On the Rollout-Training Mismatch In Modern RL Systems

Feng Yao

UCSD

August 29, 2025 – Presented at QingkeAI

Efficient RL systems are rising

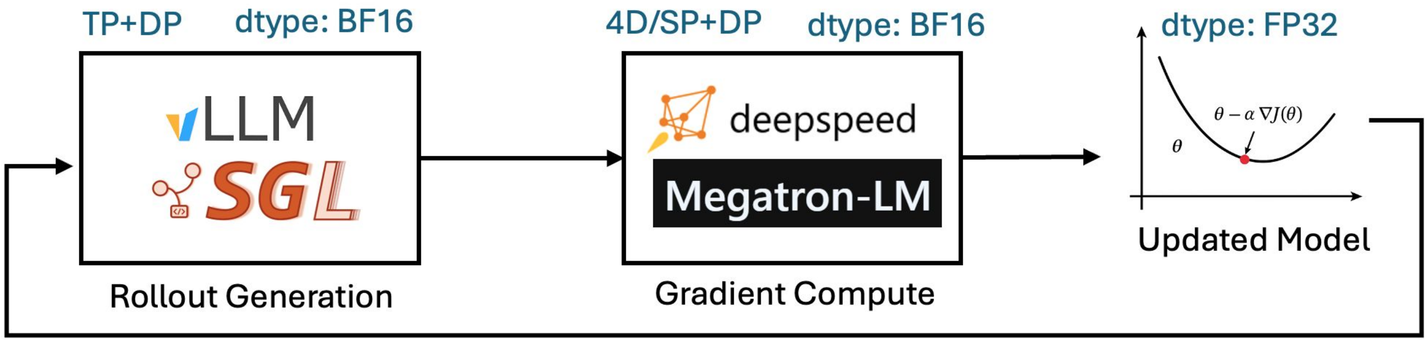
- VeRL/OpenRLHF adopts hybrid engines

Efficient RL systems are rising

- VeRL/OpenRLHF adopts hybrid engines
    - **Rollout:** Advanced LLM **inference engines** (vLLM, SGLang)
    - **Training:** Modern LLM **training backends** (FSDP, Megatron)

Efficient RL systems are rising

- VeRL/OpenRLHF adopts hybrid engines
  - **Rollout:** Advanced LLM **inference engines** (vLLM, SGLang)
  - **Training:** Modern LLM **training backends** (FSDP, Megatron)



It also brings an issue…

- **Rollout-Training Mismatch**

It also brings an issue…

- **Rollout-Training Mismatch**
  - Expected

$$\theta \leftarrow \theta + \mu \cdot \mathbb{E}_{\underbrace{a \sim \pi(\theta)}_{rollout}}[R(a) \cdot \underbrace{\nabla_\theta \log \pi(a, \theta)}_{training}]$$

It also brings an issue…

- **Rollout-Training Mismatch**
    - Expected

$$\theta \leftarrow \theta + \mu \cdot \mathbb{E}_{\underbrace{a \sim \pi(\theta)}_{rollout}}[R(a) \cdot \underbrace{\nabla_\theta \log \pi(a, \theta)}_{training}]$$

    - Implementation: Rollout engine (vLLM) + Training backends (FSDP)

$$\theta \leftarrow \theta + \mu \cdot \mathbb{E}_{a \sim \pi_{\mathrm{vllm}}(\theta)}[R(a) \cdot \nabla_\theta \log \pi_{\mathrm{fsdp}}(a, \theta)]$$

It also brings an issue…

- **Rollout-Training Mismatch**
  - Expected

$$\theta \leftarrow \theta + \mu \cdot \mathbb{E}_{\underbrace{a \sim \pi(\theta)}_{rollout}}[R(a) \cdot \underbrace{\nabla_\theta \log \pi(a, \theta)}_{training}]$$

  - Implementation: Rollout engine (vLLM) + Training backends (FSDP)

$$\theta \leftarrow \theta + \mu \cdot \mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta)}[R(a) \cdot \nabla_\theta \log \pi_{\text{fsdp}}(a, \theta)]$$

Mismatch!

It also brings an issue…

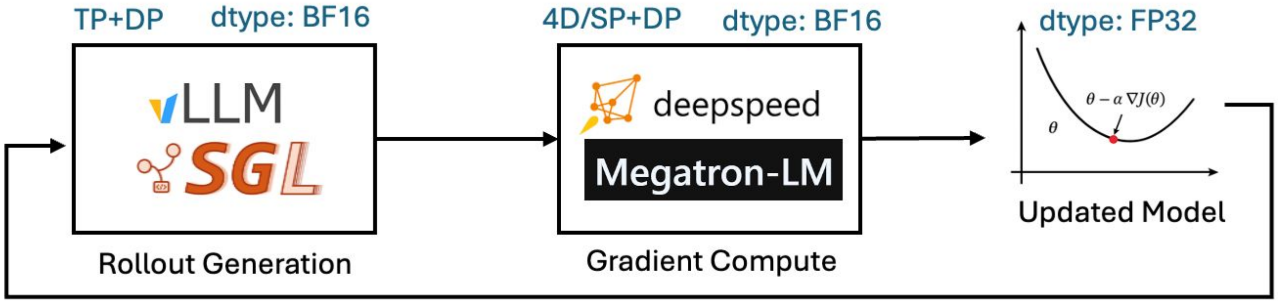
- **Rollout-Training Mismatch**
  - For the **same** rollout & model parameter

$$\theta \leftarrow \theta + \mu \cdot \mathbb{E}_{a \sim \pi_{\text{vllm}}(\theta)}[R(a) \cdot \nabla_\theta \log \pi_{\text{fsdp}}(a, \theta)]$$

It also brings an issue…

- **Rollout-Training Mismatch**
  - For the **same** rollout & model parameter

$$\theta \leftarrow \theta + \mu \cdot \mathbb{E}_{a \sim \pi_{\text{vllm}}(\theta)}[R(a) \cdot \nabla_\theta \log \pi_{\text{fsdp}}(a, \theta)]$$

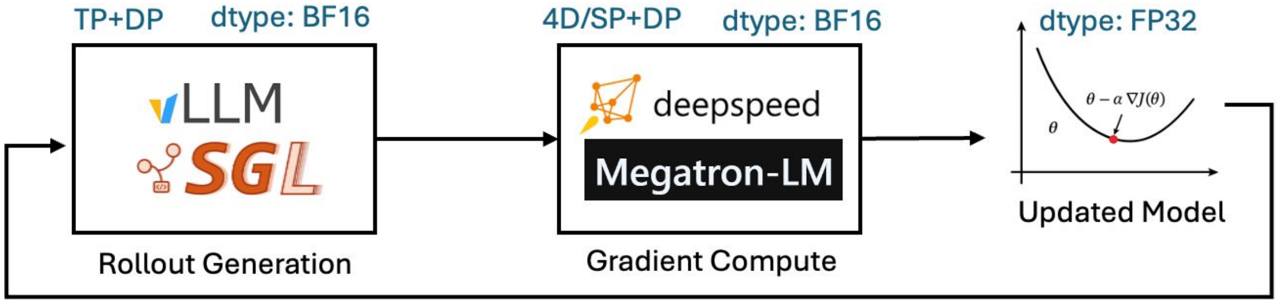


It also brings an issue…

- **Rollout-Training Mismatch**
  - For the **same** rollout & model parameter

$$\theta \leftarrow \theta + \mu \cdot \mathbb{E}_{a \sim \pi_{\text{vllm}}(\theta)}[R(a) \cdot \nabla_\theta \log \pi_{\text{fsdp}}(a, \theta)]$$

$$p^{vllm} - p^{fsdp}$$

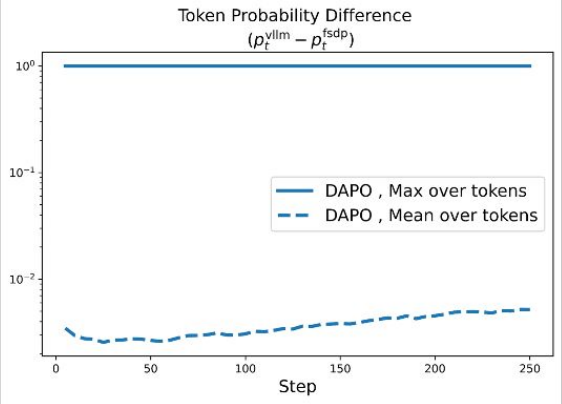


It also brings an issue…

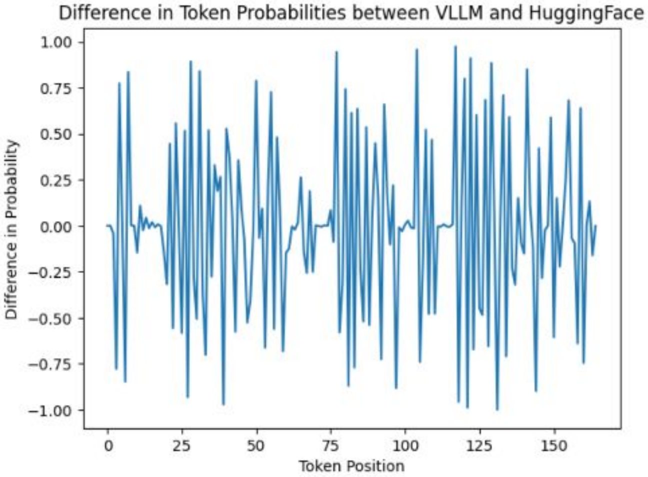
- **Rollout-Training Mismatch**
  - $p^{vllm} - p^{fsdp}$

It also brings an issue…

- **Rollout-Training Mismatch**
  - $p^{vllm} - p^{fsdp}$



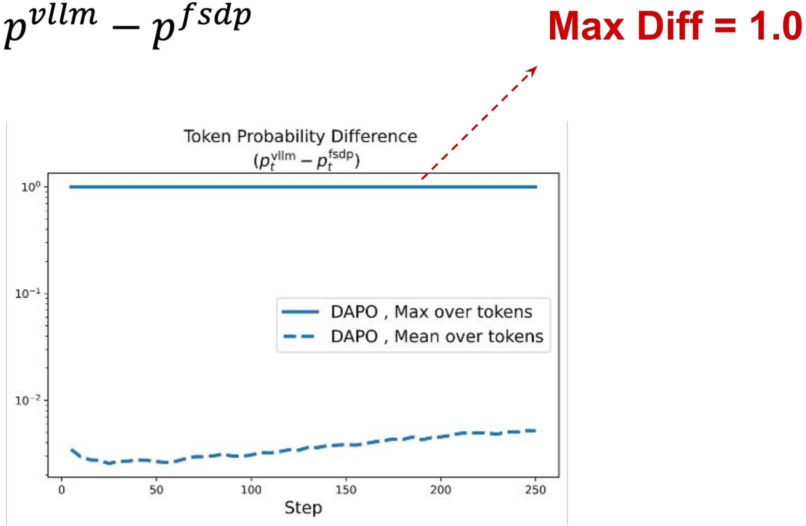
DAPO Qwen2.5-32B



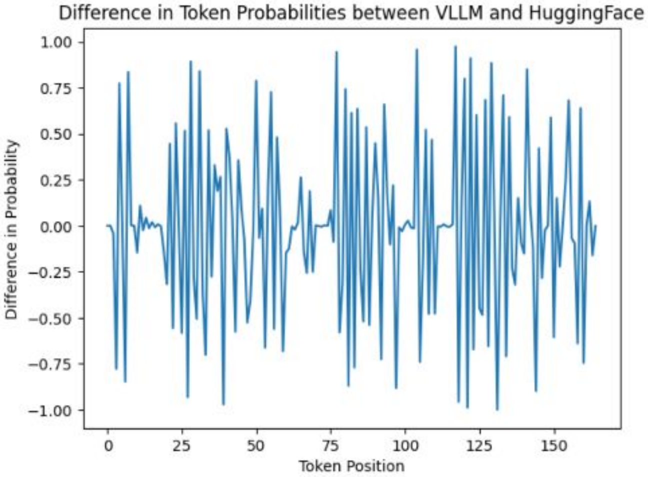
DS-Qwen2.5-1.5B

It also brings an issue…

- **Rollout-Training Mismatch**
  - $p^{vllm} - p^{fsdp}$



DAPO Qwen2.5-32B

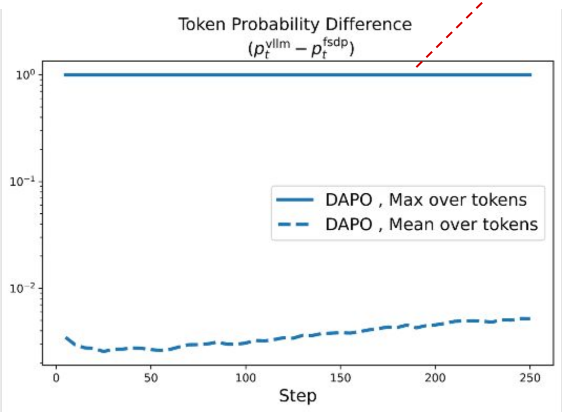


DS-Qwen2.5-1.5B

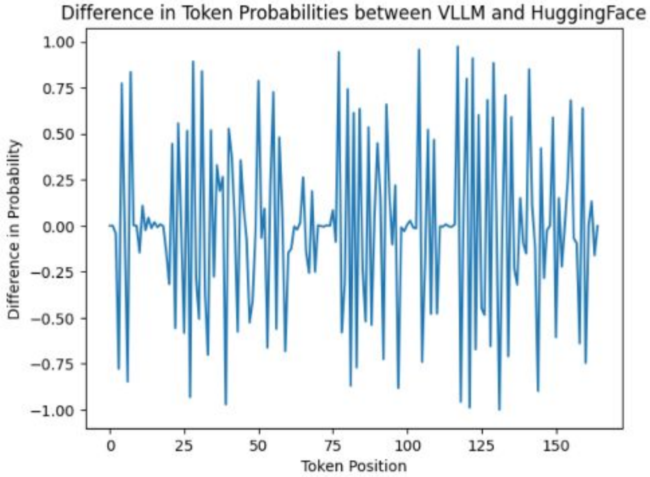
It also brings an issue…

- **Rollout-Training Mismatch**
  - $p^{vllm} - p^{fsdp}$

Max Diff = 1.0



DAPO Qwen2.5-32B



DS-Qwen2.5-1.5B

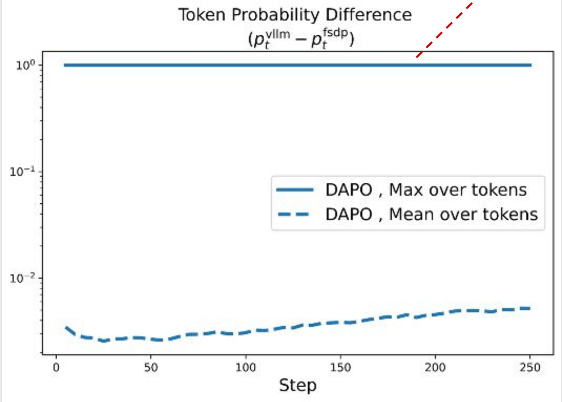
$$p^{vllm} = 1.0 \;\; \& \;\; p^{fsdp} = 0.0$$

It also brings an issue…

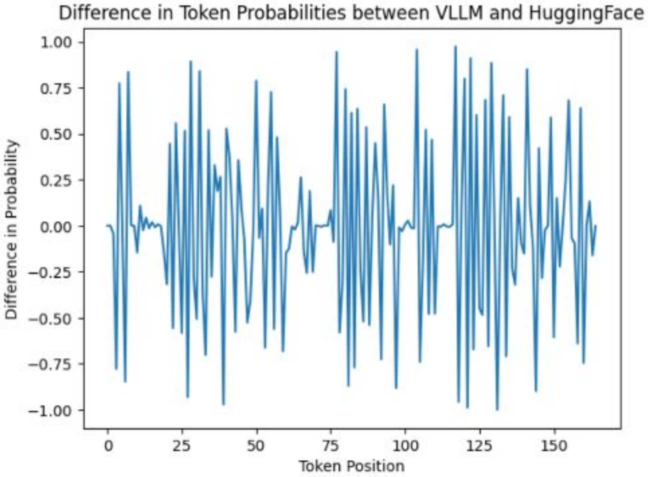
Implicitly makes RL "Off-Policy"!

- **Rollout-Training Mismatch**
  - $p^{vllm} - p^{fsdp}$

Max Diff = 1.0



DAPO Qwen2.5-32B



DS-Qwen2.5-1.5B

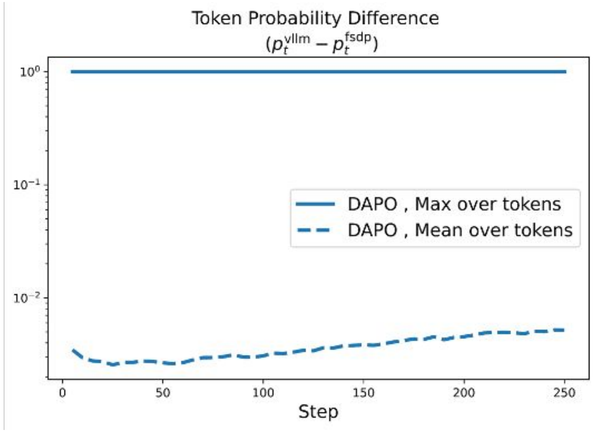
$$p^{vllm} = 1.0 \quad \& \quad p^{fsdp} = 0.0$$

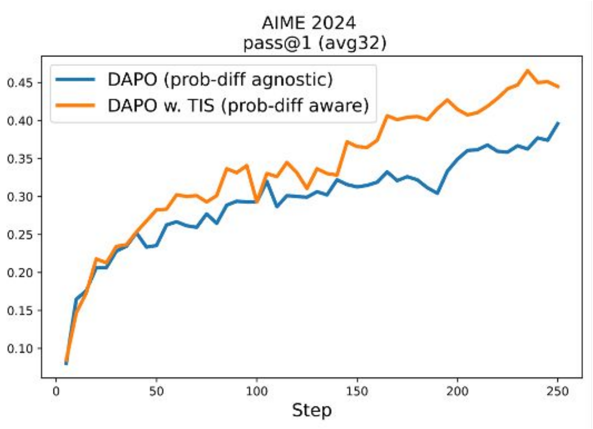
But it can be fixed effectively

- Using the classic ***Truncated Importance Sampling (TIS)*** technique

But it can be fixed effectively

- Using the classic ***Truncated Importance Sampling (TIS)*** technique
  - We show that **fix it with TIS** can improve training **effectiveness**





Harvesting the *Off-Policyness* via Quantization

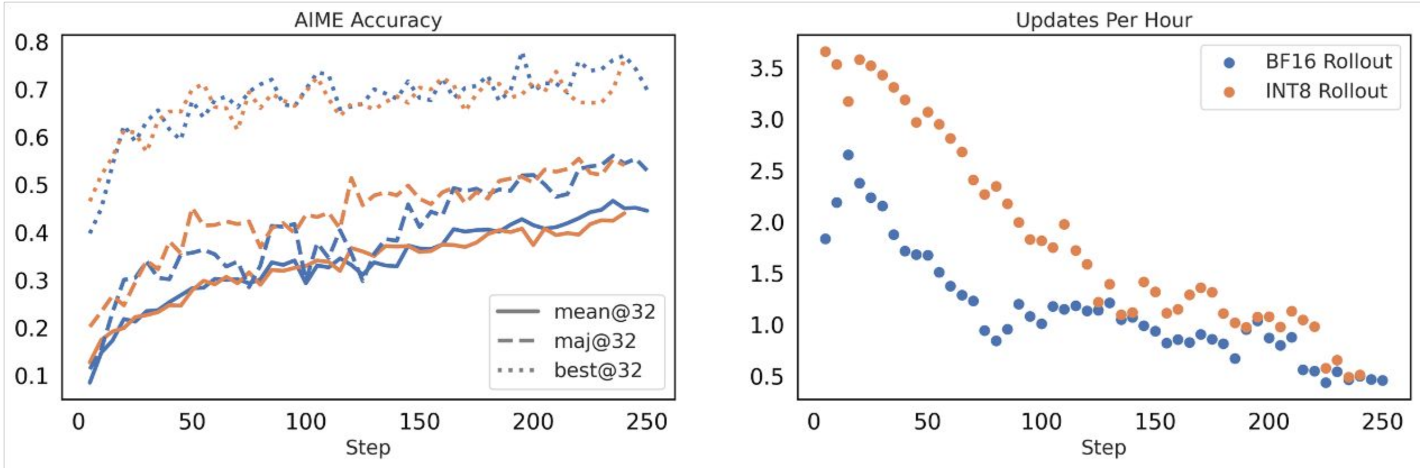
- Since TIS is able to handle the mismatch

Harvesting the *Off-Policyness* via Quantization

- Since TIS is able to handle the mismatch
  - Can we go even more “off-policy” and thus faster?

Harvesting the *Off-Policyness* via Quantization

- Since TIS is able to handle the mismatch
  - Can we go even more "off-policy" and thus faster?



Outline

- Why Rollout-Training Mismatch Occurs
- How to Fix the Off-Policy Issue It Brings
- Harvesting Rollout-Training Mismatch via Quantization
- Analyzing the Effectiveness of Different Fixes

Outline

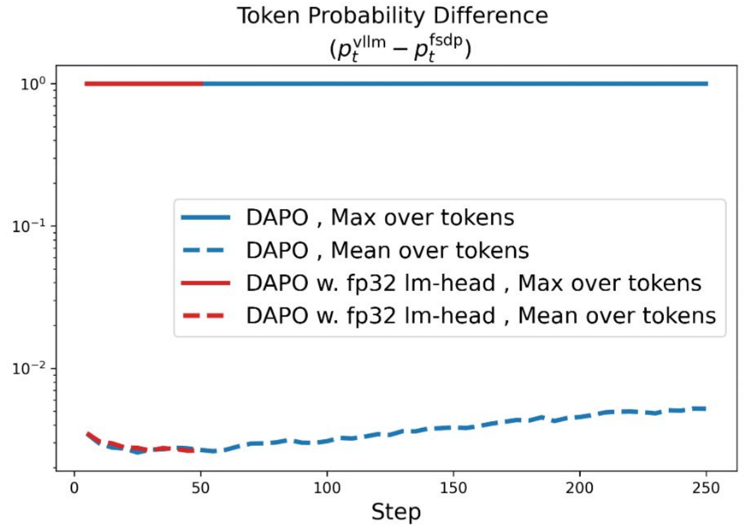
- **Why Rollout-Training Mismatch Occurs**
- **How to Fix the Off-Policy Issue It Brings**
- **Harvesting Rollout-Training Mismatch via Quantization**
- **Analyzing the Effectiveness of Different Fixes**

Why does Rollout-Training Mismatch occur?

- **Two common believes**

Why does Rollout-Training Mismatch occur?

- **Two common believes**
    - **Inaccessible true sampling probabilities**
        - Add additional gap
    - **Backend numerical differences**
        - Hard to fix

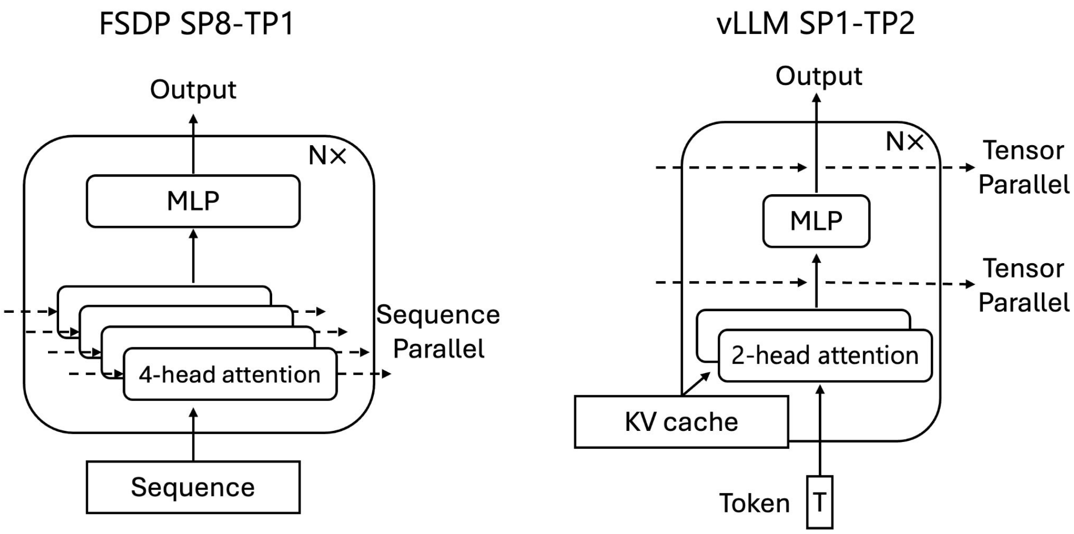


Why does Rollout-Training Mismatch occur?

- Hybrid Engine & Error Propagation
    - Different compute patterns via different backends & parallelism

Why does Rollout-Training Mismatch occur?

- **Hybrid Engine & Error Propagation**
  - **Different compute patterns via different backends & parallelism**



Outline

- Why Rollout-Training Mismatch Occurs
- How to Fix the Off-Policy Issue It Brings
- Harvesting Rollout-Training Mismatch via Quantization
- Analyzing the Effectiveness of Different Fixes

How to Fix the Off-Policy Issue It Brings

- **Trial 1 – Mitigate the system-level mismatch**
  - vLLM seems to be the root cause

How to Fix the Off-Policy Issue It Brings

- **Trial 1 – Mitigate the system-level mismatch**
  - vLLM seems to be the root cause → **Patch vLLM to**:

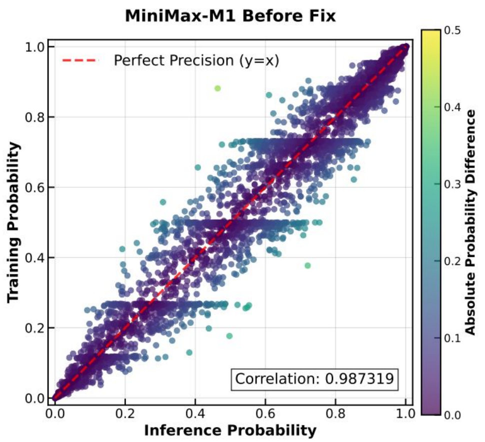
How to Fix the Off-Policy Issue It Brings

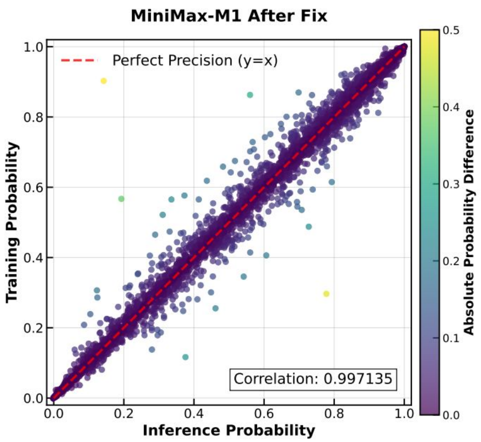
- **Trial 1 – Mitigate the system-level mismatch**
  - vLLM seems to be the root cause ⟶ **Patch vLLM to**:
    - Return the **actual sampling probabilities** for vLLM V1 engine
    - Improve the numerical precision by using **FP32 LM_Head**

How to Fix the Off-Policy Issue It Brings

- **Trial 1 – Mitigate the system-level mismatch**
  - vLLM seems to be the root cause ⟶ **Patch vLLM to**:
    - Return the **actual sampling probabilities** for vLLM V1 engine
    - Improve the numerical precision by using **FP32 LM_Head**

It helps, but …

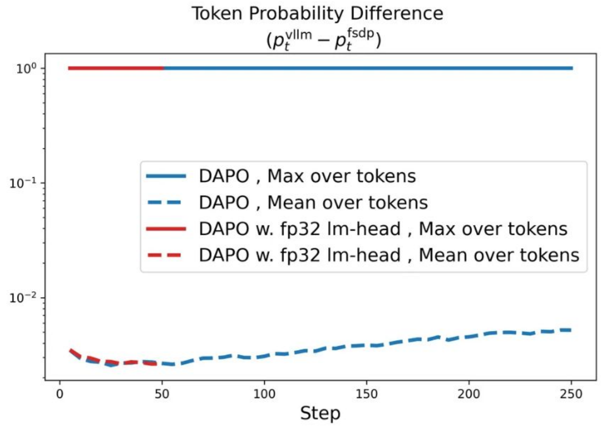


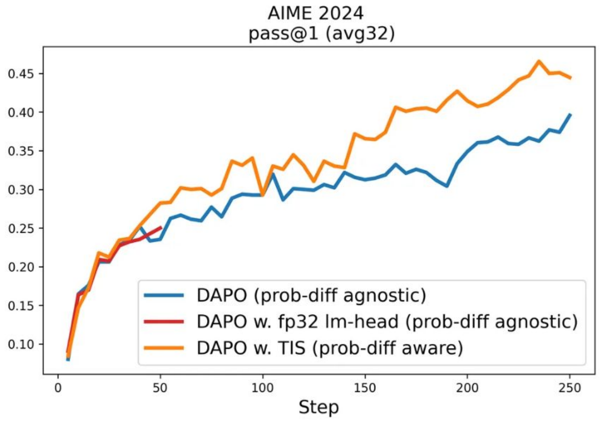


How to Fix the Off-Policy Issue It Brings

- **Trial 1 – Mitigate the system-level mismatch**
  - vLLM seems to be the root cause ⟶ **Patch vLLM to**:
    - Return the **actual sampling probabilities** for vLLM V1 engine
    - Improve the numerical precision by using **FP32 LM_Head**

It helps, but the gap still exists





How to Fix the Off-Policy Issue It Brings

- **Trial 2 – Apply algorithm-level fix**
  - Be aware of the mismatch $\longrightarrow$ **Importance sampling correction**:

How to Fix the Off-Policy Issue It Brings

- **Trial 2 – Apply algorithm-level fix**
  - Be aware of the mismatch $\longrightarrow$ **Importance sampling correction**:
    - Recall: Vanilla Importance Sampling

$$\mathbb{E}_{a\sim\pi_{\text{fsdp}}(\theta)}[R(a)] = \mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta)}\left[\underbrace{\frac{\pi_{\text{fsdp}}(a,\theta)}{\pi_{\text{vllm}}(a,\theta)}}_{\text{importance ratio}} \cdot R(a)\right]$$

How to Fix the Off-Policy Issue It Brings

- **Trial 2 – Apply algorithm-level fix**
  - Be aware of the mismatch $\longrightarrow$ **Importance sampling correction**:
    - Expected gradient

$$\mathbb{E}_{a\sim\pi_{\text{fsdp}}(\theta)}[R(a)\cdot\nabla_\theta\log\pi_{\text{fsdp}}(a,\theta)]$$

How to Fix the Off-Policy Issue It Brings

- **Trial 2 – Apply algorithm-level fix**
  - Be aware of the mismatch $\longrightarrow$ **Importance sampling correction**:
    - Expected gradient

$$\mathbb{E}_{a \sim \pi_{\text{fsdp}}(\theta)}[R(a) \cdot \nabla_\theta \log \pi_{\text{fsdp}}(a, \theta)]$$

    - But currently we have

$$\mathbb{E}_{a \sim \pi_{\text{vllm}}(\theta)}[R(a) \cdot \nabla_\theta \log \pi_{\text{fsdp}}(a, \theta)]$$

How to Fix the Off-Policy Issue It Brings

- **Trial 2 – Apply algorithm-level fix**
  - Be aware of the mismatch $\longrightarrow$ **Importance sampling correction**:
    - Expected gradient

$$\mathbb{E}_{a\sim\pi_{\text{fsdp}}(\theta)}[R(a)\cdot\nabla_\theta \log \pi_{\text{fsdp}}(a,\theta)]$$

    - But currently we have

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta)}[R(a)\cdot\nabla_\theta \log \pi_{\text{fsdp}}(a,\theta)]$$

    - So we should fix the gradient as:

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta)}\left[\frac{\pi_{\text{fsdp}}(a,\theta)}{\pi_{\text{vllm}}(a,\theta)}\cdot R(a)\cdot\nabla_\theta \log \pi_{\text{fsdp}}(a,\theta)\right]$$

How to Fix the Off-Policy Issue It Brings

- **Trial 2 – Apply algorithm-level fix**
  - Be aware of the mismatch $\longrightarrow$ **Importance sampling correction**:
    - Expected gradient

$$\mathbb{E}_{a\sim\pi_{\text{fsdp}}(\theta)}[R(a)\cdot\nabla_\theta \log \pi_{\text{fsdp}}(a,\theta)]$$

    - But currently we have

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta)}[R(a)\cdot\nabla_\theta \log \pi_{\text{fsdp}}(a,\theta)]$$

    - In practice, we use **Truncated Importance Sampling (TIS)**:

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta)}\Bigg[\underbrace{\min\left(\frac{\pi_{\text{fsdp}}(a,\theta)}{\pi_{\text{vllm}}(a,\theta)}, C\right)}_{\text{truncated importance ratio}}\cdot R(a)\cdot\nabla_\theta \log \pi_{\text{fsdp}}(a,\theta)\Bigg]$$

How to Fix the Off-Policy Issue It Brings

- **Extend to General Case**

How to Fix the Off-Policy Issue It Brings

- **Extend to General Case**
  - Expected Policy Gradient **(PPO)**

$$\mathbb{E}_{a \sim \pi_{\text{fsdp}}(\theta_{\text{old}})} \left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})} \hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon \right) \hat{A} \right) \right]$$

How to Fix the Off-Policy Issue It Brings

- **Extend to General Case**
  - Expected Policy Gradient **(PPO)**

$$\mathbb{E}_{a\sim\pi_{\text{fsdp}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

  - VeRL/OpenRLHF's Implementation **(recompute)**

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

How to Fix the Off-Policy Issue It Brings

- **Extend to General Case**
  - Expected Policy Gradient **(PPO)**

$$\mathbb{E}_{a\sim\pi_{\text{fsdp}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

  - VeRL/OpenRLHF's Implementation **(recompute)**

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

  - Truncated Importance Sampling **(TIS)**

$$\mathbb{E}_{\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\underbrace{\min\left(\frac{\pi_{\text{fsdp}}(a,\theta_{\text{old}})}{\pi_{\text{vllm}}(a,\theta_{\text{old}})}, C\right)}_{\text{truncated importance ratio}} \cdot \nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

Why Not Alternative Methods?

- **Variants of TIS**

Why Not Alternative Methods?

- **Variants of TIS**
  - PPO Importance Sampling **(PPO-IS)**

$$\mathbb{E}_{a\sim\pi_{\mathrm{vllm}}(\theta_{\mathrm{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\mathrm{fsdp}}(a,\ \theta)}{\pi_{\mathrm{vllm}}(a,\ \theta_{\mathrm{old}})}\hat{A}, \mathrm{clip}\left(\frac{\pi_{\mathrm{fsdp}}(a,\ \theta)}{\pi_{\mathrm{vllm}}(a,\ \theta_{\mathrm{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

Why Not Alternative Methods?

- **Variants of TIS**
  - PPO Importance Sampling **(PPO-IS)**

A commonly asked variant

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}\,\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

Why Not Alternative Methods?

- **Variants of TIS**
  - PPO Importance Sampling **(PPO-IS)**

A commonly asked variant

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\boxed{\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

Can break out of the trust region

Why Not Alternative Methods?

- **Variants of TIS**
  - PPO Importance Sampling **(PPO-IS)**

A commonly asked variant

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\boxed{\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

Can break out of the trust region

  - Vanilla Importance Sampling **(Vanilla-IS)**

$$\mathbb{E}_{\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\underbrace{\frac{\pi_{\text{fsdp}}(a,\theta_{\text{old}})}{\pi_{\text{vllm}}(a,\theta_{\text{old}})}}_{\text{importance ratio}} \cdot \nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

Why Not Alternative Methods?

- **Variants of TIS**
  - PPO Importance Sampling **(PPO-IS)**

A commonly asked variant

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\boxed{\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

Can break out of the trust region

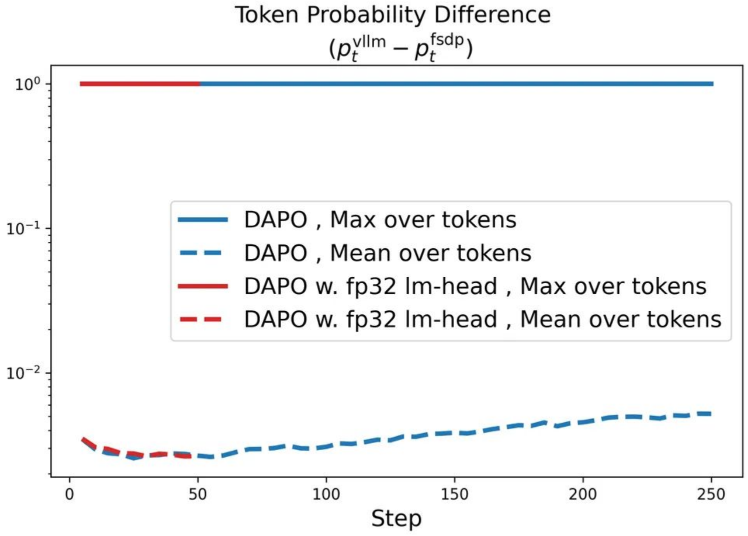
  - Vanilla Importance Sampling **(Vanilla-IS)**

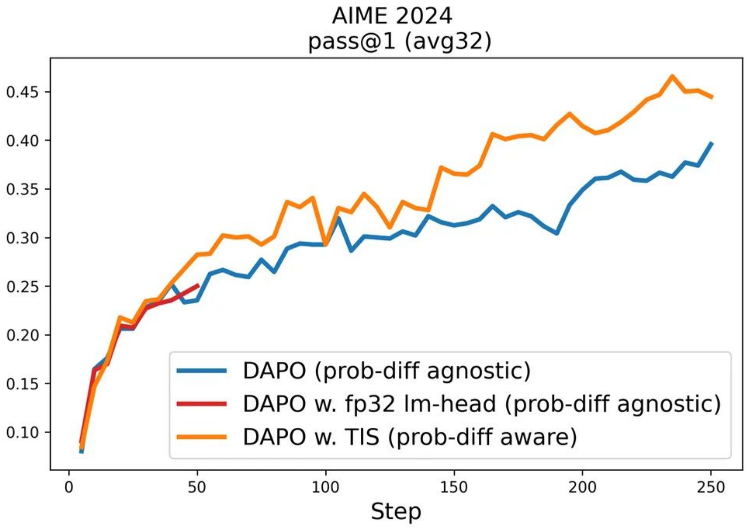
$$\mathbb{E}_{\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\boxed{\underbrace{\frac{\pi_{\text{fsdp}}(a, \theta_{\text{old}})}{\pi_{\text{vllm}}(a, \theta_{\text{old}})}}_{\text{importance ratio}}} \cdot \nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

Can be too large and makes training crash

How well can TIS fix it?

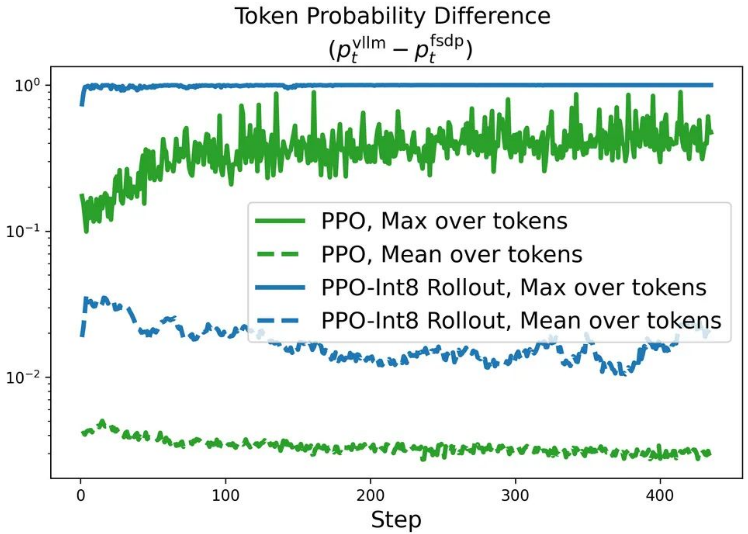
- **DAPO 32B Setting**

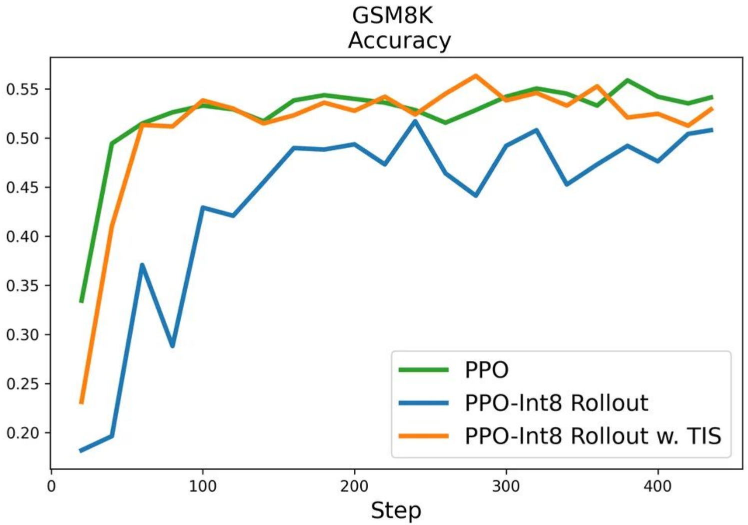




How well can TIS fix it?

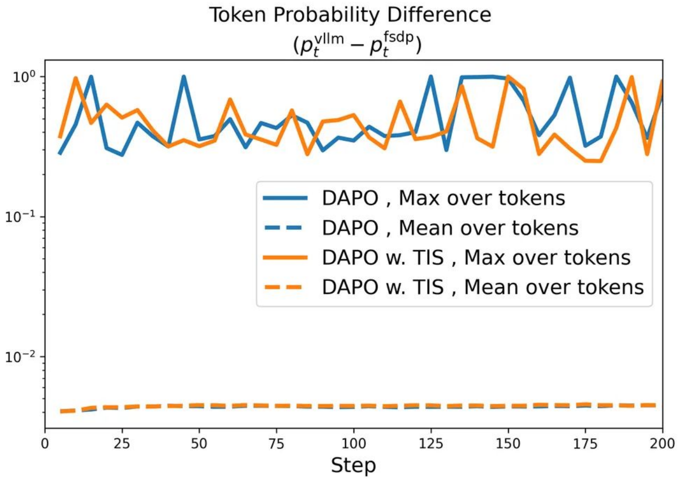
- **GSM8K 0.5B Setting**
  - **Normal RL: Max Diff is smaller (~0.4) than 1.0 (in DAPO-32B setting)**
  - **INT8 Rollout: Max Diff is larger (~1.0) than normal RL setting**

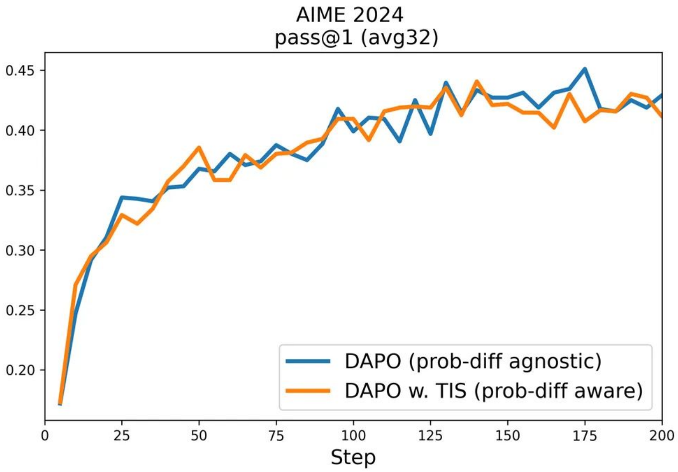




Does TIS always help?

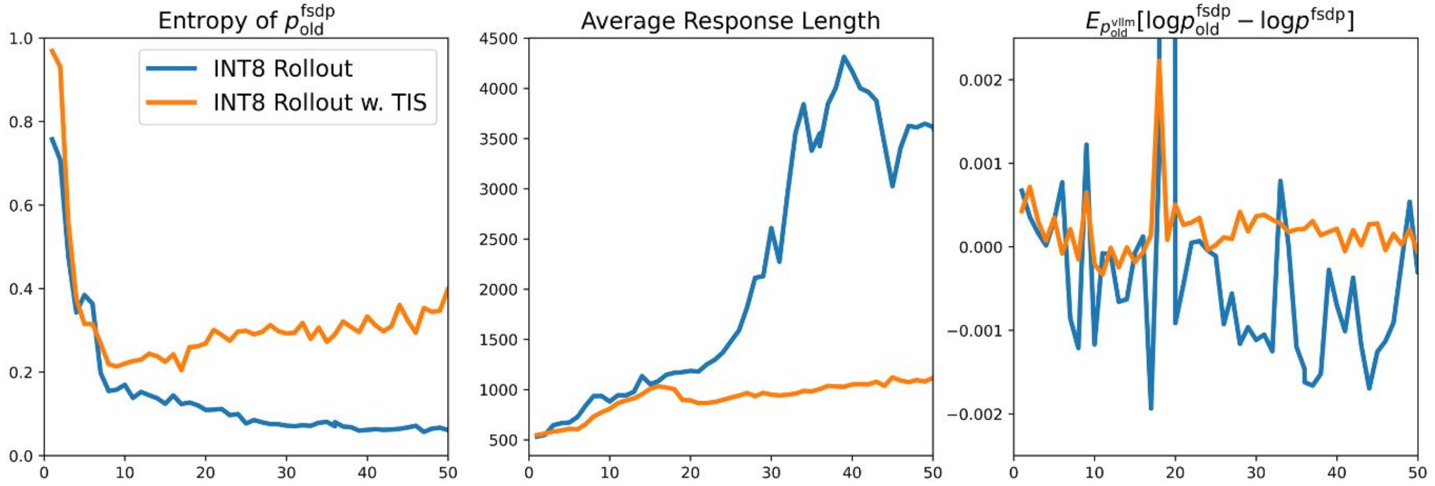
- **DAPO 1.5B Setting**
  - **In settings where prob diff is relatively small**
    - TIS does not always help, but doesn’t hurt





Does the Mismatch really matter?

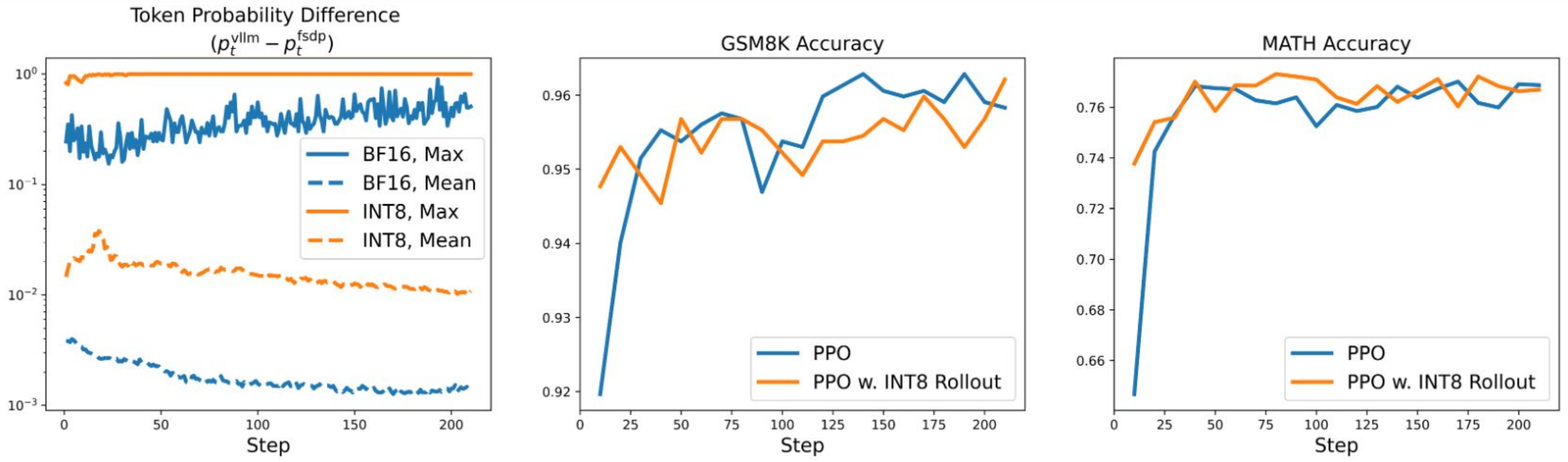
- **Unexpected training instability on challenging tasks**



DAPO Qwen2.5-32B

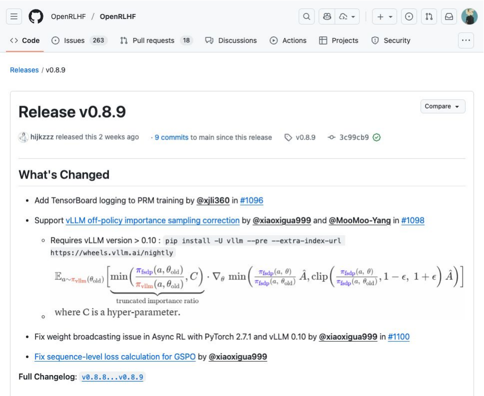
Does the Mismatch really matter?

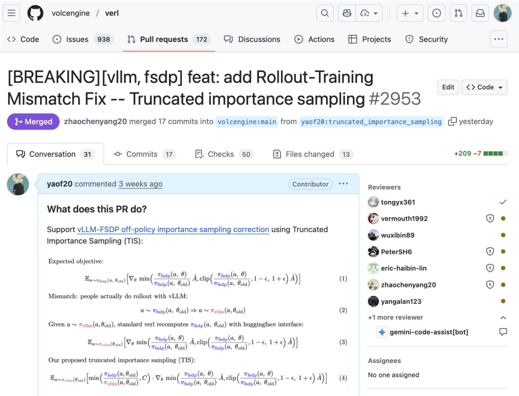
- **Possible negligible on simple tasks**

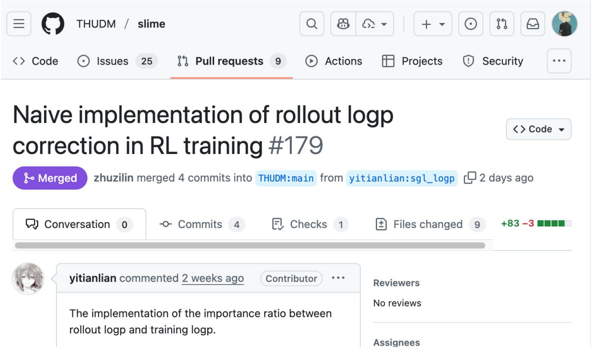


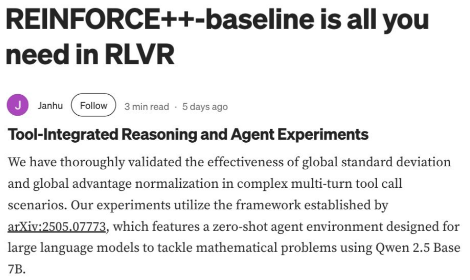
PPO GSM8K Qwen2.5-32B

Community Verification





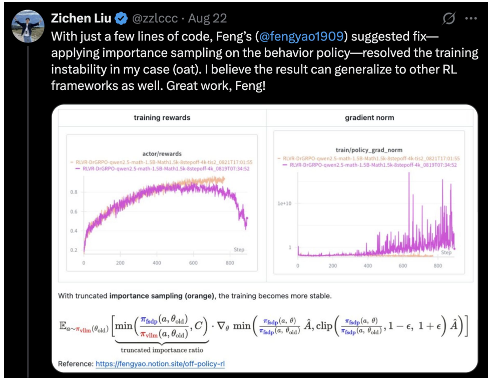


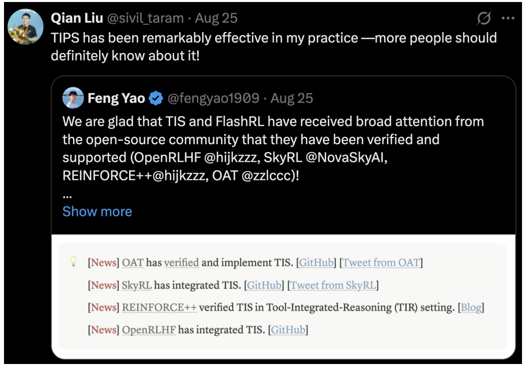


	300-step/avg@32/iternum-2(inference)	aime24	aime25	hmmt_feb_2025	hmmt_feb_2024	cimmc	avg
reinforce++	with-baseline	30.83333	27.1875	17.91667	18.95833	25.625	24.10416667
reinforce++	with-baseline,vllm-correction	31.5625	23.9583	20.625	20.1042	29.375	25.125
grpo		31.66667	21.875	16.97917	17.70833	24.6875	22.58333333
ppo		30.20833	21.66667	15	18.4375	23.95833	21.85416667

REINFORCE++-baseline achieves the best performance in the multi-turn tool call tasks.

The REINFORCE++ baseline can be combined with dynamic sampling, clip-higher, and truncated importance sampling (vLLM correction in the figure) to continuously improve performance.





Outline

- Why Rollout-Training Mismatch Occurs
- How to Fix the Off-Policy Issue It Brings
- Harvesting Rollout-Training Mismatch via Quantization
- Analyzing the Effectiveness of Different Fixes

Harvesting *Off-Policy* in Quantization

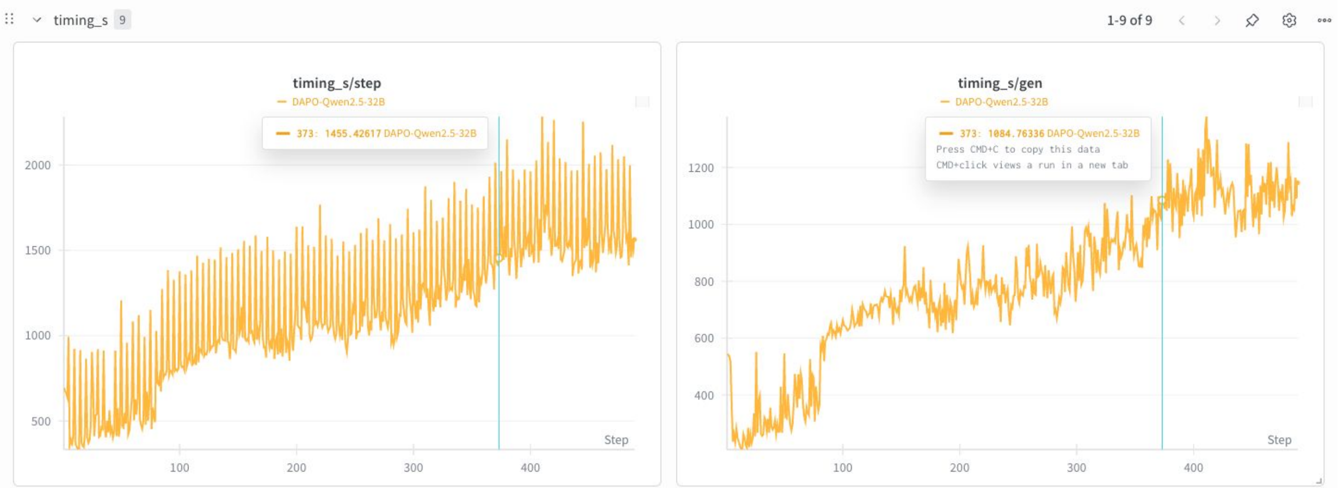
As TIS handles the gap, can we go even **further off-policy** for **speedup**?

Harvesting *Off-Policy* in Quantization

As TIS handles the gap, can we go even **further off-policy** for **speedup**?

Rollout generation is a bottleneck in RL training efficiency:

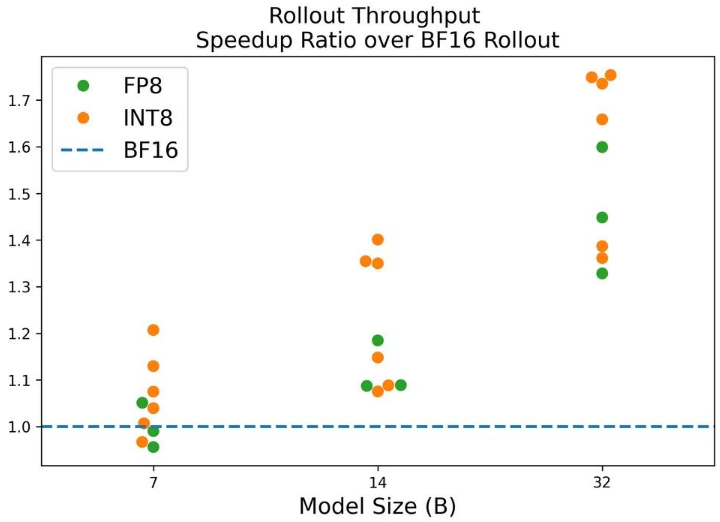
In DAPO-32B setting, rollout takes up ***~70%*** *of the training time*



Quantization helps *speedup* but hurts *performance*

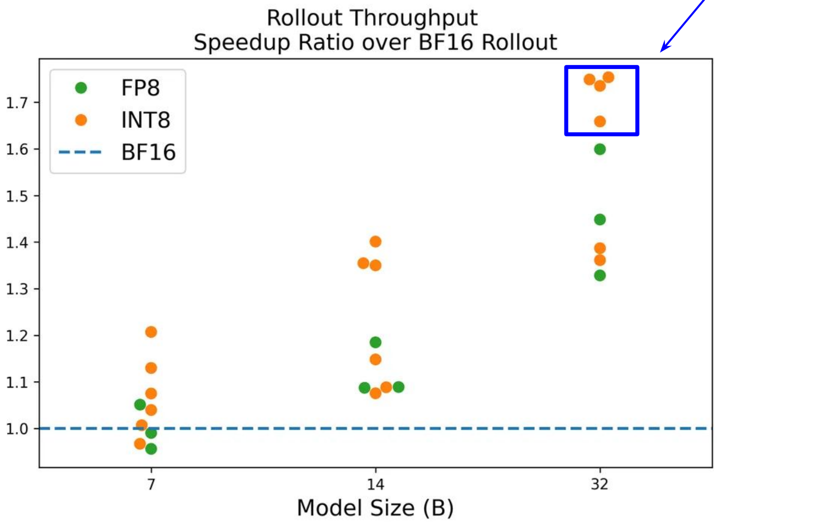
Quantization helps *speedup* but hurts *performance*

Naively applying quantization can accelerate rollout speed



Quantization helps *speedup* but hurts *performance*

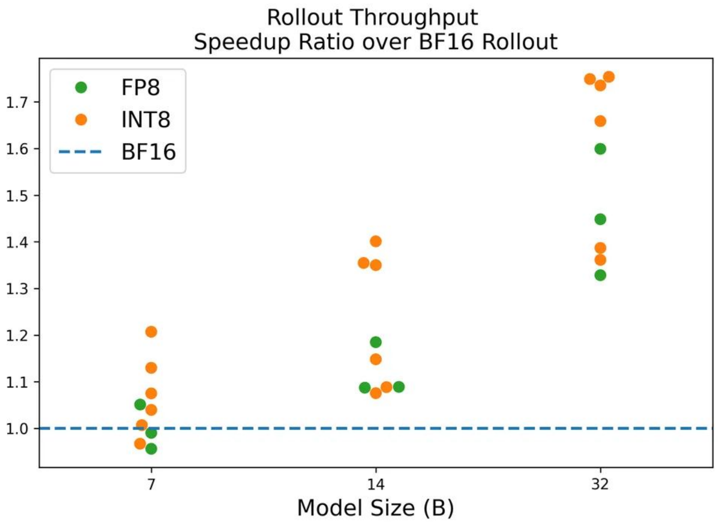
Naively applying quantization can accelerate rollout speed

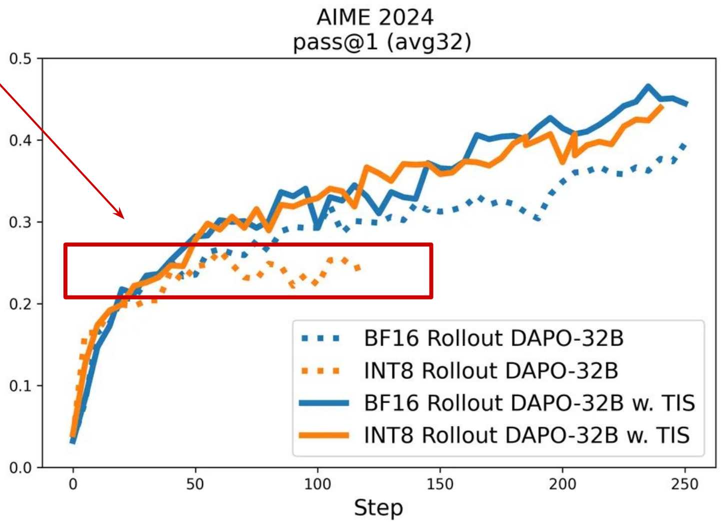


Quantization helps *speedup* but hurts *performance*

Naively applying quantization can accelerate rollout speed

But the performance is also degraded!





Quantization helps *speedup* but hurts *performance*

Naively applying quantization can accelerate rollout speed

But the performance is also degraded!

This can be expected, as quantization introduces more mismatch

$$\underbrace{\mathbb{E}_{a\sim\pi_{\mathrm{bf16}}(\theta_{\mathrm{old}})}}_{\text{int8 Rollout: }\pi_{\mathrm{bf16}}\to\pi_{\mathrm{int8}}}\left[\nabla_\theta \min\left(\frac{\pi_{\mathrm{bf16}}(a,\theta)}{\pi_{\mathrm{bf16}}(a,\theta_{\mathrm{old}})}\hat{A},\ \mathrm{clip}\left(\frac{\pi_{\mathrm{bf16}}(a,\theta)}{\pi_{\mathrm{bf16}}(a,\theta_{\mathrm{old}})},1-\epsilon,1+\epsilon\right)\hat{A}\right)\right]$$

FlashRL preserves performance with TIS

This can be expected, as quantization introduces more mismatch

$$\underbrace{\mathbb{E}_{a\sim\pi_{\text{bf16}}(\theta_{\text{old}})}}_{\text{int8 Rollout: } \pi_{\text{bf16}} \to \pi_{\text{int8}}} \left[\nabla_\theta \min\left(\frac{\pi_{\text{bf16}}(a,\theta)}{\pi_{\text{bf16}}(a,\theta_{\text{old}})} \hat{A},\ \text{clip}\left(\frac{\pi_{\text{bf16}}(a,\theta)}{\pi_{\text{bf16}}(a,\theta_{\text{old}})}, 1-\epsilon, 1+\epsilon \right) \hat{A} \right) \right]$$

FlashRL fixes it with TIS

$$\mathbb{E}_{a\sim\pi_{\text{int8}}(\theta_{\text{old}})} \Bigg[\underbrace{\min\left(\frac{\pi_{\text{bf16}}(a,\theta_{\text{old}})}{\pi_{\text{int8}}(a,\theta_{\text{old}})}, C \right)}_{\text{truncated importance ratio}} \cdot \nabla_\theta \min\left(\frac{\pi_{\text{bf16}}(a,\ \theta)}{\pi_{\text{bf16}}(a,\ \theta_{\text{old}})} \hat{A}, \text{clip}\left(\frac{\pi_{\text{bf16}}(a,\ \theta)}{\pi_{\text{bf16}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon \right) \hat{A} \right) \Bigg]$$

FlashRL preserves performance with TIS

FlashRL fixes it with TIS

$$\mathbb{E}_{a\sim\pi_{\text{int8}}(\theta_{\text{old}})}\left[\underbrace{\min\left(\frac{\pi_{\text{bf16}}(a,\theta_{\text{old}})}{\pi_{\text{int8}}(a,\theta_{\text{old}})},C\right)}_{\text{truncated importance ratio}}\cdot\nabla_\theta\min\left(\frac{\pi_{\text{bf16}}(a,\,\theta)}{\pi_{\text{bf16}}(a,\,\theta_{\text{old}})}\hat{A},\text{clip}\left(\frac{\pi_{\text{bf16}}(a,\,\theta)}{\pi_{\text{bf16}}(a,\,\theta_{\text{old}})},1-\epsilon,\,1+\epsilon\right)\hat{A}\right)\right]$$

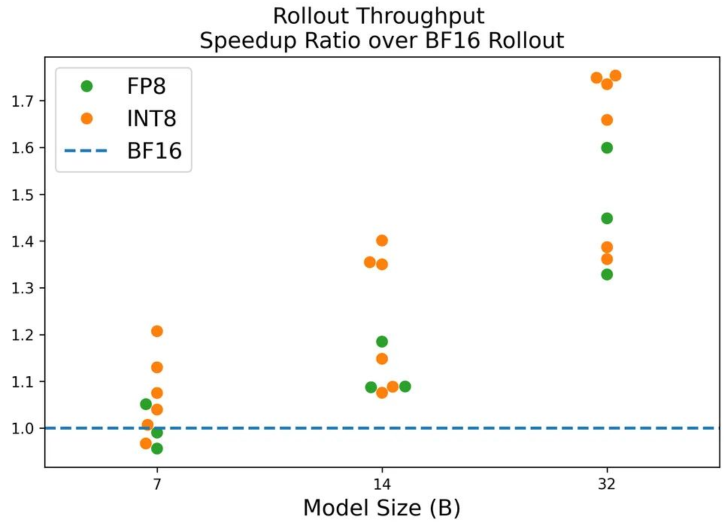
FlashRL is implemented as a PyPI package to patch vLLM

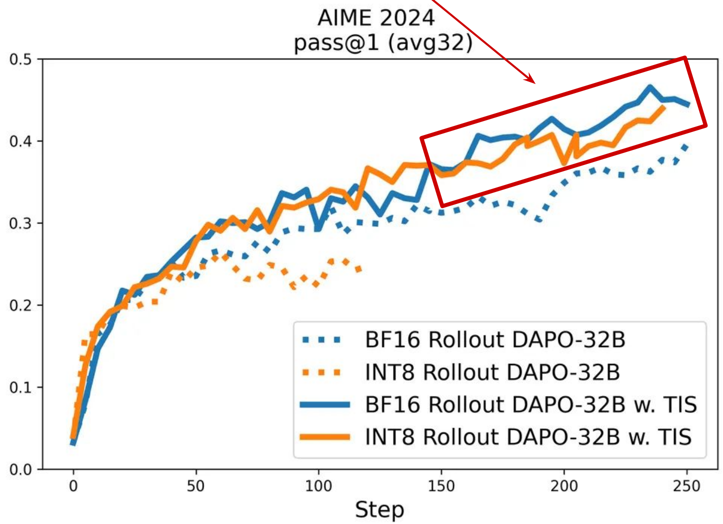
```
pip install flash-llm-rl      # install with pip
export FLASHRL_CONFIG='fp8'  # turn on env variable
bash your-rl-training-script # no code change needed!
```

FlashRL preserves performance with TIS

DAPO 32B Setting

Matches the performance of BF16 rollout with TIS



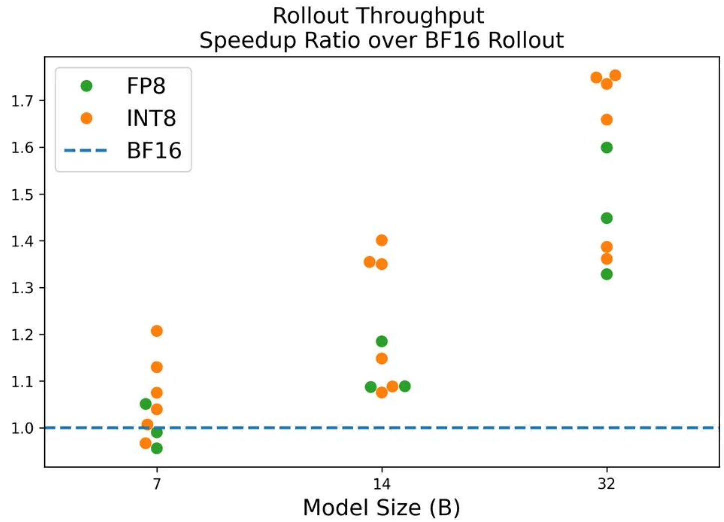


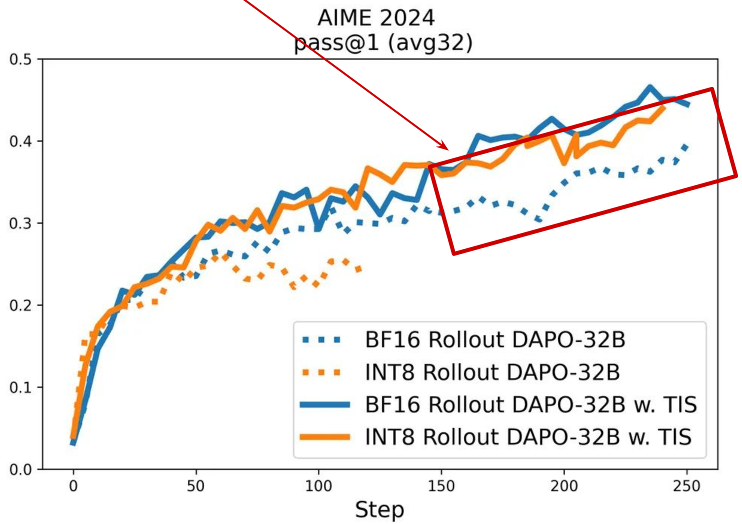
FlashRL preserves performance with TIS

DAPO 32B Setting

Matches the performance of BF16 rollout with TIS

Outperforms naive BF16 rollout (without TIS)

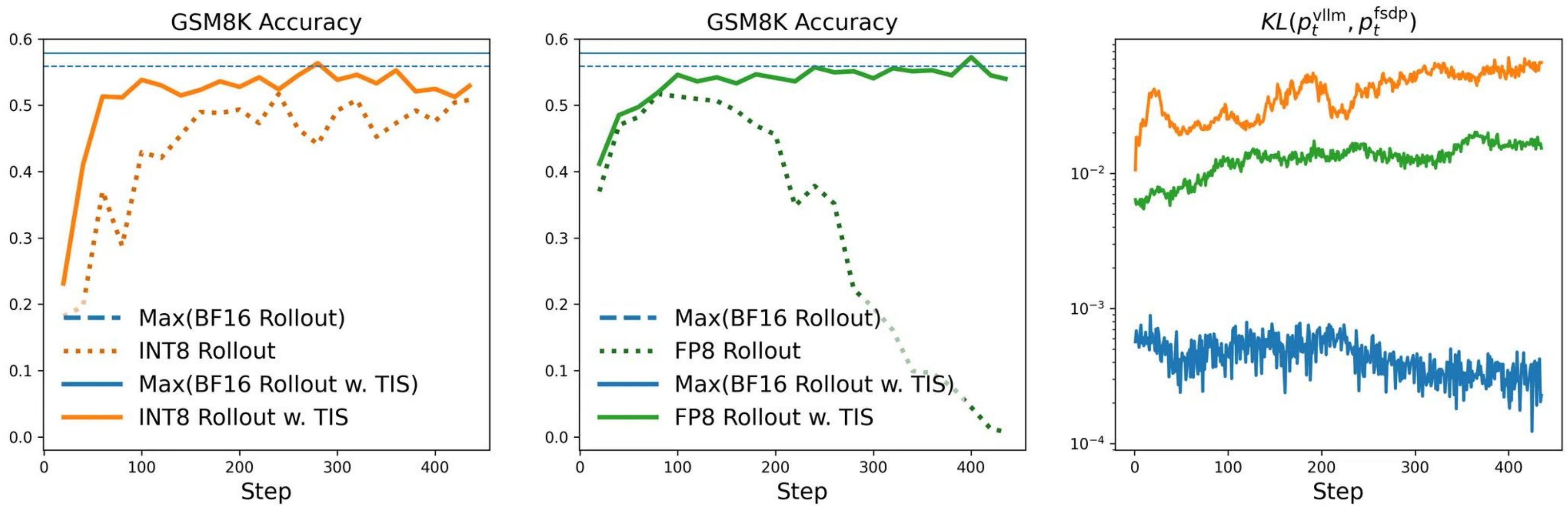




FlashRL preserves performance with TIS

GSM8K 0.5B Setting

TIS works both in INT8 and FP8 setting



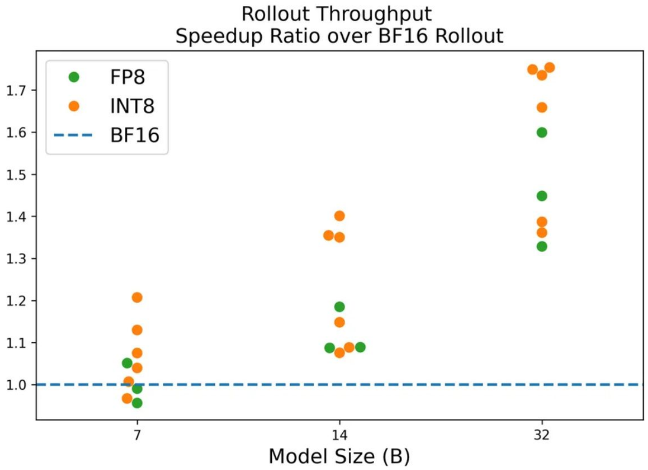
More detailed analysis

Rollout Speedup

More detailed analysis

Rollout Speedup

Regular RL Setting

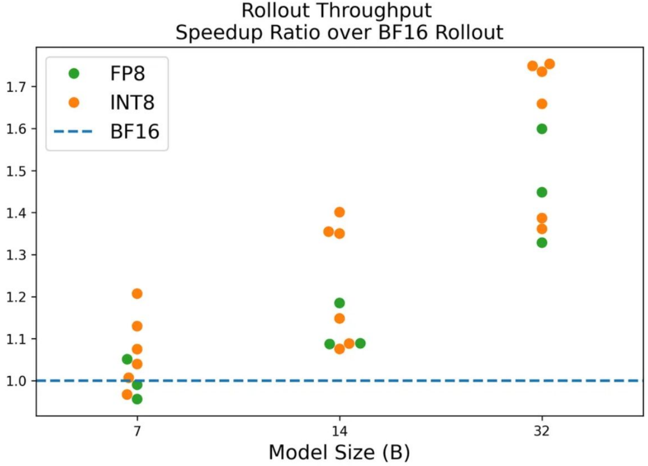


More detailed analysis

Rollout Speedup

Regular RL Setting

Standard Inference Setting



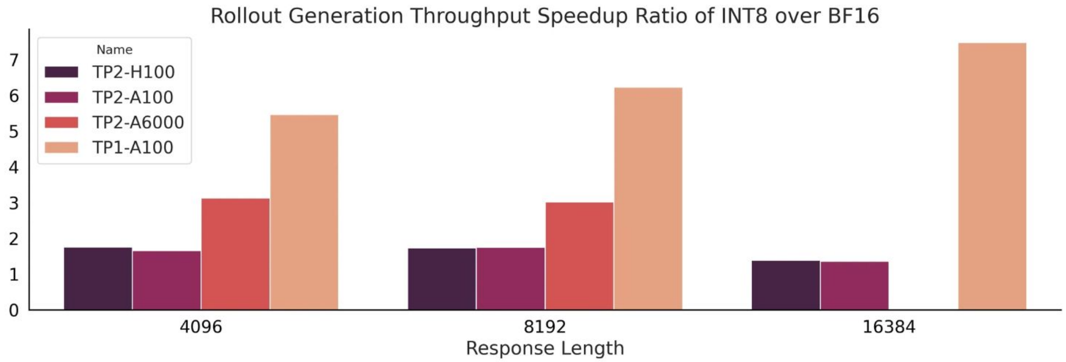


Figure 3. Throughput speedup ratio of INT8-quantized `Deepseek-R1-Distill-Qwen-32B` relative to BF16 in 4 inference-only configurations, measured across varying response lengths

More detailed analysis

End-to-End Speedup & Effectiveness

More detailed analysis

End-to-End Speedup & Effectiveness

INT8 as a pressure test

More detailed analysis

End-to-End Speedup & Effectiveness

INT8 as a pressure test

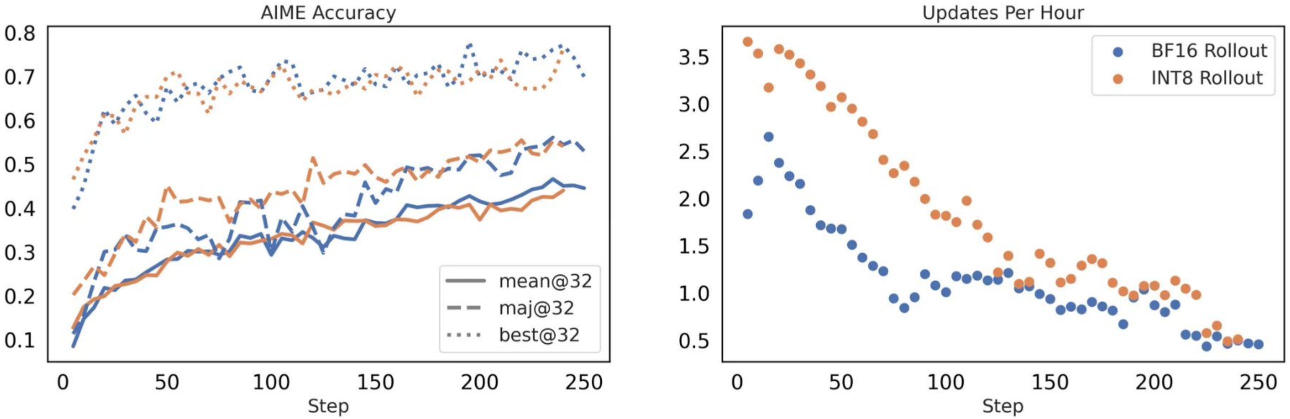


Figure 4. Left: *Downstream performance of RL training with BF16 vs. INT8 rollout precision.* ***Right:*** *Updates per hour achieved with BF16 and INT8 rollout. All experiments use the DAPO recipe on Qwen2.5-32B, trained for 250 steps on 4 nodes with 8×H100 GPUs. [wandb]*

How to perform INT8 quantization?

How to perform INT8 quantization?

FP8 quantization can be naturally conducted in an online manner

How to perform INT8 quantization?

FP8 quantization can be naturally conducted in an online manner

INT8 quantization requires complicated calibration process

How to perform INT8 quantization?

FP8 quantization can be naturally conducted in an online manner

INT8 quantization requires complicated calibration process

Our solution: ***Online INT8 Quantization via Calibration Transfer***

calculate the calibration result once at the beginning of training and reuse it at every online step

How to perform INT8 quantization?

Online INT8 Quantization via Calibration Transfer

calculate the calibration result once at the beginning of training and reuse it at every online step

Observation: RL changes model weights less aggressively comparing to SFT

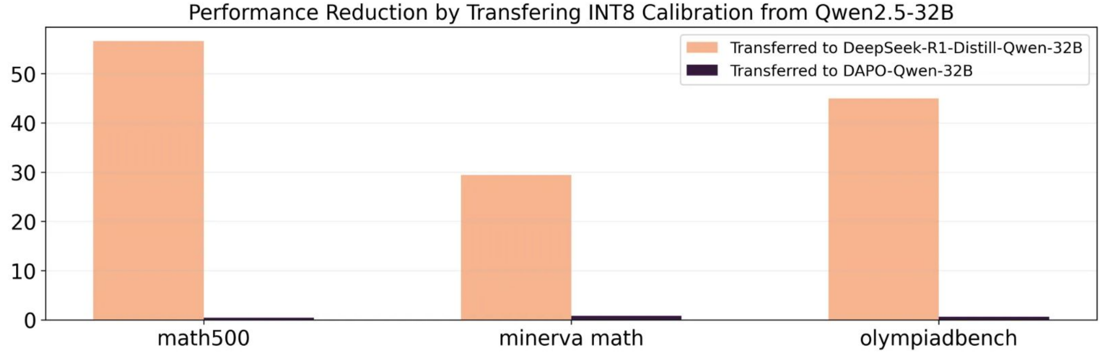


Figure 6. *We experiment on models finetuned via SFT / RL from Qwen2.5-32B base model. We find that compared to SFT, reusing the base model calibration result by applying it to RL funetuned model rarely change the performance. This indicates that INT8 online quantization is practically possible in RL by reusing previous calibration result.*

Outline

- Why Rollout-Training Mismatch Occurs
- How to Fix the Off-Policy Issue It Brings
- Harvesting Rollout-Training Mismatch via Quantization
- **Analyzing the Effectiveness of Different Fixes**

Analyzing the Effectiveness of Different Fixes

- **PPO** $$\mathbb{E}_{a\sim\pi_{\text{fsdp}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$
- **Recompute** $$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

Analyzing the Effectiveness of Different Fixes

- **PPO** $$\mathbb{E}_{a\sim\pi_{\text{fsdp}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\,\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **Recompute** $$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\,\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **PPO-IS** $$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}\,\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

Analyzing the Effectiveness of Different Fixes

- **PPO**

$$\mathbb{E}_{a\sim\pi_{\text{fsdp}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **Recompute**

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **PPO-IS**

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **Vanilla-IS**

$$\mathbb{E}_{\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\underbrace{\frac{\pi_{\text{fsdp}}(a,\theta_{\text{old}})}{\pi_{\text{vllm}}(a,\theta_{\text{old}})}}_{\text{importance ratio}} \cdot \nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

Analyzing the Effectiveness of Different Fixes

- **PPO**

$$\mathbb{E}_{a\sim\pi_{\text{fsdp}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **Recompute**

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **PPO-IS**

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **Vanilla-IS**

$$\mathbb{E}_{\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\underbrace{\frac{\pi_{\text{fsdp}}(a,\theta_{\text{old}})}{\pi_{\text{vllm}}(a,\theta_{\text{old}})}}_{\text{importance ratio}} \cdot \nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **TIS**

$$\mathbb{E}_{\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\underbrace{\min\left(\frac{\pi_{\text{fsdp}}(a,\theta_{\text{old}})}{\pi_{\text{vllm}}(a,\theta_{\text{old}})}, C\right)}_{\text{truncated importance ratio}} \cdot \nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

Comparison with TIS-Variants

GSM8K, PPO, Qwen2.5-0.5B-Instruct

Only TIS works consistently

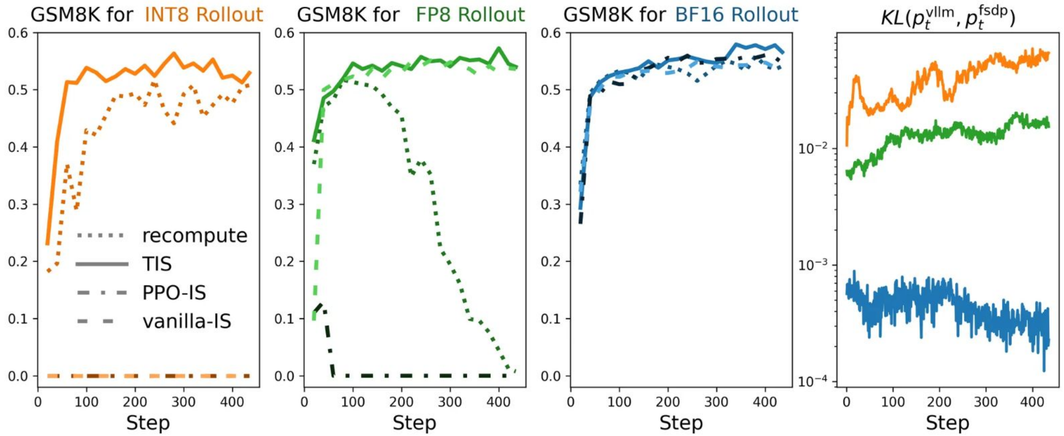


Figure 5. We ablate different rollout-training mismatch mitigation strategies on Qwen2.5-0.5B with GSM8k. Note PPO-IS and Vanilla-IS achieves near 0 accuracy for INT8 rollouts thus being highly overlapped. We also plot the KL divergence between vLLM sampled distribution and the FSDP distribution on the right. *[int8 wandb][fp8 wandb][bf16wandb]*

Why Recompute fails

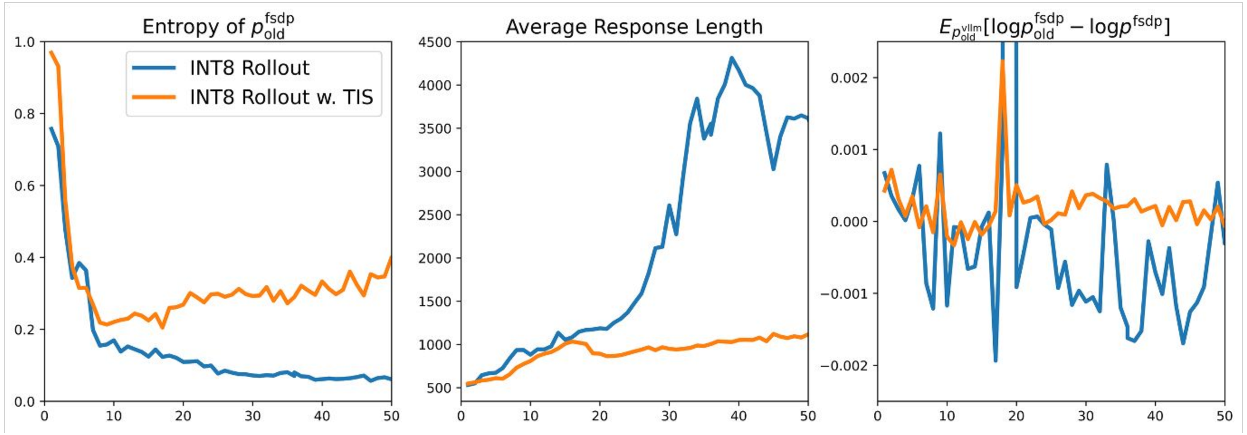
$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **Recompute**
  - The mismatch can lead to entropy collapse
    - Gradient computation vs. rollout generation

Why Recompute fails

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **Recompute**
  - The mismatch can lead to entropy collapse
    - Gradient computation vs. rollout generation

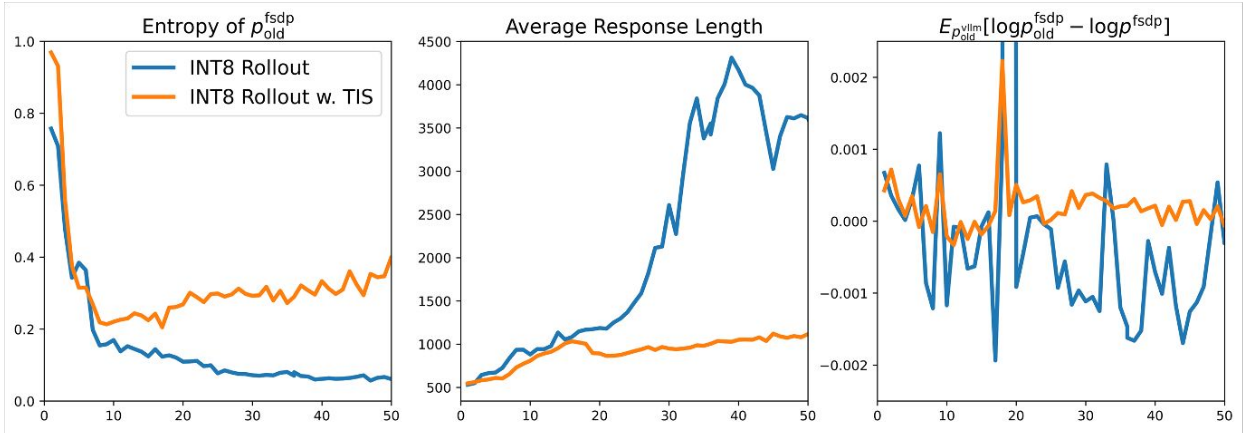


Why Recompute fails

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **Recompute**
  - The mismatch can lead to entropy collapse
    - Gradient computation vs. rollout generation

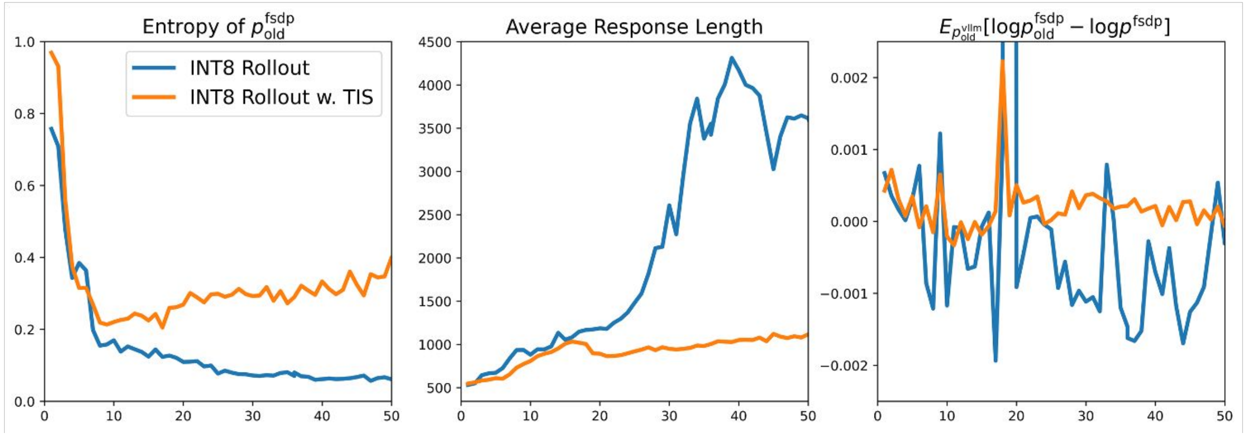
For a with $A < 0$



Why Recompute fails

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **Recompute**
  - The mismatch can lead to entropy collapse
    - Gradient computation vs. rollout generation



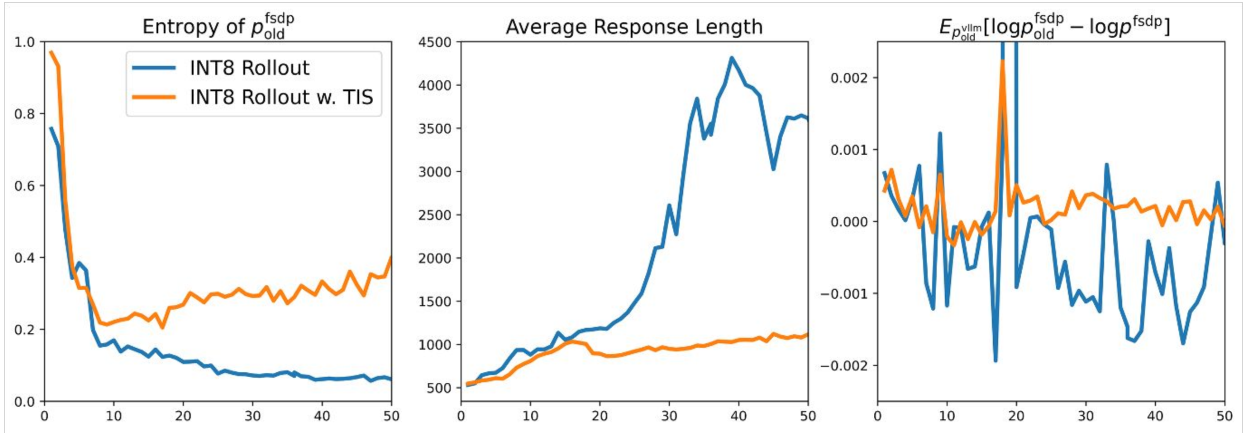
For a with $A < 0$

$\pi_{\text{fsdp}}(a)$ becomes smaller

Why Recompute fails

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **Recompute**
  - The mismatch can lead to entropy collapse
    - Gradient computation vs. rollout generation



For a with $A < 0$

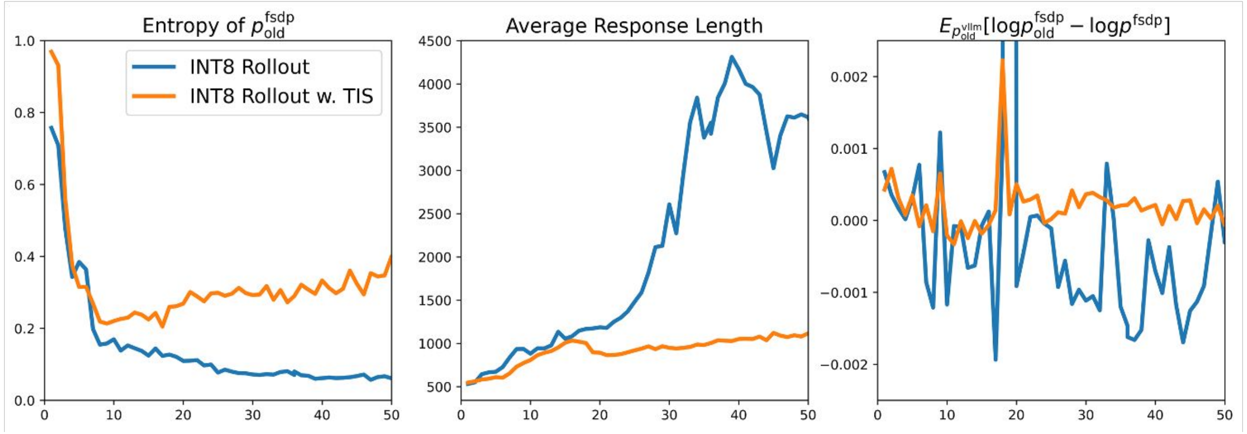
$\pi_{\text{fsdp}}(a)$ becomes smaller

With a large gap w. INT8, $\pi_{\text{vllm}}(a)$ stays the same

Why Recompute fails

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **Recompute**
  - The mismatch can lead to entropy collapse
    - Gradient computation vs. rollout generation



For a with $A < 0$

$\pi_{\text{fsdp}}(a)$ becomes smaller

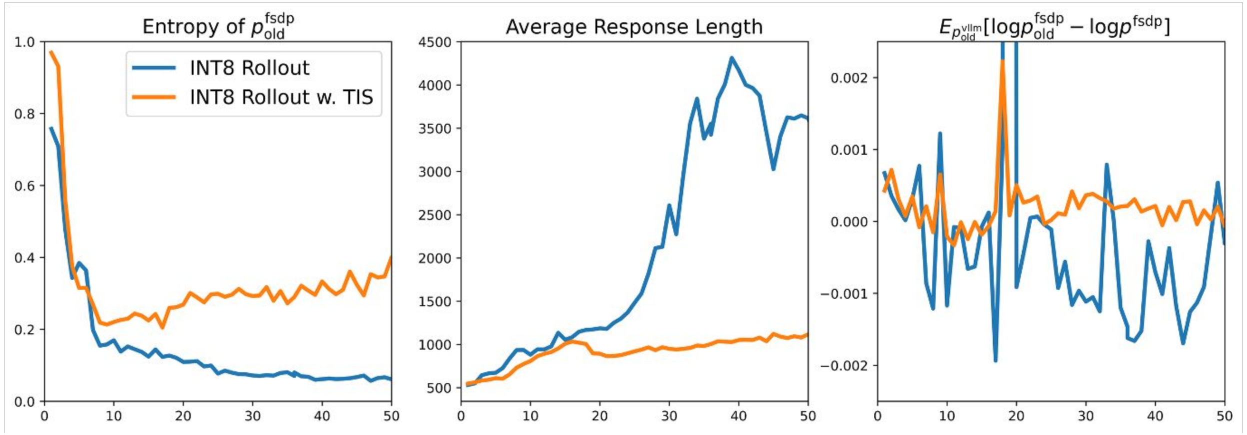
With a large gap w. INT8, $\pi_{\text{vllm}}(a)$ stays the same

$\pi_{\text{fsdp}}(a)$ is over-penalized

Why Recompute fails

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **Recompute**
  - The mismatch can lead to entropy collapse
    - Gradient computation vs. rollout generation



For a with $A < 0$

$\pi_{\text{fsdp}}(a)$ becomes smaller

With a large gap w. INT8, $\pi_{\text{vllm}}(a)$ stays the same

$\pi_{\text{fsdp}}(a)$ is over-penalized

Small entropy

Why PPO-IS fails

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **PPO-IS**
  - PPO-IS is still "biased" from the PPO gradient

$$\mathbb{E}_{a\sim\pi_{\text{fsdp}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

Why PPO-IS fails

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **PPO-IS**
  - PPO-IS is still "biased" from the PPO gradient
  - The clip in PPO is designed for "trust region"
    - At time step 0, $\theta = \theta_{\text{old}}$, we don't want to clip but PPO-IS may clip
    - PPO-clip works differently than TIS

$$\mathbb{E}_{a\sim\pi_{\text{fsdp}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

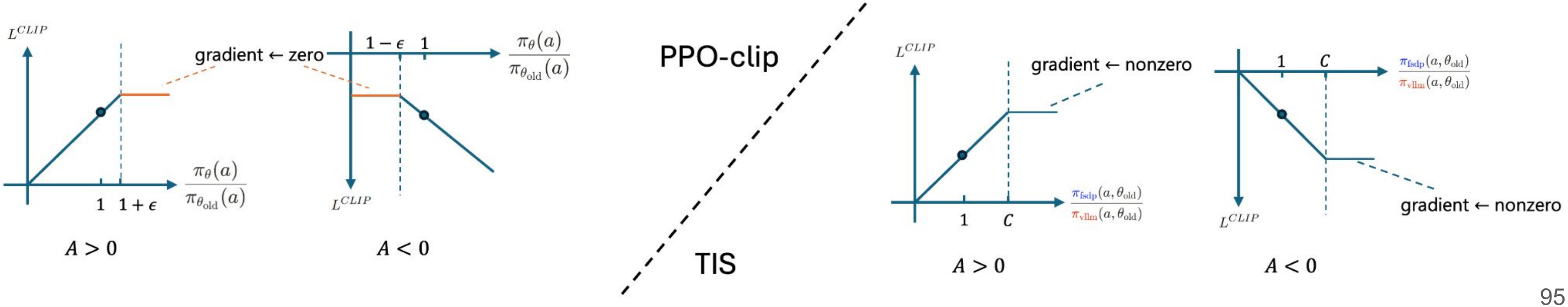
Why PPO-IS fails

$$\mathbb{E}_{a\sim\pi_{\text{vllm}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{vllm}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

- **PPO-IS**
  - PPO-IS is still "biased" from the PPO gradient

$$\mathbb{E}_{a\sim\pi_{\text{fsdp}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

  - The clip in PPO is designed for "trust region"
    - At time step 0, $\theta = \theta_{\text{old}}$, we don't want to clip but PPO-IS may clip
    - PPO-clip works differently than TIS



Why Vanilla-IS fails

$$\frac{1}{N} \cdot \sum_{a_1}^{a_N} \underbrace{\frac{\pi_{\text{fsdp}}(a, \theta_{\text{old}})}{\pi_{\text{vllm}}(a, \theta_{\text{old}})}}_{\text{importance ratio}} \cdot \nabla_\theta \min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A}, \text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}, 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)$$

- **Vanilla-IS**
  - **Uncapped importance ratio amplifies the gradient noise**
    - Leading to unstable training

Why Vanilla-IS fails

$$\frac{1}{N}\cdot\sum_{a_1}^{a_N}\underbrace{\frac{\pi_{\text{fsdp}}(a,\theta_{\text{old}})}{\pi_{\text{vllm}}(a,\theta_{\text{old}})}}_{\text{importance ratio}}\cdot\nabla_\theta\min\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})}\hat{A},\text{clip}\left(\frac{\pi_{\text{fsdp}}(a,\ \theta)}{\pi_{\text{fsdp}}(a,\ \theta_{\text{old}})},1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)$$

- **Vanilla-IS**
  - **Uncapped importance ratio amplifies the gradient noise**
    - Leading to unstable training

Gradient noise

constant factor, no BP through

$$\frac{1}{N^2}\cdot\sum_{a_1}^{a_N}\underbrace{\left(\frac{\pi_{\text{fsdp}}(a,\theta_{\text{old}})}{\pi_{\text{vllm}}(a,\theta_{\text{old}})}\right)^2}_{\text{importance ratio}}\cdot Var[\nabla_\theta\cdots] \quad = \quad Var[\qquad]$$

$$\frac{1}{N}E[Var[\nabla_\theta(x,y)]]\approx\frac{1}{N^2}\sum_{x,y}Var[\nabla_\theta(x,y)] \quad = \quad Var\left[\frac{1}{N}\sum_{x,y}\nabla_\theta(x,y)\right]$$

What's beyond?

What’s beyond?

- The gap can be amplified in MoE RL
  - Dynamic Routing
  - Specially Optimized Kernels

What’s beyond?

- **The gap can be amplified in MoE RL**
  - **Dynamic Routing**
  - **Specially Optimized Kernels**

- **TIS is orthogonal and compatible with existing GxPOs**
  - **GxPOs adjust the computation of advantage / importance ratio**
  - **TIS addresses the system-level mismatch problem**

Takeaways

- Mixing inference backend with training backends brings off-policy RL training, even if they share the same weights
- Truncated Importance Sampling (TIS) is effective mitigating the gap
- With TIS integrated, rollout generation can be accelerated via quantization without sacrificing the performance

Thanks for Listening!

Feng Yao

https://yaof20.github.io/

August 29, 2025 – Presented at QingkeAI