Latency-Conditioned Selection Bias in RLHF

How wall-clock cutoffs shape which reasoning trajectories train a model

Prakul Sunil Hiremath
Department of Computer Science
Visvesvaraya Technological University
Belagavi, Karnataka, India
prakulhiremath@vtu.ac.in

PROJECT PAGE

Outline

1. **Setting:** asynchronous partial rollouts (APR)
2. **Mechanism:** a clock-driven length filter
3. **Theory:** attenuation and preference flip
4. **Evidence:** serving, PPO, controlled intervention
5. **Diagnosis:** Effective Gradient Information (EGI)
6. **Correction:** Selection-Probability Weighting (SPW)
7. **Results** on LLaMA-3-8B and other tasks
8. **Scope, limits, takeaways**

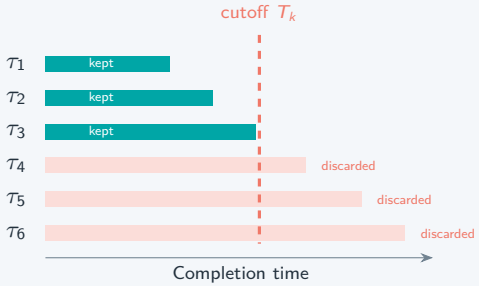
Central message

Generation time grows with length. Discard-style APR selects by time, so it can systematically under-train longer reasoning chains.

The setting: asynchronous RLHF

Generate many responses in parallel

- A policy samples N trajectories in parallel.
- The learner starts as soon as the k-th rollout completes.
- The remaining $N - k$ rollouts are *discarded* (discard-style APR).
- Truncation ratio $(N - k)/N$ ranges from 0 (synchronous) to 0.5 in our settings.



Why use APR?

It keeps the learner supplied with completed samples without waiting for the slowest worker, yielding up to 2.47× throughput in our GSM8K runs.

The hidden question: which trajectories enter the update?

A wall-clock cutoff is also a sample filter

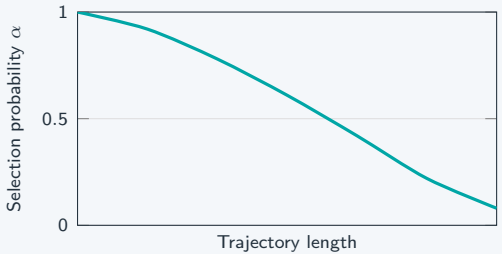
- Autoregressive decoding requires one forward pass per output token.
- Longer outputs therefore take longer to finish.
- Late trajectories are less likely to be retained.

Potential consequence

The batch over-represents short trajectories, even when longer reasoning chains are sampled by the policy and receive higher reward.

Not classical off-policy bias

All retained rollouts come from the current policy π_θ. Policy-ratio importance weights (e.g., V-trace) equal 1 and cannot correct this selection filter.



Conceptual illustration of length-dependent selection; not measured data.

Formalising latency-conditioned selection

The retained trajectories follow a reweighted policy distribution:

$$p_{\mathrm{APR}}(\tau) = \frac{p_\theta(\tau)\,\alpha(\tau)}{Z}$$

$p_\theta(\tau)$

Probability that the current policy generates trajectory τ.

$\alpha(\tau)$

Probability that τ finishes before the cutoff:

$$\alpha(\tau) = \Pr[L(\tau) \leq T_k].$$

Z

Normalising constant:

$$Z = \mathbb{E}_{p_\theta}[\alpha(\tau)].$$

Assumption: latency–length monotonicity

Shorter trajectories stochastically dominate longer ones in completion time. Thus, α is non-increasing in trajectory length.

Key distinction

The selection factor is determined by the serving system and completion time, not by reward.

Theory: selection attenuates expected reward

$$\mathbb{E}_{p_{\mathrm{APR}}}[R] = \mathbb{E}_{p_\theta}[R] + \frac{\mathrm{Cov}_{p_\theta}(R, \alpha)}{Z}$$

When reward rises with length

- Longer trajectories can receive higher reward.
- Their probability of surviving the cutoff falls.
- Reward and selection probability then covary negatively.

Under the stated monotonicity assumptions, if $R = r(|\tau|)$ with r non-decreasing, then $\mathbb{E}_{p_{\mathrm{APR}}}[R] \leq \mathbb{E}_{p_\theta}[R]$. This is a reward-attenuation result; gradient direction is analysed separately in the branching MDP on the next slide.

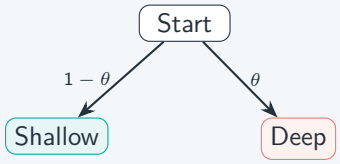
Attenuation condition

$$\mathrm{Cov}_{p_\theta}(R, \alpha) < 0 \implies \mathbb{E}_{p_{\mathrm{APR}}}[R] < \mathbb{E}_{p_\theta}[R].$$

Interpretation

APR optimises a structurally different effective objective, not merely a noisier estimate of the same one.

A minimal branching MDP: the preference can flip



The deep path is better, but slower

The policy picks the shallow path with probability $1-\theta$ and the deep path with probability θ. The on-policy gradient is $r_d - r_s > 0$. The cutoff makes the deep path less likely to enter the update.

Bias and preference-flip condition

$$\mathrm{LCSB}(\theta) = -\frac{(\alpha_s - \alpha_d)\,[(1-\theta)r_d + \theta r_s]}{(1-\theta)\alpha_s + \theta\alpha_d} \leq 0, \qquad \boxed{\frac{\alpha_s}{\alpha_d} > \frac{r_d}{r_s} \iff \nabla_\theta J_{\mathrm{APR}} < 0}$$

The update can move away from the higher-reward, longer path.

Evidence 1: the length filter exists in a real serving system

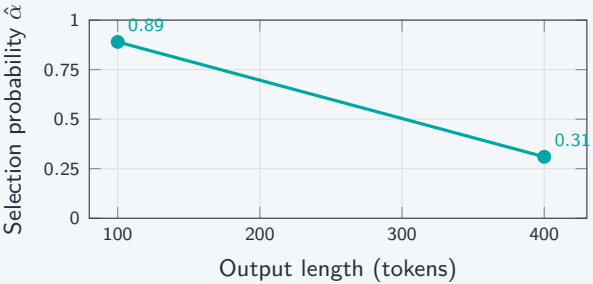
vLLM measurements

- LLaMA-3-8B; 12,000 rollouts
- HH-RLHF, MBPP, GSM8K prompts
- Concurrency $N \in \{8, 16, 32\}$; KV-cache 40–95%

Length explains latency

$$r^2 = 0.918, \qquad p < 10^{-6}.$$

91.8% of latency variance is explained by output length.



2.9×

selection disparity

Estimated $\hat{\alpha}$ at $k/N = 0.67$: 0.89 at 100 tokens vs. 0.31 at 400 tokens.

Evidence 2: truncation and horizon change what PPO learns

GPT-2 (124M) PPO

- 5,000 gradient steps, 10 seeds
- Exp. 1: GSM8K, vary $k/N \in \{1.0, 0.9, 0.8, 0.67, 0.5\}$
- Exp. 2: fix $k/N = 0.67$; vary horizon over HH-RLHF ($\approx$87 tok), MBPP ($\approx$243), GSM8K ($\approx$412)

Observed pattern

Degradation grows monotonically with truncation and with task horizon; short-horizon tasks are largely unaffected.

Throughput-matched control

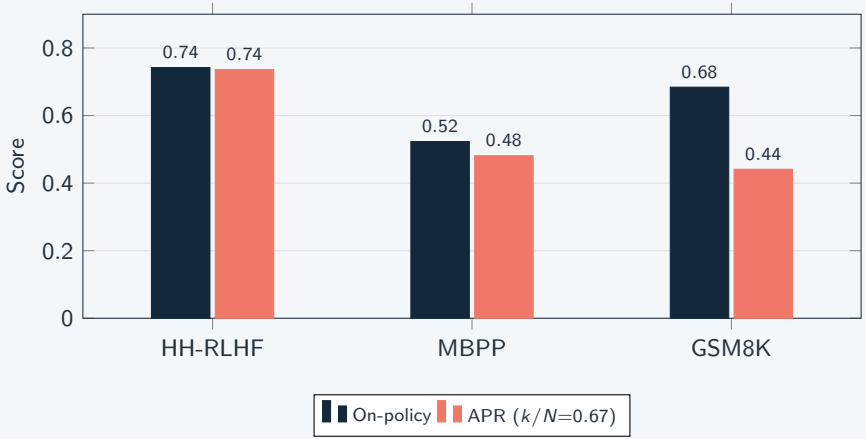
- Synchronous $N = k = 96$ vs. $N = k = 64$: within 1%.
- Both about 23 points above APR at $k/N = 0.67$.

Rules out a batch-size confound

APR's smaller training batch ($k = 64$ of $N = 96$) does not explain the gap.

The learned policy becomes shallower, not merely noisier.

Evidence 2 (cont.): training curves and horizon scaling

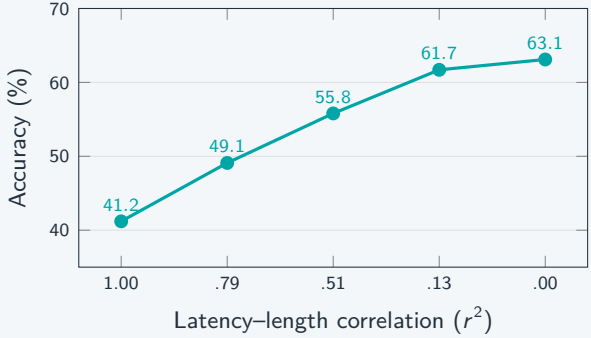


HH-RLHF and MBPP: GPT-2; GSM8K: LLaMA-3-8B. Metrics are reward, pass@1, and accuracy, respectively.

Evidence 3: a controlled latency intervention

Change only the latency–length correlation

On-policy training ($N = k = 64$), GSM8K / GPT-2, five seeds; latency injected as $c\,|\tau| + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.



Accuracy recovers as correlation weakens

- $r^2 = 1.00$: 41.2%
- $r^2 = 0.51$: 55.8%
- $r^2 = 0.00$: 63.1%

Output length

Mean output length rises from 188 to 409 tokens across these conditions.

Since only the length–latency correlation varies, this is controlled evidence that it drives the degradation.

Detecting the loss: Effective Gradient Information

$$\mathrm{EGI} = \frac{\|\mathbb{E}[\nabla J]\|^2}{\mathrm{Tr}(\mathrm{Cov}(\nabla J)) \cdot C}$$

What it measures

- Gradient signal relative to gradient variability.
- Normalised by total generated tokens C.
- Estimated via a fixed random projection to $\mathbb{R}^{512}$; 100-step EMA.

What it detects

Early warning

An alarm at $0.6\times$ baseline EGI fired roughly 600–1,700 gradient steps before reward curves clearly diverged (MBPP $\approx$600, GSM8K $\approx$1,700).

Reward curves can reveal damage only after gradient information has already degraded.

EGI is empirically, not provably, an early-warning signal and is not specific to LCSB; read it alongside the output-length distribution.

Correcting selection: Selection-Probability Weighting

Inverse-probability weighting

$$w_i = \min\left(\frac{1}{\widehat{\alpha}(\tau_i)},\, w_{\max}\right)$$

Weighted policy-gradient estimate

$$\nabla J_{\mathrm{SPW}} = \frac{\sum_{i=1}^{k} w_i \,\nabla_\theta \log \pi_\theta(\tau_i) R(\tau_i)}{\sum_{i=1}^{k} w_i}.$$

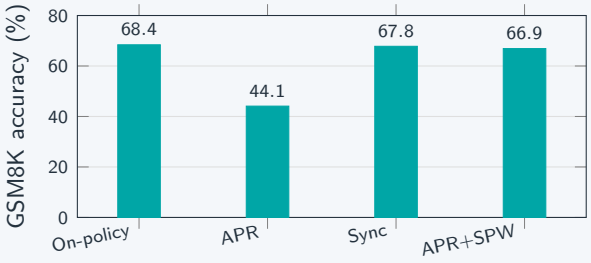
Intuition

- Inverse-probability-of-censoring weighting.
- Estimate $\hat{\alpha}$ using an online per-length-bin empirical CDF.
- Up-weight trajectories less likely to be retained; clip weights to limit variance.

When it is useful

- Consistent under the overlap condition $\alpha(\tau) > 0$.
- Adds less than 0.3% iteration-time overhead.
- $\mathrm{ESS}/n > 0.3$ keeps clipped MSE within 10% of unclipped MSE.

Does the correction work? LLaMA-3-8B on GSM8K



Values and standard deviations are reported over three seeds.

What changes with SPW?

- Accuracy rises from 44.1% to 66.9%.
- Mean chain length recovers from 193 to 402 tokens.
- Throughput remains 2.38× on-policy.

Comparison

On-policy	$68.4 \pm 1.8\%$
APR	$44.1 \pm 2.6\%$
Synchronous	$67.8 \pm 1.6\%$
APR + SPW	$66.9 \pm 1.9\%$

Relative accuracy: **97.8%** of on-policy accuracy.

Three seeds; LLaMA-3-8B, GSM8K, 500 held-out prompts. SPW restores most of the accuracy and chain length lost under APR.

Task dependence and limits

Other reported tasks (GPT-2 setting)

	On-policy	APR	APR+SPW
HH-RLHF reward	.742	.736	.739
MBPP pass@1	.523	.481	.516

HH-RLHF differences are within noise (short horizon). MBPP shows a measurable gap and strong, partial recovery by SPW.

Scope and limitations

- Discard-style APR with PPO only.
- Recycle-style APR, GRPO, and verbosity-penalised tasks are not studied.
- LLaMA-3-8B results use three seeds; serving validated on vLLM only.
- At $k/N \leq 0.4$ and horizon ≥ 500 tokens, $\mathrm{ESS}/n < 0.15$.

Practical boundary

When overlap is weak, reducing truncation is safer than relying on large importance weights.

Practical recommendations

For discard-style APR with mean horizon $\gtrsim 150$ tokens:

1. **Measure** the latency–length correlation in your serving stack.
2. **Monitor** EGI alongside tokens per second.
3. **Apply SPW** when EGI falls below its task-specific threshold.
4. **Truncate aggressively** ($k/N < 0.5$) only with SPW and ESS monitoring; reduce truncation when $\mathrm{ESS}/n < 0.15$.

When APR is safe

Mean horizon $\lesssim 100$ tokens and $k/N \geq 0.8$: the selection gap is small. Prior APR evaluations lived in this regime; our findings complement them.

When it bites

Mean horizon ≥ 200 tokens and $k/N < 0.9$, with rewards that favour longer outputs.

Limitations & Open Questions

What this work does not establish

- **Algorithmic scope:** We study discard-style APR with PPO. Recycle-style APR and GRPO remain untested.
- **Experimental scale:** The LLaMA-3-8B GSM8K evaluation uses three seeds. Broader evaluation is needed.
- **Overlap limits:** SPW requires non-zero retention probability. Severe truncation can make correction unreliable.
- **Task dependence:** Effects vary across tasks; verbosity-penalised settings remain unstudied.

Questions for future work

- **Beyond PPO:** Does latency-conditioned selection affect GRPO and other policy-gradient methods?
- **Beyond length:** Which system variables besides sequence length affect trajectory retention?
- **Beyond correction:** Can scheduling or adaptive rollout budgets prevent selection bias?
- **Beyond current scale:** How does the effect change with larger models, longer horizons, and multi-node serving?

Central open question

How can we preserve the learning signal while retaining the throughput benefits of asynchronous training?

Takeaways

1. **Asynchronous selection changes the training distribution.**
   A wall-clock cutoff makes long trajectories less likely to enter the update.
2. **The effect can change the learning direction.**
   Theory gives reward attenuation under monotonicity and a preference flip in a branching MDP.
3. **The mechanism appears in serving and training.**
   Length explains 91.8% of latency variance; a controlled intervention links weaker correlation to better accuracy.
4. **Measure and correct it.**
   EGI gives an empirical early warning; APR+SPW reaches 97.8% of on-policy accuracy on LLaMA-3-8B when overlap is adequate.

Who gets to train the model is partly decided by when they finish.

Thank you

Backup: SPW overlap and effective sample size

Effective sample size

$$\mathrm{ESS} = \frac{\left(\sum_i w_i\right)^2}{\sum_i w_i^2}$$

$\mathrm{ESS}/n > 0.3$ keeps clipped MSE within 10% of the unclipped estimator.

k/N	87 tok	243 tok	412 tok	600 tok
0.90	.81	.67	.54	.41
0.80	.73	.58	.43	.31
0.67	.61	.44	.33	.21
0.50	.44	.29	.16	.08

ESS/n by truncation level and mean horizon.

Practical implication

In the low-overlap regime, SPW is insufficient; reducing truncation is safer.

Backup: KL regularisation does not remove LCSB

Configuration	Accuracy (%)	Length (tok)	EGI
On-policy ($\beta = 0.02$)	68.4	421	1.00
APR ($\beta = 0.02$)	44.1	193	0.42
APR ($\beta = 0.05$)	49.7	228	0.56
APR ($\beta = 0.10$)	54.2	267	0.68
APR ($\beta = 0.20$)	58.3	305	0.79
APR+SPW ($\beta = 0.02$)	66.9	402	0.95

Larger β reduces the degradation but does not close the gap; SPW does (LLaMA-3-8B, GSM8K, $k/N = 0.67$, 3 seeds).

Backup: experimental details

- Serving validation: vLLM v0.4.1 with LLaMA-3-8B; 12,000 rollouts; 2× Tesla T4.
- PPO study: GPT-2 (124M), 5,000 gradient steps, $k = 64$, $N = 96$, 10 seeds.
- PPO: lr 3×10^{-5}, $\beta = 0.02$, clip $\varepsilon = 0.2$, GAE $\lambda = 0.95$, $\gamma = 1$.
- Controlled latency injection: GSM8K / GPT-2, five seeds.
- LLaMA-3-8B GSM8K comparison: three seeds; 500 held-out prompts.
- SPW: 10 length bins, $w_{\max} = 10$, 50-iteration warm-up; overhead $< 0.3\%$.
- EGI: projection $m = 512$, minibatches of 32 trajectories, 100-step EMA.

Backup: key equations

$$p_{\mathrm{APR}}(\tau) = \frac{p_\theta(\tau)\alpha(\tau)}{Z}, \quad \alpha(\tau) = \Pr[L(\tau) \le T_k].$$

$$\mathbb{E}_{p_{\mathrm{APR}}}[R] = \mathbb{E}_{p_\theta}[R] + \frac{\mathrm{Cov}_{p_\theta}(R, \alpha)}{Z}.$$

$$\nabla_\theta J_{\mathrm{APR}} = \frac{\alpha_d r_d - \alpha_s r_s}{(1-\theta)\alpha_s + \theta\alpha_d}, \qquad \frac{\alpha_s}{\alpha_d} > \frac{r_d}{r_s} \iff \nabla_\theta J_{\mathrm{APR}} < 0.$$

$$w_i = \min\left(\frac{1}{\widehat{\alpha}(\tau_i)}, w_{\max}\right), \qquad \mathrm{EGI} = \frac{\|\mathbb{E}[\nabla J]\|^2}{\mathrm{Tr}(\mathrm{Cov}(\nabla J)) \cdot C}.$$