



# On the Rollout-Training Mismatch In Modern RL Systems



**Feng Yao\***



**Liyuan Liu\***



**Dinghuai Zhang**



**Chengyu Dong**



**Jingbo Shang**



**Jianfeng Gao**

# Efficient RL systems are rising

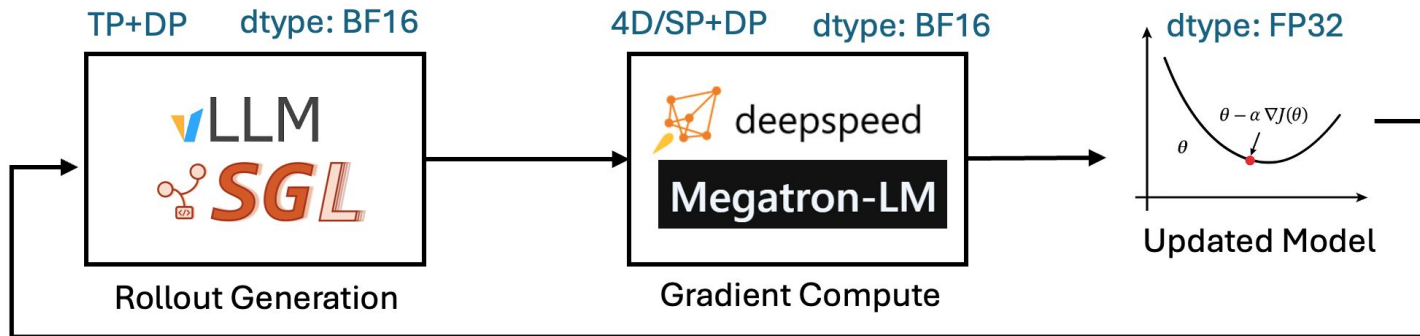
- VeRL/OpenRLHF/Slime adopts **hybrid engines**

# Efficient RL systems are rising

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

# Efficient RL systems are rising

- VeRL/OpenRLHF/Slime 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

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

# It also brings an issue...

- **Rollout-Training Mismatch**

- Expected

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

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

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

# It also brings an issue...

- **Rollout-Training Mismatch**

- Expected

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

- 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...

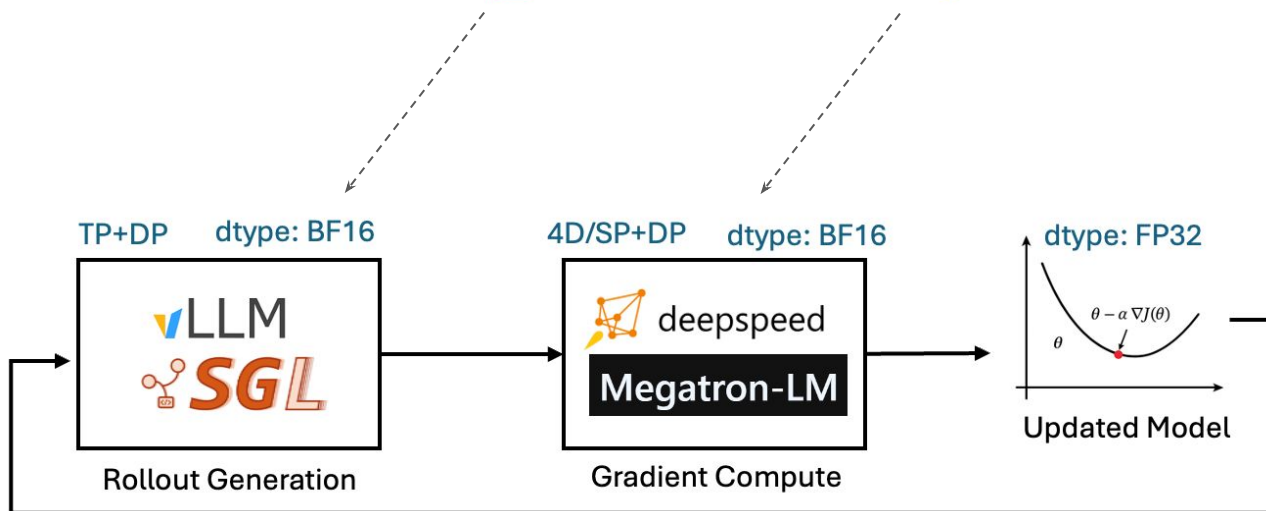
- Rollout-Training Mismatch
  - For the **same** rollout & model parameter

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

# It also brings an issue...

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

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



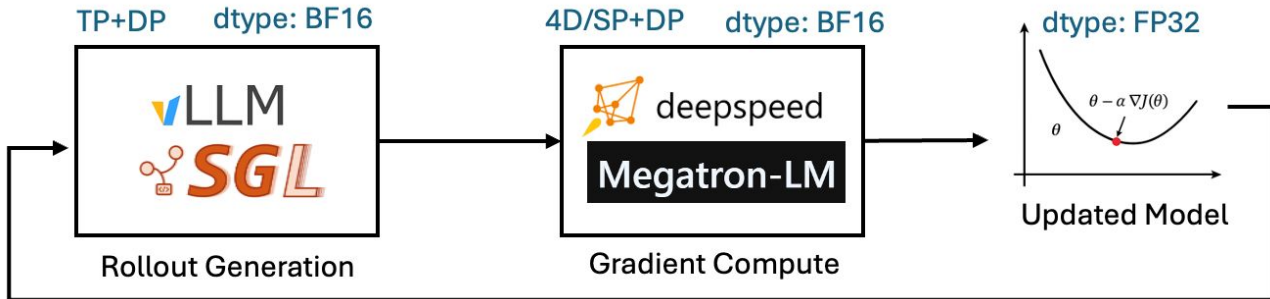
# It also brings an issue...

- **Rollout-Training Mismatch**

- For the **same** rollout & model parameter

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

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



# It also brings an issue...

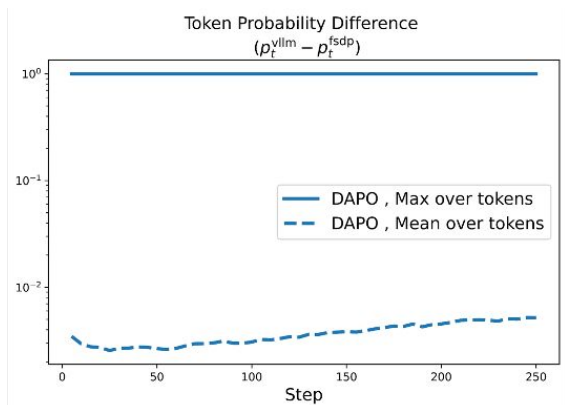
- Rollout-Training Mismatch

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

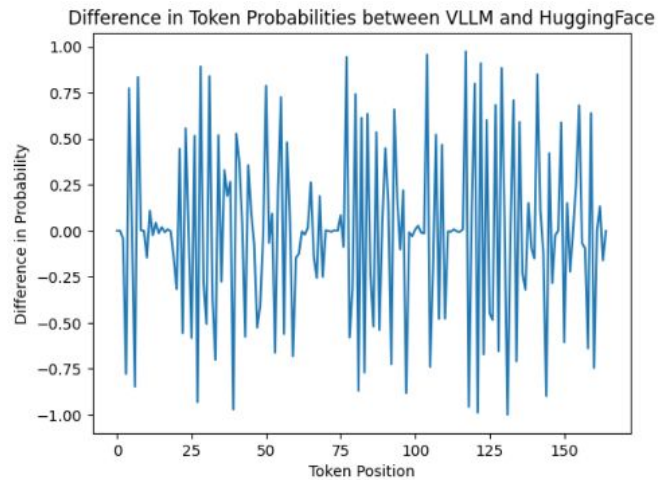
# It also brings an issue...

- Rollout-Training Mismatch

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



DAPO Qwen2.5-32B



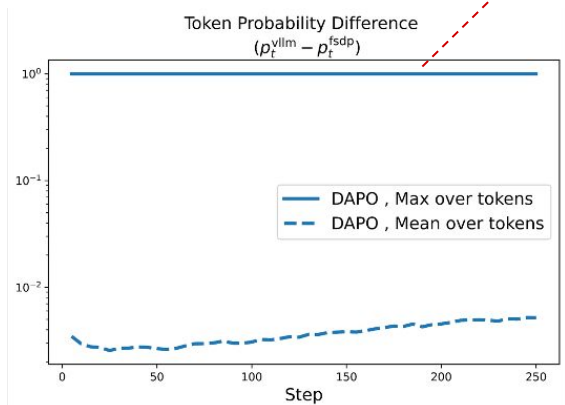
DS-Qwen2.5-1.5B

# It also brings an issue...

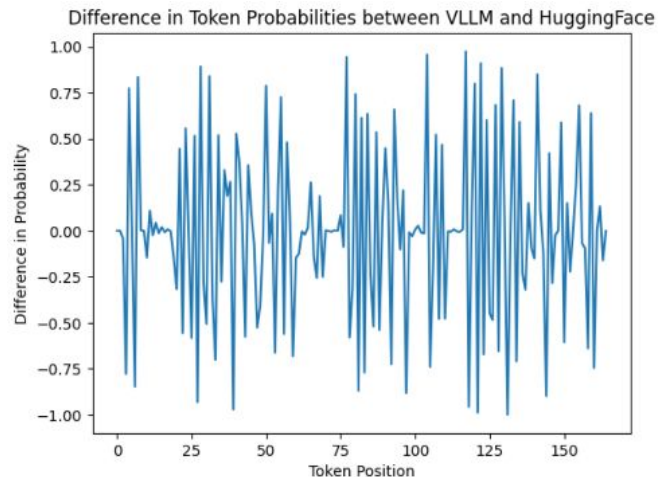
- Rollout-Training Mismatch

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

**Max Diff = 1.0**



DAPO Qwen2.5-32B



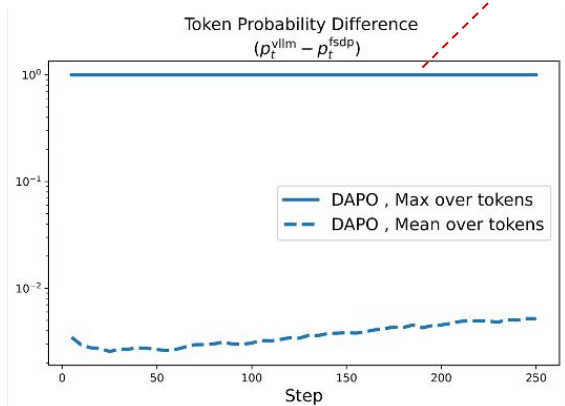
DS-Qwen2.5-1.5B

# It also brings an issue...

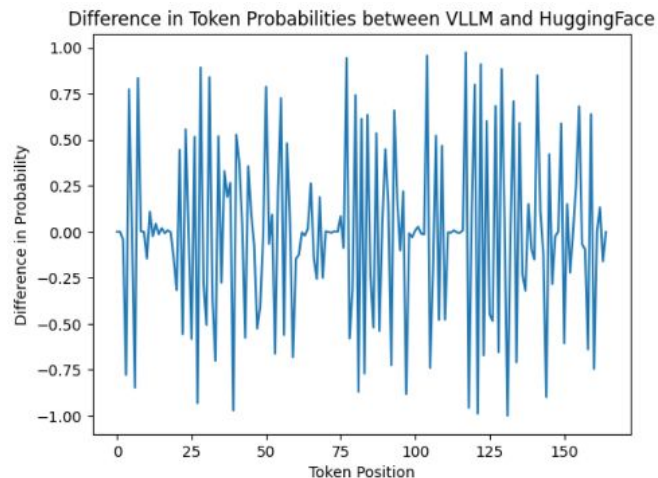
- Rollout-Training Mismatch

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

**Max Diff = 1.0**



DAPO Qwen2.5-32B



DS-Qwen2.5-1.5B

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

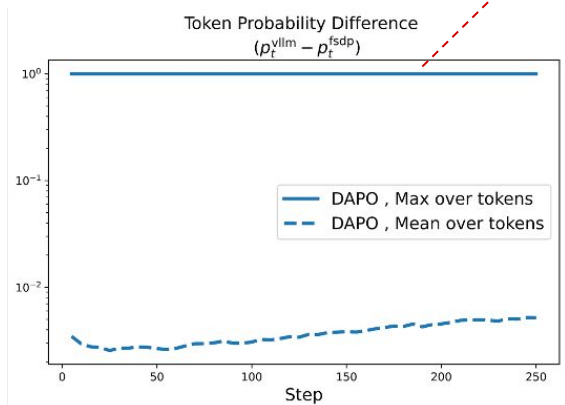
# It also brings an issue...

Implicitly makes RL “**Off-Policy**”!

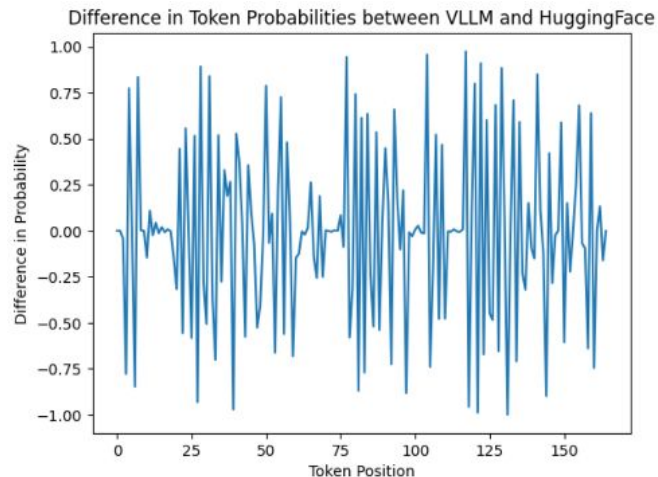
- Rollout-Training Mismatch

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

**Max Diff = 1.0**



DAPO Qwen2.5-32B



DS-Qwen2.5-1.5B

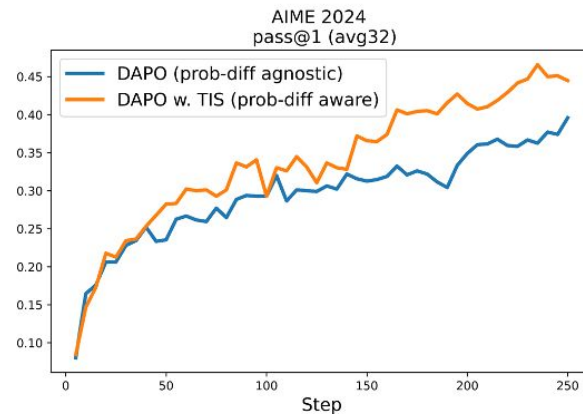
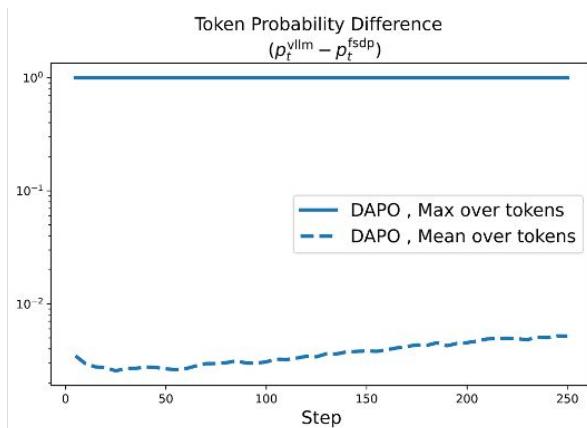
$$p^{vllm} = 1.0 \quad \& \quad p^{fsdp} = 0.0$$

# 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-Policy*ness via Quantization

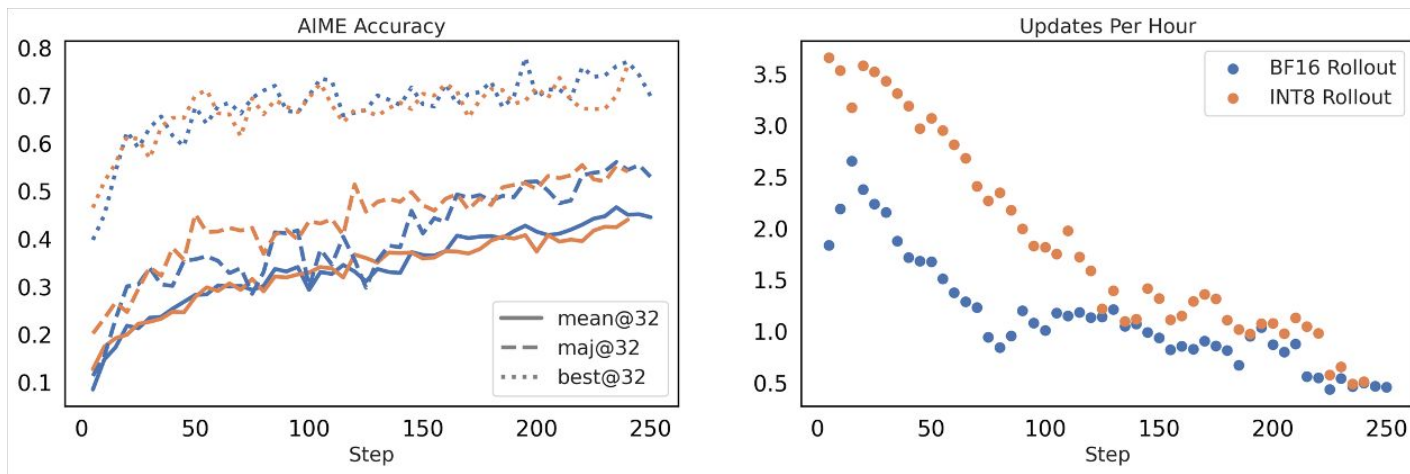
- Since TIS is able to handle the mismatch

# 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-Policy*ness 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**
- **Additional Analyses**

# Outline

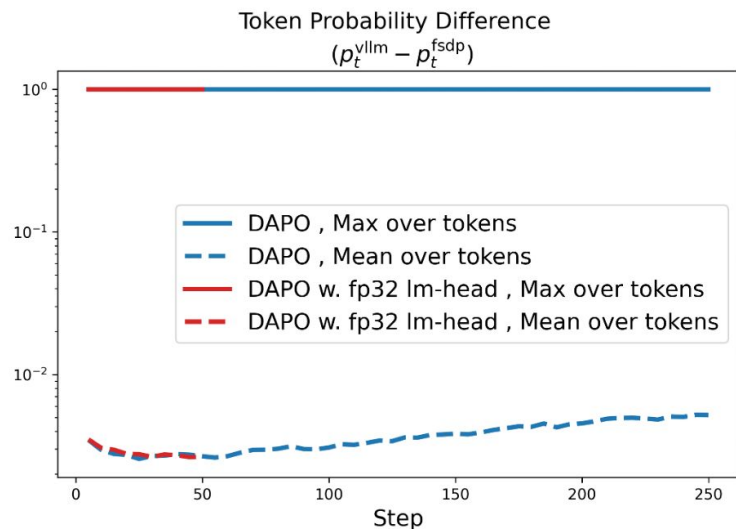
- **Why Rollout-Training Mismatch Occurs**
- How to Fix the Off-Policy Issue It Brings
- Harvesting Rollout-Training Mismatch via Quantization
- Analyzing the Effectiveness of Different Fixes
- Additional Analyses

# 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

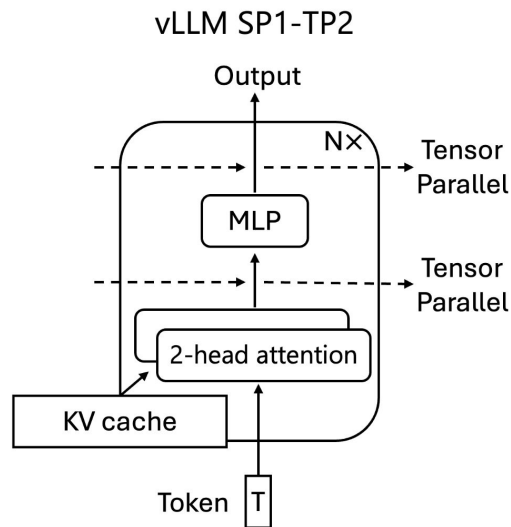
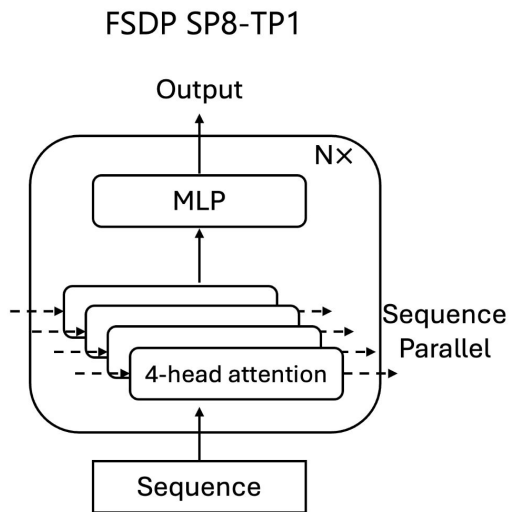


# Why does Rollout-Training Mismatch occur?

- 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
- Additional Analyses

# 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:**

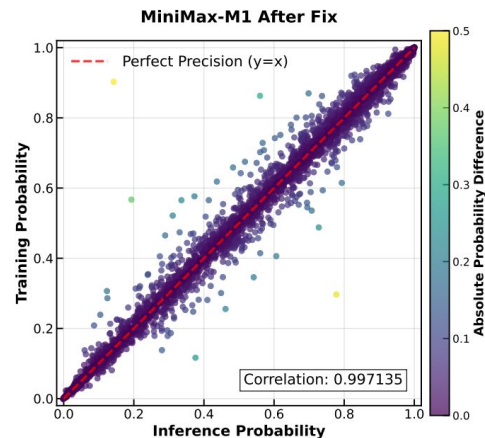
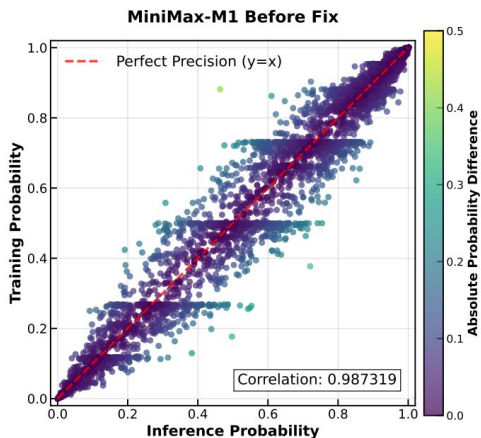
# How to Fix the Off-Policy Issue It Brings

- **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**

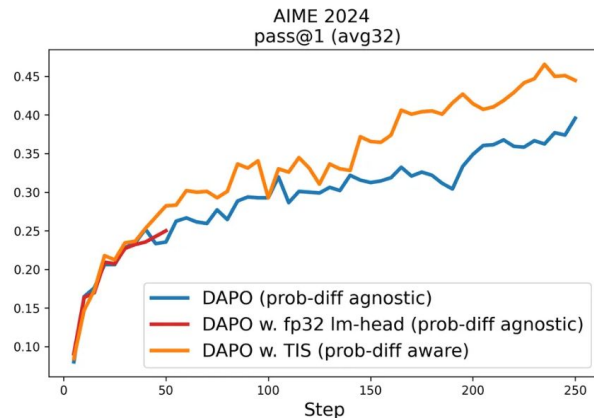
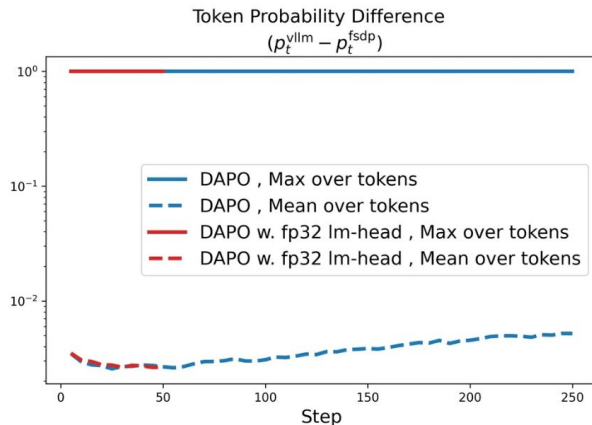
It helps, but ...



# 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 → **Importance sampling correction:**

# How to Fix the Off-Policy Issue It Brings

- Trial 2 – Apply algorithm-level fix
  - Be aware of the mismatch → **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 → **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 → **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 → **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 → **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):**

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

# 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**)

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

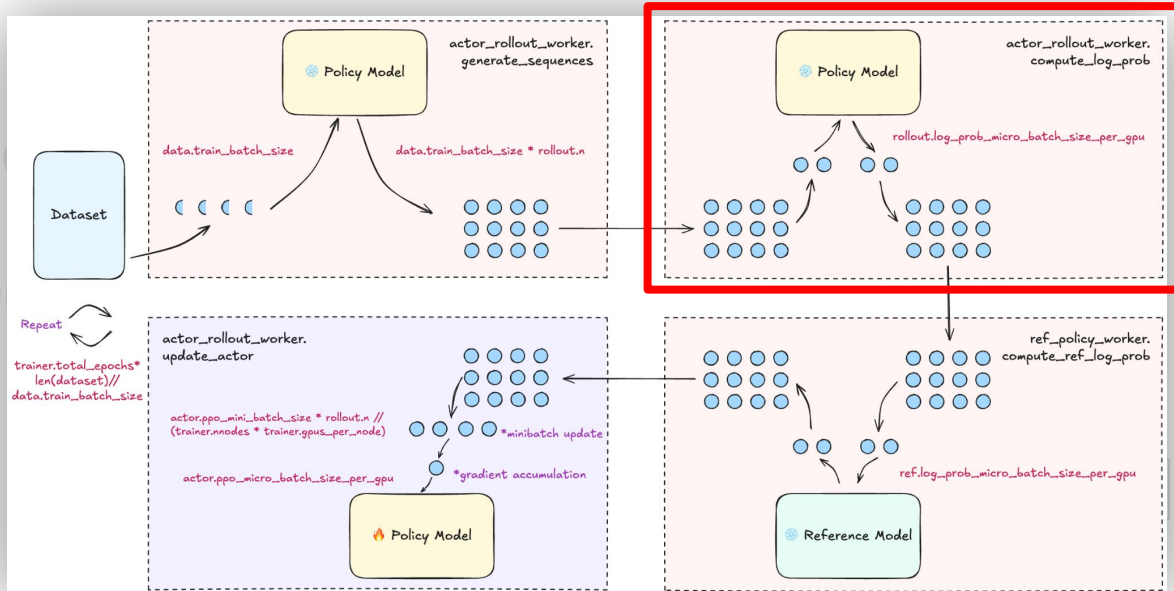
- VeRL/OpenRLHF's Implementation (**recompute**)

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



# How to

- Extending



- 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**)

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

# 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**)

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

# Why Not Alternative Methods?

- Variants of TIS

- PPO Importance Sampling (**PPO-IS**)

**A commonly  
asked variant**

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

# Why Not Alternative Methods?

- Variants of TIS

- PPO Importance Sampling (**PPO-IS**)

**A commonly  
asked variant**

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

**Can break out of the trust region**

# Why Not Alternative Methods?

- Variants of TIS

- PPO Importance Sampling (**PPO-IS**)

**A commonly asked variant**

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

**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**

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

**Can break out of the trust region**

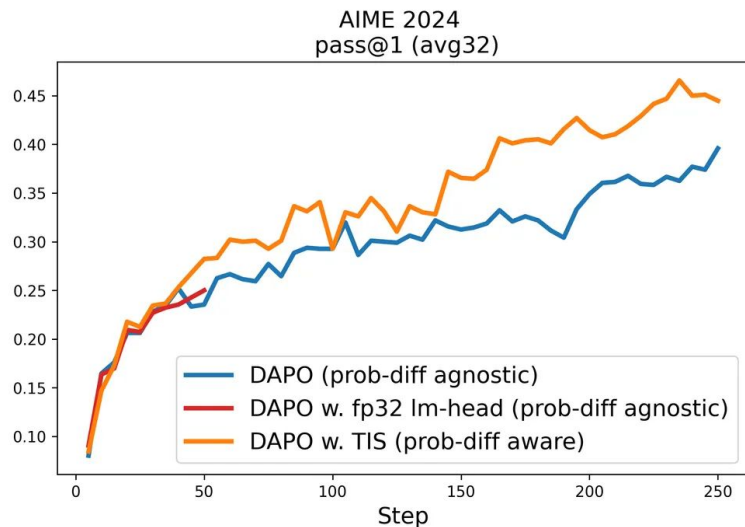
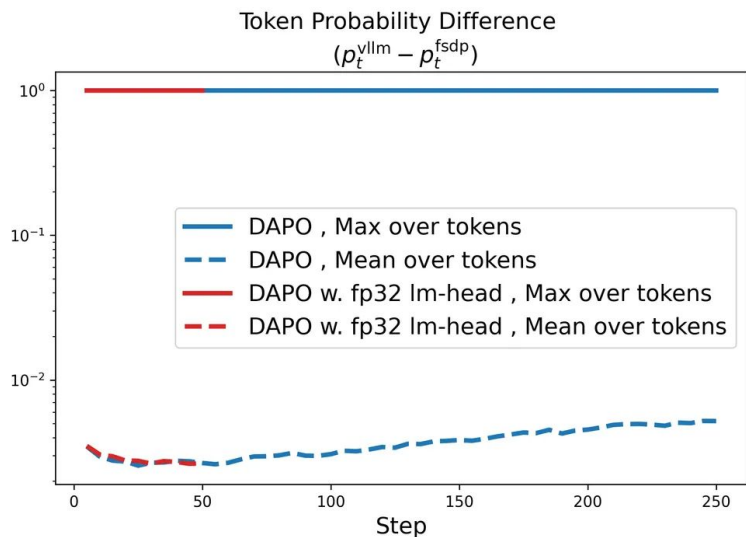
- Vanilla Importance Sampling (**Vanilla-IS**)

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

**Can be too large and makes training crash**

# How well can TIS fix it?

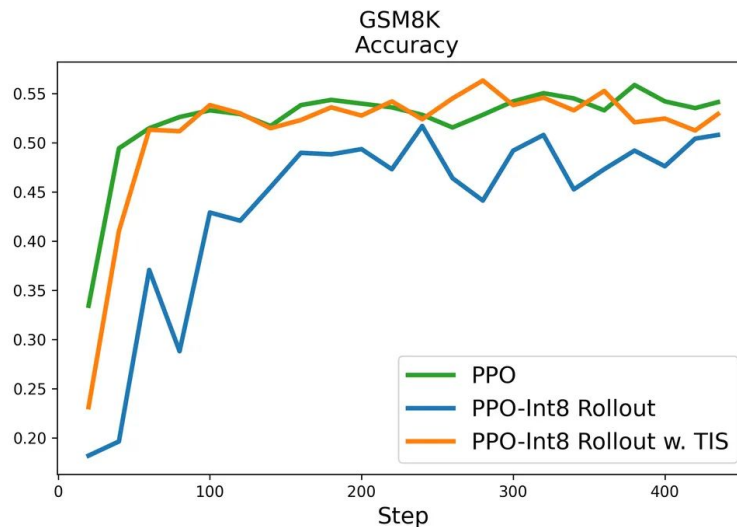
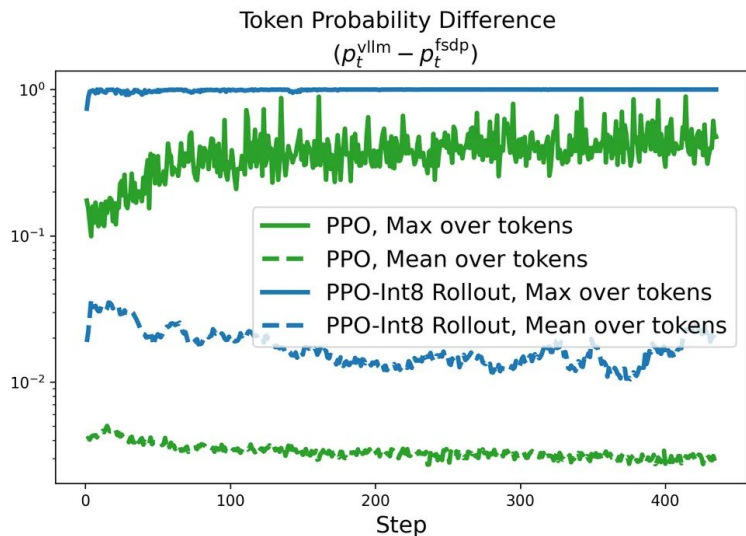
- DAPO 32B Setting



# How well can TIS fix it?

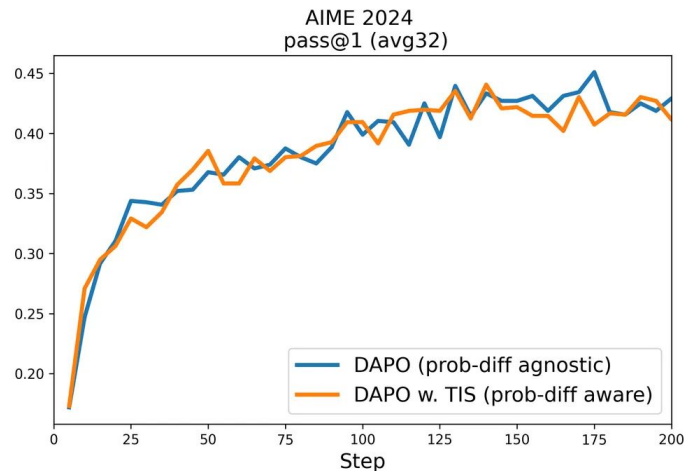
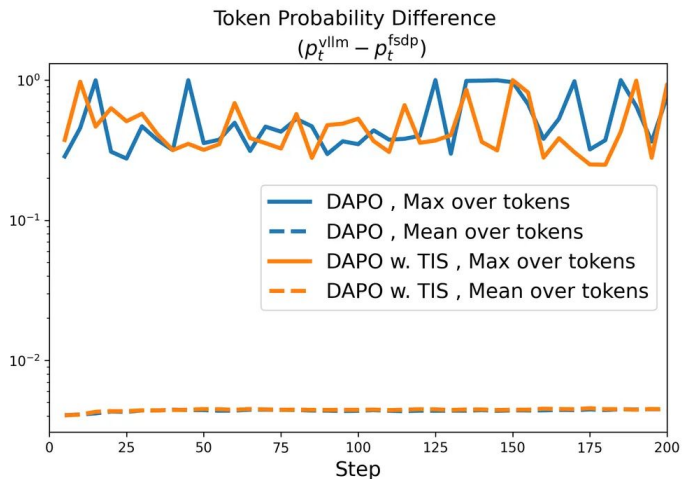
- **GSM8K 0.5B Setting**

- Normal RL: Max Diff is smaller ( $\sim 0.4$ ) than 1.0 (in DAPO-32B setting)
- INT8 Rollout: Max Diff is larger ( $\sim 1.0$ ) than normal RL setting



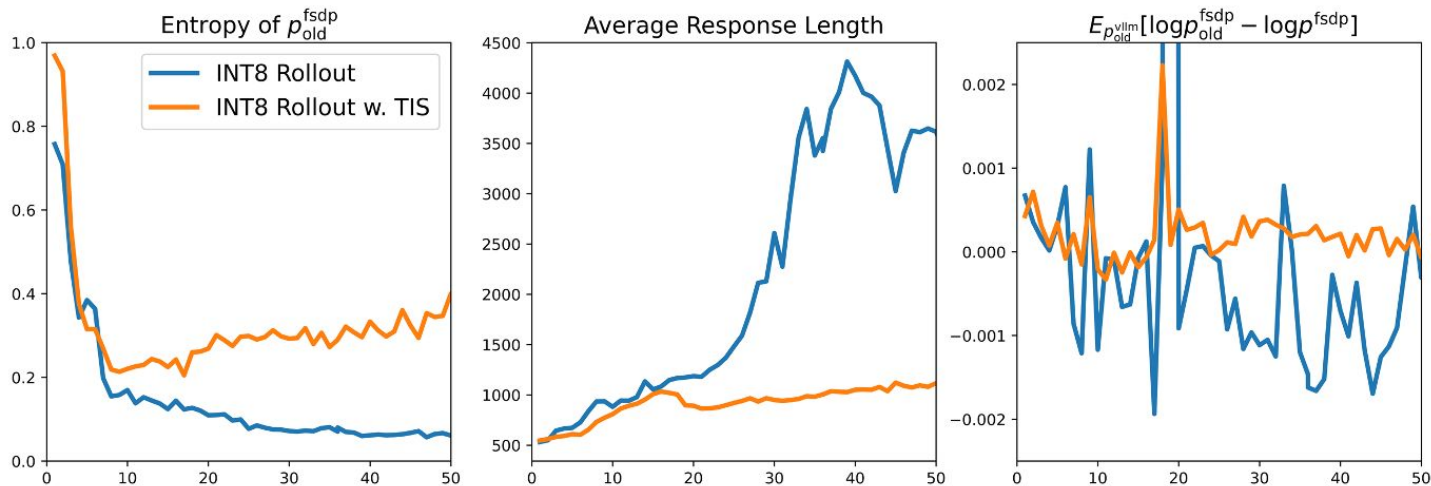
# Does TIS always help?

- 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?

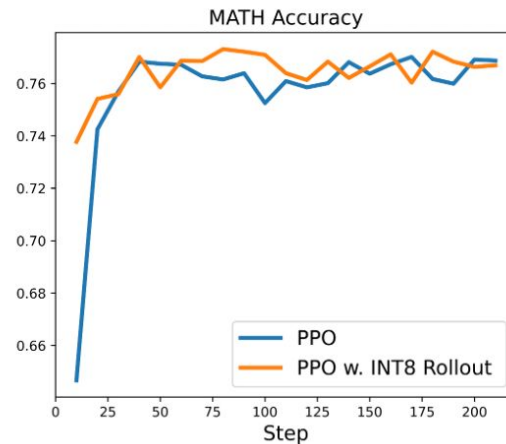
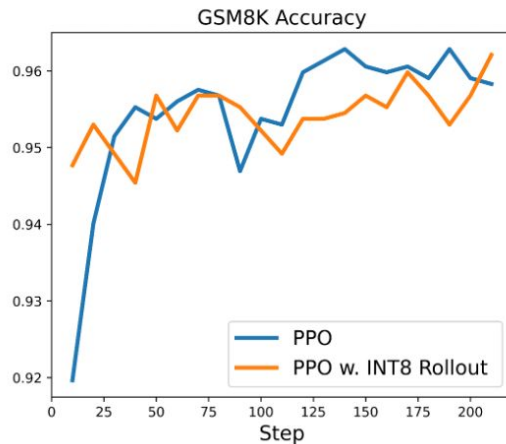
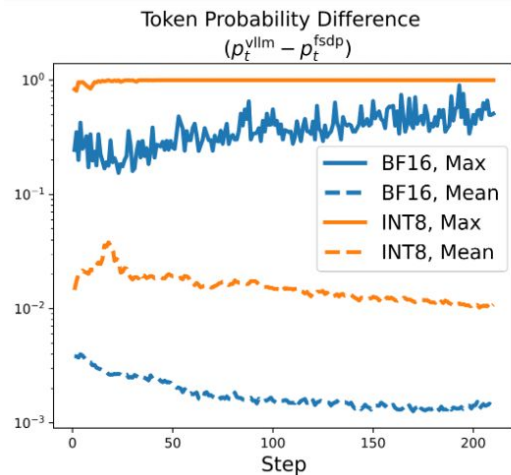
- Unexpected training instability on challenging tasks



DAPO Qwen2.5-32B

# Does the Mismatch really matter?

- Possible negligible on simple tasks



PPO GSM8K Qwen2.5-32B

# Community Verification

OpenRLHF / OpenRLHF

Releases / v0.8.9

### Release v0.8.9

hijkzzz released this 2 weeks ago · 9 commits to main since this release · v0.8.9 · 3c9c109

#### What's Changed

- Add TensorBoard logging to PRM training by @xli360 in #1096
- Support vLLM off-policy importance sampling correction by @xiaoxigua999 and @MooMoo-Yang in #1098
  - Requires vLLM version > 0.10: pip install -U vllm --pre --extra-index-url https://wheels.vllm.ai/nightly
- Fix weight broadcasting issue in Async RL with PyTorch 2.7.1 and vLLM 0.10 by @xiaoxigua999 in #1100
- Fix sequence-level loss calculation for GSPQ by @xiaoxigua999

Full Changelog: [v0.8.8...v0.8.9](#)

voicengine / vert

### [BREAKING][vllm, fsdp] feat: add Rollout-Training Mismatch Fix -- Truncated importance sampling #2953

zhaochenyang20 merged 17 commits into voicengine:main from yao20:truncated\_importance\_sampling yesterday

yaof20 commented 3 weeks ago

#### What does this PR do?

Support vLLM-FSDP off-policy importance sampling correction using Truncated Importance Sampling (TIS):

Expected objective:

$$\mathbb{E}_{\pi_{\text{train}}(\theta_{\text{old}})}[\nabla_{\theta} \min_{\pi_{\text{roll}}(\theta_{\text{roll}})} \int \hat{A} \text{clip}\left(\frac{\pi_{\text{roll}}(\theta_{\text{roll}})}{\pi_{\text{train}}(\theta_{\text{old}})} 1 - \epsilon, 1 + \epsilon\right) \hat{A}] \quad (1)$$

Mismatch: people actually do rollout with vLLM:

$$s = \pi_{\text{roll}}(\theta_{\text{roll}}) \Rightarrow s = \pi_{\text{roll}}(\theta_{\text{roll}}) \quad (2)$$

Given  $s = \pi_{\text{roll}}(\theta_{\text{roll}})$ , standard vet recognize  $\pi_{\text{roll}}(\theta_{\text{roll}})$  with Inference interface:

$$\mathbb{E}_{\pi_{\text{roll}}(\theta_{\text{roll}})}[\nabla_{\theta} \min_{\pi_{\text{roll}}(\theta_{\text{roll}})} \int \hat{A} \text{clip}\left(\frac{\pi_{\text{roll}}(\theta_{\text{roll}})}{\pi_{\text{train}}(\theta_{\text{old}})} 1 - \epsilon, 1 + \epsilon\right) \hat{A}] \quad (3)$$

Our proposed truncated importance sampling (TIS):

$$\mathbb{E}_{\pi_{\text{roll}}(\theta_{\text{roll}})}[\min_{\pi_{\text{roll}}(\theta_{\text{roll}})} \left( \frac{\pi_{\text{roll}}(\theta_{\text{roll}})}{\pi_{\text{train}}(\theta_{\text{old}})} C \right) \nabla_{\theta} \min_{\pi_{\text{roll}}(\theta_{\text{roll}})} \int \hat{A} \text{clip}\left(\frac{\pi_{\text{roll}}(\theta_{\text{roll}})}{\pi_{\text{train}}(\theta_{\text{old}})} 1 - \epsilon, 1 + \epsilon\right) \hat{A}] \quad (4)$$

THUDM / slime

Code Issues 25 Pull requests 9 Actions Projects Security

## Naive implementation of rollout logg correction in RL training #179

zhuzilin merged 4 commits into THUDM:main from yitianlian:sg\_llgp 2 days ago

Conversation 0 Commits 4 Checks 1 Files changed 9 +83 -3

yitianlian commented 2 weeks ago

The implementation of the importance ratio between rollout logg and training logg.

Reviewers: No reviews

Assignees: No assignees

## REINFORCE++-baseline is all you need in RLVR

Janhu Follow 3 min read · 5 days ago

### Tool-Integrated Reasoning and Agent Experiments

We have thoroughly validated the effectiveness of global standard deviation and global advantage normalization in complex multi-turn tool call scenarios. Our experiments utilize the framework established by [arxiv:2505.07773](#), which features a zero-shot agent environment designed for large language models to tackle mathematical problems using Qwen 2.5 Base 7B.

		300-step/avg@32/num-run-2/inference	aim@4	aim@25	hmmr_feb_2024	hmmr_feb_2024	aimmc	avg
reinforce++	with-baseline	30.83333	27.1875	17.91667	18.95833	25.625	24.1041667	
reinforce++	with-baseline,vllm-correction	31.5625	23.9583	20.625	20.1042	29.375	23.125	
gpo		31.66667	21.875	16.9167	17.70833	24.8675	22.9833333	
gpo		31.20833	21.66667	16	18.4375	23.95833	22.8541667	

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.

Zichen Liu @zclccc · Aug 22

With just a few lines of code, Feng's (@fengyao1909) suggested fix—applying importance sampling on the behavior policy—resolved the training instability in my case (oat). I believe the result can generalize to other RL frameworks as well. Great work, Feng!

With truncated importance sampling (orange), the training becomes more stable.

$$\mathbb{E}_{\pi_{\text{roll}}(\theta_{\text{roll}})}[\min_{\pi_{\text{roll}}(\theta_{\text{roll}})} \left( \frac{\pi_{\text{roll}}(\theta_{\text{roll}})}{\pi_{\text{train}}(\theta_{\text{old}})} C \right) \nabla_{\theta} \min_{\pi_{\text{roll}}(\theta_{\text{roll}})} \int \hat{A} \text{clip}\left(\frac{\pi_{\text{roll}}(\theta_{\text{roll}})}{\pi_{\text{train}}(\theta_{\text{old}})} 1 - \epsilon, 1 + \epsilon\right) \hat{A}]$$

truncated importance ratio

Reference: <https://fengyao.notion.site/off-policy-df>

Qian Liu @sivil\_taram · Aug 25

TIPS has been remarkably effective in my practice—more people should definitely know about it!

Feng Yao @fengyao1909 · Aug 25

We are glad that TIS and FlashRL have received broad attention from the open-source community that they have been verified and supported (OpenRLHF @hijkzzz, SkyRL @NovaSkyAI, REINFORCE++@hijkzzz, OAT @zclccc)!

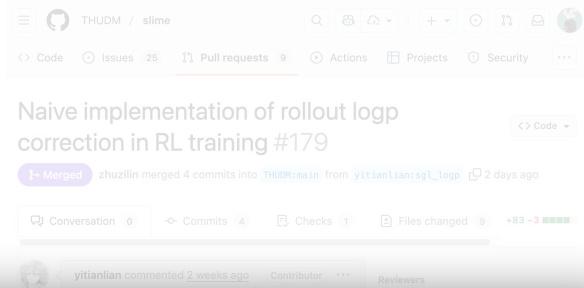
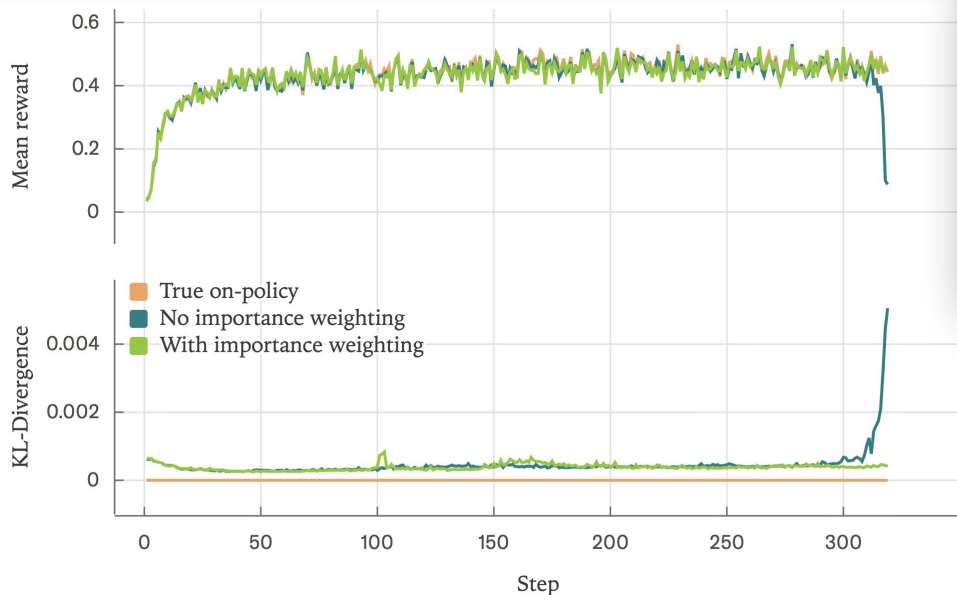
... Show more

- [News] OAT has verified and implement TIS. [GitHub] [Tweet from OAT]
- [News] SkyRL has integrated TIS. [GitHub] [Tweet from SkyRL]
- [News] REINFORCE++ verified TIS in Tool-Integrated-Reasoning (TIR) setting. [Blog]
- [News] OpenRLHF has integrated TIS. [GitHub]

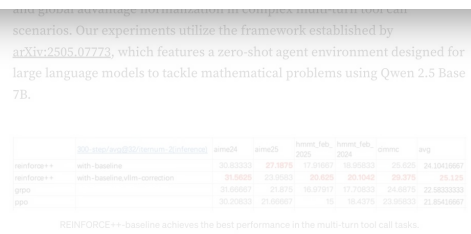
# Community Verification

## True on-policy RL

As researchers have noted, the different numerics between training and inference implicitly turns our on-policy RL into off-policy RL.



Configuration	Time (seconds)
vLLM default	26
Unoptimized Deterministic vLLM	55
+ Improved Attention Kernel	42



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

# Community Verification

## VLLM $\leftrightarrow$ Trainer logprob mismatch

Potential solutions to

**Correct, but off-policy** transformer logprobs  
**Incorrect, but on-policy** vllm logprobs

1. Change things/kernels so the logprobs do match (we didn't get this to work)
2. Calculate your logprobs in transformers, not vllm (our current approach, \$)
3. Use importance sampling

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

<https://fengyao.notion.site/off-policy-rl>



Slide credit: Michael Noukhovitch

Lambert | Olmo Thinks 27

# 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
- Additional Analyses

# Harvesting *Off-Policy* in Quantization

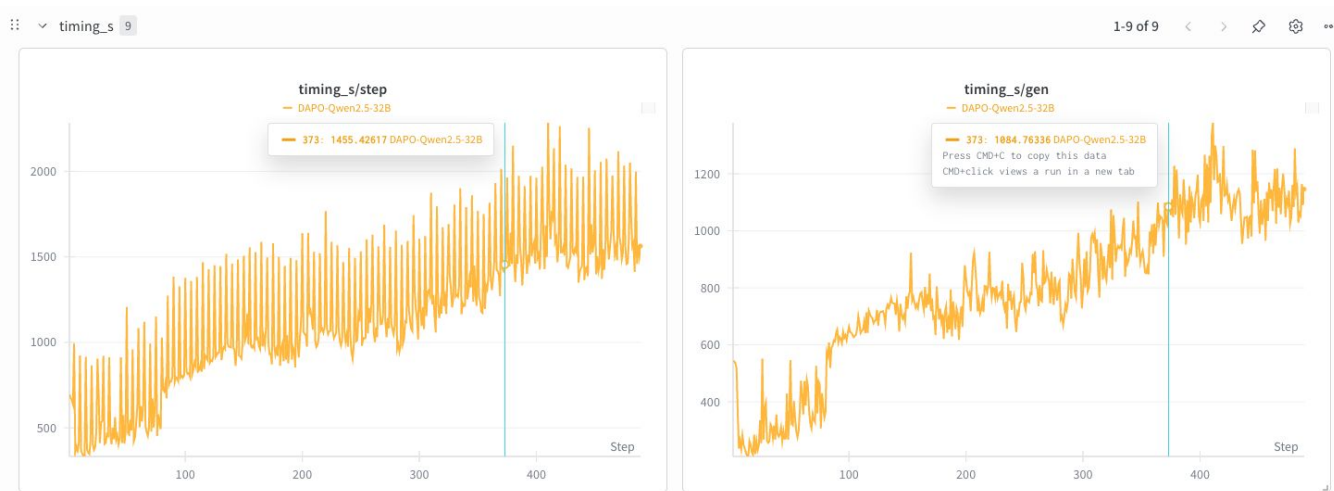
As TIS handles the gap, can we go even **further off-policy** for **speedup**?

# 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:

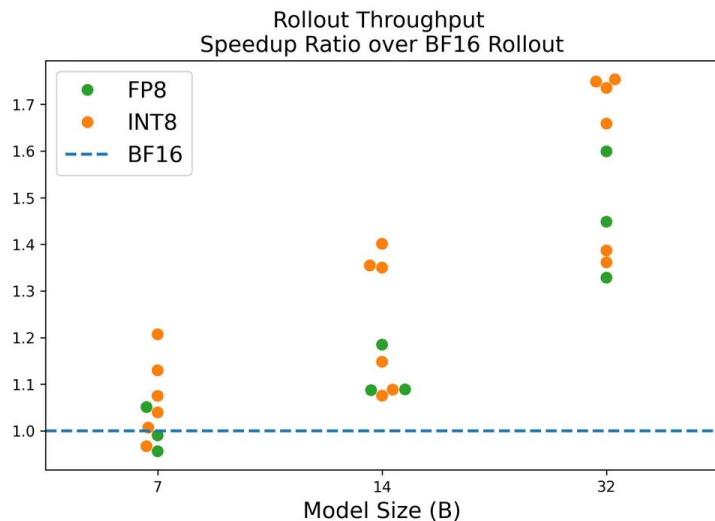
*In DAPO-32B setting, rollout takes up ~70% of the training time*



Quantization helps *speedup* **but** hurts *performance*

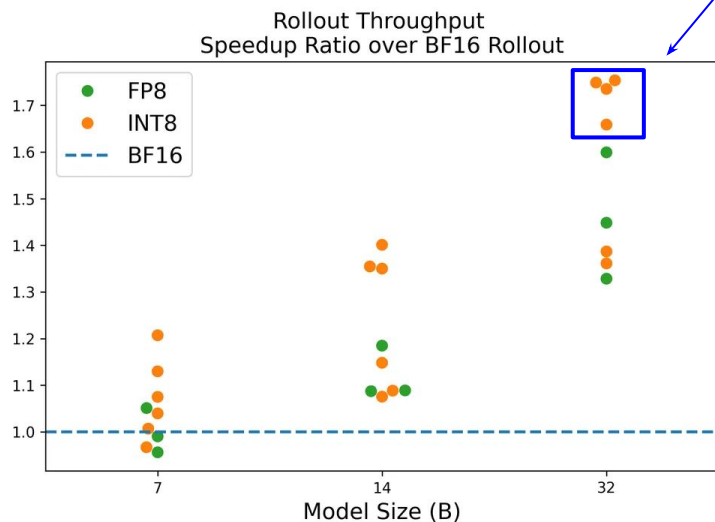
# Quantization helps *speedup* **but** hurts *performance*

Naively applying quantization can accelerate rollout speed



# Quantization helps *speedup* **but** hurts *performance*

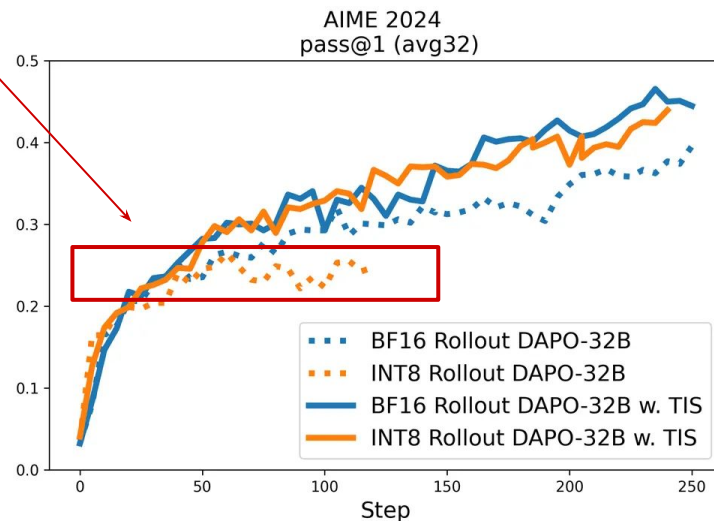
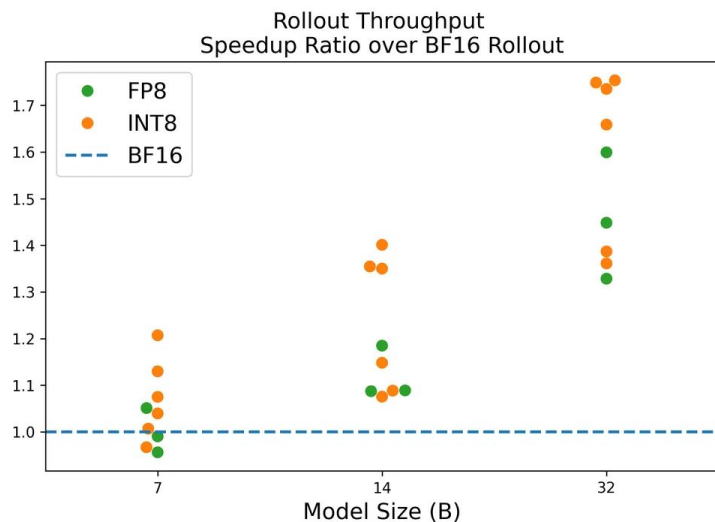
Naively applying quantization can accelerate rollout speed



# 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**

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

# FlashRL preserves performance with TIS

This can be expected, as quantization introduces more mismatch

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

## FlashRL fixes it with TIS

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

# FlashRL preserves performance with TIS

FlashRL fixes it with TIS

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

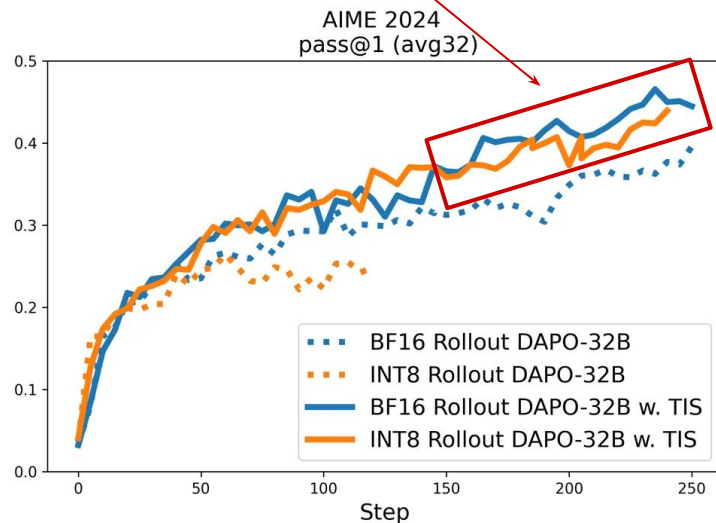
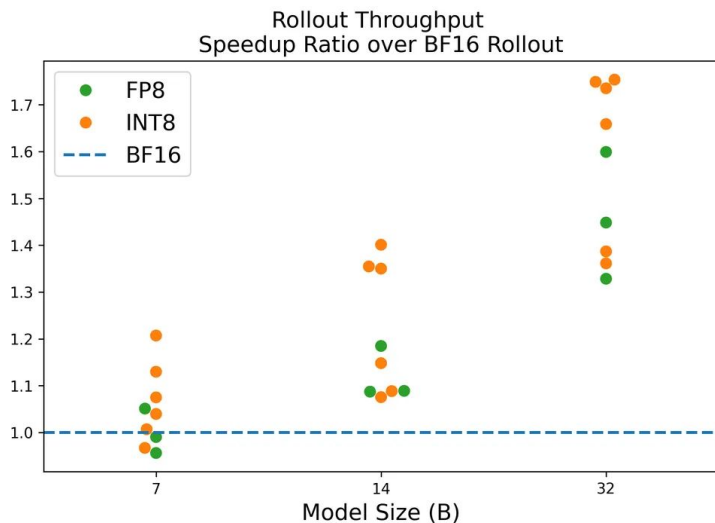
FlashRL is implemented as a PyPI package to patch vLLM

```
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*

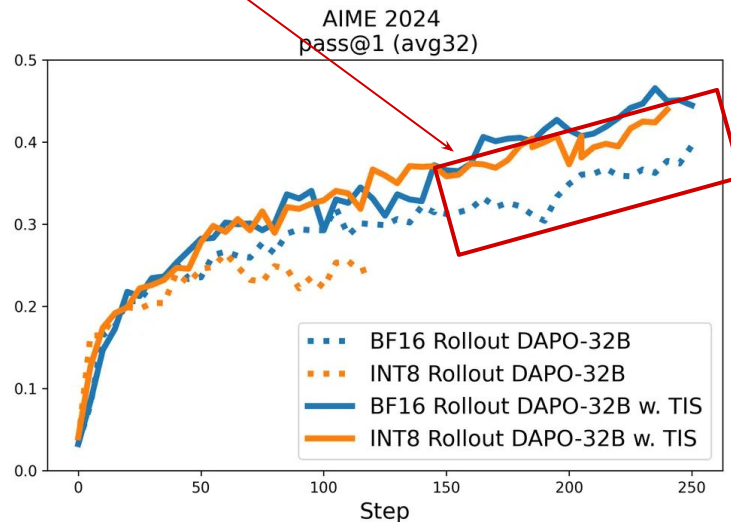
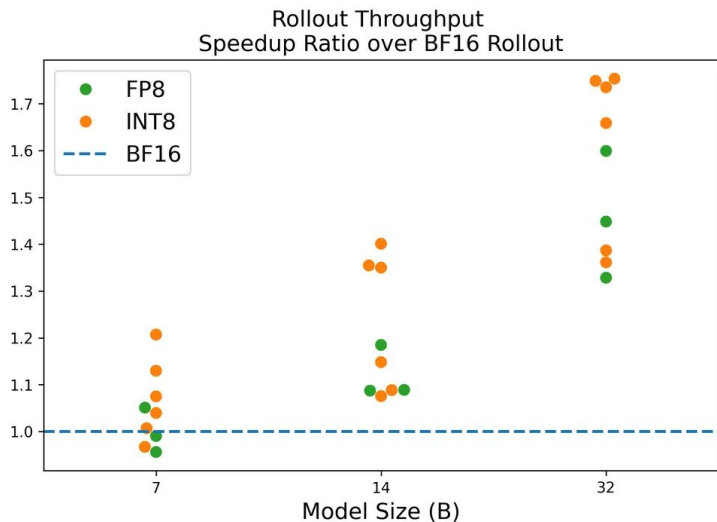


# FlashRL preserves performance with TIS

## DAPO 32B Setting

*Matches the performance of BF16 rollout with TIS*

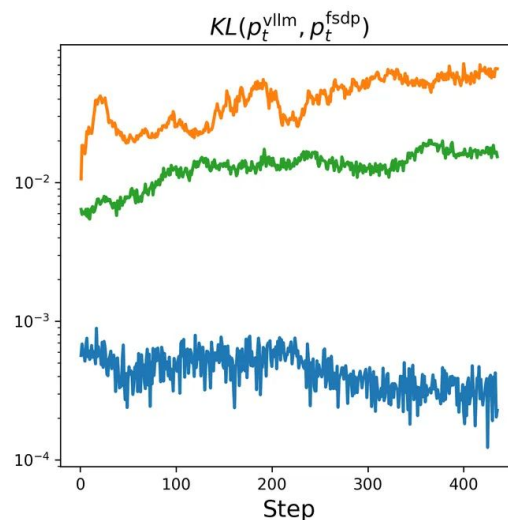
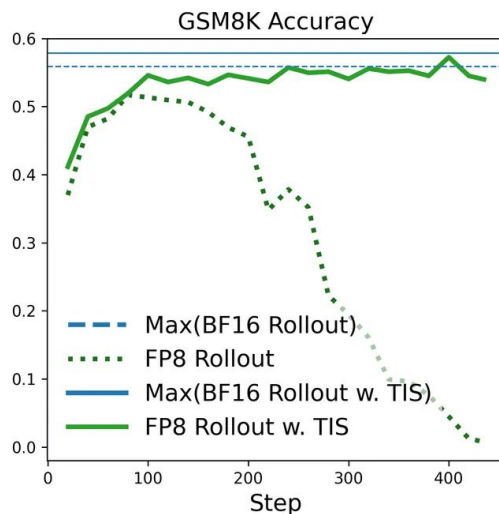
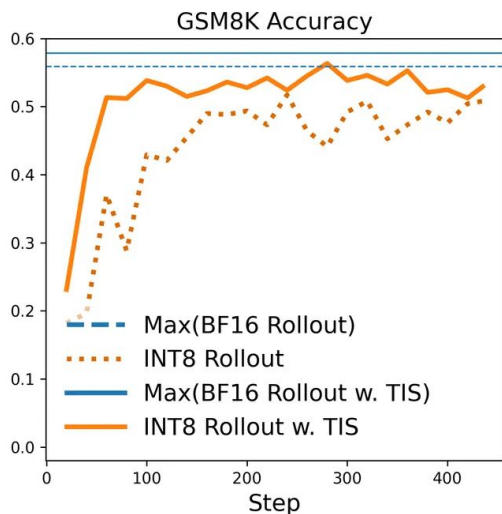
*Outperforms naive BF16 rollout (without TIS)*



# FlashRL preserves performance with TIS

## GSM8K 0.5B Setting

*TIS works both in INT8 and FP8 setting*



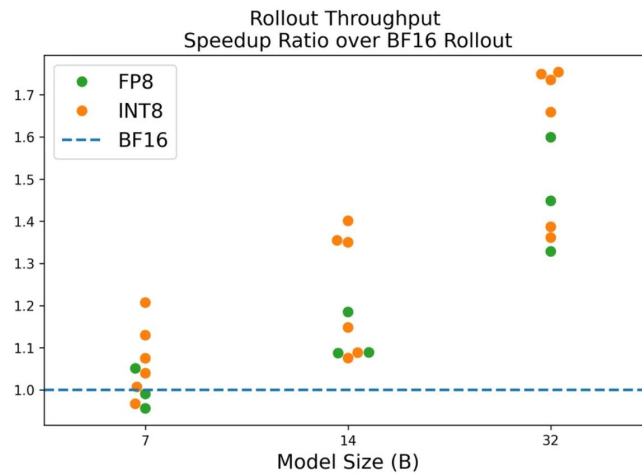
# More detailed analysis

## Rollout Speedup

# More detailed analysis

## Rollout Speedup

*Regular RL Setting*

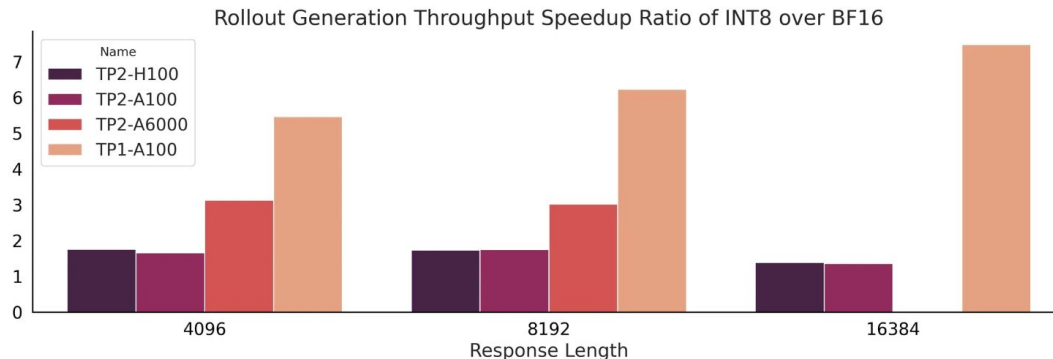
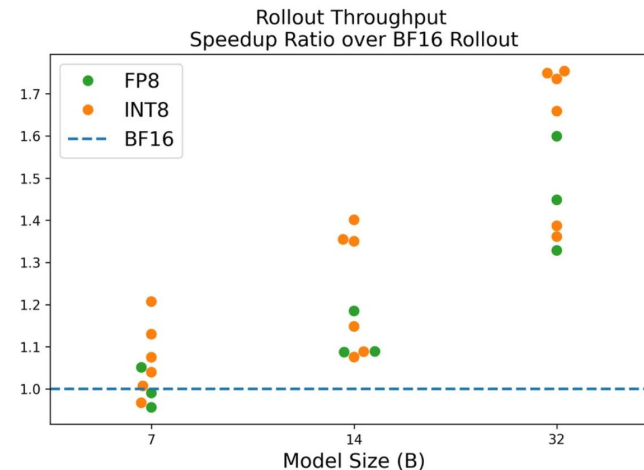


# More detailed analysis

## 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

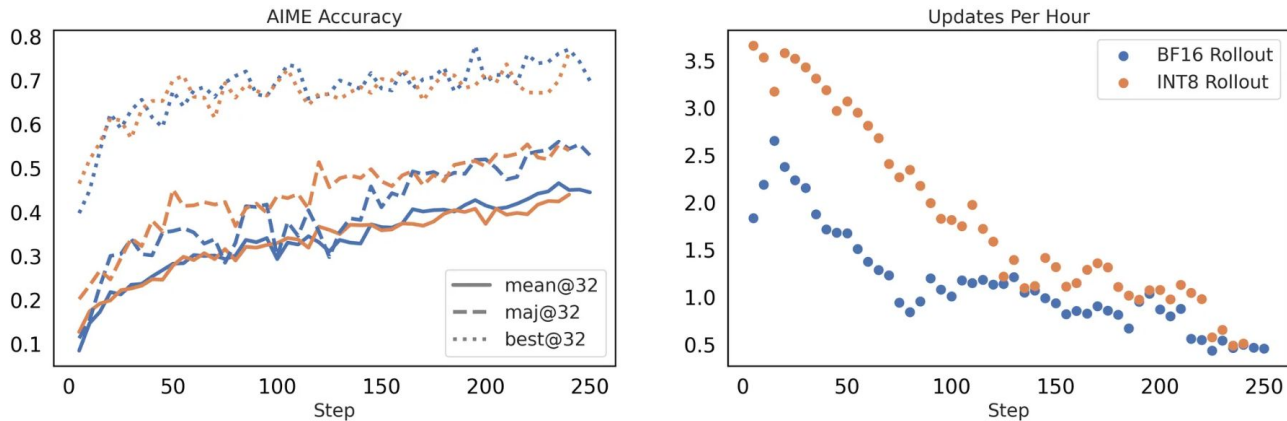
## End-to-End Speedup & Effectiveness

*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***

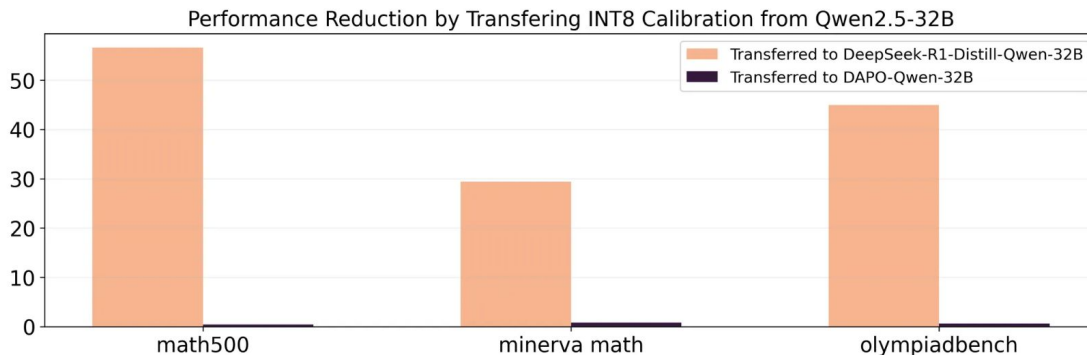
*calculate the calibration result once at the beginning of training and reuse it at every online step*

# 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 finetuned 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**
- Additional Analyses

# 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

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

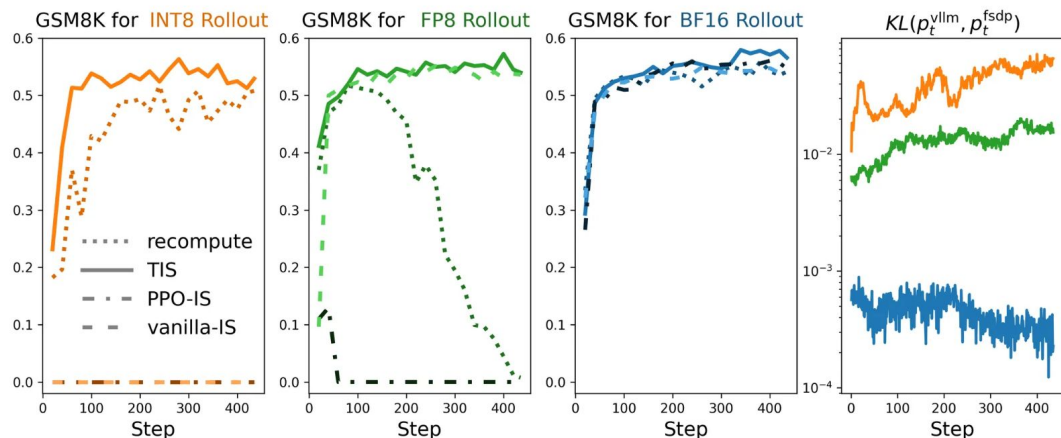
- TIS

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

# 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**

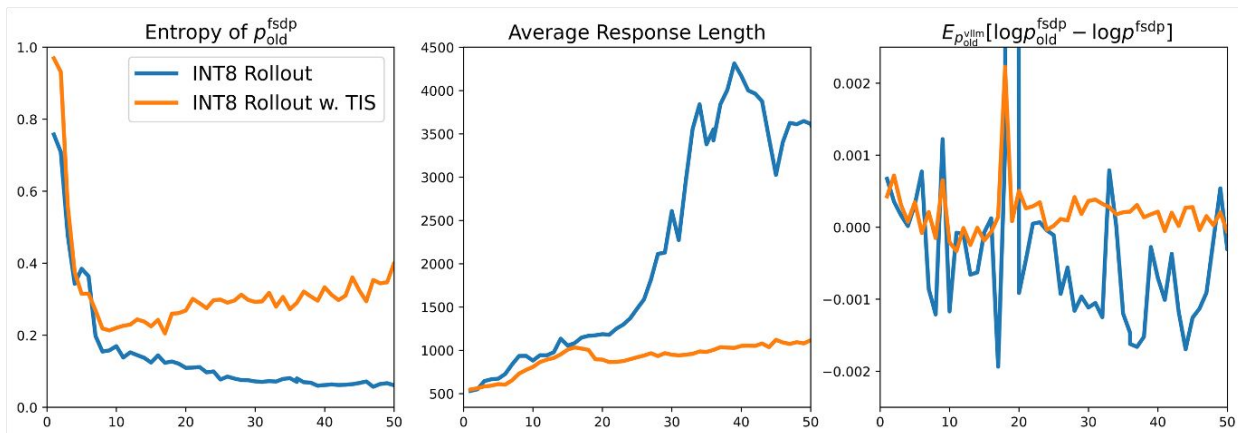
- The mismatch can lead to entropy collapse
  - Gradient computation vs. rollout generation

# Why Recompute fails

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

- **Recompute**

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

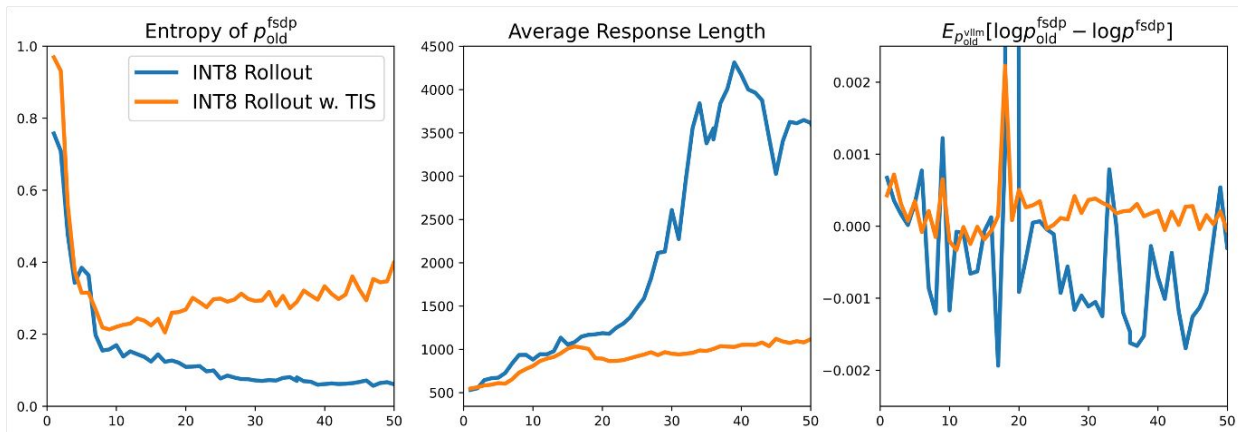


# Why Recompute fails

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

- **Recompute**

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



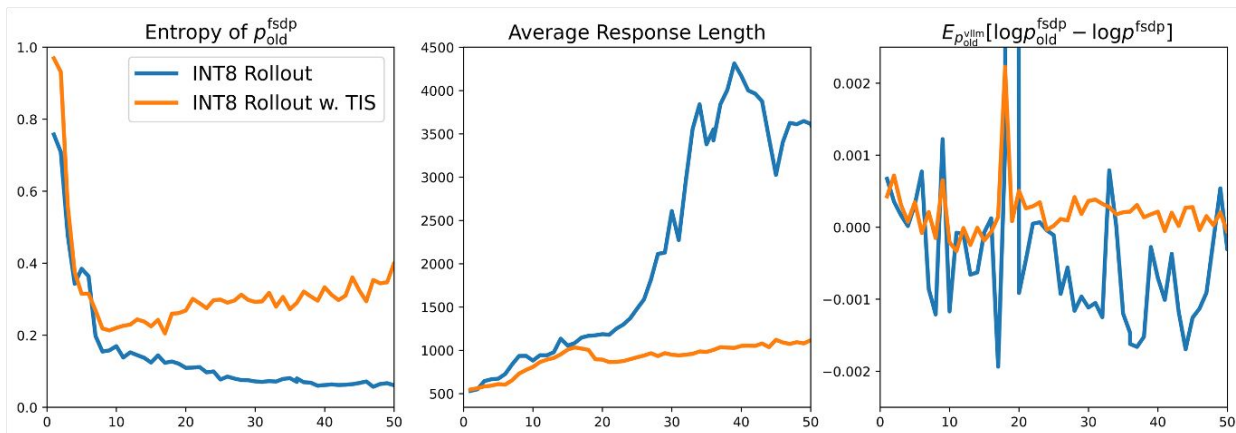
For  $a$  with  $A < 0$

# Why Recompute fails

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

- **Recompute**

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



For  $a$  with  $A < 0$

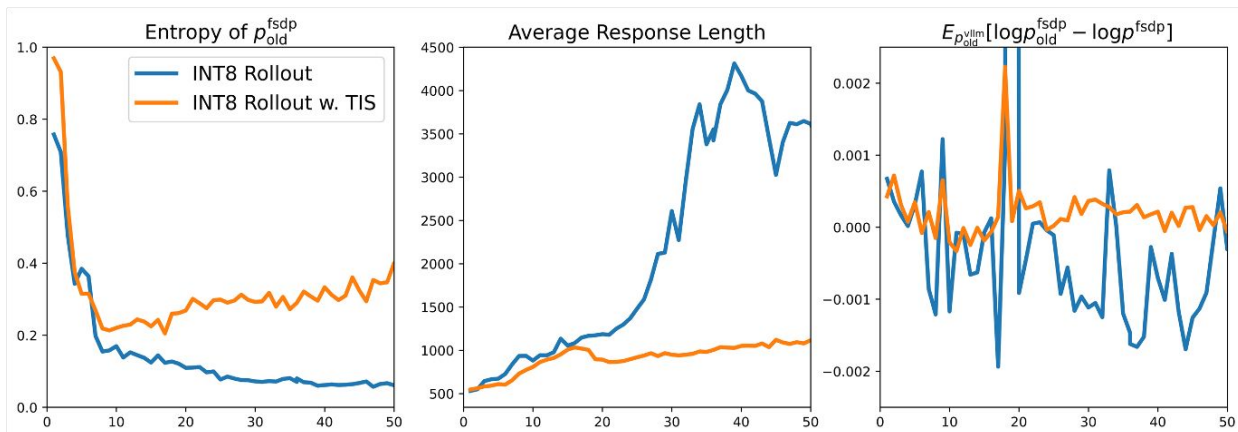
$\pi_{\text{fsdp}}(a)$  becomes smaller

# 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

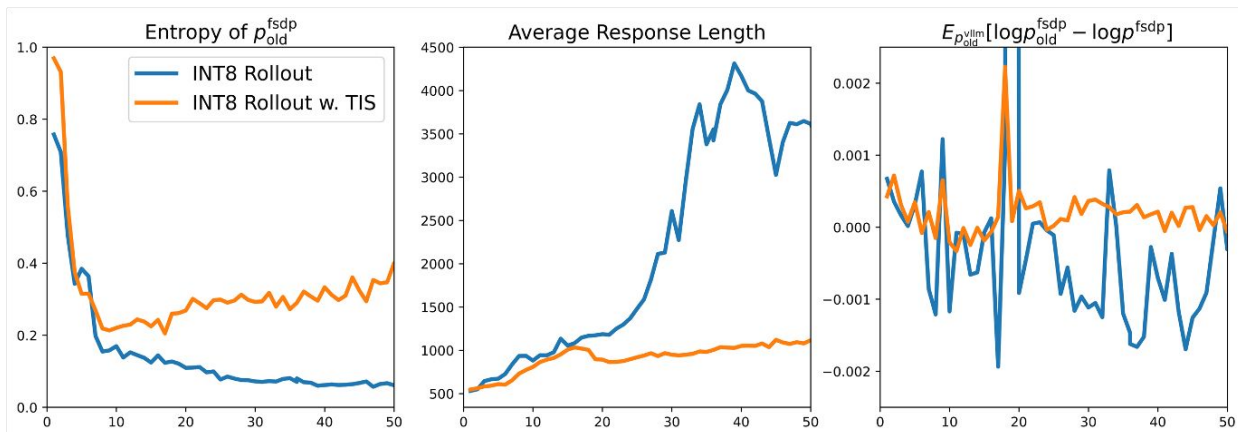
With a large gap w. INT8,  
 $\pi_{\text{vllm}}(a)$  stays the same

# 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

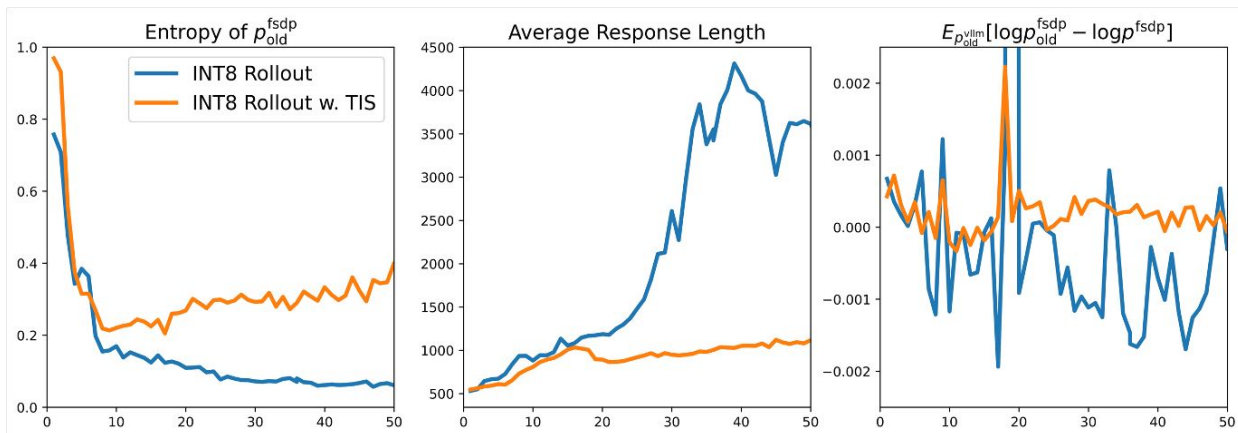
$\pi_{\text{fsdp}}(a)$  is over-penalized

# 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

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

- 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

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

- PPO-clip works differently than TIS

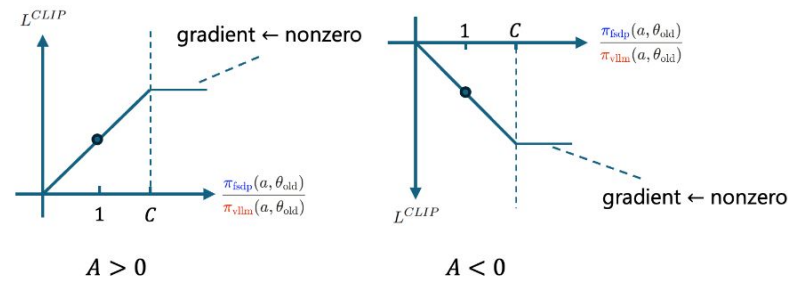
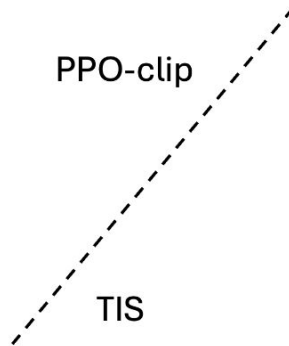
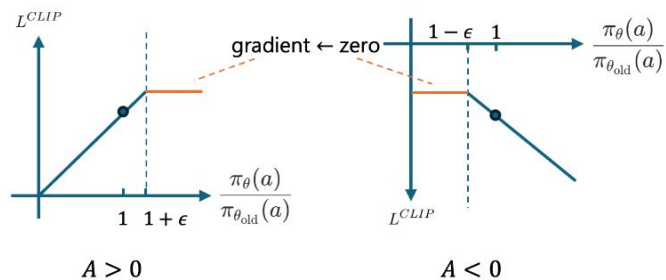
# Why PPO-IS fails

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

## • 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 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

constant factor, no BP through

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

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

Gradient noise

# 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
- **Additional Analyses**

# Factors Contributing to Mismatch

- Investigation Setup

# Factors Contributing to Mismatch

- **Investigation Setup**
  - **Model & Data:**
    - DAPO-32B / Polaris 7B
    - DAPO Training Set (first 512 samples)

# Factors Contributing to Mismatch

- Investigation Setup
  - Model & Data:
    - DAPO-32B / Polaris 7B
    - DAPO Training Set (first 512 samples)
  - Metrics:
    - Max Mismatch per response
    - Mean Mismatch per response

# Factors Contributing to Mismatch

- Investigation Setup

- Model & Data:

- DAPO-32B / Polaris 7B
    - DAPO Training Set (first 512 samples)

- Metrics:

- Max Mismatch per response

$$\max_{a \in \text{response}} |p_{\text{sampler}}(a) - p_{\text{learner}}(a)|$$

- Mean Mismatch per response

$$\frac{1}{|\text{response}|} \sum_{a \in \text{response}} |p_{\text{sampler}}(a) - p_{\text{learner}}(a)|$$

# Factors Contributing to Mismatch

- Investigation Setup

- Model & Data:

- DAPO-32B / Polaris 7B
- DAPO Training Set (first 512 samples)

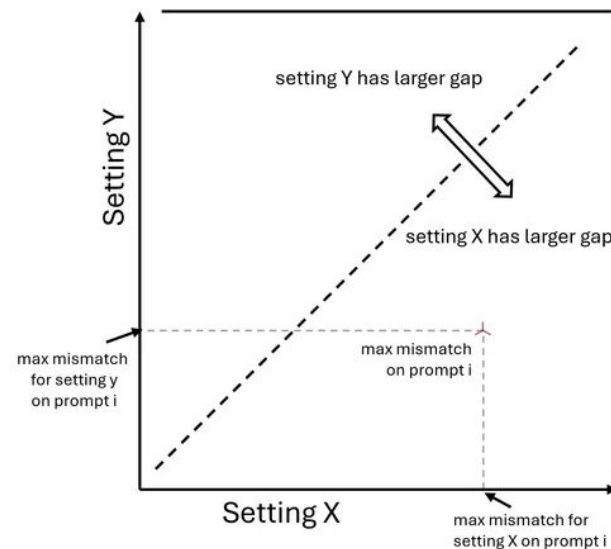
- Metrics:

- Max Mismatch per response

$$\max_{a \in \text{response}} |p_{\text{sampler}}(a) - p_{\text{learner}}(a)|$$

- Mean Mismatch per response

$$\frac{1}{|\text{response}|} \sum_{a \in \text{response}} |p_{\text{sampler}}(a) - p_{\text{learner}}(a)|$$

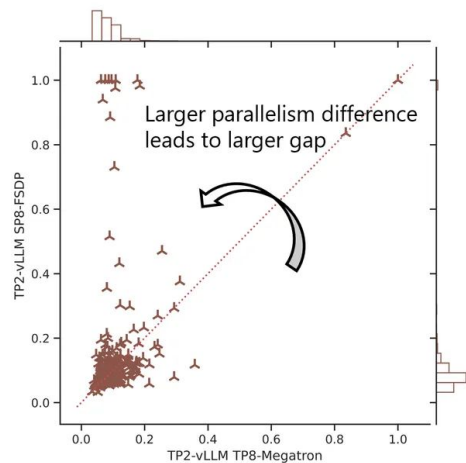


# Factors Contributing to Mismatch

- Larger **Parallelism** Difference, Larger **Max Gap**

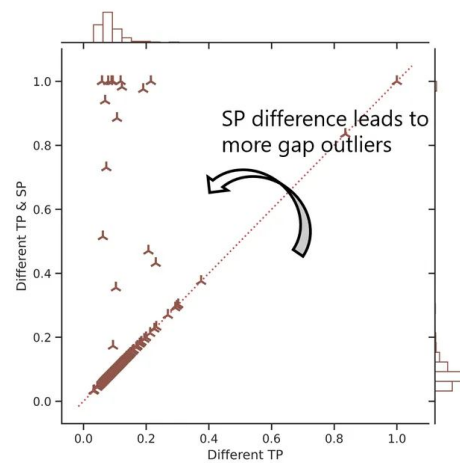
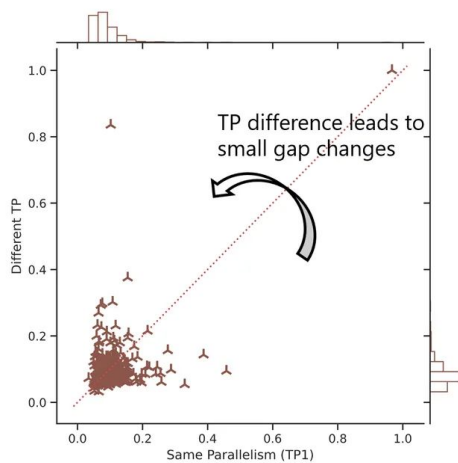
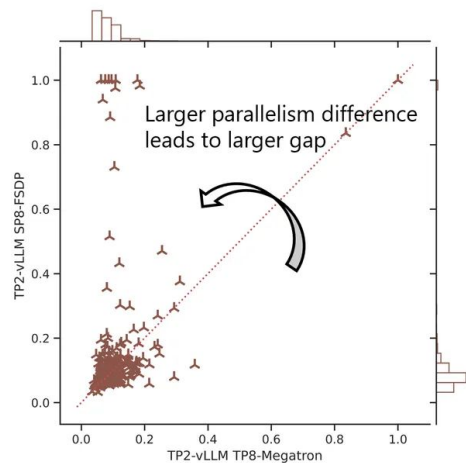
# Factors Contributing to Mismatch

- Larger **Parallelism** Difference, Larger **Max Gap**



# Factors Contributing to Mismatch

- Larger **Parallelism** Difference, Larger **Max Gap**

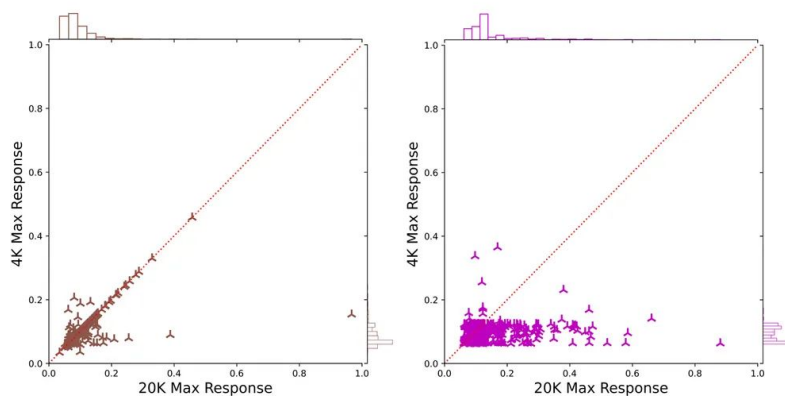


# Factors Contributing to Mismatch

- Longer **Response Length**, Larger **Max Gap**

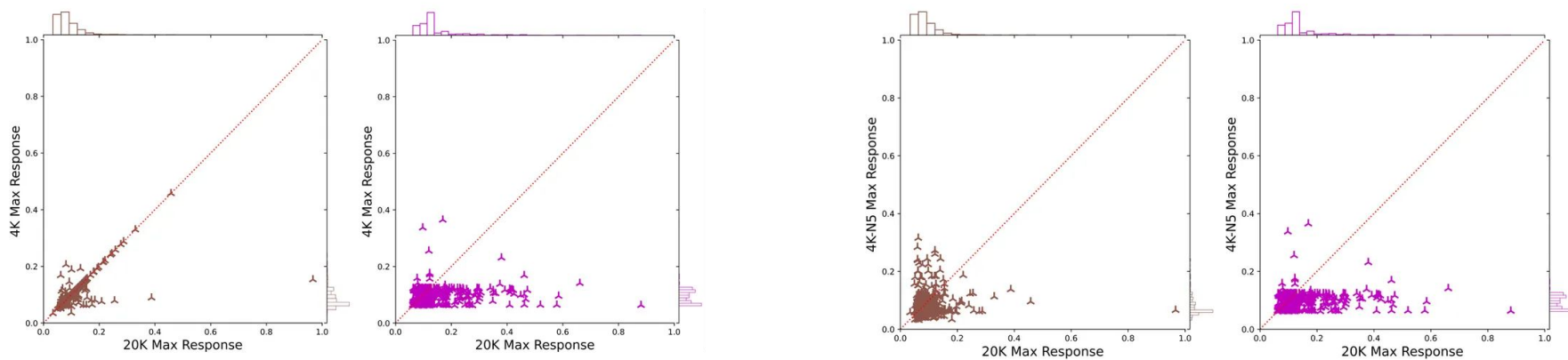
# Factors Contributing to Mismatch

- Longer **Response Length**, Larger **Max Gap**



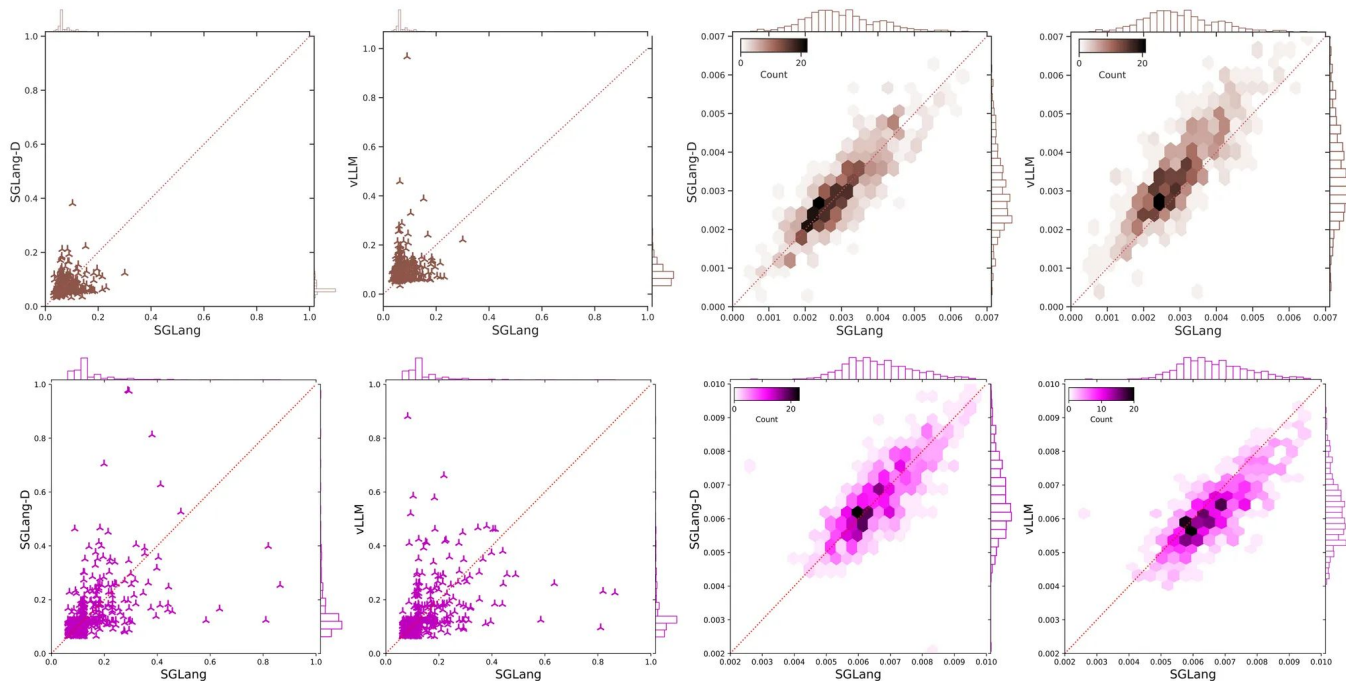
# Factors Contributing to Mismatch

- Longer **Response Length**, Larger **Max Gap**



# Factors Contributing to Mismatch

- Altering **Sampler** Alone, Gap Still There



**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**

# What's beyond?

- The gap can be amplified in MoE RL
  - Dynamic Routing
  - Specially Optimized Kernels
    - e.g., GRPO (token-level)
    - GSPO (sequence-level)
- 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/>

April 06, 2026 – Presented at UVA