B ALGORITHMS

Algorithm 1 BC-SDRPRL: Pre-Training and Initialization

STEP I: Offline Pre-Training (Behavioral Cloning)
Input:
B^E: Expert trajectory (S^E, A^E)
T: Pre-training steps
Randomly initialize: Actor network $a^A(s \mid \theta_\pi)$ and its target $a^{A'}(s \mid \theta_{\pi'})$ weights.
for pre-training steps = 1, ..., T **do**
Compute L_{BC} using eq. (1)
end for
Output:
$a^A(s \mid \theta_\pi)$: Actor-network
$a^{A'}(s \mid \theta_{\pi'})$: Target actor network

STEP II: Offline IOHMM Training (Specialization)
Input:
B^E: Expert trajectory (S^E)
n: Number of hidden states
while expectation-maximization tolerance $> 1 * 10^1$ **do**
Compute expectation-maximization ζ using eq. (3)
end while
Output:
λ: Trained IOHMM model parameters

STEP III: Offline Value Initialization & State Classification
Input:
B_e: Expert trajectory (S^E)
W: Warmup steps
Randomly initialize: Value network $V(s \mid \theta_V)$ weights.
for warmup steps = 1, ..., W **do**
Compute L_V using eq. (10)
end for
Output:
$V(s \mid \theta_V)$: Value network
$\hat{x}$: Hidden states classification based on the value function
x^*: Specialized hidden state (abnormal state)

STEP IV: Online Critic Initialization
Input:
env: Environment model
W: Warmup steps
Randomly initialize: Critic network $Q\left(\mathbf{s}, \pi\left(\mathbf{s} \mid \theta_\pi\right) \mid \theta_Q\right)$ and its target $Q'\left(\mathbf{s}, \pi\left(\mathbf{s} \mid \theta_\pi\right) \mid \theta_{Q'}\right)$weights.
Compute x_t using eq. (4).
for warmup steps = 1, ..., W **do**
if x_t is x_t^* **then**
Compute L_Q using eq. (7)
end if
end for
Output:
$\hat{Q}(x^*, s, \pi \mid \theta_Q)$: Critic network

Algorithm 2 BC-SDRPRL: Training and Inference

STEP I: Desired Hidden State Identification
Input:
s_t: State of the system
λ: IOHMM trained parameters
x^*: Specified hidden state
Compute x_t using eq. (4).
if x_t is x^* **then**
ϕ = True
end if
Output:
ϕ: Boolean activation for the residual policy learning

STEP II: Deep Residual Policy Reinforcement Learning
Input:
B^A: Initialize empty RL agent buffer
env: Environment model
$Q\left(\mathbf{s}, \pi\left(\mathbf{s} \mid \theta_\pi\right) \mid \theta_Q\right), a^A(s \mid \theta_\pi)$: Pre-trained networks
ϕ: Boolean activation for the residual policy learning
τ: Target network soft update hyperparameter
L: Training steps
Compute x_t using eq. (4).
for training steps = 1, ..., L **do**
if ϕ **then**
Take superposed action combining expert's action and residual action based on the current state s_t as in eq. (2)
Observe reward r and next state s_{t+1}
Add s_t, a_t^A, r_t, s_{t+1} in B^A
Sample batch of s_t, a_t^A, r_t, s_{t+1} from B^A
Compute L_Q and L_A using eq. (7) and eq. (9), respectively.
Update actor and critic networks using Adam optimizer.
Update target networks using eq. (15):

$$\begin{aligned} \theta_{\pi'} &\leftarrow \tau\theta_\pi + (1-\tau)\theta_{\pi'} \\ \theta_{Q'} &\leftarrow \tau\theta_Q + (1-\tau)\theta_{Q'} \end{aligned} \tag{15}$$

else
Take expert's action
end if
end for
Output:
$\pi_\theta^*(s_t)$: Optimal control policy from eq. (2)