A FRAMEWORK FOR POLICY EVALUATION ENHANCEMENT BY DIFFUSION MODELS

Tao Ma & Xuzhi Yang *
London School of Economics and Political Science

ABSTRACT

Reinforcement learning plays an important role in various fields, and has fast development in policy evaluation and learning methods, enjoying the advantages of large data size. However, when data are limited, directly applying evaluation methods does not necessarily result in a good policy evaluation. In this work, we provide a framework to generate synthetic data with diffusion models, to enhance data-efficient policy evaluation, which is supported by experiments.

1 INTRODUCTION AND RELATED WORK

Reinforcement learning (RL, Sutton & Barto, 2018) provides a comprehensive approach for an agent to find the optimal policy in a data-driven way, and has successfully boosted the development of many fields, including biology (Jumper et al., 2021), games (Silver et al., 2017), finance (Deng et al., 2016) and education (Park et al., 2019). In general, the RL problem concerns an agent trying to optimize her long-term return by adjusting the policy, according to interactions with the unknown environment. To achieve such a goal, many algorithms contain two important components, policy evaluation and policy learning (Levine et al., 2020). With either online interactions or a large pool of offline data, the performances of policy evaluation and learning are impressive, both empirically and theoretically (Riedmiller, 2005; Mnih et al., 2015; Kumar et al., 2020). But there are numerous cases where data availability is quite limited (Dulac-Arnold et al., 2019; Nie et al., 2022), largely influencing the effectiveness of algorithms, which is the first challenge.

On the other side, generative models provide a powerful tool to understand the possible distribution, from which data are sampled, and generate more similar data (Oussidi & Elhassouny, 2018). Successful methods have been proposed, like Generative Adversarial Nets (Goodfellow et al., 2014) and diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020), with analysis of the resulting distribution (Chen et al., 2022). Given a diffusion model, one can generate more data to improve training. Such successes have been observed in the accuracy of image classification (Azizi et al., 2023), the pre-training for transfer learning (He et al., 2022), digital pathology (Pozzi et al., 2023) and brain computer interface (Tosato et al., 2023). There are also a few works in RL to regularize the learned policy using diffusion models (Wang et al., 2022). However, most work focus on specific applications and lack for theoretical discussion, which is the second challenge.

Facing above significant challenges, we propose a framework to generate high-quality synthetic data by diffusion models, to enhance policy evaluation. Contributions are two-fold: 1) our method results in data-efficient policy evaluation; 2) by pointing out both advantages and limitations of diffusion models, our method serves as a framework for further theoretical investigation and implementations.

2 BACKGROUND

RL. Define a Markov Decision Process (MDP, Puterman, 2014) $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, P, R, \gamma, T\}$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ is the action space and $P(\mathbf{s}'|\mathbf{s}, \mathbf{a}) : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the transition kernel. $R(\mathbf{s}, \mathbf{a})$ is the reward function, with $\gamma \in [0, 1]$ the discount factor. Finally, T specifies the horizon of one episode. Given $\mathcal{M}$, the policy of an agent is $\pi(\mathbf{a}|\mathbf{s})$, the distribution over $\mathcal{A}$ given $\mathbf{s} \in \mathcal{S}$. With the above, an episodic MDP scenario is as follows. In the first episode, the agent starts at the fixed initial state $\mathbf{s}_0$ (it can be generalized to the case with initial distribution). She

*Equal contribution. Correspondence to: Tao Ma (t.ma9@lse.ac.uk).

Table 1: Average errors and standard deviations of the estimated values based on SYN and REAL.

	REAL	SYN		
m	/	100	500	1000
Average error	0.527	0.557	0.577	0.562
Standard deviation	0.629	0.435	0.448	0.469

selects action $\mathbf{a}_0$ according to π, receives reward $r_0 \sim R(\mathbf{s}_0, \mathbf{a}_0)$ and is moved to the next state $\mathbf{s}_1 \sim P(\mathbf{s}_1|\mathbf{s}_0, \mathbf{a}_0)$. Following that, at time t, the agent has a transition tuple $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$. Such process goes on until $t = T$. Then a new episode starts again from $\mathbf{s}_0$. The value of π is $V^\pi(\mathbf{s}_0)$, with $V^\pi(\mathbf{s}) := \mathbb{E}_\pi[\sum_{i=0}^T \gamma^t r_t|\mathbf{s}_0 = \mathbf{s}]$. RL aims to evaluate the values of policies, and then find the optimal policy to maximize the value. Our work is to enhance the first step with less real data.

Diffusion Models. Diffusion models, given data from unknwon distribution q, slowly diffuse such distribution, through Markov chain $\{\mathbf{x}_i\}_{i=0}^N$, towards a known prior $\mathcal{N}(\mathbf{x}_N; \mathbf{0}, \mathbf{I})$, and then perform a reverse process to generate more data that approximately follow q. Here we focus on one version, called Denoising Diffusion Probabilistic Model (DDPM, Ho et al., 2020), with details in Section A.

Problem Settings. Given a policy π, and n episodes of transition tuples according to π, we aim to evaluate $V^\pi(\mathbf{s}_0)$. One significant problem is the lack of enough data, i.e. n is small. In the following, we would use DDPM to generate more episodes for better evaluation of $V^\pi(\mathbf{s}_0)$.

3 Framework for Policy Evaluation with Diffusion Models

Framework. Here we only store the reward in each step, meaning that the i-th data point is a vector of $T+1$ consecutive rewards from the i-th episode, denoted by $\mathbf{r}^{(i)} := (r_0^{(i)}, ..., r_T^{(i)})$, and dataset is $\mathcal{D} = \{\mathbf{r}^{(i)}\}_{i=1}^n$. The method consists of three steps. To begin with, the data are randomly partitioned into two parts $\mathcal{D}_1, \mathcal{D}_2$, each with size $n/2$. Second, we use $\mathcal{D}_1$ to train a DDPM and generate a set of synthetic data $\mathcal{D}_{\text{syn}} = \{\tilde{\mathbf{r}}^{(i)}\}_{i=1}^m$. Finally, we concatenate $\mathcal{D}_2$ and $\mathcal{D}_{\text{syn}}$, and compute our value estimator $\widehat{V} = (\sum_{i=1}^m \widetilde{V}_i + \sum_{i=1}^{n/2} V_i)/(m+n/2)$, where $\widetilde{V}_i = \sum_{j=0}^T \gamma^j \tilde{r}_j^{(i)}$ for generated trajectories $i \in [m]$ and $V_i = \sum_{j=0}^T \gamma^j r_j^{(i)}$ for $i \in [n/2]$ from $\mathcal{D}_2$ (other evaluation methods can also be used, like FQE (Munos & Szepesvári, 2008)). In Appendix the outline is provided in Algorithm 1.

Experiments. We implement our method in the Pendulum environment (Seno & Imai, 2022) with DDPM module (Wang, 2023). A near-optimal policy π is trained by CQL (Kumar et al., 2020). First we evaluate π on $n/2$ real data concatenated with m synthetic data, i.e. $\mathcal{D}_2 \cup \mathcal{D}_{\text{syn}}$ (SYN), with $n = 10$ and $m \in \{100, 500, 1000\}$ respectively. Then perform evaluation on n real data $\mathcal{D}_1 \cup \mathcal{D}_2$ (REAL). Repeat this procedure for 10 times, we compute the average gap from each method to the true value, together with standard deviations (see Table 1). Further interpretations are in Section C. Observably, involving only half real data in mean calculation, our methods provides an estimation comparable to those from more real data with even lower standard deviations.

Theoretical analysis. For DDPM, it can be proved that the total variation distance between the generated distribution and the underlying distribution, from which real data are sampled, is small. However, such bound is not enough to restrict the estimation gap $|\widehat{V} - V|$ in terms of either high probability or expectation, because the total variation is not strong enough to bound L^p-norm. We want to point out this limitation since it shows how far in current theory diffusion models can go. Therefore, future work can either try to show the bound in L^p-norm for current models or design new methods to accommodate such guarantees. See Section D for detailed discussions.

4 Conclusion and Future Work

In this work, we propose a general framework to improve policy evaluation by synthetic data. We are aware that the scale of numerical experiments is not enough, due to our very limited computing resources. So the implementation of our method later in large-scale settings is expected.

URM STATEMENT

Both authors meet the URM criteria of ICLR 2024 Tiny Papers Track.

REFERENCES

Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J. Fleet. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*, 2023.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv preprint arXiv:2209.11215*, 2022.

Valentin De Bortoli. Convergence of denoising diffusion models under the manifold hypothesis. *arXiv preprint arXiv:2208.05314*, 2022.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3):653–664, 2016.

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

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence for score-based generative modeling with polynomial complexity. *Advances in Neural Information Processing Systems*, 35:22870–22882, 2022.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(5), 2008.

Allen Nie, Yannis Flet-Berliac, Deon Jordan, William Steenbergen, and Emma Brunskill. Data-efficient pipeline for offline reinforcement learning with limited data. *Advances in Neural Information Processing Systems*, 35:14810–14823, 2022.

Achraf Oussidi and Azeddine Elhassouny. Deep generative models: Survey. In *International Conference on Intelligent Systems and Computer Vision*, pp. 1–8. IEEE, 2018.

Hae Won Park, Ishaan Grover, Samuel Spaulding, Louis Gomez, and Cynthia Breazeal. A model-free affective reinforcement learning approach to personalization of an autonomous social robot companion for early literacy education. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 687–694, 2019.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Matteo Pozzi, Shahryar Noei, Erich Robbi, Luca Cima, Monica Moroni, Enrico Munari, Evelin Torresani, and Giuseppe Jurman. Generating synthetic data in digital pathology through diffusion models: a multifaceted approach to evaluation. *medRxiv*, 2023.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Martin Riedmiller. Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*, pp. 317–328. Springer, 2005.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241. Springer, 2015.

Takuma Seno and Michita Imai. d3rlpy: An offline deep reinforcement learning library. *Journal of Machine Learning Research*, 23(315):1–20, 2022.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Giulio Tosato, Cesare M Dalbagno, and Francesco Fumagalli. Eeg synthetic data generation using probabilistic diffusion models. *arXiv preprint arXiv:2303.06068*, 2023.

Phil Wang. Denoising-diffusion-pytorch. `https://github.com/lucidrains/denoising-diffusion-pytorch`, 2023.

Zhendong Wang, Jonathan J. Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022.

A DETAILS FOR BACKGROUND

A.1 POLICY EVALUATION

As introduced in Section 2, for an episodic MDP with a time-homogeneous transition kernel and reward function, together with an agent adopting a time-homogeneous policy π, the evaluation of such policy is by $V^\pi(\mathbf{s}_0)$. To generalize the case from fixed initial state to initial distribution, we consider $\mathbf{s}_0 \sim \nu_0$, then the value can be summarized as

$$V^\pi := V^\pi(\nu_0) = \langle V^\pi(\cdot), \nu_0(\cdot) \rangle,$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product by taking the sum of all element-wise multiplications. On the other hand, if the MDP $\mathcal{M}$ is a trajectory with infinite horizon, the evaluation can be defined as either the average reward,

$$\langle d^\pi(\cdot), V^\pi(\cdot) \rangle$$

or by average reward per time step, i.e.

$$\langle d^\pi(\cdot), \sum_{a \in \mathcal{A}} \pi(a|\cdot) R(\cdot, a) \rangle,$$

where $d^\pi(s)$ is the stationary distribution. Here we want to emphasize that even with such generalization, the proposed framework need no modification in terms of evaluation. The reason is that with real data containing multiple trajectories, the randomness due to initial distribution is also considered, so our value estimator is then a consistent estimator of V^π.

There are many approaches in policy evaluation, both for on-policy or off-policy, and examples include Conservative q-learning (Kumar et al., 2020) and Fitted q evaluation (Munos & Szepesvári, 2008). Especially, better/efficient evaluation serves as the foundation for policy learning, which, in the general sense, is to choose the policy that has the largest evaluation of value.

A.2 DDPM

For each given data $\mathbf{x}_0 \sim q$, we construct a Markov chain $\{\mathbf{x}_i\}_{i=0}^T$ with

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}),$$

where Gaussian noises are gradually injected by

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}),$$

with a pre-specified variance schedule $\{\beta_t\}_{t=1}^T$. Then the diffusion model is by a model of latent variables $\{\mathbf{x}_t\}_{t=1}^T$, sharing the same dimensionality with real data, so that we can construct the reverse process $p_\theta(\mathbf{x}_{0:T})$ parameterized by θ, which is still a Markov chain from time index T towards 0 by

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t),$$

where the starting distribution is $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$ and transitions are still normal with

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)).$$

Then to make $p_\theta(\mathbf{x}_0)$ as close as possible to q, we need to optimize the choice of θ, and we refer readers to the details of training such model in Ho et al. (2020).

A.3 THEORETICAL GUARANTEE OF DDPM

For theoretical analysis of the resulting distribution, the forward and backward processes can be accordingly expressed by stochastic differential equations (SDE), as in Chen et al. (2022). First, given the forward process $(\mathbf{x}_t)_{t \in [0,T]}$, for clarity, we denote the reverse process by $\mathbf{x}_t^{\leftarrow}$ for $t \in [0, T]$,

Algorithm 1 Evaluation with Synthetic Data by DDPM

Input: Real data $\mathcal{D} = \{\mathbf{r}^{(i)}\}_{i=1}^n$, where $\mathbf{r}^{(i)} = (r_0^{(i)}, ..., r_T^{(i)})$.

1: Randomly split $\mathcal{D}$ into two halves $\mathcal{D}_1$ and $\mathcal{D}_2$, where WLOG, assume the first half of $\mathcal{D}$ are exactly the $\mathcal{D}_2$.

2: Train a DDPM with $\mathcal{D}_1$, and generate a new synthetic data set $\mathcal{D}_{\text{syn}}$ with $|\mathcal{D}_{\text{syn}}| = m$.

3: Compute $\widehat{V}$ based on a specific evaluation method, like Monte Carlo, FQE, or conservative q, using data $\mathcal{D}_{\text{syn}} \cup \mathcal{D}_2$.

Return: $\widehat{V}$

so that $\mathbf{x}_0^{\leftarrow}$ follow a standard normal distribution and serves as the starting point of the reverse process to generate finally $\mathbf{x}_T^{\leftarrow}$. Then it can be shown that the reverse process has the form

$$\mathrm{d}\mathbf{x}_t^{\leftarrow} = \{\mathbf{x}_t^{\leftarrow} + 2\nabla \ln q_{T-t}(\mathbf{x}_t^{\leftarrow})\}\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

where q_t, for $t \geq 0$, is the law of $\mathbf{x}_t$ from the forward process, and with abuse of notation, $(B_t)_{t\in[0,T]}$ is the reversed Brownian motion. The insight here is that to perform the above reverse process, we need to know the so-called score functions $\nabla \ln q_t$. Since all laws q_t are unknown due to the unknown starting distribution q, we first discretize the reverse process with step size $h > 0$ and total steps N, so that $T = Nh$, and in the k-th step of discretized process, i.e. $t \in [kh, (k+1)h]$, its SDE is

$$\mathrm{d}\mathbf{x}_t^{\leftarrow} = \{\mathbf{x}_t^{\leftarrow} + 2w_{T-kh}(\mathbf{x}_{kh}^{\leftarrow})\}\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t, \quad t \in [kh, (k+1)h],$$

where w_t is the estimator of score function (chosen from a class of neural networks) by

$$\min_{w_t} \mathbb{E}_{q_t}[\|w_t - \nabla \ln q_t\|_2^2].$$

Then such generation can be realized, corresponding to the original definition of DDPM. Especially, some theoretical guarantees hold.

According to Chen et al. (2022), if we assume $\nabla \ln q_t$ is L-Lipschitz, q has finite second moment and the above score estimation error is uniformly bounded by $\epsilon_{\text{score}}^2$, then by choosing specific T and h (both dependent on target error level ϵ), hiding logarithmic factors, we can have

$$\mathrm{TV}(p_T, q) \leq \widetilde{O}(\epsilon + \epsilon_{\text{score}}), \quad \text{for } N = \widetilde{\Theta}\left(\frac{L^2 d}{\epsilon^2}\right),$$

where p_T is the law of $\mathbf{x}_T^{\leftarrow}$ and d is the dimension of each data point.

B Method Implementation Details

Here we provide the outline of our method described in Section 3, by the following Algorithm 1. It should be emphasized that we train our DDPM using the first half of $\mathcal{D}$ but concatenate synthetic data with the second half to guarantee independence, which could potentially lead to the theoretical properties of the estimator. In addition, although in this work we focus on Monte Carlo estimators, any appropriate evaluation method can be incorporated, justifying our proposal as a framework.

C Experiment Details and Further Results Interpretations

C.1 Environment Settings

The concerned environment has a pendulum with one end fixed, and an agent needs to sequentially apply torque to the other free-swinging end to have it closer to the upright position. At each time the reward measures the gap away from such an optimal position. The observation data is a three-dimensional vector which indicates the x-y coordinate of the pendulum's free end and the angular velocity. The action space is the torque applied to the free end of the pendulum and can range from -2.0 to 2.0. The reward function is defined as $r(\theta, v, t) = -(\theta^2 + 0.1v^2 + 0.001 * t)$, where θ is the pendulum's angle, v denotes the angular velocity and t is the applied torque. Thus the maximal possible reward is zero.

C.2 IMPLEMENTATION DETAILS

Our model is implemented using PyTorch version 2.1.1+cu121 (Paszke et al., 2017), d3rlpy version 2.3.0 (Seno & Imai, 2022), and denoising_diffusion_pytorch (Wang, 2023). We collect 500 episodes from the pendulum environment, and then based on this dataset, train a near-optimal policy π using CQL (Kumar et al., 2020). With this trained policy π, we first generate $\mathcal{D}_{\text{true}}$ with 8000 real episodes from the pendulum environment as the benchmark dataset to compute approximately the true value of π. Then we sample n real episodes again from the environment as $\mathcal{D}_1 \cup \mathcal{D}_2$, with $n = 10$, $|\mathcal{D}_1| = |\mathcal{D}_2| = 5$, and all episodes have the same horizon 128. To accommodate the randomness when splitting n real episodes into two sets, we apply a similar manner as cross-validation in the following. We first use respectively $\mathcal{D}_1$ and $\mathcal{D}_2$ to train two DDPMs, and then generate two synthetic sets of trajectories in rewards, each with size m, i.e. $\mathcal{D}_{\text{syn},1}$ and $\mathcal{D}_{\text{syn},2}$, where $m \in \{100, 500, 1000\}$, and each trajectory contains 128 simulated rewards. Then, we calculate two estimated values based on $\mathcal{D}_2 \cup \mathcal{D}_{\text{syn},1}$ and $\mathcal{D}_1 \cup \mathcal{D}_{\text{syn},2}$ respectively, and take the average of them. Finally, treating the average reward based on the benchmark dataset $\mathcal{D}_{\text{true}}$ as the underlying truth, we compare the absolute value gaps due to our estimations (SYN dataset) with the gaps based on $\mathcal{D}_1 \cup \mathcal{D}_2$ (REAL dataset). We repeat this procedure for 10 times to get the average errors and standard deviations. Given the 1D nature of the data, we employ Unet1D to estimate the score function (Ronneberger et al., 2015), and the diffusion model is trained using GaussianDiffusion1D and Trainer1D, as detailed in Wang (2023). While most tuning parameters are maintained at their default values, we set the number of iterations to train the diffusion model to 8000 due to limited computation resources.

C.3 FURTHER INTERPRETATIONS OF RESULTS

As the synthetic data are generated by diffusion models, even with larger sizes of synthetic data, it should still be checked if the estimation is robust enough. In additional to average estimation errors, here we have also provided further results in the standard deviations of estimations in Table 1. It can be seen that, across different choices of m (the size of generated data), the standard deviations due to the proposed method, using the augmented data (with a smaller number of real data, i.e. $n/2$), tend to be lower than those due to using solely real data (n), which shows the robustness of the proposed estimation method. As discussed in the above, the total number of diffusion steps we use is not perfectly high enough, due to the limited computing resources. And with future improvement in such aspect, the proposed method is expected to have even better performances.

D THEORETICAL DISCUSSIONS

D.1 TRAINING ON REWARDS

One possibly asked question is that why we train the generative models only on rewards, instead of transition tuples? Actually they are the same. Given fixed policy and environment, the stream of transition tuples $(\mathbf{s}_0, \mathbf{a}_0, r_0, ..., \mathbf{s}_T, \mathbf{a}_T, r_T)$ would follow a high dimensional joint distribution f_0. At the same time, the vector of rewards $(r_0, r_1, ..., r_T)$ follows another joint distribution f_1. It can be seen that f_1 is the marginal distribution of rewards, from the primary joint distribution f_0. When we try to directly use all transition tuples to train a generative model, we are in fact approximating f_0, and $\widehat{V}(\mathbf{s}_0)$ would be an estimator of

$$V^{\pi}_{\text{truth1}}(s_0) := \mathbb{E}_{(\mathbf{s}_0, \mathbf{a}_0, r_0, ..., \mathbf{s}_T, \mathbf{a}_T, r_T) \sim f_0} \left[\sum_{t=0}^{T} \gamma^t r_t \;\middle|\; \mathbf{s}_0 \right]$$

On the other hand, the defined $\widehat{V}(\mathbf{s}_0)$ in our proposed method is an estimator of

$$V^{\pi}_{\text{truth2}}(s_0) := \mathbb{E}_{(r_0, r_1, ..., r_T) \sim f_1} \left[\sum_{t=0}^{T} \gamma^t r_t \;\middle|\; \mathbf{s}_0 \right]$$

Without loss of generality, we assume all variables take values continuously and all density functions are well-defined. Then notice, according to Fubini's Theorem,

$$\begin{aligned}
V^{\pi}_{\text{truth1}}(s_0) &= \mathbb{E}_{(\mathbf{s}_0,\mathbf{a}_0,r_0,\ldots,\mathbf{s}_T,\mathbf{a}_T,r_T)\sim f_0}\left[\sum_{t=0}^{T}\gamma^t r_t \;\middle|\; \mathbf{s}_0\right] \\
&= \int \left[\sum_{t=0}^{T}\gamma^t r_t\right] f_0(\mathbf{s}_0,\mathbf{a}_0,r_0,...,\mathbf{s}_T,\mathbf{a}_T,r_T)\,\mathrm{d}(\mathbf{s}_0,\mathbf{a}_0,r_0,...,\mathbf{s}_T,\mathbf{a}_T,r_T) \\
&= \int \left[\sum_{t=0}^{T}\gamma^t r_t\right] \left\{\int f_0(\mathbf{s}_0,\mathbf{a}_0,r_0,...,\mathbf{s}_T,\mathbf{a}_T,r_T)\,\mathrm{d}(\mathbf{s}_0,\mathbf{a}_0,...,\mathbf{s}_T,\mathbf{a}_T)\right\}\mathrm{d}(r_0,...,r_T) \\
&= \int \left[\sum_{t=0}^{T}\gamma^t r_t\right] f_1(r_0,...,r_T)\,\mathrm{d}(r_0,...,r_T) = V^{\pi}_{\text{truth2}}(s_0)
\end{aligned}$$

leading to the following result:

Lemma 1 (Training on rewards) *For any initial state* $\mathbf{s}_0$*, we have* $V^{\pi}_{\textit{truth1}}(s_0) = V^{\pi}_{\textit{truth2}}(s_0)$.

As a result, training on just rewards would not introduce bias, but is be more efficient, which is the reason we train our DDPM only on rewards.

D.2 Performance Guarantee Questions

Regarding the theoretical assurance provided by the proposed data enhancement framework related to Section A.3, we present several favourable statements arranged in descending order of strength (which, we hope, can possibly hold) and the main difficulties that hinder us from proving them rigorously. As a result, this subsection mainly serves as a standard list of goals for future work.

We first recall that the average reward based on the synthetic dataset is calculated as $\hat{V} = (\sum_{i=1}^{m}\tilde{V}_i + \sum_{i=1}^{n/2}V_i)/(m+n/2)$. Then the most straightforward statistical guarantee is to prove that for any $\eta > 0$, the following event

$$\left\{\left|\frac{1}{n}\sum_{i=1}^{n}V_i - V^{\pi}\right| - \left|\hat{V} - V^{\pi}\right| > \eta\right\} \tag{1}$$

holds with high probability. In essence, this asserts that the proposed estimator is likely to outperform an average constructed solely from n true data points. The establishment of this high probability event requires the L^p-convergence of DDPM, however, to the best of our knowledge, the existing works concentrate on the convergence in the total variation distance or the Wasserstein distance (De Bortoli et al., 2021; Chen et al., 2022; De Bortoli, 2022; Lee et al., 2022), which are metrics only metrize weak-topology on the space of probability distributions and are not enough to imply the L^p-convergence.

The second desirable statement to come down to is to prove for any $\mathcal{V} \subset \mathbb{R}^{n/2}$,

$$\mathbb{P}\Big(\left|\hat{V} - V^{\pi}\right| < \eta \;\middle|\; (V_i)_{i=\frac{n}{2}+1}^{n} \in \mathcal{V}\Big) \geq \mathbb{P}\Big(\left|\frac{1}{n}\sum_{i=1}^{n}V_i - V^{\pi}\right| < \eta \;\middle|\; (V_i)_{i=\frac{n}{2}+1}^{n} \in \mathcal{V}\Big), \tag{2}$$

where the condition is on the source data $(V_i)_{i=n/2+1}^{n}$ for training the DDPM. This statement is noteworthy for two reasons: 1) it provides a statistical guarantee not for each instance but for the overall policy evaluation problem; 2) the conditional probability ensures that regardless of the behaviour of the source dataset, the proposed data enhancement method is likely to outperform the trivial average of the original dataset. Because $\hat{V}$ is constructed partially from $\tilde{V}_i$'s, through the average of DDPM generated rewards, to prove (2), we first need to investigate the following

$$\left|\mathbb{P}\Big(\frac{1}{m}\sum_{j=1}^{m}\tilde{\mathbf{r}}^{(j)} \in G \;\middle|\; (V_i)_{i=\frac{n}{2}+1}^{n} \in \mathcal{V}\Big) - \mathbb{P}\Big(\frac{2}{n}\sum_{i=n/2+1}^{n}\mathbf{r}^{(i)} \in G \;\middle|\; (V_i)_{i=\frac{n}{2}+1}^{n} \in \mathcal{V}\Big)\right|, \tag{3}$$

for some set G, where $\mathbf{r}^{(i)}$ is the real reward vector of the i-th episode and $\tilde{\mathbf{r}}^{(j)}$ is the j-th DDPM generated reward vector. We point out that the closeness between the distributions of $\frac{1}{m}\sum_{j=1}^{m}\tilde{\mathbf{r}}^{(j)}$ and $\frac{2}{n}\sum_{i=n/2+1}^{n}\mathbf{r}^{(i)}$ in TV-distance only implies that, by taking expectation over all possible values of source data, the distributions of $\frac{1}{m}\sum_{j=1}^{m}\tilde{\mathbf{r}}^{(j)}$ and $\frac{2}{n}\sum_{i=n/2+1}^{n}\mathbf{r}^{(i)}$ will be close. Specifically, write $\tilde{\mu}$ and μ as the distribution of $\frac{1}{m}\sum_{j=1}^{m}\tilde{\mathbf{r}}^{(j)}$ and $\frac{2}{n}\sum_{i=n/2+1}^{n}\mathbf{r}^{(i)}$ respectively, what we can have due to the bound on TV-distance is

$$\left|\mathbb{E}\left[\mathbb{P}\left(\frac{1}{m}\sum_{j=1}^{m}\tilde{\mathbf{r}}^{(j)}\in G \;\middle|\; (V_i)_{i=\frac{n}{2}+1}^{n}\right)-\mathbb{P}\left(\frac{2}{n}\sum_{i=n/2+1}^{n}\mathbf{r}^{(i)}\in G \;\middle|\; (V_i)_{i=\frac{n}{2}+1}^{n}\right)\right]\right|\leq\|\tilde{\mu}-\mu\|_{\mathrm{TV}}.$$

However, a control on the average (or expectation) does not provide a control on each specific choice of source dataset. For instance, given some poor-quality source data, the distance (3) can be unreasonably large. All the existing error bounds on the TV-distance between the source distribution and the diffusion-generated distribution are too weak to provide an upper bound for (3). Therefore, statement (2) might also be hard to obtain given current theories on DDPM.

Finally, a weaker unconditional statement is the following,

$$\mathbb{P}\big(\big|\hat{V}-V^{\pi}\big|<\eta\big)\geq\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n}V_i-V^{\pi}\right|<\eta\right). \tag{4}$$

The limitation of this unconditional statement, compared to (2), lies in its dependence on both the generated data and the source data, which may hinder the proposed method from surpassing the original average if the source data quality is low. Now recall that $\hat{V}=(\sum_{i=1}^{m}\tilde{V}_i+\sum_{i=1}^{n/2}V_i)/(m+n/2)$. According to the denoising process, $\{\tilde{V}_i\}_{i=1}^{m}$ are generated from independent white noises, but with the same model, trained from the same source data. As a result, due to the randomness of sources data, $\{\tilde{V}_i\}_{i=1}^{m}$ are not exactly independent. Since the dependence structure is mysteriously introduced by the diffusion model, the traditional statistical tools could not be easily implemented.

In summary, there is currently lack of effective tools to prove any statistical guarantees about the performance of the DDPM synthetic data on policy evaluation. We look forward to future works on the diffusion model that can provide an error bound in a stronger metric. Especially, our systematic discussion here could hopefully inspire the path to complete theories.