Offline Policy Optimization with Eligible Actions Supplementary Material

Yao Liu[*1] **Yannis Flet-Berliac**[2] **Emma Brunskill**[2]

[1]ByteDance, yao.liu.chn@gmail.com
[2]Stanford University, {yfletberliac, ebrun}@cs.stanford.edu

[*]This work is mostly done when the author was at Stanford.

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).

We first briefly describe the structure of the Appendix here. In Appendix 1 we add two more examples in the multi-step settings as supplementary to the example in Section **??**. In Appendix 2 we provide the proofs of theorems in Section **??**. In Appendix 3, we include more experiment details. In Appendices 4, 5, 6 and 7 we include more results in the considered domains including experiments with estimating the behavior policy with function approximation and experiments with an alternative policy selection procedure with best intermittent policy checkpoint and the D4RL dataset. In the real world dataset on ICU sepsis treatment, we also include in Appendix 5.2 an ablation study without ESS constraints for hyperparameter selection on the validation set and in Appendix 3.5 an investigation of the effect of eligible action constraints δ. In Appendix3.4 we also investigate the the weight given by different methods to states with low observed outcomes, and we conduct experiments on the differences in the methods under the prism of ESS and performance in Appendix 5.3. Finally, in Appendix 5.4 we include visualizations of eligible actions for high/mid/low-SOFA patients in addition to a timestep-by-timestep visualization of the two action constraints considered in this paper (based on the eligible action set in POELA and based on the probability under the behavior policy for other methods).

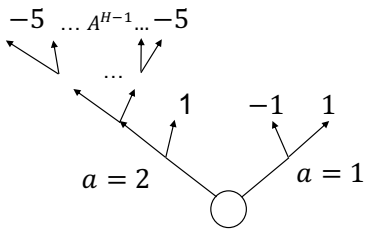
1 COUNTER EXAMPLES IN RL SETTINGS

In the main text, we gave an example about the overfitting issue in contextual bandits with large state and action space in small datasets. Here we show that it is even easier for this to occur in sequential reinforcement learning settings, even when only 2 actions are available in the next two examples with or without state aliasing.

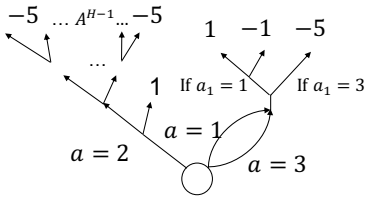
Example 1. *Consider a sequential treatment problem as shown in Figure 1a. There are two actions available in each state. From the first state, action a_1 has a 50% chance of leading to an immediate terminal positive reward $r = 1$ and a 50% chance of leading to an immediate terminal negative reward $r = -1$. From the first state, action a_2 also has 50% chance of leading to an immediate terminal positive reward $r = 1$. For the other 50% of states, action a_2 results in transitions to additional states, which are followed by additional actions, for another $H - 1$ steps; however, all transitions eventually end in a large negative outcome (e.g., $r = -5$). For example, one could consider a risky surgical procedure that results in many subsequent additional operations and but is ultimately typically unsuccessful. Assume the behavior policy is uniform over each action, yielding $\mu(a = 0|x_1) = \mu(a = 1|x_1)$ = 0.5 and a probability of each action sequence following a_2 of $\frac{1}{|A|^{H-1}}$. With even minimal data the value of $\pi(x_0) = a_1$ will be accurately estimated as 0. However, when H is large relative to a function of the dataset size, there always exists a action sequence after an initial selection of a_2 that is not observed in the dataset. This means that a policy π_2 that starts with $\pi(x_0) = a_2$ and then selects an unobserved action sequence will essentially put 0 weight on the resulting contexts that incur $r = -5$ outcomes, even though such outcomes will occur 50% of the time after taking action a_2. In this case, the value of π_2 will be overestimated significantly by IS or self-normalized IS. Thus the offline policy optimization will prefer taking action 2 at the first step as a result of overfitting even though the true value of first taking a_2 is -1.5 and the optimal policy value is 0, obtained by taking action a_1.*

Now we add a slight change in the transitions shown in Figure 1a. We can see that model/value-based approach will also fail.

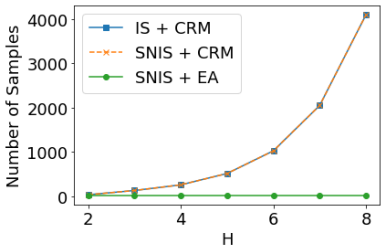
Example 2. *In this example, we add another action in the first step. The action 3 and action 1 will lead to the same next state. However in the next state, no matter which action taken, the reward will depends on the action taken in the last step: If $a_1 = 1$, then we have the same reward for $a = 1$ in the example in Figure 1a. If $a_1 = 3$ then we have a reward -5. Thus model and value based method will mix the reward for $a_1 = 1$ and $a_1 = 3$ so fail in this example. Other method is not affected by the additional structure as it only add an action with minimum reward.*



(a) Example 1.



(b) Example 2.



(c) Number of samples to solve Example 2/3 for $A = 2$.

2 PROOFS OF SECTION ??

Proof of Theorem **??**.

Proof.

$$\sum_{x_h^{(j)} \in \mathcal{B}(x_h^{(i)}, \delta)} \frac{\pi(a_h^{(j)}|x_h^{(j)})}{\mu(a_h^{(j)}|x_h^{(j)})} \geq \sum_{x_h^{(j)} \in \mathcal{B}(x_h^{(i)}, \delta)} \pi(a_h^{(j)}|x_h^{(j)}) \tag{1}$$

$$\geq \sum_{x_h^{(j)} \in \mathcal{B}(x_h^{(i)}, \delta)} \max\{0, \pi(a_h^{(j)}|x_h^{(i)}) - \delta L\} \tag{2}$$

$$\geq \sum_{a \in A_h(x_h^{(i)}; \mathcal{D}, \delta)} \max\{0, \pi(a|x_h^{(i)}) - \delta L\} = 1 - \delta L|\mathcal{A}| \tag{3}$$

□

Proof of Corollary **??**.

Proof.

$$\sum_{x_1^{(j)} \in \mathcal{B}(x_1^{(i)}, \delta)} \frac{\max\{W^{(i)}, M\}}{\sum_{i=1}^n \max\{W^{(i)}, M\}} \geq \sum_{x_1^{(j)} \in \mathcal{B}(x_1^{(i)}, \delta)} \frac{\max\{W^{(i)}, M\}}{nM} \tag{4}$$

$$\geq \frac{\max\{\sum_{x_1^{(j)} \in \mathcal{B}(x_1^{(i)}, \delta)} W^{(i)}, M\}}{nM} \tag{5}$$

$$\geq \frac{1 - \delta L|\mathcal{A}|}{nM} \tag{6}$$

□

Proof of Proposition **??**.

Proof. This is due to $\pi(a|x_h^{(i)})$ and $\mu(a|x_h^{(i)})$ are independent from history given $x_h^{(i)}$. So $W_{1:h}^{(i)}$ and $W_h^{(i)}$ are conditionally independent given $x_h^{(i)}$. □

Proof of Theorem **??**.

Proof. Let $\mathrm{P}_h(x; \mu)$ to be the distribution of context at h-th step with roll-in policy μ. For any fixed a, we can define the distribution $\mathrm{P}_h(x|a; \mu) = \mu(a|x)\mathrm{P}_h(x; \mu) / \sum_a \mu(a|x)\mathrm{P}_h(x; \mu)$. For a such that $\mu(a|x) > 0$, $\mathrm{P}_h(x|a; \mu)$ is also greater than zero. All $x_h^{(i)}$ with $a_h^{(i)} = a$ are i.i.d. samples draw from the distribution $\mathrm{P}_h(x; \mu)$. By the property of nearest neighbor [**?**], with probability 1:

$$\min_{x_h^{(i)} s.t. a_h^{(i)} = a} \mathrm{dist}(x, x_h^{(i)}) \to 0 < \delta.$$

That means with probability 1 $a \in A_h(x; \mathcal{D}, \delta)$ for all a such that $\mu(a|x) > 0$. Thus we proved the theorem statement and that the policy class will contain all π such that $\pi(a|x) > 0$ if $\mu(a|x) > 0$. □

Proof of Theorem **??**.

Proof. Given the overlap assumption and Theorem **??**, for all π we have $a \in A_h(x; \mathcal{D}, \delta)$ for all a such that $\pi(a|x) > 0$ with probability 1. Thus the solution to Equation **??** is the same as $\arg\max_\pi J(\pi, \mathcal{D}) := \hat{\pi}_{J,\mathcal{D}}$.

By the condition that $M \to \infty$ and $\frac{M}{n} \to 0$ as $n \to \infty$, we have that the truncated IS estimator is mean square consistent [**?**]:

$$\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{h=1}^{H} r_h^{(i)}\right)\min\left\{\prod_{h=1}^{H} W_h^{(i)}, M\right\} \xrightarrow{q.m.} v^{\pi}, \tag{7}$$

as $n \to \infty$. Similarly, we have that the mean of weights converge to 1 in quadratic mean:

$$\frac{1}{n}\sum_{i=1}^{n}\min\left\{\prod_{h=1}^{H} W_h^{(i)}, M\right\} \xrightarrow{q.m.} 1. \tag{8}$$

By continuous mapping theorem, we have that the self-normalized truncated IS converge to v^{π} in probability $\hat{v}_{\text{SNTIS}} \xrightarrow{p} n$. The empirical variance penalty, also converge to 0 almostly surely, since M/n converge to 0:

$$\frac{\sum_{i=1}^{n}\left(r^{(i)} - \hat{v}_{\text{SNTIS}}\right)^2 \left(\min\{W^{(i)}, M\}\right)^2}{\left(\sum_{i=1}^{n}\min\{W^{(i)}, M\}\right)^2} \leq \frac{M^2}{\left(\sum_{i=1}^{n}\min\{W^{(i)}, M\}\right)^2} \xrightarrow{q.m.} 0. \tag{9}$$

Thus the objective function $J(\pi; \mathcal{D})$ converge to v^{π} in probability:

$$\Pr\left(|J(\pi; \mathcal{D}) - v^{\pi}| > \epsilon\right) = \delta_n \to 0. \tag{10}$$

Since we assume $|\Pi| < \infty$, we have

$$\Pr\left(\forall \pi \in \Pi \, |J(\pi; \mathcal{D}) - v^{\pi}| > \epsilon\right) = |\Pi|\delta_n. \tag{11}$$

So with probability $|\Pi|\delta_n$, for any ϵ:

$$\begin{aligned} v^{\hat{\pi}_{J,\mathcal{D}}} &\geq J(\hat{\pi}_{J,\mathcal{D}}, \mathcal{D}) - \epsilon && (12)\\ &\geq J(\pi^{\star}, \mathcal{D}) - \epsilon && (13)\\ &\geq v^{\pi^{\star}} - 2\epsilon, && (14) \end{aligned}$$

where $\pi^{\star}$ is $\arg\max_{\pi \in \Pi} v^{\pi}$. As $|\Pi|\delta_n \to 0$, we proved the true value of empirical maximizer $v^{\hat{\pi}_{J,\mathcal{D}}}$ converge to the maximum of value $\max_{\pi \in \Pi} v^{\pi}$ in probability. □

3 EXPERIMENT DETAILS

For all experiments in the main text, we report the test performance of the policy saved at the end of training either through online Monte-Carlo estimation if a simulator is available, or using SNTIS estimates on a held out test set.

For all experiments reported in Appendices 4.2 and 5.1, we follow the 3-phases pipelines we describe hereafter to decide the test score we report in the corresponding Tables. To put ourselves in the more realistic situation of real-world applications where practitioners would select a policy from regular checkpoints along its training on the basis of its SNTIS score on the validation set, an algorithm is trained on the training set multiple times, using different hyperparameters and several restarts. Intermittent policies generated during the training process identified with the highest self-normalized truncated IS (SNTIS) estimates on a held-out validation set are saved at checkpoints. The pipeline is illustrated in Figure 2.

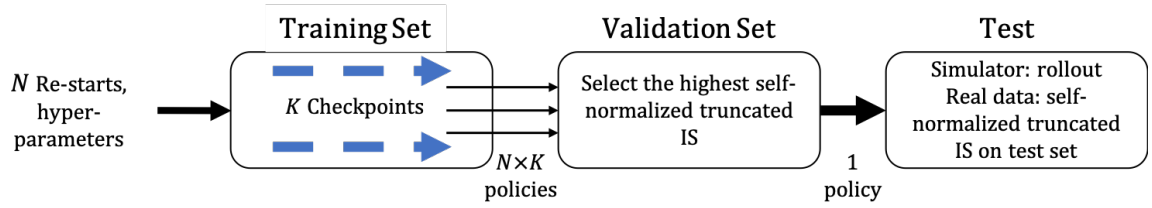


Figure 2: The process of hyperparameters search and test in the experiment.

The open-source code for POELA can be found here: https://github.com/StanfordAI4HI/poela.

3.1 EXPERIMENT DETAILS IN TGI SIMULATOR

The TGI simulator describes low-grade gliomas (LGG) growth kinetics in response to chemotherapy in a horizon of 30 months using an ordinary differential equation model. The parameter in ODEs are estimated using data from adult diffuse LGG during and after chemotherapy was used, in a horizon of 30 months. The goal in this environment is to achieve a reduction in mean tumor diameters (MTD) while reducing the drug dosage [**?**]. We includes the MTD, the drug concentration, and the number of month (time-step) in the context space. Notice that this context space is non-Markov as it does not include all parameters in the ODEs. Actions are binary representing taking the full dose or no dose which is same as prior work [**?**]. The reward at each step consist of an immediate penalty proportional to the drug concentration, and a delayed reward at the end measures the decrease of MTD compared with the beginning. Each episodes, the parameters including the initial MTD are sampled from a log-Normal distribution as [**?**] representing the difference in individuals. The behavior policy is a fixed dosing schedule of 9 months (the median duration from **?**) plus 30% of a uniformly random choice of actions. We run all algorithms on a training set with 1000 episodes with different hyperparameters (listed below), and 5 restarts, saving checkpoints along the training.The validation set is comprised of 1000 episodes as well.

Hyperparameters. In the first part of Table 1 we show the searched hyperparameters of each algorithm, except that the parameter b in PQL is set adaptively as the 2-percentile of the score on the training set as in the original paper **?**. As we know the behavior policy, we use the true behavior policy in BCQ and PQL algorithm. So BCQ threshold takes only two values as the behavior policy is ϵ-deterministic so there are only two distinct values. In the second part of Table 1 we specify some fixed hyperparameters/settings for all algorithm. All policy/Q functions are approximated by fully-connected neural networks with two hidden layers with 32 units.

Hyperparameters	used in algorithms	values
δ	POELA	$0.05, 0.1, 0.5$
$\hat{\mu}$ threshold	PO-μ	0.01, 0.05, 0.1, 0.2
CRM Var coefficient	POELA, PO-CRM	$0, 0.1, 1$
BCQ threshold	BCQ, PQL	$0.0, 0.2$
M in $\hat{v}_{\text{SNTIS}}$	All	1000
Max training steps	POELA, PO-CRM, PO-μ	500
	BCQ, PQL	1000
Number of checkpoints	All	50
Batch size	BCQ, PQL	100

Table 1: Hyperparameters in the TGI simulator experiment

The difference in the max update steps and checkpoints frequency is caused by the fact that BCQ and PQL is updated by stochastic gradient descent and all policy optimization based on SNTIS is using gradient descent.

3.2 EXPERIMENT DETAILS IN THE MIMIC III DATASET

The MIMIC III sepsis dataset is available upon application and training: https://mimic.mit.edu/iii/gettingstarted/. The code to extract the cohort is available on: https://gitlab.doc.ic.ac.uk/AIClinician/AIClinician. This cohort consists of data for 14971 patients. The contexts for each patient consist of 44 features, summarized in 4-hour intervals, for at most 20 steps. The actions we consider are the prescription of IV fluids and vasopressors. Each of the two treatments is binned into 5 discrete actions according to the dosage amounts, resulting in 25 possible actions. The rewards are defined from the 90-day mortality in the logs, 100 if the patient survives and 0 otherwise.

We now provide details of the experiment on MIMIC III sepsis dataset here. We run all algorithms on a training set with 8982 trajectories with different hyperparameters (listed below), and 3 restarts, saving checkpoints along the training.The validation set is comprised of 2994 trajectories. Finally we get the $\hat{v}_{\text{SNTIS}}$ evaluation on the test set with 2995 trajectories. In the first part of Table 2 we list the hyperparameters that we searched on the validation set for each algorithm, except that the parameter b in PQL is set adaptively as the 2-percentile of the score on the training set as in the original paper [**?**]. In the second part of Table 1 we specify some fixed hyperparameters/settings for all algorithm. All policy/Q-functions are

approximated by fully-connected neural networks with two hidden layers with 256 units.

Hyperparameters	used in algorithms	values
δ	POELA	$0.4, 0.6, 0.8, 1.0$
$\hat{\mu}$ threshold	PO-μ	0.01, 0.02, 0.05, 0.1
CRM Var coefficient	POELA, PO-CRM	$0, 0.1, 1, 10$
BCQ threshold	BCQ, PQL	$0.0, 0.01, 0.05, 0.1, 0.3, 0.5$
M in $\hat{v}_{\text{SNTIS}}$	All	1000
Max training steps	POELA, PO-CRM, PO-μ	1000
	BCQ, PQL	10000
Number of checkpoints	All	100
Batch size	BCQ, PQL	100

Table 2: Hyperparameters in the MIMIC III sepsis experiment

As we explained, the difference in the max update steps and checkpoints frequency is caused by the fact that BCQ and PQL is updated by stochastic gradient descent and all policy optimization based on SNTIS is using gradient descent.

3.3 EXPERIMENT DETAILS FOR THE BEHAVIOR POLICY ESTIMATION

In the implementation of BC, we use Multi-Layer Perceptrons (MLPs) neural networks with layer dimensions [32, 32, 32] for the LGG Tumor Growth Inhibition simulator and [256, 256, 256] for the MIMIC III dataset. All use ReLU activations. For BCRNN, we use 3-layer GRUs with a RNN hidden dimension of size 100. All networks are trained using Adam optimizer [**?**] with learning rate $3e-4$. For all experiments, BC and BCRNN are trained for 500 steps and directly serve as estimated behavior policies.

3.4 IMPORTANCE WEIGHTS IN LOW-REWARD TRAJECTORIES

To examine if the proposed overfitting phenomenon exists in real experimental datasets, we compute the importance weights of the learned policy on the low-reward trajectories in the training data for our MIMIC III dataset and our tumor simulator. Our hypothesis is that overfitting of the importance weights in policy gradient methods may result in the algorithm avoiding initial states with low rewards, which motivated our proposed algorithm.

In MIMIC III dataset the reward for a trajectory is either 0 or 100. We define the low-reward trajectories as those with 0 reward. Low-reward trajectories are over 60% of all trajectories in the dataset. In the Tumor simulation experiment we define a low-reward trajectory when reward is less than -2. Over 95% of trajectories in the Tumor simulation dataset are low-reward trajectories.

The table below shows, for each algorithm and setting, the sum of the SNTIS weights of the learned policy on the training set, for low-reward trajectory states. Our primary interest is to illustrate that alternate policy gradient methods that are also suitable for non-Markov domains, can exhibit the importance sampling overfitting of avoiding low reward trajectories. We indeed see in Table 3 that POELA has a much larger weight on low-reward trajectories than alternate offline policy search methods:

Method	POELA	PO-μ	PO-CRM
MIMIC III	0.028	0.001	0.003
Tumor non-MDP	0.054	- (fixed policy)	0.005

Table 3: Importance weights overfitting: sum of SNTIS weights of learned policy on the training set.

The Q-learning baselines we consider (BCQ and PQL) do not directly use the importance weights, but they do try to avoid actions and/or states and actions with little support. Our POELA method can be viewed as being similarly inspired, but for

non-Markovian settings where policy gradient is beneficial. We also compute the SNTIS weights of the BCQ/PQL policy on the training set in the Markov domain that satisfies the Markov assumptions of BCQ/PQL. In Table 4 we can see that POELA, BCQ and PQL all still give significantly more weight to low reward trajectories than the alternate policy gradient methods:

Method	POELA	PO-μ	PO-CRM	BCQ	PQL
Tumor MDP	0.097	- (fixed policy)	0.0004	0.083	0.124

Table 4: Importance weights overfitting: sum of SNTIS weights of learned policy on the training set.

These results help illustrate that the over avoidance of low-reward trajectories can be observed by past policy gradient methods in our datasets. Of course, one challenge is that in real settings, an excellent policy may have low importance weights in avoidable low-reward states and trajectories, but should have higher importance weights in non-avoidable low reward starting states and trajectories. To get a fuller picture of performance, it is helpful to look both at the weights on trajectories with low rewards and the test evaluation results. Compared with strong policy gradient baselines, our proposed regularization method have larger importance weights on low-reward trajectories, and the gap between training/validation evaluation and online test performance is also smaller, suggesting that we are less likely to learn policies that erroneously believe they can avoid unavoidable low reward settings.

3.5 THE EFFECT OF ELIGIBLE ACTION CONSTRAINTS δ

In this section we explore how the choice of δ, which constrains the policy class through impacting the eligible actions, impacts empirical performance. Larger δ corresponds to a less constrained policy class. Other hyperparameters are selected by the same procedure as described in previous sections.

Table 5 shows the results. As δ increases, the policy search operates with less constraints. The results show that in this case, our policy gradient method produces a policy with a higher value in the training set, but that policy may not perform as well in the test evaluation, and may have a smaller effective sample size than when a smaller δ is used. The best hyperparameter value δ lies in the middle of the explored range. δ can be selected based on performance and effective sample size.

δ	0.4	0.6	0.8	1.0
training $\hat{v}_{\text{SNTIS}}$	91.62	98.41	98.9	99.12
training ESS	3601.12	2242.07	1993.08	1769.46
test $\hat{v}_{\text{SNTIS}}$	86.62	90.07	91.46	90.23
test ESS	1278.08	819.64	624.92	542.53

Table 5: The effect of eligible action constraints δ on the results in MIMIC III sepsis dataset.

4 ADDITIONAL EXPERIMENTS: LGG TUMOR GROWTH INHIBITION SIMULATOR

In this section, we provide additional experiments to the existing LGG Tumor Growth Inhibition simulator experiments.

4.1 EXPERIMENT WITH ESTIMATING THE BEHAVIOR POLICY WITH FUNCTION APPROXIMATION

	Algorithms	POELA	PO-$\hat{\mu}$	PO-CRM	BCQ	PQL	9-mon
Non-MDP	Test v^π	92.34 ± 1.57	59.62 ± 12.71	46.66 ± 14.05	19.36 ± 5.66	30.44 ± 10.38	68.12
	$\hat{v}_{\text{SNTIS}} - v^\pi$	0.94 ± 1.66	31.38 ± 10.97	42.98 ± 12.87	72.35 ± 5.66	62.24 ± 10.94	–
MDP	Test v^π	91.04 ± 0.55	78.21 ± 4.94	78.70 ± 0.60	99.26 ± 0.59	99.66 ± 0.29	68.12
	$\hat{v}_{\text{SNTIS}} - v^\pi$	3.40 ± 2.48	15.58 ± 3.92	15.10 ± 3.97	-3.88 ± 1.60	-4.09 ± 1.75	–

Table 6: LGG Tumor Growth Inhibition simulator. Test v^π and amount of overfitting of the learned policy. Test v^π is computed from 1000 rollouts in the simulator. $\hat{v}_{\text{SNTIS}}$ on the validation set $-$ test v^π represents the amount of overfitting. All numbers are averaged across 5 runs with the standard error reported. Behavior policy $\hat{\mu} =$ BC.

	Algorithms	POELA	PO-$\hat{\mu}$	PO-CRM	BCQ	PQL	9-mon
Non-MDP	Test v^π	95.81 ± 1.68	76.64 ± 14.65	76.43 ± 14.59	19.79 ± 5.76	31.57 ± 10.63	68.12
	$\hat{v}_{\text{SNTIS}} - v^\pi$	-1.52 ± 1.79	16.35 ± 14.40	16.56 ± 14.36	73.71 ± 6.34	62.92 ± 11.19	–
MDP	Test v^π	89.25 ± 1.51	75.43 ± 8.25	73.61 ± 0.30	99.57 ± 0.29	99.96 ± 0.12	68.12
	$\hat{v}_{\text{SNTIS}} - v^\pi$	5.17 ± 2.20	17.66 ± 7.90	19.44 ± 8.52	-4.18 ± 1.76	-4.38 ± 1.78	–

Table 7: LGG Tumor Growth Inhibition simulator. Test v^π and amount of overfitting of the learned policy. Test v^π is computed from 1000 rollouts in the simulator. $\hat{v}_{\text{SNTIS}}$ on the validation set $-$ test v^π represents the amount of overfitting. All numbers are averaged across 5 runs with the standard error reported. Behavior policy $\hat{\mu} =$ BCRNN.

4.2 ALTERNATIVE SELECTION PROCEDURE: CHECKPOINT BEST INTERMITTENT POLICIES

In this section, we use the procedure of best policy checkpoint during the training described in Section 3. We report the test performance of the selected policy through online Monte-Carlo estimation.

	Algorithms	POELA	PO-μ	PO-CRM	BCQ	PQL	9-mon
Non-MDP	Test v^π	92.20 ± 1.63	76.99 ± 13.80	75.06 ± 13.22	57.77 ± 16.71	74.76 ± 9.75	68.12
	$\hat{v}_{\text{SNTIS}} - v^\pi$	-1.26 ± 1.92	16.07 ± 13.55	15.57 ± 13.07	37.55 ± 16.91	17.74 ± 9.49	–
MDP	Test v^π	89.52 ± 1.55	69.18 ± 10.17	78.79 ± 6.42	94.7 ± 3.49	96.88 ± 3.76	68.12
	$\hat{v}_{\text{SNTIS}} - v^\pi$	5.16 ± 1.78	24.92 ± 9.71	14.93 ± 5.71	2.75 ± 3.41	-0.26 ± 4.18	–

Table 8: LGG Tumor Growth Inhibition simulator. Test v^π and amount of overfitting of the learned policy. Test v^π is computed from 1000 rollouts in the simulator. $\hat{v}_{\text{SNTIS}}$ on the validation set $-$ test v^π represents the amount of overfitting. All numbers are averaged across 5 runs with the standard error reported. **Procedure: best intermittent policy checkpoints.**

	Algorithms	POELA	PO-$\hat{\mu}$	PO-CRM	BCQ	PQL	9-mon
Non-MDP	Test v^π	94.16 ± 1.82	74.76 ± 7.66	76.38 ± 7.26	92.92 ± 1.68	74.65 ± 14.5	68.12
	$\hat{v}_{\text{SNTIS}} - v^\pi$	0.95 ± 1.92	18.02 ± 7.07	15.02 ± 6.68	0.58 ± 0.27	20.49 ± 14.46	–
MDP	Test v^π	91.81 ± 1.05	84.86 ± 3.48	84.08 ± 3.46	86.22 ± 9.61	95.02 ± 4.95	68.12
	$\hat{v}_{\text{SNTIS}} - v^\pi$	2.7 ± 2.82	9.23 ± 3.85	10.01 ± 3.88	11.03 ± 10.4	2.45 ± 5.27	–

Table 9: LGG Tumor Growth Inhibition simulator. Test v^π and amount of overfitting of the learned policy. Test v^π is computed from 1000 rollouts in the simulator. $\hat{v}_{\text{SNTIS}}$ on the validation set $-$ test v^π represents the amount of overfitting. All numbers are averaged across 5 runs with the standard error reported. Behavior policy $\hat{\mu} =$ BC. **Procedure: best intermittent policy checkpoints.**

	Algorithms	POELA	PO-$\hat{\mu}$	PO-CRM	BCQ	PQL	9-mon
Non-MDP	Test v^π	96.34 ± 1.58	77.51 ± 13.87	75.73 ± 14.3	92.73 ± 1.67	74.94 ± 14.47	68.12
	$\hat{v}_{\text{SNTIS}} - v^\pi$	-2.05 ± 1.9	15.48 ± 13.62	17.27 ± 14.02	0.77 ± 0.52	20.2 ± 14.43	$-$
MDP	Test v^π	90.06 ± 1.65	79.62 ± 7.82	79.54 ± 7.65	86.38 ± 9.47	95.16 ± 4.9	68.12
	$\hat{v}_{\text{SNTIS}} - v^\pi$	4.46 ± 2.31	13.81 ± 6.96	13.89 ± 6.81	10.87 ± 10.24	2.33 ± 5.19	$-$

Table 10: LGG Tumor Growth Inhibition simulator. Test v^π and amount of overfitting of the learned policy. Test v^π is computed from 1000 rollouts in the simulator. $\hat{v}_{\text{SNTIS}}$ on the validation set $-$ test v^π represents the amount of overfitting. All numbers are averaged across 5 runs with the standard error reported. Behavior policy $\hat{\mu} =$ BCRNN. **Procedure: best intermittent policy checkpoints.**

5 ADDITIONAL EXPERIMENTS: MIMIC III SEPSIS

In this section we provide additional experiments to the existing MIMIC III sepsis experiments.

5.1 ALTERNATIVE SELECTION PROCEDURE: CHECKPOINT BEST INTERMITTENT POLICIES

In this section, we use the procedure of using checkpoints to select best policies during the training described in Section 3. We report the test performance of the selected policy using SNTIS estimates on a held out test set.

Method	POELA	PO-$\hat{\mu}$	PO-CRM	BCQ	PQL	Clinician
Test SNTIS	91.46 (90.82)	87.95	87.71	82.67	84.40	81.10
95% BCa UB	93.24 (92.61)	90.58	90.04	86.83	88.29	82.19
95% BCa LB	89.59 (88.68)	84.77	84.90	78.25	80.13	79.80
Test ESS	624.92 (586.37)	372.00	399.59	228.82	231.93	2995

Table 11: MIMIC III sepsis dataset. Test evaluation, $(0.05, 0.95)$ BCa bootstrap interval, and effective sample size. The value of POELA without a CRM variance penalty is shown in parentheses. **Procedure: best intermittent policy checkpoints.**

Method	POELA	PO-$\hat{\mu}$	PO-CRM	BCQ	PQL	Clinician
Test SNTIS	85.01 (89.62)	84.70	85.53	83.17	84.16	81.10
95% BCa UB	88.61 (92.75)	88.56	87.80	92.88	88.04	82.19
95% BCa LB	80.55 (85.57)	80.15	83.23	63.98	79.98	79.80
Test ESS	227.92 (214.12)	228.97	354.86	208.92	209.72	2995

Table 12: MIMIC III sepsis dataset. Test evaluation, $(0.05, 0.95)$ BCa bootstrap interval, and effective sample size. The value of POELA without a CRM variance penalty is shown in parentheses. Behavior policy $\hat{\mu} =$ BC. **Procedure: best intermittent policy checkpoints.**

Method	POELA	PO-$\hat{\mu}$	PO-CRM	BCQ	PQL	Clinician
Test SNTIS	88.34 (90.89)	87.98	85.12	83.20	85.06	81.10
95% BCa UB	91.65 (93.78)	91.06	92.75	91.56	89.12	82.19
95% BCa LB	83.94 (87.05)	84.41	72.96	66.27	79.76	79.80
Test ESS	201.49 (220.86)	285.82	211.20	206.11	212.36	2995

Table 13: MIMIC III sepsis dataset. Test evaluation, $(0.05, 0.95)$ BCa bootstrap interval, and effective sample size. The value of POELA without a CRM variance penalty is shown in parentheses. Behavior policy $\hat{\mu} =$ BCRNN. **Procedure: best intermittent policy checkpoints.**

5.2 ABLATION STUDY: ESS CONSTRAINTS FOR HYPERPARAMETER SELECTION ON VALIDATION SET

In the main text, we set an effective sample size threshold of 200 for a policy/hyperparameter to be selected on validation set. This is to make sure we have large enough effective sample size on the test set to provide reliable off-policy test estimates.

In Table 14, we show the results if we do not threshold the effective sample size on validation set. Generally, all algorithms will prefer a high off-policy estimates without enough effective sample size. On the test set, all algorithms yields a small effective sample size, thus unreliable off-policy estimates and large bootstrap confidence interval. The proposed methods is better than baselines but also has much smaller 95% bootstrap lower bound than with the effective sample size constraint.

Method	POELA	PO-μ	PO-CRM	BCQ	PQL
Test SNTIS	87.63(86.29)	82.36	82.36	83.28	96.32
95% BCa LB	85.06(83.51)	64.92	63.48	56.65	57.25
95% BCa UB	90.00(88.59)	94.22	93.62	100	100
Test ESS	528.18(491.71)	21.23	21.23	9.04	1.27

Table 14: Test evaluation without effective sample size constraint on the validation set, $(0.05, 0.95)$ BCa bootstrap interval, and effective sample size in the sepsis cohort of MIMIC III dataset. Value inside parenthesis of POELA is without CRM variance penalty.

5.3 THE TRADE-OFF BETWEEN ESS AND PERFORMANCE ESTIMATES

A tension in conservative offline optimization is that the most reliable and conservative policy estimates come from effectively imitating the behavior policy (which will maximize ESS). Policies that differ substantially from the behavior policy may yield higher performance, but have less overlap with the existing logged data (and lower ESS). This is illustrated in Figure 3, where the value estimates are plotted for each hyperparameter and re-start of the different algorithms. We observe that POELA achieves a better Pareto frontier between performance estimates and ESS than other algorithms. Note that for this experiment we placed ourselves in the policy selection procedure in which the best policy is selected during training based on SNTIS estimates on the validation set (cf. Table 11).

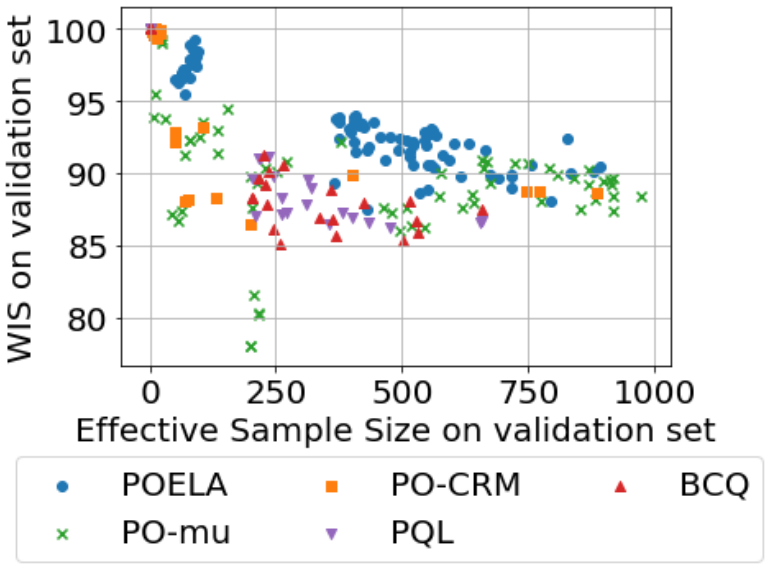
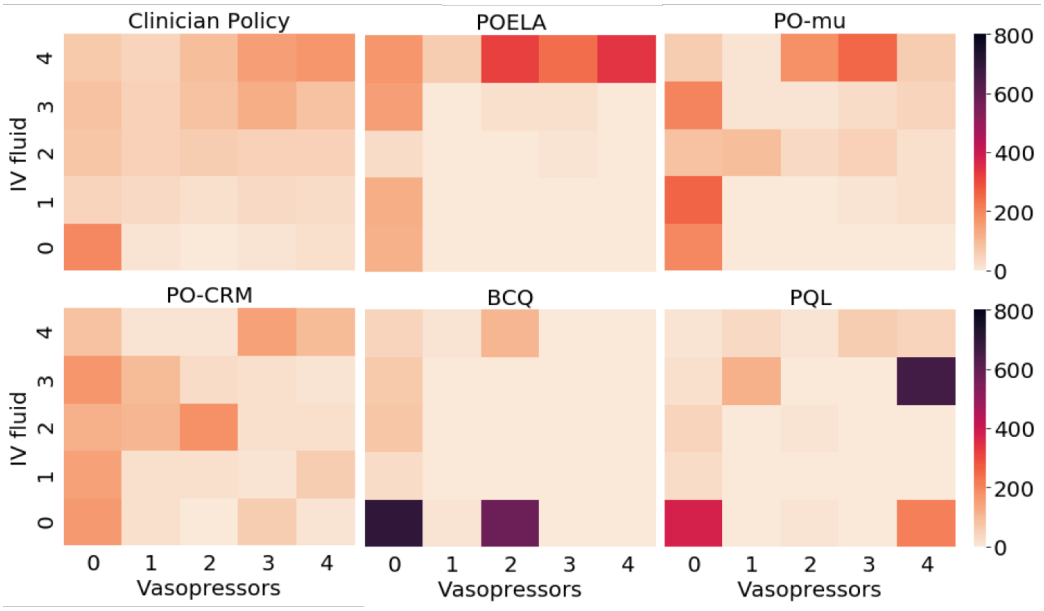


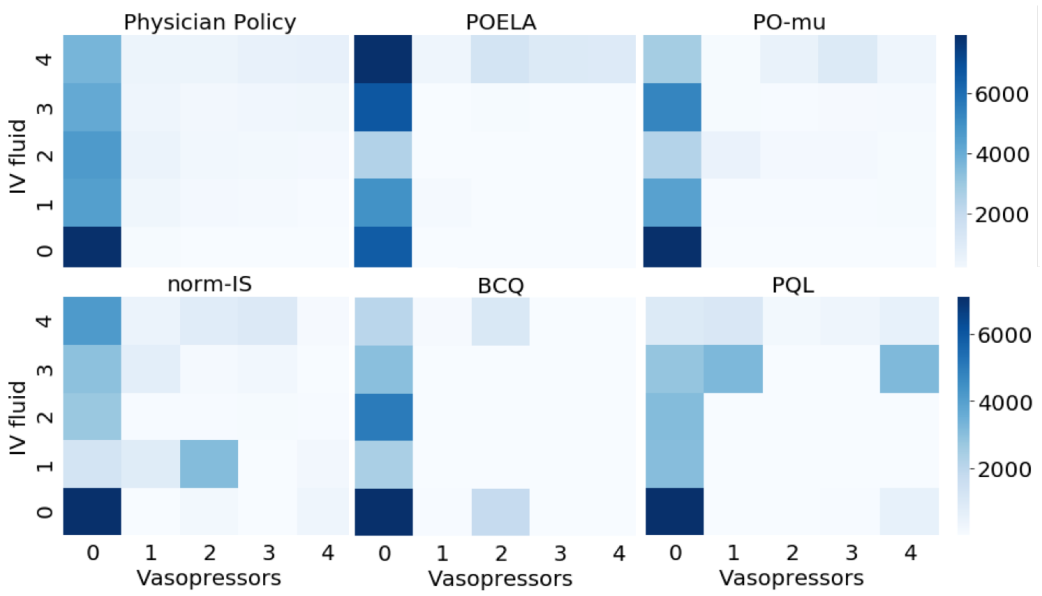
Figure 3: Trade-off between ESS and value estimates.

5.4 ELIGIBLE ACTIONS VISUALIZATION FOR HIGH/MID/LOW-SOFA PATIENTS

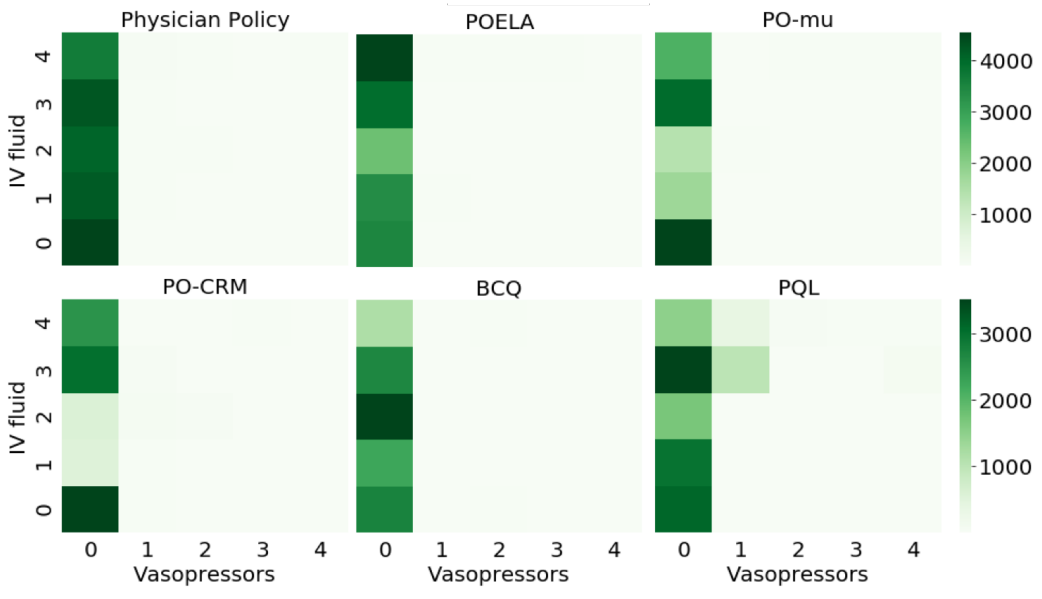
In this section, we explore the learned policies for patients with high logged SOFA scores (measuring organ failure) in the test dataset. Figure 4a illustrates the number of actions taken by different policies and the clinicians. POELA mainly takes treatments similar to the clinician's but more concentrated on high-vasopressors treatments, while PO-CRM and value-based methods take treatments different from the logged clinician decisions, suggesting these policies may be overfitting to avoid contexts with high SOFA. However, some patients arrive with high SOFA scores and a policy must have suitable treatments to support such individuals, which our method appears to ensure. For completeness, we also show the visualization of mid-SOFA $(5-15)$ and low-SOFA (<5) patient contexts in Figures 4b and 4c.



(a) Action counts in high-SOFA contexts



(b) Action counts in mid-SOFA contexts



(c) Action counts in low-SOFA contexts

6 ADDITIONAL EXPERIMENTS: OPENAI GYM ENVIRONMENT CARTPOLE

In this experiment, we collect a dataset by training DQN [**?**] on the task and saving trajectories of horizon 200 steps at regular checkpoints during the training. The dataset is composed of a mixture of sub-optimal and expert data totalling 20000 transitions. For the non-Markov modification, we keep the *Cart Position*, *Cart Velocity* and *Pole Angle* observations but remove the *Pole Angular Velocity* element. In Table 15, we report the hyperparameter used in the experiments.

Hyperparameters	used in algorithms	values
δ	POELA	$0.0001, 0.0005, 0.001, 0.005, 0.01$
$\hat{\mu}$ threshold	PO-μ	0.05, 0.1, 0.15, 0.2
CRM Var coefficient	POELA, PO-CRM	$0, 0.1, 1, 10$
BCQ threshold	BCQ, PQL	$0.0, 0.05, 0.1, 0.2, 0.5$
M in $\hat{v}_{\text{SNTIS}}$	All	1000
Max training steps	POELA, PO-CRM, PO-μ	500
	BCQ, PQL	1000
Number of checkpoints	All	50
Batch size	BCQ, PQL	64

Table 15: Hyperparameters in the CartPole experiment.

6.1 STANDARD EVALUATION PROCEDURE: USE POLICY AT THE END OF TRAINING

Method	POELA	PO-$\hat{\mu}$	PO-CRM	BCQ	PQL	Behavior policy
Test SNTIS	88.29 (86.62)	78.79	72.63	21.28	23.61	41.41
95% BCa UB	89.70 (89.81)	83.87	76.77	24.63	27.14	45.04
95% BCa LB	85.93 (85.57)	69.64	68.15	16.22	20.36	38.16
Test ESS	43.32 (40.78)	30.51	30.13	30.11	30.08	248

Table 16: CartPole dataset. Test evaluation, $(0.05, 0.95)$ BCa bootstrap interval, and ESS. The value of POELA without a CRM variance penalty is shown in parentheses.

Method	POELA	PO-$\hat{\mu}$	PO-CRM	BCQ	PQL	Behavior policy
Test SNTIS	76.18 (72.21)	68.39	67.14	12.13	5.46	41.41
95% BCa UB	89.27 (88.32)	80.22	83.72	12.89	6.63	45.04
95% BCa LB	68.97 (67.49)	57.13	57.78	9.17	5.02	38.16
Test ESS	36.41 (34.72)	34.56	31.87	31.22	30.07	248

Table 17: Non-MDP CartPole dataset. Test evaluation, $(0.05, 0.95)$ BCa bootstrap interval, and ESS. The value of POELA without a CRM variance penalty is shown in parentheses.

6.2 ALTERNATIVE SELECTION PROCEDURE: CHECKPOINT BEST INTERMITTENT POLICIES

7 ADDITIONAL EXPERIMENTS: D4RL

Although our primary focus is on application areas where the Markov assumption may not be correct or unverifiable, we also compare to an additional standard benchmark, namely D4RL.

An adaptation of the POELA algorithm is necessary to work with continuous action spaces. Practically, instead of using the eligible action set A_h, for each data sample, we pre-compute a set of similar actions and use the distance to the closest state x_h associated with the most similar action distributions in the dataset as a smooth penalty in Line 5 of Algorithm **??**.

For each dataset quality (random, medium, and expert) and task (Hopper and Walker2D), we report the performances scaled from 0 to 100 (0 corresponds to the average returns of a random policy and 100 that of an expert policy) following the

Method	POELA	PO-$\hat{\mu}$	PO-CRM	BCQ	PQL	Behavior policy
Test SNTIS	88.43 (87.56)	76.01	82.25	17.74	17.83	41.41
95% BCa UB	90.46 (90.72)	82.87	86.18	21.80	21.84	45.04
95% BCa LB	85.48 (84.63)	66.21	74.30	12.84	13.26	38.16
Test ESS	43.32 (39.66)	31.04	30.87	30.29	30.18	248

Table 18: CartPole dataset. Test evaluation, $(0.05, 0.95)$ BCa bootstrap interval, and ESS. The value of POELA without a CRM variance penalty is shown in parentheses. **Procedure: best intermittent policy checkpoints.**

Method	POELA	PO-$\hat{\mu}$	PO-CRM	BCQ	PQL	Behavior policy
Test SNTIS	75.76 (75.70)	68.66	66.34	11.73	5.70	41.41
95% BCa UB	92.35 (89.16)	79.56	82.46	12.49	6.71	45.04
95% BCa LB	68.34 (66.08)	55.49	57.50	7.98	5.08	38.16
Test ESS	37.72 (35.27)	35.15	36.02	30.12	31.77	248

Table 19: Non-MDP CartPole dataset. Test evaluation, $(0.05, 0.95)$ BCa bootstrap interval, and ESS. The value of POELA without a CRM variance penalty is shown in parentheses. **Procedure: best intermittent policy checkpoints.**

experimental protocol for D4RL with 200 episodes in each dataset. We compare with state-of-the-art methods in this dataset. The results are reported in Table 20.

Dataset	POELA	BCQ	CQL	Behavior policy
Hopper-random	10.5	10.5	10.8	9.8
Hopper-medium	43.7	42.9	41.4	29.0
Hopper-expert	58.9	59.7	52.6	43.6
Walker2D-random	6.1	4.6	5.4	1.6
Walker2D-medium	33.8	31.1	49.6	6.6
Walker2D-expert	32.2	32.8	54.7	50.2

Table 20: Additional experiments on 6 D4RL datasets.

The results in Table 20 suggest that POELA performs similarly to two other state-of-the-art methods in this setting, even though POELA does not make Markov assumptions, which are made and leveraged in BCQ and CQL.