Online Appendix to "Inpatient Overflow Management with Proximal Policy Optimization"

Jingjing Sun, J. G. Dai, and Pengyi Shi

Appendix A: Notations

$\mathcal{J}=\{1,2,\ldots,J\}$	set of pools with J being the number of pools
N_j	number of servers in pool j, $j\in\mathcal{J}$
$\lambda_j(t)$	arrival rate function of class j on a continuous-time scale
$\Lambda_j=\int_0^1\lambda_j(t)$	daily arrival rate
μ_j	probability of each customer in pool j to be discharged on the current day
h_{dis}	random variable for the time of leaving on the current day in case of departure, with CDF F_j for pool j
$t_0,t_1,\ldots,t_{m-1}$	decision epochs on day t, with m being the number of epochs per day
$S(t_k)=(X_1(t_k),\ldots,X_J(t_k)$ $,Y_1(t_k),\ldots,Y_J(t_k),h(t_k))\in\mathbb{N}^{2J+1}$	system state at decision epoch t_k (k-th decision time-point on day t)
$X_j(t_k)$	customer count in class j at decision epoch t_k
$Y_j(t_k)$	to-depart count in pool j at decision epoch t_k
$h(t_k)$	time-of-day indicator for decision epoch t_k
$\mathcal{S}$	state space
$\mathcal{S}^h$	sub-space which contains all states with time-of-day being h
$f(t_k)=\{f_{i,j}(t_k),\ i,j=1,\ldots,J\}$	overflow/wait decision at decision epoch t_k
$Q_j=(X_j-N_j)^+$	waiting count in class j
$Z_j=\max\{N_j,X_j\}$	in-service count in pool j
s	pre-action state
s^+	post-action state
$p(s'\|s,f)$	one-epoch transition probability from s to s' under action f
C_j	unit holding cost of class j waiting customers
$B_{i,j}$	unit overflow cost of assignments from class i to pool j
$g(s,f)$	one-epoch cost at state s under action f
$\pi(f\mid s)$	probability of choosing action f at state s according to policy $\pi\in\Pi$
$p_\pi(s'\|s)=\mathbb{E}_{f\sim\pi(\cdot\|s)}p(s'\|s,f)$	expected transition probability from s to s' under policy $\pi\in\Pi$
$g_\pi(s)=\mathbb{E}_{f\sim\pi(\cdot\|s)}g(s,f)$	expected one-epoch cost under policy $\pi\in\Pi$
$\mathbf{a}=\{a^1,a^2,\ldots,a^q\}$	atomic action sequence at state s, with a^n denoting the atomic action for the nth waiting customer
$\boldsymbol{\sigma}=\{\sigma^1(s),\ldots,\sigma^q(s)\}$	an order of waiting customer's classes, with $\sigma^n(s)$ denoting the class of the nth waiting customer
s^n	"atomic state" within the atomic decision process after making the assignment for the $n-1$th customer
$\mathcal{A}(s^n,\sigma^n(s))$	feasible atomic action space for a^n with atomic state s^n and customer class $\sigma^n(s)$

Table 6 Notations in the main paper (in the order of appearance).

Appendix B: Transition Dynamics under Randomized Policy

The transition dynamics are different for non-midnight epochs and midnight epoch.

Non-midnight epochs ($h\neq 0$). From the assumptions on the arrivals and departures in Section 3.1, the number of arrivals, denoted as a_j, is a realization from the random variable A_j^h, which follows $\text{Poi}\big(\int_h^{h'}\lambda_j(s)ds\big)$;

π_θ	randomized atomic policy parameterized by θ
$\kappa_\theta(a^n \mid s^n, \sigma^n(s))$	probability for choosing each a^n given the current state s^n and the customer class $\sigma^n(s)$ according to policy π_θ
$\pi_\theta(f \mid s)$	probability of choosing a system-level action f at state s under policy π_θ
$g_{j,k}(s\mid\theta)$	output of a fully-connected network with parameters θ and input s
γ_π	average cost under policy π
v_π	relative value function under policy π
A_π, A_η	advantage function under policy π and policy π_η
μ_θ	stationary distribution under policy π_θ
$r_{\theta,\eta}$	probability ratio between policies π_θ and π_η
$L(\theta)$	objective of PPO
$\text{clip}(x, 1-\epsilon, 1+\epsilon)$	clip function
$\hat{L}(\theta, \mathcal{D}_\eta^T, \hat{\mathbf{A}}_\eta(\mathcal{D}_\eta^T))$	estimated objective function using the simulation data collected under policy π_η, with $\mathcal{D}_\eta^n$ denoting trajectory of T simulated days generated under policy π_η, and $\hat{\mathbf{A}}_\eta(\mathcal{D}_\eta^T)$ denoting estimated advantage functions at each decision epoch of trajectory $\mathcal{D}_\eta^T$

Table 7 Notations for atomic policy and PPO (in the order of appearance).

and the number of departure, denoted as d_j, is a realization of the random variable D_j^h, which follows $\text{Bin}(y_j, p_j^h)$. Here, p_j^h represents the probability of a to-depart patient who is still in hospital at epoch h and will be discharged between h to h':

$$p_j^h = \frac{F_j(h') - F_j(h)}{1 - F_j(h)},$$

where $F_j(h)$ is the CDF for the discharge time.

From the transition dynamics in Equation (2), we can specify the transition probability given action f at a non-midnight epoch h as

$$p(s'|s,f) = \prod_{j=1}^{J} \mathbb{P}(D_j^h = y_j - y_j') \mathbb{P}\left(A_j^h = x_j' - \left(x_j + \sum_{i=1, i\neq j}^{J} f_{i,j} - \sum_{k=1, k\neq j}^{J} f_{j,k}\right) + y_j - y_j'\right), \quad h = 1, 2, \ldots, m-1.$$

Midnight epoch ($h = 0$). Compared with non-midnight epochs, the main differences of the dynamics in midnight epoch are: (i) there is no departure between midnight epoch and the next epoch as in Dai and Shi (2019); (ii) random draw of new to-depart patients who will leave in the coming day, i.e., y_j' is a realization of $B_j \sim Bin(\min\{x_j, N_j\}, \mu_j)$. The transition probability given action f at midnight epoch follows

$$p(s'|s,f) = \prod_{j=1}^{J} \mathbb{P}\left(A_j^0 = x_j' - x_j - \sum_{i=1, i\neq j}^{J} f_{i,j} + \sum_{k=1, k\neq j}^{J} f_{j,k}\right) \mathbb{P}(B_j = y_j'), \quad h = 0.$$

For a randomized policy π with the corresponding action probability $\pi(f \mid s)$ defined in (5), the state transition probability follows

$$p_\pi(s'|s) = \mathbb{E}_{f\sim\pi(\cdot|s)} p(s'|s,f)$$

Appendix C: PPO in Periodic Setting

Dai and Gluzman (2022) provides theoretical guarantee of performance improvement using the PPO method in the time-stationary, long-run average cost setting. However, extending it to the setting in our paper is not straightforward due to a key challenge: periodicity. In Section C.1, we elaborate on this challenge. To

address the periodicity, in Section C.2 we introduce the Markov chains (MC) with one period (e.g., one day) as one step under the periodic setting and redefine the objective functions accordingly. We call them as the *daily MC*s to contrast with the original MC where one epoch corresponds to an hour/few hours within a day (one period contains m epochs). In Section C.3, we prove that, under certain assumptions about the current policy and the updated policy, (i) the difference between the long-run average cost of the two policies can be decomposed into two terms; (ii) the decay rate of the second term is faster than that of the first term. Based on these results, we show in Section C.4 the improvement guarantee of PPO for the daily MCs.

C.1. The Key Challenge

There are two differences between the setting in Dai and Gluzman (2022) and our setting: (i) their per-epoch cost is action-independent (i.e., the action does not change the immediate holding cost), and (ii) the resulting DTMC under their considered policy class is irreducible and aperiodic. For (i), the overflow actions affect the per-epoch cost in our setting, which requires some different algebra in derivation. The critical difference lies in (ii). That is, our setting has time-varying periodicity in the state transitions. As a result, if we directly incorporate the epoch h (hour-of-day indicator) into the state s, the resulting DTMC becomes periodic. Specifically, a state $s^h \in \mathcal{S}^h$ can only transition to a state of the next epoch $s^{h+1} \in \mathcal{S}^{h+1}$ (recall that $\mathcal{S}^h$ denotes the subset of state space containing states with the time-of-day indicator h). Therefore, the results in Dai and Gluzman (2022) cannot be directly applied to our periodic environment.

C.2. Daily MC and Revised Objective

Given any policy π_η parameterized by η, we denote the transition matrix and cost vector of the original MDP, respectively, as $\mathbf{P}_\eta = \{p_\eta(s|s'), s, s' \in \mathcal{S}\}$ and $\mathbf{g}_\eta = \{g_\eta(s), s \in \mathcal{S}\}$. The matrix $\mathbf{P}_\eta$ has the following structure:

$$\mathbf{P}_\eta = \begin{pmatrix} 0 & \mathbf{P}_\eta^{0,1} & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{P}_\eta^{1,2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{P}_\eta^{m-2,m-1} \\ \mathbf{P}_\eta^{m-1,0} & 0 & 0 & \cdots & 0 \end{pmatrix}, \tag{16}$$

where $\mathbf{P}_\eta^{h,h'}$ denotes the sub-matrix for transitions from states in $\mathcal{S}^h$ to states in $\mathcal{S}^{h'}$ under policy π_η. The expected cost vector can be written as

$$\mathbf{g}_\eta = \begin{pmatrix} \mathbf{g}_\eta^0 \\ \vdots \\ \mathbf{g}_\eta^{m-1} \end{pmatrix},$$

where $\mathbf{g}_\eta^h$ denotes the one-epoch cost vector for states in $\mathcal{S}^h$.

To address the challenge of periodicity in the original MDP setting, we introduce the *daily MC*s, which are aperiodic. Under a given policy π_η, for each time-of-day indicator $h = 0, 1, ..., m-1$, if we observe states only at epoch h of each day, the resulting stochastic process $\{S(t_h), t = 0, 1, ...\}$ is still an MC. The corresponding *daily* transition matrix and expected cost vector are defined as

$$\begin{aligned} \tilde{\mathbf{P}}_\eta^h &= \mathbf{P}_\eta^{h,h+1}\mathbf{P}_\eta^{h+1,h+2}\cdots\mathbf{P}_\eta^{h-1,h}, \\ \tilde{\mathbf{g}}_\eta^h &= \mathbf{g}_\eta^h + \mathbf{P}_\eta^{h,h+1}\mathbf{g}_\eta^{h+1} + \cdots + \mathbf{P}_\eta^{h,h+1}\mathbf{P}_\eta^{h+1,h+2}\cdots\mathbf{P}_\eta^{h-2,h-1}\mathbf{g}_\eta^{h-1}. \end{aligned} \tag{17}$$

Without loss of generality, we restrict policies such that the induced daily MCs are irreducible and aperiodic. This allows us to use the proof framework in Dai and Gluzman (2022) to extend the results in time-stationary settings to our daily MC.

To start, the relative value function v_η^h under π_η is given by

$$v_\eta^h(s) := \mathbb{E}\Big[\sum_{t=1}^{\infty}\Big(\tilde{g}_\eta^h(S(t_h)) - (\mu_\eta^h)^T\tilde{\mathbf{g}}_\eta^h\Big) \mid S(1_h) = s\Big],\ \forall s \in \mathcal{S}^h. \tag{18}$$

We redefine the PPO objective as:

$$L^h(\theta) := \mathop{\mathbb{E}}_{\substack{s\sim\mu_\eta^h \\ f\sim\pi_\eta(\cdot|s)}} \max\left\{r_{\theta,\eta}(f\mid s)A_\eta^h(s,f),\quad \text{clip}\left(r_{\theta,\eta}(f\mid s), 1-\epsilon, 1+\epsilon\right)A_\eta^h(s,f)\right\}, \tag{19}$$

where A_η^h is the advantage function of the daily MC, whose definition will be specified later in this section; the vector μ_η^h is the stationary distribution of this daily MC; and $r_{\theta,\eta}$ is the policy ratio. We update the policy from π_η by minimizing this redefined objective L^h. Moreover, we update the policy only at states in $\mathcal{S}^h$. We denote the updated policy as $\pi_{\theta,h}$, which follows

$$\pi_{\theta,h}(f\mid s) = \begin{cases} \pi_\theta(f\mid s), & \forall s\in\mathcal{S}^h, f\in\mathcal{A}(s), \\ \pi_\eta(f\mid s), & \forall s\in\mathcal{S}\setminus\mathcal{S}^h, f\in\mathcal{A}(s) \end{cases}. \tag{20}$$

We denote the one-day transition matrix and one-day cost vector of the induced daily MC, under this policy update scheme, as $\tilde{\mathbf{P}}_\theta^h$ and $\tilde{\mathbf{g}}_\theta^h$, respectively. They follow

$$\tilde{\mathbf{P}}_\theta^h = \mathbf{P}_\theta^{h,h+1}\mathbf{P}_\eta^{h+1,h+2}\cdots\mathbf{P}_\eta^{h-1,h}, \tag{21}$$

and

$$\tilde{\mathbf{g}}_\theta^h = \mathbf{g}_\theta^h + \mathbf{P}_\theta^{h,h+1}\mathbf{g}_\eta^{h+1} + (\mathbf{P}_\theta^{h,h+1}\mathbf{P}_\eta^{h+1,h+2})\mathbf{g}_\eta^{h+2} + \cdots + (\mathbf{P}_\theta^{h,h+1}\mathbf{P}_\eta^{h+1,h+2}\cdots\mathbf{P}_\eta^{h-2,h-1})\mathbf{g}_\eta^{h-1}. \tag{22}$$

Note that the transition from epoch h to $h+1$ is given by $\mathbf{P}_\theta^{h,h+1}$ with θ, while all other transitions remained to be $\mathbf{P}_\eta^{\cdot,\cdot}$ with η. For notational simplicity, we use θ here in the subscript but emphasize that we only update the policy at epoch h.

Let $\tilde{g}_\eta^h(s,f)$ represent the one-day expected cost starting from state $s\in\mathcal{S}^h$, given that action f is taken at s and subsequent actions are determined by the policy π_η. The advantage function for the daily MC (under our considered policy update) is given by:

$$A_\eta^h(s,f) = \tilde{g}_\eta^h(s,f) - (\mu_\eta^h)^T\tilde{\mathbf{g}}_\eta^h + \mathbb{E}_{s'\sim\tilde{p}_\eta^h(\cdot|s,f)}[v_\eta^h(s')] - v_\eta^h(s), \tag{23}$$

where $\tilde{p}_\eta^h(s'|s,f)$ denotes the one-day transition probability from state $s\in\mathcal{S}^h$ in the current day to state $s'\in\mathcal{S}^h$ in the next day, given that action f is taken at s and subsequent actions are determined by π_η.

C.3. Average cost gap

We denote the *unclipped* PPO objective function as $N_1^h(\theta,\eta)$, which follows

$$\begin{aligned} N_1^h(\theta,\eta) :=& \mathop{\mathbb{E}}_{\substack{s\sim\mu_\eta^h \\ f\sim\pi_\eta(\cdot|s)}}[r_{\theta,\eta}(f\mid s)A_\eta^h(s,f)] = \mathop{\mathbb{E}}_{\substack{s\sim\mu_\eta^h \\ f\sim\pi_\theta(\cdot|s)}}[A_\eta^h(s,f)] \\ =& (\mu_\eta^h)^T\Big(\tilde{\mathbf{g}}_\theta^h - (\mu_\eta^h)^T\tilde{\mathbf{g}}_\eta^h\mathbf{e} + (\tilde{\mathbf{P}}_\theta^h - I)\mathbf{v}_\eta^h\Big). \end{aligned} \tag{24}$$

Our eventual goal is to examine whether minimizing N_1^h while controlling $r_{\theta,\eta}$ to be near 1 can guarantee policy improvement. To achieve this goal, we study the average-cost gap between the current policy π_η and the new policy $\pi_{\theta,h}$. We impose the following assumptions for the stability of the current policy π_η; these assumptions are from Dai and Gluzman (2022).

ASSUMPTION 1. *For any given* $h = 0, 1, ..., m-1$,

- *the daily MC with transition matrix* $\tilde{\mathbf{P}}_\eta^h$ *is irreducible and aperiodic.*
- *there exists some vector* $\mathcal{V} = \{\mathcal{V}(s),\ s \in \mathcal{S}^h\} \geq 1$, *some constants* $b \in (0,1)$, $d \geq 0$ *and finite subset* $C \subset \mathcal{S}^h$ *satisfying*

$$\tilde{\mathbf{P}}_\eta^h \mathcal{V} \leq b\mathcal{V} + d\mathbf{1}_C, \tag{25}$$

  where each element of the vector $\mathbf{1}_C = \{\mathbb{1}_C(s) \in \{0,1\}, s \in \mathcal{S}^h\}$ *is* $\mathbb{1}_C(s) = 1$ *if* $s \in C$ *and* 0 *if* $s \notin C$.
- *the one-day cost vector satisfies* $|\tilde{\mathbf{g}}_\eta^h| \leq \mathcal{V}$.

Under these conditions, one can prove that there exists a unique stationary distribution μ_η^h for the MC with transition matrix $\tilde{P}_\eta^h$; see for example, Theorem 11.3.4 and Theorem 14.3.7 in Meyn and Tweedie (2012), which is also restated in Lemma 1 of Dai and Gluzman (2022). Moreover, for any $h = 0, 1, ..., m-1$, we assume that the new policy (using the considered updating mechanism) satisfies the following condition:

$$\|(\tilde{\mathbf{P}}_\theta^h - \tilde{\mathbf{P}}_\eta^h) \sum_{n=0}^{\infty} (\tilde{\mathbf{P}}_\eta^h - \Pi_\eta^h)^n\|_{\mathcal{V}} < 1, \tag{26}$$

where every row of Π_η^h equals the stationary distribution of the MC with transition matrix $\tilde{\mathbf{P}}_\eta^h$, i.e., $\Pi_\eta^h(s, s') = \mu_\eta^h(s')$, and the $\mathcal{V}$-norm of a matrix $\Omega \in \mathcal{S}^h \times \mathcal{S}^h$ is defined as

$$\|\Omega\|_{\mathcal{V}} = \sup_{s \in \mathcal{S}^h} \frac{1}{\mathcal{V}(s)} \sum_{s' \in \mathcal{S}^h} |\Omega(s, s')| \mathcal{V}(s'). \tag{27}$$

Then one can show that the MC with transition matrix $\tilde{\mathbf{P}}_\theta^h$ also has a unique stationary distribution μ_θ^h. For the proof, see Lemma 4 of Dai and Gluzman (2022).

In the following proposition, we decompose the difference in the long-run average cost between the current policy π_η and the updated policy $\pi_{\theta,h}$ into two terms, and analyze their decay rates when θ approaches η.

PROPOSITION 1. *Assume that the current policy satisfies Assumption 1 and the new policy satisfies* (26). *The difference between the long-run average costs of the two policies equals*

$$(\mu_\theta^h)^T \tilde{\boldsymbol{g}}_\theta^h - (\mu_\eta^h)^T \tilde{\boldsymbol{g}}_\eta^h = N_1^h(\theta, \eta) + N_2^h(\theta, \eta),$$

where N_1^h *is defined in Equation* (24), *and* N_2^h *is defined as*

$$N_2^h(\theta, \eta) := (\mu_\theta^h - \mu_\eta^h)^T \Big(\tilde{\mathbf{g}}_\theta^h - (\mu_\eta^h)^T \tilde{\mathbf{g}}_\eta^h \mathbf{e} + (\tilde{\mathbf{P}}_\theta^h - I)\mathbf{v}_\eta^h \Big). \tag{28}$$

Moreover, we have $N_1^h(\theta, \eta) = O(||\boldsymbol{r}_{\theta,\eta}^h - 1||_\infty)$ *and* $N_2^h(\theta, \eta) = O(||\boldsymbol{r}_{\theta,\eta}^h - 1||_\infty^2)$, *where*

$$\big\|\boldsymbol{r}_{\theta,\eta}^h - 1\big\|_\infty := \sup_{s \in \mathcal{S}^h} \sum_{f \in \mathcal{A}(s)} \big|r_{\theta,\eta}(f \mid s) - 1\big|.$$

The proof of this proposition is in Appendix A of the Technical Companion (Anonymous 2025).

C.4. Improvement Guarantee of PPO

According to Proposition 1, if we update policy from π_η to $\pi_{\theta,h}$, the change in the average cost equals $N_1^h(\theta,\eta)+N_2^h(\theta,\eta)$. Then, the negativity of $N_1^h(\theta,\eta)+N_2^h(\theta,\eta)$ guarantees that the new policy yields an improved performance comparing with the current policy π_η in terms of the average daily cost (where this average daily cost is the same regardless the epoch we are considering). Since $N_1^h(\theta,\eta)+N_2^h(\theta,\eta) \le N_1^h(\eta,\eta)+N_2^h(\eta,\eta)=0$, to achieve maximum improvement, we can pick $\theta=\theta^*$ as

$$\theta^* = \arg\min_{\theta\in\Theta} N_1^h(\theta,\eta)+N_2^h(\theta,\eta). \tag{29}$$

However, this optimization problem cannot be directly solved because N_2^h depends on the stationary distribution of the new policy, which has no closed form and cannot be estimated when θ is to be determined. Therefore, the PPO framework proposes using an alternative way to approximately solve problem (29). That is, obtain θ via the the objective function $L^h(\theta)$, as defined in Equation (19).

Proposition 1 implies that, as the policy ratio $r_{\theta,\eta}^h$ is close to 1, $N_2^h(\theta,\eta)$ is of a smaller order compared to $N_1^h(\theta,\eta)$. Therefore, if we get θ via (i) minimizing $N_1^h(\theta,\eta)$, the unclipped objective function and (ii) keeping $r_{\theta,\eta}(f\mid s)$ to be close to 1, then policy improvement is guaranteed as $N_1^h(\theta,\eta)<0$ and $||\mathbf{r}_{\theta,\eta}^h-1||_\infty$ is sufficiently small so that $N_1^h(\theta,\eta)+N_2^h(\theta,\eta)<0$. These two goals are achieved simultaneously through the conservative updates induced by minimizing the clipped objective function $L^h(\theta)$, as we explain below.

Note that in adapting to the daily MC, $L^h(\theta)$ is not equivalent to (9) unless $m=1$. In our algorithm implementation, we nonetheless adopt (9), since our primary objective here is to illustrate the rational behind PPO, namely, its clipped objective effectively controls the two terms N_1 and N_2. For notational simplicity, we now explain how the max operator and clipping function in (9) work together to enforce conservative policy updates (dropping the index h). We can rewrite (9) in the following explicit piecewise form:

$$\begin{aligned}&\max\left\{r_{\theta,\eta}(f\mid s)A_\eta(s,f),\quad \text{clip}\left(r_{\theta,\eta}(f\mid s),1-\epsilon,1+\epsilon\right)A_\eta(s,f)\right\}\\ =&\begin{cases}(1+\epsilon)A_\eta(s,f), & \text{if } A_\eta(s,f)<0 \text{ and } r_{\theta,\eta}(f|s)>1+\epsilon,\\ (1-\epsilon)A_\eta(s,f), & \text{if } A_\eta(s,f)>0 \text{ and } r_{\theta,\eta}(f|s)<1-\epsilon,\\ r_{\theta,\eta}(f|s)A_\eta(s,f), & \text{otherwise.}\end{cases}\end{aligned}$$

In the first case, $A_\eta(s,f)<0$ indicates that action f reduces cost (desirable), so the policy update encourages increasing $r_{\theta,\eta}(f|s)$; the clipping at $1+\epsilon$ prevents an excessively large increase. In the second case, $A_\eta(s,f)>0$ indicates that action f increases cost (not desirable), and the policy update encourages reducing $r_{\theta,\eta}(f|s)$; when $r_{\theta,\eta}(f|s)<1-\epsilon$, the clipping term constrains this decrease. In the remaining case, the unclipped objective is used.

Appendix D: Basis Function Design

In Section 5.2 of the main paper, we propose a new basis function $V_d(s)=\sum_{j=1}^J \hat{v}_{\pi,j}(s_j)$ that decomposes by each pool j with $s_j=(x_j,y_j,h)$. Here, each $\hat{v}_{\pi,j}(s_j)$ denotes the *post-action* value function of a single-pool system which approximates the dynamics of pool j under policy π, and it is computed using the post-action Poisson equation for the decomposed pool-j system in Equation (13). To highlight the difference between

$s_j = (x_j, y_j, h) \in \mathcal{S}_j$	pre-action state of pool j with its state space as $\mathcal{S}_j$
s_j^+	post-action state of pool j
s_j'	pre-action state of pool j at the next epoch
$s_j'^+$	post-action state of pool j at the next epoch
$s_{-j} \in \mathcal{S}_{-j}$	pre-action state of pools excluding pool j with its state space as $\mathcal{S}_{-j}$
$f' = (f'_{i,j}, i, j = 1, \ldots, J\})$	system-level overflow action of the next epoch
$\bar{f} = (\bar{f}_{i,j}, i, j = 1, \ldots, J\})$	approximation of f'
π_j	inflow and outflow policy of decomposed pool-j system under policy π
$\bar{\kappa}_j(i\|s_j, j)$	approximation of overflow action probability at the next epoch under policy π which only uses the current post-action state information for the pool-j system
$p_j(s_j'\|s_j^+)$	transition probability of pool j from post-action state s_j^+ to next post-action state s_j'
$p(s'\|s^+, 0)$	transition probability from post-action state s^+ to next post-action state s', with $p(s'\|s^+, 0) = \prod_{j=1}^J p_j(s_j'\|s_j^+)$
$\bar{v}_\pi$	post-action value function of the original J-class J-pool system under overflow policy π
$v_{\pi,j}$	pre-action value function of pool j under policy π
$\bar{v}_{\pi,j}$	post-action value function of of pool j under policy π
$\hat{v}_{\pi,j}$	post-action value function of the approximated decomposed pool-j system under the approximated pool-independent overflow probabilities $\bar{\kappa}_j$
$V_d(s) = \sum_{j=1}^J \hat{v}_{\pi,j}(s_j)$	new design that tailors to the randomized policy
$\hat{\gamma}_{\pi,j}$	average cost of the approximated decomposed pool-j system under the approximated pool-independent overflow probabilities $\bar{\kappa}_j$

Table 8 Notations in Appendix D

the pre- and post-action states in this section, we use $s_j^+ = (x_j^+, y_j^+, h)$ to denote the post-action state for the pool-j system and restate Equation (13) as follows:

$$\hat{v}_{\pi,j}(s_j^+) = C_j q_j^+ - \hat{\gamma}_{\pi,j} + \mathbb{E}\left[\sum_{i=1}^J B_{j,i}\bar{f}_{j,i} + \hat{v}_{\pi,j}(s_j'^+)\right], \tag{30}$$

where $q_j^+ = (x_j^+ - N_j)^+$ denotes the post-action queue length, and $s_j'^+$ denotes the next post-action state. In this section, we provide detailed explanations on (i) how the pool-wise decomposition is performed, and (ii) how to estimate the parameters in Equation (30) to approximate the dynamics in the original (non-decomposed) system. Table 8 summarizes the notations used in this section.

D.1. Pool-wise Decomposition

In order to obtain pool-wise decomposed Poisson equation (30), we first derive the post-action Poisson equation for the original J-pool system. We then derive the decomposed version.

Post-action Poisson equation. Under a given randomized, stationary policy π, the pre-action Poisson equation follows

$$\begin{aligned} v_\pi(s) =& \mathbb{E}_{f\sim\pi(\cdot|s)}\left[g(s,f) - \gamma_\pi + \sum_{s'\in\mathcal{S}} p(s'|s,f)v_\pi(s')\right] \\ =& \mathbb{E}_{f\sim\pi(\cdot|s)}\left[\sum_{i,j=1}^J B_{i,j}f_{i,j} + \sum_{j=1}^J C_j q_j^+ - \gamma_\pi + \mathbb{E}_{s'\sim p(\cdot|s^+)}v_\pi(s')\right], \end{aligned} \tag{31}$$

where $s^+ = \{s_j^+\}_{j\in\mathcal{J}}$ denotes the current post-action state in the original J-pool system, q_j^+ is the post-action queue length for pool j, and $p(s'|s^+)$ denotes the transition probability from s^+ to the next pre-action state

s'. Note that here, the transitions only depend on the arrivals and departures that occurred between the current and the next epochs.

We denote the post-action value function as $\bar{v}_\pi$, where

$$\bar{v}_\pi(s^+) = \sum_{j=1}^{J} C_j q_j^+ + \mathbb{E}_{s'\sim p(\cdot|s^+)} v_\pi(s'), \tag{32}$$

using which we can rewrite (31) as

$$v_\pi(s) = -\gamma_\pi + \mathbb{E}_{f\sim\pi(\cdot|s)}\left[\sum_{i,j=1}^{J} B_{i,j} f_{i,j} + \bar{v}_\pi(s^+)\right].$$

Plugging the above back to Equation (32), we get the following post-action Poisson Equation:

$$\bar{v}_\pi(s^+) = \sum_{j=1}^{J} C_j q_j^+ - \gamma_\pi + \mathbb{E}_{s'\sim p(\cdot|s^+)}\mathbb{E}_{f'\sim\pi(\cdot|s')}\left[\sum_{i,j=1}^{J} B_{i,j} f'_{i,j} + \bar{v}_\pi(s'^+)\right], \tag{33}$$

with the pair $(\bar{v}_\pi, \gamma_\pi)$ being its solution.

Decomposition. In the post-action Poisson Equation (33), the first two terms $\sum_{j=1}^{J}(C_j q_j^+ - \gamma_{\pi,j})$ can be naturally decomposed by each pool, where $\gamma_{\pi,j}$ corresponds to the average cost of pool j under policy π. We then focus on considering the decomposition for the terms within the expectation, i.e.,

$$\mathbb{E}_{s'\sim p(\cdot|s^+)}\mathbb{E}_{f'\sim\pi(\cdot|s')}\left[\sum_{i,j=1}^{J} B_{i,j} f'_{i,j} + \bar{v}_\pi(s'^+)\right] = \sum_{j=1}^{J}\mathbb{E}_{s'\sim p(\cdot|s^+)}\mathbb{E}_{f'\sim\pi(\cdot|s')}\left[\sum_{i=1}^{J} B_{j,i} f'_{j,i} + \bar{v}_{\pi,j}(s_j'^+)\right].$$

We can observe here that the key is to decompose the probabilities, i.e., the transition probabilities p and overflow action probabilities π, by each pool. Note that the transition probability $p(s'|s^+, 0)$ from post-action state s^+ to state of next epoch s' can be decomposed by each pool naturally, that is,

$$p(s'|s^+, 0) = \prod_{j=1}^{J} p_j(s'_j|s_j^+), \tag{34}$$

where $p_j(s'_j|s_j^+)$ denotes the transition, for each pool j, from the post-action state to the next epoch pre-action state. This decomposition holds because after the overflow assignments, the state transitions only depend on the arrivals to and departures from each pool j, which are independent among j's.

Next, let $s_{-j}^+ = \{(x_i^+, y_i^+, h)\}_{i\neq j}$ denote the current post-action state of the other pools (excluding pool j) and $\mathcal{S}_{-j}$ the corresponding state space. Then, the j-th component within the expectation term can be rewritten as

$$\mathbb{E}_{s'\sim p(\cdot|s^+)}\mathbb{E}_{f'\sim\pi(\cdot|s')}\left[\sum_{i=1}^{J} B_{j,i} f'_{j,i} + \bar{v}_{\pi,j}(s_j'^+)\right] = \sum_{s'_j\in\mathcal{S}_j} p_j(s'_j|s_j^+)\mathbb{E}_{f'\sim\pi_j(\cdot|s'_j, s_{-j}^+)}\left[\sum_{i=1}^{J} B_{j,i} f'_{j,i} + \bar{v}_{\pi,j}(s_j'^+)\right] \tag{35}$$

for any $j = 1, \ldots, J$, with

$$\pi_j(f'|s'_j, s_{-j}^+) = \sum_{s'_{-j}\in\mathcal{S}_{-j}} \prod_{i\neq j} p_i(s'_i|s_i^+)\pi(f'|s') \quad \text{for } s' = (s'_j, s'_{-j}).$$

The important step for this decomposition lies in the $\pi_j(f'|s'_j, s_{-j}^+)$ term, which evaluates the marginal probability of taking overflow action f' under the policy π by taking an expectation over all the possible transition from s_{-j}^+ to s'_{-j}. This expectation is needed because in the decomposed equation (35) we only evaluated the transition for pool j via $p_j(s'_j|s_j^+)$ and did not evaluate the transitions for other pools.

To fully decompose, we still need to approximate this marginal probability $\pi_j(f'|s'_j, s^+_{-j})$ since it requires information on the states of other pools besides pool j (to get the probability of prescribing action $f' = \{f'_{i,j}\}$ based on policy π). To do so, we go back to the atomic action setup and approximate the probability of prescribing $f'_{i,j}$ with $\bar{f}_{i,j} \sim Poi(\lambda_i \bar{\kappa}_j(j|s'_j, i))$, and the probability of prescribing $f'_{j,i}$ with $\bar{f}_{j,i} \sim Poi(\lambda_j \bar{\kappa}_j(i|s'_j, j))$, using some approximated probability $\bar{\kappa}_j(\cdot|s'_j, \cdot)$ that does not require any state information from other pools beyond pool j. Note that the probability $\bar{\kappa}_j(j|s'_j, i)$ is the probability for the *atomic* action for assigning one patient from class i to j. Under the batched routing setting with the approximate probability $\bar{\kappa}_j(j|s'_j, i)$, the aggregate action $\bar{f}_{i,j}$ becomes equivalent to the Poisson random variables defined here using the Poisson thinning property given state s'_j. The rational for $\bar{f}_{j,i}$ is the same.

Given these distributions of $\bar{f}$'s, for each pool, we could then approximate the remaining state transition in (35) – from s'_j to $s^{'+}_j$ – by properly adjusting the inflows and outflows via: (a) overflow assignments from other classes to pool j, $\bar{f}_{i,j}$; (b) the overflow (diversion) from class j to other pools, $\bar{f}_{j,i}$, i.e., $x^{'+}_j = x'_j + \sum_{i=1}^J \bar{f}_{i,j} - \sum_{i=1}^J \bar{f}_{j,i}$. This eventually leads to our (approximated) pool-wise decomposed post-action Poisson equation:

$$\hat{v}_{\pi,j}(s^+_j) = C_j q^+_j - \gamma_{\pi,j} + \mathbb{E}_{s'_j \sim p_j(\cdot|s^+_j)} \left[\sum_{i=1}^J B_{j,i} \bar{f}_{j,i} + \mathbb{E}_{\bar{f}} \left[\hat{v}_{\pi,j}(s^{'+}_j) \right] \right], \tag{36}$$

Since (36) is derived from decomposing (33) by pools, then summarizing over all j's, the summation $\left(\sum_{j=1}^J \hat{v}_{\pi,j}, \sum_{j=1}^J \hat{\gamma}_{\pi,j} \right)$ can serve as an approximation for the solution to (33). Therefore, we could use $V_d = \sum_{j=1}^J \hat{v}_{\pi,j}$ as one of the basis functions to approximate relative value functions.

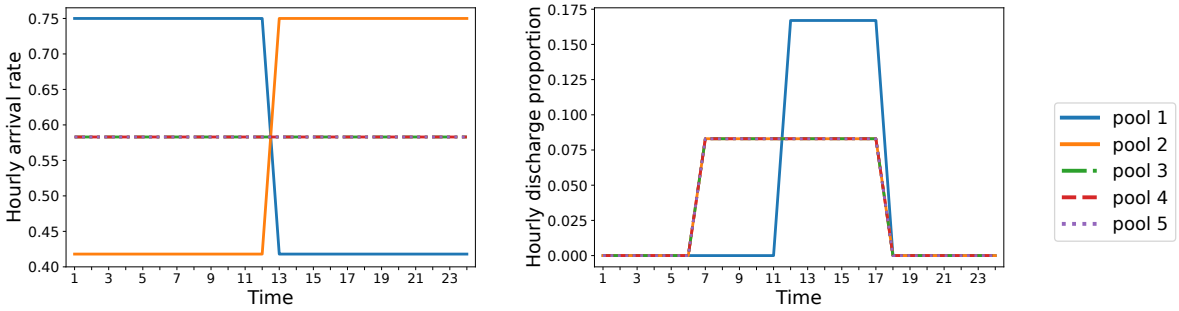
Appendix E: Additional Numerical Results

For the baseline five-pool model, we use the same setting of feasible routes as in Dai and Shi (2019), and the detailed description of the feasible routes in ten- and twenty-pool models are specified in Section 6.1. Figure 7 shows the time-varying arrival and discharge patterns of the five-pool model. The ten- and twenty-pool models can be considered as combination of two and four baseline five-pool settings, respectively, with identical discharge patterns and proportionally adjusted arrival rates. Detailed specifications are also given in Section 6.1. The number of servers per pool in the baseline five-pool model is $(N_1, \ldots, N_5) = (60, 64, 67, 62, 62)$. For the ten-pool model, we set $(N_1, \ldots, N_{10}) = (39, 43, 46, 41, 41, 81, 85, 88, 83, 83)$, and for the twenty-pool model, we set $(N_1, \ldots, N_{20}) = (32, 36, 39, 34, 34, 74, 78, 81, 76, 76, 46, 50, 53, 48, 48, 88, 92, 95, 90, 90)$.

E.1. Impact of Hyper-parameters for Neural Network Training

Tables 9 and 10 report the numeric results for those plotted in Figures 5 and 6, conducted in the baseline ten-pool model. The ten-pool model is too large to be solved by ADP. Thus, we use the best of the three heuristic policies as the benchmark policy (with an average cost of 393.36).

These results also offer guidance for tuning hyperparameters. Table 9 shows that comparable performance gains can be achieved by either increasing the number of simulation days or training epochs. However, increasing simulation days incurs significantly higher computational cost: not only due to longer data generation time but also because larger datasets slow down policy network training. Therefore, we recommend increasing the number of training epochs first. Table 10 further suggests that increasing network width generally yields better performance than increasing depth; in some cases, deeper networks may even degrade performance. Thus, we advise prioritizing wider architectures over deeper ones.



(a) Hourly arrival rate (b) Hourly discharge distribution

Figure 7 Time-varying arrival and discharge patterns in five-pool models.

(34 neurons per hidden layer)		0 depth			1 depth			2 depth		
# simulation days per actor		5k	10k	100k	5k	10k	100k	5k	10k	100k
Full separate	10 epoch	375.93±2.76	349.88±1.79	344.72±0.82	358.82±2.98	342.98±1.34	337.02±0.65	374.42±2.24	349.98±1.23	341.24±0.47
	15 epoch	364.79±2.87	345.12±1.45	343.30±0.46	352.21±2.49	337.89±1.32	336.48±0.62	369.90±2.50	344.33±1.07	342.27±0.76
Full connect	10 epoch	359.45±3.02	343.47±1.34	343.89±0.45	357.72±2.45	347.63±1.90	342.29 ±0.42	368.76±2.79	352.29±2.07	346.09±0.29
	15 epoch	357.79±2.61	343.78±1.20	342.29±0.41	354.73±3.00	345.05±1.98	343.05±0.39	359.90±2.78	348.42±1.22	346.27±0.66
Partial share	10 epoch	-			329.97±2.77	320.09±1.67	312.98±0.52	334.98±2.28	329.98±1.45	311.09±0.22
	15 epoch				321.45±1.90	313.35±1.54	309.69±0.45	327.71±2.35	321.09±1.55	312.29±0.24

Table 9 Policy network comparison in 10-pool 8-epoch

(10k data 15 epoch)		0 depth	1 depth	2 depth	3 depth
fully-separate	17 neurons	345.12±1.45	337.92±1.34	349.23±1.02	371.29±2.11
	34 neurons		337.89±1.32	344.33±1.07	379.59±1.45
	68 neurons		341.63±1.29	342.12±1.47	377.72±1.55
fully-connected	17 neurons	343.78±1.20	344.42±2.09	347.72±1.96	312.21±1.08
	34 neurons		345.05±1.98	348.42±1.22	311.48±1.35
	68 neurons		344.15±1.20	345.23±1.07	308.23±2.42
partially-shared	17 neurons	-	313.23±1.76	320.09±1.55	313.32±1.27
	34 neurons		313.35±1.54	321.09±1.55	315.78±1.20
	68 neurons		315.41±1.98	318.82±1.45	312.28±1.22

Table 10 network complexity tuning in 10-pool 8-epoch

E.2. Comparison with Additional Benchmarks

This section introduces several benchmark policies for comparative evaluation. Section E.2.1 introduces a heuristic threshold-based policy. Section E.2.2, E.2.3 and E.2.4 treat several variants tested in the ablation study as the benchmark policies. The results reported in the subsequent sections are from the baseline five-pool setting.

E.2.1. Threshold-Based Policy We introduce a heuristic threshold-based policy triggers overflow when the queue lengths exceeds a threshold. To simplify, we divide the eight decision epochs into two periods (daytime and nighttime) following the empirical policy structure. For each class and period, we search for optimal threshold values from the range 0–10, as this range covers the most frequently observed queue lengths in simulation. The resulting threshold policies yield average costs ranging from 232.87 to 356.29, with the best policy using thresholds $[0, 0, 2, 0, 0]$ at night and $[1, 4, 6, 4, 4]$ during the day. This achieves a 2.87% improvement over the empirical policy's performance (239.76). However, given the high computational cost

of threshold tuning and the relatively modest gains, we use the empirical policy as the primary benchmark in the main paper.

E.2.2. Atomic + ADP To evaluate whether the atomic decomposition can be paired with ADP instead of PPO, we implement an Atomic+ADP benchmark. As shown in Figure 8, this method performed poorly and exhibited significant "chattering" (large performance oscillations). It is due to the greedy search mechanism in ADP: when the value function is inaccurately estimated, the policy can deteriorate dramatically.

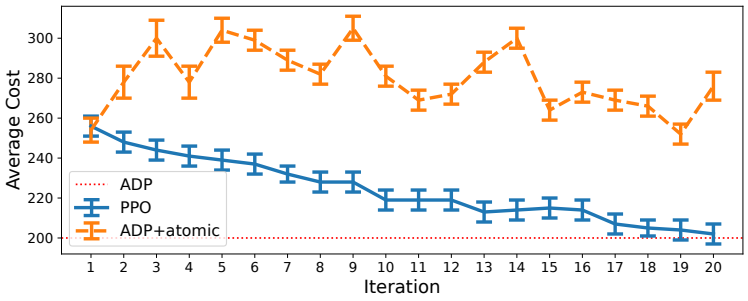


Figure 8 Performance comparison of ADP, Atomic+ADP, and PPO (full algorithm) in the five-pool setting. Atomic+ADP exhibits poor performance and significant chattering.

To fairly assess the performance of the Atomic+ADP approach, we evaluate two configurations of action representation and a few different basis function choices. The results reported in Figure 8 correspond to the best-performing setup. Specifically, in the first configuration, we adopt the same atomic action structure as in our PPO algorithm: actions are executed in FIFO order, with each action corresponding to selecting an overflow pool $[1, 2, \ldots, J]$ for the current patient. The value function approximation uses the basis set:

$$\{X_j, Y_j, W_j, X_j^2, Y_j^2, W_j^2, V_s, e_{\text{epoch}}, e_{\text{class}}\},$$

which augments the basis functions from Dai and Shi (2019) by including linear and quadratic terms for waiting patients (W_j), one-hot encodings for the epoch ($e_{\text{epoch}} \in \{0,1\}^m$), and patient class ($e_{\text{class}} \in \{0,1\}^J$). This setup results in poor performance (best average cost: 282.77) and severe chattering, likely due to instability introduced by the high-dimensional one-hot features. In the second configuration, inspired by Feng et al. (2021), actions are defined as ordered pairs (i, j), where a class i patient is assigned to ward j. This removes the need to order patients in advance. We test two basis sets: (i) the original from Dai and Shi (2019), $X_j, Y_j, X_j^2, Y_j^2, V_s$, which achieves an average cost of 268.81; and (ii) an augmented version that includes W_j, W_j^2, which further lowers the average cost (259.24) but still suffers from great instability. This latter one corresponds to the curve shown in Figure 8.

We conjecture that designing effective basis functions is particularly difficult for the Atomic+ADP approach, as the value function must capture the impact of individual action ordering (which is different from the system-level actions considered in Dai and Shi (2019) and adds an inherently complex dependency). In contrast, PPO uses proximal, conservative updates that reduce sensitivity to approximation errors and eliminate chattering, leading to significantly more stable performance in our long-run average cost setting.

E.2.3. NN-based Value Network Following the standard setup of actor-critic RL algorithms, we have conducted experiments using an NN-based critic (value network) in place of LSTD. All experiments use 10k simulation days per iteration and we test both five-pool and ten-pool settings under sequential and batched atomic action updates. Across the board, the NN critic yields higher average costs and longer computation times, with performance gaps more pronounced in the ten-pool setting. For example, in the ten-pool case without batching, the NN critic results in an average cost of 324.23 (vs. 304.52 with LSTD), requiring 31% more computation time. Using batching, the NN critic yields an average cost of 213 (vs. 207 for LSTD) in the five-pool setting with a 14% increase in runtime; and 328.67 (vs. 309.02) in the ten-pool setting, with a 19% increase in runtime.

The performance degradation under the NN critic likely stems from higher variance and less stable advantage estimates when simulation data are limited. In contrast, LSTD with tailored basis functions provides lower-variance, more accurate estimates that support more stable learning. Moreover, NN critics require extra time for training, while LSTD adds minimal overhead.

E.2.4. Action Ordering When performing sequential updates under the atomic action setup, one natural question is whether the order in which patients are processed affects performance. To investigate this, we have tested five ordering rules: Random, First-In–First-Out (FIFO), and three priority-based rules: (i) Class 1 to Class 5, (ii) Class 5 to Class 1, and (iii) Class 1,4,5,2,3 (from most- to least-loaded class). We compare performance under these different ordering rules with those from the batching approximation in two five-pool settings: the baseline setting and a more congested variant (with 3 fewer beds in ward 1 and 2 fewer in ward 2); the latter leads to longer queues and more batching.

Ordering	Random	FIFO	Priority Rule (i)	Priority Rule (ii)	Priority Rule (iii)	Batching
Baseline	207.19±4.27	207.22±3.82	203.24±4.51	205.33±4.13	203.11±4.00	208.82±4.78
Crowded	248.27±4.22	245.19±4.02	236.38±4.11	244.89±4.43	232.14±3.97	247.33±4.09

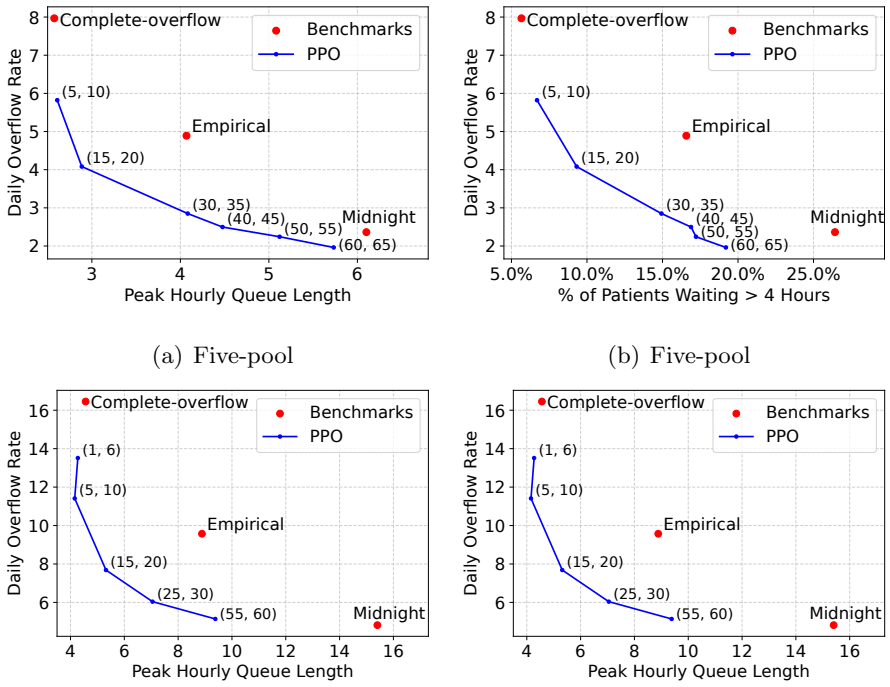
Table 11 Comparison of Different Ordering.

Results in Table 11 show minimal performance differences (about 2%) across ordering rules in the baseline case. In the congested setting, differences became more pronounced (around 8–9%). While batching performs comparably to Random and FIFO, it is generally worse than the best priority-based rule (Priority Rule iii). In summary, ordering does influence performance, but the effect is minimal when queue lengths are low (3–5 patients per class on average). When the system is more congested, the impact grows, but this could be mitigated by increasing decision frequency to reduce batch size. These findings support our use of the batching approximation as proposed in Section 5, which offers substantial computational savings (reducing per-iteration runtime by roughly 50%) with only a minor trade-off in performance.

E.3. Additional Performance Metrics

Figure 9 plots the trade-off between system congestion and overflow rates for the five- and ten-pool systems. We fix the holding cost at $C = 6$ and vary the overflow costs to construct the blue curve. Plots (a) and (c) show the daily overflow rate versus peak hourly queue length; plots (b) and (d) show the daily overflow rate versus the proportion of patients waiting longer than 4 hours, referred to as the 4-hour non-service level. Across

all settings, the PPO policy achieves a Pareto improvement: it either reduces waiting at a similar overflow rate, lowers the overflow rate without increasing congestion, or improves both dimensions simultaneously. The improvement is more pronounced in the larger ten-pool system. For example, in Figures 9(c) and 9(d), relative to the empirical policy, PPO achieves about 4 fewer overflow patients per day (approximately 16 fewer patients in non-primary wards, assuming an average LOS of 4 days) at the same peak queue length, or about 4 fewer patients waiting at the peak hour and a 15% absolute reduction in the peak 4-hour non-service level at the same overflow rate.



(a) Five-pool (b) Five-pool

(c) Ten-pool (d) Ten-pool

Figure 9 Additional Performance Metrics. The numbers in parentheses along the blue line indicate the corresponding cost parameter pairs (B_1, B_2). For the ten-pool settings, the additional cross-department overflow cost parameters are set as $(B_3, B_4) = (B_1 + 10, B_2 + 10)$.

Across the experiments, we fix the overflow cost gap $B_2 - B_1$ at 5. We also conduct additional experiments with $C = 6$ and $B_1 = 15$, while varying the gap $B_2 - B_1$. As B_2 increases, we find that the PPO policy (i) uses fewer secondary overflow assignments and (ii) shifts toward more preferred overflow assignments, while allowing more patients to wait. Thus, increasing the overflow cost gap changes not only the overall use of overflow, but also the composition of overflow assignments.

Appendix F: Policy Visualization and SHAP Analysis Details

This appendix provides a detailed analysis of the operational behavior of the learned PPO policy. Using visualization and explainable AI techniques, we illustrate its decision-making process and derive actionable managerial insights. The PPO policy examined here is trained under the baseline five-pool setting. Following Dai and Shi (2019), the baseline five-pool setting assigns each pool a capacity of $(N_1, \ldots, N_5) = (60, 64, 67, 62, 62)$.

The preferred overflow wards for classes $(1,2,3,4,5)$ are $(5,3,2,2,1)$, respectively; the secondary choices are $\{(2,3),(4,5),(1,5),(1,3),(2,3)\}$, respectively.

Section F.1 presents direct policy visualizations that reveal how different system factors influence the policy's decisions. Section F.2 further employs SHAP (SHapley Additive exPlanations) analysis to generate additional insights into the learned policy, in particular, uncover underlying network effects. Section F.3 compares PPO policy and ADP policy by direct visualization.

F.1. Direct Policy Visualization

We visualize the overflow probabilities of class 1 patients to illustrate how key state variables influence overflow decisions. In each visualization, two variables are varied (shown on the axes), while all other components are fixed at representative baseline values: for fixed X, we set $X_j = N_j$ if $j \neq 1$ and $X_1 = N_1 + 5$.

F.1.1. Impact of patient count. Figure 10 shows how the patient count (X_j) influence the policy. The first three subfigures display how the probability that a class 1 patient is not overflowed (P_{11}) changes with X_1 and X_2 at midnight, 12 pm, and 9 pm (Epoch $h = 0$, 4, 7); the fourth subfigure shows how the overflow probability to ward 5 (P_{15}) changes with X_5 and X_2 at 9 pm $(h = 7)$.

Three key patterns emerge: (i) the policy prescribes a higher probability of overflow (i.e., a lower P_{11}) when X_1 is high, indicating congestion in the source ward, or when X_2 is low, suggesting greater available capacity in the overflow ward—this aligns with operational intuition; (ii) the policy is more aggressive in overflowing patients during midnight and 9 pm (h =0, 7), but more conservative during daytime hours such as 12 pm $(h = 4)$, reflecting anticipation of discharges that typically occur during the day and free up capacity; and (iii) the right-most plot demonstrates that the policy dynamically balances medical closeness between the primary and overflow wards against system-level congestion, that is, increases the chance of assigning patients to secondary overflow wards (ward 2) when the preferred one (ward 5) is congested. These observations collectively indicate that the PPO policy captures realistic and adaptive operational behavior.

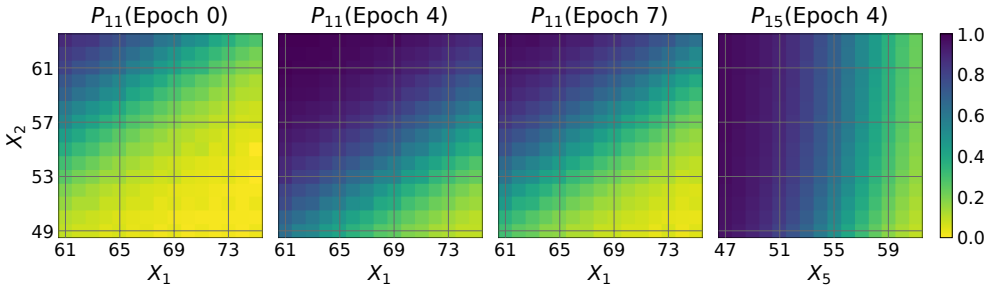


Figure 10 **Policy visualization of class 1 patients on varying X. The vertical axis shows the value of X_2; the horizontal axis shows the value of X_1 in the left three plots and of X_5 in the right-most one.**

F.1.2. Impact of to-depart quantity. Figure 11 shows that the number of patients to be discharged by the end of the day (Y_j) is also an important factor. The left figure illustrates how P_{11} changes with Y_1 and Y_2 at 12 pm (Epoch $h = 4$), while the right figure shows how P_{15} changes with Y_2 and Y_5 at the same time. For fixed Y (the ones we do not vary), we set $Y_j = 0$.

The left figure shows that the policy prescribes a lower probability of overflow (i.e., a higher P_{11}) when Y_1 is high, indicating that more capacity is anticipated to free up soon; or when Y_2 is low, implying that more beds in the overflow ward will remain occupied. The right figure reveals that the policy increases the change of overflowing patients to one of the secondary wards (ward 2) when the preferred overflow ward (ward 5) has fewer patients to be discharged.

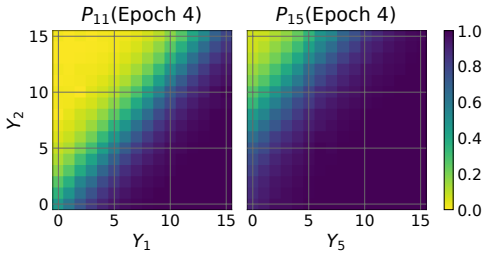


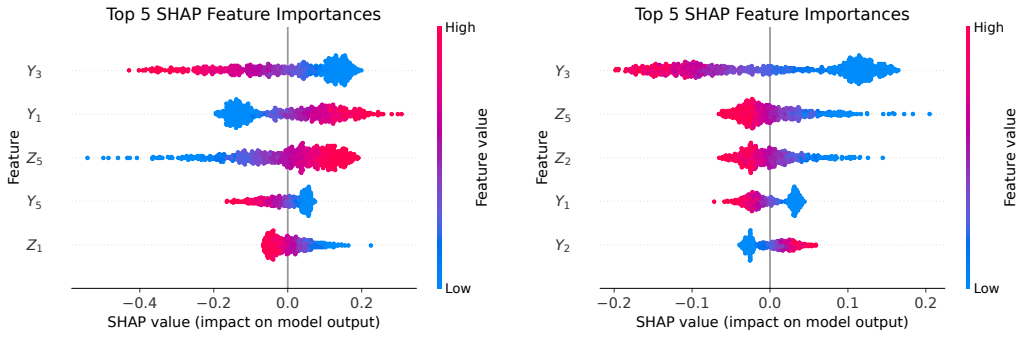
Figure 11 **Policy visualization of class 1 patients on varying Y. The vertical axis shows the value of Y_2; the horizontal axis shows the value of Y_1 in the left plot and of Y_5 in the right one.**

F.2. SHAP Analysis for Nuanced Network Effects

To uncover more subtle and non-intuitive patterns, we further employ SHAP (SHapley Additive exPlanations) analysis to interpret the PPO policy. SHAP quantifies how each input feature of the policy network affects the predicted overflow probabilities for individual samples, with the magnitude of a SHAP value indicating a feature's importance and its sign showing whether the feature increases (positive) or decreases (negative) the output. The output of interest is the probability of *not* overflowing a patient (e.g., P_{ii} for class i), while the input features include (Q, Z, Y). Recall that $Q_i = 0 \vee (X_i - N_i)$ represents the number of waiting patients, and $Z_i = N_i \wedge X_i$ represents the number of in-service patients. We use (Q, Z) instead of X directly to disentangle the effects of waiting and in-service patients, as they play distinct operational roles in overflow decisions.

Figure 12 shows the top five most influential features for the decision not to overflow a class 2 patient (P_{22}) at 12 pm ($h = 4$), and Figure 12(b) presents the corresponding results for class 4 (P_{44}). Source-ward features are excluded from both plots. In both cases, the most salient features prominently include those of the relevant overflow wards (e.g., Y_3, Z_5 and Y_5 for P_{22}; Y_3, Z_2, Y_1 and Y_2 for P_{44}), confirming that the PPO policy captures realistic operational logic consistent with the direct visualization analysis in Section F.1. Beyond these intuitive dependencies, SHAP also highlights additional network-level effects: (i) **Anticipating future congestion:** Figure 12(a) (for class 2) shows that the to-depart count from ward 1 (Y_1, positive correlation) and the occupancy of ward 1 (Z_1, negative correlation) are among the most influential features. This indicates a *proactive preparation* effect. That is, when ward 1 is more congested (higher Z_1) and/or has fewer discharges (lower Y_1), ward 2 overflows its own patients more aggressively (lower P_{22}) to reserve capacity for future class 1 arrivals, since ward 2 is frequently used as a overflow ward for class 1. (ii) **Sharing overflow pools:** For class 4 (P_{44}), congestion (Z_5) in ward 5 strongly influences decisions, despite wards 4

and 5 not directly overflowing to each other. This reflects an indirect "chain effect" arising from the fact that both classes share wards 1, 2, and 3 as overflow choices. That is, when ward 5 becomes more congested (higher Z_5), it is more likely to send patients to these shared overflow wards. Anticipating this competition, the policy instructs ward 4 to overflow its own patients more aggressively (lower P_{44}).



(a) P_{22} (b) P_{44}

Figure 12 **SHAP analysis of probabilities of staying for epoch $h = 4$. Left for class 2 patients staying (not overflow) in ward 2; right for class 4 patients staying (not overflow) in ward 4. Each figure shows the top five most important features, excluding the source-ward features.**

F.3. Comparison Between PPO Policy and ADP Policy

To directly compare what PPO and ADP policies prescribed, we visualize both under identical system states. For the deterministic ADP policy, we plot overflow proportions (Figure 13) to align with the randomized PPO policy representation (Figure 10). The resulting patterns are broadly consistent across methods. Both display time-varying and state-dependent behaviors, such as more aggressive overflow when source wards are congested or overflow wards are less occupied.

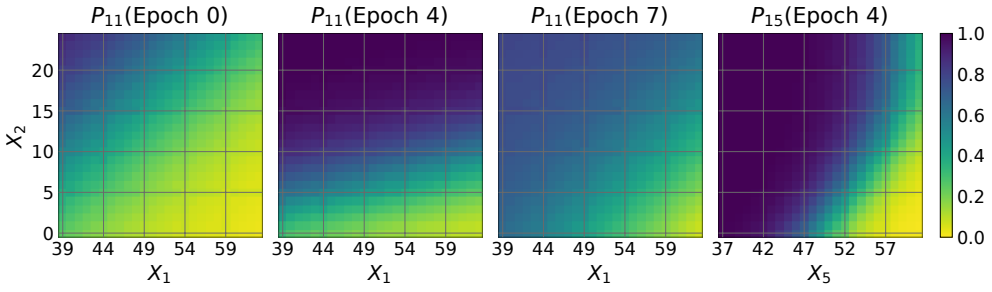


Figure 13 **Policy visualization when varying X of the ADP policy. The vertical axis shows the value of X_2; the horizontal axis shows the value of X_1 in the left three plots and of X_5 in the right-most one.**