Generative Sub-Trajectories Augmentation for Offline Policy Evaluation

Anonymous Authors

Abstract

Offline policy evaluation (OPE) sits at the epicenter of reinforcement learning (RL) research. It is particularly vital when it comes to real-world human-involved tasks, like e-learning and healthcare, where data may be scarce or not fully representative. Data augmentation has demonstrated considerable success in many tasks, including inducing RL policies, but many existing methods may not suit OPE, considering its Markovian nature and goal of generalizability over evaluation policies. We propose to facilitate OPE with Augmented Trajectories (OAT) through generative sub-trajectory learning, which would extract potential sub-trajectories and generate diverse behaviors in offline trajectories to enrich state-action space. In various simulation and real-world human-involved environments, including robotic control, healthcare, and e-learning, the effectiveness of OAT has been assessed against state-of-the-art data augmentation baselines, and our findings indicate that OAT can greatly improve OPE performance.

1. Introduction

Offline policy evaluation (OPE) has been recognized as an important part of reinforcement learning (RL), especially for human-involved RLs, in which evaluations of online policies can have high stakes (Levine et al., 2020). The objective of OPE is to evaluate target policies based on offline trajectories collected from behavioral policies different from the target ones. One major barrier often lies in the fact that the offline trajectories in human-involved tasks often only provide limited state-action coverage of the entire space. This can be caused by homogeneous behavioral policies; for example, during clinical procedures, physicians need to follow certain standardized guidelines. However, a sub-optimal autonomous control agent (*e.g.*, surgical robots under training) may deviate from such guidelines, and thus result in trajectories where the state-action visitations may not be fully covered by the offline trajectories collected, which introduces great challenges for OPE, as illustrated in Figure 1. Therefore, to improving the OPE performance, it is essential to enrich the offline trajectories.

Data augmentation is a powerful tool for data enrichment by artificially generating new data points from existing data. It has shown effectiveness in facilitating learning more robust supervised and unsupervised models (Iwana & Uchida, 2021a; Xie et al., 2020). Specifically, generative methods such as variational autoencoder (VAE) have achieved superior performance in time-series augmentation by capturing temporal and multivariate dependencies (Yoon et al., 2019; Barak et al., 2022). However, an important characteristic of OPE training data is the Markovian nature, as the environments are usually formulated as a Markov decision process (MDP) (Thomas & Brunskill, 2016; Fu et al., 2021). As a result, prior works on time-series augmentation may not be directly applicable to MDP trajectory augmentation.

Recently, though data augmentation methods have been extended to facilitate RL policy optimization, most existing works focus on enriching the state space, such as adding noise to input images to generate sufficient data and improve the generality of agents (Laskin et al., 2020b; Raileanu et al., 2021), but overlook the coverage of the joint state-action visitation distribution over time. More importantly, the goal of data augmentation towards RL policy optimization and OPE is different. Data augmentation in RL generally aims to quickly facilitate identifying and learning from high-reward regions of the state-action space (Liu et al., 2021; Park et al., 2022). In contrast, the evaluation policies considered by OPE can be heterogeneous and lead to varied performance, *i.e.*, the policies to be evaluated by OPE do not necessarily perform well; therefore, it is equally important to allow the agent learning from trajectories resulted from high- and low-reward regions. As a result, OPE methods prefer training data that provides comprehensive coverage of the state-action visitation space, including the trajectories resulting from low-performing and sub-optimal policies. To the best of our knowledge, there does not exist a method that augments historical trajectories specific to OPE.

In this paper, we propose a framework to facilitate **O**PE with **A**ugmented **T**rajectories (**OAT**) in human-involved systems. Specifically, motivated by the intrinsic nature that human-involved systems are often provided biased cover-

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

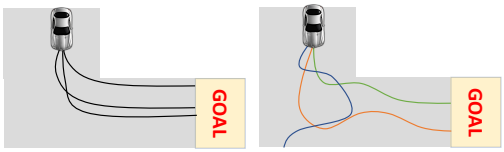

Figure 1. A conceptual illustration of the discrepancy between human demonstrations versus the empirical trajectories resulted from a sub-optimal policy to be evaluated by OPE. It can be observed that the autonomous agent may perform maneuvers unseen from the training (demonstration) trajectories, and thus can potentially hinder OPE's performance.

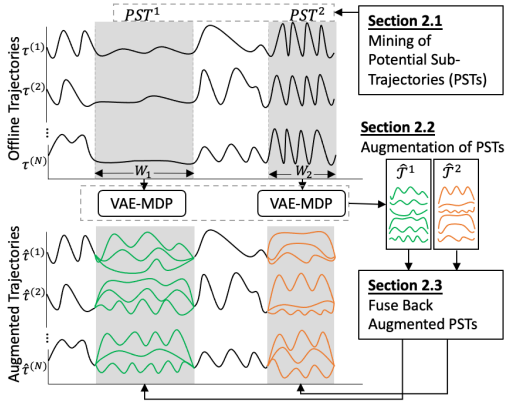

Figure 2. The illustration of OAT. It consists of three steps: (*i*) Mining of potential sub-trajectories (PSTs), where human behave similarly under behavioral policies at the grey-shaded area and may have more potential to enrich its state-action visitation space; (*ii*) VAE-MDP for augmenting PSTs; (*iii*) Stitch back augmented PSTs to trajectories.

age of the state-action visitation space and human may behave diversely when following heterogeneous policies (Yang et al., 2020b; Wang et al., 2022), we propose potential sub-trajectories (PSTs) mining to identify sub-trajectories of historical trajectories whose state-action visitation space is less covered but have great potential to enrich the space. Then a generative modeling framework is used to capture the dynamic underlying the PSTs and induce augmented sub-trajectories. Based on that, we design the stitching process by simultaneously taking the augmented sub-trajectories while maintaining the part of the state-action visitation distribution associated with non-PSTs. The proposed work is validated across various human-involved tasks, including robotic control, disease treatment, and intelligent tutoring. The key contributions of this work are summarized as follows: (*i*) To the best of our knowledge, OAT is the first method augmenting historical trajectories to facilitate OPE in human-involved systems. (*ii*) We conduct extensive experiments to validate OAT in a variety of simulation and real-world human-involved environments, including robotics, healthcare, and e-learning. (*iii*) The experimental results present that OAT can significantly facilitate OPE performance and outperforms all data augmentation baselines.

2. OPE with Augmented Trajectories (OAT)

We propose a framework to facilitate OPE with augmented trajectories (OAT) towards human-involved systems. Specifically, we first introduce offline trajectories and OPE. Then we propose a sub-trajectory mining method that identifies the sub-trajectories of trajectories that have great potential to increase the offline trajectories' coverage over the state-action space, *i.e., potential sub-trajectories (PSTs)*. A generative modeling framework is used to capture the dynamics underlying the selected PSTs, followed by a stitching process that generates augmented trajectories which will be used to train the OPE methods.

Offline Trajectories. We consider framing an agent's interaction with the environment over a sequence of decision-making steps as a Markov decision process (MDP), which is formulated as a 6-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{S}_0, r, \gamma)$. $\mathcal{S}$ is the state space. $\mathcal{A}$ is the action space. $\mathcal{P}$ defines transition dynamics from the current state and action to the next state. $\mathcal{S}_0$ defines the initial state distribution. r is the reward function. $\gamma \in (0, 1]$ is discount factor. Episodes are of finite horizon T. At each time-step t, the agent observes the state $s_t \in \mathcal{S}$ of the environment, then chooses an action $a_t \in \mathcal{A}$ following a policy π. The environment accordingly provides a reward $r_t = r(s_t, a_t)$, and the agent observes the next state s_{t+1} determined by $\mathcal{P}$. $\tau^{(i)}$ is defined as a *trajectory* where $\tau^{(i)} = [..., (s_t, a_t, r_t, s'_t, ...]_{t=1}^{T}$.

Offline Policy Evaluation (OPE). The goal of OPE is to estimate the expected total return over the *evaluation (target)* policy π, $V^{\pi} = \mathbb{E}[\sum_{t=1}^{T} \gamma^{t-1} r_t | a_t \sim \pi]$, using set of historical trajectories $\mathcal{D}$ collected over a *behavioral* policy $\beta \neq \pi$. The historical trajectories $\mathcal{D} = \{..., \tau^{(i)}, ...\}_{i=1}^{N}$ consist of a set of N trajectories.

2.1. Mining of Potential Sub-trajectories (PSTs)

The historical trajectories $\mathcal{D}$ collected from human-involved systems are often provided with biased coverage of the state-action space, due to the intrinsic nature that human may follow homogeneous behavioral policies or specific guidelines when performing their professions (Yang et al., 2020b; Wang et al., 2022). For example, a surgeon could perform appendectomy in various ways across patients depending on each patient's specific condition; however, they may strictly follow similar steps at the beginning (*e.g.*, disinfection) and the end of surgeries (*e.g.*, stitching). Therefore, the resulting trajectories may lead to limited coverage for part of the state-action space representing similar scenarios. However, a sub-optimal autonomous agent, subject to be evaluated by OPE, may visit states unseen from the trajectories collected from the surgeon, *e.g.*, towards the beginning/end of the surgery. As a result, we consider augmenting the part of trajectories, *i.e.*, the PSTs, that are more likely to be insufficiently covered by the historical trajectories $\mathcal{D}$. More-

over, the downstream generative models, such as VAEs, do not necessarily need to reconstruct entire trajectories for long horizons and over limited samples which are the common limitations of data collected from human-involved systems (Yacoby et al., 2020).

To identify the PSTs that are subject to be augmented, we introduce a three-step approach, *i.e.*, (i) discrete representation mapping (Gao et al., 2022) which encodes the original trajectories into one-dimensional temporal discrete sequences, followed by (ii) determining support from discrete representations, where the support is used in step (iii) to identify PSTs to be augmented.

Step (i) – Discrete Representation Mapping. In this step, we leverage Toeplitz inverse covariance-based clustering (TICC) (Hallac et al., 2017) to map states $s_t \in \mathcal{S}$ into C clusters, where each s_t is associated with a cluster from the set $\mathbf{K} = \{K_1, \dots, K_C\}$. The states mapped to the same cluster can be considered sharing graphical connectivity structure of both temporal and cross-attributes information captured by TICC. Discrete representation mapping has been recognized as effectively providing high-level abstractions from complex original data for supervised and unsupervised learning (Yang et al., 2021; Gao et al., 2022).

Step (ii) – Determine Support from Discrete Representations. After mapping, each state $s_{i,t}$ on trajectory $\tau^{(i)}$ is mapped to $K_{i,t} \in \mathbf{K}$. We assume that each trajectory $\tau^{(i)}$ can be mapped to a corresponding temporal discrete sequence $K^{(i)} = [K_{i,1}, \dots, K_{i,T}] \subset \mathbb{Z}^T$, based on the state mapping, where T is the horizon of the environment and $\mathbb{Z}$ is the set of integers. We also define $\mathcal{H} = \{..., K^{(i)}, ...\}_{i=1}^N$ which is the set of all temporal discrete sequences mapped from the set of original trajectories $\mathcal{D}$. We define $\delta^{(i)}_{\zeta,\zeta+W-1} = [K_{i,\zeta}, ..., K_{i,\zeta+W-1}]$ as a *temporal discrete sub-sequence (TDSS)* with length $\zeta \in [1, T-W+1]$ of $K^{(i)}$, $W \in [1, T]$, denoted as $\delta^{(i)}_{\zeta,\zeta+W-1} \sqsubseteq K^{(i)}$. Note that C is generally greatly smaller than $T \times N$ as considered in discrete representation mapping in general (Hallac et al., 2017; Yang et al., 2021). Therefore, it is possible that a temporal discrete sub-sequence $\delta^{(i)}_{\zeta,\zeta+W-1}$ is "equal" to another temporal discrete sub-sequence $\delta^{(j)}_{\zeta,\zeta+W-1}$, such that $\delta^{(i)}_{\zeta,\zeta+W-1} = \delta^{(j)}_{\zeta,\zeta+W-1}$ if every $K_{i,\zeta} = K_{j,\zeta}$ given $K_{i,\zeta}, K_{j,\zeta} \in \mathbb{Z}$. Then, the *support (or frequency)* of any TDSS $\delta^{(i)}_{\zeta,\zeta+W-1}$ appears in $\mathcal{H}$ can be calculated following the definition below.

Definition 2.1 (Support of Temporal Discrete Sub-Sequence)**.** Given the temporal discrete sequence dataset $\mathcal{H}$, the support of a temporal discrete sub-sequence $\delta^{(i)}_{\zeta,\zeta+W-1}$ is the number of $K^{(i)}$ in $\mathcal{H}$ containing $\delta^{(i)}_{\zeta,\zeta+W-1}$, i.e., $support_{\mathcal{H}}(\delta^{(i)}_{\zeta,\zeta+W-1}) = \sum_{j=1}^{N} \Big[\mathbb{1}(\delta^{(j)}_{\zeta,\zeta+W-1} \sqsubseteq K^{(j)}) \times \mathbb{1}(\delta^{(j)}_{\zeta,\zeta+W-1} = \delta^{(i)}_{\zeta,\zeta+W-1})\Big]$, where $support_{\mathcal{H}}(\cdot) \in \mathbb{Z}$ and $\mathbb{1}(\cdot)$ is the indicator function.

Step (iii) – Identify PSTs. We define $\varphi^{(i)}_{\zeta,\zeta+W-1} = [(s^{(i)}_{\zeta}, a^{(i)}_{\zeta}, r^{(i)}_{\zeta}, s'^{(i)}_{\zeta}), ..., (s^{(i)}_{\zeta+W-1}, a^{(i)}_{\zeta+W-1}, r^{(i)}_{\zeta+W-1}, s'^{(i)}_{\zeta+W-1})]$ as a sub-trajectory with length W of $\tau^{(i)}$. Given the mapping from trajectory $\tau^{(i)}$ to temporal discrete sequence $K^{(i)}$ (introduced in the step above), we define that each sub-trajectory $\varphi^{(i)}_{\zeta,\zeta+W-1}$ can be mapped to a corresponding TDSS $\delta^{(i)}_{\zeta,\zeta+W-1}$. Now we can identify the PSTs that will be used to train the generative model for reconstructing new sub-trajectories (*i.e.*, augmentation) in Section 2.2, following the definition below.

Definition 2.2 (Potential Sub-Trajectory (PST))**.** Given historical trajectories $\mathcal{D}$ and a threshold ξ, a sub-trajectory $\varphi^{(i)}_{\zeta,\zeta+W-1}$ is considered as a potential sub-trajectory if the support of its mapped temporal discrete sub-sequence $\delta^{(i)}_{\zeta,\zeta+W-1}$ satisfies $support_{\mathcal{H}}(\delta^{(i)}_{\zeta,\zeta+W-1}) \geq \xi$.

Following the step above, a set of PSTs is determined for historical trajectories $\mathcal{D}$, from which we can obtain a set of G *distinct* corresponding TDSSs $\{\delta^{g}_{\zeta,\zeta+W-1}\}$[1], $g \in [1, G]$ mapped from the PSTs. Then we can obtain G sets of PSTs, such that each set $\mathcal{T}^g = \{\varphi^{(i)}_{\zeta,\zeta+W-1}\}$, where all $\varphi^{(i)}_{\zeta,\zeta+W-1} \in \mathcal{T}^g$ satisfy that their corresponding $\delta^{(i)}_{\zeta,\zeta+W-1} = \delta^{g}_{\zeta,\zeta+W-1}$. Each set of PSTs may contain unique information captured from the original historical trajectories $\mathcal{D}$, as previous works have found that the PSTs in the same set, $\mathcal{T}^g$, are in general associated with similar temporal and cross-attributes correlations (Gao et al., 2022).

2.2. Augmenting the PSTs

In this section, we introduce how to adapt VAE to capture the MDP transitions, *i.e., VAE-MDP*, underlying each set of PSTs, $\mathcal{T}^g$, as well as reconstruct new PST samples that will be stitched with the original historical trajectories $\mathcal{D}$ for OPE methods to estimate the returns of evaluation (target) policies. The adaptation mainly consists of three parts: the latent prior, variational encoder, and generative decoder. Given a set of PSTs, $\mathcal{T}^g = \{\delta_{\zeta,\zeta+W-1}\}$[2], the formulation of VAE-MDP consists of three major components, *i.e.*, (i) the latent prior $p(z_\zeta)$ that represents the distribution of the initial latent states over $\mathcal{T}^g$, (ii) the encoder $q_\omega(z_t|s_{t-1}, a_{t-1}, s_t)$ that encodes the MDP transitions into the latent space, and (iii) the decoders $p_\eta(z_t|z_{t-1}, a_{t-1})$, $p_\eta(s_t|z_t)$, $p_\eta(r_{t-1}|z_t)$ that reconstructs new PST samples. The detailed setup can be found in Appendix A.2, and the overall encoding and decoding processes are illustrated in Figure 3.

[1]From now we use superscript ^{g} to replace $^{(i)}$ for δ's, since there may exist multiple TDSSs that are eqeuivalent.

[2]From now on we omit the superscripts of δ for conciseness.

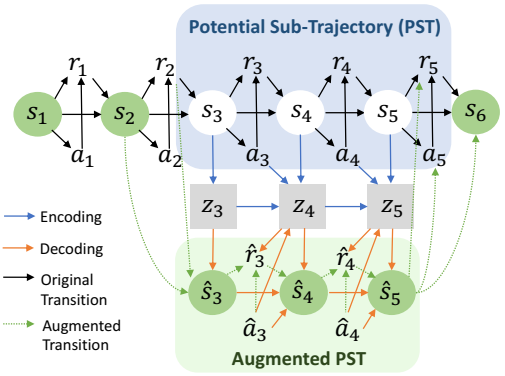

Figure 3. Illustration of VAE-MDP for Sub-Trajectory Augmentations. The PSTs are extracted by PSTs mining from original offline trajectories. Then VAE-MDP is employed to roll out new PSTs by reconstructing state-action space and inducing rewards. The augmented trajectory (colored in green) is formed by stitching back augmented PSTs to offline trajectories.

The training objective for VAE-MDP is to maximize the evidence lower bound (ELBO), which consists of the log-likelihood of reconstructing the states and rewards, and regularization of the approximated posterior, *i.e.*,

$$\mathcal{L} = -ELBO(\omega, \eta) = -\mathbb{E}_{q_\omega}\Big[\sum_{t=\zeta}^{\zeta+W-1} \log p_\eta(s_t|z_t) + \sum_{t=\zeta+1}^{\zeta+W-1} \log p_\eta(r_{t-1}|z_t) - KL\big(q_\omega(z_\zeta|s_\zeta)||p(z_\zeta)\big) - \sum_{t=\zeta+1}^{\zeta+W-1} KL\big(q_\omega(z_t|z_{t-1}, a_{t-1}, s_t)||p_\eta(z_t|z_{t-1}, a_{t-1})\big)\Big]. \tag{1}$$

The proof of Equation 1 are provided in Appendix A.3. Consequently, given a set of PSTs, $\mathcal{T}^g$, a VAE-MDP to the set can be trained to reconstruct a set of new PST samples, denoted as $\widehat{\mathcal{T}^g} = \{\hat{\varphi}^v_{\zeta,\zeta+W-1}\}$, $v \in [1, V]$, where $\hat{\varphi}^v_{\zeta,\zeta+W-1} = [(\hat{s}^v_\zeta, \hat{a}^v_\zeta, \hat{r}^v_\zeta, \hat{s}'^v_\zeta), ..., (\hat{s}^v_{\zeta+W-1}, \hat{a}^v_{\zeta+W-1}, \hat{r}^v_{\zeta+W-1}, \hat{s}'^v_{\zeta+W-1})]$ is a augmented PST and V is the total number of augmented PST samples, generated from VAE-MDP, for the set $\mathcal{T}^g$.

2.3. Stitching Augmented PSTs back to their Origins

With new augmented sub-trajectories rolled out by the VAE-MDP, we stitch them back to the original historical trajectories $\mathcal{D}$ for the OPE methods to leverage. This stitching process is designed to (i) provide enhanced coverage over the state-action visitation space where the corresponding PSTs do not explicitly capture homogeneous behaviors, and still (ii) maintain the part of the state-action visitation distribution associated with non-PSTs, since those may indicate object-specific information that is not shared across all trajectories, *e.g.*, the part of the surgical procedure specific to each patient, following from the surgery analogy above. Below we introduce how to stitch $\widehat{\mathcal{T}^g}$ with the original trajectories from $\mathcal{D}$. A graphical illustration of this step can be found in Figure 2.

Given a trajectory $\tau^{(i)} \in \mathcal{D}$, the G sets of PSTs $\{\mathcal{T}^1, ..., \mathcal{T}^G\}$ mined from $\mathcal{D}$ following Section 2.1, and G sets of augmented sub-trajectories $\{\widehat{\mathcal{T}^1}, ..., \widehat{\mathcal{T}^G}\}$ generated from G corresponding VAE-MDPs following Section 2.2, an augmented trajectory $\hat{\tau}^{(i)}$ corresponding to $\tau^{(i)}$ can be obtained by $\hat{\tau}^{(i)} = \Big[\mathbb{1}(t \in [\zeta, \zeta + W - 1])(\hat{s}^v_t, \hat{a}^v_t, \hat{r}^v_t, \hat{s}'^v_t) \vee \mathbb{1}(t \notin [\zeta, \zeta + W - 1])(s^{(i)}_t, a^{(i)}_t, r^{(i)}_t, s'^{(i)}_t)\Big]_{t=1}^{T}$; $(\hat{s}^v_t, \hat{a}^v_t, \hat{r}^v_t, \hat{s}'^v_t) \in \hat{\varphi}^v_{\zeta,\zeta+W-1} \in \widehat{\mathcal{T}^g}$, $\forall v \in [1, V]$ and $g \in [1, G]$.

3. Experiments

In this section, we first introduce augmentation baselines and OPE methods used for experiments, and environments. Then, results presented are discussed.

3.1. Setup

Baselines. We investigate a variety of general augmentation methods from prior work as baselines, including (i) RL-augmentation methods: **TDA** (Park et al., 2022) which originally incorporates with rewards learning by randomly extracting sub-trajectories from trajectories, we replace PST mining by TDA in OAT so that TDA can be used for OPE with augmentation; **permutation**, Gaussian **jittering**, and **scaling** have been broadly employed for image inputs (Laskin et al., 2020a; Liu et al., 2020; Raileanu et al., 2021); (ii) generative methods: **TimeGAN** (Yoon et al., 2019) and **VAE** (Barak et al., 2022) which are proposed towards time series; (iii) time-series augmentation methods: **SPAWNER** (Kamycki et al., 2019) and **DGW** (Iwana & Uchida, 2021b) that consider time-series similarities. We implement RL-augmentation methods strictly following original algorithms, and use open-sourced code provided by the authors for the generative and time-series augmentation methods. Since generative and time-series augmentation methods are not proposed towards trajectories, we treat trajectories as multivariate time series as their input.

Ablations. One ablation of our approach is to apply VAE-MDP to reconstruct *entire* trajectories as augmentations, *i.e.*, without PST mining (Section 2.1) and stitching (Section 2.3). Moreover, TDA (Park et al., 2022) and VAE (Barak et al., 2022) can be considered as two ablations as well, since TDA can isolate our PST mining from OAT and VAE augments entire trajectories following the vanilla VAE (Kingma & Welling, 2013), *i.e.*, without being adapted to the Markovian setting.

OPE methods considered. Outputs from all augmentation methods are fed into five OPE methods to compare the per-

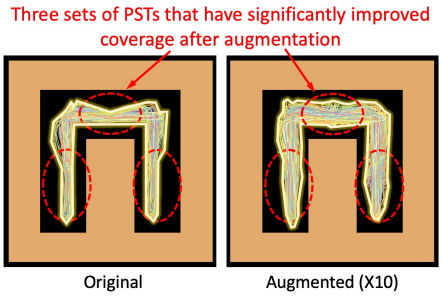

Figure 4. Visualization of trajectories in Maze2D-umaze. Left: the original 250 trajectories; Right: augmented data with ten times numbers of trajectories ($\times 10$).

formance achieved with versus without augmentations. The OPE methods we consider include importance sampling (**IS**) (Precup, 2000), fitted Q-evaluation (**FQE**) (Le et al., 2019), distribution correction estimation (**DICE**) (Yang et al., 2020a), doubly robust (**DR**) (Thomas & Brunskill, 2016), and model-based (**MB**) (Zhang et al., 2020a). We use the open-sourced implementations provided by the deep OPE (DOPE) benchmark (Fu et al., 2021).

Standard validation metrics. To validate OPE's performance (for both with and without augmentations), we use standard OPE metrics as introduced in the DOPE benchmark, which include absolute error, Spearman's rank correlation coefficient (Spearman, 1987), regret@1, and regret@5. Definitions of the metrics are described in Appendix B.3.

3.2. Environments

To evaluate our method, OAT, as well as the existing augmentation approaches for OPE, we use both simulated and real-world environments, spanning the domains of robotics, healthcare, and e-learning. The environments are human-involved which is generally challenging with highly limited quantity of human demonstrations containing underrepresented state space, due to homogeneous human interventions when collecting the historical trajectories.

Adroit. Adriot (Rajeswaran et al., 2018) is a simulation environment with four synthetic real-world robotics tasks, where a simulated Shadow Hand robot is asked to hammer a nail (`hammer`), open a door (`door`), twirl a pen (`pen`), or pick up and move a ball (`relocate`). Each task contains three training datasets with different levels of human-involvements, including full demonstration data from human (`human`), induced data from a fine-tuned RL policy (`expert`), and mixing data with a 50-50 ratio of demonstration and induced data (`cloned`).

Real-World Sepsis Treatment. We investigate a challenging task in healthcare, sepsis treatments, which has raised broad attention in OPE (Namkoong et al., 2020; Nie et al., 2022). Specifically, the trajectories are taken from Electronic Health Records containing 221,700 patient visits collected from a hospital over two years. The state space is constituted by 15 continuous sepsis-related clinical attributes that represent patients' health status, including heart rate, temperature, and creatinine etc. The cardinality of the action space is 4, *i.e.*, two binary treatment options over {`antibiotic_administration`, `oxygen_assistance`}. Given the four stages of sepsis defined by the clinicians (Delano & Ward, 2016), the rewards are set for each stage: infection (± 5), inflammation (± 10), organ failure (± 20), and septic shock (± 50). Negative rewards are given when a patient enters a worse stage, and positive rewards are given when the patient recovers to a better stage. The environment considers discrete time steps, with the horizon being 1160 steps. We use the earlier 80% trajectories (sorted by time of the first visit in patients' records) as training set and the later 20% as test set, following the common practice while splitting up time-series for training and testing (Campos et al., 2014). We assume that the clinical care team is well-trained with sufficient medical knowledge and follows standard protocols in sepsis treatments, thus we consider the behavioral policy, parameterized through behavior cloning (Azizsoltani & Jin, 2019), that generates the trajectories above as an expert policy. Five evaluation (target) policies are obtained by training 5 Deep Q Networks (DQNs) (Mnih et al., 2015) respectively over different hyper-parameters. More details are provided in Appendix D.

Real-World Intelligent Tutor Another important human-involved task for OPE is intelligent tutoring, where students interact with intelligent tutors, with the goal of improving students' engagements and learning outcomes. Such topics have been investigated in prior OPE works (Mandel et al., 2014; Nie et al., 2022). Specifically, we collect trajectories recorded from 1,307 students' interaction logs with an intelligent tutor, over seven semesters of an undergraduate course at an university. Since students' underlying learning states are considered unobservable (Mandel et al., 2014), we consult with domain experts who help defines the state space which is constituted by 142 attributes that could possibly capture students' learning status from their logs, *e.g.*, time elapsed since the start of the current working problem. During the tutoring, each student is required to solve twelve problems which cover various topics taught in course, thus the horizon of the environment is considered as 12 discrete steps. The cardinality of the action space is 3, *i.e.,* on each problem, the tutor need to decide whether the student should `solve` the next problem by themselves, `study` a solution provided by the tutor, or `work together` with the tutor to solve on the problem. Sparse rewards are obtained at the end of the tutoring, which is defined as students' normalized learning gain before and after tutoring (Chi et al., 2011). We use the trajectories collected from first six semesters as the training set, where the behavior policy follows an expert

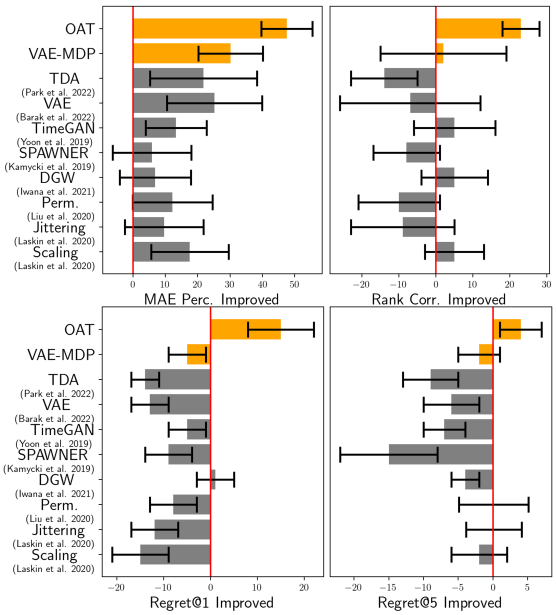

Figure 5. OPE improvement results averaging across **five OPE methods** and **four tasks** in Adroit `human` using each augmentation method. Top-left: Mean absolute error (MAE) percentage improved. Top-right: rank correlation improvements. Bottom-left & bottom-right: Regret@1 and @5 improvements, respectively.

policy commonly used in e-learning (Zhou et al., 2019), while the trajectories from the last semester constitute the test set. Four evaluation (target) policies, including three obtained by training 3 DQNs over different hyper-parameters respectively, in addition to one expert policy. More details are provided in Appendix E.

3.3. Results

Now we present and discuss the results obtained from the experimental setting introduced above.

3.3.1. The need of PSTs Mining

To better understand the need of PSTs mining (Section 2.1) conceptually, we visualize the set of augmented trajectories produced by our method, against the original set of historical trajectories $\mathcal{D}$, over the `Maze2D-umaze` environment which is a toy navigation task requiring an agent to reach a fixed goal location (Fu et al., 2020). We uniformly downsample a limited number (i.e., 250) of trajectories from the original dataset provided by D4RL (overall 12k trajectories), and use our method to augment this subset such that the total number of trajectories becomes ten times ($\times 10$) larger. The visualization is shown in Figure 4. It can be observed that there exist 3 sets of PSTs (as circled in the figure) that have significantly increased state space coverage after augmentation, benefiting from the PSTs mining methodology introduced in Section 2.1.

3.3.2. Results over Adroit

Figure 5 presents the averaged improvements across five OPE methods, over all four tasks (*i.e.*, `hammer`, `door`, `pen`, `relocate`) in Adroit `human` datasets, quantified by the percentage increases over the four validation metrics achieved by the OPE methods evaluated over the augmented against the original datasets. Overall, our method significantly improves OPE methods in terms of all standard validation metrics, and achieves the best performance compared to all augmentation baselines. This illustrates the effectiveness and robustness of our proposed methods across environments and tasks. There is no clear winner among baselines, where VAE, TimeGAN, and scaling in general perform better in terms of MAE, DGW and scaling performs better in terms of rank correlation, permutation and jittering perform better in terms of regrest@5. More specifically, besides the fact that all methods can in general improve MAE, most baselines lead to negative effects in terms of the other three metrics.

More importantly, it can be observed that the ablation baseline VAE-MDP is significantly outperformed by OAT across all metrics, which further justifies the importance of augmenting over the PSTs instead of the entire horizon. It can be also observed that VAE-MDP in general outperforms VAE Augmenter which uses the vanilla VAE instead without adaptation to the Markovian setting, illustrating the importance of the adaptation step introduced in Section 2.2. We also find that generative models achieve the best performance among the baselines over environments that have relatively shorter horizons (*e.g.*, pen), while their performance is diminished when horizons increased. That is aligned with findings in prior work, and further support our design of PSTs mining that provides much shorter and representative trajectories for generative learning.

3.3.3. Results over Real-World Healthcare and E-Learning

Figure 6 shows the intelligent tutor GUI and empirical returns of the four evaluation policies being considered. Figure 7 presents the average MAE improvements across all OPE methods in e-learning (left), and improved rank correlation in healthcare (right). Complete results for all validation metrics are provided in Appendix E. Regret@5 is not applicable to both environments, since the total number of evaluation policies are less than or equal to five.

Overall, our method can significantly improve OPE performance in terms of MAE, rank correlation, and regret@1 in both real-world human-involved environments. In both e-learning and healthcare, most augmentation baselines lead to neutral to negative percentage improvements over the metrics considered, while OAT significantly improved OPE's performance over all baselines, with the ablation VAE-MDP

Figure 6. Our intelligent tutor GUI (left) and empirical results with three RL-induced policies and one expert policy (right).

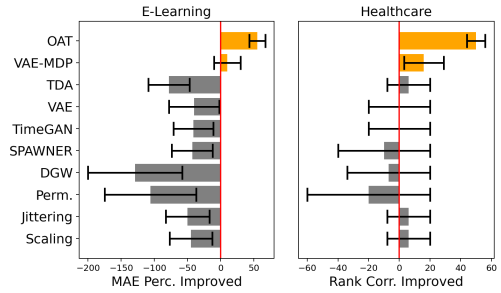

Figure 7. OPE improvement results averaging across five OPE methods in e-learning (left) and healthcare (right).

attaining the 2nd best performance. A possible reason for baselines perform worse in real-world environments than in simulations is that real-world human-involved systems are considered sophisticated, as the human mental states that may impact their behaviors implicitly. This further indicates the importance of extracting underlying information from historical trajectories $\mathcal{D}$, as did in OAT and VAE-MDP, as well as effectively enriching the state-action visitation space to provide more comprehensive coverage for OPE methods to leverage, powered by the methodologies introduced in Section 2.

3.3.4. More Discussions

To further understand the effectiveness of OAT, we discuss the following two questions that are commonly involved in analyses over human-involved systems, *i.e.*, human-involved levels and statistical significance.

Would the level of human involvements affect trajectory augmentations for OPE? As presented in Figure 8, we evaluate augmentation methods across the four tasks in Adroit environment with three different levels of human involvements (LoHI) sorted from the most to least, *i.e.,* `human`, `cloned`, and `expert`. The results show that our method achieves the best performance in terms of all validation metrics when humans are involved in data collection (i.e., `human`, `cloned`). The performance of our method is slightly attenuated (but still effective) when the LoHI decreased, while our ablation VAE-MDP leads MAE when the LoHI is 0% (i.e., `expert`). Though TDA is effective under the case when the LoHI is 0%, it still performs worse than OAT and consistently worse at other levels. Such a finding further confirms the effectiveness of PST mining. Moreover, most baselines are ineffective when the LoHI is below 50%. A possible reason is that the trajectories

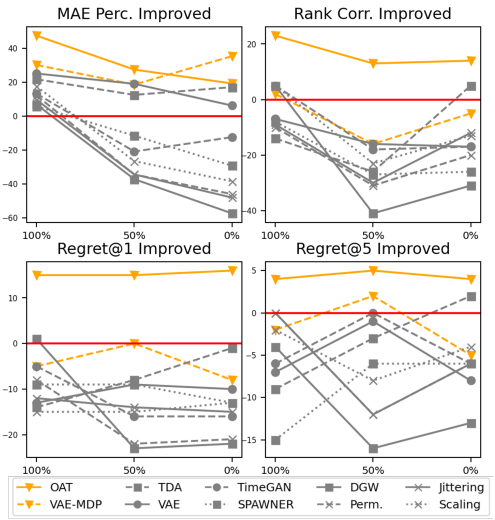

Figure 8. OPE improvement results with three human-involving levels, i.e., 100% (`human`), 50% (`cloned`), 0% (`expert`), averaging across five OPE methods and four tasks in Adroit.

obtained from human demonstrations often provide limited and/or biased coverage over the state-action space, thus any augmentation methods that can potentially increase the coverage might be able to improve OPE's performance. In contrast, the historical trajectories induced from simulations tend to result in better coverage over the state-action space in general, and the augmentation methods that do not consider the Markovian setting may generate trajectories that could be less meaningful to the OPE methods, making them less effective.

Can trajectory augmentation facilitate OPE in terms of significance test? OPE validation metrics generally focus on standard error metrics as proposed in (Fu et al., 2021; Voloshin et al., 2021b), while domain experts may emphasis statistical significance test for real-world human-involved tasks (Robertson & Kaptein, 2016; Zhou et al., 2022). For example, rank correlation summarizes the performance of a set of policies' relative rankings using averaged returns; in contrast, statistical significance tests can help determine if the relationships being found are due to randomness. Moreover, they can be easier conveyed to and interpreted by domain experts (Guilford, 1950; Ju et al., 2019).

One key measurement for RL-induced policies is whether they significantly outperform the expert policy in human-involved systems (Zhou et al., 2019; 2020). We conduct a t-test over OPE estimations (with and without augmentations) obtained from bootstrapping as introduced in (Hao et al., 2021), and measure whether there is a significant difference between the mean value of OPE estimation for each RL-induced policy against the expert policy. Interesting, the results show that IS performs the best among all 5 OPE methods we considered, in terms of all standard validation metrics in our e-learning experiments, with and without aug-

IS result	π_1 t_p	π_2 t_p	π_3 t_p
No Aug.	**7.24**$_{.00}$	**7.07**$_{.00}$	**-4.48**$_{.00}$
OAT	**3.10**$_{.00}$	1.33$_{.19}$	2.11$_{.06}$
TDA	**10.14**$_{.00}$	**5.82**$_{.00}$	**-13.58**$_{.00}$
VAE	-1.94$_{.06}$	-1.92$_{.06}$	.54$_{.13}$
TimeGAN	1.90$_{.06}$	**-2.25**$_{.03}$	**-2.25**$_{.03}$
SPAWNER	-1.00$_{.32}$	-1.00$_{.32}$	1.00$_{.32}$
DGW	-1.43$_{.06}$	-1.43$_{.16}$	1.43$_{.16}$
Perm.	-1.78$_{.08}$	-1.77$_{.08}$	1.77$_{.08}$
Jittering	-1.89$_{.06}$	-1.89$_{.06}$	1.90$_{.06}$
Scaling	-1.33$_{.06}$	-1.33$_{.06}$	1.33$_{.06}$
Empirical result	**2.01**$_{.04}$	0.61$_{.54}$	0.20$_{.84}$

Table 1. Statistical significance test at the level of $\rho < 0.05$ with bootstrapping on three RL-induced policy π_1, π_2, π_3 compared to expert policy π_{expert} from real-world intelligent tutoring. The results that show significance are in bold.

mentations using each augmentation method. We conjecture that this may be due to the fact that the behavioral policies are intrinsically similar, as shown in Figure 6 (right) that 3 out of the 4 policies (*i.e.*, π_2, π_3, π_{expert}) lead to rather similar returns, and the unbiased nature of IS estimators may dominate it's high variance downside. The statistical significance results are summarized in Table 1. It can be observed that, without augmentation, IS estimates that all RL-induced policies performs significantly different from the expert policy. However, in empirical study, only π_1 performs significantly better than expert policy, while the other two, i.e., π_2 and π_3 not. And our proposed method is the only one that improve the IS estimation to be aligned with empirical results across all three policies, while the baselines improve estimation at most one policy. Therefore, the results indicate the effectiveness of our proposed method in terms of both standard OPE validation metrics and human-centric statistical significance test.

4. Related Works

OPE A variety of contemporary OPE methods has been proposed, which can be mainly divided into three categories (Voloshin et al., 2021b): (i) Inverse propensity scoring (Precup, 2000; Doroudi et al., 2017), such as Importance Sampling (IS) (Doroudi et al., 2017). (ii) Direct methods that directly estimate the value functions of the evaluation policy (Nachum et al., 2019; Uehara et al., 2020; Xie et al., 2019; Zhang et al., 2021; Yang et al., 2022), including but not limited to model-based estimators (MB) (Paduraru, 2013; Zhang et al., 2021), value-based estimators (Munos et al., 2016; Le et al., 2019) such as Fitted Q Evaluation (FQE), and minimax estimators (Liu et al., 2018; Zhang et al., 2020b; Voloshin et al., 2021a) such as DualDICE (Yang et al., 2020a). (iii) Hybrid methods combine aspects of both inverse propensity scoring and direct methods (Jiang & Li, 2016; Thomas & Brunskill, 2016), such as DR (Jiang & Li, 2016). However, a major challenge of applying OPE to real-world is that many methods can perform unpleasant when human-collected data is highly limited as demonstrated in (Fu et al., 2020; Gao et al., 2023). Therefore, augmentation can be an important way to facilitate OPE performance.

Data Augmentation for RL In RL, data augmentation has been recognized as effective to improve generalizability of agents over various tasks (Laskin et al., 2020a;b; Kostrikov et al., 2020; Liu et al., 2021; Raileanu et al., 2021; Joo et al., 2022; Goyal et al., 2022). For instance, automatic augmentation selection frameworks are proposed for actor-critic algorithms by regularizing the policy and value functions (Raileanu et al., 2021). However, most of the prior work only consider image input which may not capture temporal dependencies in trajectories. More importantly, the prior work is proposed towards RL policy optimization by learning from high-reward regions of state-action space, while OPE aims to generalize over evaluation policies that can be heterogeneous and lead to varied performance. To the best of our knowledge, no prior work has extensively investigated various prior augmentation methods in OPE, nor proposed augmentation towards offline trajectories to scaffold OPE in real-world domains.

More comprehensive review of related works on OPE and data augmentations in general can be found in Appendix F.

5. Conclusion and Social Impact

We have proposed OAT which can capture the dynamics underlying human-involved environments from historical trajectories that provide limited coverage of the state-action space and induce effective augmented trajectories to facilitate OPE. This is achieved by mining potential sub-trajectories which have great potential to increase the historical trajectories' coverage over state-action space, as well as extending a generative modeling framework to capture dynamics under the potential sub-trajectories. We have validated OAT in both simulation and real-world human-involved environments, including robotic control, disease treatment, and intelligent tutoring, and the results have shown that OAT can generally improve OPE performance and outperform a variety of data augmentation methods.

All educational and healthcare data employed in this paper were obtained anonymously through an exempt IRB-approved protocol and were scored using established rubrics. No demographic data or class grades were collected. All data were shared within the research group under IRB, and were de-identified and automatically processed for labeling. This research seeks to remove societal harms that come from lower engagement and retention of students who need more personalized interventions and developing more robust medical interventions for patients.

References

Antoniou, A., Storkey, A., and Edwards, H. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

Azizsoltani, H. and Jin, Y. Unobserved is not equal to non-existent: Using gaussian processes to infer immediate rewards across contexts. In *In Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019.

Barak, S., Mirafzali, E., and Joshaghani, M. Improving deep learning forecast using variational autoencoders. *Available at SSRN*, 2022.

Bone, R. C., Balk, R. A., Cerra, F. B., Dellinger, R. P., Fein, A. M., Knaus, W. A., Schein, R. M., and Sibbald, W. J. Definitions for sepsis and organ failure and guidelines for the use of innovative therapies in sepsis. *Chest*, 101(6): 1644–1655, 1992.

Campos, P. G., Díez, F., and Cantador, I. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, 24:67–119, 2014.

Chi, M., VanLehn, K., Litman, D., and Jordan, P. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *User Modeling and User-Adapted Interaction*, 21(1):137–180, 2011.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 113–123, 2019.

Delano, M. J. and Ward, P. A. The immune system's role in sepsis progression, resolution, and long-term outcome. *Immunological reviews*, 274(1):330–353, 2016.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Donahue, C., McAuley, J., and Puckette, M. Adversarial audio synthesis. In *International Conference on Learning Representations*, 2018.

Doroudi, S., Thomas, P. S., and Brunskill, E. Importance sampling for fair policy selection. *Grantee Submission*, 2017.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Fu, J., Norouzi, M., Nachum, O., Tucker, G., Wang, Z., Novikov, A., Yang, M., Zhang, M. R., Chen, Y., Kumar, A., et al. Benchmarks for deep off-policy evaluation. *arXiv preprint arXiv:2103.16596*, 2021.

Gao, G., Gao, Q., Yang, X., Pajic, M., and Chi, M. A reinforcement learning-informed pattern mining framework for multivariate time series classification. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 2994–3000, 2022.

Gao, G., Ju, S., Ausin, M. S., and Chi, M. Hope: Human-centric off-policy evaluation for e-learning and healthcare. In *AAMAS*, 2023.

Goyal, A., Friesen, A., Banino, A., Weber, T., Ke, N. R., Badia, A. P., Guez, A., Mirza, M., Humphreys, P. C., Konyushova, K., et al. Retrieval-augmented reinforcement learning. In *International Conference on Machine Learning*, pp. 7740–7765. PMLR, 2022.

Guilford, J. P. Fundamental statistics in psychology and education. 1950.

Hallac, D., Vare, S., Boyd, S., and Leskovec, J. Toeplitz inverse covariance-based clustering of multivariate time series data. In *ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 215–223, 2017.

Hanna, J., Niekum, S., and Stone, P. Importance sampling policy evaluation with an estimated behavior policy. In *International Conference on Machine Learning*, pp. 2605–2613. PMLR, 2019.

Hao, B., Ji, X., Duan, Y., Lu, H., Szepesvári, C., and Wang, M. Bootstrapping statistical inference for off-policy evaluation. *arXiv preprint arXiv:2102.03607*, 2021.

Iwana, B. K. and Uchida, S. An empirical survey of data augmentation for time series classification with neural networks. *Plos one*, 16(7):e0254841, 2021a.

Iwana, B. K. and Uchida, S. Time series data augmentation for neural networks by time warping with a discriminative teacher. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 3558–3565. IEEE, 2021b.

Jiang, N. and Li, L. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pp. 652–661. PMLR, 2016.

Joo, H.-T., Baek, I.-C., and Kim, K.-J. A swapping target q-value technique for data augmentation in offline reinforcement learning. *IEEE Access*, 2022.

Ju, S., Shen, S., Azizsoltani, H., Barnes, T., and Chi, M. Importance sampling to identify empirically valid policies and their critical decisions. In *EDM (Workshops)*, pp. 69–78, 2019.

Kamycki, K., Kapuscinski, T., and Oszust, M. Data augmentation with suboptimal warping for time-series classification. *Sensors*, 20(1):98, 2019.

Kim, Y.-J. and Chi, M. Temporal belief memory: Imputing missing data during rnn training. In *In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI-2018)*, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Komorowski, M., Celi, L. A., Badawi, O., Gordon, A. C., and Faisal, A. A. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature medicine*, 24(11):1716–1720, 2018.

Kostrikov, I., Yarats, D., and Fergus, R. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., and Srinivas, A. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33: 19884–19895, 2020a.

Laskin, M., Srinivas, A., and Abbeel, P. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pp. 5639–5650. PMLR, 2020b.

Le, H., Voloshin, C., and Yue, Y. Batch policy learning under constraints. In *International Conference on Machine Learning*, pp. 3703–3712. PMLR, 2019.

Le Guennec, A., Malinowski, S., and Tavenard, R. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD workshop on advanced analytics and learning on temporal data*, 2016.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Lipton, Z. C., Kale, D., and Wetzel, R. Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. In *Machine learning for healthcare conference*, pp. 253–270. PMLR, 2016.

Liu, B., Zhang, Z., and Cui, R. Efficient time series augmentation methods. In *2020 13th international congress on image and signal processing, BioMedical engineering and informatics (CISP-BMEI)*, pp. 1004–1009. IEEE, 2020.

Liu, J., Hongyin, Z., and Wang, D. Dara: Dynamics-aware reward augmentation in offline reinforcement learning. In *International Conference on Learning Representations*, 2021.

Liu, Q., Li, L., Tang, Z., and Zhou, D. Breaking the curse of horizon: Infinite-horizon off-policy estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

Liu, V., Escobar, G. J., Greene, J. D., Soule, J., Whippy, A., Angus, D. C., and Iwashyna, T. J. Hospital deaths in patients with sepsis from 2 independent cohorts. *Jama*, 312(1):90–92, 2014.

Mandel, T., Liu, Y.-E., Levine, S., Brunskill, E., and Popovic, Z. Offline policy evaluation across representations with applications to educational games. In *AAMAS*, volume 1077, 2014.

Martin, G., Mannino, D., et al. The epidemiology of sepsis in the united states from 1979 through 2000. *New England Journal of Medicine*, 2003.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533, 2015.

Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. Safe and efficient off-policy reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

Nachum, O., Chow, Y., Dai, B., and Li, L. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in Neural Information Processing Systems*, 32, 2019.

Namkoong, H., Keramati, R., Yadlowsky, S., and Brunskill, E. Off-policy policy evaluation for sequential decisions under unobserved confounding. *Advances in Neural Information Processing Systems*, 33:18819–18831, 2020.

Nie, A., Flet-Berliac, Y., Jordan, D. R., Steenbergen, W., and Brunskill, E. Data-efficient pipeline for offline reinforcement learning with limited data. In *Advances in Neural Information Processing Systems*, 2022.

Paduraru, C. Off-policy evaluation in markov decision processes. 2013.

Park, J., Seo, Y., Shin, J., Lee, H., Abbeel, P., and Lee, K. Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. In *International Conference on Learning Representations*, 2022.

Precup, D. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, pp. 80, 2000.

Raghu, A., Komorowski, M., Ahmed, I., Celi, L., Szolovits, P., and Ghassemi, M. Deep reinforcement learning for sepsis treatment. *arXiv preprint arXiv:1711.09602*, 2017.

Raileanu, R., Goldstein, M., Yarats, D., Kostrikov, I., and Fergus, R. Automatic data augmentation for generalization in reinforcement learning. *Advances in Neural Information Processing Systems*, 34:5402–5415, 2021.

Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *Robotics: Science and Systems*, 2018.

Robertson, J. and Kaptein, M. *Modern statistical methods for HCI*. Springer, 2016.

Rue, H. and Held, L. *Gaussian Markov random fields: theory and applications*. CRC press, 2005.

Singer, M., Deutschman, C. S., Seymour, C. W., Shankar-Hari, M., Annane, D., Bauer, M., Bellomo, R., Bernard, G. R., Chiche, J.-D., Coopersmith, C. M., et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). *Jama*, 315(8):801–810, 2016.

Spearman, C. The proof and measurement of association between two things. *The American journal of psychology*, 100(3/4):441–471, 1987.

Thomas, P. and Brunskill, E. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pp. 2139–2148. PMLR, 2016.

Uehara, M., Huang, J., and Jiang, N. Minimax weight and q-function learning for off-policy evaluation. In *International Conference on Machine Learning*, pp. 9659–9668. PMLR, 2020.

Voloshin, C., Jiang, N., and Yue, Y. Minimax model learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1612–1620. PMLR, 2021a.

Voloshin, C., Le, H. M., Jiang, N., and Yue, Y. Empirical study of off-policy policy evaluation for reinforcement learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021b.

Wang, L., Tang, R., He, X., and He, X. Hierarchical imitation learning via subgoal representation learning for dynamic treatment recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 1081–1089, 2022.

Xie, Q., Dai, Z., Hovy, E., Luong, T., and Le, Q. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33: 6256–6268, 2020.

Xie, T., Ma, Y., and Wang, Y.-X. Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. *Advances in Neural Information Processing Systems*, 32, 2019.

Yacoby, Y., Pan, W., and Doshi-Velez, F. Failure modes of variational autoencoders and their effects on downstream tasks. *arXiv preprint arXiv:2007.07124*, 2020.

Yang, M., Nachum, O., Dai, B., Li, L., and Schuurmans, D. Off-policy evaluation via the regularized lagrangian. *Advances in Neural Information Processing Systems*, 33: 6551–6561, 2020a.

Yang, M., Dai, B., Nachum, O., Tucker, G., and Schuurmans, D. Offline policy selection under uncertainty. In *International Conference on Artificial Intelligence and Statistics*, pp. 4376–4396. PMLR, 2022.

Yang, X., Zhou, G., Taub, M., Azevedo, R., and Chi, M. Student subtyping via em-inverse reinforcement learning. *International Educational Data Mining Society*, 2020b.

Yang, X., Zhang, Y., and Chi, M. Multi-series time-aware sequence partitioning for disease progression modeling. In *IJCAI*, 2021.

Yoon, J., Jarrett, D., and Van der Schaar, M. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.

Zhang, M. R., Paine, T., Nachum, O., Paduraru, C., Tucker, G., Norouzi, M., et al. Autoregressive dynamics models for offline policy evaluation and optimization. In *International Conference on Learning Representations*, 2020a.

Zhang, M. R., Paine, T. L., Nachum, O., Paduraru, C., Tucker, G., Wang, Z., and Norouzi, M. Autoregressive dynamics models for offline policy evaluation and optimization. *arXiv preprint arXiv:2104.13877*, 2021.

Zhang, S., Liu, B., and Whiteson, S. Gradientdice: Rethinking generalized offline estimation of stationary values. In *International Conference on Machine Learning*, pp. 11194–11203. PMLR, 2020b.

Zhou, G., Azizsoltani, H., Ausin, M. S., Barnes, T., and Chi, M. Hierarchical reinforcement learning for pedagogical policy induction. In *International conference on artificial intelligence in education*, pp. 544–556. Springer, 2019.

Zhou, G., Yang, X., Azizsoltani, H., Barnes, T., and Chi, M. Improving student-system interaction through data-driven explanations of hierarchical reinforcement learning induced pedagogical policies. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, pp. 284–292, 2020.

Zhou, G., Azizsoltani, H., Ausin, M. S., Barnes, T., and Chi, M. Leveraging granularity: Hierarchical reinforcement learning for pedagogical policy induction. *International journal of artificial intelligence in education*, 32(2):454–500, 2022.

A. More Details on Methodology

A.1. Toeplitz Inverse Covariance-Based Clustering (TICC) Problem

Each cluster $c \in [1, C]$ is defined as a Markov random field (Rue & Held, 2005), or correlation network, captured by its Gaussian inverse covariance matrix $\Sigma_c^{-1} \in \mathbb{R}^{m \times m}$, where m is the dimension of state space. We also define the set of clusters $\mathbf{M} = \{M_1, \ldots, M_C\} \subset \mathbb{R}$ as well as the set of inverse covariance matrices $\mathbf{\Sigma}^{-1} = \{\Sigma_1^{-1}, \ldots, \Sigma_C^{-1}\}$. Then the objective is set to be

$$\max_{\mathbf{\Sigma}^{-1}, \mathbf{M}} \sum_{c=1}^{C} \Big[\sum_{o_t^{(i)} \in M_c} \big(\mathcal{L}(o_t^{(i)}; \Sigma_c^{-1}) - \epsilon \mathbb{1}\{o_{t-1}^{(i)} \notin M_c\} \big) \Big], \tag{2}$$

where the first term defines the log-likelihood of $o_t^{(i)}$ coming from M_c as $\mathcal{L}(o_t^{(i)}; \Sigma_c^{-1}) = -\frac{1}{2}(o_t^{(i)} - \mu_c k)^T \Sigma_c^{-1} (o_t^{(i)} - \mu_c) + \frac{1}{2} \log\det \Sigma_c^{-1} - \frac{n}{2} \log(2\pi)$ with μ_c being the empirical mean of cluster M_c, the second term $\mathbb{1}\{o_{t-1}^{(i)} \notin M_c\}$ penalizes the adjacent events that are not assigned to the same cluster and ϵ is a constant balancing off the scale of the two terms. This optimization problem can be solved using the expectation-maximization family of algorithms by updating $\mathbf{\Sigma}^{-1}$ and $\mathbf{M}$ alternatively (Hallac et al., 2017). There are variations of TICC targeting specific characteristics of data. Specifically, we used MT-TICC (Yang et al., 2021) which is proposed towards time-awareness and multi-trajectories.

A.2. Detailed Formulation of the VAE-MDP

The latent prior $p(z_\zeta) \sim \mathcal{N}(0, I)$ representing the distribution of the initial latent states (at the beginning of each PST in the set $\mathcal{T}^g$), where I is the identity covariance matrix.

The encoder $q_\omega(z_t | s_{t-1}, a_{t-1}, s_t)$ is used to approximate the posterior distribution $p_\eta(z_t | s_{t-1}, a_{t-1}, s_t) = \frac{p_\eta(z_{t-1}, a_{t-1}, z_t, s_t)}{\int_{z_t \in \mathcal{Z}} p(z_{t-1}, a_{t-1}, z_t, s_t) dz_t}$, where $\mathcal{Z} \subset \mathbb{R}^m$ and m is the dimension. Given that $q_\omega(z_{\zeta:\zeta+W-1} | s_{\zeta:\zeta+W-1}, a_{\zeta:\zeta+W-2}) = q_\omega(z_\zeta | s_\zeta) \prod_{t=\zeta+1}^{\zeta+W-1} q_\omega(z_t | z_{t-1}, a_{t-1}, s_t)$, both distributions $q_\omega(z_\zeta | s_\zeta)$ and $q_\omega(z_t | z_{t-1}, a_{t-1}, s_t)$ follow diagonal Gaussian, where mean and diagonal covariance are determined by multi-layer perceptrons (MLPs) and long short-term memory (LSTM), with neural network weights ω. Thus, one can infer $z_\zeta^\omega \sim q_\omega(z_\zeta | s_\zeta)$, $z_t^\omega \sim q_\omega(z_t | h_t^\omega)$, with $h_t^\omega = f_\omega(h_{t-1}^\omega, z_{t-1}^\omega, a_{t-1}, s_t)$ where f_ω represents LSTM layer and h_t^ω represents LSTM recurrent hidden state.

The decoder $p_\eta(z_t, s_t, r_{t-1} | z_{t-1}, a_{t-1})$ is used to sample new trajectories. Given $p_\eta(z_{\zeta+1:\zeta+W-1}, s_{\zeta:\zeta+W-1}, r_{\zeta:\zeta+W-2} | z_\zeta, \beta) = \prod_{t=\zeta}^{\zeta+W-1} p_\eta(s_t | z_t) \prod_{t=\zeta+1}^{T} p_\eta(z_t | z_{t-1}, a_{t-1}) p_\eta(r_{t-1} | z_t)$, where a_t's are determined following the behavioral policy β, distributions $p_\eta(s_t | z_t)$ and $p_\eta(r_{t-1} | z_t)$ follow diagonal Gaussian with mean and covariance determined by MLPs and $p_\eta(z_t | z_{t-1}, a_{t-1})$ follows diagonal Gaussian with mean and covariance determined by LSTM.

Thus, the generative process can be formulated as, *i.e.*, at initialization, $z_\zeta^\eta \sim p(z_\zeta)$, $s_\zeta^\eta \sim p_\eta(s_\zeta | z_\zeta^\eta)$, $a_\zeta \sim \beta(a_\zeta | s_\zeta^\eta)$; followed by $z_t^\eta \sim p_\eta(\tilde{h}_t^\eta)$, $r_{t-1}^\eta \sim p_\eta(r_{t-1} | z_t^\eta)$, $s_t^\eta \sim p_\eta(s_t | z_t^\eta)$, $a_t \sim \beta(a_t | s_t^\eta)$, with $\tilde{h}_t^\eta = g_\eta[f_\eta(h_{t-1}^\eta, z_{t-1}^\eta, a_{t-1})]$ where g_η represents an MLP.

A.3. Proof of Equation 1

The derivation of the evidence lower bound (ELBO) for the joint log-likelihood distribution can be found below.

$$\log p_\eta(s_{\zeta:\zeta+W-1}, r_{\zeta:\zeta+W-2}) \tag{3}$$

$$= \log \int_{z_{\zeta+1:\zeta+W-1} \in \mathcal{Z}} p_\eta(s_{\zeta:\zeta+W-1}, z_{\zeta+1:\zeta+W-1}, r_{\zeta:\zeta+W-2}) dz \tag{4}$$

$$= \log \int_{z_{\zeta+1:\zeta+W-1} \in \mathcal{Z}} \frac{p_\eta(s_{\zeta:\zeta+W-1}, z_{\zeta+1:\zeta+W-1}, r_{\zeta:\zeta+W-2})}{q_\omega(z_{\zeta:\zeta+W-1}|s_{\zeta:\zeta+W-1}, a_{\zeta:\zeta+W-2})} q_\omega(z_{\zeta:\zeta+W-1}|s_{\zeta:\zeta+W-1}, a_{\zeta:\zeta+W-2}) dz \tag{5}$$

$$\overset{Jensen's\ inequality}{\geq} \mathbb{E}_{q_\omega}[\log p(z_\zeta) + \log p_\eta(s_{\zeta:\zeta+W-1}, z_{\zeta+1:\zeta+W-1}, r_{\zeta:\zeta+W-2}|z_\zeta) - \log q_\omega(z_{\zeta:\zeta+W-1}|s_{\zeta:\zeta+W-1}, a_{\zeta:\zeta+W-2})] \tag{6}$$

$$= \mathbb{E}_{q_\omega}\Big[\log p(z_\zeta) + \log p_\eta(s_\zeta|z_\zeta) + \sum\nolimits_{t=\zeta}^{\zeta+W-1} \log p_\eta(s_t, z_t, r_{t-1}|z_{t-1}, a_{t-1}) - \log q_\omega(z_\zeta|s_\zeta) - \sum\nolimits_{t=\zeta+1}^{\zeta+W-1} \log q_\omega(z_t|z_{t-1}, a_{t-1}, s_t)\Big] \tag{7}$$

$$= \mathbb{E}_{q_\omega}\Big[\log p(z_\zeta) - \log q_\omega(z_\zeta|s_\zeta) + \log p_\eta(s_\zeta|z_\zeta) + \sum\nolimits_{t=\zeta+1}^{\zeta+W-1} \log \big(p_\eta(s_t|z_t) p_\eta(r_{t-1}|z_t) p_\eta(z_t|z_{t-1}, a_{t-1})\big) - \sum\nolimits_{t=\zeta+1}^{\zeta+W-1} \log q_\omega(z_t|z_{t-1}, a_{t-1}, s_t)\Big] \tag{8}$$

$$= \mathbb{E}_{q_\omega}\Big[\sum\nolimits_{t=\zeta}^{\zeta+W-1} \log p_\eta(s_t|z_t) + \sum\nolimits_{t=\zeta+1}^{\zeta+W-1} \log p_\eta(r_{t-1}|z_t) - KL\big(q_\omega(z_\zeta|s_\zeta)||p(z_\zeta)\big) - \sum\nolimits_{t=\zeta+1}^{\zeta+W-1} KL\big(q_\omega(z_t|z_{t-1}, a_{t-1}, s_t)||p_\eta(z_t|z_{t-1}, a_{t-1})\big)\Big]. \tag{9}$$

B. Experimental Setup

B.1. Training Resources

We implement the proposed method in Python. Training of our method and baselines are supported by four NVIDIA TITAN Xp 12GB, three NVIDIA Quadro RTX 6000 24GB, and four NVIDIA RTX A5000 24GB GPUs.

B.2. Implementation Details & Hyper-Parameters

The cluster number for discrete representation mapping can be determined by silhouette score using training data following (Hallac et al., 2017), we perform search among $[10, 20]$ for C in all datasets and the one with the highest silhouette score is selected. In our experiments, $C = 18, 10, 19, 16$ for {`pen, door, relocate, hammer`}`-human`, respectively; $C = 20, 10, 10, 10$ for {`pen, door, relocate, hammer`}`-cloned`, respectively; $C = 11, 10, 10, 10$ for {`pen, door, relocate, hammer`}`-expert`, respectively; $C = 14, 17$ for e-learning and healthcare, respectively. The experimental results are obtained with selecting the PSTs using the threshold at the top 1, *i.e.,* we use the PST with the highest support of its corresponding TDSS, for easier investigation of the PSTs mining and comparison to other augmentation baselines such as TDA, and present straightforward and general effects of our method. The percentage supports of the selected PSTs, *i.e.,* $support(\cdot)/N$, are all $\geq 82\%$ across all datasets and all experimental environments, especially can cover 100% trajectories in all Adroit `human` tasks, which may further indicates the effectiveness of PSTs mining. We choose the neural network architectures as follows. For the components involving LSTMs, which include $q_\omega(z_t|z_{t-1}, a_{t-1}, s_t)$ and $p_\eta(z_t|z_{t-1}, a_{t-1})$, their architecture include one LSTM layer with 64 nodes, followed by a dense layer with 64 nodes. All other components do not have LSTM layers involved, so they are constituted by a neural network with 2 dense layers, with 128 and 64 nodes respectively. The output layers that determine the mean and diagonal covariance of diagonal Gaussian distributions use linear and softplus activations, respectively. The ones that determine the mean of Bernoulli distributions (*e.g.*, for capturing early termination of episodes) are configured to use sigmoid activations. For training OAT and its ablation VAE-MDP, maximum number of iteration is set to 100 and minibatch size set to 4 (given the small numbers of trajectories, *i.e.,* 25 for each task) in Adroit, and 1,000 and 64 for real-world healthcare and e-learning, respectively. Adam optimizer is used to perform gradient descent. To determine the learning rate, we perform grid search among $\{1e-4, 3e-3, 3e-4, 5e-4, 7e-4\}$. Exponential decay is applied to the learning rate, which decays the learning rate by

0.997 every iteration. For OPE, the model-based methods are evaluated by directly interacting with each target policy for 50 episodes, and the mean of discounted total returns ($\gamma = 0.995$ for Adroit, $\gamma = 0.99$ for Healthcare, $\gamma = 0.9$ for e-learning) over all episodes is used as estimated performance for the policy.

B.3. Evaluation Metrics

Absolute error The absolute error is defined as the difference between the actual value and estimated value of a policy:

$$AE = |V^\pi - \hat{V}^\pi| \tag{10}$$

where V^π represents the actual value of the policy π, and $\hat{V}^\pi$ represents the estimated value of π.

Regret@1 Regret@1 is the (normalized) difference between the value of the actual best policy, and the actual value of the best policy chosen by estimated values. It can be defined as:

$$R1 = (\max_{i \in 1:P} V_i^\pi - \max_{j \in \text{best}(1:P)} V_j^\pi) / \max_{i \in 1:P} V_i^\pi \tag{11}$$

where $\text{best}(1:P)$ denotes the index of the best policy over the set of P policies as measured by estimated values $\hat{V}^\pi$.

Rank correlation Rank correlation measures the Spearman's rank correlation coefficient between the ordinal rankings of the estimated values and actual values across policies:

$$\rho = \frac{Cov(\text{rank}(V_{1:P}^\pi), \text{rank}(\hat{V}_{1:P}^\pi))}{\sigma(\text{rank}(V_{1:P}^\pi))\sigma(\text{rank}(\hat{V}_{1:P}^\pi))} \tag{12}$$

where $\text{rank}(V_{1:P}^\pi)$ denotes the ordinal rankings of the actual values across policies, and $\text{rank}(\hat{V}_{1:P}^\pi)$ denotes the ordinal rankings of the estimated values across policies.

C. Adroit

C.1. Detailed Results

	Pen				Relocate			
	MAE	Rank Corr.	Regret@1	Regret@5	MAE	Rank Corr.	Regret@1	Regret@5
NoAug.	3014	-0.104	0.184	0.03	1956.4	0.204	0.434	0.298
OAT	886.8	0.094	0.146	0.02	474.8	0.384	0.176	0.16
VAE-MDP	1527.8	0.226	0.204	0.02	430.8	0.142	0.53	0.256
VAE	1302.4	-0.03	0.334	0.062	834.8	0.04	0.654	0.484
TimeGAN	1538.2	0.006	0.216	0.022	1209	0.408	0.604	0.34
SPAWNER	1817.8	-0.192	0.218	0.144	1560.6	0.338	0.436	0.272
DGW	1578	-0.028	0.292	0.054	1226.8	0.164	0.434	0.294
Permutation	1548.6	-0.132	0.27	0.076	1628	0.338	0.51	0.152
Jittering	1632.8	-0.096	0.202	0.076	1407.4	0.038	0.574	0.168
Scaling	1308.2	-0.02	0.382	0.076	1462.8	0.244	0.72	0.212
TDA	1030.4	-0.116	0.262	0.06	832.2	0.182	0.608	0.496
	Hammer				Door			
	MAE	Rank Corr.	Regret@1	Regret@5	MAE	Rank Corr.	Regret@1	Regret@5
NoAug.	5266	0.344	0.34	0.058	603.8	0.14	0.274	0.004
OAT	2901.6	0.566	0.116	0.028	388.8	0.474	0.232	0.008
VAE-MDP	3418.6	0.02	0.454	0.126	497.2	0.336	0.224	0.052
VAE	3733.2	-0.198	0.47	0.104	642.8	0.482	0.288	0.046
TimeGAN	4681	0.262	0.34	0.24	507	0.392	0.27	0.03
SPAWNER	4244.8	-0.156	0.55	0.37	687.6	0.146	0.37	0.186
DGW	5238.8	0.242	0.296	0.144	578.2	0.342	0.134	0.052
Permutation	4103.2	-0.202	0.534	0.076	583.4	0.264	0.236	0.088
Jittering	4256.4	0.004	0.452	0.102	700.2	0.268	0.342	0.056
Scaling	3832.4	0.166	0.404	0.102	580.8	0.222	0.326	0.078
TDA	3448	-0.298	0.56	0.078	511.2	0.118	0.376	0.112

Table 2. OPE results averaging across five OPE methods without augmentation and with each augmentation method in Adroit `human`.

FQE	pen	door	relocate	hammer	MB	pen	door	relocate	hammer
NoAug.	3872±140	389±60	593±113	6000±612		1218±23	403±18	353±21	3778±78
OAT	**540±17**	**255±17**	**452±40**	**3359±113**		**556±207**	**373±59**	**263±15**	**2985±384**
VAE-MDP	843±44	498±10	**419±6**	**3358±24**		**541±310**	493±3	423±1	**2912±13**
TimeGAN	919±141	459±92	962±96	4353±495		1063±450	435±34	413±1	5811±97
VAE	614±9	531±7	520±2	3746±16		615±14	530±3	421±1	3734±9
SPAWNER	1508±52	504±35	735±131	6529±172		1107±0	473±4	698±4	3561±11
DGW	792±222	320±20	787±69	9578±431		1278±175	430±29	810±6	3779±37
Permutation	924±140	520±41	542±10	4069±56		998±0	483±4	754±4	3324±11
Jittering	1092±105	403±30	746±0	5526±323		1064±9	539±11	512±60	3257±103
Scaling	875±52	360±20	624±179	4599±306		961±337	478±5	524±84	3627±63
TDA	1185±29	469±6	805±10	3407±22		**471±248**	470±4	810±4	3402±11
IS	pen	door	relocate	hammer	DR	pen	door	relocate	hammer
NoAug.	3926±128	870±173	3926±128	7352±1118		2846±200	379±65	606±116	5768±751
OAT	**1328±69**	**502±11**	**435±2**	**3529±13**		**731±50**	**340±21**	**447±24**	**3379±458**
VAE-MDP	**1315±383**	**499±6**	**437±4**	3678±83		2954±883	502±10	**451±23**	3811±262
TimeGAN	1752±212	591±12	1995±5	5683±12		1352±282	494±70	667±122	4224±138
VAE	1896±87	513±5	930±7	3628±174		784±155	515±7	**545±16**	3832±166
SPAWNER	2769±0	1007±9	2871±19	3567±10		870±39	450±86	591±69	4008±444
DGW	2360±0	541±10	534±8	5289±11		861±99	545±55	573±38	4270±92
Permutation	2433±11	520±13	3093±20	5332±11		787±149	368±42	613±84	4467±245
Jittering	2350±0	1114±2	2111±28	5334±10		1058±127	419±12	519±28	3841±274
Scaling	**1284±40**	523±8	2118±72	3710±16		822±132	525±7	642±29	3892±239
TDA	**1269±101**	572±5	882±25	**3418±134**		991±193	477±11	857±156	3613±261
DICE	pen	door	relocate	hammer					
NoAug.	3208±22	978±10	4304±68	3432±6					
OAT	**1279±5**	**474±5**	777±14	**1256±8**					
VAE-MDP	1986±40	494±5	**424±3**	3334±9					
TimeGAN	2605±15	556±6	2008±15	3334±9					
VAE	2603±3	1125±11	1758±10	3726±18					
SPAWNER	2835±11	1004±10	2908±49	3559±12					
DGW	2599±0	1055±10	3430±63	3278±9					
Permutation	2601±2	1026±11	3138±49	3324±10					
Jittering	2600±1	1026±11	3149±51	3324±10					
Scaling	2599±0	1018±11	3406±60	3334±10					
TDA	**1236±8**	568±5	807±14	3400±11					

Table 3. MAE results of OPE without and with each augmentation method in Adroit `human` environment. Results are obtained by averaging over 3 random seeds used for training at a discount factor of 0.995, with standard deviations shown after ±.

FQE	pen	door	relocate	hammer	MB	pen	door	relocate	hammer
NoAug.	0.31±0.21	0.07±0.09	0.62±0.11	0.14±0.10		-0.12±0.33	0.13±0.13	0.16±0.10	0.29±0.23
OAT	-0.10±0.09	**0.56±0.07**	0.25±0.11	**0.53±0.12**		0.23±0.38	0.37±0.35	0.30±0.00	**0.41±0.05**
VAE-MDP	-0.02±0.64	**0.68±0.24**	-0.86±0.04	-0.44±0.72		0.12±0.34	0.14±0.58	**0.71±0.16**	0.29±0.00
TimeGAN	-0.17±0.28	0.39±0.20	**0.76±0.07**	0.37±0.06		**0.52±0.48**	0.18±0.13	-0.02±0.19	0.16±0.23
VAE	-0.52±0.38	0.39±0.20	-0.83±0.11	-0.61±0.46		0.46±0.28	**0.93±0.02**	0.25±0.30	0.11±0.67
SPAWNER	0.12±0.20	0.44±0.17	0.62±0.11	-0.11±0.31		0.13±0.44	0.04±0.79	0.19±0.60	-0.82±0.02
DGW	-0.12±0.25	0.47±0.21	0.17±0.36	0.47±0.16		-0.02±0.15	0.29±0.19	0.35±0.31	0.20±0.19
Permutation	-0.17±0.18	0.50±0.05	0.43±0.09	0.48±0.03		-0.18±0.56	0.02±0.74	0.23±0.07	-0.85±0.10
Jittering	-0.17±0.21	0.45±0.00	-0.33±0.78	-0.29±0.26		0.05±0.04	0.31±0.01	0.17±0.42	-0.22±0.20
Scaling	**0.36±0.24**	0.53±0.06	0.40±0.15	0.35±0.34		-0.35±0.38	0.21±0.65	0.24±0.33	0.35±0.15
TDA	-0.26±0.68	-0.41±0.35	-0.27±0.89	-0.12±0.52		0.09±0.09	0.72±0.11	**0.71±0.12**	-0.21±0.60
IS	pen	door	relocate	hammer	DR		door	relocate	hammer
NoAug.	0.28±0.28	0.12±0.35	0.23±0.07	0.39±0.07		0.36±0.29	0.01±0.18	0.65±0.19	0.04±0.25
OAT	0.26±0.30	0.28±0.46	**0.82±0.07**	0.18±0.44		**0.29±0.04**	**0.66±0.05**	**0.76±0.14**	**0.55±0.15**
VAE-MDP	**0.48±0.36**	0.12±0.71	0.22±0.70	-0.28±0.51		**0.34±0.68**	0.50±0.11	0.72±0.18	-0.35±0.79
TimeGAN	-0.01±0.80	0.38±0.75	-	-0.85±0.01		-0.20±0.64	0.53±0.10	0.65±0.08	0.40±0.03
VAE	-	**0.65±0.00**	-0.31±0.00	0.31±0.86		-0.17±0.64	0.29±0.29	-0.04±0.69	0.06±0.75
SPAWNER	-0.83±0	-	-	**0.81±0.16**		-0.09±0.20	0.36±0.26	0.62±0.09	0.47±0.16
DGW	-	-0.03±0.61	**0.82±0.16**	0.26±0.57		0.02±0.06	0.63±0.15	0.09±0.67	0.27±0.36
Permutation	-0.21±0.74	0.37±0.56	-0.70±0.00	-		-0.12±0.37	0.57±0.14	0.38±0.31	0.27±0.36
Jittering	-0.28±0.79	0.12±0.38	**0.85±0.00**	0.23±0.73		-0.08±0.51	**0.64±0.18**	-0.04±0.42	0.18±0.84
Scaling	-	0.24±0.42	**0.65±0.00**	0.66±0.14		-0.15±0.67	0.26±0.45	0.09±0.67	0.09±0.67
TDA	-0.15±0.53	0.37±0.28	-0.67±0.00	**0.81±0.13**		-0.11±0.63	0.18±0.34	-0.22±0.85	-0.28±0.79
DICE	pen	door	relocate	hammer					
NoAug.	-0.01±0.39	**0.61±0.34**	-0.18±0.45	**0.94±0.01**					
OAT	-0.21±0.45	0.50±0.21	**0.33±0.86**	0.52±0.67					
VAE-MDP	**0.21±0.59**	0.24±0.53	0.02±0.68	0.38±0.73					
TimeGAN	-0.11±0.19	0.48±0.09	0.27±0.84	0.38±0.73					
VAE	0.08±0.45	0.15±0.74	0.17±0.76	-0.24±0.16					
SPAWNER	-0.29±0.59	-0.11±0.71	0.26±0.83	-0.32±0.58					
DGW	-0.02±0.54	0.35±0.57	0.24±0.87	-0.55±0.28					
Permutation	0.02±0.64	-0.14±0.59	0.28±0.88	-0.21±0.75					
Jittering	0.00±0.58	-0.18±0.60	0.27±0.84	-0.50±0.45					
Scaling	0.04±0.67	-0.13±0.58	0.25±0.87	-0.61±0.30					
TDA	-0.15±0.64	-0.27±0.62	**0.32±0.91**	-0.21±0.84					

Table 4. Rank correlation results of OPE without and with each augmentation method in Adroit `human` environment. Results are obtained by averaging over 3 random seeds used for training at a discount factor of 0.995, with standard deviations shown after ±.

FQE	pen	door	relocate	hammer	MB	pen	door	relocate	hammer
NoAug.	0.07±0.05	0.05±0.08	0.17±0.14	0.46±0.23		0.15±0.15	0.44±0.42	0.73±0.36	0.15±0.17
OAT	0.23±0.17	0.17±0.15	**0.03±0.02**	**0.01±0.01**		**0.05±0.03**	0.51±0.37	0.41±0.28	0.52±0.37
VAE-MDP	0.38±0.21	**0.01±0.01**	1.00±0.01	0.81±0.30		0.13±0.11	0.39±0.45	0.41±0.28	0.42±0.16
TimeGAN	0.19±0.13	0.23±0.13	0.39±0.24	0.15±0.17		0.22±0.25	0.51±0.36	1.00±0.00	0.19±0.24
VAE	0.57±0.00	0.05±0.05	1.00±0.01	0.73±0.26		0.09±0.06	**0.00±0.01**	**0.03±0.02**	0.34±0.48
SPAWNER	**0.03±0.00**	**0.01±0.00**	0.27±0.32	0.94±0.11		0.12±0.12	0.35±0.48	0.41±0.43	1.02±0.00
DGW	0.23±0.17	0.14±0.08	0.32±0.42	0.02±0.01		0.33±0.12	0.12±0.09	0.67±0.47	**0.14±0.18**
Permutation	0.37±0.16	0.15±0.16	0.87±0.11	0.05±0.03		0.19±0.13	0.37±0.47	0.09±0.09	1.02±0.00
Jittering	0.11±0.06	0.12±0.00	0.67±0.47	0.73±0.26		**0.08±0.07**	0.63±0.10	0.48±0.41	0.45±0.31
Scaling	0.13±0.16	0.17±0.11	0.81±0.12	0.25±0.22		0.50±0.10	0.68±0.48	0.94±0.04	**0.14±0.08**
TDA	0.35±0.25	0.79±0.29	0.68±0.45	0.81±0.30		0.26±0.23	0.12±0.10	0.31±0.30	0.72±0.37
IS	pen	door	relocate	hammer	DR	pen	door	relocate	hammer
NoAug.	0.17±0.15	0.45±0.40	0.63±0.41	0.19±0.30		0.09±0	0.05±0.09	0.17±0.15	0.46±0.23
OAT	**0.00±0.00**	**0.04±0.06**	**0.02±0.02**	**0.00±0.00**		**0.04±0.04**	**0.00±0.01**	0.05±0.00	**0.02±0.01**
VAE-MDP	0.05±0.03	0.37±0.47	0.37±0.45	0.30±0.34		0.16±0.22	**0.00±0.01**	0.05±0.00	0.71±0.43
TimeGAN	0.38±0.27	0.13±0.17	1.00±0.00	1.00±0.02		0.16±0.21	0.13±0.17	0.26±0.33	0.33±0.32
VAE	0.44±0.19	0.68±0.48	1.00±0.00	0.19±0.25		0.36±0.15	0.21±0.25	0.74±0.37	0.37±0.46
SPAWNER	0.57±0.00	1.03±0.00	1.00±0.00	0.34±0.45		0.12±0.08	**0.01±0.01**	**0.00±0.00**	0.02±0.25
DGW	0.44±0.19	0.15±0.16	**0.00±0.00**	0.59±0.28		0.25±0.24	0.03±0.02	0.68±0.44	0.03±0.00
Permutation	0.44±0.19	0.05±0.06	0.91±0.13	0.74±0.39		0.10±0.06	0.08±0.12	0.31±0.19	0.50±0.40
Jittering	0.44±0.19	0.34±0.48	0.68±0.45	0.36±0.46		0.10±0.05	0.09±0.11	0.54±0.31	0.35±0.47
Scaling	0.57±0.00	0.06±0.05	0.67±0.47	0.26±0.18		0.39±0.19	0.19±0.26	0.68±0.44	0.68±0.44
TDA	0.10±0.13	0.05±0.06	1.00±0.00	**0.00±0.01**		0.39±0.19	0.35±0.46	0.68±0.44	0.60±0.43
DICE	pen	door	relocate	hammer					
NoAug.	0.44±0.19	**0.38±0.46**	0.47±0.40	0.44±0.01					
OAT	0.41±0.17	0.44±0.09	**0.37±0.45**	**0.03±0.05**					
VAE-MDP	0.30±0.22	0.35±0.15	0.82±0.26	**0.03±0.05**					
TimeGAN	**0.13±0.12**	0.35±0.15	**0.37±0.45**	**0.03±0.05**					
VAE	0.21±0.15	0.50±0.35	0.50±0.39	0.72±0.37					
SPAWNER	0.25±0.11	0.45±0.32	0.50±0.39	0.43±0.39					
DGW	0.21±0.11	0.23±0.23	0.50±0.39	0.70±0.42					
Permutation	0.25±0.11	0.53±0.22	**0.37±0.45**	0.36±0.44					
Jittering	0.28±0.15	0.53±0.22	0.50±0.39	0.37±0.43					
Scaling	0.32±0.20	0.53±0.22	0.50±0.39	0.69±0.41					
TDA	0.21±0.15	0.57±0.16	**0.37±0.45**	0.67±0.47					

Table 5. Regret@1 results of OPE without and with each augmentation method in Adroit `human` environment. Results are obtained by averaging over 3 random seeds used for training at a discount factor of 0.995, with standard deviations shown after ±.

FQE	pen	door	relocate	hammer	MB	pen	door	relocate	hammer
NoAug.	**0.01±0.02**	**0.00±0.01**	0.75±0.22	0.13±0.09		0.03±0.04	0.02±0.03	0.03±0.02	0.01±0.01
OAT	**0.01±0.02**	**0.00±0.01**	0.45±0.37	0.13±0.09		**0.01±0.01**	**0.00±0.00**	0.05±0.00	**0.00±0.00**
VAE-MDP	0.06±0.08	**0.00±0.00**	0.87±0.10	0.13±0.09		**0.01±0.01**	0.08±0.12	**0.00±0.00**	0.17±0.05
TimeGAN	0.03±0.04	**0.00±0.01**	0.47±0.35	0.13±0.09		**0.00±0.00**	0.01±0.01	0.02±0.02	0.02±0.01
VAE	0.06±0.08	**0.00±0.00**	0.75±0.22	0.13±0.09		**0.00±0.00**	**0.00±0.00**	**0.00±0.00**	0.07±0.09
SPAWNER	0.02±0.01	**0.01±0.00**	**0.00±0.00**	**0.01±0.01**		0.02±0.01	0.19±0.26	0.07±0.10	0.17±0.05
DGW	**0.01±0.02**	**0.00±0.00**	0.68±0.32	0.10±0.08		0.03±0.04	0.01±0.01	0.15±0.10	0.01±0.01
Permutation	**0.00±0.00**	**0.00±0.00**	**0.02±0.02**	**0.01±0.01**		0.10±0.06	0.19±0.26	**0.00±0.00**	0.23±0.12
Jittering	**0.00±0.00**	**0.00±0.00**	0.45±0.37	0.02±0.01		0.08±0.07	0.01±0.01	0.07±0.10	0.05±0.04
Scaling	**0.00±0.00**	**0.01±0.01**	0.03±0.02	**0.00±0.01**		0.08±0.07	0.08±0.12	0.07±0.10	0.01±0.01
TDA	0.07±0.08	0.27±0.23	0.56±0.41	**0.01±0.01**		0.10±0.06	0.00±0.00	0.07±0.10	0.15±0.17
IS	pen	door	relocate	hammer	DR	pen	door	relocate	hammer
NoAug.	**0.00±0.00**	**0.00±0.00**	0.07±0.10	0.01±0.02		0.10±0.08	**0.00±0.01**	0.32±0.46	0.14±0.18
OAT	**0.00±0.00**	**0.04±0.06**	**0.00±0.00**	**0.00±0.00**		**0.01±0.01**	**0.00±0.01**	**0.00±0.00**	**0.00±0.00**
VAE-MDP	**0.01±0.01**	0.09±0.11	0.30±0.43	0.14±0.18		**0.01±0.01**	**0.00±0.00**	0.02±0.02	0.16±0.16
TimeGAN	0.01±0.01	0.08±0.12	0.97±0.00	1.02±0.00		0.06±0.08	0.00±0.01	0.09±0.09	**0.00±0.00**
VAE	0.19±0.27	**0.00±0.00**	0.95±0.03	0.13±0.18		0.05±0.06	0.08±0.12	0.40±0.41	0.14±0.18
SPAWNER	0.57±0.00	0.56±0.00	0.97±0.00	1.02±0.00		0.04±0.04	0.01±0.01	**0.00±0.00**	0.60±0.43
DGW	0.19±0.27	0.15±0.16	**0.00±0.00**	0.46±0.40		**0.01±0.01**	0.01±0.01	0.34±0.45	**0.00±0.01**
Permutation	0.19±0.27	**0.04±0.06**	0.44±0.00	**0.00±0.00**		0.03±0.00	**0.00±0.00**	**0.00±0.00**	**0.00±0.01**
Jittering	0.19±0.27	**0.00±0.00**	**0.00±0.00**	0.15±0.17		0.05±0.06	0.00±0.01	0.02±0.02	0.13±0.18
Scaling	0.18±0.00	0.05±0.06	0.32±0.46	0.01±0.01		0.06±0.08	0.08±0.12	0.34±0.45	0.34±0.45
TDA	**0.01±0.02**	**0.00±0.01**	0.97±0.00	**0.00±0.00**		0.06±0.08	**0.00±0.00**	0.56±0.41	0.16±0.16
DICE	pen	door	relocate	hammer					
NoAug.	**0.01±0.01**	0.00±0.01	0.32±0.42	**0.00±0.00**					
OAT	0.07±0.08	**0.00±0.00**	0.30±0.43	**0.01±0.02**					
VAE-MDP	**0.01±0.01**	0.09±0.11	**0.09±0.09**	0.03±0.05					
TimeGAN	**0.01±0.02**	0.06±0.00	0.15±0.21	0.03±0.05					
VAE	**0.01±0.02**	0.15±0.16	0.32±0.42	0.05±0.03					
SPAWNER	0.07±0.08	0.16±0.12	0.32±0.42	0.05±0.04					
DGW	0.03±0.04	0.09±0.11	0.30±0.43	0.15±0.17					
Permutation	0.06±0.08	0.21±0.15	0.30±0.43	0.14±0.18					
Jittering	0.06±0.08	0.27±0.22	0.30±0.43	0.16±0.16					
Scaling	0.06±0.08	0.17±0.11	0.30±0.43	0.15±0.17					
TDA	0.06±0.08	0.29±0.21	0.32±0.46	0.07±0.05					

Table 6. Regret@5 results of OPE without and with each augmentation method in Adroit `human` environment. Results are obtained by averaging over 3 random seeds used for training at a discount factor of 0.995, with standard deviations shown after ±.

FQE	pen	door	relocate	hammer	MB	pen	door	relocate	hammer
NoAug.	715±11	359±16	371±5	3714±34		622±102	1009±92	357±6	4119±7
OAT	**248±5**	**295±35**	**356±33**	**3453±73**		522±42	**418±33**	**352±14**	**3518±28**
VAE-MDP	**323±9**	539±6	384±4	3624±16		813±674	538±3	386±1	3611±4
TimeGAN	1237±25	572±7	539±4	5268±24		1065±118	573±4	541±1	5268±15
VAE	412±8	561±6	384±2	3823±14		**409±19**	561±3	385±1	3814±10
SPAWNER	1173±0	502±5	443±2	3988±6		1173±0	502±3	446±1	3988±6
DGW	1176±0	550±5	864±8	4127±6		1176±0	550±3	871±4	4127±6
Permutation	1176±0	537±5	822±11	4128±6		1176±0	536±3	832±4	4128±6
Jittering	1176±0	537±5	820±11	4128±6		1176±0	536±3	833±4	4128±6
Scaling	1176±0	517±6	696±5	4123±6		1176±0	517±3	705±4	4123±6
TDA	610±4	526±6	438±3	3746±16		599±0	527±3	441±1	3715±14
IS	pen	door	relocate	hammer	DR	pen	door	relocate	hammer
NoAug.	636±21	1072±24	458±13	8162±91		731±115	458±34	475±22	6719±118
OAT	**259±9**	**482±8**	438±3	**3752±10**		**241±4**	**382±70**	**388±11**	**3396±166**
VAE-MDP	300±17	507±4	**406±11**	3686±47		493±76	542±20	415±40	3689±89
TimeGAN	978±192	576±4	695±27	5325±85		771±191	606±18	696±26	5514±133
VAE	300±21	517±2	433±15	**3722±15**		386±19	540±22	437±20	3764±80
SPAWNER	1173±0	611±3	506±1	3989±6		1174±0	527±14	509±14	3990±5
DGW	1176±0	567±3	1149±4	4129±6		1176±0	597±23	1180±35	4129±5
Permutation	1176±0	557±3	1107±4	4129±6		1176±0	592±25	1110±39	4129±5
Jittering	1176±0	557±3	1107±4	4129±6		1176±0	588±24	1115±89	4130±5
Scaling	1176±0	543±3	945±4	4124±6		1176±0	580±31	968±35	4125±5
TDA	604±6	501±5	445±10	**3725±11**		261±12	539±34	453±9	3765±142
DICE	pen	door	relocate	hammer					
NoAug.	1218±41	1138±7	1841±15	3752±8					
OAT	1123±169	775±164	1820±6	**3748±15**					
VAE-MDP	778±5	**538±4**	1606±8	**3614±15**					
TimeGAN	1276±42	573±4	1925±10	5252±18					
VAE	1020±14	561±5	1602±10	3813±20					
SPAWNER	1173±0	1067±7	1856±4	3988±6					
DGW	1176±0	1140±3	869±8	4127±6					
Permutation	1176±0	1140±7	832±5	4128±6					
Jittering	1176±0	1140±7	831±8	4128±6					
Scaling	1176±0	1098±7	**704±8**	4123±6					
TDA	**601±0**	1119±9	1835±6	**3731±15**					

Table 7. MAE results of OPE without and with each augmentation method in resampled Adroit `cloned` environment. The data are randomly sampled from original training data as the same data points as the corresponding task in `human` environment. Results are obtained by averaging over 3 random seeds used for training at a discount factor of 0.995, with standard deviations shown after ±.

FQE	pen	door	relocate	hammer	MB	pen	door	relocate	hammer
NoAug.	**0.51±0.25**	0.55 ± 0.27	-0.28±0.17	0.50 ± 0.09		**0.29±0.19**	0.63±0.14	0.45±0.30	0.22±0.12
OAT	0.40±0.60	**0.59±0.18**	**0.18±0.44**	**0.55±0.34**		-0.10±0.41	0.49±0.10	0.14±0.78	0.20±0.19
VAE-MDP	-0.18±0.44	0.26±0.52	-0.32±0.65	0.28±0.78		0.10±0.44	0.19±0.18	**0.88±0.11**	0.05±0.61
TimeGAN	-0.39±0.43	0.54±0.28	-0.47±0.41	-0.01±0.71		-0.12±0.63	0.58±0.34	0.46±0.71	**0.68±0.30**
VAE	-0.69±0.19	0.38±0.48	-0.40±0.62	0.16±0.50		**0.37±0.47**	0.56±0.29	0.36±0.73	-0.23±0.23
SPAWNER	-0.19±0.68	0.18±0.59	-0.32±0.73	0.02±0.79		0.05±0.70	0.62±0.06	0.79±0.19	0.40±0.82
DGW	-0.46±0.49	-0.07±0.67	-0.75±0.11	-0.32±0.82		-0.30±0.20	0.47±0.14	-0.09±0.58	0.09±0.25
Permutation	-0.36±0.69	-0.28±0.24	0.12±0.59	-0.29±0.79		0.15±0.25	0.60±0.13	0.43±0.75	0.51±0.32
Jittering	-0.32±0.80	0.54±0.49	-0.17±0.53	-0.47±0.49		0.25±0.60	**0.67±0.10**	-0.19±0.46	0.45±0.44
Scaling	-0.19±0.80	**0.63±0.22**	-0.25±0.64	-0.26±0.83		0.03±0.62	0.45±0.30	0.67±0.29	0.66±0.18
TDA	-0.26±0.77	0.37±0.26	-0.22±0.48	-0.31±0.87		0.00±0.50	0.35±0.33	0.61±0.27	-0.45±0.62
IS	pen	door	relocate	hammer	DR	pen	door	relocate	hammer
NoAug.	0.00±0.00	-0.32±0.59	0.25±0.54	0.79±0.04		**0.42±0.24**	**0.46±0.72**	-0.22±0.70	0.29±0.49
OAT	**0.78±0.00**	**0.60±0.30**	**0.98±0.00**	0.86±0.03		0.37±0.33	0.17±0.36	**0.37±0.83**	**0.58±0.17**
VAE-MDP	0.15±0.28	0.55±0.06	0.96±0.04	-0.02±0.74		-0.15±0.60	-0.57±0.06	-0.02±0.19	-0.23±0.82
TimeGAN	0.89±0.00	-0.78±0.27	0.88±0.08	0.50±0.29		-0.43±0.36	-0.42±0.14	-0.10±0.62	-0.18±0.78
VAE	0.61±0.11	-0.23±0.66	0.88±0.10	**0.98±0.01**		-0.46±0.34	-0.10±0.62	-0.02±0.70	-0.20±0.76
SPAWNER	-0.89±0.00	-0.10±0.00	-0.29±0.25	0.37±0.39		-0.23±0.66	0.02±0.64	-0.06±0.64	-0.16±0.78
DGW	-	-0.45±0.00	-0.67±0.00	-		-0.05±0.32	-0.36±0.31	-0.09±0.63	-0.20±0.77
Permutation	-	-0.50±0.00	-0.67±0.00	-		-0.18±0.36	-0.43±0.21	-0.20±0.71	-0.22±0.75
Jittering	-	-	-0.42±0.02	-		-0.31±0.26	-0.33±0.39	0.03±0.74	0.06±0.78
Scaling	-	-0.67±0.00	-0.24±0.44	-		0.07±0.55	-0.31±0.38	-0.25±0.71	-0.20±0.73
TDA	0.37±0.42	0.41±0.22	-0.48±0.35	0.79±0.13		-0.22±0.37	-0.32±0.41	-0.05±0.70	-0.25±0.79
DICE	pen	door	relocate	hammer					
NoAug.	0.03±0.56	**0.39±0.51**	0.05±0.52	**0.31±0.76**					
OAT	-0.24±0.06	**0.38±0.42**	0.39±0.66	**0.28±0.72**					
VAE-MDP	-0.61±0.38	0.26±0.65	**0.82±0.17**	-0.28±0.79					
TimeGAN	-0.36±0.36	0.06±0.70	0.77±0.24	-0.43±0.62					
VAE	-0.18±0.45	0.18±0.79	0.30±0.61	-0.11±0.64					
SPAWNER	-0.30±0.33	**0.47±0.66**	-0.18±0.78	-0.19±0.73					
DGW	**0.16±0.10**	**0.38±0.58**	0.22±0.85	-0.36±0.45					
Permutation	0.24±0.43	-0.48±0.11	**0.96±0.02**	-0.20±0.75					
Jittering	-0.39±0.61	**0.37±0.43**	-0.30±0.82	-0.14±0.77					
Scaling	**0.14±0.60**	**0.40±0.55**	0.14±0.81	-0.16±0.70					
TDA	-0.02±0.56	-0.11±0.61	0.30±0.83	-0.31±0.78					

Table 8. Rank correlation results of OPE without and with each augmentation method in resampled Adroit `cloned` environment. The data are randomly sampled from original training data as the same data points as the corresponding task in `human` environment. Results are obtained by averaging over 3 random seeds used for training at a discount factor of 0.995, with standard deviations shown after ±.

FQE	pen	door	relocate	hammer	MB	pen	door	relocate	hammer
NoAug.	0.10±0.13	0.07±0.04	1.00±0.01	**0.03±0.01**		**0.06±0.06**	0.19±0.14	0.34±0.45	0.21±0.18
OAT	**0.05±0.06**	**0.00±0.01**	**0.17±0.20**	**0.07±0.09**		**0.07±0.08**	0.17±0.11	0.63±0.26	0.36±0.24
VAE-MDP	0.30±0.22	0.13±0.17	0.67±0.47	0.26±0.37		0.20±0.26	0.79±0.19	**0.02±0.02**	0.39±0.31
TimeGAN	0.41±0.23	0.39±0.45	1.00±0.00	0.40±0.44		0.31±0.22	0.42±0.44	0.35±0.46	0.08±0.08
VAE	0.57±0.00	0.46±0.41	0.66±0.47	0.47±0.39		0.12±0.06	0.08±0.12	0.33±0.47	0.94±0.11
SPAWNER	0.20±0.26	0.44±0.32	0.67±0.47	0.37±0.46		0.13±0.16	0.21±0.25	0.26±0.33	0.26±0.37
DGW	0.05±0.03	0.43±0.43	0.99±0.02	0.68±0.48		0.57±0.00	0.52±0.41	0.90±0.12	0.01±0.02
Permutation	0.38±0.27	0.72±0.27	0.74±0.36	0.60±0.43		0.57±0.00	**0.00±0.00**	0.40±0.42	0.34±0.48
Jittering	0.35±0.25	0.09±0.11	0.67±0.47	0.47±0.39		0.14±0.11	0.02±0.03	0.81±0.12	**0.01±0.02**
Scaling	0.35±0.25	0.09±0.11	0.67±0.47	0.60±0.43		0.24±0.24	0.19±0.26	0.59±0.39	0.01±0.01
TDA	0.35±0.25	0.19±0.26	0.67±0.47	0.68±0.48		0.17±0.21	0.47±0.42	0.40±0.42	0.81±0.30
IS	pen	door	relocate	hammer	DR	pen	door	relocate	hammer
NoAug.	0.57±0.00	0.47±0.42	0.68±0.45	**0.34±0.48**		0.22±0.25	**0.26±0.36**	0.68±0.45	**0.26±0.37**
OAT	**0.19±0.27**	**0.08±0.12**	**0.00±0.00**	**0.00±0.00**		**0.08±0.07**	**0.25±0.22**	**0.37±0.45**	0.34±0.45
VAE-MDP	**0.19±0.13**	**0.07±0.04**	**0.00±0.00**	0.35±0.47		0.17±0.10	0.93±0.11	0.82±0.26	0.37±0.46
TimeGAN	0.57±0.00	1.02±0.00	0.02±0.02	0.85±0.09		0.50±0.05	0.93±0.11	0.82±0.26	0.37±0.46
VAE	0.35±0.24	0.68±0.48	**0.00±0.00**	**0.34±0.46**		0.44±0.19	0.62±0.41	0.67±0.47	0.37±0.46
SPAWNER	0.57±0.00	0.81±0.31	1.00±0.00	0.37±0.43		0.35±0.24	0.60±0.43	0.68±0.45	0.37±0.46
DGW	0.57±0.00	1.03±0.00	1.00±0.00	1.02±0.00		0.35±0.24	0.93±0.11	0.68±0.45	0.37±0.46
Permutation	0.57±0.00	1.03±0.00	1.00±0.00	1.02±0.00		0.35±0.25	0.93±0.11	0.68±0.44	0.37±0.46
Jittering	0.57±0.00	1.03±0.00	1.00±0.00	1.02±0.00		0.54±0.05	0.62±0.41	0.73±0.36	0.37±0.46
Scaling	0.57±0.00	1.03±0.00	0.91±0.13	1.02±0.00		0.17±0.21	0.62±0.41	0.81±0.26	0.37±0.46
TDA	**0.17±0.21**	**0.08±0.06**	1.00±0.00	0.07±0.09		0.25±0.24	0.62±0.41	0.68±0.45	0.37±0.46
DICE	pen	door	relocate	hammer					
NoAug.	0.26±0.18	0.34±0.48	0.73±0.36	0.34±0.48					
OAT	0.36±0.18	0.37±0.44	0.37±0.45	**0.17±0.05**					
VAE-MDP	0.36±0.18	0.36±0.30	**0.03±0.02**	0.67±0.47					
TimeGAN	0.44±0.12	0.60±0.25	0.11±0.08	0.68±0.48					
VAE	0.31±0.24	0.50±0.35	0.35±0.46	0.65±0.46					
SPAWNER	0.26±0.23	0.34±0.48	0.65±0.43	0.40±0.44					
DGW	0.29±0.23	0.36±0.47	0.41±0.43	0.53±0.35					
Permutation	0.50±0.10	0.94±0.12	**0.02±0.02**	0.40±0.44					
Jittering	0.38±0.27	0.02±0.03	0.68±0.45	0.47±0.42					
Scaling	0.50±0.10	0.34±0.48	0.73±0.36	0.40±0.44					
TDA	**0.08±0.06**	0.60±0.25	0.37±0.45	0.68±0.48					

Table 9. Regret@1 results of OPE without and with each augmentation method in resampled Adroit `cloned` environment. The data are randomly sampled from original training data as the same data points as the corresponding task in `human` environment. Results are obtained by averaging over 3 random seeds used for training at a discount factor of 0.995, with standard deviations shown after ±.

FQE	pen	door	relocate	hammer	MB	pen	door	relocate	hammer
NoAug.	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**	**0.00±0.01**		**0.00±0.00**	**0.00±0.01**	0.02±0.02	0.01±0.01
OAT	**0.01±0.02**	**0.00±0.01**	0.15±0.21	**0.00±0.01**		**0.00±0.00**	**0.00±0.00**	0.32±0.46	**0.00±0.00**
VAE-MDP	0.07±0.08	0.08±0.12	0.34±0.45	0.03±0.05		**0.00±0.00**	0.02±0.03	**0.00±0.00**	0.07±0.09
TimeGAN	0.06±0.08	**0.00±0.01**	0.36±0.43	0.07±0.09		0.01±0.02	**0.00±0.00**	**0.00±0.00**	0.00±0.00
VAE	0.08±0.07	**0.00±0.01**	0.45±0.37	0.08±0.09		0.01±0.02	**0.00±0.00**	0.02±0.02	0.01±0.01
SPAWNER	0.07±0.08	0.13±0.17	0.67±0.47	0.08±0.09		0.05±0.06	**0.00±0.00**	0.07±0.10	0.03±0.05
DGW	0.31±0.24	0.23±0.24	0.87±0.10	0.16±0.16		0.19±0.27	**0.00±0.00**	0.35±0.45	**0.00±0.00**
Permutation	0.38±0.27	0.09±0.11	0.15±0.21	0.16±0.16		0.10±0.08	**0.00±0.00**	0.24±0.34	**0.00±0.00**
Jittering	0.07±0.08	0.08±0.12	0.17±0.20	0.16±0.16		0.01±0.01	**0.00±0.00**	0.64±0.31	0.01±0.01
Scaling	0.35±0.25	0.02±0.03	0.39±0.30	0.16±0.16		0.07±0.08	0.08±0.12	**0.00±0.00**	**0.00±0.00**
TDA	0.07±0.08	0.04±0.06	0.17±0.20	0.10±0.08		0.01±0.02	**0.00±0.00**	0.09±0.09	0.14±0.08
IS	pen	door	relocate	hammer	DR	pen	door	relocate	hammer
NoAug.	0.18±0.00	0.25±0.00	0.63±0.44	0.13±0.18		**0.00±0.00**	0.13±0.18	0.56±0.41	**0.00±0.00**
OAT	**0.00±0.00**	**0.00±0.01**	**0.00±0.00**	**0.00±0.00**		**0.00±0.00**	**0.05±0.06**	0.24±0.34	**0.00±0.01**
VAE-MDP	0.01±0.01	**0.00±0.01**	**0.00±0.00**	0.14±0.18		0.09±0.06	0.25±0.18	**0.02±0.02**	0.16±0.16
TimeGAN	0.02±0.02	0.16±0.12	**0.00±0.00**	0.01±0.01		0.10±0.08	0.25±0.18	0.40±0.41	0.14±0.18
VAE	0.10±0.08	0.27±0.23	**0.00±0.00**	0.00±0.00		0.09±0.06	0.19±0.26	0.32±0.46	0.14±0.18
SPAWNER	0.18±0.00	0.56±0.00	0.02±0.02	0.00±0.01		0.06±0.08	0.13±0.18	0.26±0.33	0.14±0.18
DGW	0.18±0.00	0.56±0.00	0.97±0.00	0.39±0.00		**0.00±0.00**	0.15±0.16	0.34±0.45	0.14±0.18
Permutation	0.18±0.00	0.56±0.00	0.97±0.00	0.39±0.00		0.05±0.06	0.15±0.16	0.56±0.41	0.14±0.18
Jittering	0.18±0.00	0.56±0.00	0.97±0.00	0.39±0.00		0.01±0.01	0.13±0.17	0.24±0.34	0.13±0.18
Scaling	0.18±0.00	0.56±0.00	0.11±0.08	0.39±0.00		0.05±0.06	0.13±0.17	0.56±0.41	0.14±0.18
TDA	0.01±0.01	0.04±0.06	0.65±0.46	**0.00±0.00**		0.05±0.06	0.13±0.17	0.34±0.45	0.16±0.16
DICE	pen	door	relocate	hammer					
NoAug.	0.05±0.06	0.02±0.03	**0.00±0.00**	**0.03±0.05**					
OAT	**0.02±0.02**	**0.00±0.00**	**0.02±0.02**	**0.03±0.05**					
VAE-MDP	0.06±0.04	0.13±0.17	**0.00±0.00**	0.07±0.05					
TimeGAN	0.11±0.07	0.19±0.26	**0.00±0.00**	0.08±0.09					
VAE	0.03±0.04	0.13±0.18	0.15±0.21	0.07±0.05					
SPAWNER	**0.02±0.02**	0.13±0.18	0.56±0.41	0.07±0.05					
DGW	**0.01±0.01**	**0.00±0.00**	0.30±0.43	0.07±0.04					
Permutation	**0.00±0.00**	0.13±0.18	**0.00±0.00**	0.07±0.05					
Jittering	0.04±0.04	**0.00±0.00**	0.60±0.43	0.05±0.04					
Scaling	0.03±0.04	**0.00±0.00**	0.32±0.46	0.07±0.05					
TDA	0.06±0.04	0.23±0.24	0.30±0.43	0.07±0.05					

Table 10. Regret@5 results of OPE without and with each augmentation method in resampled Adroit `cloned` environment. The data are randomly sampled from original training data as the same data points as the corresponding task in `human` environment. Results are obtained by averaging over 3 random seeds used for training at a discount factor of 0.995, with standard deviations shown after ±.

FQE	pen	door	relocate	hammer	MB	pen	door	relocate	hammer
NoAug.	1101±47	1751±52	1729±557	2822±756		2363±135	**708±88**	621±28	13110±535
OAT	1270±39	**816±122**	**469±5**	2927±119		**1212±6**	972±172	**474±1**	11556±1335
VAE-MDP	**467±5**	**793±36**	**468±5**	**1983±45**		**1217±1**	746±10	**474±1**	**9890±105**
TimeGAN	2012±24	1008±37	994±11	6317±194		3108±2	1027±1	1007±2	19197±1121
VAE	1250±12	903±14	1605±8	2039±43		3109±1	912±6	1610±6	**10031±210**
SPAWNER	1247±11	1264±47	778±10	4738±157		3108±1	2738±6	778±10	21218±915
DGW	1245±11	1343±29	762±8	7264±117		3109±1	2885±6	762±8	36422±33
Permutation	1249±12	1270±28	763±8	6350±104		3109±1	2725±6	763±8	31965±28
Jittering	1247±13	1259±43	1772±19	6267±158		3108±2	2724±6	763±8	29471±76
Scaling	1247±11	1500±49	1495±15	5729±144		2589±367	1478±13	762±8	27187±209
TDA	1251±12	1058±45	**477±4**	3420±186		**1217±2**	1043±13	**481±1**	12350±1943
IS	pen	door	relocate	hammer	DR	pen	door	relocate	hammer
NoAug.	1881±23	1005±22	1863±43	3659±159		1632±783	970±86	1837±52	3262±856
OAT	904±48	983±25	1897±89	3809±54		1450±515	853±63	1888±145	2840±180
VAE-MDP	**676±24**	**871±23**	1858±42	**2225±110**		**1260±299**	886±71	1837±52	**2494±901**
TimeGAN	833±43	1054±44	1858±42	5929±83		1299±452	972±7	1837±52	5029±1100
VAE	1579±38	1480±41	1741±46	2481±209		2149±477	**654±63**	1734±77	**2179±158**
SPAWNER	1933±14	2400±6	1526±19	4128±6		2384±309	1089±51	1529±19	3840±606
DGW	2793±9	2571±6	1496±16	6935±6		3109±347	1172±53	**1499±16**	6556±643
Permutation	2675±12	2386±6	1497±16	5820±6		2997±344	1081±61	**1500±17**	5380±780
Jittering	2675±14	2386±6	1499±17	5821±6		3038±349	1081±50	**1503±17**	5465±701
Scaling	2683±12	2858±6	1498±16	5320±6		3046±392	1288±75	**1501±16**	4909±707
TDA	733±27	985±18	2052±113	3542±138		1325±398	963±89	2015±161	3141±902
DICE	pen	door	relocate	hammer					
NoAug.	3122±106	1250±21	2369±19	4171±47					
OAT	**1228±73**	1120±8	475±4	4201±33					
VAE-MDP	**1146±53**	812±6	473±3	**1996±44**					
TimeGAN	1943±94	570±10	694±5	6637±68					
VAE	3014±140	**428±1**	**387±2**	2021±38					
SPAWNER	3067±143	1294±7	1524±19	4985±30					
DGW	3103±37	1363±7	1492±16	7411±26					
Permutation	3108±63	1284±1	1495±16	6510±21					
Jittering	3104±57	1288±7	1495±16	6510±30					
Scaling	3097±52	1531±8	1494±16	5970±30					
TDA	**1193±54**	1087±7	479±2	3697±29					

Table 11. MAE results of OPE without and with each augmentation method in resampled Adroit `expert` environment. The data are randomly sampled from original training data as the same data points as the corresponding task in `human` environment. Results are obtained by averaging over 3 random seeds used for training at a discount factor of 0.995, with standard deviations shown after ±.

FQE	pen	door	relocate	hammer	MB	pen	door	relocate	hammer
NoAug.	**0.19±0.22**	0.87±0.07	-0.38±0.12	0.29±0.34		0.24±0.30	0.74±0.15	**0.86±0.05**	**0.06±0.35**
OAT	-0.55±0.29	0.85±0.04	**0.24±0.74**	**0.56±0.15**		**0.42±0.49**	0.15±0.11	**0.85±0.04**	-0.01±0.31
VAE-MDP	-0.18±0.53	0.33±0.47	-0.10±0.51	-0.31±0.90		0.29±0.71	**0.86±0.00**	0.30±0.73	-0.27±0.41
TimeGAN	-0.78±0.17	**0.94±0.02**	-0.10±0.51	-0.20±0.85		0.07±0.75	0.36±0.75	0.30±0.73	-0.25±0.56
VAE	-0.50±0.35	0.63±0.25	-0.47±0.35	-0.44±0.73		0.09±0.25	0.54±0.61	-0.15±0.72	-0.79±0.14
SPAWNER	-0.82±0.10	**0.96±0.04**	-0.22±0.50	-0.31±0.91		-0.40±0.53	0.29±0.60	-0.08±0.61	-0.00±0.68
DGW	-0.88±0.00	0.21±0.14	-0.96±0.03	-0.32±0.91		0.06±0.53	0.19±0.62	0.27±0.33	-0.01±0.72
Permutation	-0.79±0.13	0.80±0.19	-0.25±0.78	-0.31±0.92		-0.29±0.70	0.13±0.75	0.14±0.74	-0.65±0.32
Jittering	-0.82±0.12	**0.92±0.09**	-0.21±0.60	-0.31±0.92		**0.58±0.09**	0.23±0.83	**0.93±0.05**	-0.40±0.66
Scaling	-0.82±0.11	0.86±0.16	-0.09±0.55	-0.31±0.91		0.22±0.79	0.77±0.29	**0.92±0.07**	-0.37±0.55
TDA	-0.68±0.12	0.80±0.25	-0.06±0.80	-0.31±0.92		**0.46±0.52**	0.55±0.33	0.72±0.24	0.00±0.53
IS	pen	door	relocate	hammer	DR	pen	door	relocate	hammer
NoAug.	-0.02±0.48	0.08±0.38	**0.96±0.04**	0.34±0.50		-0.85±0.03	0.32±0.18	-0.28±0.75	0.33±0.85
OAT	-0.26±0.39	0.58±0.03	0.92±0.02	0.73±0.13		-0.85±0.05	**0.76±0.06**	**0.11±0.37**	**0.59±0.06**
VAE-MDP	-0.25±0.32	0.28±0.02	**0.96±0.00**	**1.00±0.00**		-0.83±0.08	0.41±0.31	-0.28±0.75	0.15±0.63
TimeGAN	**0.89±0.03**	**0.87±0.00**	**0.96±0.00**	0.61±0.14		**-0.82±0.05**	0.42±0.10	-0.28±0.75	0.15±0.80
VAE	-	0.19±0.04	0.75±0.00	0.49±0.45		-0.89±0.02	0.35±0.46	-0.26±0.38	0.08±0.75
SPAWNER	-0.89±0.00	-	-0.04±0.64	0.10±0.14		-0.89±0.02	0.08±0.28	**0.10±0.69**	0.23±0.85
DGW	-0.32±0.37	-	-0.59±0.26	-		-0.88±0.03	-0.16±0.28	-0.55±0.18	0.22±0.84
Permutation	-0.23±0.45	-	0.23±0.59	-		-0.90±0.02	0.19±0.16	-0.11±0.27	0.18±0.83
Jittering	0.11±0.59	-	0.50±0.07	-		-0.90±0.01	0.05±0.20	-0.75±0.16	0.22±0.84
Scaling	-0.46±0.21	-	0.27±0.62	0.00±0.00		-0.91±0.02	0.32±0.07	-0.52±0.16	0.19±0.84
TDA	0.31±0.00	0.59±0.35	0.90±0.11	0.80±0.14		-0.87±0.00	0.31±0.33	0.02±0.72	0.24±0.82
DICE	pen	door	relocate	hammer					
NoAug.	0.23±0.39	-0.33±0.89	0.11±0.62	-0.06±0.73					
OAT	**0.83±0.08**	**0.21±0.83**	0.35±0.91	**0.43±0.12**					
VAE-MDP	**0.87±0.07**	-0.23±0.82	-0.17±0.84	-0.20±0.85					
TimeGAN	**0.85±0.05**	-0.07±0.61	-0.17±0.84	-0.24±0.86					
VAE	0.65±0.19	-0.16±0.58	0.25±0.67	-0.11±0.59					
SPAWNER	0.55±0.50	0.11±0.74	-0.16±0.67	-0.10±0.78					
DGW	**0.86±0.06**	0.16±0.77	0.18±0.84	-0.07±0.75					
Permutation	**0.93±0.00**	0.13±0.73	**0.81±0.21**	-0.21±0.84					
Jittering	0.72±0.19	**0.28±0.85**	0.21±0.83	-0.12±0.79					
Scaling	0.81±0.14	**0.25±0.81**	0.15±0.80	-0.11±0.79					
TDA	0.76±0.08	0.17±0.80	0.23±0.57	-0.26±0.89					

Table 12. Rank correlation results of OPE without and with each augmentation method in resampled Adroit `expert` environment. The data are randomly sampled from original training data as the same data points as the corresponding task in `human` environment. Results are obtained by averaging over 3 random seeds used for training at a discount factor of 0.995, with standard deviations shown after ±.

FQE	pen	door	relocate	hammer	MB	pen	door	relocate	hammer
NoAug.	**0.17±0.14**	0.05±0.06	0.91±0.13	**0.05±0.04**		0.22±0.25	**0.07±0.04**	0.15±0.10	0.34±0.18
OAT	0.28±0.15	**0.03±0.02**	**0.33±0.47**	**0.10±0.08**		0.20±0.26	**0.09±0.05**	**0.00±0.00**	**0.23±0.12**
VAE-MDP	0.26±0.23	0.35±0.48	0.68±0.45	0.68±0.48		0.17±0.21	**0.01±0.01**	0.41±0.43	0.91±0.09
TimeGAN	0.54±0.05	**0.02±0.03**	0.68±0.45	0.60±0.43		0.24±0.24	0.34±0.48	0.41±0.43	0.73±0.26
VAE	0.38±0.27	0.30±0.33	1.00±0.01	0.60±0.43		0.29±0.00	0.26±0.36	0.66±0.47	0.94±0.11
SPAWNER	0.46±0.09	0.01±0.01	0.66±0.47	0.68±0.48		0.41±0.17	0.60±0.43	0.57±0.42	**0.20±0.16**
DGW	0.46±0.09	0.19±0.26	1.00±0.01	0.68±0.48		0.20±0.19	0.34±0.48	0.41±0.43	0.34±0.48
Permutation	0.37±0.16	0.05±0.06	0.67±0.47	0.68±0.48		0.36±0.23	0.67±0.48	0.57±0.42	0.94±0.11
Jittering	0.46±0.09	**0.01±0.01**	0.68±0.45	0.68±0.48		**0.07±0.08**	0.34±0.48	0.16±0.08	0.67±0.46
Scaling	0.50±0.10	**0.01±0.01**	0.68±0.45	0.68±0.48		0.16±0.22	**0.00±0.01**	**0.02±0.02**	0.74±0.39
TDA	0.37±0.16	**0.01±0.01**	0.68±0.45	0.68±0.48		0.18±0.14	0.34±0.48	0.41±0.43	0.52±0.41
IS	pen	door	relocate	hammer	DR	pen	door	relocate	hammer
NoAug.	0.14±0.03	0.94±0.12	**0.00±0.00**	0.05±0.04		**0.37±0.16**	**0.00±0.00**	0.68±0.44	0.34±0.48
OAT	0.23±0.24	**0.01±0.01**	**0.00±0.00**	**0.00±0.00**		**0.37±0.16**	0.09±0.05	**0.37±0.45**	**0.03±0.01**
VAE-MDP	0.19±0.13	0.39±0.45	0.67±0.47	**0.00±0.00**		**0.37±0.16**	0.00±0.01	0.68±0.44	0.20±0.15
TimeGAN	**0.01±0.00**	0.69±0.48	0.67±0.47	0.05±0.04		0.46±0.09	**0.02±0.03**	0.68±0.44	0.38±0.45
VAE	0.57±0.00	0.38±0.46	0.82±0.26	0.08±0.09		0.46±0.09	0.34±0.48	**0.37±0.45**	0.40±0.44
SPAWNER	0.44±0.19	1.03±0.00	0.40±0.41	0.20±0.15		0.46±0.09	0.34±0.48	**0.37±0.45**	0.34±0.48
DGW	0.29±0.23	1.03±0.00	0.81±0.26	1.02±0.00		**0.37±0.16**	0.67±0.48	1.00±0.01	0.35±0.47
Permutation	0.14±0.15	1.03±0.00	0.24±0.34	1.02±0.00		**0.37±0.16**	0.67±0.48	1.00±0.01	0.41±0.44
Jittering	0.17±0.21	1.03±0.00	0.17±0.20	1.02±0.00		**0.37±0.16**	0.34±0.48	1.00±0.01	0.40±0.44
Scaling	0.36±0.18	1.03±0.00	0.33±0.47	0.86±0.23		0.46±0.09	**0.00±0.00**	1.00±0.01	0.41±0.44
TDA	0.57±0.00	**0.01±0.01**	0.02±0.02	**0.01±0.01**		**0.37±0.16**	**0.00±0.00**	**0.37±0.45**	0.35±0.47
DICE	pen	door	relocate	hammer					
NoAug.	0.20±0.26	0.69±0.48	**0.30±0.43**	0.67±0.47					
OAT	0.02±0.01	**0.34±0.48**	**0.33±0.47**	**0.18±0.25**					
VAE-MDP	0.01±0.01	0.68±0.48	0.64±0.45	0.66±0.46					
TimeGAN	**0.00±0.00**	0.42±0.44	0.64±0.45	0.67±0.47					
VAE	0.02±0.01	0.76±0.36	0.58±0.42	0.40±0.41					
SPAWNER	0.06±0.05	0.42±0.44	0.66±0.46	0.67±0.47					
DGW	0.02±0.01	**0.34±0.48**	0.58±0.42	0.67±0.47					
Permutation	0.01±0.01	**0.34±0.48**	**0.31±0.30**	0.67±0.47					
Jittering	0.11±0.13	**0.34±0.48**	0.58±0.42	0.65±0.46					
Scaling	0.02±0.01	**0.34±0.48**	0.58±0.42	0.67±0.47					
TDA	0.03±0.00	**0.34±0.48**	0.68±0.45	0.67±0.47					

Table 13. Regret@1 results of OPE without and with each augmentation method in resampled Adroit `expert` environment. The data are randomly sampled from original training data as the same data points as the corresponding task in `human` environment. Results are obtained by averaging over 3 random seeds used for training at a discount factor of 0.995, with standard deviations shown after ±.

FQE	pen	door	relocate	hammer	MB	pen	door	relocate	hammer
NoAug.	**0.02±0.01**	**0.00±0.01**	0.24±0.34	**0.00±0.01**		**0.01±0.01**	**0.00±0.01**	**0.00±0.00**	0.07±0.09
OAT	0.07±0.08	**0.00±0.00**	0.30±0.43	**0.01±0.01**		**0.01±0.02**	0.09±0.05	**0.00±0.00**	**0.01±0.01**
VAE-MDP	0.10±0.06	**0.00±0.01**	**0.17±0.20**	0.13±0.09		0.06±0.08	**0.00±0.00**	0.02±0.02	0.03±0.00
TimeGAN	0.08±0.06	**0.00±0.00**	**0.17±0.20**	0.10±0.08		0.07±0.08	**0.00±0.00**	0.02±0.02	0.13±0.09
VAE	**0.01±0.02**	**0.00±0.01**	0.45±0.37	0.13±0.09		0.05±0.06	**0.00±0.01**	0.45±0.37	0.29±0.14
SPAWNER	0.12±0.06	**0.00±0.00**	0.26±0.33	0.13±0.09		0.09±0.07	0.02±0.03	0.09±0.09	0.07±0.09
DGW	0.14±0.03	**0.00±0.01**	0.95±0.03	0.16±0.16		0.06±0.08	**0.00±0.00**	0.07±0.10	0.13±0.18
Permutation	0.14±0.03	**0.00±0.01**	0.45±0.37	0.13±0.09		0.08±0.06	0.19±0.26	0.24±0.34	0.20±0.16
Jittering	0.08±0.06	**0.00±0.01**	**0.17±0.20**	0.13±0.09		**0.00±0.00**	0.19±0.26	**0.00±0.00**	0.07±0.05
Scaling	0.08±0.06	**0.00±0.01**	**0.17±0.20**	0.13±0.09		0.06±0.08	**0.00±0.00**	**0.00±0.00**	0.08±0.08
TDA	0.09±0.04	**0.00±0.01**	0.24±0.34	0.13±0.09		**0.00±0.00**	**0.00±0.00**	**0.00±0.00**	**0.01±0.02**
IS	pen	door	relocate	hammer	DR	pen	door	relocate	hammer
NoAug.	0.02±0.02	0.23±0.24	**0.00±0.00**	0.03±0.05		0.14±0.03	**0.00±0.00**	0.56±0.41	0.13±0.18
OAT	0.04±0.04	**0.01±0.01**	**0.00±0.00**	**0.00±0.00**		0.14±0.03	**0.00±0.00**	**0.02±0.02**	**0.00±0.01**
VAE-MDP	0.04±0.04	0.23±0.24	0.65±0.46	**0.00±0.00**		0.14±0.03	**0.00±0.00**	0.56±0.41	0.07±0.05
TimeGAN	**0.00±0.00**	0.37±0.26	0.65±0.46	**0.00±0.00**		**0.11±0.07**	**0.00±0.00**	0.56±0.41	0.13±0.18
VAE	0.18±0.00	0.21±0.25	0.79±0.25	0.01±0.01		0.14±0.03	**0.00±0.00**	0.27±0.32	0.13±0.18
SPAWNER	0.06±0.08	0.56±0.00	0.15±0.10	0.39±0.00		0.14±0.03	**0.00±0.00**	0.24±0.34	0.13±0.18
DGW	0.06±0.08	0.56±0.00	0.71±0.21	0.39±0.00		0.14±0.03	**0.00±0.01**	0.58±0.39	0.13±0.18
Permutation	0.05±0.03	0.56±0.00	0.15±0.21	0.39±0.00		0.14±0.03	**0.00±0.00**	**0.03±0.02**	0.13±0.18
Jittering	0.04±0.04	0.56±0.00	0.02±0.02	0.39±0.00		0.14±0.03	**0.00±0.00**	0.70±0.34	0.13±0.18
Scaling	0.13±0.07	0.56±0.00	0.15±0.21	0.27±0.17		0.14±0.03	**0.00±0.00**	0.41±0.40	0.13±0.18
TDA	0.13±0.07	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**		0.14±0.03	**0.00±0.00**	0.30±0.43	0.13±0.18
DICE	pen	door	relocate	hammer					
NoAug.	**0.00±0.00**	0.37±0.26	0.15±0.21	0.04±0.04					
OAT	**0.00±0.00**	**0.19±0.26**	0.30±0.43	**0.01±0.01**					
VAE-MDP	**0.00±0.00**	**0.19±0.26**	0.54±0.39	0.14±0.18					
TimeGAN	**0.00±0.00**	0.21±0.25	0.54±0.39	0.07±0.05					
VAE	**0.00±0.00**	**0.17±0.16**	0.17±0.20	0.08±0.08					
SPAWNER	**0.01±0.01**	**0.19±0.26**	0.56±0.41	0.05±0.04					
DGW	**0.00±0.00**	**0.19±0.26**	0.32±0.46	0.05±0.04					
Permutation	**0.00±0.00**	**0.19±0.26**	0.02±0.02	0.07±0.05					
Jittering	**0.00±0.00**	**0.19±0.26**	0.32±0.46	0.05±0.04					
Scaling	**0.00±0.00**	**0.19±0.26**	0.34±0.45	0.04±0.04					
TDA	**0.00±0.00**	**0.19±0.26**	**0.02±0.02**	0.14±0.18					

Table 14. Regret@5 results of OPE without and with each augmentation method in resampled Adroit `expert` environment. The data are randomly sampled from original training data as the same data points as the corresponding task in `human` environment. Results are obtained by averaging over 3 random seeds used for training at a discount factor of 0.995, with standard deviations shown after ±.

D. Real-World Sepsis Treatment

Sepsis, which is defined as life-threatening organ dysfunction in response to infection, is the leading cause of mortality and the most expensive condition associated with in-hospital stay (Liu et al., 2014). In particular, septic shock, which is the most advanced complication of sepsis due to severe abnormalities of circulation and/or cellular metabolism (Bone et al., 1992), reaches a mortality rate as high as 50% (Martin et al., 2003). It is critical to find an effective policy that can be followed to prevent septic shock and recover from sepsis.

D.1. Task Description

Labels. The hospital provided the EHRs over two years, including 221,700 visits with 35 static variables such as gender, age, and past medical condition, and 43 temporal variables including vital signs, lab analytes, and treatments. Our study population is patients with a suspected infection which was identified by the administration of any type of antibiotic, antiviral, antibacterial, antiparasitic, or antifungal, or a positive test result of PCR (Point of Care Rapid). On the basis of the Third International Consensus Definitions for Sepsis and Septic Shock (Singer et al., 2016), our medical experts identified septic shock as any of the following conditions are met:

- Persistent hypertension as shown through two consecutive readings ($\leq$ 30 minutes apart). Systolic Blood Pressure (SBP) $<$ 90 mmHg Mean Arterial Pressure (MAP) $<$ 65 mmHg Decrease in SBP $\geq$ 40 mmHg with an 8-hour period
- Any vasopressor administration.

From the EHRs, 3,499 septic shock positive and 81,398 negative visits were identified based on the intersection of the expert sepsis diagnostic rules and International Codes for Disease 9th division (ICD-9); the 36,122 visits with mismatched labels between the expert rule and the ICD-9 were excluded in our study. 2,205 shock visits were obtained by excluding the visits admitted with septic shock and the long-stay visits and then we did the stratified random sampling from non-shock visits, keeping the same distribution of age, gender, ethnicity, and length of hospital stay. The final data constituted 4,410 visits with an equal ratio of shock and non-shock visits.

States. To approximate patient observations, 15 sepsis-related attributes were selected based on the sepsis diagnostic rules. In our data, the average missing rate across the 15 sepsis-related attributes was 78.6%. We avoided deleting sparse attributes or resampling with a regular time interval because the attributes suggested by medical experts are critical to decision making for sepsis treatment, and the temporal missing patterns of EHRs also provide the information of patient observations. The missing values were imputed using Temporal Belief Memory (Kim & Chi, 2018) combined with missing indicators (Lipton et al., 2016).

Actions. For actions, we considered two medical treatments: antibiotic administration and oxygen assistance. Note that the two treatments can be applied simultaneously, which results in a total of four actions. Generally, the treatments are mixed in discrete and continuous action spaces according to their granularity. For example, a decision of whether a certain drug is administrated is discrete, while the dosage of drug is continuous. Continuous action space has been mainly handled by policy-based RL models such as actor-critic models (Lillicrap et al., 2015), and it is generally only available for online RL. Since we cannot search continuous action spaces while online interacting with actual patients, we focus on discrete actions. Moreover, in this work, the RL agent aims to let the physicians know when and which treatment should be given to a patient, rather than suggests an optimal amount of drugs or duration of oxygen control that requires more complex consideration.

Rewards. Two leading clinicians, both with over 20-year experience on the subject of sepsis, guided to define the reward function based on the severity of septic stages. The rewards were defined as follows: infection [-5], inflammation [-10], organ failures [-20], and septic shock [-50]. Whenever a patient was recovered from any stage of them, the positive reward for the stage was gained back.

The data was divided into 80% (the earlier 80% according to the time of the first event recorded in patients' visits) for training and (the later) 20% for test, as the most common task for OPE was using historical data to validate policies then applied selected policies for test.

Policies We estimate the behavior policy with behavior cloning as in (Fu et al., 2021; Hanna et al., 2019). The evaluation policies were trained using off-policy DQN algorithm with different hyper-parameter settings, where DQN was trained using default setting (learning rate $1e-3$, $\gamma = 0.99$), learning rate $1e-4$, learning rate $1e-5$, a different random seed, $\gamma = 0.9$, respectively.

D.2. Septic Shock Rate.

Since the RL agent cannot directly interact with patients, it only depends on offline data for both policy induction and evaluation. In similar fashion to prior studies (Komorowski et al., 2018; Azizsoltani & Jin, 2019; Raghu et al., 2017), the induced policies were evaluated using the septic shock rate. The assumption (Raghu et al., 2017) behind that is: when a septic shock prevention policy is indeed effective, the more the real treatments in a patient trajectory agree with the induced policy, the lower the chance the patient would get into septic shock; vice versa, the less the real treatments in a patient trajectory agree with the induced policy (more dissimilar), the higher the chance the patient would get into septic shock. Specifically, we measured agreement rate with the agent policy, $a \in [0, 1]$ was the number of events agreed with the agent policy among the total number of events in a visit; $a = 0$ if the actual treatments and the agent's recommendations are completely different in a visit trajectory, and $a = 1$ if they are the same. According to the agreement rate, the average septic shock rate is calculate, which is the number of shock visits among the visits with the corresponding agreement rate $\geq a$. If the agent policies are indeed effective, the more the actually executed treatments agree with the agent policy, the less likely the patient is going to have septic shock. This metric was first used in (Raghu et al., 2017).

E. Real-World Intelligent Tutoring

E.1. Task Description

Our data contains a total of 1,307 students' interaction logs with a web-based ITS collected over seven semesters' classroom studies. During the studies, all students used the same tutor, followed the same general procedure, studied the same training materials, and worked through the same training problems. All students went through the same four phases: 1) reading textbook, 2) pre-test, 3) working on the ITS, and 4) post-test. During reading textbook, students read a general description of each principle, reviewed examples, and solved some training problems to get familiar with the ITS. Then the students took a pre-test which contained a total of 14 single- and multiple-principle problems. Students were not given feedback on their answers, nor were they allowed to go back to earlier questions (so as the post-test). Next, students worked on the ITS, where they received the same 10 problems in the same order. After that, students took the 20-problem post-test, where 14 of the problems were isomorphic to the pre-test and the remainders were non-isomorphic multiple-principle problems. Tests were auto-graded following the same grading criteria. Test scores were normalized to the range of [0, 1].

States. During tutoring, there are many factors that might determine or indicate students' learning state, but many of them are not well understood by educators. Thus, to be conservative, we extract varieties of attributes that might determine or indicate student learning observations from student-system interaction logs. In sum, 142 attributes with both discrete and continuous values are extracted, which can be categorized into the following five groups:

(i) **Autonomy (10 features)**: the amount of work done by the student, such as the number of times the student restarted a problem;

(ii) **Temporal Situation (29 features)**: the time-related information about the work process, such as average time per step;

(iii) **Problem-Solving (35 features)**: information about the current problem-solving context, such as problem difficulty;

(iv) **Performance (57 features)**: information about the student's performance during problem-solving, such as percentage of correct entries;

(v) **Hints (11 features)**: information about the student's hint usage, such as the total number of hints requested.

Actions. For each problem, the ITS agent will decide whether the student should *solve* the next problem, *study* a solution provided by the tutor or *work together* with the tutor to solve on the problem. For each problem, the agent makes two levels of granularity: problem first and then step. For problem level, it first decides whether the next problem should be a worked example (WE), problem solving (PS), or a collaborative problem solving worked example (CPS). In WEs, students observe how the tutor solves a problem; in PSs, students solve the problem themselves; in CPSs, the students and the tutor co-construct the solution. If a CPS is selected, the tutor will then make step-level decisions on whether to elicit the next step from the student or to tell the solution step to the student directly.

Rewards. There was no immediate reward but the empirical evaluation matrix (i.e., delayed reward), which was the students' Normalized Learning Gain (NLG). NLG measured students' learning gain irrespective of their incoming competence. NLG is defined as: $NLG = \frac{score_{posttest} - score_{pretest}}{\sqrt{1 - score_{pretest}}}$, where 1 denotes the maximum score for both pre- and post-test that were

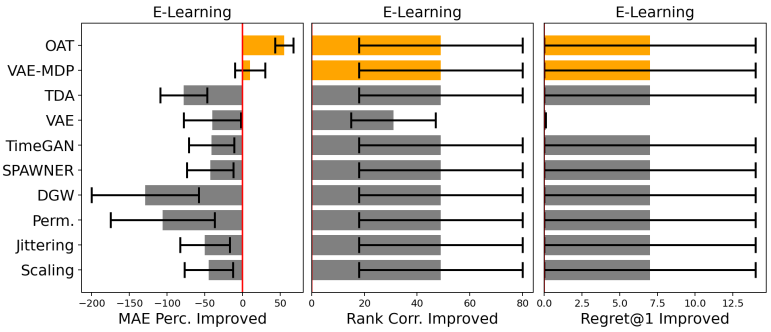

Figure 9. OPE improvement results averaging across OPE methods in e-learning.

taken before and after usage of the ITS, respectively.

Policies. The study were conducted across seven semesters, where the first six semesters' data were collected over expert policy and the seventh semester's data were collected over four different policies (three policies were RL-induced policies and one was the expert policy). The expert policy randomly picked actions. The three RL-induced policies were trained using off-policy DQN algorithm with different learning rates $lr = \{1e-3, 1e-4, 1e-5\}$.

F. More Related Works

OPE In real-world, deploying and evaluating RL policies online are high stakes in such domains, as a poor policy can be fatal to humans. It's thus crucial to propose effective OPE methods. OPE is used to evaluate the performance of a *target* policy given historical data drawn from (alternative) *behavior* policies. A variety of contemporary OPE methods has been proposed, which can be mainly divided into three categories (Voloshin et al., 2021b): (i) Inverse propensity scoring (Precup, 2000; Doroudi et al., 2017), such as Importance Sampling (IS) (Doroudi et al., 2017), to reweigh the rewards in historical data using the importance ratio between β and π. (ii) Direct methods directly estimate the value functions of the evaluation policy (Nachum et al., 2019; Uehara et al., 2020; Xie et al., 2019; Zhang et al., 2021; Yang et al., 2022), including but not limited to model-based estimators (MB) (Paduraru, 2013; Zhang et al., 2021) that train dynamics and reward models on transitions from the offline data; value-based estimators (Munos et al., 2016; Le et al., 2019) such as Fitted Q Evaluation (FQE) which is a policy evaluation counterpart to batch Q learning; minimax estimators (Liu et al., 2018; Zhang et al., 2020b; Voloshin et al., 2021a) such as DualDICE that estimates the discounted stationary distribution ratios (Yang et al., 2020a). (iii) Hybrid methods combine aspects of both inverse propensity scoring and direct methods (Jiang & Li, 2016; Thomas & Brunskill, 2016). For example, DR (Jiang & Li, 2016) leverages a direct method to decrease the variance of the unbiased estimates produced by IS. However, a major challenge of applying OPE to real world is many methods can perform unpleasant when human-collected data is highly limited as in (Fu et al., 2020; Gao et al., 2023), augmentation can be an important way to facilitate OPE performance.

Data Augmentation Data augmentation has been widely investigated in various domains, including computer vision, time series, and RL. In computer vision, images are the major target and augmentation have improved downstream models' performance (LeCun et al., 1998; Deng et al., 2009; Cubuk et al., 2019; Xie et al., 2020). However, many image-targeted methods, such as crop and rotate images, will discard important information in trajectories. In time series, a variety of data augmentation has been proposed to capture temporal and multivariate dependencies (Le Guennec et al., 2016; Kamycki et al., 2019; Yoon et al., 2019; Iwana & Uchida, 2021a). For instance, SPAWNER (Kamycki et al., 2019) and DGW (Iwana & Uchida, 2021b) augment time series by capturing group-level similarities to facilitate supervised learning. Generative models such as GAN and VAE have achieved state-of-the-art performance in time series augmentation for both supervised and unsupervised learning (Antoniou et al., 2017; Donahue et al., 2018; Yoon et al., 2019; Barak et al., 2022). However, those approaches for images and time-series do not consider the Markovian nature in OPE training data, and may not be directly applicable to MDP trajectory augmentation.