Reward-Free Action Poisoning in Offline RL via Conditional Shapley Value Estimation

Extended Abstract

Shenghong He
Sun Yat-sen University
Guangzhou, China
heshh23@mail2.sysu.edu.cn

Chao Yu*
Sun Yat-sen University
Guangzhou, China
Pengcheng Laboratory
Shenzhen, China
yuchao3@mail.sysu.edu.cn

Kaiqaing Ke
Sun Yat-sen University
Guangzhou, China
kekq@mail2.sysu.edu.cn

Yinqi Wei
Nanyang Technological University
Singapore, Singapore
WEIY0041@e.ntu.edu.sg

ABSTRACT

Recent studies show that data poisoning attacks can degrade the performance of offline reinforcement learning (RL) policies by strategically tampering with training datasets. However, existing methods generally assume the acquisition of reward signals during the generation of poisoning data, thus limiting their applications in real-world scenarios when the reward signals are not available. In this paper, we propose a novel method, called Shapley Action-Poisoning Attack (SAPA), which calculates the contribution of each state-action pair in a trajectory to identify key actions for poisoning attacks without dependence on reward signals. Theoretical analysis proves that SAPA can degrade the performance of the learned policy below a specified threshold by tampering with key actions. Numerous experimental results demonstrate that SAPA surpasses state-of-the-art poisoning methods in the attack performance under various offline algorithms.

KEYWORDS

Poisoning Attack; Offline reinforcement learning; Data Security; Reward-free Attack

ACM Reference Format:
Shenghong He, Chao Yu*, Kaiqaing Ke, and Yinqi Wei. 2026. Reward-Free Action Poisoning in Offline RL via Conditional Shapley Value Estimation: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/WJRF3793

1 INTRODUCTION

As a general paradigm for addressing sequential decision-making problems, Reinforcement learning (RL) [9] has achieved remarkable success in various domains, including autonomous driving [5], cooperative robotics [2], video games [1] and recommendation systems [13]. However, recent studies [10] have shown that RL is vulnerable to data poisoning attacks, which can degrade the cumulative return of the learned policy by tampering with the training data. Thus, exploring effective offline poisoning attack methods and finding secure defense mechanisms have become a hot topic in offline RL security research. However, existing methods [11, 12] cannot perform data poisoning on datasets where reward signals are unavailable, which is common in real life situations, such as in autonomous driving where the collected offline datasets typically contain only the observation states and driving actions of the human drivers.

In this paper, we propose a novel method called **S**hapley **A**ction-**P**oisoning **A**ttack (SAPA) to effectively attack offline RL agents by manipulating the key actions during the learning process without relying on reward signals. To achieve this, we introduce the Shapley value to assess the contribution of state-action pairs on the overall return of the trajectory at different timesteps and use this assessment to identify the key actions (i.e., high-return actions) for attack. Building on this metric, we further develop an action generator that strategically replaces the most influential actions (ranked by Shapley values) with adversarial actions. In summary, SAPA conducts computations exclusively based on the existing states and actions within the dataset during the poisoning process, which makes SAPA applicable to a wider variety of attack scenarios.

2 METHOD

Shapley values in RL. In RL settings, the return of a single state-action pair cannot be separated from its temporal dependence, which leads to the traditional Shapley values failing to accurately compute the contribution of each state-action pair. To address this issue, we propose a conditional Shapley value, which incorporates the conditional expectation of return into the traditional Shapley value framework to avoid inaccurate value assessments due to time dependencies. In order to quantify the contribution of state-action pairs, we can compute the conditional expectation of the reward model output to precisely capture the joint distribution among the data. Specifically, for a trajectory $\tau = (s_0, a_0, \ldots, s_T, a_T)$, the

* indicates the corresponding author.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/WJRF3793

Table 1: Attack performance of different action-poisoning methods. $*$ denotes the original unpoisoned algorithm.

Datasets	Environment		*	RS	AS	RFAS	TSS	SAPA
medium-expert	Halfcheetah	BCQ	86.7±3.6	24.8±5.4	24.8±3.4	35.2±3.5	34.4±3.4	**23.2±3.5**
		CQL	92.8±6.1	43.4±6.7	**17.4±4.7**	21.7±4.3	38.4±4.7	18.7±4.7
	Hopper	CQL	106.8±3.1	31.5±6.7	26.5±4.7	31.6±3.7	35.5±4.7	**23.4±3.3**
		IQL	97.4±3.5	32.3±8.5	25.3±3.5	38.5±4.2	33.3±6.1	**24.2±4.1**
	Walker2d	IQL	109.9±2.8	62.4±15.3	27.1±3.3	35.9±5.3	58.7±10.5	**25.7±4.9**
		DT	107.7±3.2	53.7±5.4	29.7±4.4	36.8±4.7	55.3±6.3	**27.6± 4.1**

conditional value function of subset C is defined as:

$$v_{\text{cond}}(C) = \mathbb{E}\left[\sum_{t=0}^{T} \hat{r}_t \,\middle|\, C\right], \tag{1}$$

where $\hat{r}$ is a value that measures the benefit of a state-action pair (e.g., an approximate reward value). The conditional Shapley value ϕ_{i_t} is then defined as:

$$\phi_{i_t}(v_{\text{cond}}) = \sum_{C \subseteq N \setminus \{i_t\}} \frac{|C|!(|N| - |C| - 1)!}{|N|!} \left[v_{\text{cond}}(C \cup i_t) - v_{\text{cond}}(C)\right], \tag{2}$$

where N is a collection of state-action pairs in the trajectory, i_t is t-th state-action pair and $|C|$ is the number of state-action pairs in a subset. By evaluating the contribution of any state-action pair based on the conditional value function, SAPA can avoid inaccurate evaluations due to time dependencies when state-action pairings are added to subset C.

To efficiently evaluate the benefit of state–action pairs, we propose a scoring function $\mathcal{F}$ that infers potential rewards solely from state–action information. By leveraging a small offline expert dataset $\mathcal{D}_E$, the scoring function $\mathcal{F}$ can learn to discriminate between the expert action a_t and suboptimal alternatives $\tilde{a}_t$ for a given state s_t:

$$\mathcal{L} = \mathbb{E}_{(s_t,a_t)\sim\mathcal{D}_E}[-\log \mathcal{F}(s_t,a_t)] + \mathbb{E}_{(s_t,\tilde{a}_t)\sim\mathcal{D}}[-\log(1-\mathcal{F}(s_t,\tilde{a}_t))], \tag{3}$$

where $\mathcal{D}$ denotes the offline dataset used for data poisoning attacks. As a result, we redefine the value function $v_{\text{cond}}(C)$ in Eq. (1) by replacing $\hat{r}$ terms with $\mathcal{F}$-based scores:

$$v_{\text{cond}}(C) = \mathbb{E}\left[\sum_{t=0}^{T} \mathcal{F}(s_t,a_t) \,\middle|\, C\right]. \tag{4}$$

Then, SAPA determines key actions by calculating the marginal contribution of each state-action pair through Eq. (4).

Worst-case action. To be able to generate covert and effective perturbation actions, we design an action generator based on a conditional variational autoencoder (CVAE) that produces actions conditioned on states and Shapley values. CVAE consists of an encoder (Enc) and a decoder (Dec), where Enc outputs the latent variable z under the Gaussian distribution, and Dec maps z to the desired space. The training method for the action generator is to maximize the evidence lower bound, which is equivalent to minimizing the loss function:

$$L_\theta = \mathbb{E}_{(s,a,v)\sim\mathcal{D},z\sim\text{Enc}(s,v)}[(a-\text{Dec}(s,v,z))^2] + D_{KL}(\text{Enc}(s,v)||\mathcal{N}(0,I))], \tag{5}$$

where $D_{KL}(\cdot\|\cdot)$ represents KL divergence and I is the unit matrix. During action generation, a latent vector z is sampled from the normal distribution, and then this vector together with the current state s and Shapley value v is input into the decoder $\text{Dec}(s,v,z)$ to generate an action.

3 EXPERIMENTS

In our attack experiments, five offline RL algorithms BC [6], BCQ [4], CQL [8], IQL [7], and DT [3] are selected as attack targets due to their widespread adoption as offline baseline methods. We investigate the impact of SAPA, the traditional Shapley solution (TSS), the advantage-based solution (AS), the reward-free advantage solution (RFAS) and the random solution (RS) on the performance of offline methods.

Table 1 compares the poisoning capabilities of SAPA under different key action selection methods across three environments: HalfCheetah, Hopper, and Walker2D. As can be seen, while RS significantly degrades the performance of offline RL algorithms, its high variance reveals unstable attack patterns. Similarly, due to neglecting temporal dependencies, TSS fails to capture critical actions, resulting in attack effectiveness that is merely comparable to RS. In contrast, AS demonstrates greater efficacy by leveraging the advantage value function to identify and perturb key actions. RFAS amplifies $\mathcal{F}$ errors due to the Q-function and V-function network, resulting in lower attack performance than AS. Most notably, SAPA outperforms existing methods by replacing critical operations without relying on reward signals from offline data, which can be applied to more scenarios of data poisoning attacks.

4 CONCLUSION

In this paper, we propose a novel method called SAPA, which generates poisoned data to reduce the cumulative return of offline methods by identifying and manipulating key actions without requiring reward signals. Future directions include refining the importance calculation of trajectory actions to improve computational efficiency and extending SAPA to offline multi-agent settings.

ACKNOWLEDGMENTS

We gratefully acknowledge the support from the Distinguished Young Scholars Project funded by the Natural Science Foundation of Guangdong Province (No. 2025B1515020060), the Basic and Applied Basic Research Program of the Guangzhou Science and Technology Plan (No. 2025A04J7141), and the Xiaomi Young Talents Program.

REFERENCES

[1] Eloi Alonso, Maxim Peter, David Goumard, and Joshua Romoff. 2021. Deep Reinforcement Learning for Navigation in AAA Video Games. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, Zhi-Hua Zhou (Ed.). ijcai.org, 2133–2139.

[2] Giorgio Angelotti, Caroline P. C. Chanel, Adam Henrique Moreira Pinto, Christophe Lounis, Corentin Chauffaut, and Nicolas Drougard. 2024. Offline Risk-sensitive RL with Partial Observability to Enhance Performance in Human-Robot Teaming. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems / ACM, 58–67.

[3] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision Transformer: Reinforcement Learning via Sequence Modeling. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems*. 15084–15097.

[4] Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-Policy Deep Reinforcement Learning without Exploration. In *Proceedings of the 36th International Conference on Machine Learning, ICML (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 2052–2062.

[5] Xinwei Gao, Arambam James Singh, Gangadhar Royyuru, Michael Yuhas, and Arvind Easwaran. 2025. CRLLK: Constrained Reinforcement Learning for Lane Keeping in Autonomous Driving. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems / ACM, 3026–3028.

[6] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. 2017. Imitation Learning: A Survey of Learning Methods. *ACM Comput. Surv.* 50, 2 (2017), 21:1–21:35.

[7] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 2022. Offline Reinforcement Learning with Implicit Q-Learning. In *The Tenth International Conference on Learning Representations, ICLR*. OpenReview.net.

[8] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative Q-Learning for Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems*.

[9] Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. 2025. Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence*. AAAI Press, 28698–28699.

[10] Zhibo Wang, Jingjing Ma, Xue Wang, Jiahui Hu, Zhan Qin, and Kui Ren. 2023. Threats to Training: A Survey of Poisoning Attacks and Defenses on Machine Learning Systems. *ACM Comput. Surv.* 55, 7 (2023), 134:1–134:36.

[11] Young Wu, Jeremy McMahan, Xiaojin Zhu, and Qiaomin Xie. 2023. Reward Poisoning Attacks on Offline Multi-Agent Reinforcement Learning. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI*. AAAI Press, 10426–10434.

[12] Yinglun Xu, Rohan Gumaste, and Gagandeep Singh. 2025. Universal Black-Box Reward Poisoning Attack against Offline Reinforcement Learning. (2025).

[13] Zhenghai Xue, Qingpeng Cai, Bin Yang, Lantao Hu, Peng Jiang, Kun Gai, and Bo An. 2025. AURO: Reinforcement Learning for Adaptive User Retention Optimization in Recommender Systems. In *Proceedings of the ACM on Web Conference 2025, WWW*. ACM, 391–401.