NON-LOCAL DATA ATTRIBUTION FOR ON-POLICY REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Data attribution has become an important tool for understanding and improving model training, but its study in reinforcement learning (RL) remains limited. Prior work has shown that local data attribution computed within a single rollout provides useful signals for data selection and hence helps accelerate training. In this work, we move beyond local attribution and introduce non-local data attribution for on-policy RL, where attribution targets are defined using future rollouts generated by a better-performing policy. We formalize this setting via a replay-based leave-one-out objective (replay-LOO) that isolates optimization effects under fixed rollout buffers. Using the well-developed training data attribution methods in supervised learning, we are able to account for the training dynamics when estimating data influence. We show that non-local attribution achieves strong correlation with ground-truth LOO retraining effects in RL. Based on this property, we further demonstrate how non-local attribution can be used for effective data selection by reusing rollout buffers, leading to improved sample efficiency without additional environment interaction. Overall, our results highlight non-local attribution as a promising tool for data-centric reinforcement learning.

1 INTRODUCTION

Reinforcement learning (RL) algorithms are notoriously sensitive to the data they train on: a small subset of transitions can disproportionately shape the learned policy, accelerate improvement, or destabilize training. Recent data-centric problems has motivated a growing line of work on *training data attribution* (Koh & Liang, 2017; Deng et al., 2025), which offers a tool for debugging, dataset curation, and improving training efficiency. In supervised learning, influence and attribution methods have become a standard way to analyze training dynamics and to guide data selection. In on-policy RL, however, attribution is harder: training data are generated online by the current policy, and the effect of a single transition can propagate through many future updates.

Most existing RL attribution or data selection approaches are *local* (Hu et al., 2025): they explain a single update, a single rollout, or the immediate effect of a sample on a short-horizon objective. While local signals are useful, they do not directly answer the counterfactual question that in practice we often care about: *how did a past training sample affect the quality of a later, better-performing policy?*

In this work, we introduce *non-local* data attribution for RL. The key idea is to define attribution targets using *future rollouts* produced by an improved policy checkpoint, and then attribute that future utility back to earlier training samples. Intuitively, we measure influence by asking how training would have changed if a single past experience had been down-weighted or removed, while keeping the rest of the collected buffers fixed.

To estimate these non-local influences efficiently, we adapt two trajectory-based training-data attribution methods from supervised learning to RL: TracIn (Pruthi et al., 2020) and SGD-influence (SGDI) (Hara et al., 2019; Wang et al., 2025). We show empirically that these estimates closely track ground-truth replay-based LOO retraining effects, despite the sequential and non-stationary nature of RL training.

Finally, we demonstrate a downstream use of non-local attribution: *data selection* without additional environment interaction. We propose *Lookahead Iterative Filtering* (LIF), which performs a short

lookahead training phase, computes non-local influence scores for the cached training curriculum, and then replays a filtered curriculum that discards the least helpful occurrences. Across `Acrobot` and `MiniGrid`, LIF improves sample efficiency over standard training and a local filtering baseline.

Below we summarize the contributions of this paper:

- We formalize *non-local* data attribution in on-policy RL by defining future-utility targets and corresponding replay-based LOO effects over an attribution horizon.
- We adapt TracIn and SGD-influence to this setting and show strong correlation with ground-truth replay-LOO retraining effects.
- We introduce LIF, a practical data selection method that reuses rollout buffers and improves sample efficiency without extra environment interaction.

2 Non-local Attribution in Reinforcement Learning

We consider an on-policy RL training process that alternates between data collection and parameter updates. For $i \geq 0$, let θ_i denote the policy parameters at rollout iteration i, and let $\mathcal{B}_i$ be the rollout buffer collected under the behavior policy π_{θ_i}. Each rollout iteration applies a training update

$$\theta_{i+1} = \mathrm{TrainRollout}(\theta_i; \mathcal{B}_i).$$

In this paper, we focus on Proximal Policy Optimization (PPO) (Schulman et al., 2017).

Each buffer $\mathcal{B}_i$ stores collected data samples (e.g., transitions). During optimization, a single data sample may participate in one or more gradient updates. We refer to the j^{th} use of a data record from the i^{th} rollout as a data *occurrence* $z_{i,j}$, which contributes to the update from θ_i to θ_{i+1}.

2.1 Non-local Data Attribution via Leave-One-Out

Our goal is to quantify the *non-local* influence of a past data occurrence $z_{i,j}$ on future training outcomes. We formalize this through a leave-one-out (LOO) lens: how does removing $z_{i,j}$ from training affect the utility of a *future* policy?

Let $H \geq 1$ be an attribution horizon, and let θ_{i+H} denote the checkpoint obtained by continuing training on buffers $\mathcal{B}_i, \ldots, \mathcal{B}_{i+H-1}$. We measure future utility using a policy-gradient surrogate objective

$$f(\theta, \pi^{\mathrm{ref}}) = \mathbb{E}_{\tau \sim \pi^{\mathrm{ref}}, (s,a) \sim \tau} \big[\log \pi_\theta(a \mid s)\, \hat{A}^{\mathrm{ref}}(s,a) \big], \tag{1}$$

where π^{ref} induces the rollout distribution used to evaluate policy utility.

LOO through replay (fixed buffers). We first consider a replay-based LOO definition, in which the rollout buffers $\mathcal{B}_i, \ldots, \mathcal{B}_{i+H-1}$ are treated as fixed. We replay the same training curriculum but remove (or set zero weight on) $z_{i,j}$ when forming the update at rollout iteration i. Let $\theta_{i+H}^{\setminus(i,j),\mathrm{replay}}$ denote the resulting parameters. The replay-LOO effect is

$$\Delta_{i,H}^{\mathrm{replay}}(z_{i,j}) := f\big(\theta_{i+H}^{\setminus(i,j),\mathrm{replay}}, \pi_{\theta_{i+H}}\big) - f\big(\theta_{i+H}, \pi_{\theta_{i+H}}\big). \tag{2}$$

LOO through recollection (counterfactual data generation). Alternatively, removing $z_{i,j}$ may change future behavior policies and hence the data distribution. In this recollection-based LOO, we rerun training from θ_i with $z_{i,j}$ removed, recollecting new buffers under the counterfactual policies. Let $\theta_{i+H}^{\setminus(i,j),\mathrm{recollect}}$ denote the resulting checkpoint. The recollection-LOO effect is

$$\Delta_{i,H}^{\mathrm{recollect}}(z_{i,j}) := f\big(\theta_{i+H}^{\setminus(i,j),\mathrm{recollect}}, \pi_{\theta_{i+H}}\big) - f\big(\theta_{i+H}, \pi_{\theta_{i+H}}\big). \tag{3}$$

While recollection-LOO captures both optimization and distributional effects, it requires additional environment interaction. In this paper, we focus on replay-LOO, which isolates optimization effects and is practical for data filtering. We leave a systematic study of recollection-LOO to future work.

2.2 Limitations of Local Attribution

We first examine whether *local* attribution methods faithfully approximate future replay-LOO effects. Specifically, we follow Hu et al. (2025) and compute the TracIn scores (see Equation (4) for detail) with attribution target $f(\theta_i, \pi_{\theta_i})$ for data in the i^{th} rollout, and compare them against ground-truth replay-LOO effects measured at the future checkpoint θ_{i+H}. We refer to this method as *local* TracIn.

Experiment Setup. We evaluate local TracIn on the `MiniGrid` environment. We set $H = 5$ and consider occurrences from rollout buffers $\mathcal{B}_0, \ldots, \mathcal{B}_4$ when attributing importance to θ_5. To obtain ground-truth replay-LOO scores, we sample 1% of occurrences and perform replay-based LOO retraining, computing Equation (2). We report the correlation between TracIn scores and replay-LOO effects.

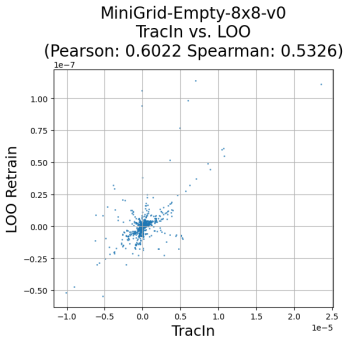


Figure 1: Correlation between local TracIn and replay-based LOO.

Results. As shown in Figure 4, although the correlation is positive, it is not consistently strong across rollouts and tends to decrease for attribution scores from earlier rollouts. We further report detailed per-rollout correlations in Figure 7 in the appendix. These results indicate that local attribution signals may fail to capture long-term data utility, which motivates the use of *non-local* attribution signals that explicitly measure future utility.

2.3 Lookahead Evaluation and Non-local Attribution Methods

To better approximate replay-LOO effects, we adopt a *lookahead evaluation* strategy, in which data influence is assessed using rollouts drawn from a future policy. Concretely, we set the reference policy to $\pi^{\text{ref}} = \pi_{\theta_{i+H}}$, so that past data occurrences $z_{i,j}$ are evaluated under the state–action distribution induced by a later stage of training. Since such future policies are typically more capable in practice, this choice emphasizes long-term utility rather than immediate, myopic effects.

Under this evaluation scheme, we adapt two trajectory-based attribution methods from supervised learning, TracIn and SGD-influence, to approximate replay-LOO effects in on-policy RL.

TracIn with lookahead evaluation. We adapt TracIn (Pruthi et al., 2020) by replacing the local target gradient used in Hu et al. (2025) with the gradient of the lookahead utility. The TracIn score of an occurrence z is defined as

$$\text{TracIn}(z) := \sum_{k=0}^{N-1} \eta_k \, \nabla_\theta \ell(\theta_k; z)^\top \nabla_\theta f(\theta_k, \pi_{\theta_{i+H}}). \tag{4}$$

Following Hu et al. (2025), for data from the i^{th} rollout, we use only checkpoint θ_i.

SGD-influence (SGDI). We also consider SGD-influence (Hara et al., 2019; Bae et al., 2024; Wang et al., 2025), which linearizes the training dynamics to approximate replay-LOO effects. Here, $t \in \{0, \ldots, T-1\}$ indexes the SGD update steps within the replayed training trajectory, where T is the total number of gradient updates. Let $g_t(z) := \nabla_\theta \ell(\theta_t; z)$ and $H_t := \nabla^2_\theta \ell(\theta_t)$ denote the per-step gradient and Hessian of the training objective. The SGD-influence of an occurrence z is given by

$$\text{SGDI}(z) := \sum_{t=0}^{T-1} \eta_t \, g_t(z)^\top \left(\prod_{s=t+1}^{T-1} (\boldsymbol{I} - \eta_s H_s) \right)^\top \nabla_\theta f(\theta_{i+H}, \pi_{\theta_{i+H}}). \tag{5}$$

We now empirically evaluate how well non-local TracIn and SGDI scores correlate with ground-truth replay-LOO effects. We use the same experimental setup as in Section 2.2, computing influence scores using the future rollout buffer $\mathcal{B}_5$ as the attribution target.

Results. As shown in Figure 2, both TracIn and SGDI exhibit strong linear correlation with replay-LOO effects. TracIn achieves a Spearman correlation of 0.9812, while SGDI slightly improves to 0.9857, substantially outperforming local attribution. These results indicate that incorporating lookahead evaluation aligns attribution scores with the optimization dynamics that govern future performance, enabling training-data attribution methods to faithfully approximate replay-LOO effects in on-policy RL.

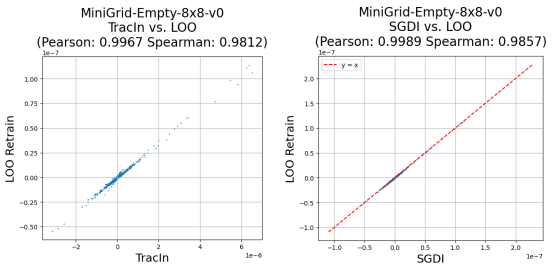


Figure 2: Correlation between lookahead attribution and replay-based LOO (left: TracIn; right: SGD-influence).

3 Data Selection with Non-local Attribution

Building on the above observations, we propose *Lookahead Iterative Filtering* (LIF), a data selection method for online reinforcement learning based on non-local data attribution, as illustrated in Algorithm 1. Each iteration focuses on an interval of H rollout buffers, which we refer to as the *attribution horizon*.

3.1 Algorithm and Designs

Lookahead stage. Starting from the current model parameters θ_i, we collect rollouts and perform standard training for H rollouts to obtain the information required for data attribution. This produces the rollout buffers $\mathcal{B}_i, \mathcal{B}_{i+1}, \ldots, \mathcal{B}_{i+H}$ and intermediate checkpoints $\theta_{i+1}, \ldots, \theta_{i+H}$, where $\mathcal{B}_i$ is collected under policy π_{θ_i} and is used to update the model from θ_i to θ_{i+1}. During this procedure, we also cache the training curriculum $c_i = \{z_{i,0}, \ldots, z_{i,T-1}\}$, where $z_{i,j} = (s_{i,j}, a_{i,j}, r_{i,j}, \log \pi_{i,j}, v_{i,j}, \hat{A}_{i,j})$ stands for a single data occurrence, as well as per-step gradients for later influence computation.

Algorithm 1 Lookahead Iterative Filtering (LIF)

Require: Initial parameters θ_i, attribution horizon H, discard percentage $p \in (0, 1)$

$\theta \leftarrow \theta_i$
▷ Lookahead Stage
for $h = 0, 1, \ldots, H-1$ **do**
  Collect rollout buffer $\mathcal{B}_{i+h}$ using policy π_θ
  Update parameters: $\theta \leftarrow \mathrm{TrainRollout}(\theta; \mathcal{B}_{i+h})$
  Cache gradients and curriculum: g_{i+h}, c_{i+h}
end for
▷ Attribution Stage
$I \leftarrow \mathrm{ComputeInfluence}(g, \theta, \mathcal{B}_{i+H-1})$
$\widetilde{c}_{i:i+H} \leftarrow \mathrm{DiscardBottomRecords}(c_{i:i+H}, I, p)$
▷ Replay Stage
$\theta \leftarrow \theta_i$
for $h = 0, 1, \ldots, H-1$ **do**
  $\theta \leftarrow \mathrm{TrainStep}(\theta; \widetilde{c}_{i+h})$
  Update log-probabilities $\log \pi_i \leftarrow \log \pi_\theta$
end for

Attribution stage. We then compute an influence score for each data *occurrence* $z_{i,j}$ along the training trajectory and filter the rollouts accordingly. Specifically, using the training gradients stored during the lookahead stage, we apply SGDI to estimate data influence as defined in Equation (1).

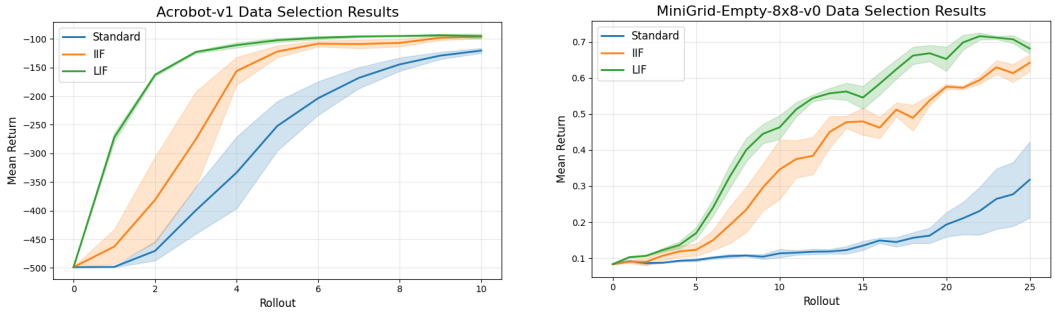
Since our goal is to measure how past data affect *future* policy performance, a natural choice is to set $\pi^{\mathrm{ref}} = \pi_{\theta_{i+H}}$ (the final checkpoint from the lookahead stage) and to estimate the expectation using samples from $\mathcal{B}_{i+H}$. In practice, we find that using the second-to-last rollout buffer $\mathcal{B}_{i+H-1}$ as the target produces comparable attribution quality. Therefore, we use $\mathcal{B}_{i+H-1}$ as a proxy to avoid additional sampling at checkpoint θ_{i+H} and further improve sample efficiency. In other words, we are setting the target function as $f(\theta_{i+H}, \pi_{\theta_{i+H-1}})$.

Replay stage. After computing per-occurrence influence scores, we discard the bottom p fraction of occurrences with negative influence scores to obtain a filtered curriculum, where p is the discard ratio. We then train from θ_i by replaying the filtered curriculum sequentially to obtain the updated model checkpoints $\theta'_{i+1}, \ldots, \theta'_{i+H}$. Notably, replaying a filtered curriculum is inherently off-policy

and can introduce a mismatch between the current policy and the logged training data. Empirically, we find that simple reweighting mitigates this issue. Specifically, at the start of rollout iteration $i + k$, we update the stored log-probabilities by replacing $\log \pi_i$ with $\log \pi_{\theta'_{i+k}}$. In other words, we are using the new updated θ'_{i+k} as a proxy for the true data generating model.

In practice, when training on a large number of time steps, we divide the whole training process into a few lookahead horizons, and for each horizon, we apply LIF individually for data selection and retraining.

3.2 Experiments in RL environments



`Acrobot-v1`. `MiniGrid-Empty-8x8-v0`.

Figure 3: Data selection results. LIF improves sample efficiency and final performance compared to standard training and local filtering (IIF). Shaded regions indicate standard error across seeds.

Experiment Setup. To evaluate our method, we use two standard RL environments: `Acrobot` (control) and `MiniGrid` (navigation). For each environment, we train a common backbone agent with a fixed interaction budget and compare our filtering procedure against two baselines: the unfiltered standard training and another local data filtering method *IIF* (Hu et al., 2025). We select the discard proportion that performs best in each environment: $p = 10\%$ for `MiniGrid` and $p = 50\%$ for `Acrobot`. We choose the lookahead horizon $H = 5$, and report learning curves averaged over 3 different random seeds, using the average environment return over 1000 episodes as the primary metric. The choice of hyperparameters for both environments are reported in Table 1 in appendix.

Results. Figure 3 reports learning curves for data selection on `Acrobot` and `MiniGrid`. Across both tasks, *LIF* consistently outperforms the unfiltered baseline and the local filtering method *IIF*, indicating that non-local attribution provides a more effective signal for selecting replay data.

Despite the improvement brougt by non-local data selection, one may observe that, within a single lookahead horizon, the LIF trajectory can plateau or even degrade at later rollouts. One plausible explanation is increasing off-policy mismatch induced by replaying filtered curricula rather than collecting fresh on-policy data. This highlighs a trade-off between convergence speed and off-policy mismatch in non-local data selection.

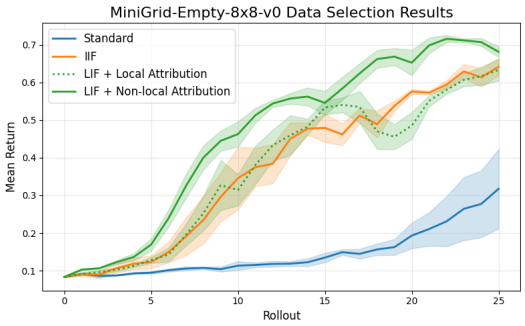
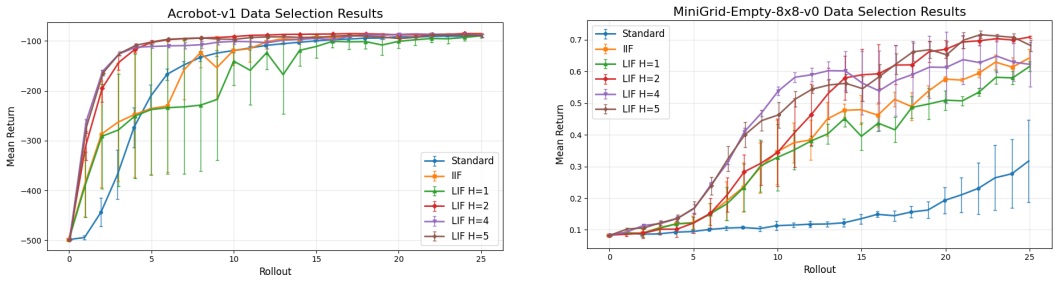


Figure 4: MiniGrid data selection ablation comparing LIF with local vs. non-local attribution. Local attribution uses TracIn with a local target rollout.

3.3 Ablation Studies

LIF with local attribution Using the same experimental settings as in Section 3, we ablate the role of the attribution target by running LIF with *local* attribution on `Minigrid`. In the attribution stage, we compute TracIn scores using



`Acrobot-v1`. `MiniGrid-Empty-8x8-v0`.

Figure 5: Effect of the attribution horizon H in LIF. We compare learning curves for LIF with $H \in \{1, 2, 4, 5\}$ on `Acrobot-v1` (left) and `MiniGrid-Empty-8x8-v0` (right), alongside standard training and the local filtering baseline (IIF). Error bars indicate standard error over 3 random seeds.

the standard local target $f(\theta_i, \pi_{\theta_i})$. Specificaly,
the reference policy π^{ref} is the current policy checkpoint π_{θ_i} and the target utility is evaluated on rollout $\mathcal{B}_i$ collected from that same checkpoint. Figure 4 shows that replacing non-local targets with local ones largely removes the benefit of LIF: the resulting learning curve degrades to the similar performance as the local filtering baseline IIF and remains substantially below LIF with non-local attribution. This suggests that the improvement of LIF is driven primarily by the *non-local* attribution target, i.e., attributing utility defined on *future* rollouts, rather than by the filtering procedure alone.

LIF with different attribution horizons We next examine how the choice of attribution horizon H affects LIF. We run LIF with $H \in \{1, 2, 4, 5\}$ and report learning curves in Figure 5. Across both `Acrobot` and `MiniGrid`, shorter horizons (e.g., $H = 1$) tend to provide a weaker selection signal and yield smaller and less stable improvements over IIF, while increasing the horizon generally improves sample efficiency. In particular, intermediate-to-long horizons ($H = 2, 4, 5$) accelerate early learning, with $H = 4$ or $H = 5$ performing best overall. These results suggest that a longer lookahead better captures delayed training benefits of past occurrences, though the gains from further increasing H appear to diminish beyond a small number of rollouts.

4 Related Work

Data filtering and experience selection have long been recognized as effective mechanisms for improving sample efficiency in reinforcement learning. A representative line of work is prioritized experience replay (PER), which biases sampling toward transitions with large temporal-difference errors, focusing learning on informative mistakes and substantially accelerating deep RL training (Schaul et al., 2015). Subsequent methods refine this idea by distinguishing between informative and unlearnable data, for example prioritizing transitions based on reducible loss or learnability to avoid over-sampling noisy or irreducible experiences (Sujit et al., 2023). Other approaches identify and replay key transitions or states that are critical for task completion, mitigating catastrophic forgetting and improving performance across actor-critic algorithms (Guo & Gao). In offline and transfer settings, data filtering has also been used to discard transitions that are inconsistent with target dynamics or value functions, preventing negative transfer (Xu et al., 2023). While these methods demonstrate that selective reuse of experience can significantly improve RL performance, they typically rely on heuristic signals such as TD error, novelty, or value consistency.

In contrast, training data attribution (TDA) provides a principled framework for quantifying the influence of individual training samples on model behavior and has been extensively studied in supervised learning. Classical influence functions estimate leave-one-out effects using second-order information but are computationally expensive (Koh & Liang, 2017). To improve scalability, prior work has proposed efficient approximations such as TracIn, which accumulates gradient inner products along the training trajectory (Pruthi et al., 2020), and LoGRA, which further improves efficiency by leveraging low-rank gradient representations with curvature approximations such as EKFAC (Grosse et al., 2023; Choe et al., 2024).

Despite their success, most existing TDA methods are developed for static supervised learning. Our work extends this line of research to reinforcement learning by introducing non-local data attribution, where influence is defined with respect to future rollouts from improved policies, and demonstrates its utility for guiding experience reuse without additional environment interaction.

5 Conclusion

We introduced *non-local data attribution* for reinforcement learning, which quantifies how individual training occurrences influence the quality of a *future* policy checkpoint by defining attribution targets on rollouts from an improved reference policy. Under a fixed-buffer replay definition of leave-one-out retraining, we showed that influence estimators adapted from supervised learning, such as TracIn and SGD-influence, are able to track ground-truth replay-LOO effects with high correlation.

Building on these estimates, we proposed *Lookahead Iterative Filtering* (LIF), a practical data selection procedure that reuses cached rollout buffers to discard low-influence occurrences and improves sample efficiency without additional environment interaction. Together, our results suggest that non-local attribution can serve as a principled tool for data-centric RL, enabling both interpretability and algorithmic improvements.

Future directions. Several directions are promising for future work. First, it is important to extend LIF to more complex environments and settings, including large-scale agents such as modern LLM-based policies, where attribution may help curate interaction traces and training curricula. Second, LIF currently relies on replaying filtered buffers, which can introduce off-policy mismatch; developing more robust replay or correction mechanisms, such as importance weighting or trust-region style constraints, is an important step toward stable long-horizon gains. Finally, our study focused on replay-LOO. A deeper investigation of recollection-LOO could provide a more faithful notion of influence in RL, including new theoretical formulations and practical applications that leverage counterfactual data generation.

References

Juhan Bae, Wu Lin, Jonathan Lorraine, and Roger B. Grosse. Training data attribution via approximate unrolling. *Advances in Neural Information Processing Systems 37*, 2024.

Sang Keun Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Minsoo Kang, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, et al. What is your data worth to gpt? llm-scale data valuation with influence functions. *arXiv preprint arXiv:2405.13954*, 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. 2025.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, et al. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*, 2023.

Youtian Guo and Qi Gao. Experience replay more when it's a key transition in deep reinforcement learning.

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

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=sYK4yPDuT1.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

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

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

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

Shivakanth Sujit, Somjit Nath, Pedro Braga, and Samira Ebrahimi Kahou. Prioritizing samples in reinforcement learning with reducible loss. *Advances in Neural Information Processing Systems*, 36:23237–23258, 2023.

Jiachen T. Wang, Dawn Song, James Zou, Prateek Mittal, and Ruoxi Jia. Capturing the temporal dependence of training data influence. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=uHLgDEgiS5.

Kang Xu, Chenjia Bai, Xiaoteng Ma, Dong Wang, Bin Zhao, Zhen Wang, Xuelong Li, and Wei Li. Cross-domain policy adaptation via value-guided data filtering. *Advances in Neural Information Processing Systems*, 36:73395–73421, 2023.

A Experimental Configuration

For all the studies in this paper, we use `Acrobot-v1` and `MiniGrid-Empty-8x8-v0` as experiment environments. For `Acrobot`, we parameterize the policy and value networks with an MLP (`MlpPolicy`). For `MiniGrid`, we use a CNN-based policy and value network (`CnnPolicy`), with an intermediate feature dimension of 64. We report the hyperparameters in PPO training in Table 1.

Hyperparameter	**Acrobot-v1**	**MiniGrid-Empty-8x8-v0**
policy_type	`MlpPolicy`	`CnnPolicy`
features_dim	–	64
total_timesteps	102400	81920
n_epochs	10	10
n_steps	2048	2048
batch_size	64	64
learning_rate	5.0×10^{-3}	5.0×10^{-3}
γ	0.99	0.99
ent_coef	0.0	0.0
clip_range	0.2	0.2
normalize_advantage	`true`	`true`
optimizer	`SGD`	`SGD`

Table 1: PPO configuration used for `Acrobot-v1` and `MiniGrid-Empty-8x8-v0` in our experiments. "–" indicates that the field is not used for that environment.

B Per-rollout Correlations

To better understand when non-local attribution is reliable during training, we report per-rollout correlations between each estimator and the corresponding replay-LOO ground truth. For each rollout index i, we compute the rank correlation (Spearman) across occurrences in buffer $\mathcal{B}_i$ between (1) the estimated influence score and (2) the replay-LOO effect measured by retraining with that occurrence removed.

Figure 6 summarizes these correlations for `MiniGrid`. Overall, the correlations are consistently positive across rollouts, with TracIn typically achieving higher and more stable correlation than SGDI. This supports our main claim that trajectory-based supervised-learning attribution methods can track replay-LOO effects in on-policy RL.

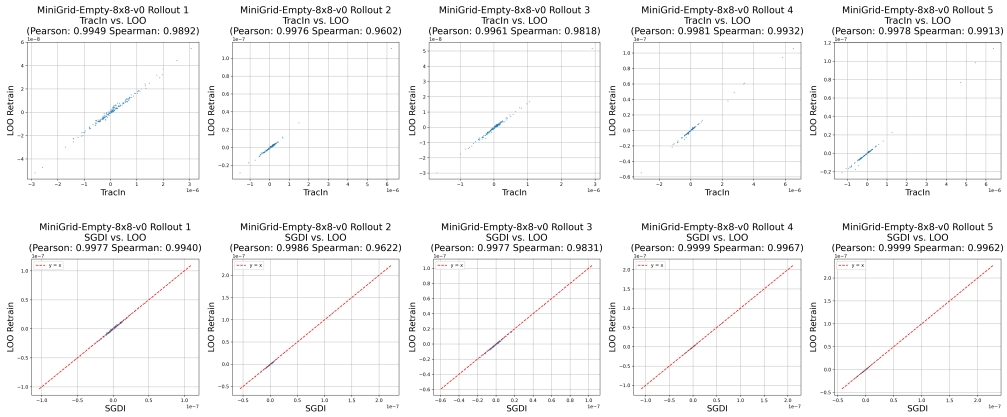


Figure 6: Per-rollout correlations on `MiniGrid` for non-local attribution. Each panel corresponds to a rollout index i and reports the Spearman rank correlation (across occurrences in $\mathcal{B}_i$) between estimated influence scores and replay-LOO effects. Top row: TracIn with non-local (future-rollout) targets; bottom row: SGDI with non-local targets.

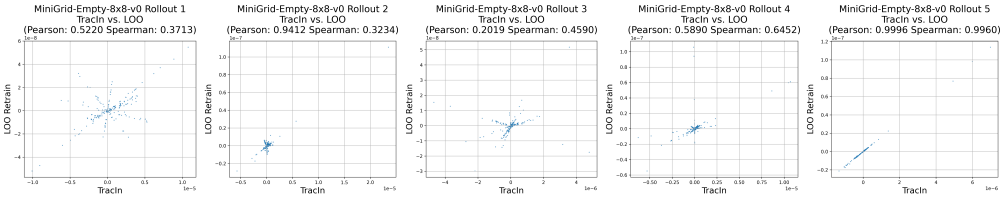


Figure 7: Per-rollout correlations for TracIn with *local* attribution targets on `MiniGrid`. Each panel corresponds to a rollout index i and reports the Spearman rank correlation between local TracIn scores for occurrences in $\mathcal{B}_i$ and their replay-LOO effects.

For comparison, we also report per-rollout correlations for *local* attribution, where TracIn uses the local target rollout. As shown in Figure 7, local correlations are generally weaker and more variable across rollouts than their non-local counterparts, consistent with our main finding that future-rollout targets provide a stronger and more predictive influence signal.