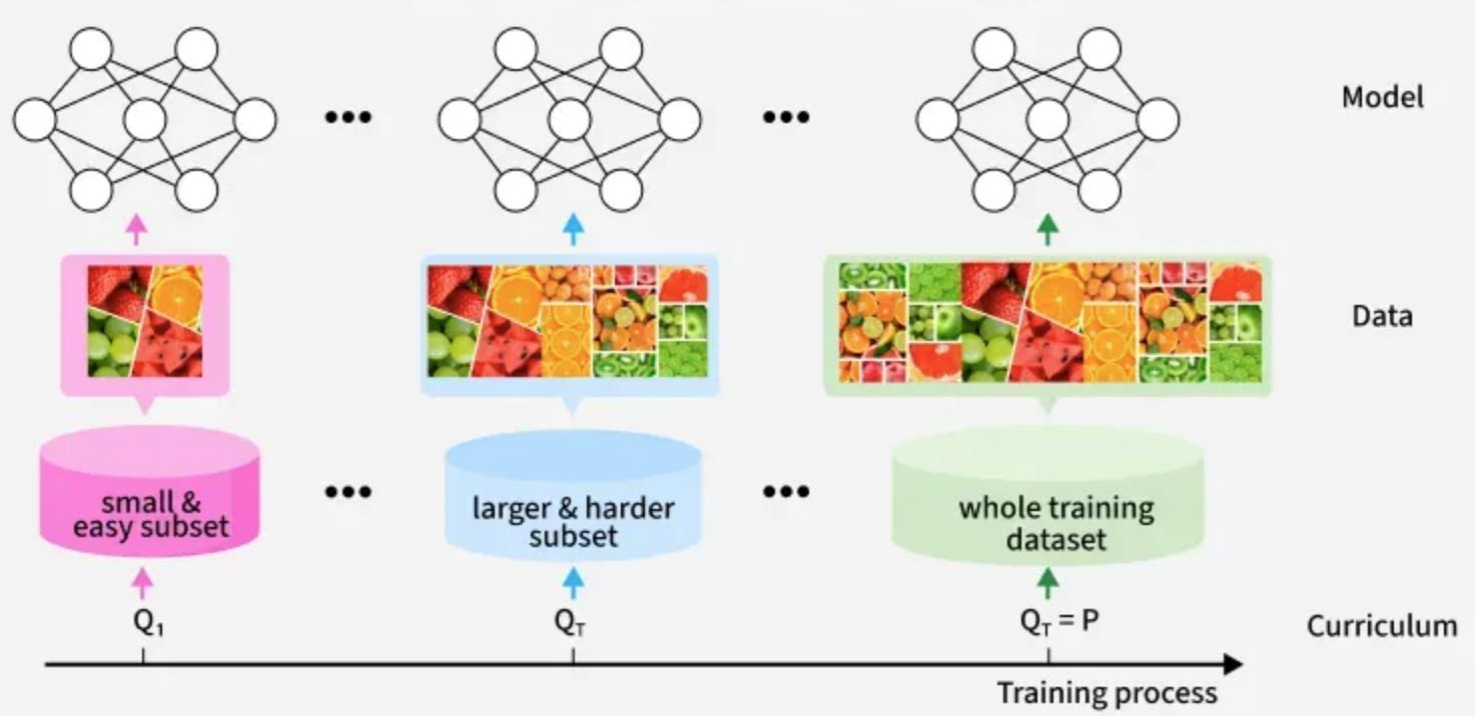
Automatic Curriculum Learning in Reinforcement Fine-tuning

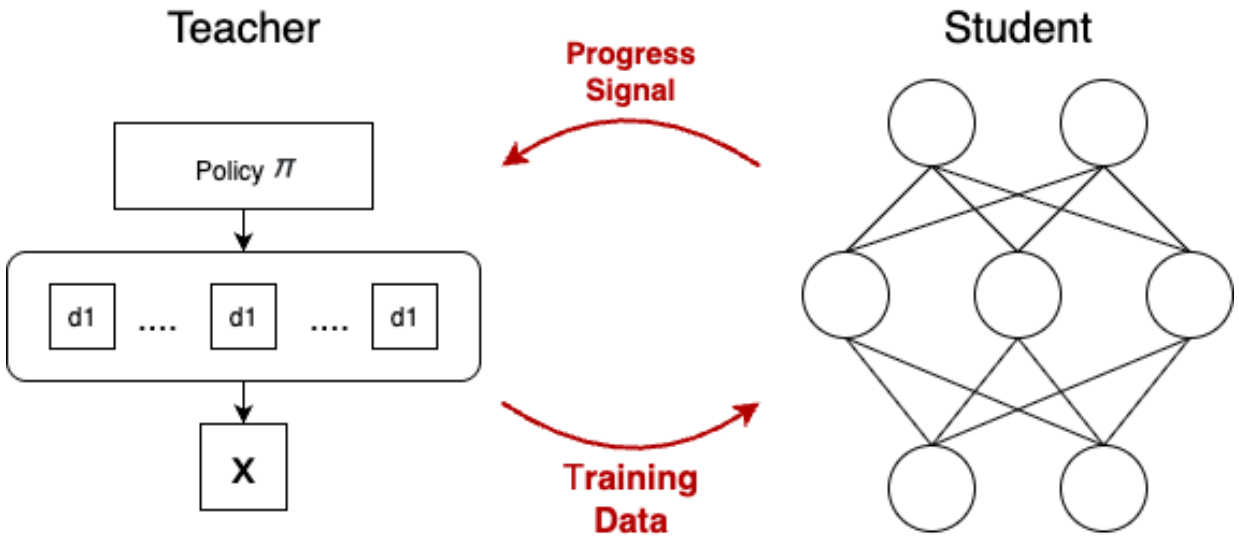
한성빈

Curriculum Learning



- 인간의 학습체계(Easy→Hard)를 적용한 학습 방법론
- Data를 난이도 별로 노출시키며 학습하는 방식
- Model이 점진적으로 능력을 쌓도록 돕는 전략

Automatic Curriculum Learning (ACL)



- Learning curriculum을 임의로 조절하는 기존 방식의 한계 지적
- Model의 상태에 따라 training data의 분포를 automatic하게 조정
- 즉, model의 현재 실력에 따라 '맞춤형' 과제를 자동으로 제공

ACL in Reinforcement Learning

$$Obj: \max_{\mathcal{D}} \int_{T \sim \mathcal{T}_{target}} P_T^N \, \mathrm{d}T$$

$\boldsymbol{T}$ (Task)

$\boldsymbol{\mathcal{T}_{target}}$ (Target Task)

$\boldsymbol{N}$ (Num of training steps)

$\boldsymbol{H}$ (History)

$\boldsymbol{D}: \boldsymbol{H} \rightarrow \boldsymbol{T}$ (Task Selection Function, Curriculum)

$\boldsymbol{P_T^N}$ (Final Performance)

- 다양한 학습 curriculum ($\boldsymbol{D}$) 중에서,
- 정해진 훈련 시간 $\boldsymbol{N}$ 이 끝났을 때,
- Target task $\boldsymbol{T} \sim \boldsymbol{\mathcal{T}_{target}}$ 에서 가장 높은 평균 reward를 기록하게 만드는 $\boldsymbol{D}$을 찾아라.

ACL in Reinforcement Learning

Surrogate Objectives

- 해당 Objective function 은 직접 최적화 하는 것은 불가능
- 이에 Surrogate Objective를 사용하여 이를 통해 최적은 curriculum D를 찾고자 함

Reward

- Agent는 reward 신호를 통해 학습하고, sparse reward 과제는 reward 신호의 가치가 더 큼
- Train batch 생성시, reward를 받은 경험의 비율을 인위적으로 더 높여 학습효율 개선

Intermediate difficulty

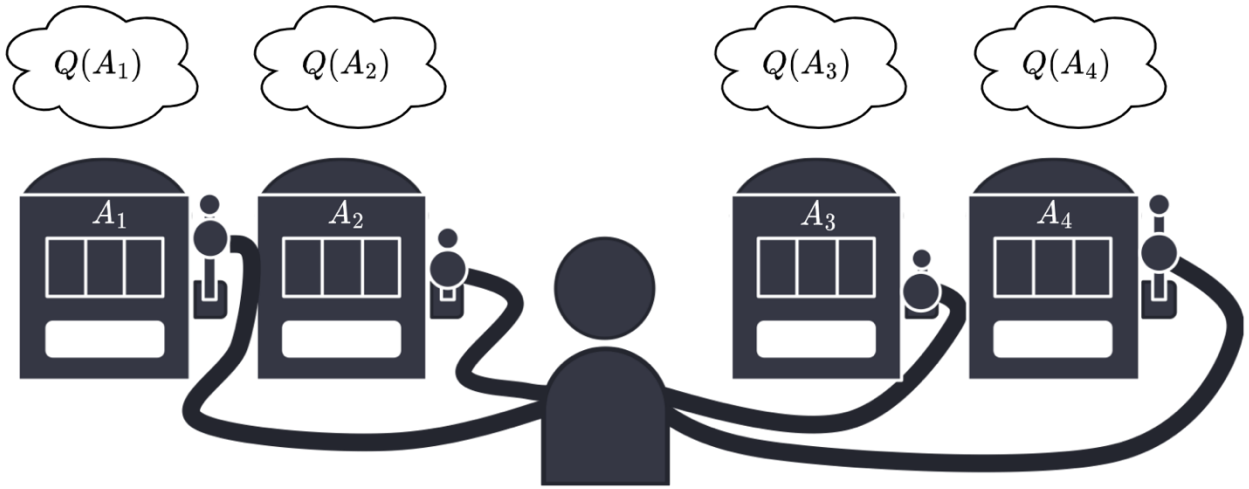
- 가장 많이 'learn' 할 수 있는 과제는 intermediate difficulty 라는 철학에 기반
- 초기 state 조절, Setter-Solver 모델, Env 조절, etc...

Learning Progress

- 가장 빠르게 learning progress가 일어나는 task로 학습
- Intermediate difficulty에서 일어날 수 있는 학습 정체 구간을 극복할 수 있음
- **Multi-armed Bandit(MAB)** 으로 구현

ACL in Reinforcement Learning

Multi-armed Bandit (MAB)



- 제한된 횟수에서, 어떤 슬롯머신의 arm을 당겨야지 최대의 reward을 얻을지 찾는 문제
- [Exploitation/Exploration] trade-off에 대한 개념

ACL in Reinforcement Learning

슬롯 머신	결과		
	Trial 1	Trial 2	Trial 3
a	0	1	5
b	1	0	0

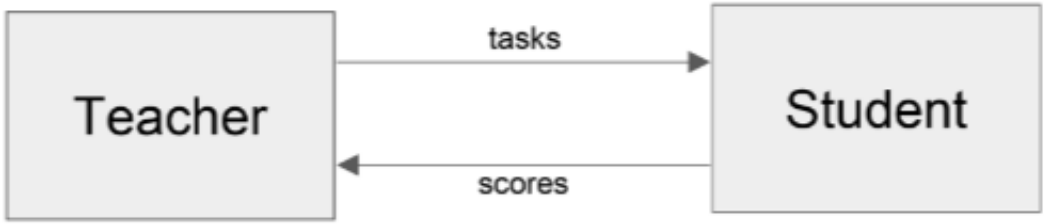
$$Q(a) = \frac{0+1+5}{3} = 2$$

$$Q(b) = \frac{1+0+0}{3} = 0.33$$

- **Exploitation**
  - 현재 결과를 바탕으로, 가장 좋다고 생각되는 슬롯을 당기자!!
  - 현재 가치가 높은 a 슬롯을 당기는 것이 exploitation

- **Exploration**
  - 슬롯머신의 가치를 정확하게 추정하기 위해 다양한 슬롯을 당겨보자.
  - 현재 가치가 낮더라도, b 슬롯을 당겨서 정확한 가치를 추정해보자.

ACL in Reinforcement Learning



Learning Progress 추정 (각 arm의 가치 추정)

- 각 arm 별로 최근 K개의 reward 저장
- K개의 reward에 대해 linear regression 수행
- Linear-regression의 기울기를 learning progress로 추정

Arm 선택 알고리즘 (Exploitation/Exploration)

- **Boltzmann Exploration**
  - 각 task마다 learning progress를 softmax
  - 이를 확률분포로 사용하여 arm 선택
  - $$p(a) = \frac{e^{Q_t(a)/\tau}}{\sum_{i=1}^{N} e^{Q_t(i)/\tau}}$$
- **Thomson Sampling**
  - 각 arm별로 저장 된 learning progress에서 sampling
  - Sampling 된 값 중, 절대값이 가장 큰 arm을 선택

Efficient Reinforcement Finetuning via Adaptive Curriculum Learning

Taiwei Shi[†], **Yiyang Wu**[†], **Linxin Song**[†], **Tianyi Zhou**[▽], **Jieyu Zhao**[†]
[†]University of Southern California, [▽]University of Maryland, College Park
{taiweish, wuwendy, linxinso, jieyuz}@usc.edu, tianyi@umd.edu

Motivation

문제 정의

- **높은 비용, 낮은 효율성**
  - 방대한 샘플 & 컴퓨팅 자원 필요
  - RL 특성상 Sample Efficiency 가 낮음
- **학습 정체**
  - 너무 쉽거나 어려운 문제에 자원을 낭비

기존 접근 방식의 한계

- **알고리즘 단순화**
  - PPO와 같은 RL 알고리즘 자체를 더 간단하게 만들려는 시도
  - 학습의 안정성을 해치거나, 성능 저하 감수
- **데이터 필터링**
  - 학습에 사용할 data를 미리 filtering, 혹은 미리 중요도를 매김
  - Data 선별 과정부터 과도한 overhead
- **초기 커리큘럼 학습**
  - Manual한 단계 설정이 대부분
  - Model의 상태에 맞춘 실시간 적응 불가

Contribution

ADARFT(Adaptive Curriculum Reinforcement Finetuning) 제안

- 학습은 작업이 너무 쉽지도 어렵지도 않을 때 가장 효과적이라는 직관에 기반
- 모델의 성공률(Reward)에 따라 목표 난이도를 실시간으로 조정
- 항상 '최적의 난이도' 구간에서 학습하도록 유도

가볍고 범용적인 설계

- 기존 RFT 파이프라인에 손쉽게 추가 가능
- 보상 함수나 모델 아키텍처 수정 불필요
- 다양한 RL 알고리즘에 바로 적용 가능

Methodology

Dynamic Curriculum Sampling

- 현재 model의 수준에 맞는 training-data를 선택하는 과정

1. 사전 난이도 점수 계산

- 훈련 데이터셋 $\boldsymbol{D}$에 포함된 모든 문제에는 사전에 계산된 난이도 점수 $\boldsymbol{d_i}$가 부여
- 다른 모델의 성공률을 기반으로 계산 (pass@k)
- 본 논문에서는 Qwen-2.5-Math-7B 를 통해 사전 난이도 계산

2. 목표 난이도 $\boldsymbol{T}$ 설정

- ADARFT는 '목표 난이도' $\boldsymbol{T}$ 라는 값을 유지
- 현재 모델이 집중적으로 훈련해야 할 문제의 난이도 수준을 의미

3. Sample 선택

- 모든 문제의 난이도 $\boldsymbol{d_i}$ 와 현재 목표 난이도 $\boldsymbol{T}$ 사이의 절대값 차이 $\Delta_i = |\boldsymbol{d_i} - \boldsymbol{T}|$ 를 계산
- Δ_i 가 작은 상위 B개의 문제를 이번 스텝의 훈련 배치 $\boldsymbol{X}$ 로 선택

Methodology

Policy Update

- 선택 된 training-data로 model을 훈련시키는 과정

1. 응답 및 보상 계산

- 샘플링 된 배치 $\boldsymbol{X}$ 에 대해, 현재 정책 모델 $\boldsymbol{\pi_\theta}$가 응답 $\boldsymbol{G}$ 생성
- 각 응답이 정답인지 여부에 따라 이진 보상(correct=1, incorrect=0)이 주어짐
- 배치 $\boldsymbol{X}$ 의 평균 보상 $\boldsymbol{R_{avg}}$ 계산

2. 정책 모델 업데이트

- 표준 RL 알고리즘을 통해 정책 모델 $\boldsymbol{\pi_\theta}$ 의 파라미터 업데이트
- 알고리즘은 PPO, GRPO와 같은 표준 RL 알고리즘 사용

ADARFT는 Policy update과정에서 알고리즘을 수정하지 않음 (wrapper)

Methodology

Target Difficulty Update

- 훈련 배치의 평균 보상 $\boldsymbol{R_{avg}}$ 을 바탕으로, 다음 스텝의 목표 난이도 $\boldsymbol{T}$ 동적으로 조절

- $\boldsymbol{T' = \text{clip}\left(T + \eta \cdot \tanh\left(\alpha \cdot \left(R_{avg} - \beta\right)\right), d_{min}, d_{max}\right)}$

  - $\boldsymbol{\beta}$ (target reward): 모델이 도달하길 바라는 목표 reward

  - $\boldsymbol{\eta}$ (step size): 목표 난이도를 한 번에 얼마나 변경할지 결정(learning rate)

  - $\boldsymbol{\alpha}$ *(sensitivity):* R_{avg}, β 의 차이에 얼마나 민감하게 반응할지 조절

- **모델의 현재 실력에 맞는 도전적인 과제를 꾸준히 제공하는 ADARFT의 핵심 원리**

Methodology

본 논문에서는 목표 보상 β=0.5로 설정

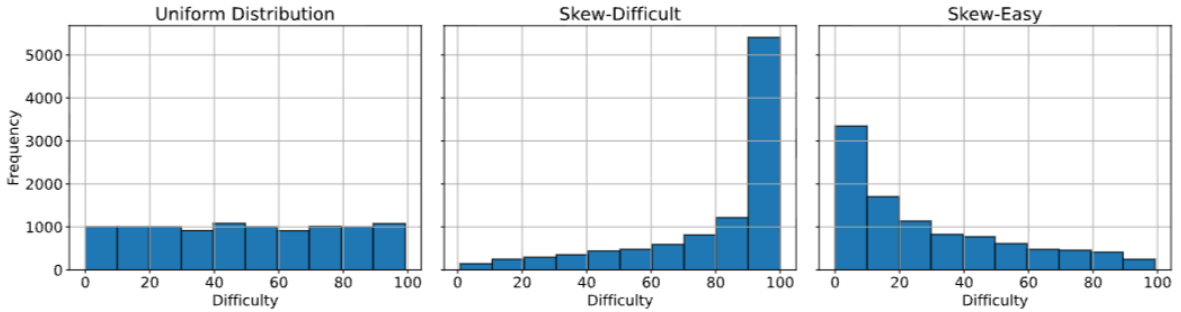
- **Intuition**
  - **너무 쉬우면?**
    - 모델이 이미 아는 것을 푸는 것이므로 새로운 것을 배우기 어렵고, learning signal이 약함
  - **너무 어려우면?**
    - 어떻게 개선해야 할지에 대한 reward를 거의 얻지 못하고, learning signal이 약함
  - **적당히 어려운 문제 (성공률 ~50%)**
    - 성공한 시도와 실패한 시도를 비교하며, 무엇이 정답으로 이끄는지를 가장 효과적으로 학습할 수 있음

- **Theoretical Proof**
  - $D_{KL}(\pi_{init}|\pi^*) \geq \frac{p(x)(1-p(x))}{2\tau^2}$
  - *KL Divergence가 크다는 것은,* π^*와 차이가 크다는 것.
  - 즉, KL Divergence가 곧 learning signal의 크기
  - 이때, p(x) 가 0.5 일 때, KL-Divergence 하한선의 최대값이 된다.

Experiment Setup

Dataset

- **훈련 데이터:** DeepScaleR (10,000 samples each)
- **난이도 분포:** Skew-Difficult, Skew-Easy, Uniform



Model

- Qwen 2.5 7B
- Qwen 2.5 MATH 1.5B

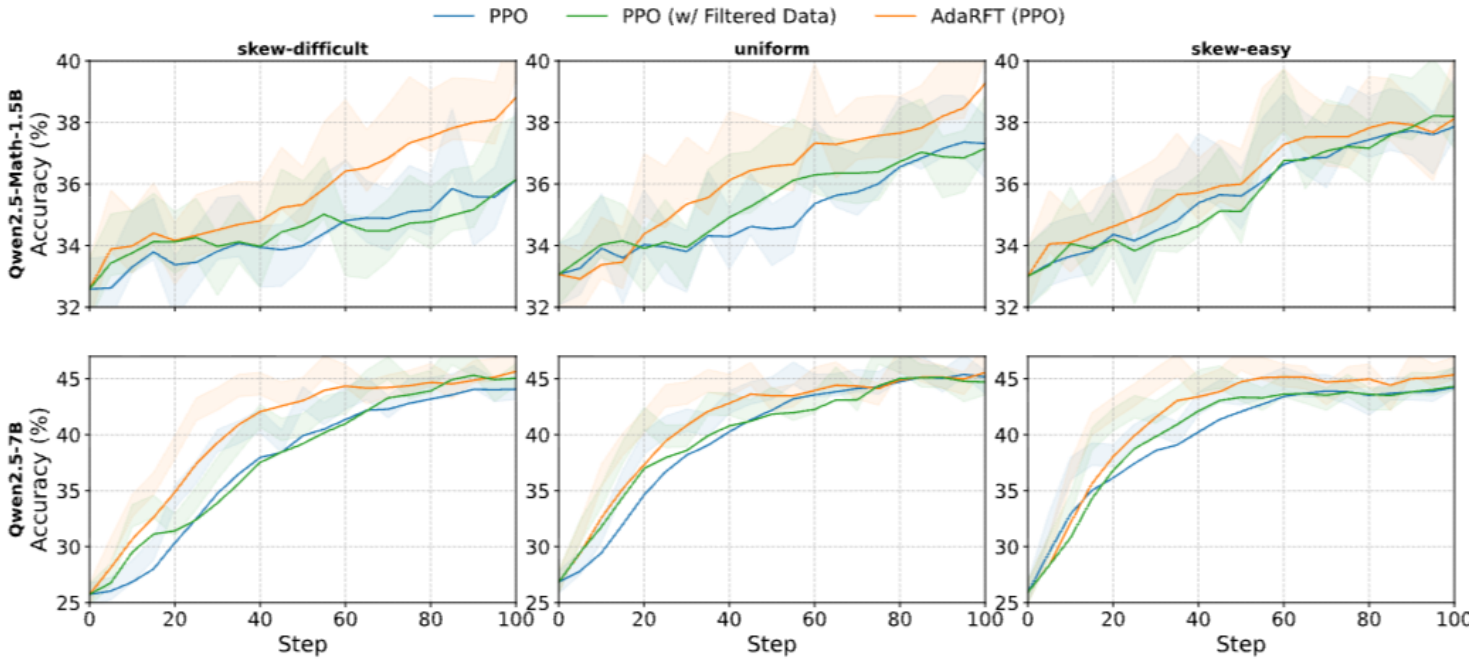
Policy train 방법

- 표준 PPO
- Filtered-Data PPO
- ADARFT (PPO)

Hyper-Parameter

- **배치 크기:** 1024
- **목표 보상:** $\beta = 0.5$
- **민감도 & 스텝 크기:** $\alpha = 2$, $\eta = 50$
- **훈련 환경:** 8개 A100 GPU, 약 100 step

Result



- **전반적인 성능 우위**
  - 학습 효율성과 최종 성능 모두 뛰어남

- **데이터 분포의 영향**
  - skew-difficult에서 성능 향상이 두드러짐
  - 쉬운 문제부터 시작하여 점차 난이도를 높여가므로 안정적으로 성능이 향상

- **모델 크기의 영향**
  - Qwen2.5-Math-1.5B에서 더 크게 나타남
  - Small model에서 체계적인 가이드가 학습에 더 큰 도움을 준다는 것을 시사

Result

Model	Setup	Method	Avg Step Time (s)	Extra Steps (%)
Qwen2.5 Math 1.5B	skew-difficult	ADARFT	**122.24**	**+0 (0.0%)**
		PPO	132.95	+43 (71.7%)
		PPO (w/ Filter)	128.20	+49 (81.7%)
	uniform	ADARFT	**121.31**	**+0 (0.0%)**
		PPO	126.82	+34 (56.7%)
		PPO (w/ Filter)	126.35	+52 (86.7%)
	skew-easy	ADARFT	120.52	**+0 (0.0%)**
		PPO	121.15	+16 (26.7%)
		PPO (w/ Filter)	**115.12**	+21 (35.0%)
Qwen2.5 7B	skew-difficult	ADARFT	**239.92**	**+0 (0.0%)**
		PPO	246.21	+24 (60.0%)
		PPO (w/ Filter)	254.22	+25 (62.5%)
	uniform	ADARFT	**234.16**	**+0 (0.0%)**
		PPO	243.82	+13 (32.5%)
		PPO (w/ Filter)	263.11	+23 (57.5%)
	skew-easy	ADARFT	247.44	**+0 (0.0%)**
		PPO	235.27	+20 (50.0%)
		PPO (w/ Filter)	**233.13**	+17 (42.5%)

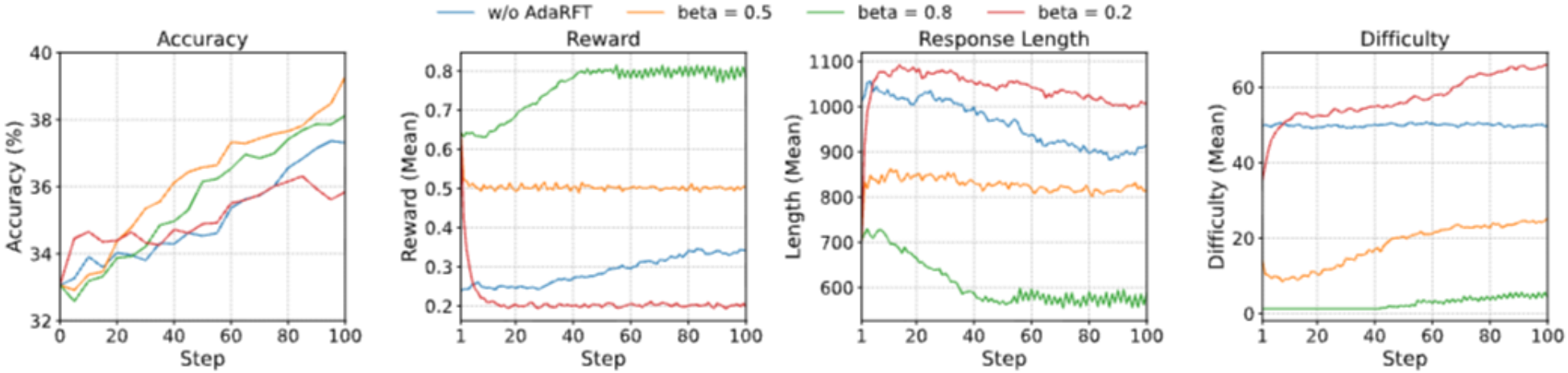
Avg Step Time
- 100 step 까지 소요시간 평균

Extra Steps
- AdaRFT의 60-step의 성능을 도달하기까지 필요한 추가 step 횟수

계산 효율성 (Avg Step Time)
- 대부분의 경우, ADARFT는 스텝 당 평균 훈련 시간도 가장 짧음
- 훈련 초기에 상대적으로 쉬운 문제들을 먼저 학습하여, roll-out 이 짧기 때문

샘플 효율성 (Extra Steps)
- 모든 실험 환경에서 ADARFT는 가장 빠르게 목표 성능에 도달
- 특히 skew-difficult 환경에서 효과가 두드러짐
- ADARFT가 어려운 문제만 있는 환경에서도 효율적으로 성능을 끌어올린다는 것을 의미

Result



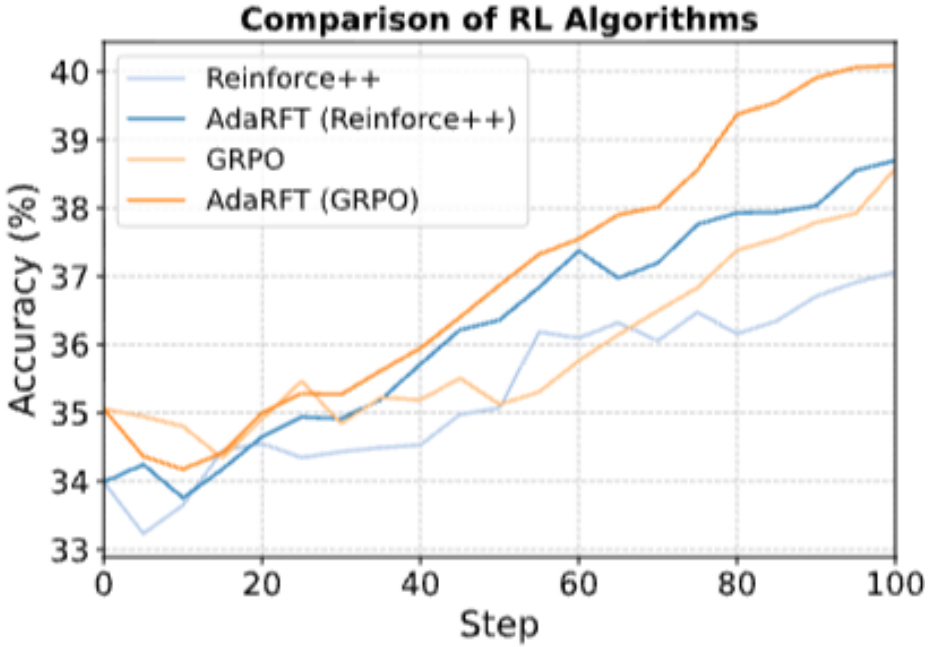
Accuracy (정확도)
- β = 0.5 일때 모델의 최종 정확도가 가장 높게 나타남
- 50%의 성공률을 보이는 데이터로 학습시, 성능이 향상된다는 것을 의미

Reward (보상)
- β 설정에 따라 모델이 받는 평균 보상 값이 목표치 근처에서 유지

Response Length (응답 길이)
- 문제의 난이도와 응답 길이는 비례하는 경향

Difficulty (난이도)
- 낮은 성공률을 목표로 할 수록, 더 어려운 문제를 제시

Result



- ADARFT는 기반이 되는 RL 알고리즘의 종류와 상관없이 성능개선
- RL 최적화 방법과 독립적으로 작동함을 보여줌
- ADARFT가 다양한 강화학습 미세 조정(RFT) 파이프라인에 쉽게 통합할 수 있음을 보여줌

Self-Evolving Curriculum for LLM Reasoning

Xiaoyin Chen[*1,2], **Jiarui Lu**[1,2], **Minsu Kim**[1,3], **Dinghuai Zhang**[1,4], **Jian Tang**[1,5]
Alexandre Piché[6], **Nicolas Gontier**[6] ,**Yoshua Bengio**[1,2], **Ehsan Kamalloo**[6]
[1]Mila – Quebec AI Institute, [2] Université de Montréal
[3]KAIST, [4]Microsoft Research, [5]HEC Montréal, [6]ServiceNow Research
xiaoyin.chen@mila.quebec

Motivation

임계값 방식(AdaRFT)의 한계

- **순서 의존성**
  - 반드시 '난이도' 라는 단일 축 위에 정렬 되어 있어야 함
  - 같은 평가 기준으로 평가 가능한 task에만 적용 가능

- **비순서형 task 적용 어려움**
  - '수학', '코딩', '논리 추론' 과 같은 모든 task에 대해서 학습이 되어야 함
  - 그러나, 순서가 없는 여러 종류의 문제를 단일 난이도 축으로 표현하기 어려움

Contribution

Curriculum 선택 문제를 Multi-armed Bandit 문제로 정의

- Curriculum 선택을 MAB 문제로 공식화
- 각 문제 범주를 개별적인 Arm으로 취급

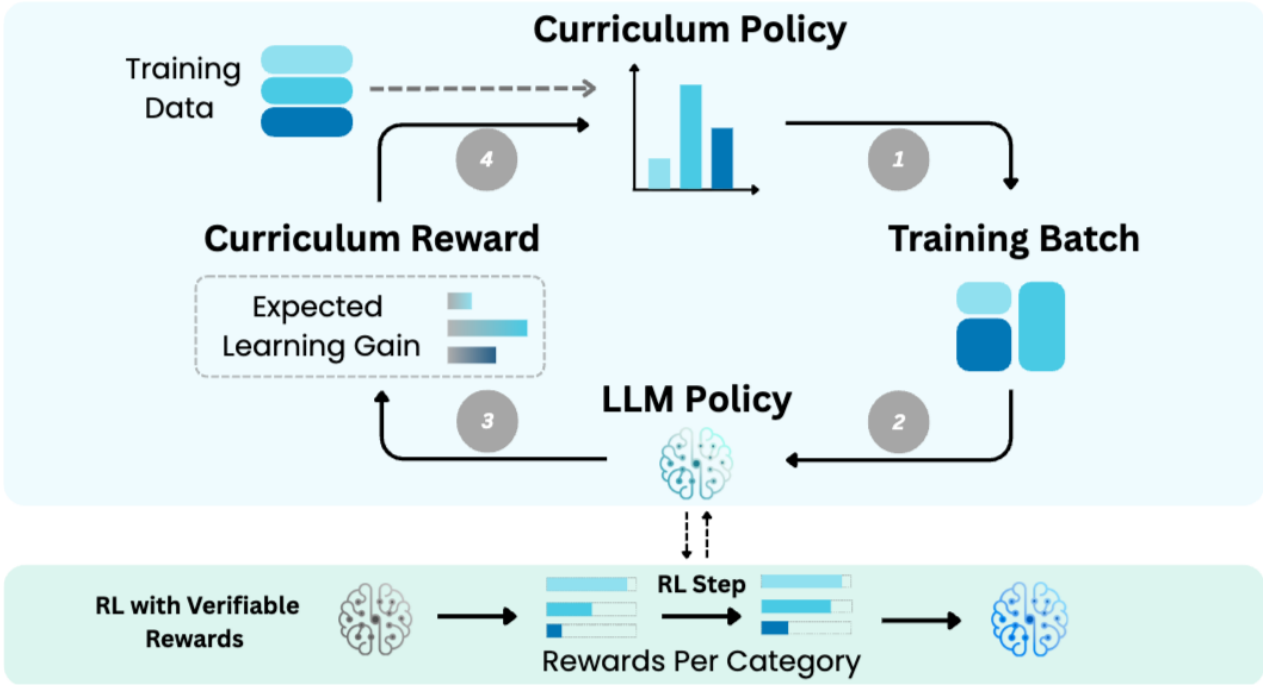
범주의 유연성

- 각 arm은 서로 독립적으로 평가
- 따라서, 순서가 없는 다른 범주로도 curriculum learning이 가능

다차원 curriculum 설계

- 범주의 유연성을 통해 복잡한 curriculum 설계 가능
- 실제로 본 논문에서는 '난이도 3개'x'3가지 문제유형' = 9개 arm을 생성

Methodology



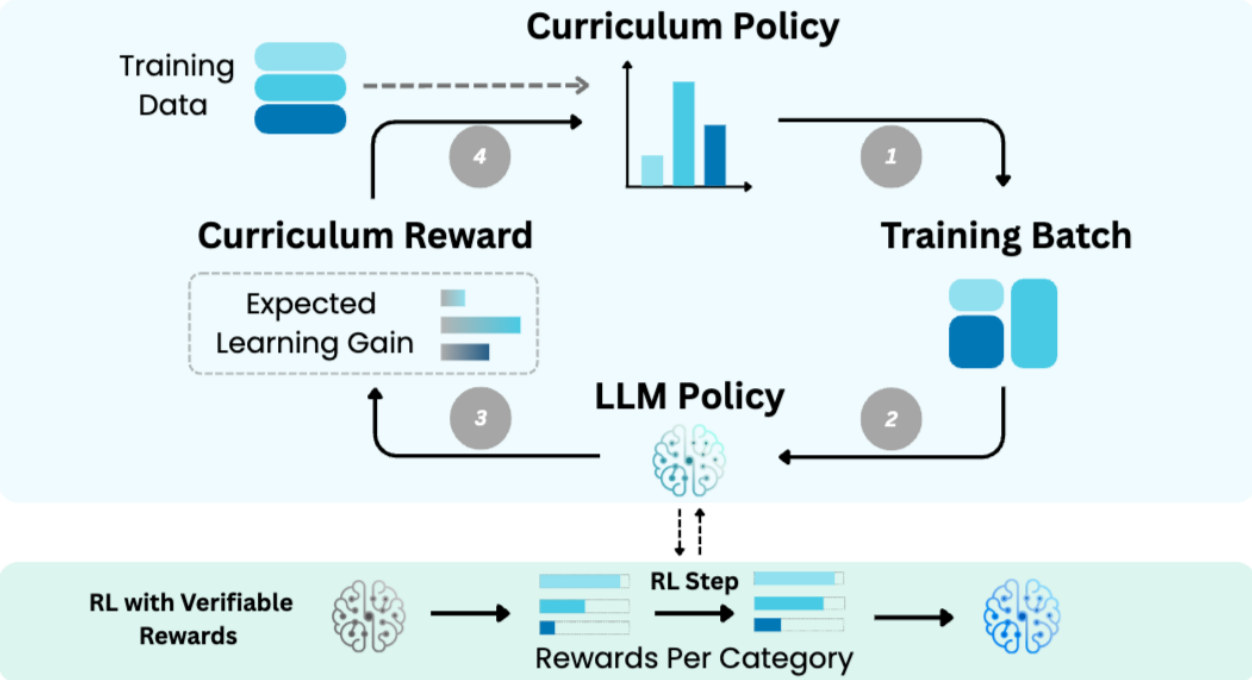
1. Training batch sampling

- Curriculum Policy를 통해 각 category 에서 data sampling *(Boltzmann Exploration)*
- Curriculum Policy는 현재 모델에서, 큰 학습효과를 줄 것이라 기대되는 category에서 자주 sampling 하도록 설계

2. LLM 학습

- Sampling 된 train-batch로 LLM 업데이트
- 논문에서는 GRPO, PPO, RLOO 사용

Methodology



3. Curriculum Reward 계산

- 직전 training의 학습 성과(Learning gain)를 계산하는 단계
- Learning gain의 surrogate objective로 Absolute Advantage 사용
- 각 arm 별로 어떤 arm이 현재 model에 유익한지 평가

Policy Gradient Loss

$$\mathcal{L}_{\mathrm{PG}}(\theta) = -\mathbb{E}_{(s_t,a_t)\sim\pi_\theta}\left[\log \pi_\theta(a_t \mid s_t)\ \widehat{A}_t\right]$$

Policy gradient

Policy Gradient L2-Norm

$$\|\nabla_\theta \mathcal{L}_{\mathrm{PG}}(\theta, s_t, a_t)\|_2 = \left\|\mathbb{E}_{(s_t,a_t)\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t)\ \widehat{A}_t\right]\right\|_2$$

$$\approx |\widehat{A}_t| \|\nabla_\theta \log \pi_\theta(a_t \mid s_t)\ \|_2$$

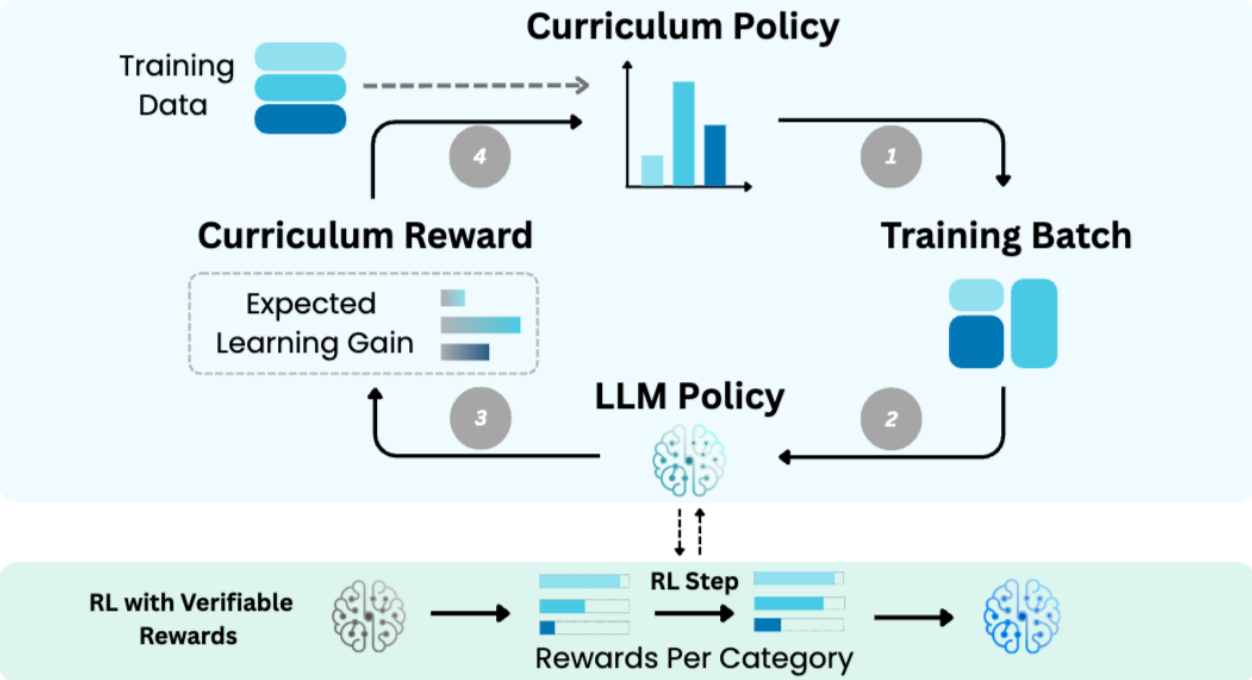
학습이 잘 된다
➔ *Gradient의 크기가 크다 (저자의 주장!!)*
➔ *Gradient의 크기는 Advantage의 절대값과 비례한다*
➔ *Advantage의 절대값이 크면, 학습이 잘 된다*

Absolute Advantage

$$r(c) = \mathbb{E}_{(s_t,a_t)\sim\pi_\theta(x_i),x_i\sim c}\ |\widehat{A}_t|$$

r(c) : category c의 평균 Absolute advantage
r(c)가 크다는 것은, category c의 학습성과가 좋다는 뜻

Methodology



4. Curriculum Policy update

- **3** 에서 계산 된 Curriculum Reward로 Curriculum Policy 갱신
- Non-stationarity(비정상) Multi-armed Bandit 문제로 접근
  - Model이 학습됨에 따라 최적의 category가 달라지기 때문
- TD(0)를 사용하여, category의 기대보상 업데이트

TD(0) Update

$$Q_{t+1}(c) = \alpha r_t(c) + (1-\alpha)Q_t(c)$$

Qt(c) : 시간 t에서 category c의 기대보상
rt(c) : 시간 t에서 category c를 학습시킬 때 얻는 reward

EMA를 통해 최근 reward에 더 높은 가중치 부여
(Model의 성능이 바뀌는 비정상 문제이기 때문!!)

Experiment Setup

Dataset

- **Planning**
  - Countdown (Synthetic)
  - Zebra Puzzles (Synthetic)
- **Inductive reasoning**
  - ARC-1D (Synthetic)
- **Mathematics**
  - MATH50
  - AMC 22-23
  - AIME 24

Countdown

Using the numbers [5, 17, 91], create an equation that equals 113.

You can only use basic arithmetic operations (+, -, *, /) and each number should be used exactly once.

Return the final answer in ₩boxed{}, for example ₩boxed{(1 +2) / 3}.

Zebra Puzzle

This is a logic puzzle. There are 3 houses (numbered 1 on the left, 3 on the right), from the perspective of someone standing across the street from them. Each has a different person in them.
They have different characteristics:

- Each person has a unique name: arnold, bob, alice
- Everyone has a different favorite cigar: dunhill, prince, pall mall
- The people keep different animals: cat, dog, bird

1. The bird keeper is directly left of the Dunhill smoker.
2. Alice is the dog owner.
3. Arnold is in the second house.
4. Alice is the Prince smoker.
5. Arnold is the cat lover.

What is Name of the person who lives in House 1?

Experiment Setup

Dataset

- **Planning**
  - Countdown (Synthetic)
  - Zebra Puzzles (Synthetic)
- **Inductive reasoning**
  - ARC-1D (Synthetic)
- **Mathematics**
  - MATH50
  - AMC 22-23
  - AIME 24

ARC-1D

Find the common rule that maps an input grid to an output grid, given the examples below.

Example 1:
Input: 1 2 1 2 1 0 0 1 2 0
Output: 0 0 0 1 1 1 1 2 2 2

Example 2:
Input: 1 2 0 0 0 0 2 0 1 2
Output: 0 0 0 0 0 1 1 2 2 2

Example 3:
Input: 0 0 2 0 0 0 0 1 1 0
Output: 0 0 0 0 0 0 0 1 1 2

Below is a test input grid. Predict the corresponding output grid by applying the rule you found. Describe how you derived the rule and your overall reasoning process in detail before you submit your answer. Your final answer must be placed in ₩boxed{} and should be just the test output grid itself.

Input: 0 0 2 0 0 1 1 0 0 1

Experiment Setup

Dataset

- Synthetic Dataset은 4개의 난이도로 생성
- Train시에는 하위 3개의 난이도만 사용
- 제일 어려운 난이도는 OOD 평가용으로 사용

Model

- Qwen 2.5 3B
- Qwen 2.57B

Policy Algorith,

- GRPO

평가 지표

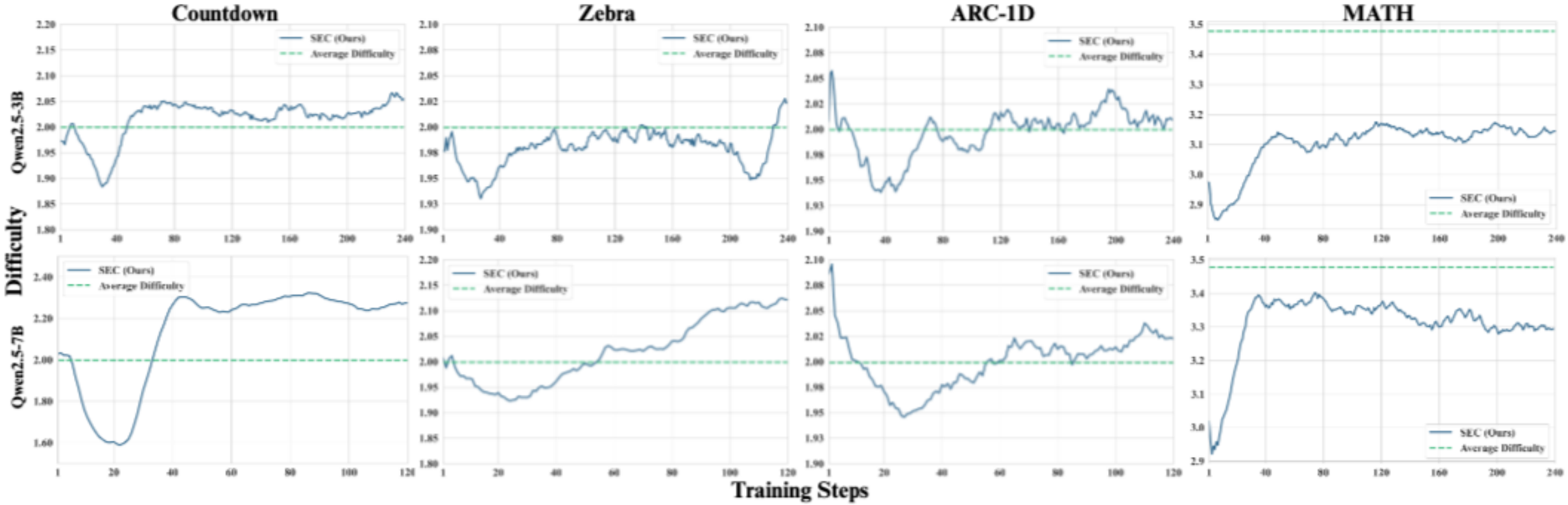
- pass@1 rate
- 8번 독립적인 답변의 평균

Result

Task	Split	Qwen2.5 3B			Qwen2.5 7B		
		Random	**Ordered**	**SEC (Ours)**	**Random**	**Ordered**	**SEC (Ours)**
Countdown	ID	0.859	0.551	**0.866**	0.858	0.820	**0.872**
	OOD	0.479	0.321	**0.542**	**0.566**	0.442	0.555
Zebra	ID	0.517	0.534	**0.547**	0.573	0.572	**0.587**
	OOD	0.285	0.329	**0.345**	0.321	0.311	**0.355**
ARC-1D	ID	**0.501**	0.476	**0.500**	0.512	**0.526**	0.514
	OOD	0.313	0.363	**0.381**	**0.436**	0.428	0.418
Math	MATH500	0.668	**0.672**	**0.672**	**0.774**	0.759	0.761
	AMC22-23	0.345	**0.352**	**0.351**	0.486	0.477	**0.511**
	AIME24	0.075	0.054	**0.100**	0.138	0.150	**0.175**

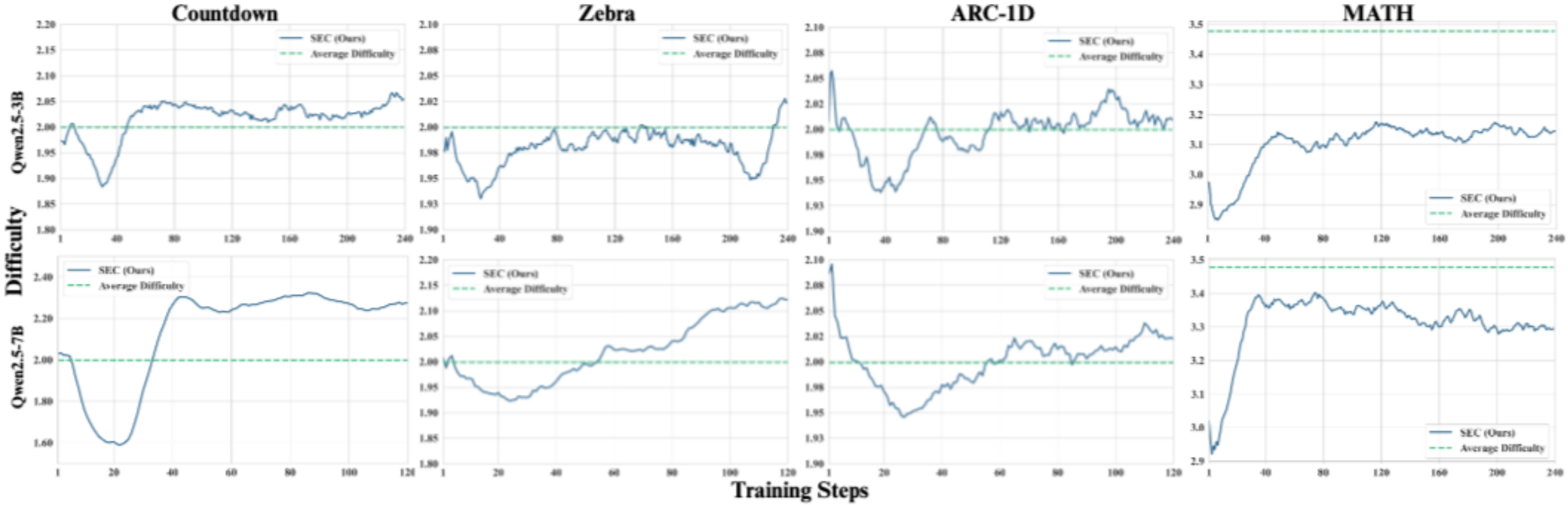
- 대부분의 과제에서 SEC가 높은 성능을 보임
- 특히, OOD 데이터셋에서 성능 향상이 두드러짐 (일반화 능력 향상)
- 7B에서는 성능 차이가 감소 (이미 model에서 강력한 추론 능력 보유)

Result



- **Easy-to-Hard 경향**
- **적응적 난이도 조절**
  - 단순히 직선으로 상승하지 않고 계속해서 변동
  - 학습에 어려움을 겪는다면, 쉬운 문제로 돌아간 후 다시 어려운 문제로 나아감
- **모델 능력에 따른 맞춤형 커리큘럼**
  - 3B 모델의 평균 난이도가 7B보다 대체로 낮음

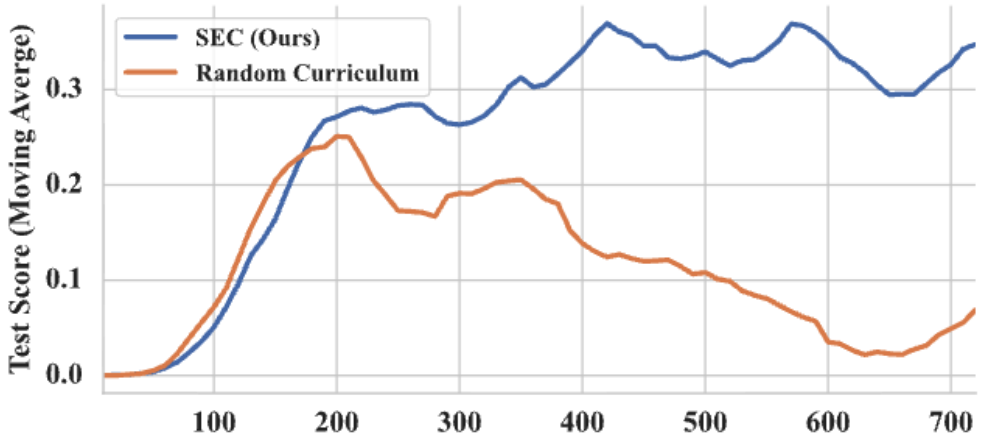
Result



- **Easy-to-Hard 경향**
- **적응적 난이도 조절**
  - 단순히 직선으로 상승하지 않고 계속해서 변동
  - 학습에 어려움을 겪는다면, 쉬운 문제로 돌아간 후 다시 어려운 문제로 나아감
- **모델 능력에 따른 맞춤형 커리큘럼**
  - 3B 모델의 평균 난이도가 7B보다 대체로 낮음

Result

Task	Split	Random	SEC-2D
Countdown	ID	0.837	**0.839**
	OOD	0.418	**0.428**
Zebra	ID	0.513	**0.539**
	OOD	0.254	**0.312**
ARC	ID	0.380	**0.418**
	OOD	0.251	**0.327**



Multiple tasks training

- 3가지 task, 3가지 난이도에 대해 9개의 arm 생성
- 여러 종류의 추론 능력을 동시에 향상 시켜야 하는 task

성능 향상

- 모든 task에서 random curriculum에 비해 일관 된 성능 향상

성능 붕괴 (OOD data)

- Random curriculum
  - 여러 task를 무작위 하게 학습시켜, 학습 능력을 잃은 forgetting 현상 발생
- SEC
  - SEC는 모든 task에 대해 균형있게 학습을 조절하여, forgetting 현상 방지

Conclusion

Contributions

- SEC는 특히 OOD data에 대한 일반화 성능을 일관되게 향상
- 여러 task를 동시에 학습시킬 때, SEC는 각 task의 균형을 맞추는 능력을 보여줌
- 이는 범용적인 학습 상황에 있어, 더욱 효과적으로 작용 될 것임

Limitations

- 학습률(α), temperature(τ)의 hyperparameter에 민감하여, 섬세한 조정 필요

Future work

- Embedding 기반의 clustering을 통해, data의 특성까지 반영하여 cluster 구성
- Light weight 모델을 사용해, curriculum reward를 계산하여, LLM이 직접 계산해야 하는 overhead 감소

감사합니다.