Hierarchical Imitation Learning with Contextual Bandits for Dynamic Treatment Regimes

Lu Wang [1] **Wenchao Yu** [2] **Wei Cheng** [2] **Bo Zong** [2] **Haifeng Chen** [2]

Abstract

Imitation learning has been proved to be effective in mimicking experts' behaviors from their demonstrations without access to explicit reward signals. Meanwhile, complex tasks, e.g., dynamic treatment regimes for patients with comorbidities, often suggest significant variability in expert demonstrations with multiple sub-tasks. In these cases, it could be difficult to use a single flat policy to handle tasks of hierarchical structures. In this paper, we propose the hierarchical imitation learning model, HIL, to jointly learn latent high-level policies and sub-policies (for individual sub-tasks) from expert demonstrations without prior knowledge. First, HIL learns sub-policies by imitating expert trajectories with the sub-task switching guidance from high-level policies. Second, HIL collects the feedback from its sub-policies to optimize high-level policies, which is modeled as a contextual multi-arm bandit that sequentially selects the best sub-policies at each time step based on the contextual information derived from demonstrations. Compared with state-of-the-art baselines on real-world medical data, HIL improves the likelihood of patient survival and provides better dynamic treatment regimes with the exploitation of hierarchical structures in expert demonstrations.

1. Introduction

Dynamic Treatment Regimes (DTRs) for patients involve a sequence of tailored treatment decision rules (Murphy, 2003). In practice, such decision rules can be grouped as actions for individual sub-policies (e.g., treatment rules for COVID-19 patients with different underlying conditions) where a high-level treatment policy, that controls how to switch between sub-policies in a right way, is needed so that patients with a wide variety of symptoms can be optimally treated. For example, possible actions in a high-level policy of DTRs could be "pain treatment", "fever treatment" and "inflammation treatment", each of which corresponds to a sub-policy.

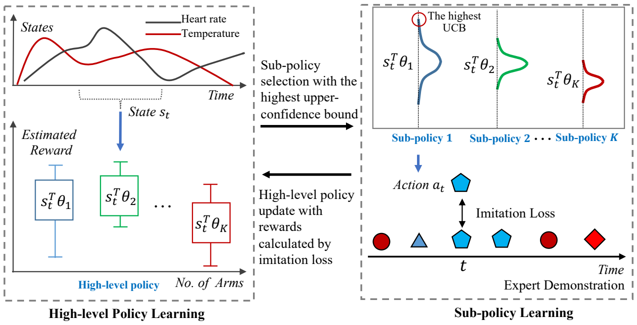


Figure 1. Illustration of the HIL model for learning optimal dynamic treatment regimes

Recent developments in discovering DTRs have heightened the importance of reinforcement learning (RL) and imitation learning (IL), which are performed to recover the doctor's treatment policies. However, in order to get a well-performed policy, RL requires user-defined reward signals which are either very sparse or need clinical guidance. IL has the potential to mimic experts' behaviors directly from demonstrations, but it is intrinsically difficult to imitate the aforementioned hierarchical policies in DTRs by a single flat policy. Hierarchical policies have been proposed to improve the quality of policy learning by dividing a complex task into several simpler sub-tasks, with success in hierarchical reinforcement learning (Ahilan & Dayan, 2019; Rizzolatti et al., 2001; Sutton et al., 1999; Tang et al., 2018; Vezhnevets et al., 2017). However, existing hierarchical imitation learning methods (Le et al., 2018; Hausman et al., 2017) need prior knowledge in high-level sub-task switching policies (high-level policies for short) that control how to switch between sub-tasks. Such prior knowledge may not be applicable in DTRs since it is not practical to ask doctors to label high-level policies, which limits the feasibility of the existing hierarchical IL approaches in health care domain.

[1] East China Normal University [2] NEC Laboratories America. Correspondence to: Wenchao Yu <wyu@nec-labs.com>.

Reinforcement Learning for Real Life (RL4RealLife) Workshop in the 38th International Conference on Machine Learning, 2021.

To address the limitation above, in this paper, we propose the hierarchical imitation learning model, HIL, to jointly learn latent high-level policies and sub-policies of individual sub-tasks from expert demonstrations using contextual bandits without prior knowledge in high-level policies. As shown in Fig. 1, we overview HIL from the following two aspects. First, HIL uses high-level policy to sequentially select the best sub-policy (i.e. high-level action) at each time step based on contextual information derived from demonstrations where the selected sub-policy takes a low-level action. Second, HIL collects the low-level action's reward feedback from sub-policies to update the high-level policy. The underlying optimization task is formulated as a contextual multi-arm bandit (Li et al., 2010) and the selected sub-policy will be updated accordingly by imitating expert demonstrations. In the phase of high-level policy learning, HIL encourages *smoothness* in the sequential sub-task selection inspired by the observation that sub-task switch is less frequently in practice. In the phase of sub-policy learning, the reward function is designed to enforce *diversity* in learned sub-policies such that complex tasks can be decomposed into multiple modes with hierarchical structures. We quantitatively validate the performance of HIL on dynamic treatment recommendation task with real-world medical data, and the empirical results demonstrate the effectiveness of HIL. Our main contributions are summarized as follows:

- We propose a new hierarchical imitation learning model, HIL, to jointly learn latent high-level policies and sub-policies from expert demonstrations using contextual bandits without the assumption of prior knowledge in high-level policies.
- The reward functions are carefully designed to encourage *diversity* among learned sub-policies and *smoothness* in terms of sub-task switching in high-level policies.
- Quantitative experiments and qualitative case studies on MIMIC-III (Johnson et al., 2016) demonstrate that HIL could further reduce the estimated mortality and provides better dynamic treatment regimes with the exploitation of hierarchical structures from the treatment demonstrations, compared with competitive state-of-the-art baselines.

The rest of this paper is organized as follows. Section 2 introduces the preliminaries of contextual bandits and imitation learning. Section 3 describes the HIL model. Section 4 empirically evaluates HIL on dynamic treatment recommendation task. We summarize the related work in Section 5, followed by the conclusions in Section 6.

2. Preliminaries

2.1. Contextual Bandits

The goal of multi-armed bandit (Auer et al., 2002) is to find an optimal strategy in finite steps which minimizes the regret $\mathbb{E}\left[\sum_{t=1}^{T} r_{t,a_t^*}\right] - \mathbb{E}\left[\sum_{t=1}^{T} r_{t,a_t}\right]$ between the agent's action a_t and the optimal action a_t^* at each time step t. Multi-armed bandit algorithms solve this problem by balancing two factors: 1) *exploiting* the agent's past experience to select the best arm so far; 2) *exploring* more find a better arm. Upper confidence bound (UCB) (Auer, 2002) is one of the most effective bandit algorithms. It selects the action with highest upper confidence bound $a_t = \arg\max_a(\hat{r}_{t,a} + c_{t,a})$, where $\hat{r}_{t,a}$ is the mean reward of action a and $c_{t,a}$ is the confidence interval. Contextual bandits consider the contextual information s_t to take an action which is particularly useful in many real-world problems such as personalized recommendation (Li et al., 2010). It takes actions with highest upper confidence bound by considering contextual information: $a_t = \arg\max_a(s_t^\top \theta_a + c_{t,a})$, where θ_a is parameter of the arm.

2.2. Imitation Learning

Generally, given a set of experts' demonstration trajectories τ, which consists of sequences of states and actions $(s_1, a_1, s_2, a_2, ...)$ drawn from the expert policy π_E, the goal of imitation learning is to learn a policy $\pi_\theta(a|s)$ which can replicate experts' behaviors. The imitation learning methods can be generally grouped into three categories: behavior cloning (BC), inverse reinforcement learning (IRL) and adversarial imitation learning (AIL). In this paper, we focus on BC and AIL which learn policies directly from the demonstrations.

2.2.1. Behavior Cloning (BC)

BC (Bain & Sammut, 1995) aims to learn the policy $\pi_\theta(a|s)$ via supervised learning. Given the fixed action space or classes, BC learns a policy mapping from states to experts' actions with the tuple datasets $\{(s_1, a_1), (s_2, a_2), ...\}$,

$$\arg\min_\theta \mathbb{E}_{(s,a)\sim P^*} L(a, \pi_\theta(s)),$$

where $P^* = P(s|\pi^*)$ is the distribution of states visited by expert. Due to the standard *i.i.d.* assumption in the supervised learning, the errors induced by BC are compounding over the length of the trajectories.

2.2.2. Adversarial Imitation Learning

Adversarial imitation learning (Ho & Ermon, 2016) directly learns π_θ by minimizing the Jensen-Shannon divergence between expert's policy π_E and the learned policy π_θ,

$$D_{JS}(\rho_{\pi_\theta}, \rho_{\pi_E}) = D_{KL}(\rho_{\pi_\theta} \| \frac{\rho_{\pi_\theta} + \rho_{\pi_E}}{2}) + D_{KL}(\rho_{\pi_E} \| \frac{\rho_{\pi_\theta} + \rho_{\pi_E}}{2}),$$

where the occupancy measure $\rho_\pi = \pi(a|s)\sum_{t=0}^{T}\gamma P(s_t = s|\pi)$ is the distribution of state-action pairs that the policy π interacts with the environment. γ is the discounting factor, and the successor states are drawn from $P(s|\pi)$. AIL utilizes a generative adversarial network to minimize the Jensen-Shannon divergence via a generator π_θ and a discriminator $D(\cdot)$ with the following objective function:

$$\max_{D\in(0,1)^{\mathcal{S}\times\mathcal{A}}} \mathbb{E}_{\rho_{\pi_\theta}}[\log(D(s,a))] + \mathbb{E}_{\rho_{\pi_E}}[\log(1-D(s,a))],$$

where $\mathcal{S}$ is the state set, and $\mathcal{A}$ is the action set.

3. Approach

3.1. The Formulation of HIL

The architecture of HIL is motivated by the dynamic treatment regimes. Different sub-policies can be viewed as treatment skills for different symptoms and the high-level policy acts as a scheduler of the sub-policies. As depicted in Fig. 1, HIL consists of two components: 1) High-level policy π^h learning with contextual bandit to compose multiple high-level actions $\{a_k^h\}_{k=1}^K$ (e.g., symptom treatments). Each of the high-level action corresponds to one sub-policy π_k^l; 2) Sub-policy learning via mimicking low-level actions a^l (e.g., medications). The high-level policy π^h is implemented with a K-arm contextual bandit, where K is the number of arms or high-level actions. Let T be the number of time steps and $r_{t,k} \in [0,1]$ is the reward of high-level action $a_{t,k}^h$, $t \in [1,T]$. At time step t, the agent observes the current state $s_t \in \mathbb{R}^d$, selects a high-level action $a_{t,k}^h$ and receives the reward $r_{t,k}$. Conditioned on the selected high-level action $a_{t,k}^h$, the agent chooses k-th sub-policy π_k^l which generates an primitive action a_t^l conditioned on s_t. The goal of the high-level policy π^h is to find an effective way to compose the sub-policies π_k^l to mimic the expert demonstrations. With the linear realizability assumption of the reward (Li et al., 2010), there exits K unknown weight vectors $\Theta = \{\theta_1^*, \theta_2^*, ..., \theta_K^*\}$ for each high-level action, $\theta_k^* \in \mathbb{R}^d$ with $\| \theta_k^* \| \leq 1$ and $r_{t,k} \in [0,1]$ such that,

$$\mathbb{E}[r_{t,k}|s_t] = s_t^\top \theta_k^*. \tag{1}$$

Let s_t be the state vector at t, we define the regret of the high-level policy π^h to be,

$$regret = \mathbb{E}[\sum_{t=1}^{T} r_{t,k^*}] - \mathbb{E}[\sum_{t=1}^{T} r_{t,k}], \tag{2}$$

where k^* is index of the best high-level action at step t and k indicates the index of the best high-level action selected by the agent. Our goal is to learn an optimized high-level policy and sub-policies by minimizing the regret.

3.2. High-level Policy Learning with Contextual Bandits

We extend the traditional imitation learning framework by assuming there exits finite sub-policies with different behaviors $\Pi = \{\pi_k^l\}_{k=1}^K$. HIL learns a high-level policy π^h parameterized with $\Theta = \{\theta_1, ..., \theta_K\}$ to compose these K sub-policies via contextual bandits algorithm, and the agent learns the K sub-policies by imitation learning methods such as behavioral cloning and adversarial imitation learning. Given the expert demonstrations $\tau = (s_1, a_1, ..., s_T, a_T)$, HIL selects a set of high-level actions $a_{t,k}^h$ which corresponds to the sub-policies π_k^l to take primitive actions and generates the trajectory $(s_1, a_{1,k}^h, a_1^l ..., s_T, a_{T,k}^h, a_T^l)$. As mentioned in Eq. (1), given the state $s_t \in \mathbb{R}^d$, we assume the expected reward of arm $a_{t,k}^h$ is linear with the state s_t with some unknown coefficients $\theta_k^* \in \mathbb{R}^d$. Let $X_{t,k} \in \mathbb{R}^{m\times d}$ be the matrix of the observed states, where rows indicate the historical m observed states up to time step t. In addition, $R_{t,k} \in \mathbb{R}^m$ is used to denote the reward vector, where each element indicates the reward $r_{t,k}$ (described in Eq. (7) and (10)) obtained by high-level action $a_{t,k}^h$. Our goal is to learn the unknown parameters θ_k to minimize the difference between the estimated rewards $X_{t,k}\theta_{t,k}$ and the observed rewards $R_{t,k}$,

$$\mathcal{L}_{\theta_k} = \|X_{t,k}\theta_k - R_{t,k}\|_2^2 + \lambda\|\theta_k\|_2^2, \tag{3}$$

where $\lambda > 0$ regulates the weight decay term. Applying ridge regression to the training data $(X_{t,k}, R_{t,k})$ gives the estimation of the arm parameters:

$$\theta_k = (X_{t,k}^\top X_{t,k} + \lambda I_{t,k})^{-1} X_{t,k}^\top R_{t,k}, \tag{4}$$

where $I_{t,k} \in \mathbb{R}^{d\times d}$ is an identity matrix. At each step t, we select the high-level action with the largest upper confidence bound, which is described as follows,

$$a_{t,k}^h = \arg\max_{a_{t,i}^h, i\in\{1,2,...,K\}} (s_t^\top \theta_k + \alpha\sqrt{s_t^\top A_k^{-1} s_t}) \tag{5}$$

where $A_k = X_{t,k}^\top X_{t,k} + I_{t,k}$, α is a constant to determine the confidence interval. With the assumption that the expectation of reward signals is linear with the state s_t, we can obtain the probability $1-\delta$ to choose the valid sub-policy for the state s_t such that the regret described in Eq. (2) can be bounded in a small value, as shown in the "Theoretical Properties" Section in Appendix.

Smoothness Constraint. With the observation that states are changing gradually, HIL encourages smoothness in the sequential sub-policy selection with a smooth operator $\Delta(\cdot)$,

$$\Delta_\beta(a_{t,k}^h) := \left\{ \begin{array}{ll} a_{t-1,k}^h; & \mathrm{UCB}_k \leq \beta \\ a_{t,k}^h; & \mathrm{UCB}_k > \beta \end{array} \right.$$

where $\beta \geq 0$ is the reward threshold and $\text{UCB}_k = s_t^\top \theta_k + \alpha\sqrt{s_t^\top A_k^{-1} s_t}$ is the upper confidence bound of the k-th sub-policy. When the upper confidence bound of the selected arm (sub-policy) is larger than β, we update the arm selection to $a_{t,k}^h$, otherwise we keep the previous selection $a_{t-1,k}^h$.

3.2.1. Discussion.

Previous hierarchical imitation learning and reinforcement learning researches learn the high-level policy by formulating it as a Markov decision process problem. However, by leveraging contextual bandits in high-level policy learning, HIL has the following advantages,

- *Simplify the environment*: K-armed bandit problem offers simplified environment compared with reinforcement learning. The simplification is offered by the “immediate” nature of the action, because the “immediate” reward can be directly used to evaluate the action. In reinforcement learning, action taken at a given time can affect expected reward in the future.

- *Avoid “credit assignment” problem*: If RL is used in high-level policy learning, the high-level actions and low-level actions would both affect the state transition, it is hard to assignment the future rewards to these two types of actions.

3.3. Sub-policy Learning via Imitation

The second phase of HIL is to learn sub-policies with imitation learning. HIL can incorporate any imitation learning method to mimic the expect demonstrations. Here we instantiate the proposed HIL model with behavioral cloning and adversarial imitation learning.

HIL with Behavioral Cloning. Behavior cloning suffers from distribution mismatch between behavior policy in long-horizon task. The use of HIL framework can help reduce this compounding error with a shortened horizon because π^h allocates the sub-policies only with a limited number of states. Given expert trajectories $\tau = (s_1, a_1^l, s_2, ..., s_T)$ generated from policy π_E, the high-level policy π^h sequentially selects sub-policies to take an action, which explicitly breaks down τ into a set of sub-trajectories $(\tau^1, ..., \tau^K)$ for sub-policies. The agents collect the states from the allocated sub-trajectories by π^h. Let $\pi_{t,k}^l$ (parameterized with ϕ_k, $\phi_k \in \Phi$, $\Phi = \{\phi_1, ..., \phi_K\}$) indicates the agent selected by π^h at time step t, we jointly learn each sub-policy π_k^l by maximizing the log-likelihood,

$$\mathcal{L}_{\phi_k} = -\sum_{t=1}^{T} \log \pi_{t,k}^l(a_t^l | s_t), \tag{6}$$

As we mentioned in the “High-level Policy Learning” Section, imitation reward also serves as feedback to the high-level policy training. The imitation reward $r_{t,k}$ of HIL with behavioral cloning is obtained by the cross entropy,

$$r_{t,k} = -\sum_{v=1}^{V} \pi_E(a_v^l | s_t) \log \pi_{t,k}^l(a_v^l | s_t), \tag{7}$$

where a_v^l is the v-th action and V is the total number of actions. The model is trained with coordinate descent by jointly minimizing the regression loss defined in Eq. (3) of high-level policy π^h and the maximizing the likelihood defined in Eq. (6).

HIL with Adversarial Imitation Learning. Adversarial imitation learning mimics the expert policy by matching the distributions state-action pairs from expert policy and the learned policy (Ho & Ermon, 2016). The output of the discriminator can be considered as a reward signal.

Adversarial imitation learning is utilized to learn $\pi_{t,k}^l$ at time step t by minimizing the Jensen-Shannon divergence between expert’ policy π_E and the learned policy $\pi_{t,k}^l$,

$$\mathcal{L}_{\phi_k} = D_{KL}\left(\rho_{\pi_{t,k}^l} \| \frac{\rho_{\pi_{t,k}^l} + \rho_{\pi_E}}{2}\right) + D_{KL}\left(\rho_{\pi_E} \| \frac{\rho_{\pi_{t,k}^l} + \rho_{\pi_E}}{2}\right), \tag{8}$$

where $\rho_\pi(s, a^l) = \pi(a^l|s) \sum_{t=1}^T \gamma P(s_t = s|\pi)$ is the distribution of state-action pairs with policy π. γ is the discounting factor, and the successor states are drawn from $P(s|\pi)$. A three-layer generative adversarial network is used to minimize the Jensen-Shannon divergence via a generator π_k^l parameterized with ϕ_k and a discriminator D_ω with the following objective function,

$$\max_\omega \min_{\phi_k} \mathbb{E}_{\rho_{\pi_{t,k}^l}}\left[\log(D_\omega(s, a^l))\right] + \mathbb{E}_{\rho_{\pi_E}}[\log(1 - D_\omega(s, a^l))], \tag{9}$$

The reward $r_{t,k}$ can then be obtained from the discriminator as described below,

$$r_{t,k} = \log D_\omega(s_t, a_t^l|_{a_t^l = \pi_k^l(s_t)}). \tag{10}$$

Diversity Constraint. Diversified sub-policies help learn complex task with multiple modes or hierarchical structure (Eysenbach et al., 2018). Thus the reward function is designed to enforce diversity of the learned K sub-policies. The diversity constraint is defined as follow,

$$\begin{aligned} r_t^{div} &= \sum_{i=1}^{K} \sum_{j=1}^{K} \sum_{s} W(\pi_i^l(a^l|s), \pi_j^l(a^l|s)), \\ &= \sum_{i=1}^{K} \sum_{j=1}^{K} \inf_{\kappa \in \Pi(\pi_i^l(a^l|s), \pi_j^l(a^l|s))} \mathbb{E}_{(x,y)\sim\kappa}[\| x - y \|], \end{aligned} \tag{11}$$

where $W(\pi^l_i, \pi^l_j)$ is the Wasserstein distance (Villani, 2008) between π^l_i and π^l_j. $\Pi(\pi^l_i(a^l|s), \pi^l_j(a^l|s))$ denotes the set of joint distributions $\kappa(x, y)$ whose marginals are π^l_i and π^l_j, respectively. r^{div}_t is used as feedback to help update π^h where $R_{t,:} \leftarrow R_{t,:} + r^{div}_t$.

Full Objective Function. To summarize, the loss of HIL that we are minimizing is:

$$\mathcal{L}_{\text{HIL}} = \zeta \mathcal{L}_{\phi_k} + (1-\zeta)\sum_{k=1}^{K} \mathcal{L}_{\theta_k} \tag{12}$$

where ζ is the balancing factor. The training of HIL is summarized in Algorithm 1 for sub-policy and high-level policy learning. We simultaneously optimize the high-level policy π^h with parameters $\{\theta_1, ..., \theta_K\}$ and the low-level policies $\{\pi^l_k\}_{k=1}^K$ with parameters $\{\phi_1, ..., \phi_K\}$.

Algorithm 1 Policy Learning in HIL

```
Require: Trajectories $\tau$; buffer $\mathcal{B}_k$; horizon $T$; smooth factor $\beta$.
for $t = 1$ to $T$ do
    $a^h_{t,k} \leftarrow \pi^h(s_t, \beta, \Theta)$,
    Take the sub-policy $\pi^l_k$ which corresponds to high-level action $a^h_{t,k}$,
    Calculate imitation reward $r_{t,k}$ based on Eq. (7) or Eq. (10)
    ▷ sub-policy learning
    Update $\pi^l_k$ by the gradient $\mathbb{E}_{\mathcal{B}_k}[\nabla_{\phi_k} \log \pi^l_k(a^l|s)]$ or $\mathbb{E}_{\mathcal{B}_k}[\nabla_{\phi_k} \log \pi^l_k(a^l|s)]D(s, a^l)$,
    Calculate diversity reward $r^{div}_t$ among sub-policies with Eq. (11),
    ▷ high-level policy learning
    Update $\pi^h$ via $\theta_k = (X^\top_{t,k} X_{t,k} + \lambda I_{t,k})^{-1} X^\top_{t,k} R_{t,k}$.
end for

function $\pi^h(s_t, \beta, \Theta)$
    Calculate high-level action $a^h_{t,k}$ with Eq. (5),
    Select the sub-policy $\pi^l_k$ corresponding to $a^h_{t,k}$.
end function
```

4. Experiments

We evaluate HIL with the task of recommending dynamic treatments for patients on a public EHRs dataset MIMIC-III (Johnson et al., 2016). A dynamic treatment is a sequence of tailored treatment decision rules that specify how the treatments should be recommended through time according to the dynamic states of patients (Murphy, 2003; Chakraborty & Murphy, 2014).

Dataset. The experiments are conducted on a public EHRs dataset MIMIC-III (Johnson et al., 2016), which contains 43,000 patients in critical care units during 2001 and 2012. There are 6,695 distinct diseases and 4,127 drugs in MIMIC-III. We extracted the *Comorbidity* patients following the procedure in (Bajor & Lasko, 2016), and we extract *Sepsis* patients conforming to the Sepsis-3 criteria (Singer et al., 2016). The statistics of the extracted datasets used in this paper are summarized in Table 1. More details of data pre-processing can be found in Section B of Appendix.

Table 1. Dataset statistics

Dataset	#Patient	#Medication
Comorbidity	16,508	23
Sepsis	6,620	25

Metrics and Baselines. We evaluate the treatment policy using both long-term measurements and short-term metrics: 1) The long-term performance measurement, *mortality rate* and *expected reward*, for reporting the policy performance via off policy policy evaluation (Precup et al., 2001; Dudík et al., 2011) (OPPE). OPPE utilizes a set of previously-collected trajectories to estimate the value of the learned policy without interacting with the environment. In this paper we use the Doubly Robust Off-policy Value Evaluation (Jiang & Li, 2015) to obtain an unbiased estimation of the value of the learned policies. 2) We also consider mirror metrics for short-term evaluation, *AUC scores* and *Jaccard coefficient*, to evaluate the short-term (i.e., one-step) consistency with expert trajectories. We use macro and micro average of the AUC scores (denoted as MA-AUC and MI-AUC) and Jaccard coefficient for short-term evaluation. The mean Jaccard is defined as $\frac{1}{Q\times T}\sum_{q=1}^{Q}\sum_{t=1}^{T}\frac{|U_{m,t}\cap U^*_{m,t}|}{|U_{m,t}\cup U^*_{m,t}|}$, where Q is the number of patients and U is the set of medications recommended by agent.

HIL is compared with the state-of-the-art imitation learning methods and the treatment recommendation methods which are described as follows:

- Behavior Cloning (BC): BC bins the trajectories into state-action tuples and learns a policy from the demonstrations via supervised learning.
- SRL-RNN (Wang et al., 2018): It manually designs a sparse reward function that assigns $r_T = 15$ if a patient is discharged, and $r_T = -15$ if the patient dies; $r_t = 0$ when $t = 0, 1, ..., T-1$.
- D3Q (Raghu et al., 2017): This method designed a reward function as SRL-RNN and trains the policy via deep Q-learning.
- GAIL (Ho & Ermon, 2016): GAIL utilizes GAN to learn a policy directly via the reward signal produced by discriminator.
- Directed Info-GAIL (D-GAIL) (Sharma et al., 2018): Directed Info-GAIL considers the multiple modes in

Table 2. Model comparisons on Sepsis

Method	Mortality	Rewards	MI-AUC	MA-AUC	Jaccard
BC	0.459	3.93	0.886	0.745	0.781
SRL-RNN	0.422	3.79	0.801	0.625	0.601
D3Q	0.485	3.42	0.703	0.527	0.472
GAIL	0.267	4.25	0.802	0.621	0.599
D-GAIL	0.259	4.33	0.797	0.618	0.575
HIL$_A$	**0.206**	**5.58**	0.892	0.802	0.847
HIL$_{BC}$	0.217	5.46	**0.915**	**0.817**	**0.863**

Table 3. Model comparisons on Comorbidity

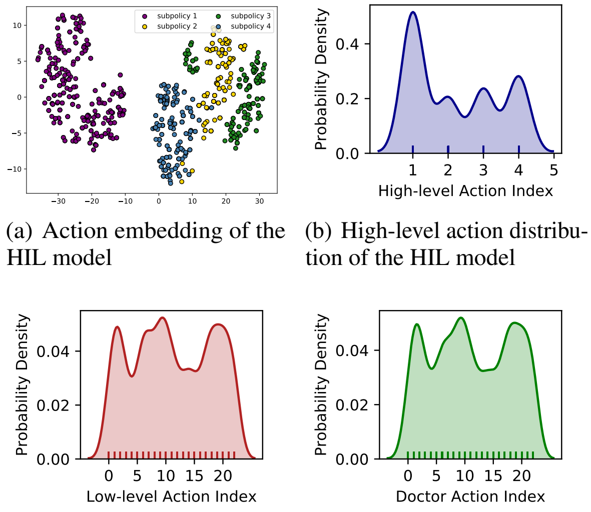
Method	Mortality	Rewards	MI-AUC	MA-AUC	Jaccard
BC	0.427	3.73	0.831	0.820	0.648
SRL-RNN	0.515	3.58	0.801	0.792	0.605
D3Q	0.539	3.27	0.692	0.688	0.514
GAIL	0.418	3.69	0.785	0.789	0.598
D-GAIL	0.411	3.75	0.787	0.791	0.604
HIL$_A$	**0.264**	**4.97**	0.881	0.872	0.708
HIL$_{BC}$	0.275	4.84	**0.893**	**0.883**	**0.719**

expert trajectories and mimic the expert policy by maximizing the directed information via the pre-generated high-level policies.

Our proposed models are denoted as HIL$_A$ and HIL$_{BC}$ with *adversarial imitation learning* and *behavior cloning*, respectively. To ensure fair comparisons, we set BC, HIL$_{BC}$, the actor network of SRL-RNN, GAIL, D-GAIL and HIL$_A$ with as a 3-layer neural network with the same active function and hidden variable size. The discriminator networks of GAIL, D-GAIL, SRL-RNN and HIL$_A$ are represented by a 3-layer MLPs with the same hidden variable size. The loss balancing factor ζ is set to 0.5.

4.1. Performance Evaluation

Table 2 and 3 summarize the performance of the baselines evaluated by the aforementioned metrics. We observe that: 1) Our models HIL$_A$ and HIL$_{BC}$ outperform other baselines on all metrics. Note that HIL$_{BC}$ utilizes the contextual bandits to select a subset of states for each sub-policy which shortens the horizon for each sub-policy and reduce the compounding errors (compared with BC). HIL$_A$ also explores the hierarchical structure of the expert demonstrations to compose multiple sub-policies in a meaningful way (compared with GAIL, SRL-RNN, D3Q and BC) and introduces diversity and smoothness constraints to achieve more effective sub-policies (compared with D-GAIL). 2) As for short-term performance (evaluated with MI-, MA-AUC and Jaccard), BC and HIL$_{BC}$ outperform other baselines by 12% to 30%. It is because BC considers the sequential treatments as i.i.d and mimics the expert's behaviors step-by-step. But BC fails in the evaluation of *mortality rate* and *expected rewards*. The reason is that BC ignores the long-term goal and introduces compounding errors. 3) D-GAIL and GAIL perform relatively good compared with other baselines in mortality rate and expected rewards. They learn to match the whole trajectory distributions between the learned policy and the expert's policy and consider the discriminator as a reward function to alleviate the sparse reward problem.



(a) Action embedding of the HIL model

(b) High-level action distribution of the HIL model

(c) Overall action distribution of the HIL model

(d) Overall action distribution of Doctors' policy

Figure 2. Visualization the actions (multi-hot vectors) generated by doctors and HIL in the Comorbidity dataset. Different colors in (a) indicate the action embedding generated by different sub-policies. (b-c) The high-level and overall action distributions of HIL. (d) The action distributions of the doctors in the test dataset. HIL can learn diversified sub-policies which can be clearly distinguished in (a) and the actions taken by these sub-policies match doctors' behaviors as shown in (c) and (d).

4.2. Visualization of the Learned Policies

Learned Policies on Comorbidity. In this part, we study the policies learned by HIL on Comorbidity test dataset. The action in Comorbidity dataset is a combination of medications (a multi-hot vector). The doctor gave a set of medications for the patients at different time steps. We collect the actions predicted by HIL on test dataset and plot their embedding and frequency.

From Fig. 2(a), we observe that HIL can learn diversified sub-policies where different sub-policies generate clear action embedding clusters. Fig. 2(b) and Fig. 2(c) are distributions of high-level actions and low-level actions generated by HIL, where x-axis is the action index, and y-axis is the probability density of the action which specify the probability of taking that action. All these sub-policies (high-level actions) are frequently selected as indicated in Fig. 2(b). Fig. 2(c) and 2(d) show the *consistency* of HIL's policy and the doctors' policy, as we can observe that the distribution

Table 4. Learned sub-policies on Comorbidity

Sub-policy	Top Action ID	Explanation
1	{0,6,9,20}	blood pressure
2	{1,8,9,20,21}	cardiovascular
3	{0,1,17,18}	hypokalemia
4	{16,17}	nutrition deficiencies

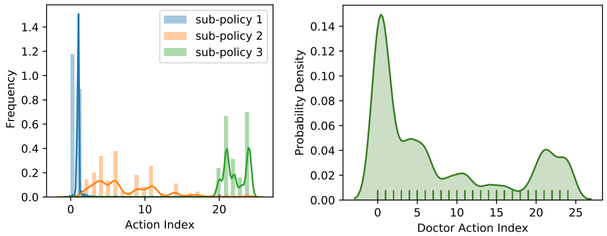


Figure 3. The medication distribution of HIL (left) and doctor's policy (right) in Sepsis

of the HIL' low-level action is roughly matching doctor's action distribution.

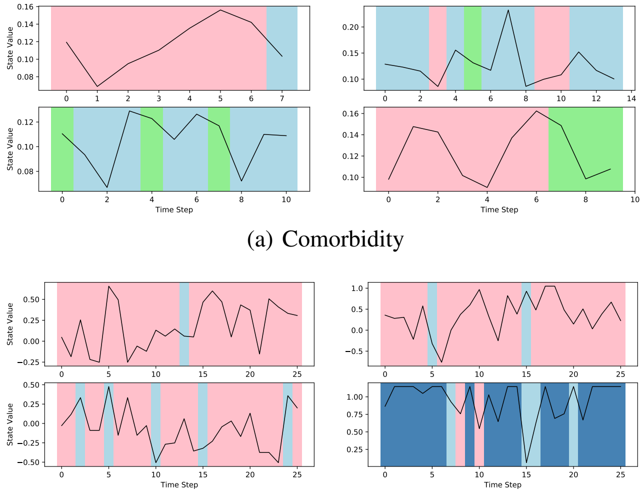
We further analyze the frequent actions used in each sub-policy, which are summarized in Table 4. We observe that different sub-policies may associate with different treatment focuses. For example, the *sub-policy 1* with frequent medication IDs $[0, 6, 9, 20]$ is mainly used to reduce the blood pressure of the patients. The details of the medication and action ID mapping can be found in Table 5 of Appendix.

Table 5. Learned sub-policies on Sepsis

Sub-policy	Top Action ID	Explanation
1	{0,1}	low-dose
2	{4,6,11}	mid-dose
3	{21,24}	high-dose

Learned Policies on Sepsis. In this part, we study the policies of HIL on the Sepsis dataset. The actions in Sepsis are one-hot vectors. Thus we only consider the action frequencies. First we evaluate the *consistency* of the learned policy and the doctors' policy. It can be seen from Fig. 3 that the learned sub-policies can be roughly divided into three categories: low-level dosage (*Sub-policy 1* with actions 0-1), mid-level dosage (*Sub-policy 2* with actions 2-20) and high-level dosage (*Sub-policy 3* with actions 20-24), as summarized in Table 5. Fig. 3 also demonstrates that the actions distribution generated by HIL is consistently with doctors' behavior.

Policy Smoothness. This part studies the smoothness of the actions generated by the high-level policy. We select the samples that HIL takes the same actions as the doctors' in test datasets, so that they have the same state transitions. Fig. 4 shows that the HIL's high-level policy takes high-level actions (sub-policies) smoothly. Additionally, we can observe some interesting patterns from Fig 4(a) that when π^h takes the sub-policy colored in red, the state values roughly tend to increase. In contrast, when it takes the sub-policy colored in green, the state value tend to decrease. Similar patterns can be found in Fig 4(b) on Sepsis treatment. π^h is more likely to take the sub-policy colored in red when the state value is lower than 0.5. However, it takes the sub-policy colored in dark blue when the value is above about 0.75.



(a) Comorbidity

(b) Sepsis

Figure 4. High-level action visualization. Different colors represent different high-level actions. The black curve shows the symptom of the patient.

4.3. Ablation Study

Number of High-level Actions. Results with different high-level action numbers (from 1 to 7) in Comorbidity dataset are shown in Fig. 5, where x-axis is the number of high-level actions, and y-axis represents the metric values. We observe that the best performance is achieved when the number of high-level agents is set to 4. The performance improves when the sub-policy number increases from 2 to 4, which shows the benefits of shrinking cluster size. However, when we further increase sub-policy number from 5 to 7, the performance of HIL drops. The reason is that, as we increase the number of sub-policies (high-level actions), there will be more redundant sub-policies which adds the complexity of training. After sub-policy number tuning, we set the sub-policy number to 4 for the follow-up experiments on Comorbidity dataset. Similarly, we set the sub-policy number to 3 for the Sepsis dataset as the model performs the best under this setting.

Sample Complexity. To analyze HIL's sensitivity to the number of expert sample size, we tested the training dataset

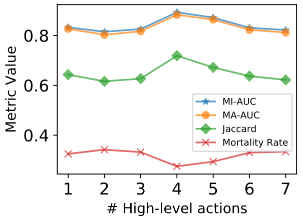


Figure 5. Performance of HIL on different number of high-level actions

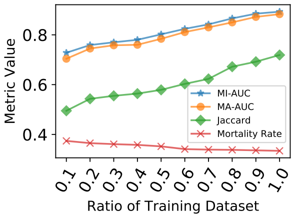
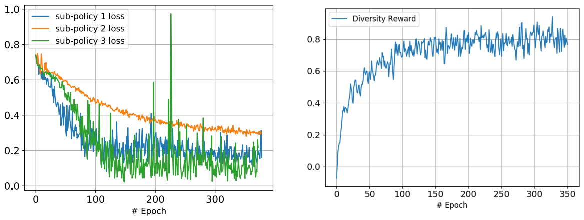


Figure 6. Performance of HIL on different training ratios of Comorbidity dataset



(a) Sub-policy loss

(b) Reward Diversity

Figure 7. Convergence analysis

proportion from 10% to 100% with a step of 10%. Fig. 6 depicts the changes of Jaccard, MI-, MA-AUC and mortality rate with different ratio of training dataset. As expected, the performance of HIL on the Comorbidity dataset increases with more expert trajectories because of more information can be used to mimic the experts' behavior. Notably, HIL can still achieved satisfactory results with only 10% of the training dataset.

4.4. Convergence Analysis

Fig. 7(a) and 7(b) show the loss of the discriminators of different sub-policies, and the diversity reward of the contextual bandits on Sepsis dataset. We observe that both high-level policy and low-level policy training are converged. In terms of the sub-policy training, typically around 100 to 200 iterations are necessary to yield a useful approximation. Similar converge patterns can be found in the reward diversity plot in Fig. 7(b).

5. Related Work

Imitation Learning Imitation learning, also known as *learning from demonstrations*, can be generally grouped into three categories: behavior cloning (BC), inverse reinforcement learning (IRL) and adversarial imitation learning (AIL). BC (Pomerleau, 1991) is a form of supervised learning which is to learn a direct mapping from the states to the actions. BC can avoid interacting with the environment. However, without substantial correction during training, BC is known to have compounding error (Ross et al., 2011). Instead of directly utilizing the supervised signal from demonstrations, inverse reinforcement learning (Zhifei & Meng Joo, 2012; Abbeel & Ng, 2004) finds a reward function that models the intention of the demonstrator. The learned reward function gives feedback to the states that were un-visited. A policy can be learned by reinforcement learning methods (Sutton et al., 1998) with this reward function. Maximum entropy IRL (Ziebart et al., 2008; Wulfmeier et al., 2015; Finn et al., 2016) seeks to find a reward function that makes the demonstrations achieve highest total reward as well as maximize the entropy of the resultant policy. Adversarial imitation learning (Ho & Ermon, 2016) leverages generative adversarial networks (Goodfellow et al., 2014) to directly learn the policy and reward function simultaneously, where the policy corresponds to the generator and the discriminator plays as the reward function. In order to learn from complex expert trajectories, a set of prior work aims to decompose demonstrations into primitive sub-policy or skills, e.g. using segmentation labels (Le et al., 2018), latent variable models (Sharma et al., 2018; Hausman et al., 2017; Kipf et al., 2019). Unlike these approaches which learn sub-policies for individual tasks, we jointly learn the high-level policies and sub-policies.

Hierarchical Reinforcement Learning Long-term decision making is an important problem in traditional RL domains. The hierarchical reinforcement learning (HRL) is developed to decompose the long-term decision process into hierarchically structured short-term decision processes so that the policy can be organized along the hierarchy to manipulate the agent's behavior at multiple control levels. For example, some HRL methods use some domain knowledge to design a hierarchy over the actions (Parr & Russell, 1998; Sutton et al., 1999; Dietterich, 2000) to reduce the search space. Another branch of HRL methods adopts options (Sutton et al., 1999; Dietterich, 2000; Precup, 2000), which uses a two-layer hierarchy containing a manager policy to choose option and many sub-policies corresponding to different options. A typical example is the Feudal network (Vezhnevets et al., 2017) that learns a goal (option) embedding and computes some intrinsic rewards based on the goal to motivate the agent to act. However, the above HRL methods focus on reinforcement learning setting, which requires to interacting with the environment and accessing to the reward signal.

6. Conclusion

In this paper, we present the hierarchical imitation learning model, HIL, to handle complex tasks with hierarchical structures. Unlike existing methods that learn sub-policies for sub-tasks with prior knowledge in high-level policies for sub-task switching, HIL jointly learns latent high-level policies and sub-policies from expert demonstrations without accessing prior knowledge. Using real-world medical data, we empirically show that HIL is capable to learn meaningful high-level policies and sub-policies that accurately reproduce complex expert trajectories. Compared with state-of-the-art baselines, HIL could further improve the likelihood of patient survival and provide better dynamic treatment regimes with the exploitation of hierarchical structures in expert demonstrations.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1. ACM, 2004.

Ahilan, S. and Dayan, P. Feudal multi-agent hierarchies for cooperative reinforcement learning. *arXiv preprint arXiv:1901.08492*, 2019.

Auer, P. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

Bain, M. and Sammut, C. A framework for behavioural cloning. In *Machine Intelligence*, pp. 103–129, 1995.

Bajor, J. M. and Lasko, T. A. Predicting medications from diagnostic codes with recurrent neural networks. 2016.

Chakraborty, B. and Murphy, S. A. Dynamic treatment regimes. *Annual review of statistics and its application*, 1:447–464, 2014.

Dietterich, T. G. Hierarchical reinforcement learning with the maxq value function decomposition. *JAIR*, pp. 227–303, 2000.

Dudík, M., Langford, J., and Li, L. Doubly robust policy evaluation and learning. In *ICML*, pp. 1097–1104, 2011.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Finn, C., Levine, S., and Abbeel, P. Guided cost learning: Deep inverse optimal control via policy optimization. In *ICML*, pp. 49–58, 2016.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Hausman, K., Chebotar, Y., Schaal, S., Sukhatme, G., and Lim, J. J. Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets. In *NIPS*, pp. 1235–1245, 2017.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pp. 4565–4573, 2016.

Jiang, N. and Li, L. Doubly robust off-policy value evaluation for reinforcement learning. *arXiv preprint arXiv:1511.03722*, 2015.

Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

Kipf, T., Li, Y., Dai, H., Zambaldi, V., Sanchez-Gonzalez, A., Grefenstette, E., Kohli, P., and Battaglia, P. Compile: Compositional imitation learning and execution. In *International Conference on Machine Learning*, pp. 3418–3428, 2019.

Le, H. M., Jiang, N., Agarwal, A., Dudík, M., Yue, Y., and Daumé III, H. Hierarchical imitation and reinforcement learning. *arXiv preprint arXiv:1803.00590*, 2018.

Li, L., Chu, W., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670, 2010.

Murphy, S. A. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(2):331–355, 2003.

Parr, R. and Russell, S. J. Reinforcement learning with hierarchies of machines. In *NeurIPS*, pp. 1043–1049, 1998.

Pomerleau, D. A. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, 1991.

Precup, D. *Temporal abstraction in reinforcement learning*. University of Massachusetts Amherst, 2000.

Precup, D., Sutton, R. S., and Dasgupta, S. Off-policy temporal-difference learning with function approximation. In *ICML*, pp. 417–424, 2001.

Raghu, A., Komorowski, M., Celi, L. A., Szolovits, P., and Ghassemi, M. Continuous state-space models for optimal sepsis treatment-a deep reinforcement learning approach. *arXiv preprint arXiv:1705.08422*, 2017.

Rizzolatti, G., Fogassi, L., and Gallese, V. Neurophysiological mechanisms underlying the understanding and imitation of action. *Nature reviews neuroscience*, 2(9): 661–670, 2001.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635, 2011.

Sharma, A., Sharma, M., Rhinehart, N., and Kitani, K. M. Directed-info gail: Learning hierarchical policies from unsegmented demonstrations using directed information. *arXiv preprint arXiv:1810.01266*, 2018.

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

Sutton, R. S., Barto, A. G., et al. *Introduction to reinforcement learning*, volume 2. MIT press Cambridge, 1998.

Sutton, R. S., Precup, D., and Singh, S. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2): 181–211, 1999.

Tang, H., Hao, J., Lv, T., Chen, Y., Zhang, Z., Jia, H., Ren, C., Zheng, Y., Fan, C., and Wang, L. Hierarchical deep multiagent reinforcement learning. In *AAAI*, 2018.

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. Feudal networks for hierarchical reinforcement learning. In *ICML*, pp. 3540–3549, 2017.

Villani, C. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Wang, L., Zhang, W., He, X., and Zha, H. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In *SIGKDD*, pp. 2447–2456. ACM, 2018.

Wulfmeier, M., Ondruska, P., and Posner, I. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888*, 2015.

Zhifei, S. and Meng Joo, E. A survey of inverse reinforcement learning techniques. *International Journal of Intelligent Computing and Cybernetics*, 5(3):293–311, 2012.

Ziebart, B. D., Maas, A., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. 2008.

Appendix

A. Theoretical Properties

In this section we show that the high-level policy of HIL can select the valid sub-policy with probability $1-\delta$ for any $\delta > 0$ (Li et al., 2010). With probability at least $1-\delta$, $|s_t^\top \hat{\theta}_k - \mathbb{E}[r_{t,k}|s_t]| \leq \alpha\sqrt{s_t^\top (X_{t,k}^\top X_{t,k} + \lambda I_{t,k})^{-1} s_t}$ for any $\delta > 0$ and $s_t \in \mathbb{R}^d$ where $\alpha = 1 + \sqrt{\ln(2/\delta)/2}$ is a constant.

We will prove it with Hoeffding inequality. We first interpreted the ridge regression problem described in Eq. (3) as a Bayesian point estimation, where the prior distribution of θ_k is Gaussian distribution $p(\theta|\alpha) = \mathcal{N}(\theta|0, \lambda I)$ and λ is hyperparameter to control the distribution of model parameters. The posterior distribution of θ is with mean $\hat{\theta}$ and covariance A_k^{-1}, $A_k := X_{t,k}^\top X_{t,k} + \lambda I_{t,k}$. Given the current model θ, the predictive variance of the expected reward $s_t^\top \theta_k^*$ is evaluated as $s_t^\top A_k^{-1} s_t$ and $\sqrt{s_t^\top A_k^{-1} s_t}$ is the standard deviation.

Hoeffding Inequality . Let $X_1, X_2, ..., X_n$ be i.i.d observations such that $\mathbb{E}(X_i) = \mu$ and $a \leq X_i \leq b$. Then, for any $\epsilon > 0$, $P(|\bar{X} - \mu| > \epsilon) < 2\exp(-2n\epsilon^2/(b-a)^2)$.

In this paper, we set the reward signal $\mathbb{E}[r_t]$ in the range $[0,1]$ in order to bound the probability $P(|s_t^\top \hat{\theta}_k - \mathbb{E}[r_{t,k}|s_t]| > \epsilon)$ via Hoeffding inequality for all the arms, where the variance of the estimated reward is $\sigma^2 = s_t^\top A_k^{-1} s_t$.

Hoeffding inequality tells us that for any $\epsilon > 0$,

$$P(|s_t^\top \hat{\theta}_t - \mathbb{E}[r_{t,k}|s_t]| > \epsilon) < 2\exp(-2n\epsilon^2/\sigma^2) < \delta. \tag{13}$$

Based on the aforementioned inequality, we can drive the lower bound of ϵ as follows,

$$\epsilon \geq \sigma\sqrt{1/2\ln(1/\delta)}. \tag{14}$$

The ϵ in Eq. (13) can be replaced by $\sigma\sqrt{1/2\ln(1/\delta)}$ and we have,

$$P\left(|s_t^{\mathrm{T}} \hat{\theta}_k - \mathbb{E}[r_{t,k}|s_t]| \leq \sigma\sqrt{1/2\ln(1/\delta)}\right) > (1-\delta) \tag{15}$$

This concludes the proof.

The above theorem insures a reasonably tight upper confidence bound where π^h can be derived. That is, at trial t, the following arm is a valid sub-policy to be taken,

$$a_{t,k}^h = \arg\max_{k\in[0,K]} (s_t^\top \hat{\theta}_k + \alpha\sqrt{s_t^\top A_k^{-1} s_t}), \tag{16}$$

where $s_t^\top \theta_k$ is the mean of the estimated reward and $\sqrt{s_t^\top A_k^{-1} s_t}$ is the standard deviation, $a_{t,k}^h$ is the arm which achieves highest upper confidence bound at step t.

B. Data Pre-Processing

For each patient, we extract relevant physiological parameters with the suggestions from clinicians, which include static variables and time-series variables, as summarized in Table 6. These variables are first rescaled to z-scores, then rescaled to $[0,1]$. We impute the missing variable with k-nearest neighbors and remove admissions with more than 10 missing variables. Each hospital admission of a patient is regarded as a treatment plan. Time-series data in each treatment plan is divided into different units, each of which is set to 4 hours since it is the median of the prescription frequency in MIMIC-III. If several data points are in one unit, we use their average values instead. We remove patients less than 18 years old because of the special conditions of minors. Finally, we obtain 20,193 hospital admissions of comorbidity patients (16,508 survived patients and 3,685 deceased patients). And 10,189 sepsis patients (6,620 survived patients and 3,569 deceased patients). This paper only uses survived patients for the evaluations. Specifically, we distilled the following information from data, such as,

$$([\texttt{patient ID}], [\texttt{state}], [\texttt{Medication}], [\texttt{time}]),$$

where *state* consists of demographic feature, clinical variables. The *action* of Comorbidity dataset is a multi-hot vector where each entry indicates the medication usage. The *action* of Sepsis dataset is a one-hot vector with a length of 25, as described in Table 6.

C. Experimental Details

Settings. In this paper, the following neural network architectures are used for the implementation,

- The actor networks of HIL$_A$ and HIL$_{BC}$ are a three-layer MLP with hidden node size 2011-128-128-64-23 (Comorbidity) and 49-128-64-32-25 (Sepsis).
- The discriminator network of HIL$_A$ and HIL$_{BC}$ is a three layer MLP with size 2034-128-128-64-1 (Comorbidity) and 74-128-64-32-1 (Sepsis).
- HIL is optimized by Adam, with learning rate 6e-04
- Loss balance weight ζ is set as 0.5. on the Actor and 6e-05 on the critic.

D. Additional Results

The action distribution of each Comorbidity's sub-policy is shown in Fig. 8, which presents a detailed description of

Table 6. Description of demographics, clinical variables and medications

Demographics	gender, age, weight, height, religion, language, marital status, and ethnicity
Lab Tests & Vital Signs	**Comorbidity:** diastolic blood pressure, Glasgow coma scale, urine output, systolic blood pressure, fraction of inspired O_2, heart rate, pH, respiratory rate, blood glucose, body temperature, blood oxygen saturation, and blood glucose **Sepsis:** Shock Index, Elixhauser, SIRS,Re-admission, GCS - Glasgow Coma Scale, SOFA - Sequential Organ Failure Assessment, 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,Diastolic Blood Pressure, Systolic Blood Pressure, Mean Blood Pressure, PaCO2, PaO2, FiO2, PaO/FiO2 ratio, Respiratory Rate, Temperature (Celsius), Weight (kg), Heart Rate, SpO2, Fluid Output - 4 hourly period, Total Fluid Output, Mechanical Ventilation
Medications	**Comorbidity (ID: Medication):** 1: Furosemide, 2: Potassium, 3: Chloride, 4: Sodium, 5: Chloride, 6: Acetaminophen$_1$, 7: Lorazepam, 8: Heparin, 9: Docusate Sodium, 10: Bisacodyl, 11: Acetaminophen$_2$, 12: Magnesium Sulfate, 13: Senna, 14: Aspirin, 15: Pantoprazole, Calcium 16: Gluconate, Dextrose 50%, 17: Fentanyl Citrate, 18: Docusate Sodium, 19: Benicar, 20: Bisacodyl, 21: Potassium Chloride, 22: Ropinirole, 23: Acetaminophen$_3$ **Sepsis Medication**: five dosages (0-4) of IV fluid and Vasopressor, where 0 represents no drug given, and 4 represents the maximum dosage. We flat this 5 × 5 space into 25 actions with IDs 0-24 (0: Vasopressor = 0, IV fluid = 0; 1: Vasopressor = 1, IV fluid = 0; ...)

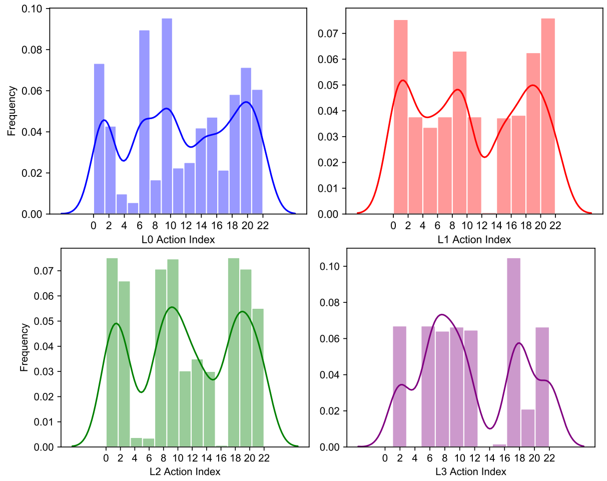
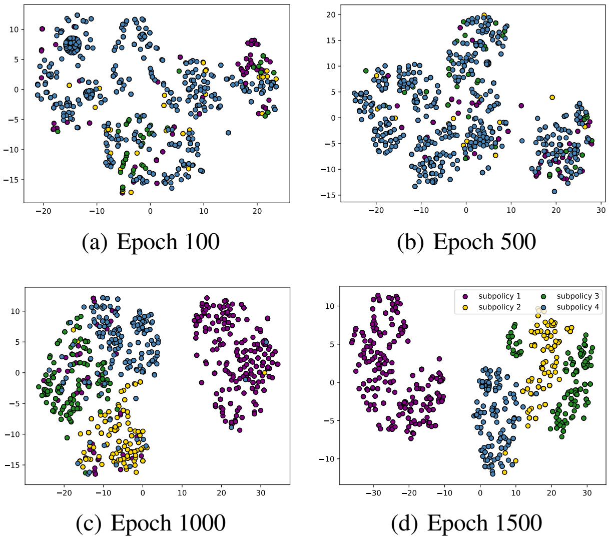


Figure 8. Medication distribution generated by HIL's sub-policies with Comorbidity dataset



(a) Epoch 100

(b) Epoch 500

(c) Epoch 1000

(d) Epoch 1500

Figure 9. Visualization of sub-policy learning process at different training epochs with Comorbidity dataset. Different colors represent different sub-policies learned by HIL.

Table 4. It demonstrated that HIL can learn sub-policies which focus on different actions.

We also visualize the sub-policy learning process by plotting the action embedding at different time steps with Comorbidity dataset. From Fig. 9 we observe that in the early stage, there is a large overlap among the actions from different sub-policies. As we have more training epochs, the actions generated by different sub-policies form clusters, which indicate that we have diversified sub-policies learned.