 잘못 예측을 하더라도 Negative Signal 을 받는 sample 들이 생기게 되어 학습이 진행 됨
- 모델의 성능이 낮은 학습 초기에 오히려 Reward Accuracy 는 높게 나타남

Discussion

Discussion 2: When might TTRL fail?



1. RL Hyperparameters

- RL 은 원래도 hyperparameter 에 민감하지만, GT 가 없이 학습하는 TTRL 은 그 만감도가 더 높음

- **Temperature**
  - Temperature를 1.0 으로, 상대적으로 높게 유지하는 것이 유리.
  - 더 많은 exploration을 유도하는 게 적합.
- **Episodes**
  - 데이터셋의 크기가 작거나, 난이도가 높을 수록 더 많은 epoch 필요.

Discussion

Discussion 2: When might TTRL fail?

Table 3: Performance of **TTRL** across the five difficulty levels of MATH-500.

Metric	Name	MATH-500-L1	MATH-500-L2	MATH-500-L3	MATH-500-L4	MATH-500-L5
Accuracy	Backbone	25.9	33.0	36.3	32.5	22.3
	w/ **TTRL**	71.2	76.2	76.3	58.7	39.2
	Δ	+45.4 ↑ 175.3%	+43.2 ↑ 130.8%	+40.0 ↑ 110.2%	+26.2 ↑ 80.4%	+16.8 ↑ 75.3%
Response Len.	Backbone	2,339.2	2,125.1	2,120.6	1,775.1	1,751.3
	w/ **TTRL**	624.3	614.4	672.3	783.5	985.3
	Δ	−1,715.0 ↓ 73.3%	−1,510.6 ↓ 71.1%	−1,448.3 ↓ 68.3%	−991.6 ↓ 55.9%	−766.0 ↓ 43.7%

2. Lack of prior knowledge

- Background knowledge 가 전혀 없다면, 정답인 경로를 찾기가 불가능 함
- 어느정도 문제를 풀 능력이 있는 difficulty 의 문제에서 가장 큰 상승률 보임
- Difficulty 조절 없이, 학습을 시도하면 실패할 확률이 높음.

Distribution-Aware Reward Estimation for Test-Time Reinforcement Learning

Bodong Du [1] Xuanqi Huang [1] Xiaomeng Li [1]

ARXIV

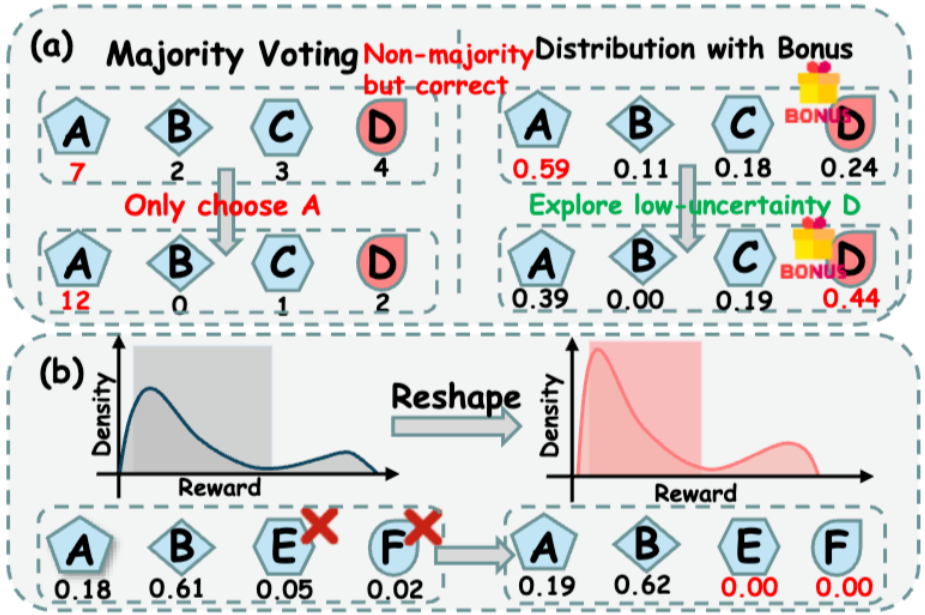
Motivation

TTRL 의 한계

- **Information Collapse**
  - Rollout 들이 형성하는 Empirical Distribution를 단 하나의 최빈값으로 축소
  - 빈도수는 낮지만 올바른 추론 과정을 담고 있을 수 있는 Rollout의 학습 신호를 배제
  - 유의미한 minority signal 이 있어도, Majority Voting 은 해당 signal 을 아예 제거.

- **Confirmation Collapse**
  - Rollouts 상관관계가 있을 때 MV가 편향 될 수 있음
  - 실제로 한 모델이 같은 질문에 여러 번 답하면, 샘플들이 비슷한 Reasoning path에 묶이게 됨
  - 이런 상황에선 Majority-Voting reward 가 잘못된 reasoning 으로 인해 나온 경로를 강화할 수 있음
  - 이는 학습 초기의 우연한 오답 모드가 강화되는 **confirmation collapse** 로 이어질 수 있음

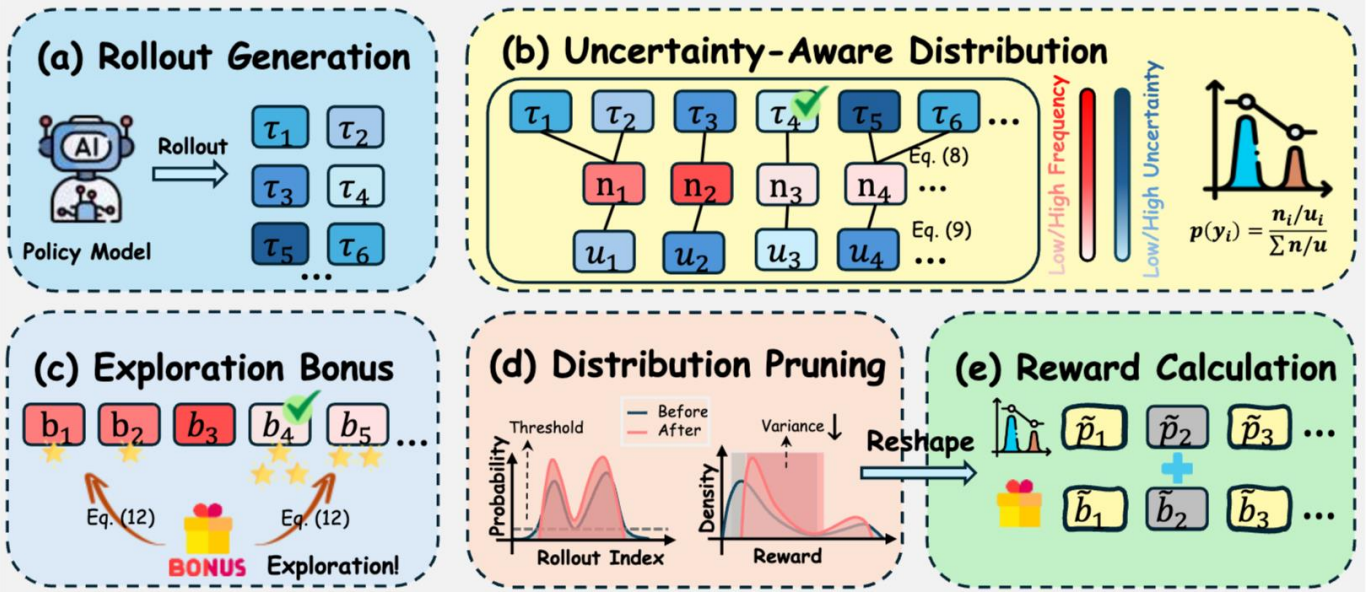
Contribution



DARE (Distribution-Aware Reward Estimation)

- Rollout prediction 들이 만들어낸 distribution 자체를 reward 로 사용
- 직관적으로 "확률이 높은 답변을 더 보상" 하는 형태
- Distribution 의 신뢰도를 높이기 위해 uncertainty, Exploration bonus, pruning 을 추가

Methodology



1. Uncertainty-Aware Distribution

- 현재 policy 에서 M개의 rollout 생성

$$\{\tau_1, \ldots, \tau_M\} \sim \pi_\theta(\cdot \mid q),$$

- Rollout 들에서 특정 response 의 frequency count 계산

$$n(\hat{y}) = \sum_{k=1}^{M} \mathbf{1}[\hat{y}_k = \hat{y}],$$

- Response의 모든 rollout의 avg entropy 계산 (Uncertainty)

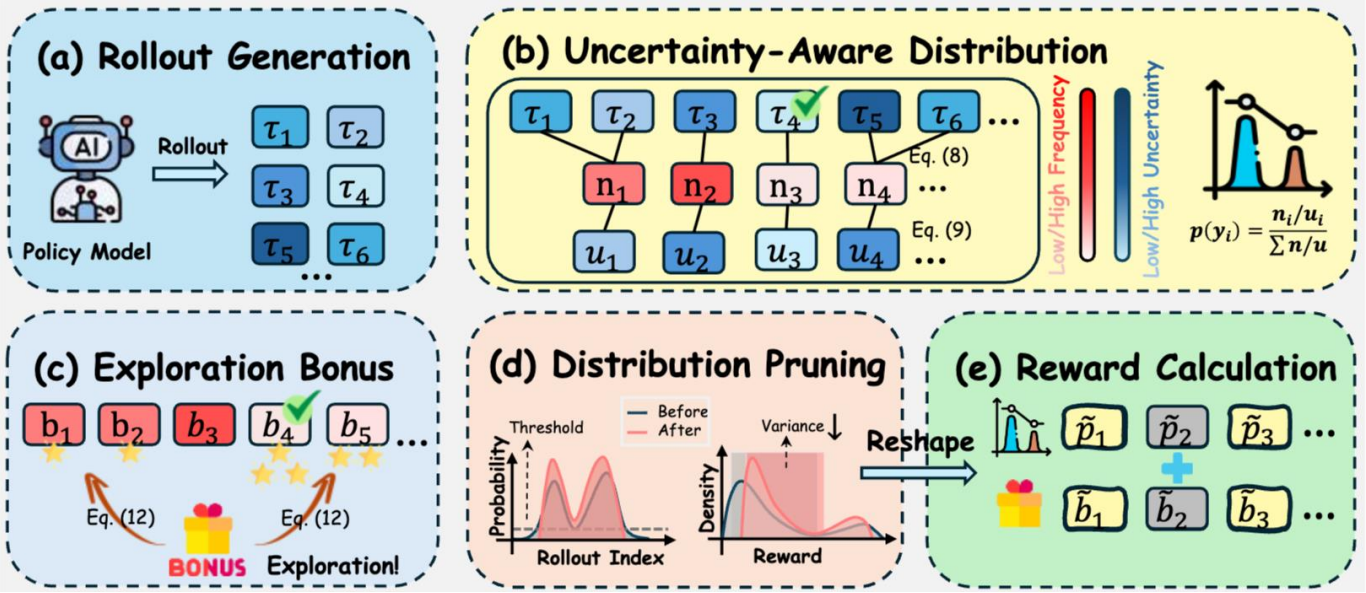
$$u(\hat{y}) = \frac{1}{n(\hat{y})} \sum_{k:\hat{y}_k=\hat{y}} \frac{1}{|\tau_k|} \sum_{i \in \tau_k} \sum_{j} -P_i(j) \log P_i(j),$$

- Frequency 와 uncertainty를 결합하여 답변들의 최종 distribution 구성

$$\hat{p}(\hat{y}) = \frac{n(\hat{y})/(u(\hat{y}) + \epsilon)}{\sum_{\hat{y}'} n(\hat{y}')/(u(\hat{y}') + \epsilon)},$$

- Frequency가 높을수록, uncertainty 가 낮을수록 더 높은 probability 부여

Methodology



2. Exploration Bonus

- 앞선 단계의 각 response 의 probability를 default reward 로 설정

$$r_{\text{dis}}(y_i) = \hat{p}(y_i),$$

- Frequency가 낮지만, uncertainty가 낮은 답안에 추가 점수

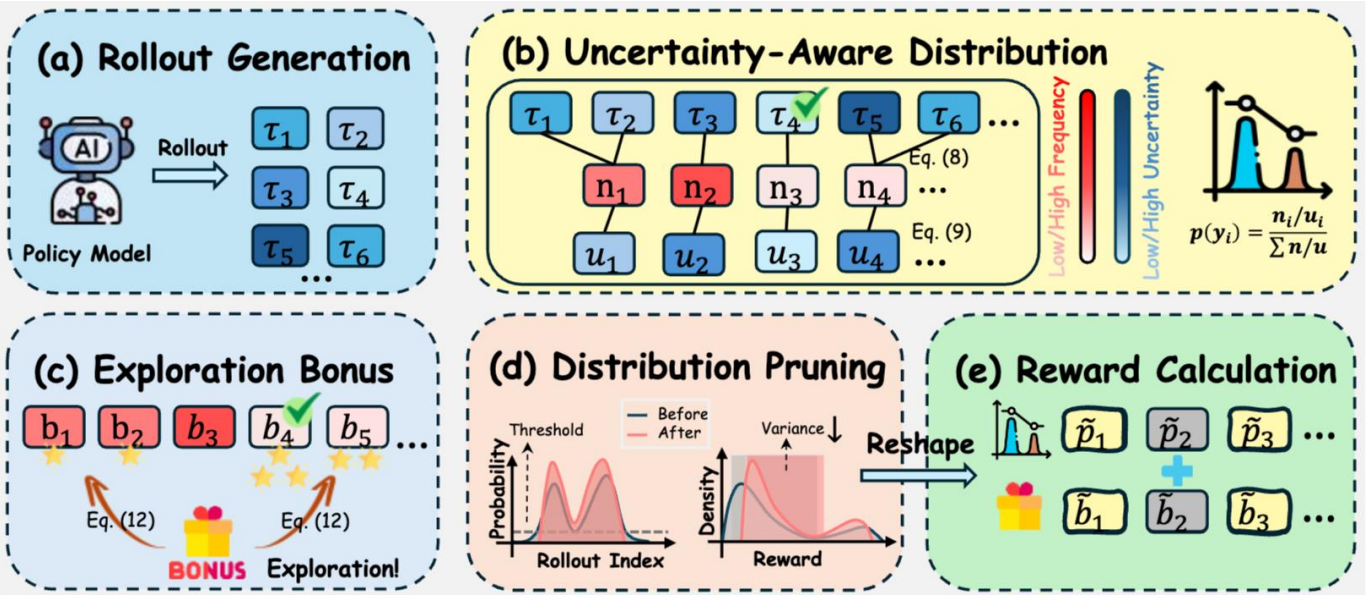
$$b(y_i) = \left(1 - \frac{n(y_i)}{M}\right) \cdot \left(1 - u(y_i)\right).$$

- 추가 점수를 적용한 최종 reward 계산

$$r(y_i) = r_{\text{dis}}(y_i) + \alpha\, b(y_i), \quad \alpha \in [0, 1]$$

- Frequency가 낮더라도, Uncertainty 가 낮다면 해당 response를 활용할 수 있도록 Exploration 을 장려

Methodology



3. Distribution pruning

- Stability 를 위해, 임계값 이하의 항목을 pruning

$$\tilde{p}(y_i) = \frac{\hat{p}(y_i)\,\mathbf{1}[\hat{p}(y_i) \geq \tau]}{\sum_{k=1}^{M} \hat{p}(y_k)\,\mathbf{1}[\hat{p}(y_k) \geq \tau]}.$$

- Pruning 후, response 들의 reward를 다시 re-compute

$$r(y_i) = \tilde{r}_{\text{dis}}(y_i) + \alpha\,\tilde{b}(y_i) = \tilde{p}(y_i) + \alpha\,\tilde{b}(y_i).$$

- 계산된 response 들의 reward 를 통해 Policy Update

Result

	Method	General	Mathematical Reasoning			Sci. Reasoning	Avg
		MMLU-Pro	MATH-500	AIME 2024	AMC	GPQA	
Qwen2.5-Math-1.5B	***Prompt methods (training-free)***						
	Raw model (Yang et al., 2024)	25.8	32.7	7.7	28.6	27.8	24.5
	CoT (Wei et al., 2023)	27.5	34.1	8.5	29.8	28.6	25.7
	RL methods on training dataset						
	GRPO (Shao et al., 2024)	33.5	72.8	14.3	46.5	26.1	42.3
	REINFORCE (Williams, 1992)	32.5	72.0	14.6	46.0	25.2	40.1
	REINFORCE++ (Hu et al., 2025)	34.3	72.8	15.2	46.8	25.9	41.0
	Test-time scaling methods						
	INTUITOR (Zhao et al., 2025a)	31.1	70.0	12.0	44.5	23.5	38.2
	RLPR (Yu et al., 2025a)	33.9	73.2	16.0	47.0	26.5	41.3
	CO-REWARDING-I (Zhang et al., 2025b)	33.1	72.5	15.5	46.5	25.8	40.7
	TTRL (Zuo et al., 2025)	35.6	73.0	15.8	47.3	26.1	41.5
	DARE (Ours)	**38.9**	**73.6**	**19.8**	**50.2**	**28.5**	**44.2**
Qwen3-1.7B	***Prompt methods (training-free)***						
	Raw model (Yang et al., 2025)	36.8	55.8	11.4	32.6	26.8	32.6
	CoT (Wei et al., 2023)	38.5	57.2	12.3	33.8	27.6	33.8
	RL methods on training dataset						
	GRPO (Shao et al., 2024)	44.5	77.3	24.1	53.2	31.1	48.2
	REINFORCE (Williams, 1992)	44.2	77.5	23.2	52.3	30.2	47.4
	REINFORCE++ (Hu et al., 2025)	45.4	78.0	23.9	52.8	30.8	48.1
	Test-time scaling methods						
	INTUITOR (Zhao et al., 2025a)	42.1	77.5	20.5	50.5	28.5	45.8
	RLPR (Yu et al., 2025a)	45.7	75.5	24.5	53.0	31.0	48.9
	CO-REWARDING-I (Zhang et al., 2025b)	43.2	78.9	23.8	52.5	30.5	47.7
	TTRL (Zuo et al., 2025)	46.9	78.2	24.0	52.9	31.2	48.6
	DARE (Ours)	**48.8**	**79.6**	**26.3**	**55.7**	**32.7**	**50.6**

Table 1. Performance Comparison across two backbones, evaluated on five benchmarks from three task categories. The best and second best results are **highlighted** and underlined, respectively.

- 모든 benchmark 에서 baseline 대배 가장 높은 성능 달성
- 특히 높은 difficulty (AIME) 에서 성능 향상폭이 큼
- 다른 Test-time Scaling methods 뿐만 아니라, Ground Truth가 있는 일반적인 RL methods 들 보다 성능이 더 높음

Result

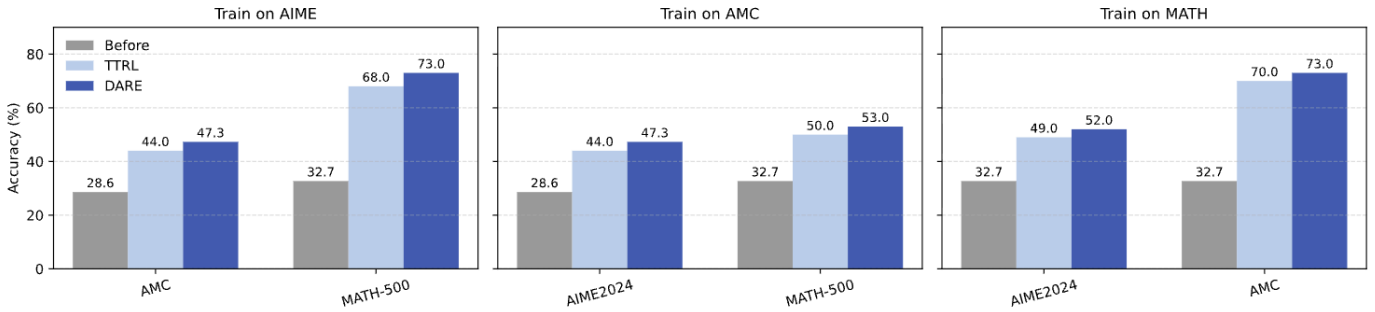


Figure 3. **OOD generalization of Qwen2.5-Math-1.5B.** Each subfigure shows evaluation on OOD benchmarks after adaptation on a training set. Bars indicate pass@1 accuracy for the original model, TTRL, and DARE, with DARE consistently improving performance.

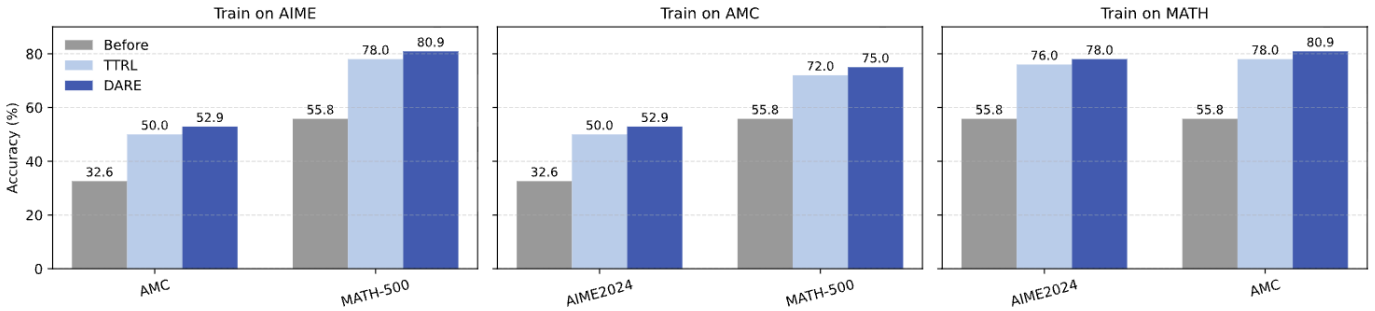
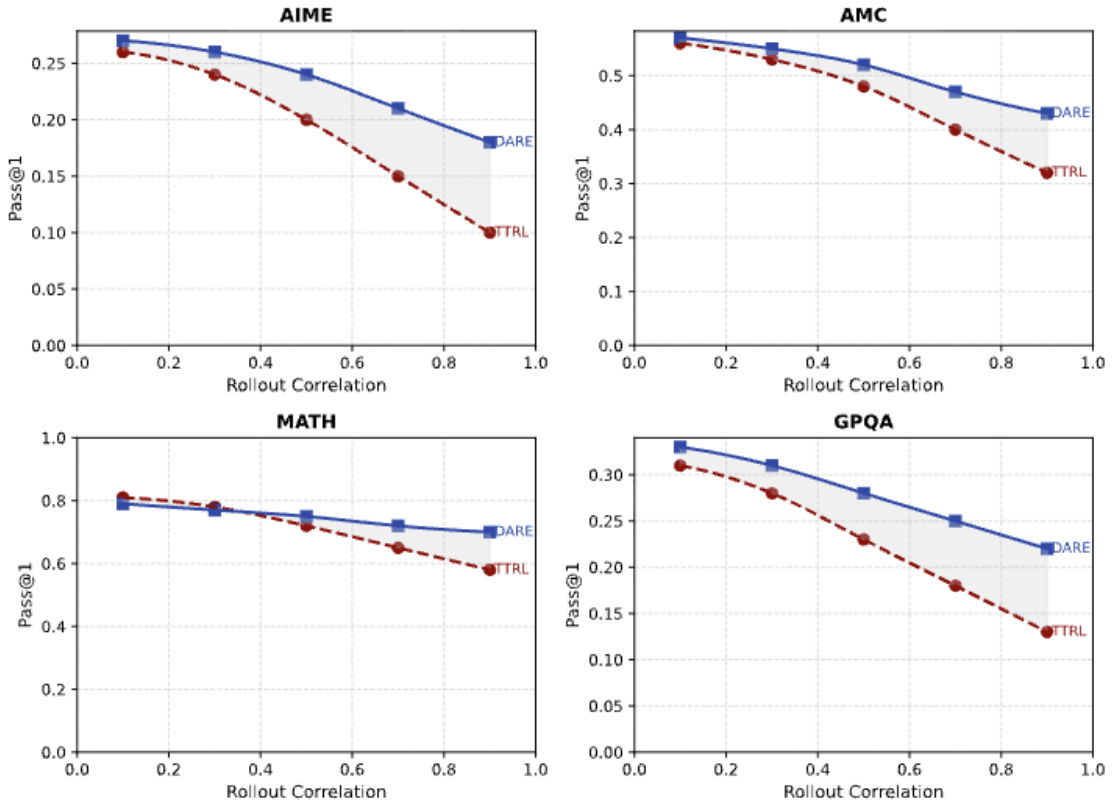


Figure 4. **OOD generalization of Qwen3-1.7B.** Each subfigure shows evaluation on OOD benchmarks after adaptation on a training set. Bars indicate pass@1 accuracy for the original model, TTRL, and DARE, with DARE consistently improving performance.

- OOD 성능에서도 TTRL 보다 더 높은 성능
- Generalization 능력 입증

Result



Rollout Correlation

- Model이 생성한 response 들의 비슷한 정도
- 통계적 correlation의 proxy로 pairwise token overlap 측정
- Decoding hyperparameter 조절을 통해 rollout correlation 통제

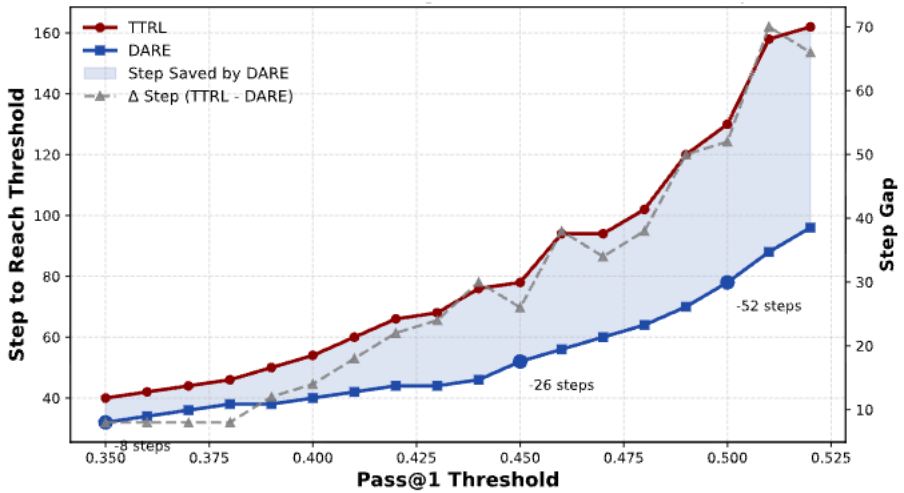
TTRL

- Rollout Correlation 이 증가할수록 빠르게 성능하락
- Majority-Voting 특성상 다수의 오답에 취약해짐을 의미

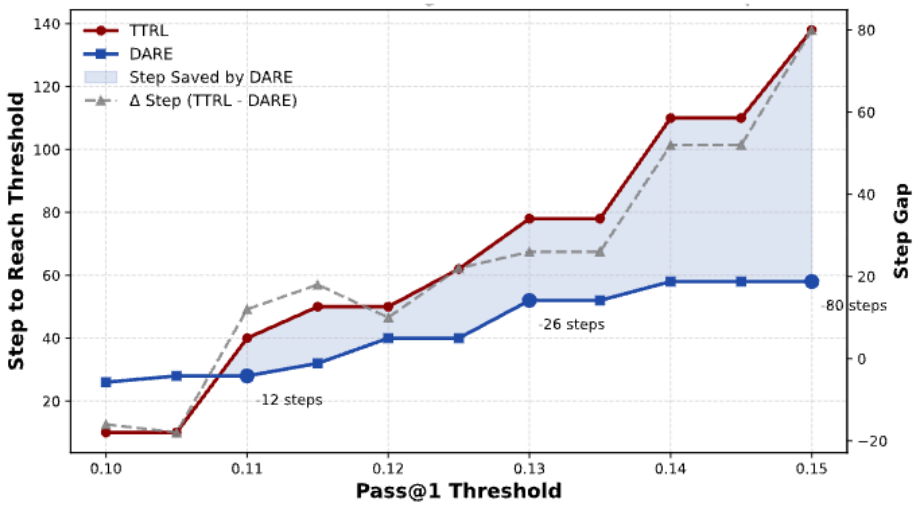
DARE

- 성능하락 폭이 상대적으로 완만
- Frequency 뿐만 아니라 Uncertainty 까지 고려하기 때문
- Exploration bonus 통해서도 low frequent response 도 reward 부여

Result



(a) AMC (Qwen3-1.7B)



(b) AIME (Qwen2.5-Math-1.5B)

- TTRL, DARE 가 목표성능 도달에 필요한 training step 수
- TTRL 대비 적은 training step 이 소요
- Reward signal을 얻는데 더 많은 resource 를 소비하지만, 빠른 성능 향상이 이를 커버

Discussion

1. **Uncertainty=Token-Entropy 가정**
   - Confidently Wrong / Uncertainly Right 인 case가 존재할 수 있음
   - 'Low Entropy=High Reasoning Quality' 가정이 성립하지 않으면, 잘못된 방향으로 shaping

2. **Computational Overhead 증가**
   - 모든 rollout 들의 모든 token 의 entropy를 계산
   - 성능향상 대비 비용 Trade-off (특히 LRM 에서는)

3. **Pruning**
   - Minority에 정답이 있을 수 있음을 강조하지만, 동시에 낮은 확률을 제거
   - Pruning threshold 기준 값에 따라 민감하지 않을까..?? (이에 대한 Ablation은 없음)

Thank you