RL-IF: Tracking Non-local Training Data Influence for Online Reinforcement Learning

Shixuan Liu[*], **Cheng Tang**[*], **Yuzheng Hu**[*], **Fan Wu, Han Zhao, Jiaqi W. Ma**

University of Illinois Urbana-Champaign

{shixuanl,chengt6,yh46,fanw6,hanzhao,jiaqima}@illinois.edu

Abstract

Online reinforcement learning (RL) has achieved significant success in decision-making tasks, but modern online RL remains highly sensitive to the quality of training experience. Understanding the contribution of individual training instances is therefore central to data-centric RL, with applications ranging from experience selection to policy interpretation. Existing data attribution methods for online RL primarily focus on local influence within a single training round, overlooking the non-local data influence across multiple training rounds. In this work, we formalize data attribution in online RL through a trajectory-level leave-one-out quantity, *RL-LOO*, which measures how one training sample influences a downstream target after subsequent online updates have taken place. We then derive *RL-IF*, a first-order estimator that propagates data influence through both optimization effects and policy-induced sampling effects, and provide theoretical guarantees on its approximation accuracy under smoothness assumptions. Our analysis further disentangles these two effects and shows both theoretically and empirically that the full RL-IF estimator can often be well approximated using the optimization effect alone. Building on this observation, we develop **RINSE**, a practical data filtering method for online RL. Experiments on standard control benchmarks and an RLHF-style toxicity-mitigation setting show that RINSE improves training efficiency and final performance over standard PPO and local attribution baselines. These results suggest that RL-IF provides a practical and principled framework for understanding and improving online RL.

1 Introduction

Online reinforcement learning (RL) has become a central paradigm for learning decision-making policies through repeated interaction with an environment, from robotic control [Todorov et al., 2012, Andrychowicz et al., 2020] to the alignment and enhancement of large language models (LLMs) [Ouyang et al., 2022, Shao et al., 2024]. Despite this success, modern online RL remains highly sensitive to the quality of training experience: training often requires millions of interactions, while the quality of collected experience can vary substantially across time and trajectories, leading to well-known challenges such as sample inefficiency, high variance, and training instability [Yu, 2018, Henderson et al., 2018, Dulac-Arnold et al., 2019]. Hence, understanding the contribution of individual training instances from these experiences is fundamental to data-centric studies of RL and has broad implications for applications such as data selection [Hu et al., 2026, Zhu et al., 2026], experience replay [Schaul et al., 2016], or interpretation [Hu et al., 2025].

Data attribution offers a natural lens for understanding the role of individual training instances. In supervised learning, data attribution aims to identify the training examples that most strongly drive

[*]Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

model behavior [Koh and Liang, 2017, Deng et al., 2025], with applications in model debugging [Koh and Liang, 2017], mislabeled data detection [Pruthi et al., 2020], data valuation [Ghorbani and Zou, 2019, Jia et al., 2019], and data selection [Xia et al., 2024]. Recently, Hu et al. [2025] introduce data attribution to online RL, and develop a local attribution framework for PPO. The proposed framework measures the influence of rollout records on a local policy update within a round, and shows that such local scores can identify harmful training time steps and improve sample efficiency through data filtering.

However, the existing attribution methods are not yet principled and capture only a partial notion of data influence in online RL. First, they mainly rely on heuristic influence estimation, such as gradient dot products, which are intuitive and scalable, but do not fully account for the training dynamics or correspond to a clear counterfactual notion of data value [Koh and Liang, 2017]. Second, the scope of existing attribution methods is restricted to a single training round, leaving the influence of training data across rounds largely unexplored. Characterizing such inter-round influence is more challenging in online RL because the effect of a single training occurrence can propagate across future rounds through subsequent optimization updates and, in principle, through changes in the policy-induced distribution of future rollout data.

This motivates the development of a framework that directly estimates counterfactual influence beyond the local scope of individual RL training rounds. To address this, we formulate data attribution in online RL through a principled leave-one-out objective defined over multiple training rounds, and use it to study the non-local value of rollout data. Our main contributions are summarized as follows:

- **A non-local attribution framework for online RL (Section 3.1).** Prior online-RL attribution methods are local and heuristic; we formulate non-local data attribution in online RL through an RL leave-one-out objective, *RL-LOO*, defined over the training trajectory.
- **A first-order estimator with optimization and sampling effects (Section 3.2).** We derive *RL-IF*, a first-order estimator of RL-LOO, and show that online-RL data influence decomposes into **optimization effect** and **sampling effect**. This yields both a conceptual lens for understanding long-horizon influence and a computationally efficient approximation, *RL-IF-opt*, that is well supported by our theory and experiments.
- **A practical data-selection algorithm with downstream gains (Section 4).** Building on an efficient approximation to RL-IF, we develop **RINSE**, a non-local attribution-based filtering method for online RL. Across standard benchmark environments and an RLHF-style toxicity-mitigation setting, RINSE improves training efficiency and final performance relative to standard PPO and local filtering baselines.

2 Preliminaries

2.1 Background

Online RL algorithms. We consider an online reinforcement learning (RL) process indexed by rounds $t = 1, \ldots, T$. At round t, the current policy π_{θ_t} interacts with the environment to collect a rollout buffer D_t, after which the policy is updated for M gradient steps. In this work, we focus on a specific RL algorithm, proximal policy optimization (PPO) [Schulman et al., 2017]. Given the state action pair (s, a), the standard clipped PPO objective maximizes

$$L^{\text{clip}}(\theta) := \mathbb{E}_{(s,a)\sim D_t}\Big[\min\big(r_t(\theta)\hat{A}, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}\big)\Big], \quad r_t(\theta) := \frac{\pi_\theta(a \mid s)}{\pi_{\theta_t}(a \mid s)}. \tag{1}$$

Here, $r_t(\theta)$ is the policy ratio, $\hat{A}$ is an advantage estimate, π_{θ_t} is the behavior policy used to collect the rollout, and ϵ is the clipping threshold. Schulman et al. [2017] also introduced an alternative formulation of the PPO objective, in which the clipping mechanism is replaced by KL regularization. Specifically, let β be the coefficient of the KL penalty, and the KL-regularized form of PPO maximizes

$$L^{\text{KL}}(\theta) := \mathbb{E}_{(s,a)\sim D_t}\big[r_t(\theta)\hat{A} - \beta\,\text{KL}\big(\pi_{\theta_t}(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s)\big)\big].$$

Let $P_t(\cdot \mid \theta_t)$ denote the distribution over rollout buffers induced by π_{θ_t}, where a rollout buffer D_t is the collection of trajectories sampled in round t. Write $D_t^{(j)} \subseteq D_t$ for the j-th minibatch in round

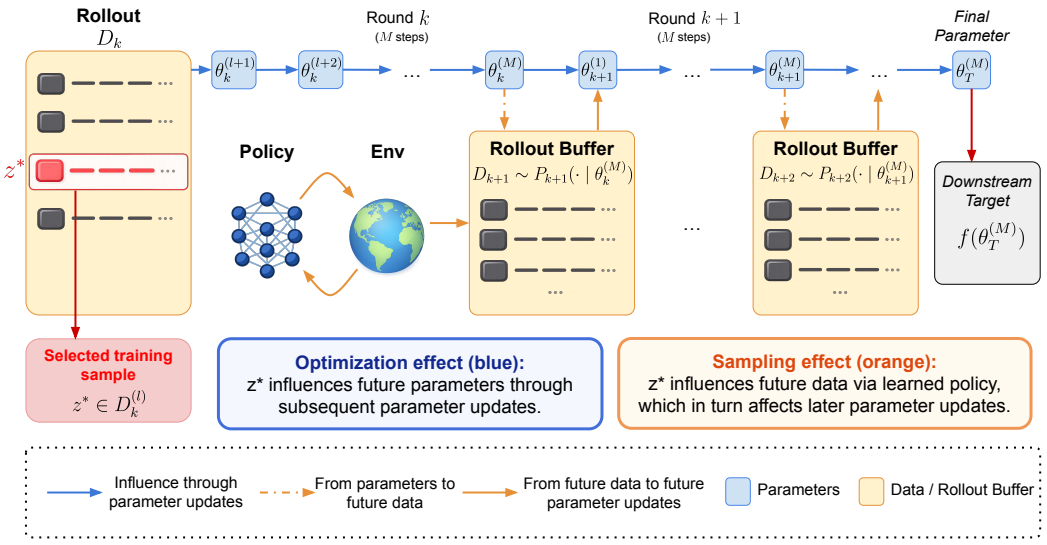


Figure 1: Two-fold data influence in RL. A training occurrence $z^* \in D_k^{(l)}$ affects learning through (i) the *optimization path* (blue): propagating via parameter updates to $\theta_T^{(M)}$, and (ii) the *sampling path* (orange): altering future rollout buffers $D_{k+1}, D_{k+2}, \ldots$. These effects are coupled through the policy-induced data distribution.

t, and $\theta_t^{(j)}$ for the parameter at the j-th gradient step, with $\theta_t = \theta_t^{(0)} = \theta_{t-1}^{(M)}$. For simplicity, we assume that the policy is updated with stochastic gradient descent, and each update takes the form:

$$\theta_t^{(j)} = \theta_t^{(j-1)} - \frac{\eta_t}{b} \sum_{x \in D_t^{(j)}} \nabla_\theta \ell(\theta_t^{(j-1)}, x), \qquad D_t \sim P_t(\cdot \mid \theta_t), \tag{2}$$

where η_t is the learning rate, b is the minibatch size, and ℓ denotes the PPO loss.

Local data attribution in online RL. Hu et al. [2025] propose a *local* data attribution framework for online RL, which measures how much a training sample contributes to a downstream target function f. In this setting, the training samples are scored by gradient inner products between the per-sample gradient and gradient on an attribution target f. The local attribution score is estimated as

$$I^{(k,l)}(z^*; f) := -\eta_k \, \langle \nabla_\theta \ell(\theta, z^*), \nabla_\theta f(\theta) \rangle \,, \tag{3}$$

A key design choice is the attribution target f. Hu et al. [2025] propose to leverage a policy-gradient surrogate for downstream return, defined on a reference validation distribution. Given a reference policy π^{ref} and a reference advantage estimator $\hat{A}^{\text{ref}}$, Hu et al. [2025] define

$$f^{\text{return}}(\theta) := \mathbb{E}_{\tau \sim \pi^{\text{ref}}, (s,a) \sim \tau} \big[\log \pi_\theta(a \mid s) \, \hat{A}^{\text{ref}}(s,a) \big]. \tag{4}$$

The target gives the attribution score a direct behavioral interpretation: a positive score means the sample pushes the policy toward higher surrogate return, whereas a negative score suggests that removing the sample may help.

Discrepancy between short-term and long-term influence. A major limitation of local data attribution is its myopic view of data influence. To illustrate this point, we empirically estimate the data influence[2] in `LunarLander` [Brockman et al., 2016], and track how the influence of the same sample changes as the *horizon* H grows, where horizon means the number of training rounds over which the sample's effect is propagated and evaluated. As shown in Figure 2, a good sample that stabilizes the lander may appear harmful from a short-term viewpoint even though it improves downstream behavior and is beneficial in the long run, and vice versa. Systematically, we also observed a bias induced by the local viewpoint. In particular, local attribution tends to overestimate the long-term value of late-episode samples, those from late time steps in an episode.

[2]We estimate data influence by removing the sample from the training trajectory and measuring the resulting change in the final attribution target, as formally defined in Section 3.1.

Two-fold data influence in online RL. In online RL, the influence of a training sample is not limited to its direct effect on a parameter update. A rollout affects learning through two coupled channels, as demonstrated in Figure 1. First, it influences the checkpoint by contributing to the gradient update that produces the next parameter. This is the usual **optimization effect** of data: removing one sample perturbs the parameter trajectory and therefore the final checkpoint. Second, the same rollout influences future data sampling. Since subsequent rollout buffers are collected by policies obtained from earlier checkpoints, perturbing one training sample changes not only the next iterate but also the state-action distribution from which later data are drawn. This is referred to as the **sampling effect**: a sample influences future learning by altering the distribution of later data. Hence, in on-policy RL, data influence is inherently two-fold: it acts on both the optimization path and the sampling path. Moreover, the two effects are generally entangled rather than separable: a perturbation to one sample changes the data distribution of future rollouts, and this distributional shift in turn alters the subsequent optimization trajectory.

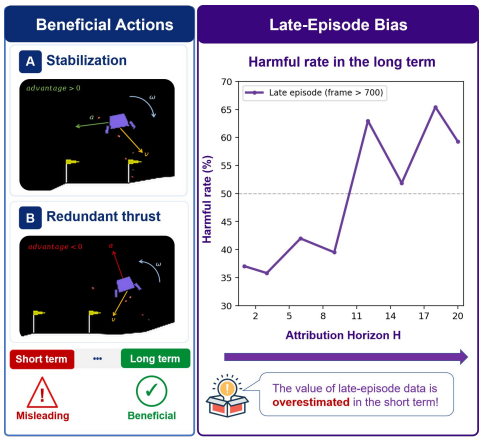


Figure 2: Left: two LunarLander samples that appear harmful locally but are beneficial in the long run. Right: fraction of late-episode samples labeled harmful vs. attribution horizon H.

3 Measuring Data Influence in Online RL

In this section, we systematically formulate and approximate data influence in online reinforcement learning. We first introduce a leave-one-out counterfactual objective, RL-LOO, which measures the long-term effect of removing a training occurrence. Then, we derive a first-order influence estimator, RL-IF, that approximates this counterfactual without retraining by propagating perturbations through the training trajectory. The resulting recursion naturally decomposes data influence into the aforementioned optimization effect and sampling effect, and admits theoretical guarantees under suitable smoothness assumptions. Finally, we analyze both the effectiveness of RL-IF and the relative contributions of these two effects through theoretical analysis and empirical evaluation.

3.1 Problem Formulation

Following prior work, we study data attribution in an online PPO-style training pipeline, and extend it beyond individual RL training rounds. Let $\theta_T^{(M)}$ denote the final policy parameter after all rounds of training. Our goal is to systematically measure how one training sample would affect a downstream target after subsequent online updates have taken place.

We formalize this through a leave-one-out (LOO) [Koh and Liang, 2017] counterfactual specific to RL, or *RL-LOO*, following a similar setup in supervised learning literature [Hara et al., 2019, Wang et al., 2025]. The unit of attribution is a particular *occurrence* of a sample in a training minibatch, written as $z^* \in D_k^{(l)}$, which specifies both the sample identity and the minibatch in which it is used. Let $\widetilde{\theta}_T^{(M;-z^*)}$ denote the final parameter obtained by removing this occurrence from the training trajectory and replaying the remaining optimization process, including the induced changes in future sampling. For a downstream target function f, RL-LOO is defined as

$$\text{RL-LOO}^{(k,l)}(z^*; f) := f(\widetilde{\theta}_T^{(M;-z^*)}) - f(\theta_T^{(M)}).$$

This quantity measures the effect of removing one training occurrence on a downstream target after subsequent training. As for the attribution target, we adopt the negative of the target defined in (4), so that negative scores identify occurrences whose removal improves the surrogate objective. In the non-local setting, the reference policy is naturally chosen as the final policy $\pi_{\theta_T}^{(M)}$, so that the target measures how removing one training occurrence changes the surrogate return of the final policy on its own induced distribution. $\hat{A}^{\text{ref}}$ can be approximated using the corresponding critic and the same GAE

procedure and hyperparameters as PPO. When exact advantages are used, we show that RL-LOO with this target serves as a local proxy for the change in final expected return in Appendix A.4.

3.2 Approximating Leave-One-Out in RL

We now derive a first-order estimator, *RL-IF* (RL InFluence), for approximating RL-LOO. At a high level, RL-LOO asks how the final policy would change if a training sample were removed from a rollout-based training trajectory. Rather than explicitly retraining the policy, RL-IF approximates this counterfactual effect through first-order perturbation analysis.

Consider the RL training process in (2). Let $z^* \in D_k^{(l)}$ be a sample removed from update (k, l), and let $\theta_t^{(j)}(\varepsilon)$ denote the perturbed training trajectory obtained by continuously downweighting z^* by a scalar ε. A first-order Taylor expansion gives

$$\text{RL-LOO}^{(k,l)}(z^*; f) \approx \nabla_\theta f(\theta_T^{(M)})^\top \Delta_T^{(M)}, \qquad \Delta_t^{(j)} := \left.\frac{\partial \theta_t^{(j)}(\varepsilon)}{\partial \varepsilon}\right|_{\varepsilon=0}, \tag{5}$$

which motivates the first-order estimator

$$\text{RL-IF}^{(k,l)}(z^*; f) := \nabla_\theta f(\theta_T^{(M)})^\top \Delta_T^{(M)}.$$

The key step is to characterize the perturbation trajectory $\Delta_t^{(j)}$. In online RL, perturbations propagate through two distinct channels, the *optimization effect* and the *sampling effect*, respectively. To unify the effects from both channels, we define the stage update operator

$$F_t^{(j)}(\vartheta, \beta) := \vartheta - \frac{\eta_t}{b} \sum_{x \sim S_t^{(j)}(\beta)} \nabla_\theta \ell(\vartheta, x), \quad S_t^{(j)}(\beta) := \begin{cases} D_t^{(j)}, & \text{if } t = k \text{ and } j > l, \\ P_t(\cdot \mid \beta), & \text{if } t \geq k+1. \end{cases} \tag{6}$$

where ϑ denotes the parameter being updated and β denotes the policy parameter controlling data collection. Differentiating the update recursion with respect to ε yields

$$\Delta_t^{(j)} = \frac{\partial F_t^{(j)}}{\partial \vartheta} \Delta_t^{(j-1)} + \frac{\partial F_t^{(j)}}{\partial \beta} \Delta_t^{(0)},$$

where the resulting recursion naturally distinguishes the two channels: the first term associated with the current parameters ϑ captures the optimization effect, and the second term related to the sampling parameters β captures the sampling effect. With further simplification, the resulting first-order perturbation recursion is summarized below.

Proposition 3.1: First-order perturbation recursion for RL-IF

Let $\Delta_t^{(j)}$ denote the perturbation induced by downweighting sample $z^* \in D_k^{(l)}$. The perturbation is initialized as

$$\Delta_k^{(l)} = \frac{\eta_k}{b} \nabla_\theta \ell(\theta_k^{(l-1)}, z^*),$$

and evolves according to

$$\Delta_t^{(j)} = \begin{cases} A_t^{(j)} \Delta_t^{(j-1)}, & \text{if } t = k,\ j > l, \\ A_t^{(j)} \Delta_t^{(j-1)} + B_t^{(j)} \Delta_t^{(0)}, & \text{if } t \geq k+1, \end{cases} \tag{7}$$

where

$$A_t^{(j)} := I - \frac{\eta_t}{b} H_t^{(j)}, \qquad B_t^{(j)} := -\frac{\eta_t}{b} \sum_{x \in D_t^{(j)}} \nabla_\theta \ell(\theta_t^{(j-1)}, x) \nabla_\theta \log P_t(x \mid \theta_t)^\top.$$

Here $A_t^{(j)}$ and $B_t^{(j)}$ correspond to the optimization and sampling effects, respectively.

The recursion formula of RL-IF in Proposition 3.1 propagates the first-order effect of removing z^* through both later optimization steps and future sampling decisions. The term $A_t^{(j)} \Delta_t^{(j-1)}$ captures

the *optimization effect*, while $B_t^{(j)}\Delta_t^{(0)}$ captures the additional *sampling effect* caused by policy-dependent data collection. Notably, the sampling effect vanishes when the perturbation remains within the same training round. In practice, we use $\nabla \log \pi_{\theta_t}(a \mid s)$ to approximate $\nabla \log P_t((s,a) \mid \theta_t)$. Note that both matrix multiplications in this formula can be tackled efficiently: multiplying by $A_t^{(j)}$ can be computed using Hessian Vector Product (HVP), and multiplying by $B_t^{(j)}$ is essentially two vector products. We further provide a detailed derivation of the recursion, together with a first-order approximation guarantee for RL-IF under smoothness assumptions, in Appendix A.1 and Appendix B.1.

3.3 Sampling vs. Optimization Effects in RL-IF

A more detailed analysis of RL-IF can be obtained by unrolling the recursion and separating the optimization and sampling effects. In particular, the recursion in (7) admits a compact closed form.

$$\text{RL-IF}^{(k,l)}(z^*; f) = \nabla_\theta f(\theta_T^{(M)})^\top \left(\prod_{t=T}^{k+1} C_t\right) \left(\prod_{j=M}^{l+1} A_k^{(j)}\right) \frac{\eta_k}{b} \nabla_\theta \ell(\theta_k^{(l-1)}, z^*). \tag{8}$$

Here, for each stage $t > k$,

$$A_t := \prod_{j=M}^{1} A_t^{(j)}, \qquad C_t := A_t + \sum_{q=1}^{M} \left(\prod_{j=M}^{q+1} A_t^{(j)}\right) B_t^{(q)}.$$

Equation (8) is the closed-form solution of the recursion. Here, the product of A_t captures the accumulated *optimization effect*, while the terms involving $B_t^{(j)}$ capture the accumulated *sampling effect*. When $B_t^{(q)} = 0$, the expression reduces to the optimization-only case. The following theorem formalizes the gap between the optimization-only and full sampling-aware effects.

Theorem 3.1: Accumulated relative gap

Assume that $\ell(\theta; x)$ is twice differentiable, that its gradient norm is bounded by G, that its Hessian norm is bounded by Λ. Assume in addition that for each future stage $t > k$, the sampling family $P_t(\cdot \mid \theta)$ is KL-smooth with score second moment bounded by $L_{\max}$. Assume the learning rate takes the harmonic form $\eta_t = \frac{\eta_0}{M(t-k)}$ with $\eta_0\Lambda < 1$. Then, for any $T > k$ with $H := T - k$, the accumulated gap satisfies

$$\left\| \prod_{t=T}^{k+1} C_t - \prod_{t=T}^{k+1} A_t \right\|_2 \le (e^\alpha H^\alpha - 1) \prod_{t=k+1}^{T} \|A_t\|_2, \tag{9}$$

where $\alpha := \frac{G\sqrt{L_{\max}}\,\eta_0}{(1-\eta_0\Lambda)^2}$.

The bound in (9) shows that the sampling effect remains a *relative perturbation* to the optimization effect. In particular, if $\eta_0\Lambda \ll 1$, the accumulated gap over H stages grows only logarithmically

$$e^\alpha H^\alpha - 1 \approx \alpha(1 + \log H)$$

Hence, in this regime, the accumulated gap remains well-conditioned over long horizons, and the sampling effect acts as a controlled perturbation to the optimization dynamics. Accordingly, we define the optimization-only variant **RL-IF-opt** as the estimator obtained by setting $B_t^{(j)} = 0$ in the recursion, or equivalently by dropping the sampling terms in the closed-form operator C_t. The original estimator that retains both optimization and sampling terms is denoted **RL-IF-full**. The gap suggests that RL-IF is often dominated by its optimization component, making RL-IF-opt a compelling simplification in practice. The proof of Theorem 3.1 is deferred to Appendix B.3.

Table 1: Linear data modeling scores of attribution methods on `Ant-v4` and `HalfCheetah-v4`.

Score	Ant-v4 LDS ↑	HalfCheetah-v4 LDS ↑
Random	-0.031 ± 0.058	-0.063 ± 0.032
TracIn	0.055 ± 0.102	0.138 ± 0.037
RL-IF-opt	0.191 ± 0.037	0.218 ± 0.027
RL-IF-full	0.192 ± 0.033	0.218 ± 0.026

Experimental comparison. To further validate the theoretical findings, we evaluate attribution quality using *Linear Data Modeling Scores* (LDS) [Park et al., 2023]. LDS measures the Spearman correlation between aggregated attribution scores and the observed effect of removing subsets of training data under retraining. We compare the two variants of RL-IF, along with TracIn [Pruthi et al., 2020], an attribution method that does not account for optimization or sampling dynamics, and a random baseline. We consider KL-regularized PPO on two consecutive training rounds, where retraining is performed with future rollouts resampled using the updated policy. The results are shown in Table 1. Both RL-IF-opt and RL-IF-full substantially outperform local attribution methods such as TracIn, while exhibiting nearly identical performance to each other. This small gap suggests that, empirically, the sampling effect contributes only marginally beyond the optimization effect. Additional setup and evaluation details are provided in Appendix D.1.

4 Data Selection with RL-IF

We demonstrate a direct downstream application of RL-IF in online RL: using RL-IF attribution scores to guide data selection over multiple training rounds. The goal is to identify and remove training samples that are harmful to downstream performance, thereby improving both compute efficiency and sample efficiency. We build our data selection algorithm on KL-regularized PPO, since the clipping objective in (1) is piecewise-defined and introduces non-smooth behavior at the clipping boundary, which is less compatible with the first-order approximation used in RL-IF. Our source code is available at `https://github.com/sx-liu/RL-IF`.

RINSE: RL-IF-based Sample Selection. The RL-IF formulation naturally leads to a data-selection algorithm, which we call **RINSE** (**R**L-**IN**fluence-based **S**ample s**E**lection), over a finite horizon window. Suppose we aim to improve PPO training for H rounds, starting from round k. We first run the standard PPO pipeline for H subsequent rounds, obtaining policies $\pi_{\theta_{k+1}}, \ldots, \pi_{\theta_{k+H}}$ and the corresponding training trajectory $\{D_t^{(j)}\}_{t=k}^{k+H-1}$. This rollout trajectory provides the reference path on which long-horizon influence is evaluated.

Algorithm 1 RINSE: RL-IF-based Sample Selection

Require: Initial policy π_{θ_k}, horizon H, discard ratio ρ

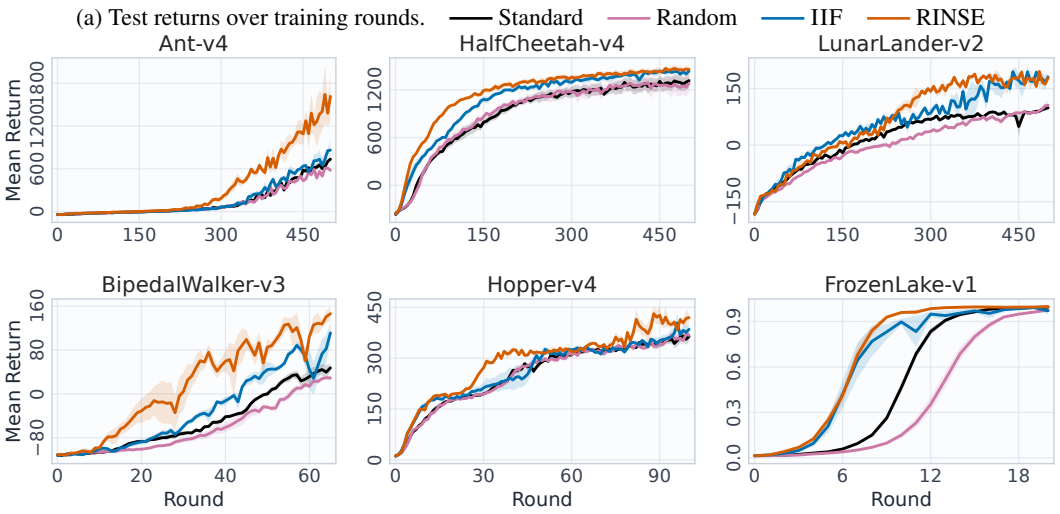
$\theta \leftarrow \theta_k$
for $h = 1, \ldots, H$ **do**
  $D_{k+h-1} \leftarrow \textsc{CollectRollout}(\theta)$
  $r(h) \leftarrow \textsc{Evaluate}(D_{k+h-1})$
  if $h > 1$ and $r(h-1) > r(h)$ **then**
    break
  end if
  $\widehat{H} \leftarrow h$
  $\theta \leftarrow \textsc{TrainRollout}(\theta; D_{k+h-1})$
end for
Compute optimization-only RL-IF-opt scores $\{s(x)\}$ for all samples $x \in \bigcup_{t=k}^{k+\widehat{H}-1} D_t$
$\mathcal{R} \leftarrow \textsc{NegBottomFraction}(\{s(x)\}, \rho)$
Construct filtered datasets $\widetilde{D}_t = D_t \setminus \mathcal{R}$ for $t = k, \ldots, k+\widehat{H}-1$
$\theta \leftarrow \theta_k$
for $h = 1, \ldots, \widehat{H}$ **do**
  $\theta \leftarrow \textsc{TrainRollout}(\theta; \widetilde{D}_{k+h-1})$
end for
return $\pi_{\theta_{k+\widehat{H}}}$

Given this trajectory, we estimate the influence of samples in $\{D_t\}_{t=k}^{k+H-1}$ using RL-IF, with the final policy $\pi_{\theta_{k+H}}$ in this window as the reference policy in (4). Because the optimization-only approximation achieves comparable empirical performance at substantially lower cost, RINSE uses the optimization-only variant, RL-IF-opt, in the selection procedure. We rank samples by their attribution scores, remove those with the most negative influence based on discard ratio ρ, and then retrain on the filtered datasets to obtain updated policies $\pi_{\theta'_{k+1}}, \ldots, \pi_{\theta'_{k+H}}$. Since this retraining stage is inherently off-policy, we use an adjusted retraining objective,

$$L^{\text{KL-Retrain}}(\theta) := \mathbb{E}\big[r_t(\theta)\hat{A}_t - \beta\,\mathrm{KL}\big(\pi_{\theta'_t}(\cdot \mid s_t) \,\|\, \pi_\theta(\cdot \mid s_t)\big)\big],$$

which limits policy drift from the reference policies along the retraining trajectory.

Design of attribution horizon. The effectiveness of attribution-based selection depends on the attribution horizon and the choice of reference policy. Short horizons lead to overly myopic attribution that reduces to local effects, while excessively long horizons introduce severe off-policy mismatch between the selected data and later training dynamics. Moreover, the reference policy must be sufficiently strong to provide a reliable attribution target.



(b) Relative efficiency and peak reward. TE/SE are normalized to standard PPO; lower is better.

Metric	Method	Ant	HalfCheetah	LunarLander	BipedalWalker	Hopper	FrozenLake
TE% ↓	IIF	110.2±3.7	75.0±11.0	**76.1±8.5**	94.9±6.0	94.5±18.8	133.4±26.0
	RINSE	**103.0±4.9**	**63.9±10.4**	89.9±6.8	**67.9±13.6**	**88.9±16.7**	**116.3±12.5**
SE% ↓	IIF	86.3±3.3	57.7±8.5	52.6±6.0	74.8±5.0	72.2±14.6	64.2±12.2
	RINSE	**74.9±3.7**	**45.4±7.3**	**49.8±3.7**	**47.1±9.7**	**63.9±11.9**	**53.5±5.0**
Max Reward↑	Standard	734.3±14.4	1316.6±91.6	99.1±4.3	47.5±6.7	365.3±11.6	0.99±0.00
	IIF	861.6±146.2	1437.9±20.2	195.0±8.3	111.1±18.4	403.7±31.3	0.99±0.00
	RINSE	**1642.8±408.6**	**1470.2±39.3**	**195.3±3.8**	**146.0±13.3**	**431.0±20.7**	**1.00±0.00**

Figure 3: Data-selection results under rollout filtering. (a) Test returns over training rounds. (b) Relative time efficiency, sample efficiency, and peak mean return across environments. RINSE largely improves return and efficiency over standard training and local baselines.

To balance these considerations, we adopt a **window-based data selection** strategy, which partitions training into consecutive windows and performs attribution within each window independently. This maintains proximity to the current training dynamics while still capturing non-local effects over multiple future updates.

We further use an **adaptive horizon selection** procedure. Given a maximum window size $H_{\max}$, we progressively extend the horizon under standard training and monitor the mean episode return $r(h)$ of policy $\pi_{\theta_{k+h-1}}$. We stop when either $h = H_{\max}$ or performance ceases to improve, i.e., $r(h-1) > r(h)$, and select the largest valid horizon for attribution and data selection.

The complete procedure is summarized in Algorithm 1, while ablations of our design choices are presented in Appendix D.3.

Experimental setup. We evaluate our method on six standard RL benchmarks spanning continuous control and navigation tasks from the Gym/Gymnasium ecosystem, including MuJoCo domains [Todorov et al., 2012, Brockman et al., 2016, Towers et al., 2025]. We use PPO with a KL-regularized objective as the underlying training algorithm and fix the maximum horizon to $H_{\max} = 5$ without tuning. We compare RINSE against three baselines: standard PPO, **Random** filtering, and **IIF** [Hu et al., 2025], a local-attribution filtering method based on TracIn scores. For fairness, Random and IIF use the same discard ratio ρ as RINSE.

We report both **time efficiency (TE)** and **sample efficiency (SE)** relative to standard PPO. For each method, we measure the runtime and the amount of training data required to reach a set of target performance levels, normalized such that the PPO baseline corresponds to 100%. We also report full training curves. Additional setup and metric details are provided in Appendix C.1 and Appendix D.4.

Results. Figure 3 shows that RINSE consistently outperforms both standard PPO and IIF. Although RINSE invokes TRAINROLLOUT twice per round and incurs additional influence-computation overhead, it reduces the number of required rollouts by an average of 42.8% to achieve the same performance as standard training. Consequently, RINSE remains more efficient than the local-attribution baseline on nearly all environments.

RINSE achieves the best sample efficiency and highest peak return across all six tasks, with especially large improvements on challenging continuous-control environments such as `Ant` and `BipedalWalker`. These results suggest that RL-IF-opt captures the most useful component of long-term data influence for multi-round training improvement. We further report the wall-clock breakdown in Appendix D.4.1.

4.1 Extension to RLHF

We also consider an RLHF-style toxicity-mitigation task. Starting from GPT-Neo 2.7B [Black et al., 2021], each training round collects prompt-response trajectories from the current policy, assigns a toxicity-based reward to each response using a pretrained classifier [Vidgen et al., 2021, Gehman et al., 2020], and then updates the policy following standard RLHF pipelines [Ouyang et al., 2022]. We use the KL-regularized PPO objective described in Section 2.1. For additional efficiency, we use the Generalized Gauss–Newton matrix (GGN) [Martens, 2020, Wang et al., 2025] to approximate the Hessian term in (7), completely reducing the HVP to efficient vector product operations. Further details are in Appendix C.2.

Results. Figure 4 shows the toxicity trajectory during training: RINSE achieves lower toxicity probability than the baselines across most of training. Notably, in terms of relative TE on reducing test toxicity, RINSE achieves an average training speedup of more than $3\times$ over the standard training pipeline and more than $2\times$ over IIF. We also provide detailed training details, additional analysis and runtime breakdown in Appendix D.5. These results suggest that RINSE extends naturally beyond continuous-control benchmarks to large-scale language-model alignment.

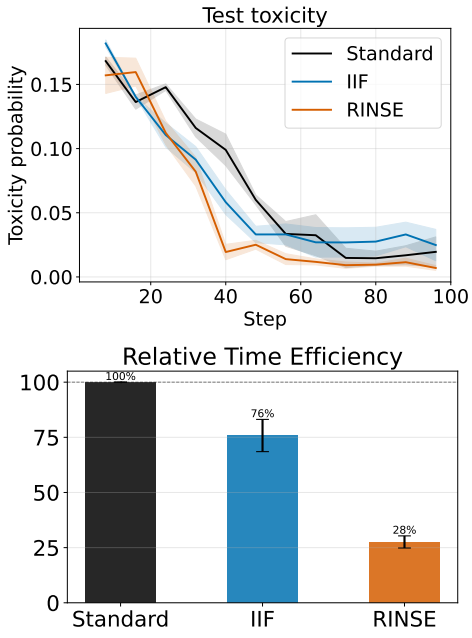


Figure 4: Toxicity-mitigation results in the RLHF setting.

5 Related Work

Training-data attribution and data valuation. Our work is closely related to the literature on training-data attribution and data valuation [Deng et al., 2025]. Classical influence functions approximate leave-one-out effects through second-order sensitivity analysis [Koh and Liang, 2017], while later methods improve scalability by tracing gradient alignment along optimization trajectories, as in TracIn [Pruthi et al., 2020], by using representer-point decompositions [Yeh et al., 2018], or by designing scalable estimators for modern large-scale models [Park et al., 2023]. Recent work further studies attribution in temporally dependent or sequential training settings [Wang et al., 2025]. A closely related line studies *data valuation*, where training examples are assigned value according to their contribution to model quality or downstream utility [Ghorbani and Zou, 2019, Jia et al., 2019, Kwon and Zou, 2022]. Other work uses per-example influence signals for data cleaning, robust reweighting, and data selection [Hara et al., 2019, Xia et al., 2024, Wang et al., 2024].

Most existing attribution methods assume a fixed training distribution. In contrast, our work studies attribution in online RL, where the training distribution evolves with the policy itself. We focus specifically on *non-local* attribution: instead of measuring only the immediate effect of a sample on a nearby update, we study how perturbations propagate through future optimization and future data collection over an extended training trajectory.

Data-centric perspectives in RL. A growing line of work studies RL from a data-centric perspective, asking which trajectories, transitions, or training contexts are most useful for improving policy learning. In off-policy RL, this question appears naturally in experience replay and replay prioritization, where informative transitions are replayed more frequently to improve sample efficiency [Schaul et al., 2016]. Related ideas also appear in curriculum learning and curriculum design, where training tasks or experiences are structured adaptively to improve learning dynamics [Florensa et al., 2017, Jiang et al., 2021]. Recent work also explores trajectory filtering, selective data reuse, and replay-buffer management for improving robustness and efficiency in RL training.

Our work is most closely related to recent attribution methods for online RL. In on-policy training, trajectories are generated by the current policy and immediately reused for optimization [Williams, 1992, Sutton et al., 1999, Schulman et al., 2015, 2017]. Consequently, perturbing a single training occurrence can influence not only later parameters, but also the future rollout distribution induced by the updated policy. Hu et al. [2025] take an important first step in this direction by proposing a local attribution framework for PPO that measures the influence of rollout records relative to a nearby checkpoint within a fixed rollout buffer. Our work builds directly on this line, but studies a fundamentally different target: instead of estimating the immediate local effect of a sample, we study its *non-local* influence on future learning outcomes over subsequent training rounds.

6 Conclusion and Limitations

We study data attribution in online reinforcement learning, where the training data are generated by the evolving policy rather than drawn from a fixed distribution. To capture this setting, we formulate a trajectory-level leave-one-out quantity, *RL-LOO*, and derive a first-order estimator, *RL-IF*, that propagates the effect of one training occurrence through both later optimization and future sampling. This yields a counterfactual perspective on data value in online RL that goes beyond local attribution. Empirically, the resulting optimization-only approximation, *RL-IF-opt*, is already effective enough to support a practical filtering method, **RINSE**, which improves efficiency and final performance on standard benchmarks and in an RLHF-style toxicity-mitigation setting.

Our work also has several limitations. First, the current development is specialized to PPO-style on-policy training [Schulman et al., 2017] and to a first-order approximation of the leave-one-out trajectory. Extending the framework to other online RL algorithms, including closely related trust-region methods such as TRPO [Schulman et al., 2015] and other settings with substantially different sampling or update structures, remains an important direction. Second, our derivation is based on SGD-style [Amari, 1993] updates, whereas modern deep RL and RLHF pipelines often rely on adaptive optimizers such as Adam [Kingma and Ba, 2015]. Although the method works empirically in our Adam-based experiments, a derivation that explicitly accounts for adaptive preconditioning would make the estimator more principled [Deng et al., 2026]. Finally, our analysis and experiments suggest that the sampling term is often smaller than the optimization term in the regimes we study, but this need not hold universally; understanding when sampling effects become dominant is an open question.

Acknowledgements

This project was partially supported by an ACCESS allocation CIS260068. JM would like to acknowledge support by NSF CAREER Award No. 2542856. YH and HZ are partially supported by an NSF grant No. 2416897 and an NSF CAREER Award No. 2442290.

References

Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4–5): 185–196, 1993.

Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. GPT-Neo: Large scale autoregressive language modeling with Mesh-TensorFlow. Zenodo, 2021.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, 2024.

Junwei Deng, Yuzheng Hu, Pingbang Hu, Ting-Wei Li, Shixuan Liu, Jiachen T. Wang, Dan Ley, Qirun Dai, Benhao Huang, Jin Huang, et al. A survey of data attribution: Methods, applications, and evaluation in the era of generative AI. SSRN Electronic Journal, 2025.

Junwei Deng, Pingbang Hu, Suliang Jin, Hao Lu, Jiachen T. Wang, Shichang Zhang, and Jiaqi W. Ma. How faithful is trajectory-based data attribution? Error sources, remedies, and practical guidelines. *arXiv preprint arXiv:2605.18814*, 2026.

Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *Conference on Robot Learning*, pages 482–495, 2017.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, 2020.

Amirata Ghorbani and James Zou. Data Shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251, 2019.

Satoshi Hara, Atsushi Nitanda, and Takanori Maehara. Data cleansing for models trained with SGD. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Pingbang Hu, Xueshen Liu, Z. Morley Mao, and Jiaqi W. Ma. Dr. Post-Training: A data regularization perspective on LLM post-training. *arXiv preprint arXiv:2605.07063*, 2026.

Yuzheng Hu, Fan Wu, Haotian Ye, David Forsyth, James Zou, Nan Jiang, Jiaqi W. Ma, and Han Zhao. A snapshot of influence: A local data attribution framework for online reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J. Spanos. Towards efficient data valuation based on the Shapley value. In *International Conference on Artificial Intelligence and Statistics*, pages 1167–1176, 2019.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *International Conference on Machine Learning*, pages 4940–4950, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894, 2017.

Yongchan Kwon and James Zou. Beta Shapley: A unified and noise-reduced data valuation framework for machine learning. In *International Conference on Artificial Intelligence and Statistics*, pages 8780–8802, 2022.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK: Attributing model behavior at scale. In *International Conference on Machine Learning*, pages 27074–27113, 2023.

Amalie Brogaard Pauli, Maria Barrett, Ophélie Lacroix, and Rasmus Hvingelby. DaNLP: An open-source toolkit for Danish natural language processing. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 460–466, 2021.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, pages 19920–19930, 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-Baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016.

John Schulman. Approximating KL divergence. `http://joschu.net/blog/kl-approx.html`, 2020. Accessed: 2026-05-06.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Yang Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12, pages 1057–1063, 1999.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 38, 2025.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1667–1682, 2021.

Jiachen T. Wang, Tong Wu, Dawn Song, Prateek Mittal, and Ruoxi Jia. GREATS: Online selection of high-quality data for LLM training in every iteration. In *Advances in Neural Information Processing Systems*, volume 37, pages 131197–131223, 2024.

Jiachen T. Wang, Dawn Song, James Zou, Prateek Mittal, and Ruoxi Jia. Capturing the temporal dependence of training data influence. In *International Conference on Learning Representations*, 2025.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1391–1399, 2017.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning*, pages 54104–54132, 2024.

Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K. Ravikumar. Representer point selection for explaining deep neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Yang Yu. Towards sample efficient reinforcement learning. In *International Joint Conference on Artificial Intelligence*, pages 5739–5743, 2018.

Erle Zhu, Dazhi Jiang, Yuan Wang, Xujun Li, Jiale Cheng, Yuxian Gu, Yilin Niu, Aohan Zeng, Jie Tang, Minlie Huang, and Hongning Wang. Data efficient RLVR via off-policy influence guidance. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46167–46192, 2026.

A Additional Details on RL-IF

A.1 Derivation of RL Leave-One-Out and RL-IF

In this section, we derive the recursion underlying RL-IF. To estimate the RL-LOO effect of a sample $z^* \in D_k^{(l)}$, we inject a perturbation at round k, step l, while keeping all subsequent updates unchanged:

$$\theta_k^{(l)}(\varepsilon) = \theta_k^{(l-1)} - \frac{\eta_k}{b} \sum_{x \in D_k^{(l)} \setminus \{z^*\}} \nabla_\theta \ell(\theta_k^{(l-1)}, x) - \frac{\eta_k}{b}(1-\varepsilon)\nabla_\theta \ell(\theta_k^{(l-1)}, z^*). \tag{10}$$

The injected perturbation at (k, l) is obtained by differentiating (10):

$$\Delta_k^{(l)} := \left.\frac{\partial \theta_k^{(l)}(\varepsilon)}{\partial \varepsilon}\right|_{\varepsilon=0} = \frac{\eta_k}{b} \nabla_\theta \ell(\theta_k^{(l-1)}, z^*).$$

To characterize the terminal perturbation $\Delta_T^{(M)}$, we introduce a stage-update operator that makes explicit the dependence on the current iterate and the rollout distribution. For each update (t, j), define

$$F_t^{(j)}(\vartheta, \beta) := \vartheta - \frac{\eta_t}{b} \sum_{x \sim S_t^{(j)}(\beta)} \nabla_\theta \ell(\vartheta, x), \qquad S_t^{(j)}(\beta) := \begin{cases} D_t^{(j)}, & \text{if } t = k \text{ and } j > l, \\ P_t(\cdot \mid \beta), & \text{if } t \geq k+1. \end{cases}$$

Here $D_t^{(j)}$ denotes a fixed minibatch, while $P_t(\cdot \mid \beta)$ denotes the rollout distribution induced by β. The perturbed trajectory follows $\theta_t^{(j)}(\varepsilon) = F_t^{(j)}\big(\theta_t^{(j-1)}(\varepsilon), \theta_t^{(0)}(\varepsilon)\big)$. Recall the definition of perturbation $\Delta_t^{(j)}$ in (5), and by the chain rule, we have

$$\Delta_t^{(j)} = \frac{\partial F_t^{(j)}}{\partial \vartheta} \Delta_t^{(j-1)} + \frac{\partial F_t^{(j)}}{\partial \beta} \Delta_t^{(0)}. \tag{11}$$

Optimization effect. Differentiating $F_t^{(j)}$ with respect to the current iterate ϑ gives $\frac{\partial F_t^{(j)}}{\partial \vartheta} = A_t^{(j)} = I - \frac{\eta_t}{b} H_t^{(j)}$, where $H_t^{(j)} := \sum_{x \in D_t^{(j)}} \nabla_\theta^2 \ell(\theta_t^{(j-1)}, x)$.

Sampling effect. For $t = k$ and $j > l$, the minibatch is fixed, so $S_t^{(j)}(\beta) = D_t^{(j)}$ and $\frac{\partial F_t^{(j)}}{\partial \beta} = 0$. Thus, by (11) we have $\Delta_k^{(j)} = A_k^{(j)} \Delta_k^{(j-1)}$. For future stages $t \geq k+1$, the sampling distribution depends on β. Using the score-function identity,

$$\nabla_\beta \mathbb{E}_{x \sim P_t(\cdot|\beta)}[g(x)] = \mathbb{E}_{x \sim P_t(\cdot|\beta)} \left[g(x) \nabla_\beta \log P_t(x \mid \beta)^\top \right],$$

with $g(x) = \nabla_\theta \ell(\vartheta, x)$, we obtain

$$\frac{\partial F_t^{(j)}}{\partial \beta} = -\eta_t \mathbb{E}_{x \sim P_t(\cdot|\beta)} \left[\nabla_\theta \ell(\vartheta, x) \nabla_\beta \log P_t(x \mid \beta)^\top \right].$$

Approximating the expectation with the sampled minibatch yields $\frac{\partial F_t^{(j)}}{\partial \beta} \approx B_t^{(j)}$, where

$$B_t^{(j)} := -\frac{\eta_t}{b} \sum_{x \in D_t^{(j)}} \nabla_\theta \ell(\theta_t^{(j-1)}, x) \nabla_\theta \log P_t(x \mid \theta_t^{(0)})^\top.$$

Substituting into (11), we obtain the unified recursion $\Delta_t^{(j)} = A_t^{(j)} \Delta_t^{(j-1)} + B_t^{(j)} \Delta_t^{(0)}$, where the second term is active only for $t \geq k+1$.

A.2 Efficient Computation through Adjoint Formulation

Even with the recursive formulation of RL-IF, a key challenge is efficient computation. Directly materializing $\Delta_t^{(j)}$ for each candidate sample is computationally prohibitive. Instead, we evaluate RL-IF by propagating an adjoint vector backward through the training trajectory. This converts the forward perturbation recursion into a single backward pass involving Hessian–vector products and score-function vector products. The resulting procedure is summarized in Algorithm 2.

The algorithm returns $\nabla_\theta f(\theta_T^{(M)})^\top \Delta_T^{(M)}$, which approximates the RL leave-one-out effect $\text{RL-LOO}^{(k,l)}(z^*; f)$. Notably, all influence scores across the trajectory can be computed within a single backward pass.

Algorithm 2 One-pass RL-IF

Require: Training trajectory $\{\theta_t^{(j)}, D_t^{(j)}\}$, attribution target f, target sample $z^* \in D_k^{(l)}$

$U \leftarrow \nabla_\theta f(\theta_T^{(M)})$
for $t = T$ down to 1 **do**
  $U_{\text{prev}} \leftarrow 0$
  for $j = M$ down to 1 **do**
    if $t = k$ and $j = l$ **then**
      return $\left\langle U,\ \frac{\eta_k}{b} \nabla_\theta \ell(\theta_k^{(l-1)}, z^*) \right\rangle$
    end if
    // sampling contribution
    $U_{\text{prev}} \leftarrow U_{\text{prev}} + B_t^{(j)\top} U$
    // optimization step
    $U \leftarrow \left(I - \frac{\eta_t}{b} H_t^{(j)} \right) U$
  end for
  $U \leftarrow U + U_{\text{prev}}$
end for

We prove that Algorithm 2 returns the first-order RL-IF estimator. Recall the forward perturbation recursion in (7). For fixed-minibatch updates, the sampling term is inactive, which is equivalently captured by setting $B_t^{(j)} = 0$. The target scalar is $\nabla_\theta f(\theta_T^{(M)})^\top \Delta_T^{(M)}$. We initialize

$$U_T^{(M)} = \nabla_\theta f(\theta_T^{(M)}).$$

Suppose that, before processing update (t, j) in the backward pass, the current adjoint vector is $U_t^{(j)}$. Then, using (7), we have

$$\begin{aligned}\left(U_t^{(j)}\right)^\top \Delta_t^{(j)} &= \left(U_t^{(j)}\right)^\top \left(A_t^{(j)}\Delta_t^{(j-1)} + B_t^{(j)}\Delta_t^{(0)}\right) \\ &= \left(A_t^{(j)\top}U_t^{(j)}\right)^\top \Delta_t^{(j-1)} + \left(B_t^{(j)\top}U_t^{(j)}\right)^\top \Delta_t^{(0)}. \end{aligned} \tag{12}$$

Therefore, the adjoint contribution to the previous minibatch iterate is $U_t^{(j-1)} = A_t^{(j)\top}U_t^{(j)}$, which gives the optimization update:

$$U \leftarrow A_t^{(j)\top}U = \left(I - \frac{\eta_t}{b}H_t^{(j)}\right)U,$$

where we use the symmetry of $H_t^{(j)}$. The second term in (12) contributes to the previous-stage perturbation $\Delta_t^{(0)}$ rather than to $\Delta_t^{(j-1)}$. Hence, during the backward sweep over stage t, all such cross-stage contributions are accumulated as

$$U_{\text{prev}} \leftarrow U_{\text{prev}} + B_t^{(j)\top}U.$$

After all updates in stage t have been processed, the adjoint propagated through the optimization dynamics and the accumulated sampling contribution both act on the same previous-stage perturbation $\Delta_t^{(0)}$. Thus they are combined by

$$U \leftarrow U + U_{\text{prev}}.$$

By repeated application of (12), the algorithm maintains the invariant that the current adjoint U satisfies

$$\nabla_\theta f(\theta_T^{(M)})^\top \Delta_T^{(M)} = U^\top \Delta_t^{(j)} + U_{\text{prev}}^\top \Delta_t^{(0)}$$

whenever the backward pass is positioned at iterate $\theta_t^{(j)}$. In particular, when the backward pass reaches the injection point (k, l), $\Delta_k^{(0)} = 0$, so we have $\nabla_\theta f(\theta_T^{(M)})^\top \Delta_T^{(M)} = U^\top \Delta_k^{(l)}$. The injected perturbation is $\Delta_k^{(l)} = \frac{\eta_k}{b}\nabla_\theta \ell(\theta_k^{(l-1)}, z^*)$. Substituting this expression gives

$$\nabla_\theta f(\theta_T^{(M)})^\top \Delta_T^{(M)} = \left\langle U,\ \frac{\eta_k}{b}\nabla_\theta \ell(\theta_k^{(l-1)}, z^*)\right\rangle.$$

This is exactly the value returned by Algorithm 2. Therefore, Algorithm 2 correctly computes the first-order RL-IF estimator for the RL-LOO effect.

A.3 Closed-Form RL-IF: Proof of Equation (8)

In this section, we show that the recursion in (7) directly leads to a compact closed form. For each stage $t > k$, define

$$A_t := \prod_{j=M}^{1} A_t^{(j)}, \qquad C_t := A_t + \sum_{q=1}^{M}\left(\prod_{j=M}^{q+1} A_t^{(j)}\right) B_t^{(q)}. \tag{13}$$

Here, products are ordered from right to left along the forward trajectory; in particular,

$$\prod_{j=M}^{1} A_t^{(j)} = A_t^{(M)}A_t^{(M-1)}\cdots A_t^{(1)}.$$

Then

$$\text{RL-IF}^{(k,l)}(z^*; f) = \nabla_\theta f(\theta_T^{(M)})^\top \left(\prod_{t=T}^{k+1} C_t\right)\left(\prod_{j=M}^{l+1} A_k^{(j)}\right)\frac{\eta_k}{b}\nabla_\theta \ell(\theta_k^{(l-1)}, z^*). \tag{14}$$

Proof. We prove the closed form in (14) by induction.

First, consider a future stage $t > k$. Recall $\Delta_t^{(j)} = A_t^{(j)}\Delta_t^{(j-1)} + B_t^{(j)}\Delta_t^{(0)}$. We claim that, for every $r = 1, \ldots, M$,

$$\Delta_t^{(r)} = \left(\prod_{j=r}^{1} A_t^{(j)}\right)\Delta_t^{(0)} + \sum_{q=1}^{r}\left(\prod_{j=r}^{q+1} A_t^{(j)}\right) B_t^{(q)}\Delta_t^{(0)}, \tag{15}$$

where an empty product is the identity. For $r = 1$, (15) gives

$$\Delta_t^{(1)} = A_t^{(1)}\Delta_t^{(0)} + B_t^{(1)}\Delta_t^{(0)},$$

which is exactly (7). Assume (15) holds for some $r - 1$. Then

$$\begin{aligned}
\Delta_t^{(r)} &= A_t^{(r)}\Delta_t^{(r-1)} + B_t^{(r)}\Delta_t^{(0)} \\
&= A_t^{(r)}\left[\left(\prod_{j=r-1}^{1} A_t^{(j)}\right)\Delta_t^{(0)} + \sum_{q=1}^{r-1}\left(\prod_{j=r-1}^{q+1} A_t^{(j)}\right) B_t^{(q)}\Delta_t^{(0)}\right] + B_t^{(r)}\Delta_t^{(0)} \\
&= \left(\prod_{j=r}^{1} A_t^{(j)}\right)\Delta_t^{(0)} + \sum_{q=1}^{r-1}\left(\prod_{j=r}^{q+1} A_t^{(j)}\right) B_t^{(q)}\Delta_t^{(0)} + B_t^{(r)}\Delta_t^{(0)} \\
&= \left(\prod_{j=r}^{1} A_t^{(j)}\right)\Delta_t^{(0)} + \sum_{q=1}^{r}\left(\prod_{j=r}^{q+1} A_t^{(j)}\right) B_t^{(q)}\Delta_t^{(0)}.
\end{aligned}$$

Thus the claim holds for r, and therefore for all $r = 1, \ldots, M$. Setting $r = M$ gives

$$\Delta_t^{(M)} = \left(\prod_{j=M}^{1} A_t^{(j)}\right)\Delta_t^{(0)} + \sum_{q=1}^{M}\left(\prod_{j=M}^{q+1} A_t^{(j)}\right) B_t^{(q)}\Delta_t^{(0)}.$$

Factoring out $\Delta_t^{(0)}$, we obtain

$$\Delta_t^{(M)} = \left[\prod_{j=M}^{1} A_t^{(j)} + \sum_{q=1}^{M}\left(\prod_{j=M}^{q+1} A_t^{(j)}\right) B_t^{(q)}\right]\Delta_t^{(0)} = C_t\Delta_t^{(0)}. \tag{16}$$

Next, for the injection stage k, the perturbation after (k, l) is $\Delta_k^{(l)} = \frac{\eta_k}{b}\nabla_\theta \ell(\theta_k^{(l-1)}, z^*)$. For $j = l + 1, \ldots, M$, the minibatches are fixed, so $\Delta_k^{(j)} = A_k^{(j)}\Delta_k^{(j-1)}$. By another direct induction over j, this gives

$$\Delta_k^{(M)} = \left(\prod_{j=M}^{l+1} A_k^{(j)}\right)\Delta_k^{(l)}. \tag{17}$$

Finally, applying (16) inductively over stages $t = k + 1, \ldots, T$ yields

$$\Delta_T^{(M)} = \left(\prod_{t=T}^{k+1} C_t\right)\Delta_k^{(M)}.$$

Substituting (17) and the expression for $\Delta_k^{(l)}$ gives

$$\Delta_T^{(M)} = \left(\prod_{t=T}^{k+1} C_t\right)\left(\prod_{j=M}^{l+1} A_k^{(j)}\right)\frac{\eta_k}{b}\nabla_\theta \ell(\theta_k^{(l-1)}, z^*).$$

Multiplying both sides by $\nabla_\theta f(\theta_T^{(M)})^\top$ proves

$$\text{RL-IF}^{(k,l)}(z^*; f) = \nabla_\theta f(\theta_T^{(M)})^\top\left(\prod_{t=T}^{k+1} C_t\right)\left(\prod_{j=M}^{l+1} A_k^{(j)}\right)\frac{\eta_k}{b}\nabla_\theta \ell(\theta_k^{(l-1)}, z^*).$$

This completes the proof.

A.4 Construction and Justification of the Reference Target

For an attribution window starting at round k, we use its terminal checkpoint $\bar{\theta} := \theta_{k+H}$ as the reference policy. We collect an on-policy rollout D_{k+H} from $\pi_{\bar{\theta}}$ and compute the reference advantages using the critic associated with this checkpoint and the same generalized advantage estimation (GAE) procedure and hyperparameters as PPO. In particular, writing V^{ref} for the reference critic, the estimates take the form

$$\delta_t^{\mathrm{ref}} = r_t + \gamma V^{\mathrm{ref}}(s_{t+1}) - V^{\mathrm{ref}}(s_t), \tag{18}$$

$$\hat{A}_t^{\mathrm{ref}} = \sum_{j=0}^{L_t-1} (\gamma\lambda)^j \delta_{t+j}^{\mathrm{ref}}, \tag{19}$$

where L_t is the number of available steps used in the estimate, with the same termination masking and truncation bootstrapping as in PPO. The reference rollout, critic, and advantage estimates are held fixed during attribution. This construction does not involve tuning over candidate reference advantages.

The terminal checkpoint is used because RL-LOO evaluates the effect of removing a training occurrence on the policy obtained at the end of the window. The following result explains the connection between the reference target and the expected return of that policy.

Ideal reference target. For the formal statement, let $0 \le \gamma < 1$ and define the normalized discounted return and discounted state occupancy by

$$J(\theta) := (1-\gamma)\mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t r_t\right], \tag{20}$$

$$d^{\pi_\theta}(s) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr_{\pi_\theta}(s_t = s). \tag{21}$$

The initial-state distribution is fixed. Let $A^{\pi_{\bar{\theta}}}$ denote the exact advantage for the usual discounted return. Following the negative-target convention in the main text, define

$$f_{\mathrm{ref}}(\phi) := -\mathbb{E}_{\substack{s\sim d^{\pi_{\bar{\theta}}} \\ a\sim\pi_{\bar{\theta}}(\cdot|s)}}\left[\log \pi_\phi(a \mid s) A^{\pi_{\bar{\theta}}}(s,a)\right]. \tag{22}$$

The reference distribution and advantages in this expression are fixed as ϕ varies. This normalization makes the return-gradient identity explicit; the corresponding statement for unnormalized discounted return differs by the constant factor $1-\gamma$.

Proposition A.1: Reference target and terminal return

Assume the policy-gradient identity holds and that f_{ref} and J are twice continuously differentiable in a neighborhood of $\bar{\theta}$. Let $\bar{\theta}_{-z}$ be the terminal parameter after deleting occurrence z, and set $\Delta_z := \bar{\theta}_{-z} - \bar{\theta}$. For sufficiently small Δ_z,

$$\mathrm{RL\text{-}LOO}(z; f_{\mathrm{ref}}) = J(\bar{\theta}) - J(\bar{\theta}_{-z}) + O(\|\Delta_z\|_2^2). \tag{23}$$

Proof. Since the reference distribution and advantages are fixed, the policy-gradient theorem gives

$$\nabla_\phi f_{\mathrm{ref}}(\phi)|_{\phi=\bar{\theta}} = -\mathbb{E}_{\substack{s\sim d^{\pi_{\bar{\theta}}} \\ a\sim\pi_{\bar{\theta}}(\cdot|s)}}\left[\nabla_\theta \log \pi_{\bar{\theta}}(a \mid s) A^{\pi_{\bar{\theta}}}(s,a)\right] \tag{24}$$

$$= -\nabla J(\bar{\theta}). \tag{25}$$

Taylor expansion at $\bar{\theta}$ therefore yields

$$f_{\mathrm{ref}}(\bar{\theta}_{-z}) - f_{\mathrm{ref}}(\bar{\theta}) = -\nabla J(\bar{\theta})^\top \Delta_z + O(\|\Delta_z\|_2^2), \tag{26}$$

$$J(\bar{\theta}_{-z}) - J(\bar{\theta}) = \nabla J(\bar{\theta})^\top \Delta_z + O(\|\Delta_z\|_2^2). \tag{27}$$

Combining the two expansions proves (23).

Thus, with this negative target, a negative RL-LOO value locally corresponds to an increase in terminal return after deletion. RL-IF estimates this counterfactual through the first-order parameter

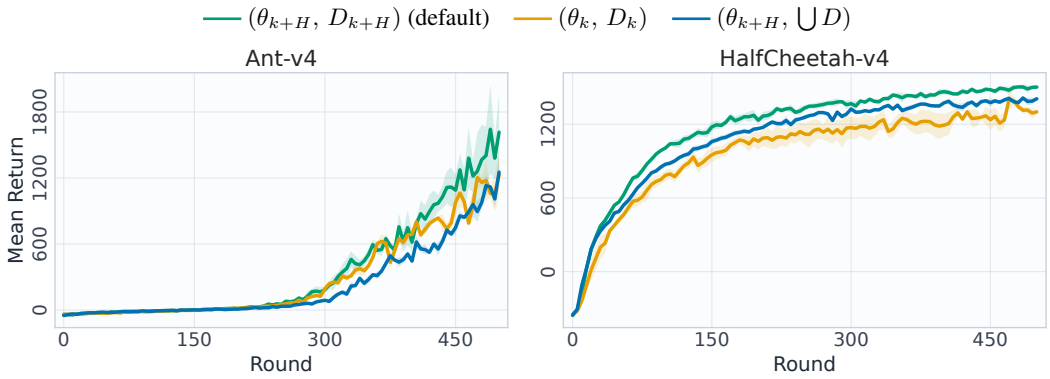


Figure 5: Reference-construction ablation on Ant and HalfCheetah. Curves show mean return at the reported training rounds; shaded regions indicate one standard error.

perturbation. In practice, we replace the population target and exact advantages with the empirical reference rollout and GAE estimates, introducing sampling and advantage-estimation error in addition to the first-order approximation. Sensitivity to the reference construction is evaluated in Appendix A.4.1.

A.4.1 Choice of Reference Policy and Rollout

We examine sensitivity to the reference construction by comparing the default terminal checkpoint and its own rollout, (θ_{k+H}, D_{k+H}), with two alternatives. The first uses the initial checkpoint of the attribution window and its corresponding rollout, (θ_k, D_k), following the IIF-style reference construction. The second retains the terminal checkpoint but uses the union of rollout data collected within the window. Although the latter uses more reference data, these rollouts were generated by earlier policies rather than solely by the terminal policy.

Figure 5 shows that the default reference achieves the highest final mean return on both environments. These results support pairing the terminal reference policy with its own on-policy rollout.

B Detailed Theorem Derivations

In this section, we provide detailed proofs for the theorems.

B.1 Uniform Approximation Error of First-Order RL-IF

Theorem B.1: Uniform approximation error of first-order RL-IF

Assume that $\ell(\theta; x)$ is twice differentiable, that its gradient norm is bounded by G, that its Hessian norm is bounded by Λ, and that its Hessian is L_H-Lipschitz continuous in θ. Assume also that the rollout distribution $P_t(x \mid \theta)$ has score norm bounded by S and score Hessian norm bounded by L_P. Furthermore, assume that the learning rate satisfies $M\overline{\Lambda}\eta_t \le \frac{1}{T-k}$ for every $k \le t \le T$. Then for the first-order RL-IF estimator $\Delta_T^{(M)}$, the approximation satisfies

$$\left\|\widetilde{\theta}_T^{(M)} - \theta_T^{(M)} - \Delta_T^{(M)}\right\|_2 \le \frac{C_{\text{smooth}}}{M^2 b^2},$$

where $C_{\text{smooth}} = \frac{cG^2(L_H + \overline{\Lambda}S + GL_P)}{\overline{\Lambda}^3}$ and $\overline{\Lambda} = \Lambda + GS$ are smoothness constants.

Proof. We analyze the population update map and we do not consider minibatch noise in this theorem. We define the true counterfactual perturbation as

$$\delta_t^{(j)} := \widetilde{\theta}_t^{(j)} - \theta_t^{(j)}.$$

The first-order RL-IF perturbation $\Delta_t^{(j)}$ is defined by (5) and satisfies the recursion in Proposition 3.1. At the injection point (k, l), the two coincide exactly:

$$\delta_k^{(l)} = \Delta_k^{(l)} = \frac{\eta_k}{b} \nabla_\theta \ell(\theta_k^{(l-1)}, z^*).$$

Our goal is to bound the discrepancy

$$e_t^{(j)} := \delta_t^{(j)} - \Delta_t^{(j)}$$

at the terminal point (T, M). Recall the stage update map $F_t^{(j)}(\vartheta, \beta)$ defined in (6). By construction, the post-intervention trajectory is generated by

$$\theta_t^{(j)}(\varepsilon) = F_t^{(j)}\left(\theta_t^{(j-1)}(\varepsilon), \theta_t^{(0)}(\varepsilon)\right).$$

Since $\ell(\theta; x)$ has an L_H-Lipschitz Hessian, and the rollout distribution $P_t(x \mid \theta)$ has score norm bounded by S and score Hessian norm bounded by L_P, the map $F_t^{(j)}(\vartheta, \beta)$ is jointly smooth in (ϑ, β). We next prove a bound for the remainder:

$$\left\| F_t^{(j)}(\vartheta + u, \beta + v) - F_t^{(j)}(\vartheta, \beta) - \partial_\vartheta F_t^{(j)}(\vartheta, \beta) u - \partial_\beta F_t^{(j)}(\vartheta, \beta) v \right\|_2 \le \frac{\eta_t K}{2} (\|u\|_2 + \|v\|_2)^2, \tag{28}$$

where $K := L_H + \overline{\Lambda} S + G L_P, \overline{\Lambda} := \Lambda + GS$. We denote $m_t^{(j)}(\vartheta, \beta) = \mathbb{E}_{x \sim P_t(\cdot|\beta)}[\nabla_\theta \ell(\vartheta, x)]$, for a future round $t > k$, write $m = m_t^{(j)}$ and define

$$R_m(u, v) := m(\vartheta + u, \beta + v) - m(\vartheta, \beta) - \partial_\vartheta m(\vartheta, \beta) u - \partial_\beta m(\vartheta, \beta) v.$$

We decompose this remainder as

$$\begin{aligned} R_m(u, v) = & \underbrace{m(\vartheta + u, \beta + v) - m(\vartheta, \beta + v) - \partial_\vartheta m(\vartheta, \beta + v) u}_{\text{(I)}} \\ & + \underbrace{\big(\partial_\vartheta m(\vartheta, \beta + v) - \partial_\vartheta m(\vartheta, \beta)\big) u}_{\text{(II)}} \\ & + \underbrace{m(\vartheta, \beta + v) - m(\vartheta, \beta) - \partial_\beta m(\vartheta, \beta) v}_{\text{(III)}}. \end{aligned}$$

Bounding (I). For fixed $\beta + v$,

$$\partial_\vartheta m(\vartheta, \beta + v) = \mathbb{E}_{x \sim P_t(\cdot|\beta+v)}\big[\nabla_\theta^2 \ell(\vartheta, x)\big].$$

Since the loss Hessian is L_H-Lipschitz, we have

$$\|(\mathrm{I})\|_2 \le \frac{L_H}{2} \|u\|_2^2.$$

Bounding (II). Using the score-function identity, the Hessian bound Λ, and the score bound S, we obtain

$$\big\|\partial_\vartheta m(\vartheta, \beta + v) - \partial_\vartheta m(\vartheta, \beta)\big\|_2 \le \Lambda S \|v\|_2.$$

Consequently,

$$\|(\mathrm{II})\|_2 \le \Lambda S \|u\|_2 \|v\|_2.$$

Bounding (III). The sampling derivative is

$$\partial_\beta m(\vartheta, \beta) = \mathbb{E}_{x \sim P_t(\cdot|\beta)} \left[\nabla_\theta \ell(\vartheta, x) \nabla_\beta \log P_t(x \mid \beta)^\top\right].$$

Differentiating once more in β produces a score-Hessian term and a score outer-product term. The gradient bound G, score bound S, and score-Hessian bound L_P therefore imply

$$\|(\mathrm{III})\|_2 \le \frac{G(L_P + S^2)}{2} \|v\|_2^2.$$

Combining the three bounds yields

$$\begin{aligned}\|R_m(u,v)\|_2 &\le \frac{L_H}{2}\|u\|_2^2 + \Lambda S\|u\|_2\|v\|_2 + \frac{G(L_P+S^2)}{2}\|v\|_2^2 \\ &\le \frac{K}{2}(\|u\|_2+\|v\|_2)^2,\end{aligned}$$

where $K = L_H + \Lambda S + G(L_P + S^2)$. Since $F_t^{(j)}(\vartheta,\beta) = \vartheta - \eta_t m_t^{(j)}(\vartheta,\beta)$, this proves (28) for $t > k$. For the remaining updates of round k, the map is independent of β, and the fixed-minibatch average has an L_H-Lipschitz Jacobian. Its remainder is bounded by $\eta_k L_H\|u\|_2^2/2$, so (28) holds there as well. Applying (28) with $u = \delta_t^{(j-1)}$ and $v = \delta_t^{(0)}$ gives

$$\delta_t^{(j)} = A_t^{(j)}\delta_t^{(j-1)} + B_t^{(j)}\delta_t^{(0)} + r_t^{(j)},$$

where

$$\|r_t^{(j)}\|_2 \le \frac{\eta_t K}{2}\left(\|\delta_t^{(j-1)}\|_2 + \|\delta_t^{(0)}\|_2\right)^2. \tag{29}$$

Subtracting the first-order recursion yields

$$e_t^{(j)} = A_t^{(j)}e_t^{(j-1)} + B_t^{(j)}e_t^{(0)} + r_t^{(j)}. \tag{30}$$

This also applies to the remaining updates of round k, with $B_k^{(j)} = 0$. We next control the perturbation size. The uniform Jacobian bounds are

$$\|\partial_\vartheta F_t^{(j)}(\vartheta,\beta)\|_2 \le 1 + \eta_t\Lambda, \qquad \|\partial_\beta F_t^{(j)}(\vartheta,\beta)\|_2 \le \eta_t GS.$$

Integrating these bounds along the line segment between the two inputs gives

$$\|F_t^{(j)}(\vartheta+u,\beta+v) - F_t^{(j)}(\vartheta,\beta)\|_2 \le (1+\eta_t\Lambda)\|u\|_2 + \eta_t GS\|v\|_2.$$

For round k, the second term can be omitted. Thus both the true perturbation and its first-order approximation grow by at most a factor $q_t := 1 + \eta_t\overline{\Lambda}$ per update, when measured relative to the maximum perturbation encountered so far. The learning-rate assumption implies

$$\sum_{t=k}^{T} M\eta_t\overline{\Lambda} \le \frac{T-k+1}{T-k} \le 2.$$

Starting from the injection bound $\|\delta_k^{(l)}\|_2 = \|\Delta_k^{(l)}\|_2 \le \eta_k G/b$, we therefore obtain

$$D := \max_{(t,j)\text{ at or after }(k,l)} \left\{\|\delta_t^{(j)}\|_2, \|\Delta_t^{(j)}\|_2\right\} \le \frac{\eta_k G}{b}\prod_{t=k}^{T} q_t^M \le \frac{e^2\eta_k G}{b} \le \frac{e^2 G}{M\overline{\Lambda}b}. \tag{31}$$

Here we used $T - k \ge 1$ in the last inequality. The sampling perturbation at the start of round k is zero. Consequently, (29) gives, for every update after the injection,

$$\|r_t^{(j)}\|_2 \le R_t, \qquad R_t := 2\eta_t K D^2.$$

Define

$$E_k := \max_{l\le j\le M}\|e_k^{(j)}\|_2, \qquad E_t := \max_{0\le j\le M}\|e_t^{(j)}\|_2 \quad (t>k),$$

and let $a_t := M\eta_t\overline{\Lambda}$. Taking norms in (30) and applying the triangle inequality yields

$$\|e_t^{(j)}\|_2 \le q_t \max\left\{\|e_t^{(j-1)}\|_2, \|e_t^{(0)}\|_2\right\} + R_t.$$

For $t > k$, using $e_t^{(0)} = e_{t-1}^{(M)}$ and iterating within the round gives

$$\begin{aligned}E_t &\le q_t^M E_{t-1} + R_t\sum_{i=0}^{M-1} q_t^i \\ &\le e^{a_t}\big(E_{t-1} + MR_t\big).\end{aligned}$$

In round k, we instead start from $e_k^{(l)} = 0$. The remaining $M - l$ updates satisfy

$$E_k \le R_k \sum_{i=0}^{M-l-1} q_k^i \le M e^{a_k} R_k,$$

where the sum is empty when $l = M$. In particular, the error accumulated during round k is included. Unrolling the round-wise bounds therefore gives

$$\begin{aligned} E_T &\le M \sum_{t=k}^{T} R_t \exp\left(\sum_{s=t}^{T} a_s\right) \\ &\le 2e^2 M K D^2 \sum_{t=k}^{T} \eta_t. \end{aligned}$$

Finally,

$$\sum_{t=k}^{T} \eta_t \le \frac{T-k+1}{(T-k)M\overline{\Lambda}} \le \frac{2}{M\overline{\Lambda}}.$$

Combining this with (31), we conclude

$$\begin{aligned} \left\|\widetilde{\theta}_T^{(M)} - \theta_T^{(M)} - \Delta_T^{(M)}\right\|_2 &= \|e_T^{(M)}\|_2 \\ &\le E_T \\ &\le \frac{4e^2 K D^2}{\overline{\Lambda}} \\ &\le \frac{4e^6 G^2 \left(L_H + \overline{\Lambda} S + G L_P\right)}{M^2 b^2 \overline{\Lambda}^3}. \end{aligned}$$

This proves the result with the universal constant $c = 4e^6$. □

B.2 Relative Bound between Stage-Wise Operators

Theorem B.2: Nonconvex relative gap between sampling-aware and optimization-only operators

Under the smoothness and bounded-gradient assumptions in Theorem B.1, assume in addition that, for each future stage $t > k$, the sampling family $P_t(\cdot \mid \theta)$ is KL-smooth with score second moment bounded by L_t, and that $\eta_t \Lambda < 1$.
Then, for every future stage $t > k$,

$$\|C_t - A_t\|_2 \le \rho_t \|A_t\|_2, \qquad \rho_t := \frac{G\sqrt{L_t}\,\frac{(1+\eta_t\Lambda)^M - 1}{\Lambda}}{(1-\eta_t\Lambda)^M}. \tag{32}$$

Proof. Fix a future stage $t > k$. Recall that

$$C_t = A_t + \sum_{q=1}^{M} \left(\prod_{j=M}^{q+1} A_t^{(j)}\right) B_t^{(q)}, \qquad A_t = \prod_{j=M}^{1} A_t^{(j)}.$$

Therefore,

$$C_t - A_t = \sum_{q=1}^{M} \left(\prod_{j=M}^{q+1} A_t^{(j)}\right) B_t^{(q)}.$$

Taking operator norms and using submultiplicativity,

$$\|C_t - A_t\|_2 \le \sum_{q=1}^{M} \left\|\prod_{j=M}^{q+1} A_t^{(j)}\right\|_2 \|B_t^{(q)}\|_2.$$

By smoothness,

$$\|A_t^{(j)}\|_2 = \left\|I - \frac{\eta_t}{b} H_t^{(j)}\right\|_2 \le 1 + \eta_t \Lambda.$$

Thus,

$$\left\| \prod_{j=M}^{q+1} A_t^{(j)} \right\|_2 \le (1 + \eta_t \Lambda)^{M-q}.$$

By the bounded-gradient assumption and the KL-smoothness assumption,

$$\|B_t^{(q)}\|_2 \le \eta_t G \sqrt{L_t}.$$

Therefore,

$$\|C_t - A_t\|_2 \le \eta_t G \sqrt{L_t} \sum_{q=1}^{M} (1 + \eta_t \Lambda)^{M-q}.$$

Using the geometric sum,

$$\sum_{q=1}^{M} (1 + \eta_t \Lambda)^{M-q} = \frac{(1 + \eta_t \Lambda)^M - 1}{\eta_t \Lambda},$$

we obtain

$$\|C_t - A_t\|_2 \le G \sqrt{L_t} \frac{(1 + \eta_t \Lambda)^M - 1}{\Lambda}.$$

It remains to relate this bound to $\|A_t\|_2$. Since $\eta_t \Lambda < 1$, for every vector v,

$$\|A_t^{(j)} v\|_2 = \left\|v - \frac{\eta_t}{b} H_t^{(j)} v\right\|_2 \ge \|v\|_2 - \frac{\eta_t}{b} \|H_t^{(j)} v\|_2 \ge (1 - \eta_t \Lambda) \|v\|_2.$$

Applying this bound through the product gives

$$\|A_t v\|_2 \ge (1 - \eta_t \Lambda)^M \|v\|_2.$$

Hence,

$$\|A_t\|_2 \ge (1 - \eta_t \Lambda)^M.$$

Combining this with the absolute bound yields

$$\|C_t - A_t\|_2 \le \frac{G \sqrt{L_t} \frac{(1+\eta_t \Lambda)^M - 1}{\Lambda}}{(1 - \eta_t \Lambda)^M} \|A_t\|_2.$$

This proves (32). □

B.3 Accumulated Relative Gap

Theorem B.3: Accumulated nonconvex relative gap

Under the assumptions of Theorem B.2, for any $T > k$,

$$\left\| \prod_{t=T}^{k+1} C_t - \prod_{t=T}^{k+1} A_t \right\|_2 \le \left[\prod_{t=k+1}^{T} (1 + \rho_t) - 1 \right] \prod_{t=k+1}^{T} \|A_t\|_2. \tag{33}$$

Furthermore, if the learning rate takes the harmonic form

$$\eta_t = \frac{\eta_0}{M(t-k)},$$

with $\eta_0 \Lambda < 1$, and if $L_{\max} := \sup_{k<t\le T} L_t$, then, with $H := T - k$,

$$\left\| \prod_{t=T}^{k+1} C_t - \prod_{t=T}^{k+1} A_t \right\|_2 \le (e^{\alpha} H^{\alpha} - 1) \prod_{t=k+1}^{T} \|A_t\|_2, \tag{34}$$

where

$$\alpha := \frac{G \sqrt{L_{\max}} \eta_0}{(1 - \eta_0 \Lambda)^2}.$$

Proof. We first prove the general accumulated product bound. Define

$$P_C(k,T) = \prod_{t=T}^{k+1} C_t, \qquad P_A(k,T) = \prod_{t=T}^{k+1} A_t.$$

Using the telescoping expansion for products,

$$P_C(k,T) - P_A(k,T) = \sum_{s=k+1}^{T} \left(\prod_{t=T}^{s+1} C_t \right) (C_s - A_s) \left(\prod_{t=s-1}^{k+1} A_t \right).$$

Taking norms gives

$$\|P_C(k,T) - P_A(k,T)\|_2 \le \sum_{s=k+1}^{T} \left(\prod_{t=s+1}^{T} \|C_t\|_2 \right) \|C_s - A_s\|_2 \left(\prod_{t=k+1}^{s-1} \|A_t\|_2 \right).$$

By Theorem B.2,

$$\|C_s - A_s\|_2 \le \rho_s \|A_s\|_2.$$

Moreover,

$$\|C_t\|_2 \le \|A_t\|_2 + \|C_t - A_t\|_2 \le (1+\rho_t)\|A_t\|_2.$$

Substituting these inequalities gives

$$\|P_C(k,T) - P_A(k,T)\|_2 \le \sum_{s=k+1}^{T} \left(\prod_{t=s+1}^{T} (1+\rho_t)\|A_t\|_2 \right) \rho_s \|A_s\|_2 \left(\prod_{t=k+1}^{s-1} \|A_t\|_2 \right).$$

Factoring out the full product of $\|A_t\|_2$ gives

$$\|P_C(k,T) - P_A(k,T)\|_2 \le \left[\sum_{s=k+1}^{T} \rho_s \prod_{t=s+1}^{T} (1+\rho_t) \right] \prod_{t=k+1}^{T} \|A_t\|_2.$$

Finally, using the identity

$$\sum_{s=k+1}^{T} \rho_s \prod_{t=s+1}^{T} (1+\rho_t) = \prod_{t=k+1}^{T} (1+\rho_t) - 1,$$

we obtain

$$\left\| \prod_{t=T}^{k+1} C_t - \prod_{t=T}^{k+1} A_t \right\|_2 \le \left[\prod_{t=k+1}^{T} (1+\rho_t) - 1 \right] \prod_{t=k+1}^{T} \|A_t\|_2.$$

We next prove the harmonic-learning-rate refinement. From Theorem B.2,

$$\rho_t = G\sqrt{L_t} \frac{(1+\eta_t\Lambda)^M - 1}{\Lambda(1-\eta_t\Lambda)^M}.$$

Let $x_t = \eta_t\Lambda$. Since $\eta_t = \eta_0/(M(t-k))$, we have

$$Mx_t = \frac{\eta_0\Lambda}{t-k} \le \eta_0\Lambda < 1.$$

Using

$$(1+x_t)^M - 1 \le \frac{Mx_t}{1-Mx_t}, \qquad (1-x_t)^{-M} \le \frac{1}{1-Mx_t},$$

we get

$$\rho_t \le G\sqrt{L_t} \frac{M\eta_t}{(1-M\eta_t\Lambda)^2}.$$

Since

$$1 - M\eta_t\Lambda = 1 - \frac{\eta_0\Lambda}{t-k} \ge 1 - \eta_0\Lambda,$$

Table 2: Summary of RL environments used in experiments. MuJoCo environments are from Todorov et al. [2012], while Box2D and FrozenLake environments are from Gym/Gymnasium [Brockman et al., 2016, Towers et al., 2025].

Env	Env ID	Type	Action Space
Ant	`Ant-v4`	MuJoCo locomotion	Continuous
HalfCheetah	`HalfCheetah-v4`	MuJoCo locomotion	Continuous
Hopper	`Hopper-v4`	MuJoCo locomotion	Continuous
LunarLander	`LunarLander-v2`	Box2D control	Discrete
BipedalWalker	`BipedalWalker-v3`	Box2D locomotion	Continuous
FrozenLake	`FrozenLake-v1`	Tabular navigation	Discrete

we have

$$\rho_t \le \frac{G\sqrt{L_t}M\eta_t}{(1-\eta_0\Lambda)^2} = \frac{G\sqrt{L_t}\eta_0}{(1-\eta_0\Lambda)^2}\frac{1}{t-k}.$$

Using $L_t \le L_{\max}$ gives

$$\rho_t \le \frac{G\sqrt{L_{\max}}\eta_0}{(1-\eta_0\Lambda)^2}\frac{1}{t-k} = \frac{\alpha}{t-k}.$$

Therefore,

$$\sum_{t=k+1}^{T} \rho_t \le \alpha \sum_{h=1}^{H} \frac{1}{h} \le \alpha(1+\log H).$$

Finally,

$$\prod_{t=k+1}^{T}(1+\rho_t) - 1 \le \exp\left(\sum_{t=k+1}^{T}\rho_t\right) - 1 \le \exp\big(\alpha(1+\log H)\big) - 1.$$

Substituting this into (33) proves (34). □

C Detailed Experimental Setup

In this section, we provide details of the experimental setup, hyperparameter selection, and implementation details. All experiments are conducted using Amazon Web Services (AWS) c7a high-performance CPUs and 4 NVIDIA A40 GPUs, each with 46GB of memory.

C.1 Experimental Setup for Traditional RL

Environment description. Table 2 summarizes the RL environments used in our experiments. The MuJoCo environments used in this work are distributed under the Apache 2.0 License, while the Gymnasium environments and wrappers, including Box2D and FrozenLake environments, are distributed under the MIT License.

Training setup. We use `Stable-Baselines3` [Raffin et al., 2021] (MIT license) as our training framework for all the traditional RL tasks. We implement a KL-regularized PPO [Schulman et al., 2017], using KL estimated via the k_3 approximation [Schulman, 2020]

$$\widehat{\mathrm{KL}}(\pi_{\mathrm{ref}} \,\|\, \pi_\theta) = (r-1) - \log r, \quad r = \pi_\theta(a \mid s)/\pi_{\mathrm{ref}}(a \mid s)$$

We summarize the detailed hyperparameter selection in table 3. For base hyperparameters selection, we refer to RL Baselines3 Zoo[3], with some minor tuning. We perform a grid search over KL coefficients β ranging from 0.0 to 5.0, and select the value that yields the best standard training performance. Similarly, we search for the ρ that produces the highest and most stable performance for the baseline method IIF. For network architecture, we use the default `MlpPolicy` provided by `Stable-Baselines3`.

[3] `https://github.com/DLR-RM/rl-baselines3-zoo`

Table 3: PPO/training hyperparameters per environment.

Hyperparameter	Ant	HalfCheetah	LunarLander	BipedalWalker	Hopper	FrozenLake
N_{envs}	8	8	1	1	8	1
Total rollouts	500	500	500	100	100	50
n_{steps}	512	512	1024	8192	512	2048
n_{epochs}	20	20	4	10	20	10
Learning rate	1e-4	1e-4	5e-3	1e-4	1e-4	5e-3
Optimizer	Adam	Adam	SGD	Adam	SGD	SGD
γ	0.98	0.98	0.99	0.99	0.98	0.99
λ (GAE)	0.92	0.92	0.95	0.95	0.92	0.95
β_{KL}	5.0	1.0	0.5	1.0	1.0	0.0
ρ (Filtering only)	10%	10%	10%	5%	10%	50%

Evaluation setup. During evaluation, we evaluate the stochastic performance of each policy averaged over multiple evaluation episodes. For most continuous control environments (Ant, HalfCheetah, Hopper), we average over 32 episodes by default; for BipedalWalker, we use 100 episodes to reduce variance due to its longer horizon; and for light-weight environments, LunarLander and FrozenLake, we use 1000 episodes. Each setup is run with three independent seeds; we report mean $\pm$ standard error.

C.2 Experimental Setup for RLHF

Base model. We use GPT-Neo 2.7B [Black et al., 2021] (`EleutherAI/gpt-neo-2.7B`, MIT license) as the base language model. We apply LoRA [Hu et al., 2022] for parameter-efficient fine-tuning, with rank $r = 16$, scaling factor $\alpha = 32$, and dropout rate 0.05.

Training data. We train on the Real Toxicity Prompts dataset [Gehman et al., 2020] (`allenai/real-toxicity-prompts`, Apache-2.0 license). We retain examples with a prompt toxicity score above 0.3, use the 80% training split, and shuffle with a fixed seed. For each example, we concatenate the prompt and continuation fields, tokenize, and truncate to a length sampled uniformly in $[10, 15]$ tokens to form the PPO prompt. The policy then generates a 30-token continuation with temperature 1.0 and top-$p = 1.0$.

Reward model. The reward signal is provided by the RoBERTa-based hate-speech classifier `facebook/roberta-hate-speech-dynabench-r4-target` [Vidgen et al., 2021]. Following the reference implementation, we use the raw logit of the `nothate` class (index 0) as the scalar reward: a higher logit indicates less toxic output.

Training setup. We optimize a PPO objective with a fixed KL penalty and no probability-ratio clipping, with a learning rate of 5×10^{-6}. The per-token KL divergence from the reference policy is penalized with coefficient $\beta = 0.4$, estimated via the k_1 approximation [Schulman, 2020]. Each rollout collects a batch of 64 samples; PPO updates use a mini-batch size of 8 over 4 epochs, yielding 32 gradient steps per rollout. We use the Adam optimizer with gradient norm clipping at 1.0 and no weight decay. For each run, we train for 300 rollout steps. All runs use bfloat16 precision with Flash Attention 2 [Dao, 2024].

For data attribution setups, we use $H = 2$ for RINSE, and $\rho = 10\%$ for both IIF and RINSE for fairness.

Evaluation setup. We evaluate on the Wiki-Toxic dataset [Wulczyn et al., 2017] (`OxAISH-AL-LLM/wiki_toxic`, test split), drawn from a different distribution than the training data. We select the first 200 toxic examples (label $= 1$) and use up to the first $2{,}000$ characters of each comment as the evaluation prompt—substantially longer than training prompts and thus more likely to elicit toxic continuations. Continuations of 30 tokens are generated under the same sampling parameters as training. To guard against reward hacking specific to the training signal, we score continuations with a *separate* classifier, `DaNLP/da-electra-hatespeech-detection` [Pauli et al.,

2021], and report the mean toxicity probability across the 200 samples. Each method is run with three independent seeds; we report mean ± standard error.

D Additional Experimental Results

D.1 Further Details on Linear Data Modeling Scores and Evaluation

Directly validating attribution methods using single-sample leave-one-out (LOO) correlation can be challenging in online RL due to the extremely low signal-to-noise ratio of individual sample effects. In particular, removing a single rollout record often induces only a negligible change in the final policy behavior, while stochasticity from environment interaction introduces substantial variance across retraining runs. To obtain a more stable evaluation protocol, we additionally adopt the linear data modeling score (LDS), which evaluates whether attribution scores can linearly predict the effect of removing a subset of training data. Concretely, LDS samples random subsets of rollout records, sums the attribution scores within each subset, and compares the aggregated score against the actual performance change after retraining without that subset. This evaluation implicitly assumes approximate linear additivity of data influence across samples, and therefore measures how well attribution scores behave as a linear proxy for collective data influence.

Experimental setup. For the LDS evaluation, we consider model training with KL-regularized PPO on two consecutive rounds. We construct 100 random subsets of training data, each containing 4096 samples. For each subset, we remove it and retrain the model with future rollout data resampled using the updated policy for 32 independent runs, and compute the average change in validation loss. All results are further averaged over 3 random seeds. We conduct the experiments on two MuJoCo benchmark environments [Todorov et al., 2012], `Ant-v4` and `HalfCheetah-v4`.

Table 4: Linear data modeling scores of attribution methods on `Ant-v4` and `HalfCheetah-v4`.

	Ant-v4		HalfCheetah-v4	
Score	Pearson ↑	Spearman ↑	Pearson ↑	Spearman ↑
Random	-0.020 ± 0.041	-0.031 ± 0.058	-0.069 ± 0.037	-0.063 ± 0.032
TracIn	0.029 ± 0.081	0.055 ± 0.102	0.166 ± 0.043	0.138 ± 0.037
RL-IF-opt	0.179 ± 0.017	0.191 ± 0.037	0.266 ± 0.015	0.218 ± 0.027
RL-IF-full	0.180 ± 0.023	0.192 ± 0.033	0.267 ± 0.016	0.218 ± 0.026

Table 4 reports LDS results on `Ant-v4` and `HalfCheetah-v4`. We additionally report Pearson correlations for completeness, which exhibit the same overall trend as the Spearman correlations in the main text. Both RL-IF-opt and RL-IF-full substantially outperform TracIn and random baselines across environments and correlation metrics, indicating that the proposed non-local attribution framework better captures the long-term effect of rollout data on downstream policy behavior.

D.1.1 Direct Leave-One-Out Evaluation

To complement the subset-removal LDS evaluation, we directly evaluate LOO attribution on FrozenLake with attribution horizon $H = 2$. Unlike the fixed-rollout experiment reported separately, this evaluation allows future rollout data to change with the perturbed policy. We use 256 retraining runs to estimate the deletion effects and compare them with the attribution scores of TracIn, RL-IF-opt, and RL-IF-full.

Table 5: Direct LOO attribution evaluation on FrozenLake with $H = 2$. Correlations are reported as per-seed means with standard errors.

Method	Pearson ρ (mean ± SE)
TracIn	$+0.538 \pm 0.165$
RL-IF-opt	$+0.461 \pm 0.161$
RL-IF-full	$\mathbf{+0.687 \pm 0.041}$

RL-IF-full achieves the highest mean correlation in this experiment, outperforming both RL-IF-opt and TracIn. This provides a setting in which explicitly accounting for policy-induced sampling changes improves agreement with the estimated LOO effects.

D.1.2 Empirical Characterization of the Sampling Effect

To examine why the sampling contribution differs across environments, we measure the top Hessian eigenvalue of the KL-regularized loss, denoted by Λ, the rollout-score second moment, summarized by $\sqrt{L_{\max}}$, and the learning rate η within the attribution window. Table 6 reports these measurements and the diagnostic ratio $\eta\sqrt{L_{\max}}/\Lambda$.

Table 6: Empirical diagnostics of the sampling contribution.

Environment	Λ	$\sqrt{L_{\max}}$	η	$\eta\sqrt{L_{\max}}/\Lambda$
FrozenLake	1.3	2.4	5×10^{-3}	9.2×10^{-3}
HalfCheetah	65	12.0	1×10^{-4}	1.8×10^{-5}
Ant	2.4×10^{3}	12.2	1×10^{-4}	5.0×10^{-7}

FrozenLake has a much smaller measured curvature scale and a larger learning rate than Ant and HalfCheetah. Its reported diagnostic ratio is approximately four orders of magnitude larger than Ant's. Together with the direct LOO comparison in Table 5, these measurements identify FrozenLake as an empirical setting in which retaining the sampling correction is useful. In contrast, the full and optimization-only estimators have nearly identical LDS on Ant and HalfCheetah.

D.2 Additional Results for LOO

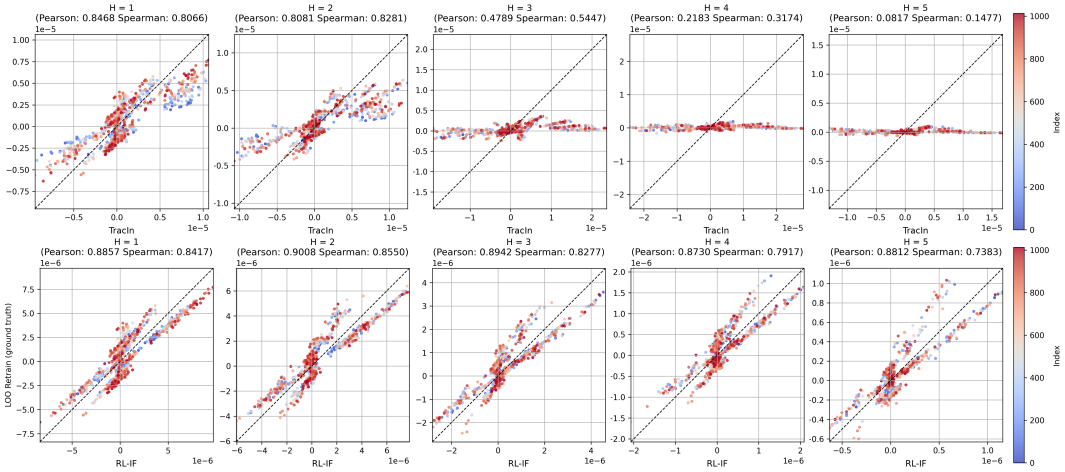


Figure 6: Comparison between TracIn and RL-IF. The top row shows the comparison between the TracIn method and the ground-truth LOO values, while the bottom row shows the comparison between RL-IF and the ground-truth LOO values under the same held-out settings. From left to right, the panels show the attribution horizon H from 1 to 5. The colorbar (point index) is shared across each row and shown once on the right. Each panel keeps its own axes because the attribution and LOO magnitudes differ across horizons.

Additional LOO results are presented in Figure 6. We compare our RL-IF score with TracIn and ground truth (LOO Retrain), under the fixed rollout setting, where we do not consider the sampling effect. RL-IF shows high correlation with the ground truth LOO retrain, with correlation decaying slowly with respect to the attribution horizon. TracIn shows high correlation with ground truth when attribution horizon equals to 1, but fails to capture the long horizon effect.

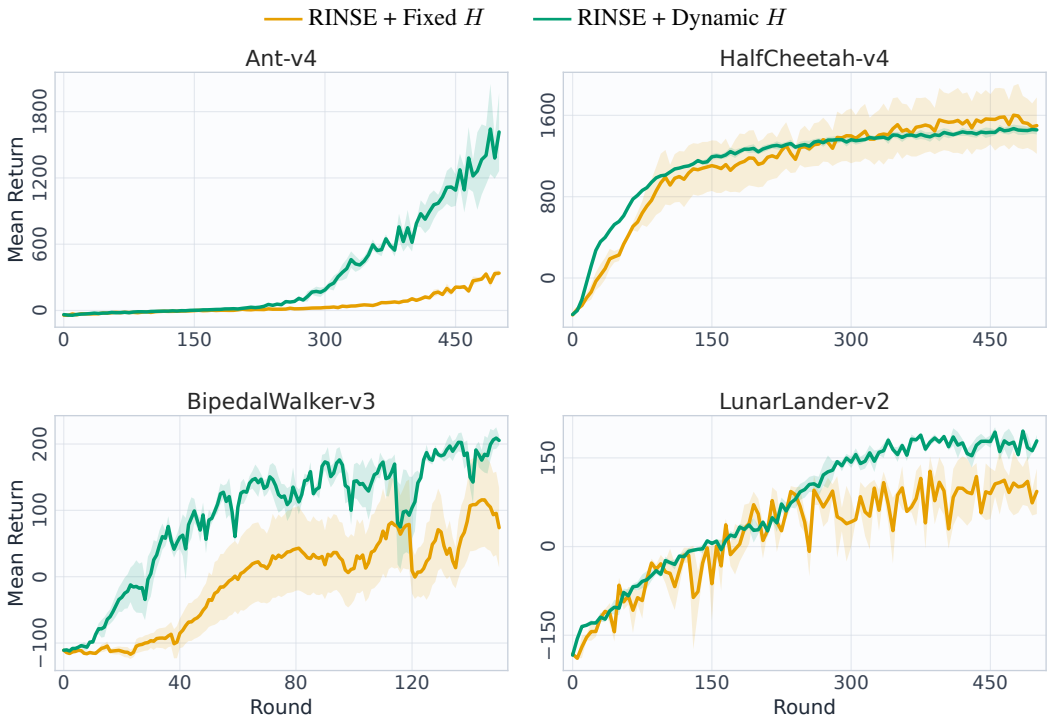


Figure 7: **Dynamic-horizon ablation.** Fixed horizon vs. the adaptive (dynamic) horizon. The dynamic horizon matches or outperforms the fixed variant on all four environments, with the largest gains on the harder-exploration tasks.

D.3 Ablation on Algorithmic Design

In this section, we conduct ablation studies to justify our design choices of the horizon selection H in RINSE.

D.3.1 Ablation on Adaptive Horizon

We ablate the effect of the adaptive horizon by adopting a fixed horizon instead, on `Ant-v4`, `HalfCheetah-v4`, `BipedalWalker-v3`, and `LunarLander-v2`. As shown in Figure 7, the adaptive horizon matches or exceeds the fixed variant in every case, and the gap is largest on the harder-exploration tasks: on `Ant-v4` and `BipedalWalker-v3` the fixed-horizon runs progress slowly and erratically, while on `LunarLander-v2` the adaptive horizon reaches a higher return with lower variance. This shows that the adaptive horizon is necessary: it stabilizes training and yields consistently higher-quality data selection without per-environment tuning.

D.3.2 Effect of Horizon in RINSE

We sweep the horizon over $H \in \{1, 2, 4, 5, 8, 10\}$ on `Ant-v4` and `HalfCheetah-v4`, holding all other hyperparameters fixed. Figure 8 shows a consistent trend that emerges across both environments. As the horizon H grows, the quality of the selected data improves gradually. This improvement, however, saturates at a moderate horizon of around $H = 5$. Beyond this point, additional horizon brings no further benefit, and for the largest values it can even slightly degrade selection quality. This motivates some medium horizons, such as $H = 5$, as a robust default.

D.3.3 Effect of the Discard Ratio

We ablate the discard ratio ρ on `Ant-v4` and `HalfCheetah-v4`, sweeping $\rho \in \{0\%, 5\%, 10\%, 20\%, 50\%, 100\%\}$, where $\rho = 0\%$ recovers standard PPO (no filtering) and $\rho = 10\%$ is our default. Figure 9 shows that filtering improves performance across the tested ratios on `Ant-v4`, while `HalfCheetah-v4` is more sensitive to aggressive filtering. On `Ant-v4`, any non-zero discard

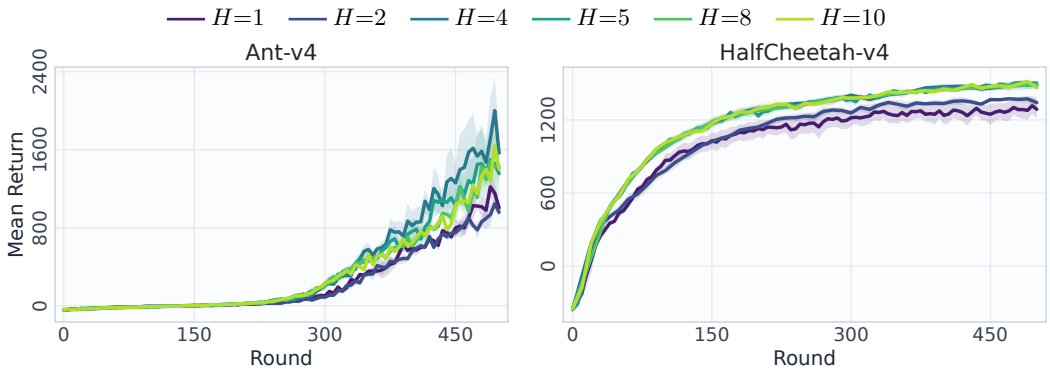


Figure 8: **Horizon ablation.** Effect of the different horizons $H \in \{1, 2, 4, 5, 8, 10\}$. Longer horizons speed up early learning, but the benefit saturates around $H \approx 4$–5.

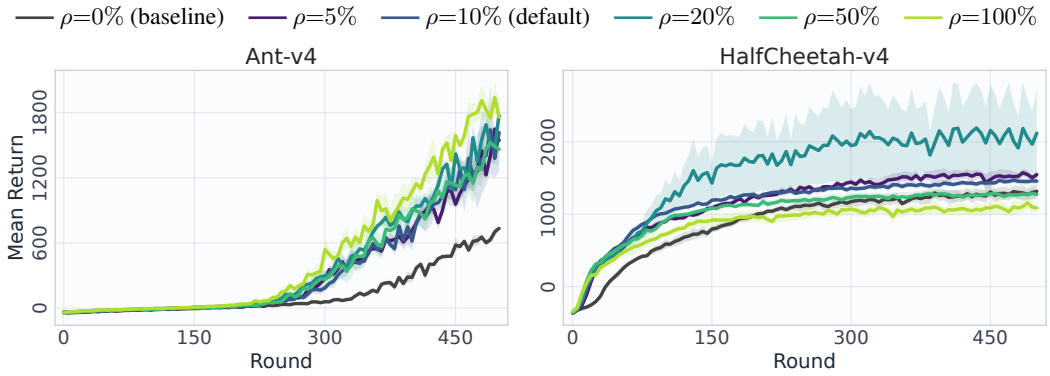


Figure 9: **Discard-ratio ablation.** Effect of the discard ratio ρ on `Ant-v4` and `HalfCheetah-v4`. Moderate ratios ($\rho \approx 10\%$–20%) are best, while very aggressive filtering can degrade performance.

ratio substantially outperforms the no-selection baseline, confirming that removing low-influence data is beneficial. On `HalfCheetah-v4`, $\rho = 50\%$ and $\rho = 100\%$ yield lower final mean returns than the no-selection baseline. These results support using a moderate discard ratio.

D.4 Further Details on Time and Sample Efficiency

We provide additional details on the computation of time efficiency (TE) and sample efficiency (SE), along with extended results. As described in the main text, both metrics are defined relative to the standard PPO baseline, which is normalized to 100%. Lower values therefore indicate better efficiency.

Evaluation protocol. For each method, we track the training trajectory and record the first time step at which a set of target performance levels is reached. In the main results, we use three thresholds corresponding to 90%, 95%, and 100% of the peak performance achieved by the standard PPO baseline. TE and SE are then computed by averaging the relative runtime and data usage required to reach these thresholds.

In addition to this multi-level evaluation, we also report a stricter variant based on a single threshold at 100% performance. This provides a complementary view that focuses on the final stage of training.

Results. Table 7 summarizes both versions of the efficiency metrics. The upper block ("100% only") corresponds to the single-threshold evaluation, while the lower block ("90/95/100%") matches the results reported in the main text.

Across both evaluation protocols, RINSE improves mean sample efficiency over IIF on all six environments and mean time efficiency on five of the six environments. The gains are particularly

Table 7: Relative time efficiency (TE) and sample efficiency (SE) across environments. All values are normalized to the PPO baseline (100%); lower is better. Best results per column are in bold.

Setting	Metric	Method	Ant	HalfCheetah	LunarLander	BipedalWalker	Hopper	FrozenLake
100% only	TE↓	IIF	109.7±2.0	77.5±8.5	**80.3±6.0**	93.9±3.3	114.7±4.0	105.2±7.9
		RINSE	**102.9±2.4**	**60.2±6.7**	88.6±3.9	**71.1±9.5**	**107.5±4.5**	**104.2±2.7**
	SE↓	IIF	85.8±1.8	59.5±6.5	55.4±4.3	74.1±2.8	87.9±3.0	51.0±3.9
		RINSE	**74.7±1.8**	**42.8±4.7**	**49.1±2.1**	**49.7±7.0**	**77.2±3.4**	**48.0±1.6**
90/95/100%	TE↓	IIF	110.2±3.7	75.0±11.0	**76.1±8.5**	94.9±6.0	94.5±18.8	133.4±26.0
		RINSE	**103.0±4.9**	**63.9±10.4**	89.9±6.8	**67.9±13.6**	**88.9±16.7**	**116.3±12.5**
	SE↓	IIF	86.3±3.3	57.7±8.5	52.6±6.0	74.8±5.0	72.2±14.6	64.2±12.2
		RINSE	**74.9±3.7**	**45.4±7.3**	**49.8±3.7**	**47.1±9.7**	**63.9±11.9**	**53.5±5.0**

pronounced in continuous control tasks such as Ant and HalfCheetah, where RINSE reduces sample usage by a substantial margin.

Comparing the two settings, the multi-threshold evaluation (90/95/100%) yields slightly more stable estimates, as it averages performance across different stages of training. In contrast, the 100%-only metric is more sensitive to late-stage behavior but leads to similar overall conclusions.

Overall, these results suggest that the efficiency improvements of RINSE are not limited to a specific performance regime, but persist throughout the training process.

D.4.1 Wall-Clock Runtime Breakdown

Table 8 reports the wall-clock runtime per round for each control environment, separating rollout collection, the initial training pass, attribution, and filtered replay. The baseline cost is the sum of collection and training, while the additional cost is the sum of attribution and replay. The final column reports this additional cost relative to the baseline.

Table 8: Wall-clock runtime in seconds per round on the traditional RL environments.

Environment	Collect	Train	Attribution	Replay	Base	Overhead	Overhead / Base
Ant-v4	1.46	17.66	15.90	5.51	19.12	21.41	112%
HalfCheetah-v4	0.86	17.13	13.85	5.36	17.99	19.21	107%
Hopper-v4	1.06	26.45	14.28	5.66	27.51	19.94	72%
BipedalWalker-v3	7.24	26.95	15.56	5.66	34.19	21.22	62%
LunarLander-v2	0.75	0.97	0.62	1.44	1.72	2.06	120%
FrozenLake-v1	1.43	3.20	3.07	2.87	4.63	5.94	128%

Attribution and replay add between 62% and 128% of the baseline per-round cost in these measurements. This overhead differs from TE, which measures total time required to reach specified performance thresholds. Faster progress in return can offset the additional per-round computation, but the resulting wall-clock benefit depends on the environment.

D.5 Additional Results for RLHF

We provide additional details on RLHF training dynamics and efficiency.

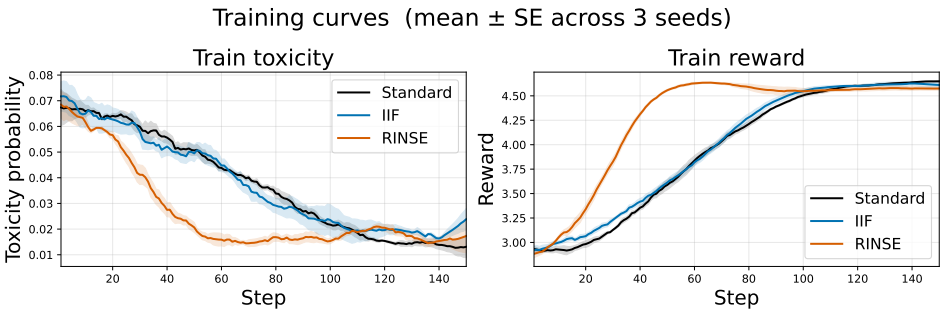


Figure 10: Training curves of Standard, IIF, and RINSE on RLHF toxicity reduction.

Training dynamics. Figure 10 shows the training curves of Standard, IIF, and RINSE on training toxicity and reward. Standard and IIF exhibit similar learning dynamics, with IIF slightly faster than Standard, while RINSE achieves more significant toxicity reduction together with higher reward under the same number of training steps.

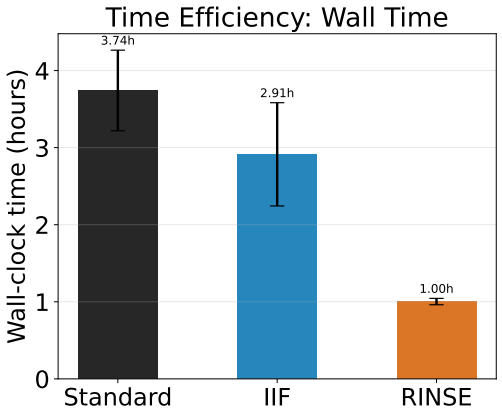


Figure 11: Average wall time of Standard, IIF, and RINSE on RLHF.

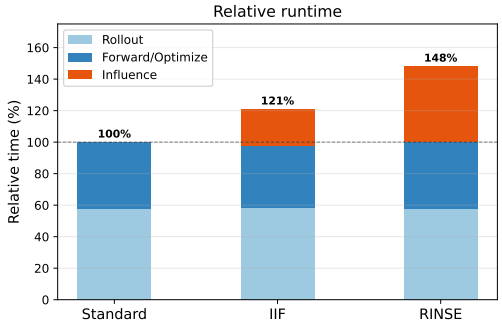


Figure 12: Per-step wall-clock time breakdown for Standard, IIF, and RINSE on RLHF.

Time efficiency and runtime breakdown. Figure 11 reports the wall-clock time required for each method to first reach the best test toxicity achieved by Standard on each seed. RINSE reaches the target substantially faster than Standard, demonstrating that non-local data filtering accelerates convergence.

Figure 12 further breaks down the per-step wall-clock time by training phase. With the Generalized Gauss–Newton matrix Martens [2020] approximation, RINSE is only ~1.5× slower than standard training per step. Although RINSE introduces this additional computation per training step, the overhead is compensated by the reduction in required training steps, leading to improved overall wall-clock efficiency.

D.5.1 Generation Quality at Matched Toxicity

We evaluate whether the reduction in toxicity is accompanied by a degradation in generation quality on held-out Wiki-Toxic prompts. We compare RINSE with the unperturbed baseline at a matched toxicity level, so that differences in quality are not explained solely by different degrees of toxicity reduction.

We assess fluency using the perplexity of the generated continuation under a separate language model, GPT-2 Large [Radford et al., 2019], conditioned on the prompt. This evaluator is independent of both the policy and the training reward model. We assess diversity using corpus-level Distinct-2 and Distinct-3, and repetition using the within-sample repeated-4-gram rate, Seq-rep-4.

Table 9: Generation quality on held-out Wiki-Toxic prompts at matched toxicity.

Metric	RINSE	Unperturbed baseline
Median perplexity ↓	$\mathbf{43.4 \pm 2.6}$	50.5 ± 5.3
Distinct-2 ↑	$\mathbf{0.79 \pm 0.01}$	0.72 ± 0.03
Distinct-3 ↑	$\mathbf{0.96 \pm 0.00}$	0.92 ± 0.02
Seq-rep-4 ↓	$\mathbf{0.0037 \pm 0.0013}$	0.0046 ± 0.0010

At matched toxicity, RINSE achieves lower median perplexity, higher Distinct-2 and Distinct-3, and a lower repeated-4-gram rate than the unperturbed baseline. These results provide evidence that the toxicity improvement does not come at the cost of the evaluated fluency and diversity measures.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We tried our best to ensure that the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the theorems along with the assumptions and proof in Appendix B

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed information on the experimental setups in Appendix C, and release our code at `https://github.com/sx-liu/RL-IF`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our code is available at `https://github.com/sx-liu/RL-IF`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We discuss the results in Section 4 and provide the detailed setups in Appendix C

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We use at least 3 random seeds in all of our experiments. We report the standard errors in tables and include error bars in figures.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We discuss the compute in Appendix C, and also provide detailed runtime analysis in Section 4, Appendix D.4 and Appendix D.5

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: To the best of our knowledge, the research presented in this paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper conducts fundamental research in theory, and there is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not release high-risk models, datasets, or generative systems.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the datasets, models, code frameworks and included their licenses in Appendix C.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not include crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not include research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: This paper does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.