Generative AI-based Offline Reinforcement Learning for Trajectory Planning

Chaemoon Im, Bohyeon Kim and Joongheon Kim
Department of Electrical and Computer Engineering, Korea University, Seoul, Republic of Korea
E-mails: {anscodla0314,mattia0515,joongheon}@korea.ac.kr

***Abstract*—Conventional reinforcement learning (RL) algorithms faces several problems such as danger and cost for collecting data from the environment, degrading its potential for real-world application. As an alternative, offline RL, which aims to learn policy from the dataset, is proposed to remove the danger from real-time interaction with the environment. Among them, this paper proposes diffusion-based offline reinforcement learning algorithm which adopts diffusion model to better extract useful policy from the dataset . Experiment results shows the potential of the diffusion-based RL, in terms of the maximized reward.**

***Index Terms*—Generative AI, Offline Reinforcement Learning, Trajectory Planning**

I. Introduction

Reinforcement learning (RL) has been an useful solution for decision-making task, such as autonomous driving, drone manipulation and so on [1], [2]. However in many cases, trial-and-error based learning, which is the essential part of the algorithm, can pose a risk during the training, degrading the algorithm's potential and applicability [3]. Other modification aims to enhance the safety of the RL by encompassing safety-related constraints into the reward settings [4]. However, as long as this safety constraints are hand-crafted, the adaptability of RL still cannot be enhanced, because it requires continuous manual adaptation.

Offline RL is an alternative method for solving the problem of RL originating from trial-and-error [5]. Instead of directly interacting with the environment, offline RL gathers data from experts and aims to learn the policy from that dataset, which successfully removes the risk from real-time interaction with the environment. However, addressing out-of-distribution data is still a challenging task for offline RL [6].

Recent advances in generative artificial intelligence (GAI) models have opened a new possibility of offline RL, which is GAI-based offline RL [7]. By utilizing the ability of GAI to approximate an arbitrary probability function, GAI-based offline RL aims to extract the probability function of the expert dataset. Specifically, diffusion model is applied to the fields of trajectory planning as described in Fig. 1 [8]. These GAI models are reported to have ability to generalize in unknown data, compensating the offline RL's deficit.

This research was supported by Mechanical Equipment Industry Technology Development Program through the Korea Planning & Evaluation of Industrial Technology (KEIT) funded by the Korea government (Ministry of Trade, Industry and Energy (MOTIE)) (RS-2024-00442168).*(Corresponding author: Joongheon Kim)*

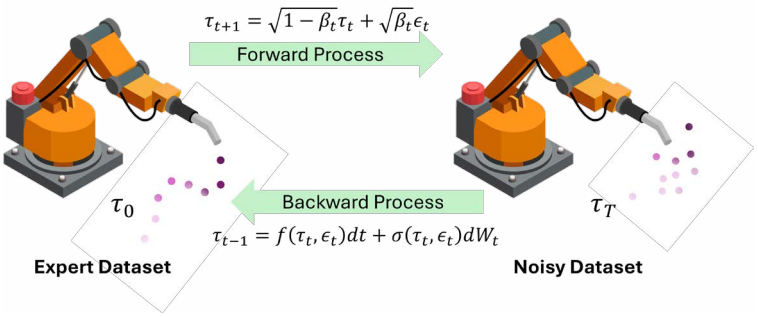

Fig. 1: An schematic illustration of diffusion-based trajectory planner. Firstly, initial trajectory data τ_0 is transformed into noisy data τ_T, by the forward process. After that, the model learns to denoise τ_T using backward process.

Motivated by this, this paper examines the performance of the GAI-based trajectory planning algorithms with other baselines. In addition, this paper analyzes the property of these algorithms.

II. Preliminaries

A. Diffusion Models

A diffusion model is proposed to approximate dataset's probability distribution, enabling sampling from it. Diffusion model first gradually adds noise to original state x_0, yielding noisy state x_T. After that, the model learns to reconstruct the original state from the noisy state by approximating ϵ_t, which is the noise at timestep t. Here, the former is called as forward process and the latter is called as backward process.

To guide diffusion model to sample with specific condition such as label, several guidance methods are developed. Classifier-guidance is the first algorithm proposed for guiding diffusion model [9]. Classifier-guidance independently trains another module that predicts the label of the noisy data. After that, it provides the gradient $\nabla \log p(y|x_t)$ to the diffusion model so that it can sample from the distribution $p(x_0|y) \simeq p(x_0)p(y|x_0)$.

B. Related Work

Before the advent of learning-based control, decision-making in dynamic systems relied on methods such as Proportional–Integral–Derivative (PID), Linear Quadratic Regulator (LQR), $\mathcal{H}_\infty$ robust control, Model Predictive Control (MPC), and Sliding Mode Control (SMC) [10]. While these approaches offered advantages in simplicity, optimality, robustness, or

constraint handling, they shared key drawbacks: strong model dependence, poor adaptability to nonlinear or uncertain dynamics, and high computational or tuning burdens. These limitations motivated the shift toward learning-based approaches [11].

Among the learning-based methods, reinforcement learning (RL) addresses aforementioned issues by enabling agents to learn policies through environment interaction without explicit mathematical calculation [12]. To reduce online interaction demands, offline RL leverages static datasets, improving safety and cost-effectiveness, though generalization beyond dataset distributions remains challenging [13].

To further address data inefficiency and distributional shift, generative model-based RL has emerged. Diffusion approaches such as Diffuser generate action trajectories through denoising, enabling multimodal behavior modeling but with heavy computational cost [14]. These GAI-based RL models reframe RL as sequence modeling, capturing long-term dependencies and reusing offline datasets effectively [15].

III. Algorithm Details

A. Problem Formulation

Optimal control problem can be modeled as inference problem over MDP defined as tuple $\langle S, A, R, \gamma, P\rangle$. Here, each component in the tuple denotes the set of states, set of actions, reward function, discount factor, and transition probability. Main objective of the optimal control problem is to find $\pi(a|s)$ such that maximizes cumulative reward of the agent, i.e., $\mathbb{E}[\sum r(s_t, a_t)]$. To explicitly calculate the probability of the trajectories with high reward, let $\mathcal{O}_t$ a binary random variable that denotes the optimality of the action a_t in the situation s_t. In addition, define the conditional probability of $\mathcal{O}_t$ as follows,

$$p(\mathcal{O}_t = 1|s_t, a_t) = \exp(r(s_t, a_t)). \tag{1}$$

Above equation implies that the higher the reward is, higher the probability of that trajectory τ. Then, the goal of the control is to sample the trajectory that has highest $p(\tau|\mathcal{O})$. Using similar defactorization with classifier-guidance, $\log p(\tau|\mathcal{O})$ can be stated as $\log p(\tau|\mathcal{O}) \simeq \log p(\tau) + \log p(\mathcal{O}|\tau)$. To find a trajectory that satisfies above property, the model should learn these two kinds of attributes from the data.

B. Algorithm Details

This paper introduces diffusion-based trajectory planning algorithm, as described in Fig. 2. The main objective of the algorithm is to approximate the original distribution $p(\tau)$. To solve aforementioned problem using diffusion model, this paper defines forward process as follows,

$$\tau_{t+1} = \sqrt{1-\beta_t}\tau_t + \sqrt{\beta_t}\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \mathbb{I}). \tag{2}$$

where β_t is pre-defined monotonic increasing function. Noise ϵ_t sampled from Gaussian process is added to the original trajectory, making it noisy. To reverse this noisy state into denoised state, this paper utilizes DDIM method, which greatly

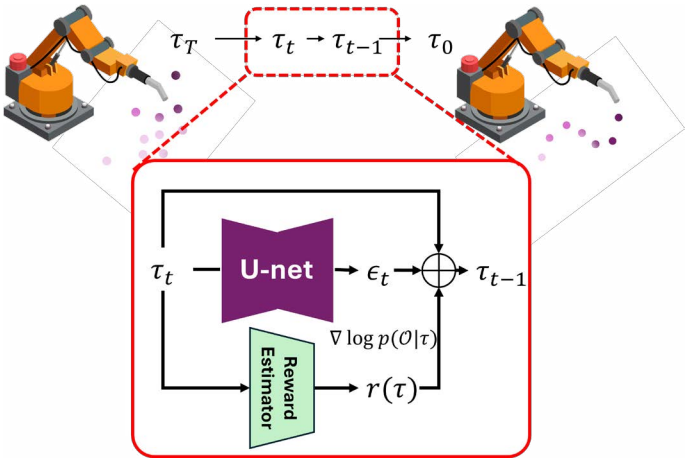

Fig. 2: An illustration of the algorithm structure.

enhances the inference speed of the model [16]. Thus, backward process is defined as follows,

$$\tau_{t-1} = \sqrt{\alpha_{t-1}}\left(\frac{\tau_t - \sqrt{1-\alpha_t}\cdot\hat{\epsilon}_t(\tau_t)}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\hat{\epsilon}^t(\tau_t) + \sigma_t\hat{\epsilon}_t. \tag{3}$$

where $\sigma_t \sim \mathcal{N}(0, \mathbb{I})$ and $\alpha_t = 1 - \beta_t$. Here, $\hat{\epsilon}_t$ is the output of the model given input τ_t. In order to get accurate denoised state, the model minimizes below loss,

$$\mathcal{L} = \|\hat{\epsilon}_t - \epsilon_t\|^2. \tag{4}$$

The diffusion model is trained to estimate ϵ_t, which is the noise applied to the trajectory τ_t. Additional reward estimator is independently trained to provide a gradient $\nabla \log p(\mathcal{O}|\tau)$. Reward estimator minimizes below loss,

$$\mathcal{L}_r = \|\hat{r}(s_t, a_t) - r(s_t, a_t)\|^2, \tag{5}$$

where $\hat{r}(s_t, a_t)$ is the output of the reward estimator. This reward estimator provides $\nabla \log p(\mathcal{O}|\tau)$, which is the second term of the objective function.

Because generated trajectory $\hat{\tau}_0$'s starting point $\hat{s}_0$ is tailored to s_0, Diffuser algorithm uses same technique as impainting. In particular, after the backward step $t = 1, \cdots, T$, starting point $\hat{s}_t$ is replaced to s_t, forcing the diffusion model to generate trajectory that satisfies $\hat{s}_t = s_t$.

For the implementation of the diffusion model, this paper adopts U-net structure utilizing 1D convolution. Moreover, for the implementation of the reward estimator, this paper chooses CNN structure with multi-head attention, utilizing 1D convolution. For the inference,

IV. Performance Evaluation

A. Experiment Setup

This paper verifies the proposed algorithm's performance using the door environment in D4RL dataset, which is represented by high-dimensional state and action space [17]. For comparison, this paper selects DDPG and SAC, which are RL algorithms for continuous actions [18], [19].

TABLE I: A comparison between the GAI-based RL with conventional RL algorithms.

Algorithms	Final Rewards (Normalized)
DDPG	-40.87
SAC	-28.59
Diffusion-based	**-0.202**

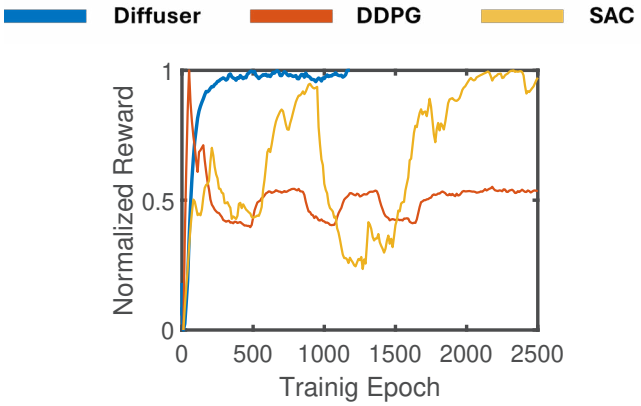

Fig. 3: Normalized Rewards of the GAI-based RL and conventional RL algorithms.

B. Experiment Results

As can be seen in Table I and Fig. 3, the proposed diffusion-based algorithm shows equal or better performance compared to other baselines. In particular, the proposed diffusion-based RL shows 99.5% and 99.2% higher performance compared to DDPG, SAC. In addition, the proposed diffusion-based RL shows faster convergence compared to other algorithms. Considering that DDPG and SAC are online algorithms, the result implies that the proposed diffusion-based RL can successfully replace conventional algorithms, eliminating the risk from direct interaction with the environment. Note that diffuser model rapidly achieves very high performance, unlike other RL algorithms.

V. Concluding Remarks

Conventional RL algorithms face danger during the process, which is from real-time interaction with the environment. To remove the interaction from the learning process, GAI technologies are integrated with the offline RL. Experiment results show that the proposed diffusion-based RL algorithm achieves similar performance with previous RL algorithms without interaction with the environment.

References

[1] W. Huang, H. Liu, Z. Huang, and C. Lv, "Safety-aware human-in-the-loop reinforcement learning with shared control for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 11, pp. 16 181–16 192, Nov. 2024.

[2] Y. Cho, H. Lee, S. Park, and J. Kim, "Joint multi-agent reinforcement learning and message-passing for distributed multi-uav network management using conflict graphs," in *Proc. IEEE Network Operations and Management Symposium*. Honolulu, HI, USA: IEEE, May 2025, pp. 1–5.

[3] R. Figueiredo Prudencio, M. R. O. A. Maximo, and E. L. Colombini, "A survey on offline reinforcement learning: Taxonomy, review, and open problems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 8, pp. 10 237–10 257, Aug. 2024.

[4] S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, and A. Knoll, "A review of safe reinforcement learning: Methods, theories, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 11 216–11 235, Sept. 2024.

[5] J. Zhu, C. Du, and G. E. Dullerud, "Model-based offline reinforcement learning with uncertainty estimation and policy constraint," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 12, pp. 6066–6079, Dec. 2024.

[6] C. Zhang, S. R. Kuppannagari, and V. K. Prasanna, "BRAC+: Improved Behavior Regularized Actor Critic for Offline Reinforcement Learning," in *Proc. Asian Conference on Machine Learning, ACML 2021, 17-19 November 2021, Virtual Event*, ser. Proc. Machine Learning Research, V. N. Balasubramanian and I. W. Tsang, Eds., vol. 157. PMLR, 2021, pp. 204–219.

[7] K. A. Yau, Y. Chong, X. Fan, F. Nejati, M. K. Chamran, and S. Darmaraju, "Combinations of generative adversarial network and reinforcement learning: A survey," *Neurocomputing*, vol. 650, p. 130847, 2025.

[8] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with Diffusion for Flexible Behavior Synthesis," in *Proc. International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ser. Proc. Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 2022, pp. 9902–9915.

[9] P. Dhariwal and A. Q. Nichol, "Diffusion models beat gans on image synthesis," in *Proc. Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., Virtual, Dec. 2021, pp. 8780–8794.

[10] R. Roy, M. Islam, N. Sadman, M. A. P. Mahmud, K. D. Gupta, and M. M. Ahsan, "A Review on Comparative Remarks, Performance Evaluation and Improvement Strategies of Quadrotor Controllers," *Technologies*, vol. 9, no. 2, pp. 1–21, May 2021.

[11] R. R. Faria *et al.*, "Where Reinforcement Learning Meets Process Control: Opportunities and Challenges," *Processes*, vol. 10, no. 11, pp. 1–29, 2022.

[12] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[13] R. F. Prudencio, M. R. O. A. Máximo, and E. L. Colombini, "A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 35, no. 8, pp. 10 237–10 257, 2024.

[14] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion Policy: Visuomotor Policy Learning via Action Diffusion," in *Proc. Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, K. E. Bekris, K. Hauser, S. L. Herbert, and J. Yu, Eds., 2023.

[15] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision Transformer: Reinforcement Learning via Sequence Modeling," in *Proc. Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 15 084–15 097.

[16] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. International Conference on Learning Representations, Virtual Event, Austria, May 3-7, 2021*, Virtual, May 2021.

[17] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, "D4RL: datasets for deep data-driven reinforcement learning," *CoRR*, vol. abs/2004.07219, 2020.

[18] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *Proc. International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016.

[19] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *Proc. International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proc. Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 1856–1865.