Exploring Time-Step Size in Reinforcement Learning for Sepsis Treatment

Yingchuan Sun YINGCHUAN.SUN@EMORY.EDU
Shengpu Tang SHENGPU.TANG@EMORY.EDU
Department of Computer Science, Emory University, USA

Abstract

Existing studies on reinforcement learning (RL) for sepsis management have mostly followed an established problem setup, in which patient data are aggregated into 4-hour time steps. Although concerns have been raised regarding the coarseness of this time-step size, which might distort patient dynamics and lead to suboptimal treatment policies, the extent to which this is a problem in practice remains unexplored. In this work, we conducted empirical experiments for a controlled comparison of four time-step sizes ($\Delta t = 1, 2, 4, 8$ h) on this domain, following an identical offline RL pipeline. To enable a fair comparison across time-step sizes, we designed action re-mapping methods that allow for evaluation of policies on datasets with different time-step sizes, and conducted cross-Δt model selections under two policy learning setups. Our goal was to quantify how time-step size influences state representation learning, behavior cloning, policy training, and off-policy evaluation. Our results show that performance trends across Δt vary as learning setups change, while policies learned at finer time-step sizes ($\Delta t = 1$ h and 2 h) using a static behavior policy achieve the overall best performance and stability. Our work highlights time-step size as a core design choice in offline RL for healthcare and provides evidence supporting alternatives beyond the conventional 4-hour setup.

Keywords: time step discretization, reinforcement learning, sepsis treatment, offline RL

Data and Code Availability This study uses the MIMIC-III v1.4 critical care database, which is publicly available to credentialed researchers through PhysioNet. The code for our experiments is available at https://github.com/ysun564/rl4h_timestep, which builds upon two publicly available code bases.[1],[2]

Institutional Review Board (IRB) This study does not require IRB approval.

1. Introduction

Reinforcement learning (RL) has shown great promise for sequential decision-making in healthcare, enabling data-driven treatment policies for complex medical conditions such as sepsis (Komorowski et al., 2018; Tang, 2024; Jayaraman et al., 2024). Unlike typical RL problems in which states and actions are implicitly assumed to occur at regular intervals, time series data in the electronic health record (EHR) are collected at irregular intervals. This irregularity poses significant challenges for the direct application of RL to such data.

A common workaround is to discretize irregularly sampled data into fixed-length time steps. For example, in the landmark work by Komorowski et al. (2018), patient data were aggregated into 4-hour time steps. However, it has been demonstrated that this kind of time discretization could introduce biases and obscure rapid physiological changes, negatively impacting policy learning and evaluation (Schulam and Saria, 2018). So far, this bias has been studied only in theory; nearly all work in this domain has continued to use 4 hours as the time step size and has not systematically studied the impact of other time-step sizes on the entire RL pipeline (see Table 1).

In this work, we explore the impact of using four different time-step sizes ($\Delta t = 1, 2, 4, 8$ h) in the MIMIC-III sepsis treatment task. While this may seem to be a simple change in preprocessing, we note that this has important implications for the problem formulation, the study cohort, and the definition of the action space, which pose challenges for establishing a “fair” comparison. To facilitate analysis across time-step sizes, we used the same cohort, designed normalized action spaces, and learned and evaluated treatment policies separately for each Δt,

[1] https://github.com/microsoft/mimic_sepsis
[2] https://github.com/MLD3/OfflineRL_FactoredActions

following an identical offline RL pipeline that includes latent state representation learning, behavior cloning, batch-constrained Q-learning (BCQ), hyperparameter selection, and off-policy evaluation (OPE) using weighted importance sampling (WIS) and effective sample size (ESS). To enable a "fair" comparison across Δt during OPE, we introduce a policy evaluation procedure that uses mapping strategies to transform actions across time-step sizes and evaluate policies learned at a Δt on test data that were preprocessed at a different Δt. We conducted cross-Δt model selection for policies trained under two BCQ architectures and evaluated the final selected policies. Our results show that performance trends across Δt vary across different BCQ architectures, and finer policies ($t_\pi = 1$ h and 2 h) trained under BCQ with a static behavior policy tend to exhibit overall good and stable performance. Our work highlights that time-step size is a core design choice for healthcare RL that affects problem formulation, learning and evaluation, and provides empirical evidence for adopting alternatives beyond the conventional 4-hour setup.

2. Related Work

When applying RL to ICU sepsis management, most studies discretize each admission's EHR into 4-hour time steps ($\Delta t = 4$ h), and model each interval as a single Markov decision proces (MDP) step. This commonly used design choice is popularized by the "AI Clinician" paper (Komorowski et al., 2018). In this setting, treatments administered within each 4-hour interval are aggregated into the action, and observations are mapped to a state, forming a trajectory for each admission. In Table 1 we summarize recent RL for sepsis studies. Nearly all of them adopted $\Delta t = 4$ h, inherited from Komorowski et al. (2018).

Table 1: Time-step sizes in prior work that studied RL for sepsis (see Table 8 for full description).

Paper	Δt
Raghu et al. (2017)	4 h
Komorowski et al. (2018)	4 h
Jeter et al. (2019)	4 h
Yu et al. (2019)	1 h
Tang et al. (2020)	4 h
Killian et al. (2020)	4 h
Lu et al. (2021)	1 h, 4 h
Fatemi et al. (2021)	4 h
Satija et al. (2021)	4 h
Ji et al. (2021)	4 h
Liang et al. (2023)	4 h
Choudhary et al. (2024)	4 h
Tu et al. (2025)	1 h

Whereas most of the studies adopt the 4 h setting, there are also differing viewpoints and attempts regarding the design choice. Jeter et al. (2019) criticizes the coarse discretization for potentially failing to capture rapid physiological changes, thereby providing justification for exploring alternative time-step sizes. Lu et al. (2021) found that using 1 h time steps significantly altered the learned policy, suggesting that a 4 h step might obscure important decision timing. To our knowledge, no controlled study has been conducted to compare different Δt values in otherwise identical setups.

3. Background & Problem Setup

3.1. Time Step Discretization

Suppose the patient timeline starts at an anchor time t_0 and ends at an ending time T. To convert continuous-time time-series data into a discrete-step trajectory, we discretize the timeline into non-overlapping windows of size Δt. We define the boundaries between consecutive windows

$$t_k = t_0 + k\Delta t,\ k = 0, \dots, L,$$

where $L = \lceil (T - t_0)/\Delta t \rceil$ represents the total number of time steps. The k-th time step is the half-open interval $[t_k, t_{k+1})$ for $k = 0, \dots, T-1$.

3.2. Offline RL Objective

We model sequential clinical treatments as a partially observable Markov decision process (POMDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, P, \Omega, R, \gamma)$, where:

- $\mathcal{S}$ (state space): the set of true patient states, which are latent and unobservable.
- $\mathcal{A}$ (action space): the set of possible treatments.
- $\mathcal{O}$ (observation space): the set of observable patient measurements (e.g., vitals, labs) per Δt.
- P (transition dynamics): $P(s_{k+1}|s_k, a_k)$ gives the probability of transitioning to next state s_{k+1} from state s_k after action a_k.
- Ω (observation function): $\Omega(o_k|s_k)$ gives the probability of observing o_k given the underlying state s_k. In our setting, o_k represents all clinical information recorded in the k-th time window.
- R (reward function): $R(s_k, a_k)$ is the reward obtained after taking action a_k in state s_k.
- γ (discount factor): $\gamma \in [0, 1)$ balances immediate and future rewards.

In our setting, the true transition dynamics $P(s_{k+1}|s_k, a_k)$ and the observation function $\Omega(o_k|s_k)$ are unknown. We only have access to logged transitions (o_k, a_k, o_{k+1}) from offline EHR data. We aggregate the information **within the window** $[t_k, t_{k+1})$ into o_k and use a learned encoder f to infer a compact latent state from the history of observations, $s_k = f(o_{0:k})$. This latent state s_k serves as the agent's belief state, and we assume it is a sufficient statistic of history and use it interchangeably with the true state. The treatment action executed **within the subsequent window** $[t_{k+1}, t_{k+2})$ is denoted as a_k, selected based on the state s_k following some policy $\pi(a_k|s_k)$, which leads to a reward r_k and influences the transition to the next state s_{k+1} (Tang et al., 2025). The process repeats until a terminal state s_T (e.g., discharge or death) is reached, yielding a trajectory $\tau = (s_0, a_0, r_0, \ldots, s_{T-1}, a_{T-1}, r_{T-1}, s_T)$. Given a trajectory τ with rewards $r_0, \ldots, r_{T-1}$, the discounted return is defined as $R(\tau) = \sum_{k=0}^{T-1} \gamma^k r_k$. The goal of RL is to learn an optimal policy π^* that maximizes the expected return: $\pi^* = \operatorname{argmax}_\pi \mathbb{E}_{\tau\sim\pi}[R(\tau)]$, i.e., the policy that achieves the highest expected return. In practice, we approximate π^* as π_μ by applying a learning algorithm to offline data, consisting of trajectories generated by following a behavior policy π_b.

4. Experimental Setup

To empirically study the impact of time step size on the MIMIC sepsis domain, we conducted experiments following an identical offline RL pipeline (Figure 1) to data discretized at $\Delta t \in \{1, 2, 4, 8\}$ h, including the following stages: *1. Cohort Construction and Preprocessing* → *2. State Representation Learning* → *3. Behavior Cloning* → *4. Policy Learning and Selection* → *5. Policy Evaluation.* Finally, we conducted *6. Policy Analysis* to summarize the results of our selected policy.

4.1. Cohort Construction and Preprocessing

Cohort Construction. We used the MIMIC-III v1.4 critical care database (Johnson et al., 2016), focusing on adult ICU patients who developed sepsis following the code of Subramanian and Killian (2020). For all patients, we extracted data from their first ICU stay during each hospitalization. Patient data include indicators of infection, patient demographics, and time-series data such as vitals, laboratory results, and interventions (intravenous [IV] fluids, vasopressors, and mechanical ventilation). Using the Sepsis-3 criteria (Singer et al., 2016), we identified the presumed onset of infection for each ICU stay.

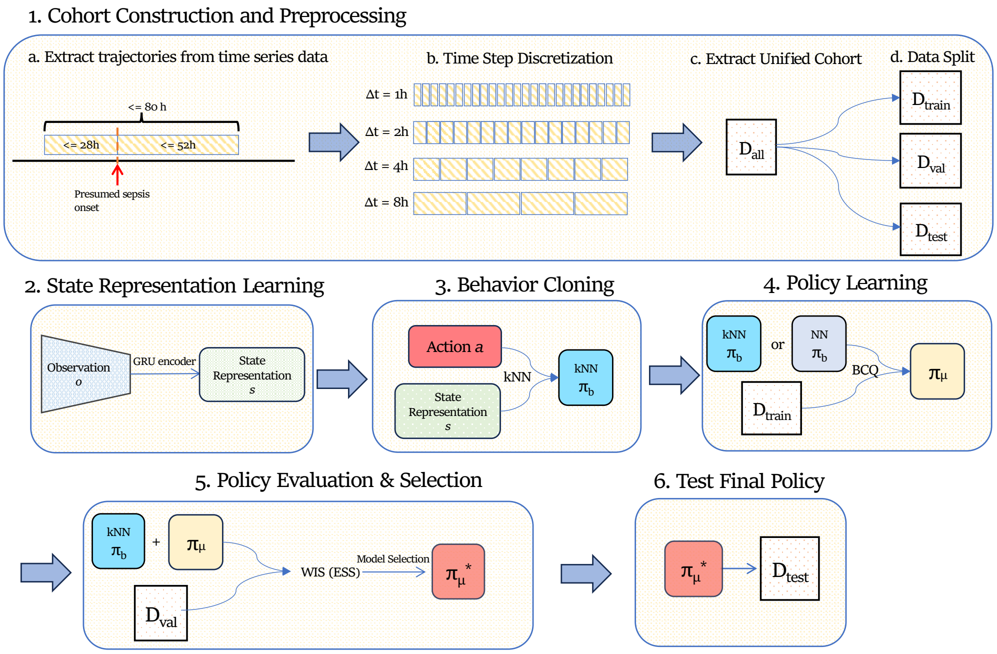


Figure 1: Overview of the offline RL pipeline.

After that, we assembled each ICU stay's time series from up to 28 hours before the first sepsis onset to up to 52 hours post-onset, yielding trajectories of up to 80 hours for each ICU admission. We discretized each extracted patient trajectory into fixed-length time windows for each time-step size $\Delta t \in \{1, 2, 4, 8\}$. To ensure valid transitions, trajectories shorter than one step at a given Δt were excluded. Following Subramanian and Killian (2020), we handled outliers, missing values and implausible measurements, and finally created a separate sepsis cohort for each Δt. Since different numbers of patients were excluded for each Δt, the resulting cohort sizes were also different. To enable a fair comparison, we defined a unified cohort consisting of ICU stays that were present across all Δt cohorts. We then split the cohort into 70/15/15% for train/validation/test.

POMDP Setup. For each ICU stay, we extracted 33 time-varying continuous features per time step, in addition to 5 static demographic and contextual features (see Table 9). Each 38-dimensional feature vector was considered an observation o, while the observation space $\mathcal{O}$ comprises the set of all o. Based on this, we learned the (approximate) state space $\mathcal{S}$ as described in Section 4.2. Following Komorowski et al. (2018), each action is defined with the interventions applied within each time step, including IV fluids and vasopressors. Specifically, the total volume of IV fluids and the maximum dose of vasopressors applied simultaneously constitute the action taken in that time step. The dosage of fluids and vasopressors was each divided into 5 levels using clinically relevant dosage boundaries (Tang et al., 2020), yielding an action space $\mathcal{A}$ with $5 \times 5 = 25$ possible actions. Notably, the dosage levels were normalized by time per Δt, resulting in a different action space for each Δt (Table 2). We used a sparse reward signal that reflects patient's terminal outcomes (Tang et al., 2020; Shi et al., 2025): a sparse reward of +100 was given for survival (at discharge or at end of trajectory) and 0 otherwise. We set $\gamma = 0.99$ in policy learning and $\gamma = 1$ in policy evaluation (Lee et al., 2025).

Table 2: Normalized action space $\mathcal{A}$ across Δt.

Level	IV fluids (mL/Δt)	Vasopressors (μg kg^{-1} min^{-1})
0	$= 0$	$= 0$
1	(0, 125Δt)	(0, 0.08)
2	[125Δt, 250Δt)	[0.08, 0.20)
3	[250Δt, 500Δt)	[0.20, 0.45)
4	$\geq$ 500Δt	$\geq$ 0.45

4.2. State Representation Learning

To address partial observability in patient trajectories, we learned a compact latent state representation with a recurrent neural network using the approximate information state (AIS) (Subramanian et al., 2022; Killian et al., 2020). The learned state representation constitutes an approximation of the state space $\mathcal{S}$ in the POMDP setup. Specifically, we trained a gated recurrent unit (GRU) encoder (Cho et al., 2014) that, at each step k, maps the observations $o_{0:k}$ up to step k and the actions $a_{0:k-1}$ taken up to step $k-1$ to a D-dimensional latent state s_k. The GRU encoder was optimized via a dual-head objective: one decoder head reconstructs the current observation vector o_k, while another head predicts the next observation o_{k+1} given the current latent state s_k and action a_k in the form of a parameterized distribution $p(o_{k+1}|s_k, a_k) = P(s_{k+1}|s_k, a_k)\Omega(o_{k+1}|s_{k+1})$. We trained the representation model on the training set trajectories to minimize the negative log-likelihood (NLL) loss and monitored the learning curve on the validation set. For each Δt, we ran an identical grid search over five latent dimension sizes and six different learning rates (see Appendix A). The checkpoint with the lowest validation NLL was selected to extract the latent states at each time step. We treat the D-dimensional latent state s_k as the AIS summarizing the patient's history up to time k.

4.3. Behavior Cloning

During policy learning and evaluation, we require access to the clinician's action probability distribution. Since the observational dataset only contained the observed deterministic actions, we estimated a stochastic behavior policy π_b from data to approximate the clinicians' non-deterministic treatment decisions. The policy takes the patient's state representation s_k as input and predicts the clinicians' action distribution $\pi_b(a|s_k)$. After considering discriminative performance and calibration, we implemented k-nearest neighbors (kNN) classifiers for behavior cloning separately for each data partition (train/validation/test) (Raghu et al., 2018). We transformed the episodic dataset into a flattened dataset of n state-action pairs (s_k, a_k), where each sample corresponds to one step in a trajectory. Each state vector s_k is treated as a feature input, and its associated action a_k is the label used for kNN classification. We then performed a hyperparameter grid search over the number of neighbors k and the distance metric (see Table 10). Best classifiers were selected based on their macro and micro averaged area under the receiver operating characteristic curve (AUROC) via 5-fold cross validation, and were used as the π_b for BCQ and OPE. We report the result of grid search and policy performance in Section 5.3.

4.4. Policy Learning

In healthcare where exploration of new treatments is infeasible, it is critical that we do not learn a policy that extrapolates to actions (more specifically, state-action combinations) not observed in the data (Gottesman et al., 2019). To address the issue, we used an offline RL algorithm, namely (discrete-action) batch-constrained Q-learning (BCQ) (Fujimoto et al., 2019). In our BCQ implementation, the Q-network is a three-layer feed-forward network that estimates $Q(s, a)$, together with a target network of identical architecture updated via Polyak averaging. At each update, the Q-network selects the action for the next state from a set generated by π_b of the behavior cloning stage, where actions whose estimated behavior probability falls below a threshold ε are masked out. The target network then evaluates the selected action when forming the bootstrapping target. We trained the Q-network with the Huber loss between the current and target values. In classic BCQ implementations, behavior policy π_b is modeled by a neural network and is typically trained concurrently with the Q-network (Liu and Brunskill, 2022; Tang et al., 2022). To ensure a consistent π_b throughout the policy training and evaluation stages, we used the static kNN-based π_b in BCQ (the resulting policies are called **kNN-policies**). For comparison, we also considered a standard neural network for π_b in BCQ training (the resulting policies are called **NN-policies**). For each Δt, we ran experiments with both BCQ architectures by training for a fixed number of epochs and conducted a grid search using five different random seeds and eight values of ε (see Appendix A), which yielded a set of learned policies π_μ.

4.5. Policy Evaluation & Selection

Off-policy Evaluation (OPE). We evaluated the performance of the learned policy using OPE, specifically weighted importance sampling (WIS). The standard WIS estimator used importance weights to reweight the returns of test trajectories under the assumption that test data were generated by the behavior policy π_b; using our learned π_b, we computed cumulative per-step importance ratios w_i for each action a_k taken by clinicians, and then took a weighted average of the observed returns G_i normalized by the sum of the importance weights across all evaluation trajectories (Eqn. (1)) (Mahmood et al., 2014).

$$w_i = \prod_{k=0}^{T_i-1} \frac{\pi_\mu(a_{i,k} \mid s_{i,k})}{\pi_b(a_{i,k} \mid s_{i,k})}, \quad G_i = \sum_{k=0}^{T_i-1} \gamma^k r_{i,k} \tag{1a}$$

$$\widehat{V}_{\text{WIS}} = \frac{\sum_{i=1}^n w_i G_i}{\sum_{i=1}^n w_i} \tag{1b}$$

To handle trajectories with different lengths, we implemented the per-horizon WIS estimator (Doroudi et al., 2018), which normalizes the cumulative importance weights separately at each time step k over trajectories that survive to k. We note that the same patient trajectory has a larger H when discretized at a smaller Δt and this directly increases estimator variance. In order to make results comparable across Δt, we truncated the cumulative importance ratios $W = \prod_{k=1}^H \rho_k$ at $W \leq 1.438^H$ (Ionides, 2008). As WIS requires the action distribution $\pi_\mu(a_k|s_k)$ of treatment policy π_μ to calculate importance ratio ρ, we applied ϵ-softening to convert deterministic greedy action recommended by the policy into stochastic policy probabilities (Tang and Wiens, 2021): $\tilde{\pi}(a|s) = (1-\varepsilon)\,\mathbf{1}\{a = a^\star\} + \varepsilon\,\hat{\pi}_b(a|s)$ where the greedy action is $a^\star = \arg\max_a Q(s, a)$. This avoids zero weights in WIS and thereby prevents ESS from becoming too small. Throughout the experiments, we use $\epsilon = 0.1$.

To quantify the reliability and variance of WIS, we also recorded the effective sample size (ESS) of WIS (Elvira et al., 2022):

$$\text{ESS} = \frac{\left(\sum_{i=1}^n w_i\right)^2}{\sum_{i=1}^n w_i^2}, \tag{2}$$

which reflects how many trajectories contribute meaningfully after weighting. Although we have two BCQ architectures that use different behavior policies π_b for training, for OPE we only present results for kNN-based π_b learned in Section 4.3 as NN-based π_b yielded low ESS.

Cross-Δt OPE. By default, the policy learned at a particular time-step size Δt will be evaluated on test datasets processed at the same Δt. This poses a challenge for directly comparing policies learned at different Δt since they would be effectively evaluated on a "different test set". To allow for a fairer comparison, we evaluated policy learned at some $\Delta t = t_\pi$ using a test dataset with a different $\Delta t = t_D$. This includes two cases, where (i) $t_\pi > t_D$ and (ii) $t_\pi < t_D$. We achieve this by defining how to map the t_π action taken by the policy to the t_D action observed in data. For simplicity, we assume that t_π and t_D are integer multiples of each other, such that either (i) $t_\pi = Mt_D$ or (ii) $Nt_\pi = t_D$ for integers M, N.

- **Mapping when $t_\pi > t_D$.** Here, each action taken by the policy spans a time interval of size t_π, which corresponds to $M = t_\pi / t_D$ intervals of size t_D. Conceptually, for each action taken by the policy at t_π, we broadcast that action over the entire interval, yielding a sequence of M identical t_D actions (see case 1 in Figure 3). Because the action space is normalized across Δt, broadcasting does not change the action index.
- **Mapping when $t_\pi < t_D$.** In this case, each t_D interval corresponds to a sequence of $N = t_D / t_\pi$ finer steps. However, as the data has been discretized at t_D hours, the intermediate states that would be observed at the N fine steps (which are t_π hours apart) are unobserved. Thus, within each t_D window, the policy can only predict an action for the first t_π interval. In this case, we repeat that action over all N fine steps and aggregate it to a single t_D action using mapping rules below (see case 2 in Figure 3). For IV fluids, we used an **expected-overlap rule**. We first compute the total fluid volume interval $[L, U]$ by summing the upper and lower bounds (Table 2) of the N fluid actions at t_π, and then identify the fluid volume interval at t_D with the largest overlap. Since fluid volume level-4 does not specify an upper bound, we used the 95^{th}-percentile empirical threshold on the training set (for each Δt). For vasopressors, we take the maximum level across the N fine steps. The indices of fluids and vasopressors combined yield a joint t_D action (Table 2).

Model Selection. For each (t_π, t_D) pair where $t_\pi, t_D \in \{1, 2, 4, 8\}$ h under two BCQ architectures, we conduct OPE via per-horizon WIS and mapping rules. In the model selection stage, we present the validation ESS–WIS Pareto frontier for candidate policies, which consists of the set of candidate policies for which no other policy simultaneously achieves both higher WIS and higher ESS. We picked the policy on each frontier that balanced both metrics, together with an ESS cutoff that constrains minimum value for ESS. This approach prevents the selection of models that achieve high WIS but with excessive variance. We select one policy for each (t_π, t_D) pair, resulting in $4^2 = 16$ policies for each BCQ architecture.

4.6. Policy Analysis

For the selected policies, we report WIS and ESS on the test set with bootstrapped confidence intervals (CIs), and present results separately for each (t_π, t_D) pair. We compared performance of policies with different t_π using dataset with the same t_D to ensure a fair comparison (i.e., using an identical test set). To eliminate variations inherent to the WIS estimator itself, we additionally report OPE results measured using fitted-Q evaluation (FQE). To complement these metrics, we further include heatmaps in Appendix B showing how the learned policies redistribute action probabilities relative to the clinician policy.

5. Results

We applied our experimental pipeline to four time-step sizes ($\Delta t = 1, 2, 4, 8$ h) under two BCQ architectures, then conducted OPE for each (t_π, t_D) pair. For each Δt we report the following: (i) cohort statistics, (ii) AIS reconstruction error across latent dimensions, (iii) BC performance, and (iv) performances of the kNN-policies and NN-policies learned by two BCQ architectures, measured by per-horizon WIS and action frequency heatmaps.

5.1. Cohort Statistics

In Appendix A, we compare the cohort sizes across Δt. The cohort sizes decrease with coarser Δt, reflecting the exclusion of trajectories shorter than one step. For all experiments, we report results on a unified cohort (Table 3) that includes trajectories present at all Δt, which contains 18,377 ICU stays with a mortality rate of 5.9%.

Table 3: Cohort statistics across Δt.

	N	% Female	Mean ICU Hours
Survivors	17,288	44.2%	45.7
Non-survivors	1,089	44.6%	61.4

5.2. State Representation Learning

Following Section 4.2, we first evaluated the quality of our AIS encoder across time-step sizes. Table 4 reports the selected latent dimension, learning rate and the resulting minimum validation mean squared error (MSE) with 95% confidence intervals (CI). The same latent dimension (128) was selected for all Δt. For the learning rate, 0.0001 was selected for $\Delta t = 8h$, whereas 0.001 was selected for the other Δt. We also

Table 4: AIS encoder (GRU) results across time-step sizes: selected latent dimension, learning rate, and minimum validation MSE with 95% confidence intervals from 1000 bootstrap samples.

Δt (h)	latent dim	learning rate	MSE [95% CI]
1	128	0.001	0.2288 [0.2181, 0.2424]
2	128	0.001	0.2678 [0.2655, 0.2702]
4	128	0.001	0.4011 [0.3940, 0.4110]
8	128	0.0001	0.4356 [0.4290, 0.4426]

Table 5: Estimated performance (Macro and Micro AUROC) of kNN behavior policy on the validation sets across time-step sizes, with 95% confidence intervals from 1000 bootstrap samples.

Δt (h)	Macro AUROC [95% CI]	Micro AUROC [95% CI]
1	0.7715 [0.7678, 0.7753]	0.9449 [0.9443, 0.9456]
2	0.8047 [0.7998, 0.8095]	0.9491 [0.9482, 0.9500]
4	0.8143 [0.8071, 0.8211]	0.9507 [0.9496, 0.9518]
8	0.7754 [0.7601, 0.7883]	0.9483 [0.9466, 0.9501]

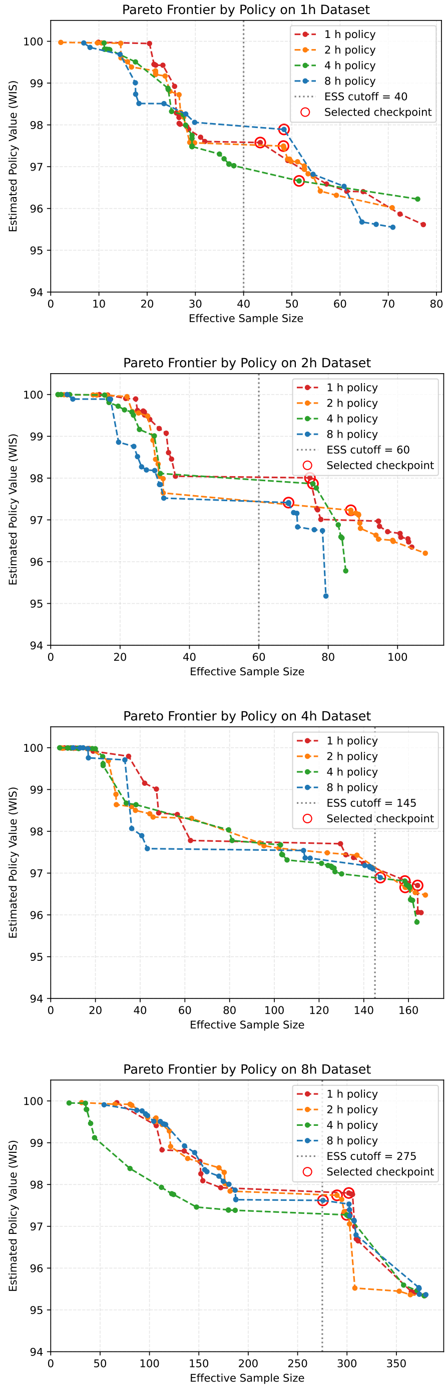
observe that validation MSE tends to increase with larger Δt. This is likely because the task is inherently more difficult for longer prediction horizons, as the AIS encoder can be seen as a forecaster that predicts future observations with a prediction horizon of Δt (e.g., average heart rate over the next time step).

5.3. Behavior Cloning

Across all settings, as k increases from 21 to $5\sqrt{n}$, the cross-validation macro- and micro-AUROC generally improve. Based on the cross-validation performance, we selected kNN classifiers for each data partition with Euclidean distance and $k = 5\sqrt{n}$ as π_b, yielding macro-AUROC > 0.75 and micro-AUROC ≈ 0.95. We summarize the performance in Table 5. While class imbalance can reduce macro-AUROC and inflate micro-AUROC, the overall performance is comparable to past work (Jeong et al., 2024).

5.4. Policy Learning, Evaluation & Selection

Figures 2 and 4 show validation ESS–WIS Pareto frontiers for candidate kNN/NN policies across $t_D \in \{1, 2, 4, 8\}$ h. For all figures, a similar trend appears: WIS is close to 100 when ESS is small, and then declines as ESS increases, reflecting the trade-off between high value and high confidence. In general, evaluation using validation data at a larger t_D led to a wider range of ESS (up to >350 for 8 h) compared to a smaller t_D (up to 80 for 1 h). This is because a


Pareto Frontier by Policy on 1h Dataset

Pareto Frontier by Policy on 2h Dataset

Pareto Frontier by Policy on 4h Dataset

Pareto Frontier by Policy on 8h Dataset

Figure 2: kNN-policies' Pareto frontiers of performance (WIS vs ESS) across evaluation t_d. Each curve corresponds to policies trained at t_π; hollow markers denote the model selected for testing; dotted lines with different colors represent the thresholds used as the boundary for model selection across Δt.

Table 6: Cross-Δt evaluation results with 95% CI for selected kNN-Policies (left) and NN-Policies (right).

		kNN-Policy						**NN-Policy**					
t_D	t_π	PHWIS		ESS		FQE		PHWIS		ESS		FQE	
		Test	CI	Test	CI	Test	CI	Test	CI	Test	CI	Test	CI
1	**1**	95.82	[93.47, 97.90]	47.74	[38.00, 59.66]	94.38	[94.69, 94.90]	95.69	[94.19, 96.83]	160.31	[136.36, 181.49]	96.38	[96.38, 96.55]
	2	96.26	[94.12, 97.98]	46.61	[35.25, 60.63]	96.87	[96.87, 97.05]	93.83$^{\downarrow}$	[90.76, 96.96]	68.25	[52.10, 86.72]	92.22$^{\downarrow}$	[92.07, 92.29]
	4	97.58	[94.03, 99.39]	25.65†	[18.03, 36.72]	94.11	[93.77, 94.03]	90.62$^{\downarrow}$	[87.40, 95.23]	30.14†	[21.96, 40.36]	96.43	[96.27, 96.46]
	8	93.33$^{\downarrow}$	[91.60, 96.42]	49.61	[37.34, 61.57]	93.38$^{\downarrow}$	[92.57, 92.79]	96.35	[93.75, 98.92]	15.99†	[9.29, 23.77]	96.19	[95.74, 95.91]
2	1	93.48$^{\downarrow}$	[91.07, 95.77]	73.76	[60.81, 87.40]	95.30	[95.11, 95.43]	93.45$^{\downarrow}$	[90.82, 98.77]	21.07†	[13.28, 29.43]	94.91	[94.69, 94.98]
	2	95.16	[92.64, 97.11]	81.45	[65.75, 95.39]	95.43	[94.98, 95.30]	93.71$^{\downarrow}$	[91.28, 96.28]	84.51	[71.47, 99.00]	95.77	[95.17, 95.46]
	4	91.35$^{\downarrow}$	[85.93, 95.94]	45.07†	[34.37, 56.14]	95.43	[95.03, 95.30]	91.14$^{\downarrow}$	[87.09, 96.06]	21.83†	[14.16, 32.26]	95.47	[95.58, 95.84]
	8	91.80$^{\downarrow}$	[88.74, 95.36]	80.92	[64.06, 93.96]	95.16	[95.36, 95.60]	96.88	[91.58, 99.89]	2.19†	[1.07, 5.28]	94.55	[94.37, 94.64]
4	1	94.87	[92.66, 96.90]	163.28	[146.77, 184.35]	95.30	[95.40, 95.66]	96.44	[92.14, 99.48]	16.05†	[12.47, 24.27]	93.13$^{\downarrow}$	[93.04, 93.35]
	2	94.87	[92.65, 96.80]	163.28	[143.44, 183.71]	95.30	[95.37, 95.61]	95.43	[92.45, 98.33]	62.35	[52.69, 76.50]	92.91$^{\downarrow}$	[92.86, 93.10]
	4	94.92	[92.53, 96.73]	161.81	[145.37, 181.07]	95.30	[95.37, 95.66]	94.01$^{\downarrow}$	[91.23, 97.15]	66.61	[56.10, 77.72]	94.15	[93.69, 94.10]
	8	94.37	[92.02, 96.30]	144.43	[127.35, 163.27]	95.30	[95.40, 95.69]	96.48	[90.71, 99.72]	18.72†	[13.46, 25.35]	94.66	[94.56, 94.87]
8	1	94.12	[92.16, 95.70]	340.91	[320.00, 367.78]	96.66	[96.49, 96.97]	93.55$^{\downarrow}$	[88.41, 98.54]	48.78†	[40.66, 60.13]	92.59$^{\downarrow}$	[92.40, 92.69]
	2	94.13	[92.48, 96.00]	326.67	[304.63, 353.73]	96.66	[96.47, 96.87]	91.71$^{\downarrow}$	[87.74, 95.80]	135.41†	[123.96, 151.25]	96.06	[95.93, 96.35]
	4	92.66$^{\downarrow}$	[90.72, 94.78]	311.25	[287.34, 330.94]	96.66	[96.48, 96.93]	97.08	[95.39, 98.58]	213.58	[190.51, 232.46]	97.78	[97.62, 98.00]
	8	94.06$^{\downarrow}$	[92.00, 95.71]	305.91	[288.45, 326.83]	96.66	[96.47, 96.88]	94.12	[92.43, 95.60]	431.44	[405.32, 454.87]	95.32	[94.93, 95.35]

$^{\downarrow}$ Performance lower than the clinician baseline (94.09). † Low ESS relative to the corresponding ESS cutoff.

finer t_D yields a POMDP with more decision points, giving the evaluation policy more opportunities to diverge from the behavior policy and thereby increasing the variance of the importance weights.

All Pareto figures of the kNN-policies (Figure 2) show a visually similar shape, where the curves for $t_\pi \in \{1, 2, 4, 8\}$ h exhibit substantial overlap, and no single t_π policy curve consistently dominates the others. All curves achieve validation WIS results higher than that of the clinicians (94.09). For model selection, we selected a different ESS cutoff of 40/60/145/275 respectively for each $t_D \in \{1, 2, 4, 8\}$ h, and chose the policy that achieved the highest WIS and has an ESS $\geq$ the cutoff.

The Pareto figures of the NN-policies (Figure 4) noticeably differ in shape from those of the kNN-policies. When the policies are learned at the same time-step size as validation data ($t_\pi = t_D$), the corresponding curves generally have the widest ESS range compared to policies learned at a different Δt. We set a ESS cutoff of 75/70/60/300 respectively for each $t_D \in \{1, 2, 4, 8\}$ h to ensure comparable variance of the selected policies learned at different t_π.

5.5. Test Performance

Table 6 summarizes final policy performance on the test set, while Figures 5 and 6 show action frequency heatmaps of the final kNN and NN policies. We summarize the key findings below.

How do performance trends differ for kNN- vs. NN-policies? Overall, we find that the kNN-policies are more stable than NN-policies. The ESS values of kNN-policies are similar within each t_D and almost always exceed the corresponding t_D-specific ESS cutoff with rare exceptions (2 out of 16 total cases). In contrast, the ESS for NN-policies drops below the cutoff in 9 cases. Interestingly, the highest ESS of NN-policies for each t_D always occurs when $t_\pi = t_D$. For example, when $t_\pi = t_D =$ 1h, the NN-policy achieves the highest ESS of 160.31. In terms of policy value as measured by PHWIS and FQE, kNN-policies exhibit fewer cases than NN-policies (7 vs. 12 out of 32 cases) where the policy value is lower than the clinician baseline, indicating overall better performance. In summary, kNN-policies tend to have generally better performance and stability than NN-policies. Therefore, we focus our subsequent results on kNN-policies.

Which t_π has the best performance? When t_D = 1 h or 2 h, the finer policies (t_π = 1, 2 h) tend to perform better; they achieve better WIS and FQE scores than the t_π = 8 h policy with a comparable ESS. This may be because policies learned at a coarser timescale struggle to accurately predict actions at a more granular timescale, leading to higher variance (as seen for t_π = 4 h) and degraded performance (as for t_π = 8 h). Among policies learned at a finer timescale, the t_π = 2 h policy achieves higher values in both WIS and FQE than the t_π = 1 h policy, possibly reflecting a balance between time granularity and performance stability.

When $t_D = 2$ h and 4 h, the finer policies ($t_\pi = 1, 2$ h) tend to perform similarly with coarser ones. With $t_\pi = 1$ h or 2 h, the kNN-policies at $t_D = 4$ h show WIS and ESS values close to those of the $t_\pi = 4$ h policy, with identical FQE scores, indicating similar policy behaviors after the cross-Δt mapping. This is also reflected in the corresponding heatmaps (see Figure 5). At $t_D = 8$ h, the $t_\pi = 1$ h and 2 h kNN-policies show slightly higher WIS and ESS values than the $t_\pi = 4$ h policy, while their FQE scores remain the same.

Overall, we found that kNN-policies learned at finer time-step sizes ($\Delta t = 1, 2$ h) were the most stable and performant across the evaluation settings.

6. Conclusion & Discussion

While nearly all prior work on RL for sepsis has universally followed the AI Clinician (Komorowski et al., 2018) with 4 h time steps, this work presents, to our knowledge, the first systematic comparison across four different time-step sizes (1, 2, 4, 8 h) under two policy training setup (BCQ with kNN and NN behavior cloning) using an identical offline RL pipeline that includes preprocessing, representation learning, behavior cloning, policy learning, and off-policy evaluation. To enable a fair comparison of different time step sizes beyond simply altering preprocessing, we controlled for several aspects of our pipeline. We used the same cohort and a fixed set split across all settings, preventing the impact of data differences on training and evaluation. We designed a normalized action space for each Δt to facilitate comparison. We conducted AIS grid searches and selected the best latent dimension independently for each Δt. We tuned BC policies, selecting different k for each Δt to ensure downstream stability. For OPE, we clipped importance ratios based on each trajectory's horizon, thereby mitigating effects arising solely from differing trajectory lengths resulting from differing time-step sizes. Together, these choices provide a robust reference for future studies that intend to conduct similar investigations on this domain.

Furthermore, we introduced a method for cross-Δt policy evaluation. We developed action mappings for IV fluids and vasopressors based on their definitions and conducted model selection on all t_π policies across all t_D datasets. Our results show that finer policies ($t_\pi = 1, 2$ h) trained under BCQ with a kNN behavior policy tend to exhibit overall good and stable performance. These findings underscore that time-step size fundamentally changes the task, thus shaping learned policies and the evaluation process.

Still, our work has several limitations and challenges. During cohort construction stage, in order to build a unified cohort for direct comparison across Δt, we only included admissions that are present in *all* four initial cohorts (Table 7 vs. Table 3). This reduced the amount of data available in the experiment and might introduce selection bias since we only kept trajectories that span at least 8 hours, which may in turn affect the performance of the learned policies. In the behavior cloning stage, there is no widely accepted standard for measuring the quality of the estimated behavior policy. Although we followed prior studies and used AUROC as the metric for hyperparameter selection during behavior cloning, future work should explore how other behavior cloning method and hyperparameter selection metrics may influence policy learning and evaluation. In the evaluation stage, we introduced a cross-Δt evaluation method. However, this approach is based on our current action space design and may not directly be applicable to a more general setting. In future work, we plan to investigate alternative mapping strategies that apply more broadly to different types of action spaces. In addition, the choice of the ESS cutoffs in our model selection stage was manually determined, as we currently lack a standardized criterion for specific Δt. This process might introduce human bias and skew the results. Future work should explore and design fairer model selection methods both theoretically and empirically. There also remain challenges in interpreting the final results. Although many of our learned policies outperform the clinician baseline on the test set in terms of the evaluation metrics, the heatmap action distributions differ substantially from those of clinicians. This discrepancy may limit the clinical utility of the learned policies. While wes believe that having a robust understanding of technical differences across Δt is a crucial step before potential real-life use, future work should aim to strengthen the clinical validation of the approach.

Our results demonstrate that time-step size is a crucial design choice for clinical RL tasks that can substantially shape the learned policies. Our work advocates for careful reconsideration from the community of different time-step sizes in sepsis management beyond the conventional 4 h setup, in order to learn better policies and enable fairer evaluation across time-step sizes.

Acknowledgments

This work was supported in part by computing resources provided by the Department of Computer Science at Emory University. The authors thank members of the TAIL group (Aayam Kc, Yuxuan Shi, Matthew Lafrance, Sixing Wu, Canyu Cheng, Hanqi Chen, Yilang Ding, Kaixuan Liu, Tina Piltner, Owen Tucker, Olin Gilster, Zixin Lin) for helpful discussions regarding this work. We also appreciate the constructive feedback from the anonymous reviewers of ML4H 2025, NeurIPS 2025 TS4H workshop and ARLET workshop, and RLC 2025 RL4RS workshop.

References

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179/.

Kartik Choudhary, Dhawal Gupta, and Philip S. Thomas. ICU-Sepsis: A benchmark MDP built from real medical data. *Reinforcement Learning Journal*, 4:1546–1566, 2024.

Shayan Doroudi, Philip S. Thomas, and Emma Brunskill. Importance sampling for fair policy selection. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 5239–5243. AAAI Press, 2018. ISBN 9780999241127.

Víctor Elvira, Luca Martino, and Christian P Robert. Rethinking the effective sample size. *International Statistical Review*, 90(3):525–550, 2022.

Mehdi Fatemi, Taylor W. Killian, Jayakumar Subramanian, and Marzyeh Ghassemi. Medical dead-ends and learning to identify high-risk states and treatments. In *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=4CRpaV4pYp.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2052–2062. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/fujimoto19a.html.

Omer Gottesman, Fredrik Johansson, Matthieu Komorowski, Aldo Faisal, David Sontag, Finale Doshi-Velez, and Leo Celi. Guidelines for reinforcement learning in healthcare. *Nature Medicine*, 25, 01 2019. doi: 10.1038/s41591-018-0310-5.

Edward L. Ionides. Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311, 2008. ISSN 10618600. URL http://www.jstor.org/stable/27594308.

Pushkala Jayaraman, Jacob Desman, Moein Sabounchi, Girish N Nadkarni, and Ankit Sakhuja. A primer on reinforcement learning in medicine for clinicians. *npj Digital Medicine*, 7(1): 337, 2024.

Hyewon Jeong, Siddharth Nayak, Taylor Killian, and Sanjat Kanjilal. Identifying differential patient care through inverse intent inference, 2024. URL https://arxiv.org/abs/2411.07372.

Russell Jeter, Christopher Josef, Supreeth Shashikumar, and Shamim Nemati. Does the "Artificial Intelligence Clinician" learn optimal treatment strategies for sepsis in intensive care? *arXiv preprint arXiv:1902.03271*, 2019. URL https://arxiv.org/abs/1902.03271.

Christina X Ji, Michael Oberst, Sanjat Kanjilal, and David Sontag. Trajectory inspection: A method for iterative clinician-driven design of reinforcement learning studies. *AMIA Summits on Translational Science Proceedings*, 2021:305, 2021.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Liwei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3: 160035, 2016.

Taylor W. Killian, Haoran Zhang, Jayakumar Subramanian, Mehdi Fatemi, and Marzyeh Ghassemi. An empirical study of representation learning for reinforcement learning in healthcare. In Emily Alsentzer, Matthew B. A. McDermott, Fabian Falck, Suproteem K. Sarkar, Subhrajit Roy, and Stephanie L. Hyland, editors, *Proceedings of the Machine Learning for Health NeurIPS Workshop*, volume 136 of *Proceedings of Machine Learning Research*, pages 139–160. PMLR, 11 Dec 2020. URL https://proceedings.mlr.press/v136/killian20a.html.

Matthieu Komorowski, Leo A Celi, Omar Badawi, Anthony C Gordon, and A Aldo Faisal. The Artificial Intelligence Clinician learns optimal treatment strategies for sepsis in intensive care. *Nature Medicine*, 24(11):1716–1720, 2018. URL https://doi.org/10.1038/s41591-018-0213-5.

Jung Min Lee, Shengpu Tang, Michael Sjoding, and Jenna Wiens. Optimizing loop diuretic treatment for mortality reduction in patients with acute dyspnea using a practical offline reinforcement learning pipeline for health care: Retrospective single-center simulation study. *JMIR Medical Informatics*, 13:e69145, 2025.

Dayang Liang, Huiyi Deng, and Yunlong Liu. The treatment of sepsis: an episodic memory-assisted deep reinforcement learning approach. *Applied Intelligence*, 53(9):11034–11044, 2023.

Yao Liu and Emma Brunskill. Avoiding overfitting to the importance weights in offline policy optimization, 2022. URL https://openreview.net/forum?id=dLTXoSIcrik.

Mingyu Lu, Zachary Shahn, Daby Sow, Finale Doshi-Velez, and Li-wei Lehman. Is deep reinforcement learning ready for practical applications in healthcare? a sensitivity analysis of duel-DDQN for hemodynamic management in sepsis patients. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, 2020:773–782, 01 2021.

A. Rupam Mahmood, Hado van Hasselt, and Richard S. Sutton. Weighted importance sampling for off-policy learning with linear function approximation. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3014–3022, Cambridge, MA, USA, 2014. MIT Press.

Aniruddh Raghu, Matthieu Komorowski, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. Continuous state-space models for optimal sepsis treatment: a deep reinforcement learning approach. In Finale Doshi-Velez, Jim Fackler, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 2nd Machine Learning for Healthcare Conference*, volume 68 of *Proceedings of Machine Learning Research*, pages 147–163. PMLR, 18–19 Aug 2017. URL https://proceedings.mlr.press/v68/raghu17a.html.

Aniruddh Raghu, Omer Gottesman, Yao Liu, Matthieu Komorowski, Aldo Faisal, Finale Doshi-Velez, and Emma Brunskill. Behaviour policy estimation in off-policy policy evaluation: Calibration matters. *arXiv preprint arXiv:1807.01066*, 2018.

Harsh Satija, Philip S Thomas, Joelle Pineau, and Romain Laroche. Multi-objective SPIBB: Seldonian offline policy improvement with safety constraints in finite MDPs. *Advances in Neural Information Processing Systems*, 34:2004–2017, 2021.

Peter Schulam and Suchi Saria. Discretizing logged interaction data biases learning for decision-making, 2018. URL https://arxiv.org/abs/1810.03025.

Yuxuan Shi, Matthew Lafrance, and Shengpu Tang. Between life and death: Examining sparse reward designs in healthcare RL. In *RLC 2025 Workshop on Practical Insights into Reinforcement Learning for Real Systems*, 2025. URL https://openreview.net/forum?id=B8TLToCmfi.

Mervyn Singer, Clifford S Deutschman, Christopher W Seymour, Manu Shankar-Hari, Djillali Annane, Michael Bauer, Rinaldo Bellomo, Gordon R Bernard, Jean-Daniel Chiche, Craig M Coopersmith, et al. The third international consensus definitions for sepsis and septic shock (Sepsis-3). *JAMA*, 315(8):801–810, 2016. doi: 10.1001/jama.2016.0287.

Jayakumar Subramanian and Taylor Killian. Sepsis cohort from MIMIC dataset. https://github.com/microsoft/mimic_sepsis, 2020. Accessed: 2025-05-22.

Jayakumar Subramanian, Amit Sinha, Raihan Seraj, and Aditya Mahajan. Approximate information state for approximate planning and reinforcement learning in partially observed systems. *Journal* *of Machine Learning Research*, 23(12):1–83, 2022. URL http://jmlr.org/papers/v23/20-1165.html.

Shengpu Tang. *Towards Clinically Applicable Reinforcement Learning*. PhD thesis, University of Michigan, 2024.

Shengpu Tang and Jenna Wiens. Model selection for offline reinforcement learning: Practical considerations for healthcare settings. In *Machine Learning for Healthcare Conference*, pages 2–35. PMLR, 2021. URL https://proceedings.mlr.press/v149/tang21a.

Shengpu Tang, Aditya Modi, Michael Sjoding, and Jenna Wiens. Clinician-in-the-loop decision making: Reinforcement learning with near-optimal set-valued policies. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9387–9396. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/tang20c.html.

Shengpu Tang, Maggie Makar, Michael W. Sjoding, Finale Doshi-Velez, and Jenna Wiens. Leveraging factored action spaces for efficient offline reinforcement learning in healthcare. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Shengpu Tang, Jiayu Yao, Jenna Wiens, and Sonali Parbhoo. Off by a beat: Temporal misalignment in offline RL for healthcare. In *RLC 2025 Workshop on Practical Insights into Reinforcement Learning for Real Systems*, 2025. URL https://openreview.net/forum?id=yRMY2a1rjR.

Rui Tu, Zhipeng Luo, Chuanliang Pan, Zhong Wang, Jie Su, Yu Zhang, and Yifan Wang. Offline safe reinforcement learning for sepsis treatment: Tackling variable-length episodes with sparse rewards. *Human-Centric Intelligent Systems*, 5(1): 63–76, 2025.

Chao Yu, Guoqi Ren, and Jiming Liu. Deep inverse reinforcement learning for sepsis treatment. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 1–3, 2019. doi: 10.1109/ICHI.2019.8904645.

Appendix A. Extended Methods

Table 7: Extracted cohort size of MIMIC-Sepsis at different time steps.

Δt (h)	Cohort Size
1	18,995
2	18,987
4	18,906
8	18,783

Table 8: RL studies for sepsis care, summarizing time-step choices and key design aspects.

Paper	**Δt**	**Algorithm**	**Dataset**	**Cohort**	**Notes**
Raghu et al. (2017)	4 h	Dueling DDQN	MIMIC-III	17.9k	Continuous state; 5×5 IV/vaso bins; first DL-RL policy (–3.6 % mortality).
Komorowski et al. (2018)	4 h	Batch Q-learning	MIMIC-III (+eRI*)	17.1k	AI Clinician; 750 states, 25 actions; external validation.
Jeter et al. (2019)	4 h	Reproduction study	MIMIC-III	5.4k	Finds no-action policy often rivals AI Clinician; urges caution.
Yu et al. (2019)	1 h	Deep IRL	MIMIC-III	14.0k	Learns reward; highlights mortality factors (e.g. PaO_2).
Tang et al. (2020)	4 h	Set-valued DQN	MIMIC-III	20.9k	Returns top-k near-optimal dose sets for clinician choice.
Killian et al. (2020)	4 h	Offline DQN	MIMIC-III	17.9k	Sequential latent encodings outperform raw features.
Lu et al. (2021)	1 h, 4 h	Dueling DDQN	MIMIC-III	17k+	Sensitivity study on features, reward, time discretization.
Fatemi et al. (2021)	4 h	Dead-end discovery	MIMIC-III	17k+	Identifies high-risk states; secures policy to avoid them.
Satija et al. (2021)	4 h	MO-SPIBB	MIMIC-III	17k+	Safe policy improvement under performance constraints.
Ji et al. (2021)	4 h	Trajectory inspection	MIMIC-III	17k+	Clinician “what-if” review reveals policy flaws; validation tool.
Liang et al. (2023)	4 h	Episodic-memory DQN	MIMIC-III	17.9k	Memory module boosts sample efficiency, lowers est. mortality.
Choudhary et al. (2024)	4 h	Tabular MDP	MIMIC-III	∼18k	ICU-Sepsis benchmark: 715 states, 25 actions.
Tu et al. (2025)	1 h	CQL (offline)	MIMIC-III	14.0k	Safety-aware CQL with dense rewards for variable-length stays.

*eRI: Philips eICU Research Institute cohort for external validation; DDQN: Double Deep Q-Network; DQN: Deep Q-Network; IRL: Inverse Reinforcement Learning; CQL: Conservative Q-Learning; MO-SPIBB: Multi-Objective Safe Policy Improvement with Baseline Bootstrapping.

Table 9: Observed features extracted from the MIMIC-III database. The upper panel lists the 33-dimensional time-varying continuous variables fed to the GRU encoder, following the default code configuration. The lower panel lists the 5 static demographic / contextual variables appended to each trajectory.

33-d Time-varying continuous features

Glasgow Coma Scale	Heart Rate	Sys. BP
Dia. BP	Mean BP	Respiratory Rate
Body Temp (°C)	FiO_2	Potassium
Sodium	Chloride	Glucose
INR	Magnesium	Calcium
Hemoglobin	White Blood Cells	Platelets
PTT	PT	Arterial pH
Lactate	PaO_2	$PaCO_2$
PaO_2/FiO_2	Bicarbonate (HCO_3)	SpO_2
BUN	Creatinine	SGOT
SGPT	Bilirubin	Base Excess

5-d Demographic and contextual features

Age • Gender • Weight • Ventilation Status • Re-admission Status

Table 10: Hyperparameter values used for training GRU encoder and BCQ models.

Hyperparameter	**Searched Settings**
RNN:	
– Embedding dimension, d_S	$\{8, 16, 32, 64, 128\}$
– Learning rate	$\{1\times10^{-5}, 3\times10^{-5}, 1\times10^{-4}, 3\times10^{-4}, 5\times10^{-4}, 1\times10^{-3}\}$
kNN:	
– Number of neighbors, k	$k_i = \exp\left(\ln 21 + \frac{i}{7}(\ln(5\sqrt{n}) - \ln 21)\right)$[a]
– Distance metric	{Euclidean, Manhattan}
BCQ (with 5 random restarts):	
– Threshold, ε	$\{0, 0.01, 0.05, 0.1, 0.3, 0.5, 0.75, 0.999\}$
– Learning rate	3×10^{-4}
– Weight decay	1×10^{-3}
– Hidden layer size	256

[a] $i = 0, 1, \ldots, 7$. n denotes the size of the flattened dataset.

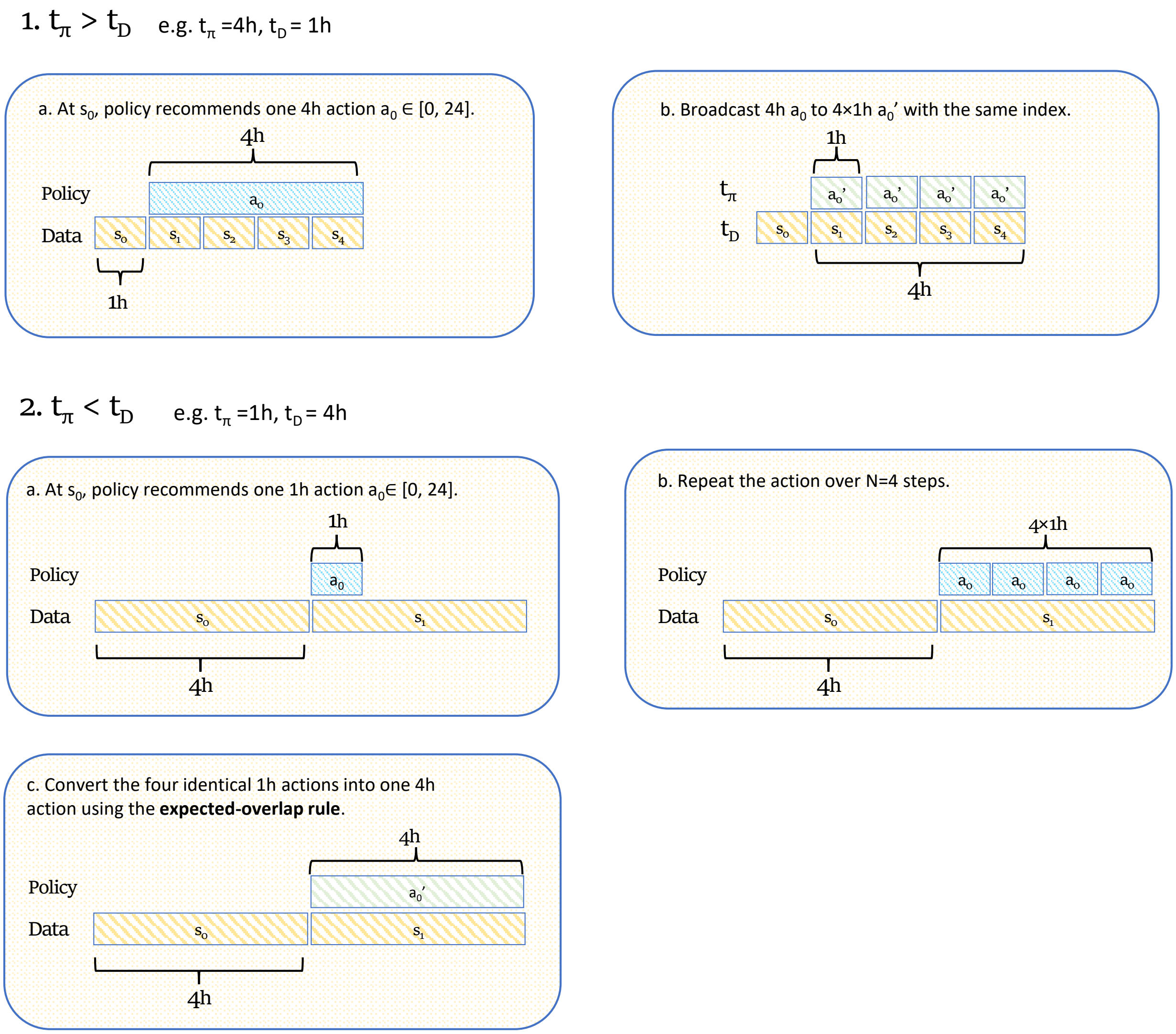
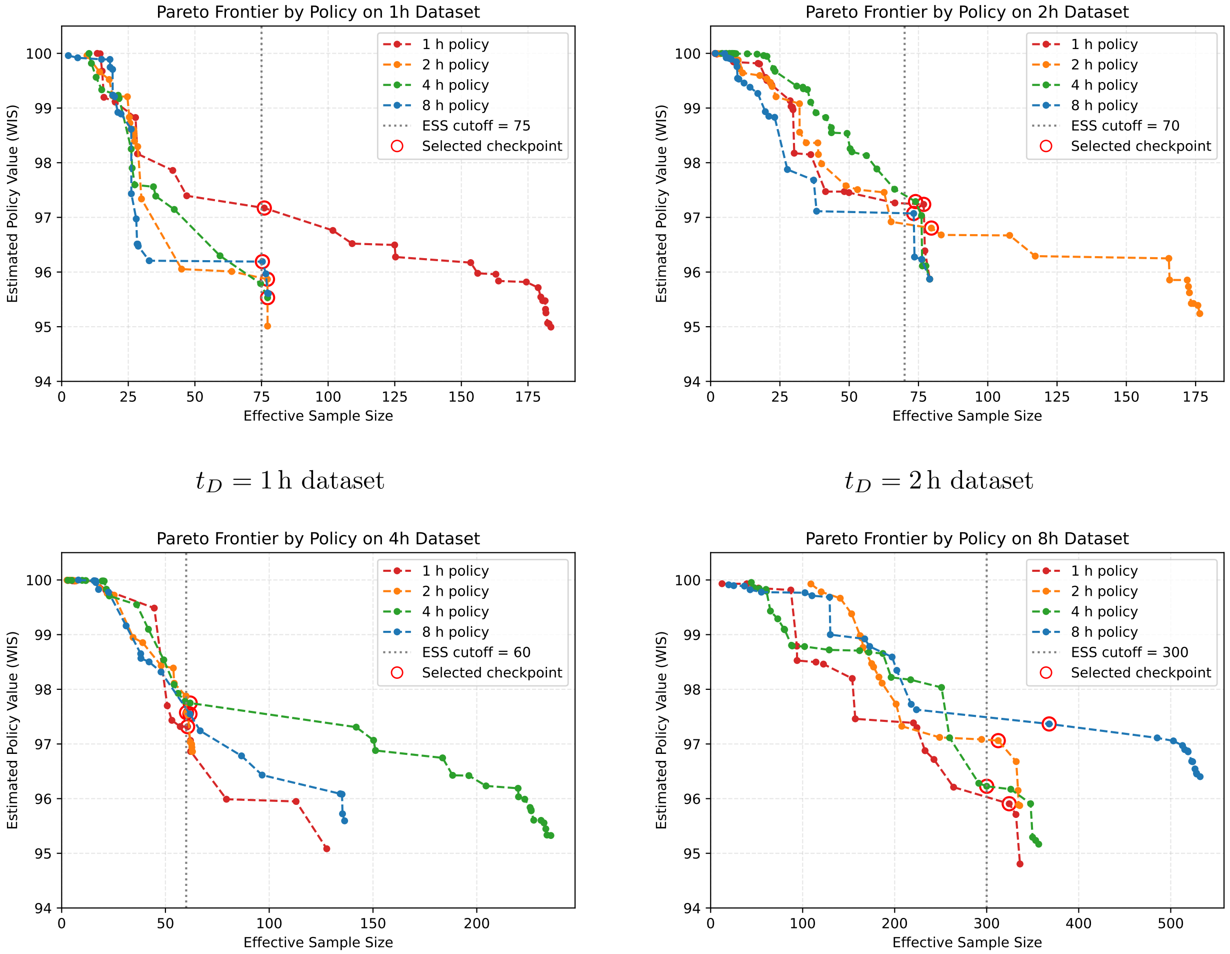

1. $t_\pi > t_D$ e.g. t_π =4h, t_D = 1h
a. At s_0, policy recommends one 4h action $a_0 \in [0, 24]$.
4h
Policy
a_0
Data
s_0
s_1
s_2
s_3
s_4
1h
b. Broadcast 4h a_0 to 4×1h a_0' with the same index.
1h
t_π
a_0'
a_0'
a_0'
a_0'
t_D
s_0
s_1
s_2
s_3
s_4
4h
2. $t_\pi < t_D$ e.g. t_π =1h, t_D = 4h
a. At s_0, policy recommends one 1h action $a_0 \in [0, 24]$.
1h
Policy
a_0
Data
s_0
s_1
4h
b. Repeat the action over N=4 steps.
4×1h
Policy
a_0
a_0
a_0
a_0
Data
s_0
s_1
4h
c. Convert the four identical 1h actions into one 4h action using the **expected-overlap rule**.
4h
Policy
a_0'
Data
s_0
s_1
4h

Figure 3: The illustration of our cross-Δt mapping.

Appendix B. Extended Results



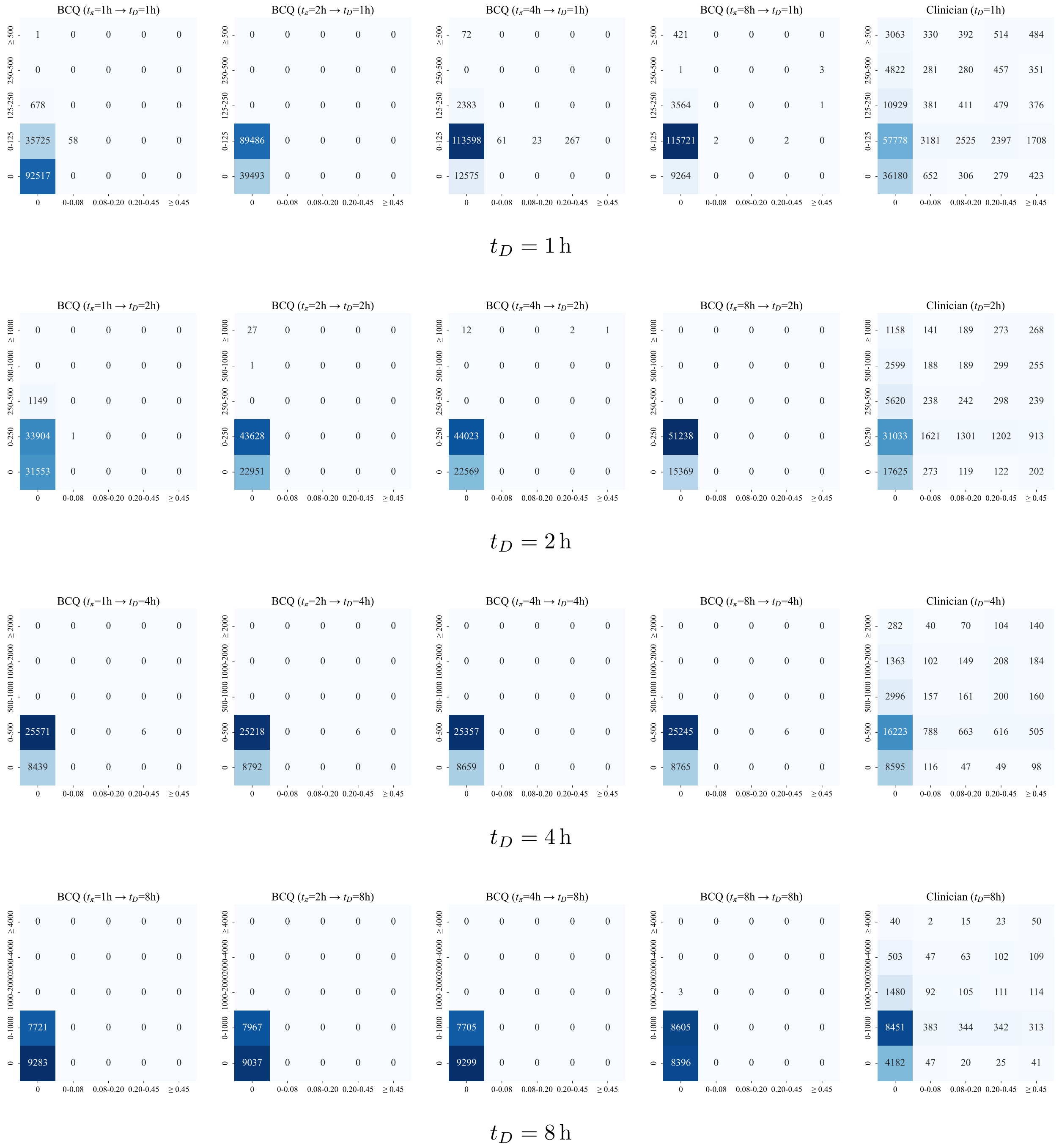
$t_D = 1\,\mathrm{h}$ dataset

$t_D = 2\,\mathrm{h}$ dataset

$t_D = 4\,\mathrm{h}$ dataset

$t_D = 8\,\mathrm{h}$ dataset

Figure 4: NN-policies' pareto frontiers of performance (WIS vs. ESS) across evaluation time steps t_D. Each curve corresponds to a policy trained at a specific t_π; hollow markers denote the model selected for testing; dotted lines with different colors represents the thresholds used as the boundary for model selection across Δt.



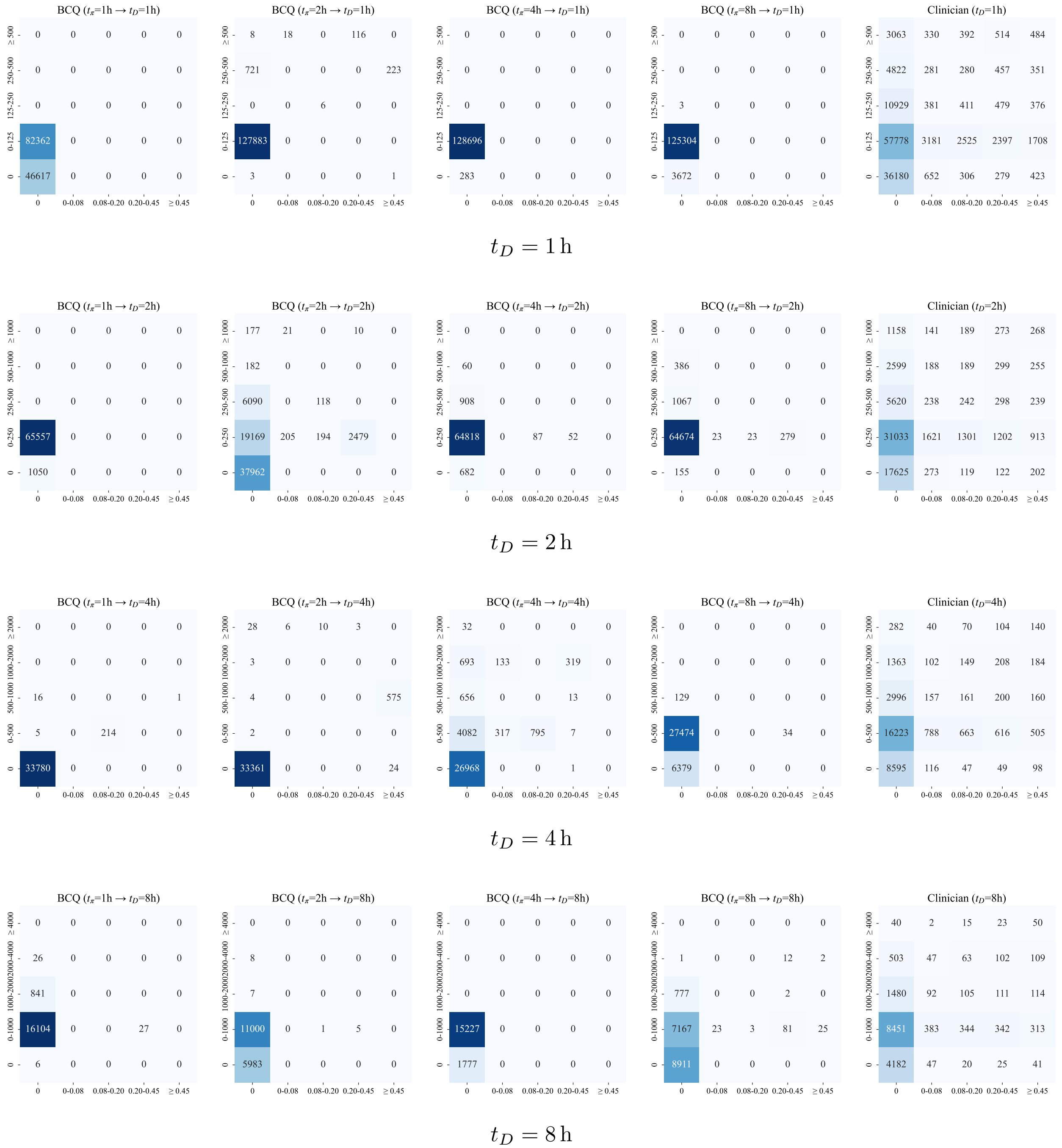
$t_D = 1\,\text{h}$

$t_D = 2\,\text{h}$

$t_D = 4\,\text{h}$

$t_D = 8\,\text{h}$

Figure 5: kNN policies' frequency heatmap of IV fluids (y-axis; mL) and vasopressors (x-axis; μg kg^{-1} min^{-1}) doses on validation set. The columns from left to right represent respectively: Policies at $t_\pi \in \{1, 2, 4, 8\}$ h, clinician policy. Darker cells indicate more frequent selections. Almost all policies most frequently select actions with zero vasopressor and low IV fluids doses.



$t_D = 1\,\text{h}$

$t_D = 2\,\text{h}$

$t_D = 4\,\text{h}$

$t_D = 8\,\text{h}$

Figure 6: NN policies' frequency heatmap of IV fluids (y-axis; mL) and vasopressors (x-axis; $\mu\text{g}\,\text{kg}^{-1}\,\text{min}^{-1}$) doses on validation set. The columns from left to right represent respectively: policies at $t_\pi \in \{1, 2, 4, 8\}$ h, clinician policy. Darker cells indicate more frequent selections. Almost all policies most frequently select actions with zero vasopressor and low IV fluids doses.