Truly Batch Apprenticeship Learning with Deep Successor Features - Appendix Additional Supplementary Material

1 Classical Control Tasks

We used OpenAI Gym for our classical control experiments that are well-known. The scale of the environments is relatively small and the dynamics have some stochasticity.

environment	dim(s)	dim(a)
MountainCar-v0	2	3
Cartpole-v0	4	2
Acrobot-v1	6	3

Table 1: **Classical Control Environments:** state and action space dimensions on classical control environments.

1.1 Experimental Details

How is batch data collected in control experiments ? In all our control experiments, we collect batch data by solving the environment using DDQN and collect transitions from this optimal policy learned by DDQN. Once the transitions from this policy are collected, they are labeled expert data and provided to the batch IRL models as D_e and at this point, access to the simulator is turned off, thus simulating batch settings. We acknowledge the fact that DDQNs are near-optimal MDP solvers and do not produce ”the” expert policy. Having said that, this resembles several real-world tasks where all human experts are almost optimal (like clinicians in the ICU) and hence our control environments mirror what we observe in our sepsis management task. Besides, we found that the final IRL policy learned on the control environments were of almost the same quality (in terms of accumulated rewards) as the known top-performing solutions to these environments.

For LSTD-μ and SCIRL, we also tried other variants of feature engineering by obtaining basis features using the means and standard deviations of the state samples uniformly sampled from the environment. The performance results obtained for the baselines were in the same range as those tabulated in the main paper and hence we do not state the same again. For MountainCar-v0, we used a Gaussian kernel of 25 components for $\phi(s)$ and subsequently we onehot-encoded $\phi(s)$ based on the 3 actions to represent $\phi(s, a)$ so its dimension becomes 75. For Acrobot-v1 and Cartpole-v0, we used RBF Kernel of 100 components (25 components each $\gamma = 0.1, 0.5, 1.0, 5.0$).

Algorithm 1 Deep Successor Feature Network (DSFN)

```
Input: $D_e$ (Demo.), $\phi$ (Feature Map), $\pi$ (Eval. Policy), $\delta$
Parameter: $\theta$, Output: $\mu^\pi_\theta$
Feed $D_e$ to Experience Replay Buffer (ERB).
Initialize $\theta$ for $\mu_\theta(\pi)$
while $\mathcal{L}_{\text{val}} > \delta$ do
    Sample a batch $B = \{(s, a, s')\}$ from ERB.
    Set TD targets according to Eqn. (??).
    Compute the gradient on $B$ using Eqn. (??) and update
    $\theta$ with mini-batch gradient descent.
end while
return $\theta = \mu^\pi_\theta$
```

We set the maximum of 10 iterations with two stopping conditions: first is when feature expectation margin at 0.1 and second is when the difference in validation accuracy for action prediction for the two consecutive iterations drops lower than 5%. We found the latter stopping condition to be useful in keeping the training loop stable. Unlike typical inverse reinforcement learning routines, there is no correcting mechanism that's based on the ground-truth information (typically achieved by on-policy evaluation) and hence, the training loop may diverge in the complete batch apprenticeship learning.

2 Model Details

2.1 DSFN Training

The pseudo-code for training a DSFN model can be found in Algorithm 1.

2.2 Neural Network Model Architectures

We used different neural network models in our work such as DSFN (off-policy feature expectations estimator), TRIL (warm-starter and representation learner) and DQN (MDP solver) whose architecture and parametric details can be found in Table 2 in the next page. These parameters were identified by grid search within reasonable limits for each parameter.

3 Clinical Task: Sepsis Management

In comparison to the classical control tasks, the clinical task had three crucial differences. First, the ground-truth

Hyperparameters	TRIL	DSFN	DQN
number of hidden layers	2 (+1)	2	2
hidden node size	128	64	128
max training iterations	50000	50000	30000
activation function	tanh	tanh	tanh
optimizer	Adam	Adam	Adam
adam epsilon	1e-4	1e-4	1e-4
adam learning rate	3e-4	3e-4	3e-4
mini-batch size	64	32	64
λ (regularization)	1.4	-	-
state normalizer	Y	Y	N
prioritized experience replay	N	N	Y
prioritized experience replay alpha	-	-	0.6
prioritized experience replay beta0	-	-	0.9
moving average for target network	-	0.01	0.01
discount rate	0.99	0.99	0.99
stopping condition (validation)	5e-3	5e-3	1e-2

Table 2: **The Hyper-parameters of Neural Networks**

evaluation is almost impossible, as there is no reliable simulator for the task. Second, the demonstrations data is noisy and the expert policy is not necessarily deterministic or optimal. For a challenging treatment problem like our task, clinicians may not have a consensus on the optimal treatment. Third, interpretability matters. As is common in the domains like health-care, the batch constraint arises due to safety concerns. Thus, offering the interpretability of a learned solution via a reward function (IRL) is preferred to pure imitation learning.

Here we share the details for the sepsis management experiment. The features that were chosen with a view to represent represent the most important parameters that clinicians would examine when deciding treatment and dosage for sepsis patients. The features broadly could be categorized into four groups as below.

3.1 Patient Features

1. **Index Measures** - Shock Index, Elixhauser, SIRS, Gender, Re-admission, GCS - Glasgow Coma Scale, Age
2. **Lab Values** - Albumin, Arterial pH, Calcium, Glucose, Hemoglobin, Magnesium, PTT - Partial Thromboplastin Time, Potassium, SGPT - Serum Glutamic-Pyruvic Transaminase, Arterial Blood Gas, BUN - Blood Urea Nitrogen, Chloride, Bicarbonate, INR - International Normalized Ratio, Sodium, Arterial Lactate, CO2, Creatinine, Ionised Calcium, PT - Prothrombin Time, Platelets Count, SGOT - Serum Glutamic-Oxaloacetic Transaminase, Total bilirubin, White Blood Cell Count
3. **Vital Signs**: Diastolic Blood Pressure, Systolic Blood Pressure, Mean Blood Pressure, PaCO2, PaO2, FiO2, Respiratory Rate, Temperature (Celsius), Weight (kg), Heart Rate, SpO2
4. **Intake and Output Events**: Fluid Output - 4 hourly period, Total Fluid Output, Mechanical Ventilation, IV Fluids

3.2 Experimental Details

When several data points were present in one window, appropriate statistics (mean or sum) deemed apt by clinicians were used for aggregation. The trajectories of clinical measurements have no state space—effective state construction itself is an important problem in health care—so we modeled the data as coming from a continuous state space that consisted of 46 features, including important non-vasopressor interventions such as mechanical ventilation and IV fluids. We consider in-hospital mortality and leaving the ICU (alive) absorbing states. (Each patient's treatment trajectory comprises an episode of expert demonstrations for our agent to learn from. Our trajectory lengths are less than or equal to 20 steps (about 80 hours of ICU stay since the data was collated over 4 hour bins). Vasopressor actions were discretized into 5 bins: one bin for no dose and 4 associated with quartiles from data. We used a discount factor γ of 0.99.

4 Code Repository

Link to the github repo that contains our code for reproducing the experiments :

```
https://github.com/dtak/
batch-apprenticeship-learning
```