Sharing is Caring: Efficient LM Post-Training with Collective RL Experience Sharing

2025.09.16

송건학

https://huggingface.co/papers/2509.08721

https://arxiv.org/abs/2509.08721v1

Sharing is Caring: Efficient LM Post-Training with Collective RL Experience Sharing

"비동기 기반 분산형 Foundation Model의 Experience Sharing 기반 RL 학습의 효율성 제안경험 공유(SAPO), **학습 속도 가속화 및 성능 94% 향상**"

Algorithm 1: SAPO

Input: For each $n \in [N]$: dataset $\mathcal{D}^n$, metadata $\mathcal{M}^n$, policy π^n, reward model ρ^n, policy update algorithm, number of local samples I^n, number of external samples J^n

Output: Updated policy parameters

for *each round* t **do**

  for *each node* n **do** `// Can be fully decentralized and run in parallel`

    `// Sample questions`

    $\mathcal{B}^n \leftarrow \text{SampleBatch}(\mathcal{Q}^n)$

    `// Generate rollouts`

    for *each* $q \in \mathcal{B}^n$ **do**

      $\mathcal{R}^n(q) \leftarrow \{a_1^n(q), \ldots, a_{L^n}^n(q)\}$

    `// Share rollouts and associated data`

    $\mathcal{S}^n \leftarrow \text{SelectSubset}(\mathcal{B}^n)$

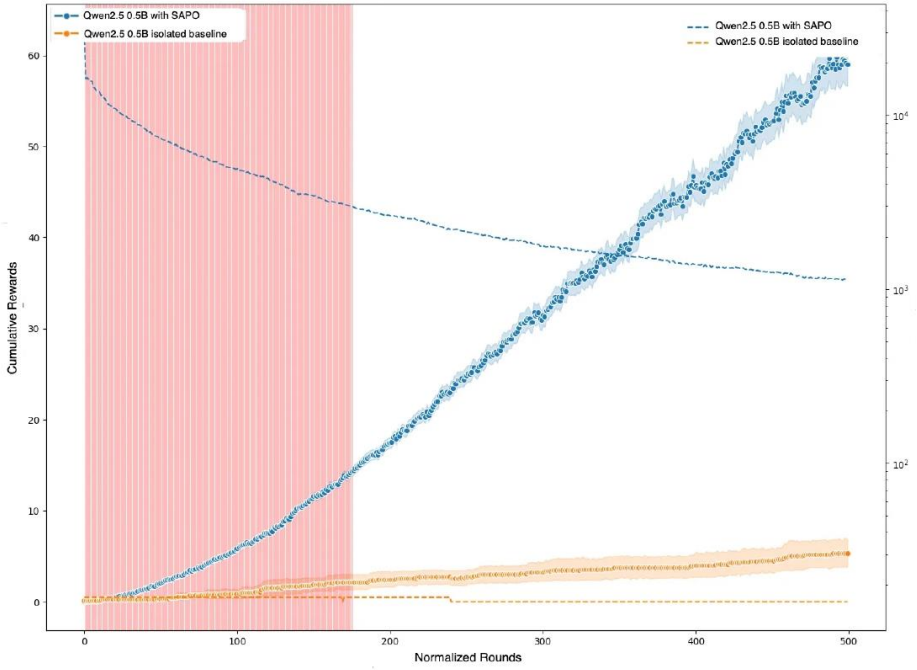
    $\text{Communicate}(\{\mathcal{C}^n(q) \mid q \in \mathcal{S}^n\})$

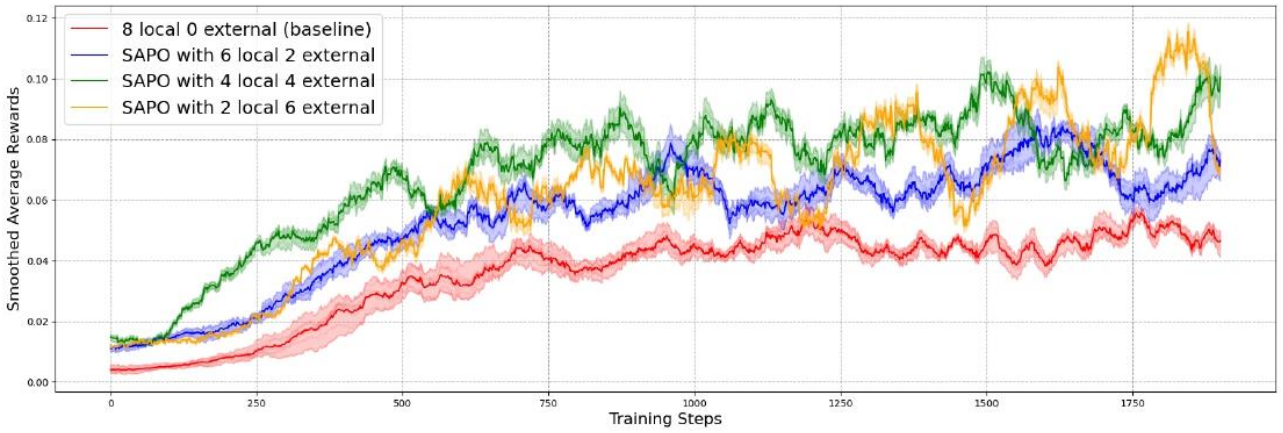
    `// Assemble training set`

    $\mathcal{T}^n \leftarrow \text{SampleSelf}(\{\mathcal{C}^n(q) \mid q \in \mathcal{B}^n\}, I^n)$
    $\hookrightarrow \text{SampleExternal}(\cup_{m \neq n}\{\mathcal{C}^m(q) \mid q \in \mathcal{S}^m\}, J^n)$

    `// Policy update`

    $\pi^n \leftarrow \text{PolicyGradient}(\pi^n, \rho^n, \mathcal{T}^n)$





Problem Definition

- **배경**: 추론을 위한 LLM 학습을 위한 중앙집중형 HW와 동기화 SW를 통한 분산 학습의 병목 현상
- **핵심 문제**: LLM 및 Foundation Model의 Scalability에 따른 학습 비효율 문제
- **세부 이슈**: 높은 **지연 시간(Latency), 메모리, 비용, 병목 현상** 발생으로 학습 효율 저하

Methodology : SAPO (Swarm Sampling Policy Optimization)

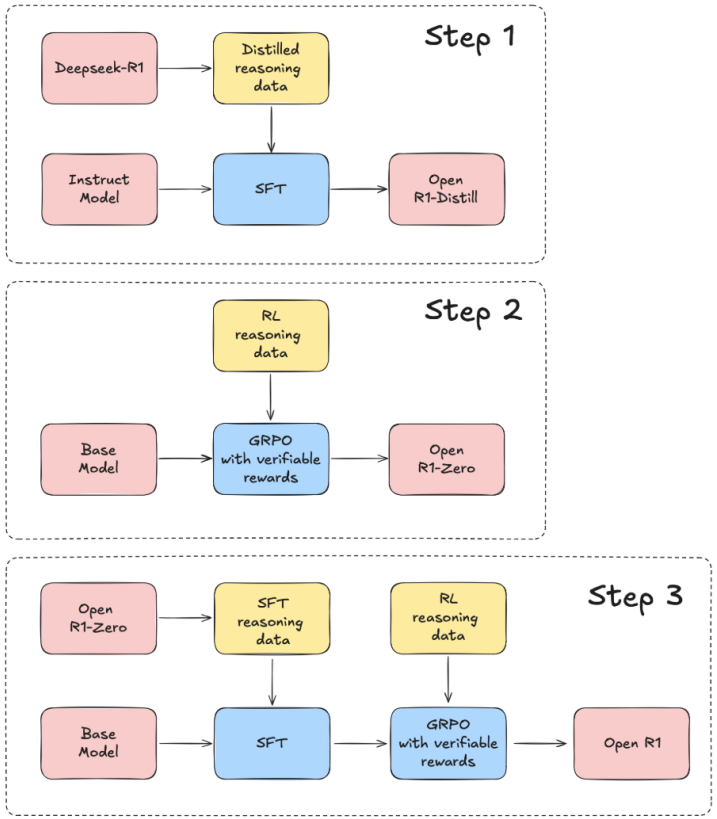
- **핵심 아이디어**: 기존 학습 방법 대비 가벼운 **Rollout 기반 RL**을 통한 Experience Sharing 효율성 증대
- **작동 방식**: 각 Node(LLM)는 공유된 경험을 바탕으로 **독립적, 비동기적**으로 자신의 Policy Update

Experimental Results

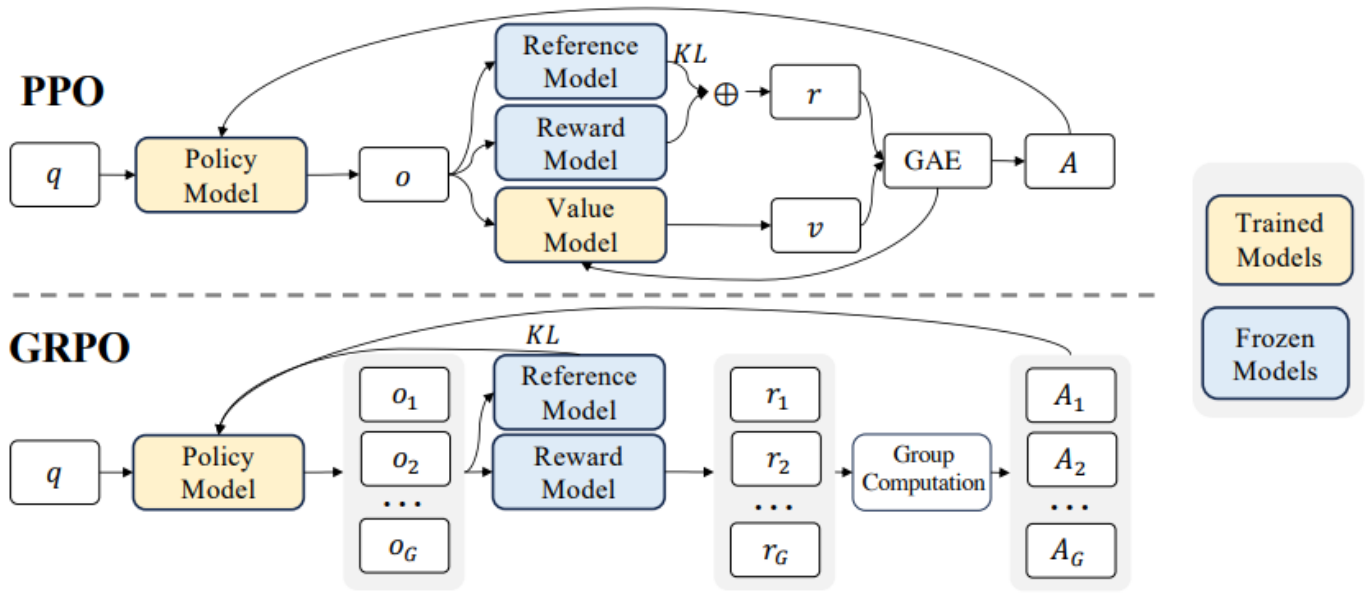
- **정량적 성과**: 단독 학습 대비 **최대 94% 누적 보상(Cumulative Reward) 증가**
- **통계적 검증**: 초기 학습 이후, **통계적으로 유의미한($p < 0.05$) 성능 격차** 확인
- **결론**: SAPO는 분산 환경에서 **집단 학습을 효과적으로 가속**함을 입증

Table of Contents

Related Work (RL)



DeepSeek-R1(25.01)



DeepSeekMath(24.02)

- LM의 fine-tuning 진행에 있어 RL은 매우 중요한 기법
- RLHF(RL from Human Feedback), RLVR (RL from verifiable Rewards)
- Policy : PPO(Proximal Policy Optimization), GRPO(Group Relative Policy Optimization), DAPO, VAPO etc

Related Work (Multi-Agent Methods)

Multi-Agent 방법론은 여러 모델이 상호작용하며 문제를 해결하는 방식

- **Debate:** 여러 LM이 하나의 질문에 대해 독립적으로 답변을 생성한 뒤, 서로 대화를 통해 답변을 개선하고 정제해 나가는 방식. 최종 결과물은 투표나 별도의 검증 모델(verifier)을 통해 선택.

- **Specialization / Role-Playing:** 각 Agent에게 생성, 검증, 개선 등 명확한 역할을 부여하는 방식.

- **Self-Improvement:** 모델이 스스로 데이터를 생성하고 평가하는 self-play를 통해 반복적으로 성능을 개선하는 방식.

Motivation

최근 동향

언어 모델(LM)의 추론 능력을 향상시키기 위해 최근 **강화학습을 활용하는 효율적 연구** 제안됨
하지만 기존의 RL 확장 방식은 **대규모의 중앙집중형 인프라에 의존**한 한계에 직면

Limitation

- 대규모 GPU 클러스터 구축 및 운영에 막대한 재정적 비용이 발생
- 추론 규모 확장을 위한 병렬화 과정에서 병목현상 발생
- policy weights를 지속적으로 동기화하는 과정에서 통신 병목이 발생
- 지연 시간(latency), 메모리, 신뢰성 등 해결하기 어려운 기술적 과제들이 수반
- 고사양의 동질적인(homogeneous) 하드웨어를 요구

근본적 질문 :

“중앙 서버나 가중치 동기화 없이, 서로 다른 사양의 분산된 컴퓨팅 네트워크 환경에서
효율적으로 협력하여 언어 모델을 훈련할 수 있는 새로운 RL 프레임워크를 구축할 수 있는가?"

Methodology

Decentralized Network

Heterogeneous Nodes

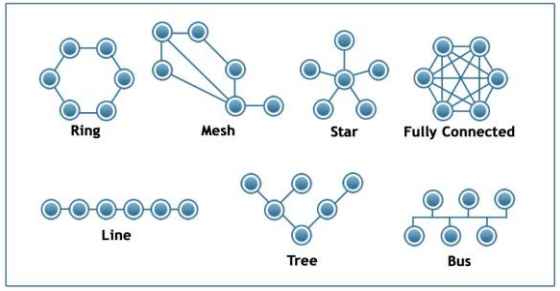
Collective Experience Sharing

Swarm

SAPO (Swarm sAmpling Policy Optimization)

"Aha Moment" Propagation

Methodology : Swarm



Swarm 정의: Swarm은 N개의 Nodes로 구성된 분산 네트워크(Decentralized Network)로 중앙 서버나 컨트롤러 없이 각 Node가 자율적으로 작동하며 서로 통신

- 특징 1 : **Asynchronous**: Node들은 서로를 기다릴 필요 없이 독립적인 속도로 작동.
- 특징 2 : **Heterogeneous**: 각 Node는 서로 다른 모델, 하드웨어, 네트워크 지연 시간을 가질 수 있음. 모델이나 하드웨어의 동질성에 대한 가정이 전혀 필요 없음

Node : Swarm에 참여하는 개별 컴퓨터, 서버이지만 여기서는 LM 모델을 돌릴 수 있는 HW, SW

각 Node 구성 요소:

- Model & Policy : Node n은 모델 (e.g. LM)을 보유하고 있으며, 각 모델별 Policy π^n 활용
- Dataset $D^n = \{(q, y_q) \mid q \in Q^n\}$: Node n의 Question & Ground-Truth ($q \in Q^n$, $y \in Y^n$)쌍으로 구성
  - 진행 실험 : Question과 Ground-Truth는 ReasoningGYM의 Rule-based Algorithm에 의해 생성
- Rollout $\mathcal{R}^n(q) := \{a_1^n(q), \ldots, a_{L^n}^n(q)\}$, : Node n는 주어진 문제(q)에 대해 Model이 생성한 여러 답변
- Metadata M^n: D^n의 Metadata M^n 는 Dataset 내 각 task가 어떻게 검증될 수 있는지에 대한 정보를 명시.
  - 진행 실험 : ReasoningGYM의 Rule-based Verifier에 대한 소개. (후속 연구 : LLM-as-Judge)

Methodology : SAPO (Swarm sAmpling Policy Optimization)

- 단계 1. 각 Training Round t에서 각 Node n은 자신의 질문 풀 Q^n에서 질문 Batch를 Subsampling 하고, 이에 답변하는 Rollout 생성함
  - 현재 실험 : Reasoning GYM 9개 도메인 중 모델이 내부적으로 활용한 Rollout 개수만큼의 도메인을 선택하여 도메인별 질문 하나씩 생성하여 B를 구성

- 단계 2. Task-Rollout 다른 Node들과 공유(Shared) or 샘플링(Sampled)
  - 2-1 Shared : 공유되는 정보 (question, answers, rollout, metadata)

$$C^n(q) := (q, y_q, \mathcal{R}^n(q), \mathcal{M}^n) \text{ for } q \in \mathcal{S}^n \subseteq \mathcal{B}^n.$$

- 단계 3. Node별 Training set 구성

$$\mathcal{T}^n = \underbrace{\left\{ I^n\text{-many samples from } \bigcup_{q \in \mathcal{B}^n} C^n(q) \right\}}_{\text{self-rollouts}} \cup \underbrace{\left\{ J^n\text{-many samples from } \bigcup_{m \neq n,\ q \in \mathcal{S}^m} C^m(q) \right\}}_{\text{external rollouts}}.$$

- 단계 4. Policy update
  - Policy = Policy Gradient(Policy, Reward Model, Training set)
  - 현재 실험 : Policy Optimization(GRPO), Reward Model(RLVR, ReasoningGYM Verifier)

Methodology : SAPO (Swarm sAmpling Policy Optimization)

Algorithm 1: SAPO

Input: For each $n \in [N]$: dataset $\mathcal{D}^n$, metadata $\mathcal{M}^n$, policy π^n, reward model ρ^n, policy update algorithm, number of local samples I^n, number of external samples J^n

Output: Updated policy parameters

for *each round* t **do**

  for *each node* n **do** `// Can be fully decentralized and run in parallel`

    `// Sample questions`

    $\mathcal{B}^n \leftarrow \text{SampleBatch}(\mathcal{Q}^n)$

    `// Generate rollouts`

    for *each* $q \in \mathcal{B}^n$ **do**

      $\mathcal{R}^n(q) \leftarrow \{a_1^n(q), \ldots, a_{L^n}^n(q)\}$

    `// Share rollouts and associated data`

    $\mathcal{S}^n \leftarrow \text{SelectSubset}(\mathcal{B}^n)$

    $\text{Communicate}(\{\mathcal{C}^n(q) \mid q \in \mathcal{S}^n\})$

    `// Assemble training set`

    $\mathcal{T}^n \leftarrow \text{SampleSelf}(\{\mathcal{C}^n(q) \mid q \in \mathcal{B}^n\}, I^n)$

      $\hookrightarrow \text{SampleExternal}(\cup_{m \neq n}\{\mathcal{C}^m(q) \mid q \in \mathcal{S}^m\}, J^n)$

    `// Policy update`

    $\pi^n \leftarrow \text{PolicyGradient}(\pi^n, \rho^n, \mathcal{T}^n)$

Controlled Experiment Setup : Dataset

ReasoningGYM Dataset

- Algebra, Logic, Graph Reasoning과 같은 도메인에서 On-Demand로 다양한 난이도의 문제를 생성하는 Dynamic Dataset.

  - `base_conversion`: converting numbers between different bases;
  - `basic_arithmetic`: performing elementary arithmetic operations;
  - `arc_1d`: abstract reasoning over one-dimensional sequences (a simplified version of the ARC benchmark);
  - `bf`: tasks involving Brainf*ck programs or similar algorithmic reasoning;
  - `propositional_logic`: solving propositional logic questions;
  - `fraction_simplification`: simplifying fractions as much as possible;
  - `decimal_arithmetic`: enforcing proper operator precedence during arithmetic with decimal contraints;
  - `calendar_arithmetic`: puzzle solving on word problems involving calendar dates;
  - `binary_matrix`: abstract reasoning on binary square matrices.

- 가장 큰 특징 : 생성된 모든 문제에 대해 **프로그램적으로 정답을 검증** 가능
- Baseline 실험 설정:
  - 1) 각 Training Round에서 Agent는 위 9개 목록 중 8개의 Specialities를 임의의 Sampling
  - 2) 각 전문 분야별(8개)로 하나의 ReasoningGYM 질문 제공
  - 3) 각 Agent는 질문당 8개의 completions을 생성 (한 Round에 8개 질문 * 8개 답변 = 총 64개 답변)
  - 4) 8개의 Rollout 생성 (= # of Specialities)

$$\mathcal{R}^n(q) := \{a_1^n(q), \ldots, a_{L^n}^n(q)\},$$

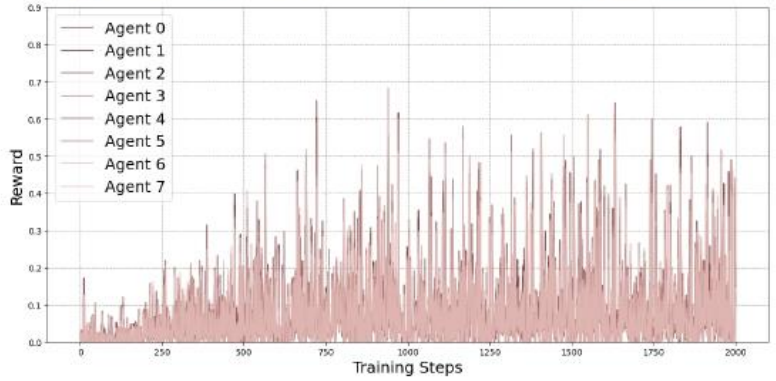
Controlled Experiment Setup : Policy, Reward Model

- **Swarm 구성:** 8개의 Qwen2.5 0.5B 모델 (동일 모델로 구성하여 공유 효과만 측정).
- **Dataset: ReasoningGYM** (수학, 논리 등 9개 도메인의 검증 가능한 추론 문제).
- **Policy Optimization : GRPO** (Group Relative Policy Optimization) 사용.
- **Reward Model :** ReasoningGYM의 **Rule-based Verifier (RLVR 패러다임)**.

Experiments Results

(a) Baseline case, i.e. 8 local / 0 external rollouts.

(b) 6 local / 2 external rollouts.



(c) 4 local / 4 external rollouts.

(d) 2 local / 6 external rollouts.

- **Result 1 :** Experience Sharing(경험 공유)는 학습 성능을 향상시킴.
- **Quantitative Analysis:**
  - **4 Local / 4 External** 구성이 Baseline 대비 **누적 보상 94% 향상**으로 최고의 성능 기록.
  - Rollout **(a) 8/0 (561.79)**, (b) 6/2 (854.43), **(c) 4/4 (1093.31)**, **(d) 2/6 (945.87)**

Experiments Results

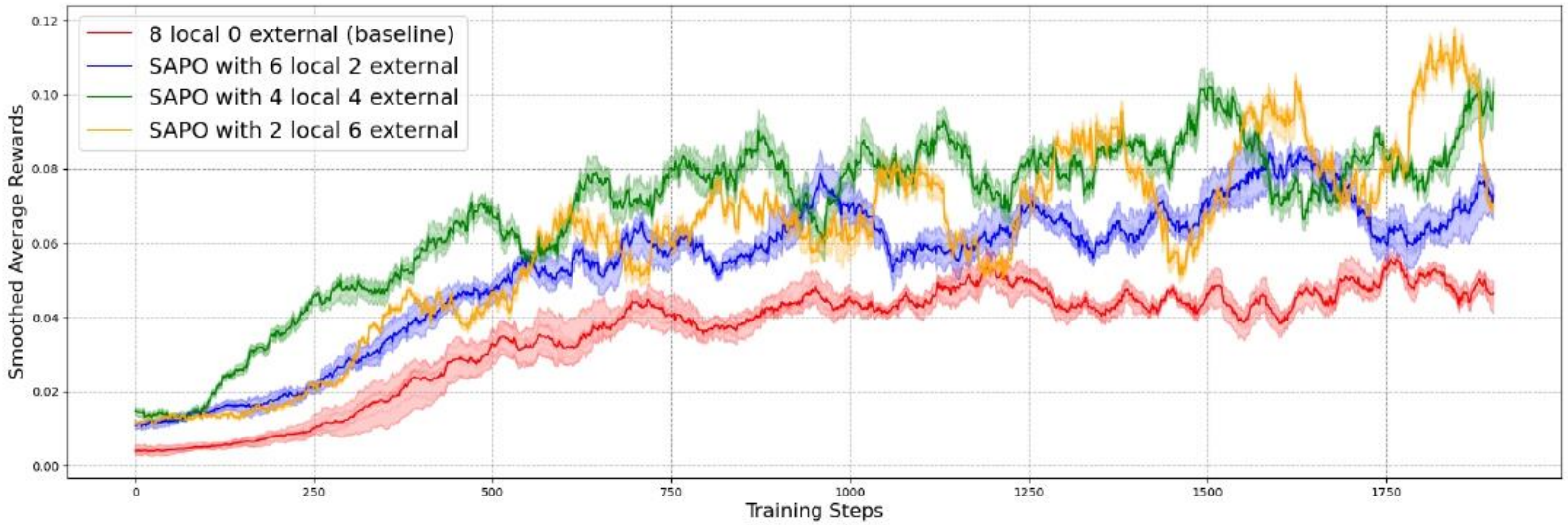


Figure 2 Average agent rewards for each configuration across training, smoothed with a moving average (window size 100). The 4 local / 4 external configuration consistently outperforms the baseline and, in nearly all rounds, also exceeds the 6 local / 2 external configuration in expected average reward. The 4 local / 4 external configuration also surpasses the 2 local / 6 external setup for most rounds, though the difference is smaller compared to the other cases.

- **"Aha Moment"의 전파:** 개별 Agent의 성공적인 발견(높은 보상의 Rollout)이 Swarm 전체로 빠르게 전파되어 집단 지성(Collective Intelligence)을 통한 학습 가속화.
- **과도한 의존의 위험성: External Rollout** 의존도가 높은 2/6 구성에서 **학습 불안정성 및 진동(Oscillation)** 현상 관찰.
- **추정 원인:**
  - 고성능 Agent가 저성능 Agent 의 Rollout에 의해 학습이 저해됨.
  - Swarm 전체의 Rollout 생성량 대비 소비량이 많아져 Shared Pool의 질적 저하 발생.

Training in a large Swarm: Insights from an Open-Source Demo

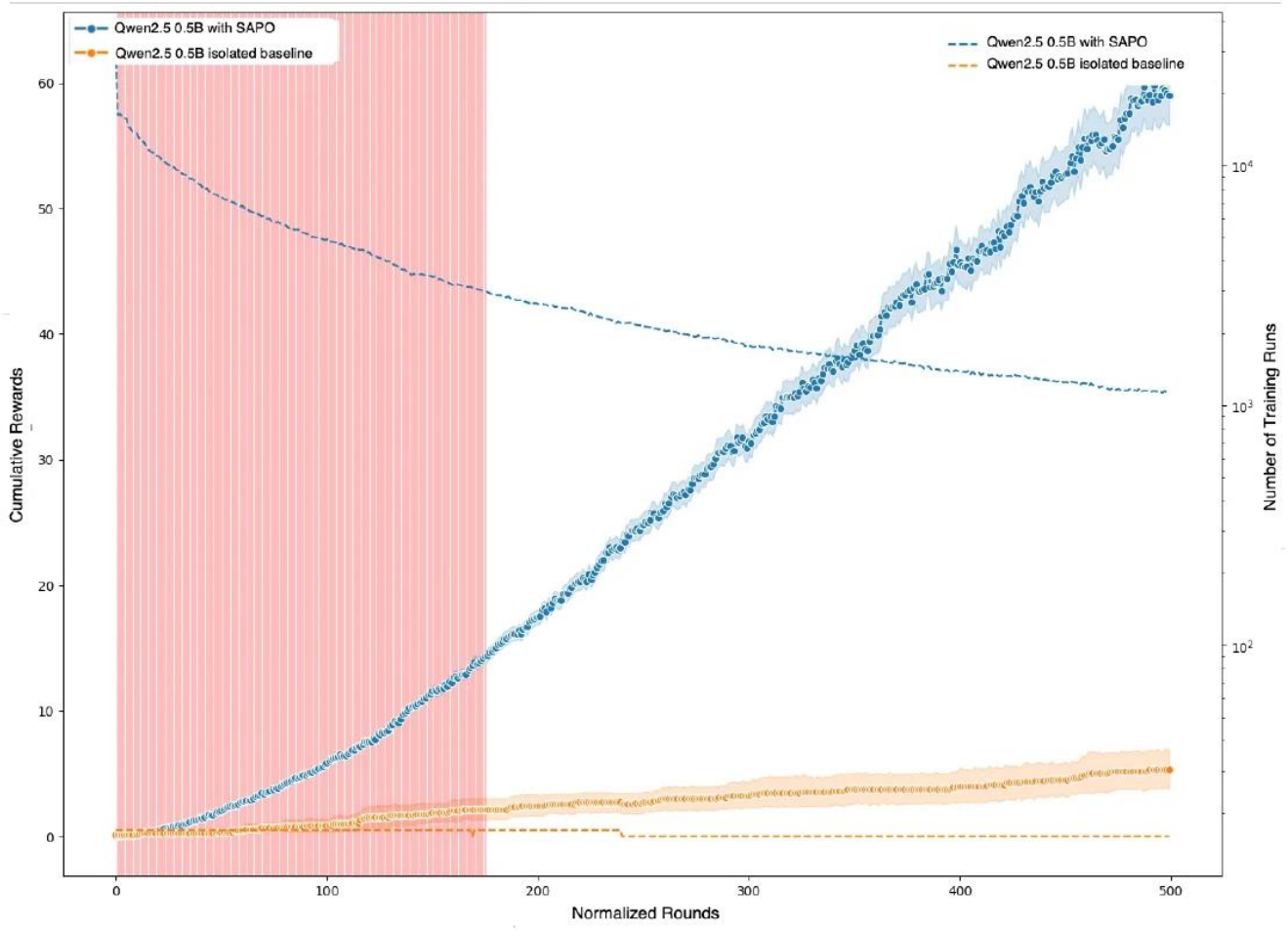


Figure 3 Shown in red are the regions where the adjusted p-value is greater than 0.05. After a certain number of rounds, in this case approximately 175, the performance per round of the models in the swarm significantly exceeds that of the model trained in isolation.

수천 명의 커뮤니티 멤버가 참여한 대규모 분산 환경에서 이기종 네트워크 (다양한 모델, 하드웨어) 속 데이터 수집.

Result 1 (Qwen2.5 0.5B): Swarm 참여가 독립 훈련 대비 유의미한 성능 향상을 보임.
특히, 175 Normalized Rounds 기점으로 확연히 상승(?)

Result 2 (Qwen3 0.6B):
Swarm 참여 여부에 따른 성능 차이가 미미.

Hypothesis: 고성능 모델은 단순한 Random Sampling 방식의 External Rollout에서 유의미한 학습 신호를 얻기 어려웠을 가능성.
> **더 정교한 Sampling/Filtering 전략의 필요성** 시사.