Multi-Agent Reinforcement Learning for Intraday Operating Rooms Scheduling under Uncertainty

Kailiang Liu

Department of Mathematics, National University of Singapore

liukl@u.nus.edu

Ying Chen

Department of Mathematics & Risk Management Institute & Center for Quantitative Finance, National University of Singapore

matcheny@nus.edu.sg

Ralf Borndörfer

Department of Mathematics and Computer Science, Freie Universität Berlin

Department of Network Optimization, Zuse Institute Berlin

borndoerfer@zib.de

Thorsten Koch

Faculty of Mathematics and Natural Sciences, Technische Universität Berlin

Department of Applied Algorithmic Methods, Zuse Institute Berlin

koch@zib.de

Abstract

Intraday surgical scheduling is a multi-objective decision problem under uncertainty—balancing elective throughput, urgent and emergency demand, delays, sequence-dependent setups, and overtime. We formulate the problem as a cooperative Markov game and propose a multi-agent reinforcement learning (MARL) framework in which each operating room (OR) is an agent trained with centralized training and decentralized execution. All agents share a policy trained via Proximal Policy Optimization (PPO), which maps rich system states to actions, while a within-epoch sequential assignment protocol constructs conflict-free joint schedules across ORs. A mixed-integer

pre-schedule provides reference starting times for electives; we impose type-specific quadratic delay penalties relative to these references and a terminal overtime penalty, yielding a single reward that captures throughput, timeliness, and staff workload. In simulations reflecting a realistic hospital mix (six ORs, eight surgery types, random urgent and emergency arrivals), the learned policy outperforms six rule-based heuristics across seven metrics and three evaluation subsets, and, relative to an ex post MIP oracle, quantifies optimality gaps. Policy analytics reveal interpretable behavior—prioritizing emergencies, batching similar cases to reduce setups, and deferring lower-value electives. We also derive a suboptimality bound for the sequential decomposition under simplifying assumptions. We discuss limitations—including OR homogeneity and the omission of explicit staffing constraints—and outline extensions. Overall, the approach offers a practical, interpretable, and tunable data-driven complement to optimization for real-time OR scheduling.

Keywords: Healthcare operations management; Operating room scheduling; Dynamic scheduling under uncertainty; Multi-objective optimization; Multi-agent reinforcement learning (MARL); Centralized training–decentralized execution.

1 Introduction

Operating rooms (ORs) are the financial and clinical nexus of the modern hospital. Surgical services are frequently reported to account for a large share of hospital finances—often *60–70% of revenue* and roughly *35–40% of costs*—placing extraordinary pressure on managers to deploy OR time efficiently while protecting access and outcomes [Rothstein and Raval, 2018]. At the same time, the cost of operating room (OR) time typically ranges from approximately $20 per minute to over $100 per minute, with direct and indirect components that rise quickly when schedules run long [Childers and Maggard-Gibbons, 2018]. Compounding these pressures, same-day surgical cancellations—a common form of intraday disruption—occur at material rates (meta-analytic prevalence *about 18%* globally), eroding throughput and patient experience [Abate et al., 2020]. Together, these facts underscore the managerial stakes of day-of-surgery decision making.

Intraday (day-of) OR scheduling is inherently a *multi-objective* decision problem under pervasive *uncertainty*. Managers must balance *patient (case) throughput* and *timely access*—especially for urgent and emergency demand—against *patient waiting times*, *OR idle time*, and *staff overtime*, while sustaining *revenue* and ensuring effective use of surgical teams and equipment. Even with a sound pre-day elective plan, day-of realities routinely erode static schedules: procedure durations are stochastic; urgent cases arrive unpredictably; emergency

batches periodically materialize; and sequence-dependent setup work (cleaning, instrumentation, room turnover) creates frictions that amplify small delays [Denton et al., 2007, Silva and de Souza, 2020]. These dynamics generate a large, evolving state space and a stream of time-critical, sequential decisions that are difficult to solve optimally in real time.

Approach. We address intraday OR scheduling as a *sequential* decision problem and develop a modern learning-based solution rooted in *multi-agent reinforcement learning* (MARL). We model the system as a *cooperative Markov game* in which each OR is an agent acting toward a shared hospital objective. The design follows *centralized training and decentralized execution* (CTDE): during training, a central learner has access to the global state to induce cooperative behavior; at run time, each OR executes the learned policy locally and in real time. We instantiate CTDE with *Proximal Policy Optimization* (PPO), a policy-gradient method known for training stability [Schulman et al., 2017].

To produce *conflict-free* joint decisions without enumerating exponentially large joint action spaces, we employ a *within-epoch sequential assignment* mechanism: at each decision epoch, available ORs take turns (in a fixed or randomized order) selecting actions; later-acting ORs observe earlier choices, eliminating collisions such as assigning the same job twice. The policy observes a rich state summarizing time; room occupancy/progress; type-specific queues; and elective reference start times derived from a pre-day plan.

Crucial operational elements are embedded directly in the *reward design*. First, we generate an *elective pre-schedule* via a mixed-integer program (MIP), which provides reference start times[1]; the reinforcement signal then imposes *type-specific quadratic delay penalties* relative to those references, discouraging excessive waits and reflecting that marginal disutility rises with delay (particularly for higher-urgency classes). Second, we include *sequence-dependent setup times* to capture cleaning and preparation overhead between heterogeneous case types; this incentivizes batching of similar types when clinically appropriate and aligns with well-known turnover sensitivities in OR practice. Third, a *terminal overtime penalty* captures labor and downstream costs of running past regular hours (with implications for staff well-being too); idle penalties can also be incorporated to temper unnecessary slack, though they are not included in the reward function in this study. This single scalar reward induces policies that implicitly trade off *throughput*, *timeliness*, and *workload/costs* without continual retuning.

[1]For *non-elective patients*, the arrival time itself serves as the reference, ensuring that their waiting time is explicitly penalized in the reward.

Empirical setting and evaluation. We evaluate the approach in a high-fidelity hospital simulation that reflects a realistic OR setting: six parallel rooms; eight surgery types with diverse duration distributions; and a stochastic mix of elective, urgent, and emergency arrivals over an intraday horizon of 100 discrete time units (slot length set to match the operating day; e.g., 6 minutes for a 10-hour day). We benchmark the learned policy against six rule-based heuristics across seven metrics (elective and non-elective throughput, unserved emergencies, overtime, delay, revenue, and cumulative reward) and report results overall (across all simulated days), and separately for the emergency-day and non-emergency-day subsets. We further compare the learned policy with an *ex-post MIP oracle* that has full foresight of all future stochastic information—including actual surgery durations and the arrival times and types of non-elective patients. While unattainable in practice, it serves as an upper bound of performance, evaluating overtime, delay, revenue, and cumulative reward on representative test instances.

Case duration distributions, emergency arrival rates, and other parameters are calibrated from hospital data and literature. This test environment allows us to assess how the MARL policy performs under realistic load and variability. The results show that our learning-based policy can improve multiple performance metrics compared to baseline scheduling heuristics and deterministic optimization benchmarks. In particular, the MARL approach reduces average patient waiting times and surgical delays while maintaining high revenue and throughput, illustrating a more balanced trade-off between access and efficiency. Moreover, by adjusting in real-time, the policy handles surge scenarios (e.g. multiple emergencies back-to-back) more gracefully than fixed schedules, thereby improving responsiveness. Beyond quantitative gains, the *policy analytics* offer qualitative understanding – for example, we find that the learned policy implicitly implements a smart strategy of batching similar procedures to limit sequence-dependent setups, and it dynamically resequences elective cases to accommodate high-priority arrivals with minimal disruption. These behaviors align with expert intuition but are achieved here automatically via learning, demonstrating the potential of AI agents to discover effective scheduling heuristics in a complex environment.

Contributions. This study makes the following contributions:

1. New Formulation – MARL for OR Scheduling: We cast intraday OR scheduling as a Markov decision process and design a CTDE MARL framework with a *sequential in-epoch* assignment that guarantees conflict-free joint actions while keeping each agent's action space compact.

2. Comprehensive Reward Design with Operational Metrics: We integrate sequence-

dependent setup dynamics, an elective MIP pre-schedule, *type-specific* urgency via quadratic delay penalties, and a terminal overtime penalty into a unified learning objective, aligning the reward with core hospital performance measures.

3. Empirical Realism and Evaluation: We learn real-time scheduling policies and benchmark them in a realistic six-OR, eight-type testbed against multiple heuristics and an ex post MIP oracle across seven metrics and three evaluation subsets.

4. Insights and Policy Interpretability: We develop *policy analytics* that distill learned regularities into managerially interpretable patterns, clarifying what the policy does and why.

5. Theory: Under simplifying assumptions, we state a *suboptimality bound* for the sequential decomposition relative to a joint assignment, offering initial insight into optimality gaps.

Positioning within the literature. OR planning and scheduling have been extensively studied via *stochastic and robust mixed-integer programming* and related exact/approximate methods, yielding high-quality pre-day elective plans and block allocations under uncertain durations and emergency accommodation. Representative contributions include stochastic sequencing/assignment with hedging against duration uncertainty [Denton et al., 2007], elective block allocation under uncertainty [Majthoub Almoghrabi and Sagnol, 2025, Denton et al., 2010], and models that jointly consider elective and emergency demand [Lamiri et al., 2008]. Distributionally robust formulations have recently advanced planning under ambiguity, incorporating penalties for overtime, idle OR time, and downstream capacity [Shehadeh et al., 2026, Shehadeh, 2022, Shehadeh and Padman, 2021]. These models offer interpretability and control but face acute *intraday* challenges: as operations unfold, the state space explodes (cases start, finish, overrun, or arrive), and re-solving large stochastic programs within short decision windows becomes impractical.

To address real-time reactivity, *approximate dynamic programming* (ADP), rolling-horizon, and simulation-based heuristics approximate downstream value or embed myopic reoptimization [Silva and de Souza, 2020]. While effective in certain settings, these approaches typically require strong structural assumptions, handcrafted contingency rules, or problem-specific value approximations—and can struggle to scale with richer state/action representations and multiple interacting ORs.

Recent work explores *reinforcement learning* (RL) for sequential decision-making in healthcare operations. In OR scheduling, early studies show promise in simplified elective-only contexts using tabular methods [Ribino et al., 2022]. Our contribution advances this

Table 1: Overview of OR Scheduling Research and Reinforcement-Learning Applications

		Objective			Uncertainty		Numerical setup		
Study	**Method**	PT	WT	OT	PA	SD	#ORs	#Types	T
Lamiri et al. [2008]	MIP			✓			–	–	–
Min and Yih [2010]	MIP	✓		✓		✓	–	–	–
Saadouli et al. [2015]	MIP	✓	✓	✓		✓	–	–	–
Riise et al. [2016]	MIP		✓	✓		✓	–	–	–
Wang et al. [2018]	MIP	✓	✓	✓		✓	–	–	–
Fairley et al. [2019]	MIP			✓		✓	–	–	–
Arab Momeni et al. [2022]	MIP	✓		✓		✓	–	–	–
Wang et al. [2023]	MIP	✓		✓		✓	–	–	–
Silva and de Souza [2020]	ADP+MIP	✓	✓		✓	✓	6	4	24
Gökalp et al. [2023]	ADP	✓	✓		✓	✓	3	3	20
Ribino et al. [2022]	MARL	✓	✓		✓	✓	2	1	32
Zhao et al. [2024]	HRL	✓	✓	✓	✓	✓	4	6	48
This study	**MARL**	✓	✓	✓	✓	✓	6	8	100

PT = patient throughput; WT = waiting time cost; OT = overtime cost; PA = non-elective patient arrivals; SD = surgery durations. MIP = mixed-integer programming; ADP = approximate dynamic programming; RL = reinforcement learning; MARL = multi-agent reinforcement learning. Most MIP works consider uncertainty in surgery durations only, not stochastic non-elective arrivals; therefore, numerical comparisons with the intraday MARL are omitted.

line along three dimensions. *First*, we make *ORs*—not patients—the agents, aligning with capacity control and keeping per-agent actions compact while preserving a global view during training. *Second*, we adopt a CTDE deep RL pipeline with a *sequential in-epoch assignment* that yields conflict-free joint actions without enumerating exponentially large joint action spaces; this design also clarifies the information structure (later-acting ORs observe earlier choices within an epoch). *Third*, we embed operational elements central to practice (setup frictions, elective references, urgency-weighted delay, overtime) directly in state and reward, enabling interpretable policy summaries rather than treating RL as a black box. Tables 1 synthesize the surgical-scheduling literature by objectives and uncertainty modeling, and—specifically for RL—by agent/policy design and numerical scale.

Managerial relevance. The proposed scheduler can be viewed as a decision-support layer atop existing planning processes. Hospitals can continue to generate elective plans via established optimization models; the learned policy then adaptively *resequences and reallocates* throughout the day in response to realized durations and non-elective arrivals, thereby protecting timely access while tempering overtime and preserving revenue. Because the policy is trained offline in a digital twin and executed locally per OR in milliseconds, it is deployable without heavy intraday computation. The accompanying policy analytics make its behavior transparent (e.g., urgency-first responses and type batching), which is essential for clinician trust and governance.

Roadmap. Section 2 formalizes the Markov game and reward design. Section 3 presents the MARL architecture, the *sequential in-epoch* mechanism, and the suboptimality bound. Section 4 describes the simulation testbed and benchmarks and reports results with policy analytics. Section 5 discusses limitations (e.g., OR homogeneity, omitted explicit staffing constraints, surgery cancellations), scalability considerations, and extensions; Section 6 concludes.

2 Problem Formulation: Markov Game and Reward

We formalize intraday OR scheduling as a finite-horizon *Markov game* played over a single operating day. The time is divided into T equal-length slots indexed by $t = 1, 2, \ldots, T$, spanning the regular-hours horizon.[2] We consider J operating rooms (ORs), indexed by is $j = 1, 2, \ldots, J$, assumed homogeneous unless stated otherwise. Table 2 summarizes the symbols, units, and modeling conventions used in this section; additional details are provided below.

Patients, types, arrivals, and durations. There are K surgery types. Each patient i is associated with a type $k_i \in \{1, 2, \ldots, K\}$, an arrival time $\alpha_i \in \{0, 1, \ldots, T-1\}$, and a random procedure duration $\xi_i \sim \mathcal{D}_{k_i}$. Let $\mathcal{P} = \bigcup_{t=0}^{T} \mathcal{P}_t$ be the set of all patients. Elective patients $\mathcal{P}_0$ are known pre-day, and we set $\alpha_i = 0$ for $i \in \mathcal{P}_0$; non-elective arrivals occur stochastically during the day, forming sets $\mathcal{P}_t$ at the beginning of slot t. Service is *non-preemptive*. If an OR runs a case of type k immediately after a case of type k', a sequence-dependent setup $\sigma_{k' \to k} \geq 0$ is incurred (we set $\sigma_{k \to k} = 0$).

Queues and eligibility. At decision epoch t, each surgery type k has a queue Q_k^t of *eligible* patients (arrived and not started). When an OR selects type k at t, the first eligible patient in Q_k^t is assigned to that OR and removed from the queue (following the First-In, First-Out (FIFO) rule). Differences across types/queues are handled via the reward (see below), rather than by imposing predefined queue preferences as in rule-based heuristics.

Elective pre-schedule and reference times. A pre-day mixed-integer program (MIP) produces *reference start times* $\{\tau_i\}_{i \in \mathcal{P}_0}$ (and possibly nominal room assignments) for electives. Intraday, the policy is *not* constrained to follow τ_i; instead, τ_i anchors type-specific delay penalties.

[2] In our experiments, $T = 100$ with slot lengths of 6 minutes, but the formulation is invariant to finer discretizations up to rounding.

Agents, state, actions. Each OR is an agent. The *global state* at time t, denoted $s_t \in \mathcal{S}$, aggregates: (i) the time index t; (ii) Q_k^t summaries for all surgery types (e.g., counts and the earliest reference in the queue); and (iii) each OR's occupancy (busy/idle), elapsed time if busy, and last completed type (to compute the next setup). An OR that is *available* at t chooses an action $a_t^j \in \mathcal{A}_j := \{0, 1, \ldots, K\}$, where $a_t^j = 0$ denotes no start (idle) and $a_t^j = k \in \{1, \ldots, K\}$ denotes starting the first eligible patient from Q_k^t at t. An OR that is busy has no effective choice at t (equivalently, $a_t^j = 0$ until the current case completes). The *joint action* is $a_t = (a_t^j)_{j \in \mathcal{J}} \in \mathcal{A} := \prod_{j \in \mathcal{J}} \mathcal{A}_j$. Because multiple available ORs could target the same patient, we realize a_t via a *within-epoch sequential assignment*: free ORs act in a fixed (or randomized) order, each observing earlier selections and removing assigned patients, thereby producing a conflict-free joint decision without enlarging per-OR action alphabets (algorithmic details appear in Section 3).

State transition. Given (s_t, a_t), the next state s_{t+1} is drawn from $P(\cdot \mid s_t, a_t)$ as follows: (i) any OR that starts a case at t becomes busy for the effective processing time $\xi_i + \sigma_{k' \to k_i}$ if the previous completed type on that OR was k'; (ii) ongoing cases decrement their residual by one slot and complete when the residual reaches zero, freeing their ORs for $t+1$; (iii) selected patients are removed from queues; and (iv) new arrivals $\mathcal{P}_{t+1}$ are added to the corresponding Q_k^{t+1}. Let β_i denote the begin time of the served patient i. For each OR j, define its last completion $f_j = \max\{\beta_i + \xi_i + \sigma_{k' \to k_i} : \ i \text{ ran on } j\}$.

Per-patient utility and delay. Each patient i has a type-dependent utility $u_{k_i} \geq 0$ that credits throughput and a type-dependent delay coefficient $\delta_{k_i} \geq 0$ that penalizes waiting. For electives $i \in \mathcal{P}_0$, the waiting time is $\omega_i := [\beta_i - \tau_i]^+$; for non-electives $i \notin \mathcal{P}_0$, the waiting time is $\omega_i := \beta_i - \alpha_i \geq 0$ (patients cannot start before arrival). The quadratic penalty $c_{k_i}\, \omega_i^2$ captures increasing marginal disutility with delay (higher for more urgent types). The per-patient reward is then defined as $u_k - c_{k_i}\, \omega_i^2$.

Overtime. Let $C_o > 0$ be the overtime cost per slot. Define suite-level overtime as the excess of the latest completion beyond regular hours,

$$\mathrm{OT} = \sum_{j=1}^{J} \Big[f_j - T \Big]^+, \tag{1}$$

and penalize it by $C_o \cdot \mathrm{OT}$ at the end of regular hours.

Reward and objective. We use a single *scalar* day-level reward to internalize throughput, timeliness, and overtime. If $\mathcal{I}$ denote the set of patients who receive surgery during the day, the total reward is

$$\sum_{i\in\mathcal{I}} \Big(u_{k_i} - c_{k_i}\,\omega_i^2\Big) \;-\; C_o \cdot \mathrm{OT}. \tag{2}$$

A (stochastic) policy π specifies, at each t, a distribution over joint actions given the state, $\pi(a_t \mid s_t)$. The objective is to maximize expected reward:

$$\max_{\pi} \;\; \mathbb{E}_\pi \Big[\sum_{i\in\mathcal{I}} \Big(u_{k_i} - c_{k_i}\,\omega_i^2\Big) \;-\; C_o \cdot \mathrm{OT}\Big], \tag{3}$$

where the expectation is taken over arrivals, durations, and the policy's internal randomization. The cooperative nature of the game (all agents share the same expected reward) aligns with centralized training and decentralized execution (CTDE): a centralized learner can condition on the full state during training, while each OR executes its local policy in real time; the within-epoch sequential mechanism ensures conflict-free joint actions (Section 3).

$$\begin{aligned}
\max_{\pi} \;\; & \mathbb{E}_\pi \Big[\sum_{i\in\mathcal{I}} \Big(u_{k_i} - c_{k_i}\,\omega_i^2\Big) \;-\; C_o \cdot \mathrm{OT}\Big] \\
\text{s.t.} \;\; & \mathrm{OT} \;=\; \sum_{j=1}^{J} \big[\, f_j - T \,\big]^+, \\
& f_j = \max\{\beta_i + \xi_i + \sigma_{k'\to k_i} : i \text{ ran on } j\},\; j \in [J] \\
& \beta_i \ge \alpha_i, \qquad i \notin \mathcal{P}_0 \\
& \xi_i \sim \mathcal{D}_{k_i}, \qquad i \in \mathcal{P} \\
& \omega_i \;=\; \begin{cases} [\beta_i - \tau_i]^+, & i \in \mathcal{P}_0, \\ \beta_i - \alpha_i, & i \notin \mathcal{P}_0, \end{cases} \\
& s_{t+1} \sim P(\cdot \mid s_t, a_t), \qquad t = 0, \ldots, T-1, \\
& a_t \in [K], \qquad t = 0, \ldots, T-1, \\
& \beta_i \ge 0,\; \omega_i \ge 0, \qquad i \in \mathcal{P} \\
& f_j \ge 0, \qquad j \in [J].
\end{aligned}$$

Assumptions. (i) Homogeneous ORs (any type may run in any room; specialized rooms can be handled by restricting admissible types per room); (ii) non-preemptive service (once started, a surgery runs to completion); (iii) FIFO within type queues (urgency handled through $\{p_k, \delta_k\}$ rather than hard priorities); (iv) setup times $\sigma_{k'\to k}$ are known (or known

in distribution and incorporated into effective durations); and (v) No explicit staff coupling (surgeon/anesthesia/nursing constraints are not modeled explicitly). Relaxations are discussed in Section 5.

Rationale for a unified scalar reward. Equation (2) integrates *multiple operational objectives* into a single learning signal, avoiding repeated hand-tuning of scenario-specific scalarization weights. The type weights u_k preserve heterogeneity in case value (revenue/priority); the convex delay term $c_k\,\omega_i^2$ encodes timeliness against a_i or arrival-anchored term τ_i for non-electives; and the terminal overtime term $C_o \cdot \mathrm{OT}$ internalizes staffing and downstream costs. Fixing (u_k, c_k, C_o) at design time yields a stable training objective across demand and duration regimes, while allowing the learned policy to *implicitly* trade off throughput, timely access, and overtime. Alternative formulations—such as explicit multiobjective optimization (e.g., ε-constraint or Pareto-front methods) or service-level constraints for non-electives—can be accommodated in the same framework (e.g., via penalties or constrained RL), but are beyond the scope of this paper.

From pre-day plans to intraday control. The pre-day MIP produces the elective reference starts $\{\tau_i\}_{i\in P_0}$ that reflect surgeon availability and clinic commitments (for the full MIP formulation and parameterization, see Appendix A.2. Intraday, the policy π may resequence electives and allocate capacity to non-electives as uncertainty unfolds; the convex delay penalty ensures that deviations from τ_i are driven by realized conditions (e.g., overruns, urgent arrivals) rather than arbitrary reordering. Section 4 evaluates these trade-offs empirically.

3 Multi-Agent Reinforcement Learning and Sequential Assignment

Building on the Markov game in Section 2, we develop a multi-agent reinforcement learning (MARL) solution trained with *proximal policy optimization* (PPO) [Schulman et al., 2017]. We describe a centralized-training, decentralized-execution (CTDE) MARL approach in which each OR j acts as an agent. Agents share a common policy (actor) π_θ and a common value function (critic) V_ϕ with θ and ϕ parameterizing the actor and critic networks respectively. At runtime, we realize a feasible *joint* decision by a within-epoch *sequential assignment* protocol that prevents conflicts without enumerating the exponential joint-action space. We detail the learning algorithm and then provide a theoretical guarantee relative to an optimal centralized planner.

Table 2: Notation

Symbol	Meaning	Units/Notes
T	Number of discrete slots in the day	Slots
J	Number of operating rooms (agents)	ORs
K	Number of surgery types	Types
$\mathcal{P}_t$	Set of non-elective patients arriving at slot t	Arrival set
$\mathcal{P}_0$	Set of elective patients known pre-day	Electives
$\mathcal{I}$	Set of patients who received surgery	
α_i	Arrival time of patient i	Slots; $\alpha_i = 0$ for $i \in P_0$
τ_i	Elective reference start time from pre-schedule	Slots (anchor for delays)
$\xi_i \sim \mathcal{D}_{k_i}$	Random duration of patient i	Slots (or minutes)
$\sigma_{k' \to k}$	Setup time from type k' to k	Slots; $\sigma_{k \to k} = 0$
β_i	Actual start time of i	Slots
f_j	Last completion time on OR j	Slots
ω_i	Waiting time of i	Elective: $[\beta_i - \tau_i]^+$; non-elective: $\beta_i - \alpha_i$
u_k	Throughput utility weight for type k	Dimensionless (value/revenue/priority)
c_k	Delay penalty coefficient for type k	Weight on ω_i^2
C_o	Overtime cost per slot	Overtime penalty $C_0 \cdot \mathrm{OT}$
s_t	Global state at time t	Queues, OR statuses, time
a_t^j, a_t	Action of OR j; joint action at time t	$a_t^j \in \{0, 1, \ldots, K\}$
r_t^j	Immediate reward of OR j at time t	Defined in (5).
r_T	Terminal reward	$-C_0 \cdot \mathrm{OT}$
$P(\cdot \mid s_t, a_t)$	State transition kernel	Markovian dynamics
π	Scheduling policy	$\pi(a_t \mid s_t)$

Key symbols used in Section 2. Positive part $[x]^+ = \max\{x, 0\}$. Queues follow FIFO within type. Overtime penalty OT is defined in (1).

3.1 CTDE with a Shared Policy and Local Observations

Let π_θ be a stochastic policy with parameters θ, shared across agents, and let V_ϕ be a value function with parameters ϕ. At decision epoch t, the global state is s_t (Section 2). Agent j augments s_t with its *local information* ℓ_t^j to form

$$s_t^j \;=\; [\, s_t,\; \ell_t^j \,],$$

where ℓ_t^j includes agent-specific features (e.g., the actions $\{a_t^1, \ldots, a_t^{j-1}\}$ of earlier agents in our sequential protocol below).

The shared policy maps s_t^j to a distribution over the per-agent action space $\mathcal{A}_j = \{0, 1, \ldots, K\}$, such that

$$\begin{aligned} &a_t^j \sim \pi_\theta(\cdot \mid s_t^j), \\ &a_t^j = \begin{cases} 0, & \text{idle}, \\ k \in \{1,\ldots,K\}, & \text{start next patient of type } k. \end{cases} \end{aligned} \tag{4}$$

Agents that are busy at t have no effective choice (equivalently, $a_t^j = 0$ until completion).

3.2 Within-Epoch Sequential Assignment (Conflict-Free Joint Action)

At each epoch t, agent j observes (s_t, ℓ_t^j), draws a_t^j, and if $a_t^j = k > 0$ assigns the first eligible patient from Q_k^t to OR j, removing that patient from the queue. Agents that are busy at t have their action space restricted to $\{0\}$. This produces a conflict-free joint action

$$a_t \;=\; (a_t^1, \ldots, a_t^J) \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_J$$

without expanding per-agent action alphabets. The environment then advances to $t+1$ under the Markovian dynamics in Section 2.

3.3 Reward

In MARL algorithm, when agent j starts a patient i of type k at time t (i.e., $a_t^j = k > 0$), it receives the *immediate* reward

$$r_t^j = u_k - c_k\, \omega_i^2, \qquad \omega_i = \begin{cases} [\beta_i - \tau_i]^+, & i \in \mathcal{P}_0, \\ \beta_i - \alpha_i, & i \notin \mathcal{P}_0, \end{cases} \tag{5}$$

and $r_t^j = 0$ for $a_t^j = 0$. The per-epoch reward is $r_t = \sum_{j=1}^{J} r_t^j$. At the end of regular hours ($t = T$), a *terminal* penalty captures overtime in (1):

$$r_{T+1} \;=\; -\,C_o \cdot \mathrm{OT}, \qquad \mathrm{OT} = \sum_{j \in \mathcal{J}} \Big[f_j - T \Big]^+. \tag{6}$$

Consistent with Section 2, the goal is to maximize $\mathbb{E}_\pi[\sum_{t=1}^{T} r_t + r_{T+1}]$.

3.4 Policy Gradient with PPO and Generalized Advantage Estimation

We train (π_θ, V_ϕ) on simulated day-long trajectories $(s_0, a_0, r_1, \ldots, s_{T-1}, a_{T-1}, r_T, s_T)$. Let $\gamma \in [0,1]$ be the discount factor. Define the temporal-difference residuals $\delta_t = r_{t+1} + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$ and the generalized advantage estimator (GAE) with parameter $\lambda \in [0,1]$:

$$\hat{A}_t \;=\; \sum_{\ell=0}^{T-t} (\gamma\lambda)^\ell \, \delta_{t+\ell} \quad \text{with} \quad \hat{A}_T = 0. \tag{7}$$

The critic minimizes a mean-squared value error,

$$L^{\mathrm{VF}}(\phi) \;=\; \mathbb{E}\Big[\big(V_\phi(s_t) - \hat{R}_t \big)^2 \Big], \; \hat{R}_t = \hat{A}_t + V_{\phi_{old}}(s_t), \tag{8}$$

and the actor maximizes the PPO clipped surrogate,

$$\begin{aligned} L^{\mathrm{CLIP}}(\theta) &= \mathbb{E}\big[\min\{ r_t(\theta), \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon) \} \hat{A}_t \big], \\ r_t(\theta) &= \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t \mid s_t)}. \end{aligned} \tag{9}$$

Here, $r_t(\theta)$ is the importance sampling ratio and ϵ controls how much the new policy is allowed to deviate from the previous policy.

To encourage exploration and prevent premature policy collapse, we additionally include an entropy bonus $\lambda_{\mathrm{ent}} \, \mathcal{H}(\pi_\theta(\cdot \mid s_t^j))$ in the actor objective, where

$$\mathcal{H}\big(\pi_\theta(\cdot \mid s_t)\big) = -\sum_a \pi_\theta(a \mid s_t) \, \log \pi_\theta(a \mid s_t). \tag{10}$$

is the policy entropy and $\lambda_{\mathrm{ent}} > 0$ controls exploration strength. The resulting actor loss becomes

$$L_{\mathrm{actor}}(\theta) = L^{\mathrm{CLIP}}(\theta) \;+\; \lambda_{\mathrm{ent}} \, \mathbb{E}\Big[\mathcal{H}\big(\pi_\theta(\cdot \mid s_t)\big) \Big], \tag{11}$$

Gradients are computed on mini-batches drawn from a replay buffer built from multiple

Algorithm 1 Trajectory Collection with Shared Policy (CTDE, Sequential Assignment)

Initialize environment; observe s_0
for $t = 0, \ldots, T-1$ **do**
  for $j = 1$ to J **do** ▷ Within-epoch sequential assignment
    Build local obs ℓ_t^j (incl. earlier actions $\{a_t^1, \ldots, a_t^{j-1}\}$); set $s_t^j = [s_t, \ell_t^j]$
    Sample $a_t^j \sim \pi_\theta(\cdot \mid s_t^j)$, or set $a_t^j \leftarrow 0$ if agent j is busy at t
    if $a_t^j = k > 0$ **then**
      Assign first patient in Q_k^t to OR j; compute r_t^j via (5)
    else
      $r_t^j \leftarrow 0$
    end if
  end for
  Form joint $a_t = (a_t^1, \ldots, a_t^J)$ and team reward $r_t = \sum_j r_t^j$
  Environment step to s_{t+1}
end for
Compute terminal r_T via (6)
return Trajectory $(s_0, a_0, r_1, \ldots, s_{T-1}, a_{T-1}, r_T, s_T)$

day-long trajectories. See Alg. 1 for action sampling. Alg. 2 outlines the overall procedure of our multi-agent training framework, highlighting how shared policy and critic components interact with environment dynamics across time steps.

Remark (centralized planner vs. MARL). From another perspective, one may consider a centralized agent that makes scheduling decisions on behalf of all ORs. In this setting, the objective is to maximize the cumulative reward over time by selecting *joint* action $a_t \in \mathcal{A}(s_t)$, leading to an exponential action space. Our sequential protocol preserves feasibility and coordination (via shared parameters and conditioning on earlier actions) while remaining tractable; see Appendix A.1 for more details.

3.5 Theory: Dynamic Programming, CTDE Optimality, and Suboptimality Bounds

We establish the theoretical foundations of our multi-agent reinforcement learning (MARL) framework through results connecting centralized reinforcement learning, weak-coupling decomposition, and the optimality of centralized training with decentralized execution (CTDE). We also derive regret bounds that quantify potential suboptimality when these conditions are partially violated.

Algorithm 2 MARL Training Loop (Shared Actor–Critic, PPO)

Initialize π_θ, V_ϕ, empty buffer $\mathcal{B}$
for $m = 1$ to MaxIters **do**
    for $e = 1$ to M_{traj} **do**
        Run Alg. 1; append all transitions to $\mathcal{B}$
    end for
    for $u = 1$ to M_{upd} **do**
        Sample mini-batch from $\mathcal{B}$
        Update critic by GD step on (8)
        Update actor by GD step on (9) (+ entropy)
    end for
    Clear $\mathcal{B}$
end for
return trained π_θ

Bellman optimality and joint planning. Let $V_t^*(s)$ denote the optimal value function at time t given state s. The centralized Bellman recursion can be expressed as

$$\begin{aligned} V_t^*(s) &= \max_{a \in \mathcal{A}(s)} \big\{ R(s,a) + \mathbb{E}[V_{t+1}^*(s') \mid s, a] \big\}, \\ V_T^*(s) &= 0. \end{aligned} \tag{12}$$

where $\mathcal{A}(s)$ is the set of conflict-free joint actions, and $R(s,a)$ aggregates the per-patient rewards and terminal overtime penalties. Equation (12) defines the centralized reinforcement learning for the hospital scheduling problem, in which all operating rooms (ORs) are jointly optimized at each decision epoch.

Lemma 1 (Equivalence under weak coupling)**.** *Fix an epoch t and assume: (A1) no sequence-dependent setups ($\sigma_{k' \to k} \equiv 0$); (A2) starting a case at t affects only its own queue (OR-separable transitions); (A3) immediate rewards add across ORs (no cross-OR penalties); and (A4) no contention for a single remaining patient in any type selected by multiple ORs.*[3] *Then the per-epoch joint maximization in* (12) decomposes*, and greedy sequential selection (our within-epoch protocol) attains the same joint maximizer as the centralized selection.*

Proof. Proof. Under (A1)–(A4), conditional on earlier picks, marginal gains of distinct ORs are independent; selecting the top-J' marginals yields the joint maximizer by a standard exchange argument. □ □

[3] A4 can be enforced operationally by preventing simultaneous pulls when $|Q_k^t| = 1$, which our sequential protocol already guarantees.

Proposition 2 (Optimality of CTDE under Weak Coupling)**.** *Consider the cooperative Markov game defined in Section 2 over horizon T, with joint state $s_t \in \mathcal{S}$, joint action $a_t = (a_t^1, \ldots, a_t^{J'}) \in \mathcal{A}(s_t)$, and reward $R(s_t, a_t)$. Suppose Assumptions (A1)–(A6) hold: (A5) Centralized training, sufficient local information: During training, the critic $V_\phi(s)$ conditions on the full state s_t; during execution, each agent observes a sufficient local statistic s_t^j such that its local argmax coincides with the centralized argmax. (A6) Exact representation and convergence: The actor–critic networks are expressive enough to represent V^* and π^*, and PPO converges to the fixed point of policy iteration.*

Then the CTDE policy with shared actor–critic and sequential within-epoch assignment satisfies

$$\pi^*_{\mathrm{CTDE}}(s_t) = \arg\max_{a_t \in \mathcal{A}(s_t)} Q_t^*(s_t, a_t),$$

and thus achieves the same performance as the centralized optimum.

Proof. Proof. By Lemma 1, under (A1)–(A4) the joint value decomposes additively across ORs, and sequential greedy selection achieves the same maximizer as the centralized planner. Under (A5)–(A6), the centralized critic V_ϕ accurately estimates V^π, and the PPO update step maximizes the expected advantage $\mathbb{E}[\hat{A}_t^{(j)}]$, where $\hat{A}_t^{(j)} \approx q_t^{(j)}(s_t, a_t^j \mid a_t^{1:j-1}) - V_\phi(s_t)$. Because all agents share the same actor and critic, each performs an identical local improvement step as in the centralized Bellman recursion. Hence, the CTDE improvement step is equivalent to centralized policy iteration, which converges to π^*. Therefore, $\pi^*_{\mathrm{CTDE}} = \pi^*$. □ □

Theorem 3 (Suboptimality bound)**.** *When weak coupling is violated (e.g., due to setup dependencies, queue tightness, or overtime coupling), the sequential CTDE policy may incur a bounded loss relative to the centralized optimum. Let $Q^*(s_t, a)$ denote the optimal one-step lookahead value and define the per-epoch externality*

$$\Gamma_t(s_t) = \max_{a \in \mathcal{A}(s_t)} Q^*(s_t, a) - \sum_{j=1}^{J'} \max_{a^j \in \mathcal{A}_j} Q^{(j)}(s_t, a^j \mid a^{1:j-1}). \tag{13}$$

Then along any trajectory,

$$V_1^*(s_1) - V_1^{\mathrm{seq}}(s_1) \le \mathbb{E}\left[\sum_{t=1}^{T} \Gamma_t(s_t)\right]. \tag{14}$$

Moreover, for the reward defined in Eqs. (5)–(6),

$$V_1^*(s_1) - V_1^{\text{seq}}(s_1) \le \mathbb{E}\Big[\sum_{i\in\mathcal{I}} \delta_{k_i}\big((\omega_i^{\text{seq}})^2 - (\omega_i^*)^2\big) + C_o\big(\text{OT}^{\text{seq}} - \text{OT}^*\big)\Big], \tag{15}$$

and since $\omega_i \le T$,

$$V_1^*(s_1) - V_1^{\text{seq}}(s_1) \le \mathbb{E}\Big[2T\sum_{i\in\mathcal{I}} \delta_{k_i}\big(\omega_i^{\text{seq}} - \omega_i^*\big) + C_o\big(\text{OT}^{\text{seq}} - \text{OT}^*\big)\Big]. \tag{16}$$

Proof. Proof. Decompose the regret epoch-by-epoch and apply the performance-difference identity; $\Gamma_t(s_t)$ measures the loss from not selecting the globally best joint action. For the quadratic delay penalty and additive overtime cost, the loss can occur only via (i) additional waiting for some patients and (ii) additional overtime, yielding (15). Bound (16) follows from $(\omega_i^{\text{seq}})^2 - (\omega_i^*)^2 = (\omega_i^{\text{seq}} + \omega_i^*)(\omega_i^{\text{seq}} - \omega_i^*) \le 2T(\omega_i^{\text{seq}} - \omega_i^*)$. □ □

Corollary 4 (Sequential policy gap for one urgent case)**.** *Consider a single high-priority arrival at time* $\bar{t}$ *that the optimal policy serves immediately (zero wait). If under* π_{seq} *the patient waits* Δ *slots (all ORs occupied at* $\bar{t}$*), then*

$$\mathbb{E}_{\pi^*}[\mathcal{R}] - \mathbb{E}_{\pi_{\text{seq}}}[\mathcal{R}] \le \delta_k\, \Delta^2, \tag{17}$$

where k *is the patient's type.*

Proof. Proof. Immediate from (15) by specializing to one affected case and noting that other contributions cancel or are dominated in this stylized setting. □ □

Remarks and interpretation. Lemma 1 and Proposition 2 jointly establish that under weak coupling, the sequential CTDE policy exactly reproduces the centralized Bellman-optimal policy. When coupling is present, Theorem 3 quantifies the deviation, showing that performance loss arises solely from excess waiting and overtime externalities. Corollary 4 provides an intuitive special case illustrating this loss. In our OR scheduling environment, the weak-coupling assumptions hold approximately because setup times are OR-local, queues are disjoint, and overtime is terminal. Empirically (Section 4), the learned MARL policy achieves near-optimal performance with small oracle-normalized regret, demonstrating that CTDE offers a scalable and theoretically sound approximation to centralized dynamic programming for large-scale hospital scheduling.

4 Numerical Experiments

This section evaluates the proposed MARL policy under realistic intraday operating conditions using a detailed simulation environment. The design, metrics, and statistics are chosen to (i) test the day-level value of the method under practically relevant loads/uncertainty; (ii) investigate the mechanisms suggested by our model and theory; and (iii) assess robustness. All simulator primitives (durations, arrivals, setup times, elective mix, and reward weights) are parameterized from proprietary hospital data and perioperative consulting engagements[4].

4.1 Objectives

We evaluate whether the learned MARL policy improves the scalar objective (2) and its clinical components-urgent access, elective timeliness, and overtime—relative to established heuristics and an ex post MIP oracle computed with full *future knowledge*, including actual surgery durations and the arrival times and types of non-elective patients. The oracle thus represents an *unfair comparator*—since such information is unavailable to any implementable policy—but provides a valuable *upper bound* for benchmarking. By comparing against this ex post optimum, we quantify both the *regret* and the attainable performance headroom of the learned policy.

4.2 Environment, Data, and Calibration

Suite/horizon. We simulate $J = 6$ homogeneous ORs over $T = 100$ slots (slot length 6 minutes for a 10 hour day), yielding 600 assignable slot-units of regular-hours capacity. Sequence-dependent setups $\sigma_{k' \to k}$ encode cleaning, turnover, and instrumentation.

Types and durations. We consider $K = 8$ clinical surgery types but model procedural time via *four duration classes—minor*, *moderate*, *long*, and *complex*—with two clinical types per class. For each surgery type k, the duration law $\mathcal{D}_k$ is Gamma, calibrated to empirical class-wise means which also aligns with [Azar et al., 2022, Choi and and, 2012]; durations are rounded up to the nearest integer time unit. For the four elective types, expected duration increases monotonically with type index.

Arrivals and case mix. Electives ($\mathcal{P}_0$) are fixed pre-day via the MIP plan (Appendix A.2); Non-electives arrive stochastically: *urgent* cases follow a homogeneous Poisson process with per-slot rate λ_u (shared across urgent types), while *emergencies* follow a separate day-level Bernoulli event with probability ϵ; if the event occurs, a batch of five emergency patients is

[4]Due to NDAs, the raw data are not public; however, the synthetic generator matches the first/second moments, and time-of-day profiles observed operationally.

released (at most one event/day). At each epoch, idle ORs choose between eligible electives and any waiting non-electives via the unified reward and conflict-free sequential assignment; surgeries are non-preemptive.

Mixing of electives and non-electives. Because electives are present from $t = 0$ while non-electives arrive stochastically, the composition of queues evolves endogenously. When there are no urgent or emergency patients waiting, the policy schedules electives (potentially batching by type to mitigate setups); upon an urgent or emergency arrival, the new patient becomes immediately eligible and competes with electives via the shared reward (higher p_k and/or δ_k can pull capacity toward high-priority cases). This mechanism yields realistic day-to-day variation: some simulated days contain no emergency event, whereas others include a sudden emergency batch.

Why fix intensities within arrivals? Non-elective demand is exogenous; fixing λ_u and ϵ isolates policy effects and enables fair, low-variance paired comparisons (via common random numbers), while reflecting the operational stability of hour-of-day arrival profiles.

Reward weights. Type utilities u_k, delay coefficients c_k, and overtime cost C_o reflect value/priority, timeliness sensitivity, and labor/recovery burdens.

4.3 Simulation Design

Table 3 displays the parameters used in the simulation.

Electives. $N_e = 55$ cases distributed by duration class $(28, 19, 5, 3)$ from *minor* to *complex*, consistent with typical hospital mix. Pre-day references $\{\tau_i\}$ are obtained via MIP and act as nonbinding anchors intraday.

Urgent arrivals. Three urgent types arrive as a homogeneous Poisson process with per-slot rate $\lambda_u = 0.08$; each arrival is assigned uniformly to one of the three urgent types.

Emergency arrivals. Emergencies represent sudden, high-priority cases (e.g., major traffic accidents). Each time slot has a $\epsilon = 5\%$ chance of triggering an "emergency event," which immediately generates a batch of five patients (at most one event per day).

Workload and regimes. Under this scenario, the expected volume is

$$\mathbb{E}[N] \;=\; 55 \;+\; 100\lambda_u \;+\; 5(1-(1-\epsilon)^{100}) \;\approx\; 65,$$

with expected cumulative workload ≈ 677 slot-units, i.e., above regular-hours capacity, inducing realistic overtime pressure.

Evaluation sets. The test set comprises **500** independent day-episodes (disjoint seeds) with different non-elective arrival realizations and type mixes, while electives volume and pre-schedule are fixed. Because the emergency event is stochastic, only a subset of days contains emergencies (empirically $\approx 39.42\%$). Results are reported for (i) *All Days* (all 500 test days), (ii) *Emergency Days* (subset with an event), and (iii) *Non-Emergency Days* (subset without an event).

Table 3: Simulation configuration

Parameter	**Description**	**Value / Law**
T	Horizon (slots)	100
J	Operating rooms	6
K	Clinical types	8 (two per duration class)
N_e	Elective volume	55 (28/19/5/3 by class)
λ_u	Urgent arrival rate (per slot)	0.08 (homogeneous Poisson)
ϵ	Emergency event probability (per day)	0.40 (if yes: batch of 5)
d_k	Duration by class (Gamma)	rounded up to slots
	Minor ($k \in \{1, 5\}$)	$\Gamma(\text{shape} = 2,\ \text{scale} = 2)$
	Moderate ($k \in \{2, 6\}$)	$\Gamma(3, 3)$
	Long ($k \in \{3, 7\}$)	$\Gamma(6, 4)$
	Complex ($k \in \{4, 8\}$)	$\Gamma(7, 5)$
p_k	Utility by class	
	Minor ($k \in \{1, 5\}$)	8
	Moderate ($k \in \{2, 6\}$)	12
	Long ($k \in \{3, 7\}$)	30
	Complex ($k \in \{4, 8\}$)	50
C_0	Overtime cost per slot	0.005
δ_k	Delay penalty coefficient by class	
	Elective ($k \in \{1, 2, 3, 4\}$)	0.002
	Urgent ($k \in \{5, 6, 7\}$)	0.004
	Emergency ($k = 8$)	0.005
$\sigma_{k' \to k}$	Setup time	type-pair specific

4.4 Policies and Baselines

We benchmark the proposed MARL policy against five transparent rule-based heuristics, a pre-schedule–driven policy, and a ex post MIP oracle. The heuristics capture common

dispatch logics in OR control (duration- and urgency-based), while the oracle offers a challenging upper bound.

4.5 Alternative Scheduling Methods for Comparison

MARL (Proposed). Shared actor–critic PPO with within-epoch *sequential assignment* to produce conflict-free joint actions (Section 3). One shared policy and value function are trained centrally; agents execute locally at runtime.

Heuristics.

- SPT-U (shortest processing time first; non-electives before electives).
- LPT-U (longest processing time first; non-electives before electives).
- NE-LPT (non-elective first; long-first tie-break within group).
- E-LPT (elective first; long-first tie-break within group).
- NE-SPT (non-elective first; short-first tie-break within group).
- PRE-S. (follow pre-scheduling plan; start non-electives immediately if capacity).

Oracles (unfair comparators). The EX POST MIP ORACLE has access to all future arrivals and realized case durations and determines the optimal daily schedule by solving a mixed-integer programming model. Though infeasible in practice, this benchmark provides a rigorous upper bound for quantifying regret and attainable performance improvements (details in Appendix A.3.

The heuristics above were selected because they span the principal decision logics seen in OR control—duration-based dispatching (SPT/LPT), urgency-based precedence (non-elective vs. elective first), and adherence/batching rules (pre-schedule, setup-aware)—while remaining fast, transparent, and representative of practices used by hospitals. The MIP-based *oracle* is included solely as a challenging upper bound: it assumes perfect foresight of future arrivals and realized durations, which is unattainable operationally, and is therefore an *unfair* comparator used only to quantify regret and headroom; it is not a feasible operational benchmark.

4.6 Evaluation, and Statistics

PPO uses fixed hyperparameters across scenarios; training/validation splits use disjoint seeds. Each trained policy is evaluated on **500** independent day-episodes per scenario. We employ *common random numbers* (paired seeds) across policies and oracles to reduce

variance in paired comparisons. For every metric, we report the sample mean $\pm$*(sd)* over the 500 episodes; when stratifying by emergency incidence, results are shown for *All Days*, *Emergency Days*, and *Non-Emergency Days*.

Primary outcome. Cumulative reward (**CR**) $\mathbb{E}[\sum_{i\in\mathcal{I}}\left(u_{k_i} - c_{k_i}\,\omega_i^2\right) - C_o \cdot \mathrm{OT}]$ (Eq. (2))

Secondary outcomes. Throughput:**PT(E)** (completed electives) and **PT(NE)** (completed non-electives); **Unserved(NE)**: count of non-elective patients not started within the day; **OT**:overtime in slots; **Delay**: waiting time in slots, measured as the mean of absolute elective delay $|\beta_i - \tau_i|$ and non-elective wait $\beta_i - \alpha_i$ over all cases within a day; **Revenue**: proxy value $\sum_{i\in\mathcal{I}} u_{k_i}$ generated by completed cases.

4.7 Main Results

We compare MARL against all baselines on seven metrics—PT(E), PT(NE), Unserved(NE), OT, Delay, Revenue, and the unified objective CR—using the full test set and the two strata defined by emergency incidence. Raw results appear in Table 4; normalized scores and radar plots are shown in Table 5 and Fig. 1.

Overall performance. Across *All Days*, MARL attains the highest cumulative reward (CR) among all alternative methods, indicating superior overall efficiency. Its variability (74.28; Table 4) remains relatively low among alternative methods at a similar performance level (CR above 700), suggesting stable performance. In particular, while SPT_U (651.80 ± 57.81) and SPT_E (672.90 ± 65.00) show slightly smaller variance, they do so at substantially lower reward levels. Overall, MARL achieves the best average normalized score (**0.81**; Table 5), reflecting balanced gains rather than single-metric dominance. In the *Emergency Days* stratum, E_LPT achieves the largest CR but does so by sacrificing non-elective service (low PT(NE), high Unserved) and incurring high overtime (OT). MARL achieves the *best overall balance* (top average score = **0.73**) with strong CR and materially lower OT/Delay than elective-first rules. In *Non-Emergency Days*, MARL again leads on CR and posts the top average score (**0.87**), with low Unserved(NE), moderate OT, and low Delay.

Trade-offs. Heuristics exhibit clear trade-offs visible in Fig. 1. E_LPT maximizes PT(E) and Revenue but under-serves non-electives (high Unserved) and often incurs more OT in emergency days; SPT_U boosts PT(E) yet suffers from OT and Delay, depressing CR. The Pre-s policy controls OT and Delay but leaves reward on the table (lower Revenue/CR). NE_LPT performs relatively well when no emergency occurs but degrades when an emergency arrives. In contrast, MARL's polygon is consistently well-rounded: it maintains high

PT(NE) and low Unserved while limiting OT and Delay, yielding superior CR.

Policy interpretability. Ex-post analytics (Section 4.9) reveal that the learned policy (i) *Batches* compatible types when setup penalties would otherwise accumulate, (ii) *Length-aware*: longer or more complex cases are placed earlier in the day, while shorter ones fill later slots, following an "early-long, late-short" pattern, (iii) *Adaptive prioritization*: the policy raises the priority of urgent and emergency cases, inserting them adaptively, and (iv) *Load balancing* long or complex cases are staggered across operating rooms, preventing simultaneous congestion and promoting balanced utilization throughout the system. These patterns align with our unified reward design and provide face-valid, managerial explanations for the balanced performance observed in Fig. 1.

Regret Analysis. To assess how closely the learned policy approaches the theoretical upper bound, we conduct a regret analysis relative to an oracle benchmark with full knowledge of future arrivals and realized durations (Section 4.10) . The oracle represents an unattainable but informative performance ceiling, and the difference in objective value between the learned policy and the oracle quantifies the regret. Two dominant components account for this gap: (i) the oracle's knowledge of future arrivals enables timely and precise responses to non-elective cases, substantially reducing waiting-related penalties and service delays, and (ii) with full information about all arrivals and durations, the oracle can globally coordinate the schedule to maximize room utilization and revenue, achieving near-perfect case coverage within capacity limits.

Taken together, the tables and radar plots demonstrate that MARL is *robust and effective* for the multi-objective OR scheduling problem: it dominates or nearly dominates across metrics, delivers the highest unified reward, and preserves service quality for non-electives without incurring excessive overtime—precisely the balanced behavior sought in practice.

4.8 Computing Facility and Runtime

All experiments were conducted on an Apple M2 chip with an 8-core CPU, an 8-core integrated GPU, and 8 GB of unified memory. Training each scenario's MARL policy required 2.07 hours of wall-clock time over 1,000 epochs, with 16 simulated day-episodes per epoch (i.e., 16 full daily scheduling simulations used for gradient updates). The average simulation throughput was 4,269 environment steps per second. During inference, per-epoch action selection required 0.374 milliseconds on average, and each full-day simulation took 0.192 seconds including the within-epoch sequential assignment.

Table 4: Performance comparison of the proposed MARL scheduling framework and rule-based scheduling heuristics under all samples and emergency-case samples

	All Days						
	PT(E)	PT(NE)	Unserved(NE)	Overtime	Delay	Revenue	CR
SPT_U	54.07 ± 1.33	8.80 ± 3.12	0.74 ± 1.38	93.67 ± 45.51	36.20 ± 2.07	892.72 ± 69.78	651.80 ± 57.81
LPT_U	39.19 ± 13.95	8.68 ± 2.92	0.86 ± 1.57	39.11 ± 37.67	22.78 ± 4.62	824.90 ± 73.45	727.64 ± 85.36
NE_LPT	38.51 ± 14.20	9.33 ± 3.52	0.22 ± 0.78	40.45 ± 37.34	21.45 ± 4.29	823.17 ± 72.11	738.21 ± 80.75
E_LPT	54.49 ± 2.18	5.48 ± 2.67	4.07 ± 4.23	77.42 ± 61.18	22.73 ± 2.67	896.03±101.95	758.64 ± 96.19
NE_SPT	52.97 ± 3.58	9.18 ± 3.57	0.36 ± 0.98	83.97 ± 41.17	33.32 ± 3.67	879.99 ± 63.22	672.90 ± 65.00
Pre-s.	37.09 ± 6.99	9.39 ± 3.59	0.15 ± 0.67	31.87 ± 37.66	14.21 ± 3.35	708.67 ± 94.95	669.28 ± 96.72
MARL	45.05 ± 11.19	9.14 ± 3.36	0.41 ± 1.27	46.38 ± 34.47	17.91 ± 3.14	849.93 ± 69.79	772.64 ± 74.28
	Emergency Days						
	PT(E)	PT(NE)	Unserved(NE)	Overtime	Delay	Revenue	CR
SPT_U	52.83 ± 1.15	10.98 ± 2.44	1.86 ± 1.79	122.54 ± 41.02	35.81 ± 2.67	945.80 ± 75.29	689.90 ± 65.84
LPT_U	24.95 ± 9.45	10.55 ± 2.15	2.29 ± 1.86	67.02 ± 46.11	21.74 ± 6.12	848.28 ± 85.96	774.33 ± 85.46
NE_LPT	24.00 ± 9.51	12.25 ± 2.57	0.59 ± 1.22	69.20 ± 44.65	20.78 ± 6.02	850.66 ± 80.08	781.13 ± 84.31
E_LPT	54.43 ± 2.42	5.33 ± 2.34	7.51 ± 3.86	134.05 ± 60.54	22.90 ± 2.92	985.13±104.50	837.76±106.11
NE_SPT	49.70 ± 4.35	11.97 ± 2.92	0.87 ± 1.47	92.62 ± 40.92	30.52 ± 3.99	909.32 ± 79.98	723.72 ± 67.46
Pre-s.	31.26 ± 6.58	12.41 ± 2.64	0.43 ± 1.08	59.89 ± 47.15	13.98 ± 3.99	817.24 ± 56.82	776.87 ± 61.82
MARL	33.78 ± 11.05	11.82 ± 2.55	1.02 ± 1.93	68.50 ± 43.05	17.86 ± 4.79	884.95 ± 86.47	814.30 ± 84.41
	Non-emergency Days						
	PT(E)	PT(NE)	Unserved(NE)	Overtime	Delay	Revenue	CR
SPT_U	54.73 ± 0.87	7.64 ± 2.81	0.14 ± 0.45	78.25 ± 39.94	36.40 ± 1.62	864.39 ± 46.12	631.46 ± 40.28
LPT_U	46.79 ± 9.20	7.69 ± 2.78	0.10 ± 0.51	24.21 ± 20.10	23.34 ± 3.44	812.42 ± 62.32	702.71 ± 74.10
NE_LPT	46.25 ± 9.42	7.76 ± 2.91	0.02 ± 0.16	25.11 ± 19.95	21.81 ± 2.91	808.50 ± 62.71	715.30 ± 68.55
E_LPT	54.52 ± 2.04	5.56 ± 2.82	2.23 ± 3.12	47.19 ± 34.05	22.63 ± 2.52	848.48 ± 60.12	716.41 ± 55.29
NE_SPT	54.72 ± 0.87	7.70 ± 2.90	0.09 ± 0.31	79.36 ± 40.55	34.82 ± 2.34	864.33 ± 44.84	645.77 ± 44.00
Pre-s.	40.20 ± 4.91	7.78 ± 2.93	0.00 ± 0.00	16.91 ± 18.61	14.34 ± 2.96	650.72 ± 49.53	611.86 ± 53.21
MARL	51.07 ± 5.11	7.71 ± 2.89	0.08 ± 0.27	34.57 ± 21.51	17.93 ± 1.70	831.23 ± 50.38	750.40 ± 57.11

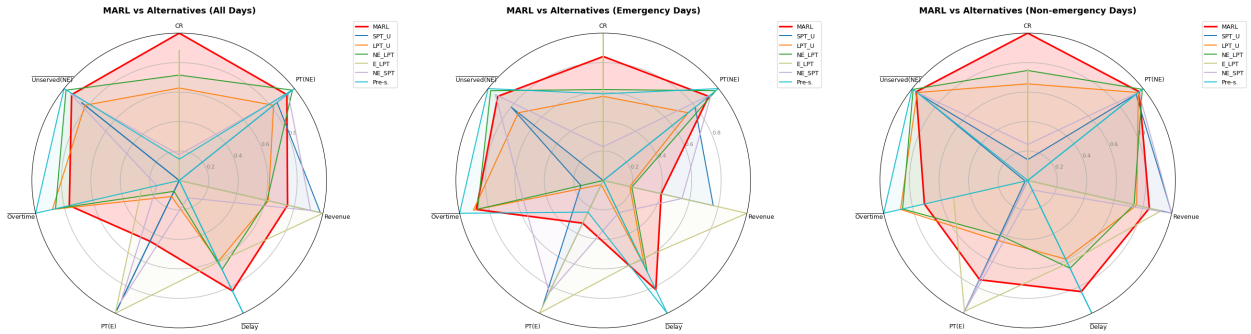
4.9 Policy Interpretability and Behavioral Insights

Beyond aggregate metrics, we visualize representative daily schedules to qualitatively examine how the learned policy behaves under different operational conditions. Figure Figure 2 presents three illustrative Gantt charts corresponding to (a) a regular non-emergency day, (b) a day with an emergency batch arriving late in the schedule ($t = 66$), and (c) a day with an early emergency arrival ($t = 1$).

These visualizations highlight how the MARL policy adapts its sequencing and resource allocation in response to real-time uncertainty.

Batching compatible types. The policy tends to group setup-compatible elective cases—typically shorter procedures-consecutively within each operating room, minimizing setup transition that would otherwise accumulate between heterogeneous case types. This batching behavior indicates that the policy has internalized setup-related penalties and implicitly learned to reduce unnecessary switching.

Length-aware scheduling. Longer or more complex surgeries are typically placed earlier in the day, while shorter ones occupy later slots, forming an "early-long, late-short" temporal structure. This pattern mitigates the risk of overtime by ensuring that long cases finish before the end-of-day horizon, while remaining capacity can flexibly absorb shorter



(a) All Days (b) Emergency Days (c) Non-emergency Days

Figure 1: Radar charts comparing the performance of alternative scheduling methods.

cases.

Adaptive prioritization. The policy internalizes the prioritization rule directly from the reward structure, raising the priority of urgent and emergency arrivals and inserting them adaptively into the remaining schedule. This learned behavior enhances responsiveness and ensures timely treatment of high-value or time-critical patients without causing excessive disruption to elective throughput.

Cross-OR load balancing. Long or resource-intensive cases are staggered across operating rooms rather than concentrated in a few, preventing simultaneous congestion and promoting balanced utilization system-wide. This spatial diversification reduces bottlenecks and improves overall robustness to stochastic variation in case duration.

Taken together, these patterns demonstrate that the MARL policy has learned an operationally coherent strategy. The resulting behaviors—batching similar cases, front-loading longer ones, adaptively prioritizing urgent arrivals, and distributing load across ORs—collectively explain the policy's superior and well-balanced performance across metrics.

4.10 Regret Analysis Relative to the Oracle

To better understand the remaining performance gap, we compare the learned MARL policy with an ex post MIP oracle that possesses complete foresight of all future stochastic information—namely, the realized surgery durations and the arrival times and types of non-elective patients. The oracle corresponds to the optimal schedule derived from the mixed-integer programming (MIP) formulation, solved to a 1.15% optimality gap[5]. It therefore represents an idealized yet unrealizable upper bound: computationally near-optimal, but infeasible for

[5]The reported optimality gap reflects the solver's internal tolerance, meaning the MIP solution is guaranteed to be within 1.15% of the true optimal value of the formulated problem. It measures solver precision, not the difference between the MIP oracle and MARL.

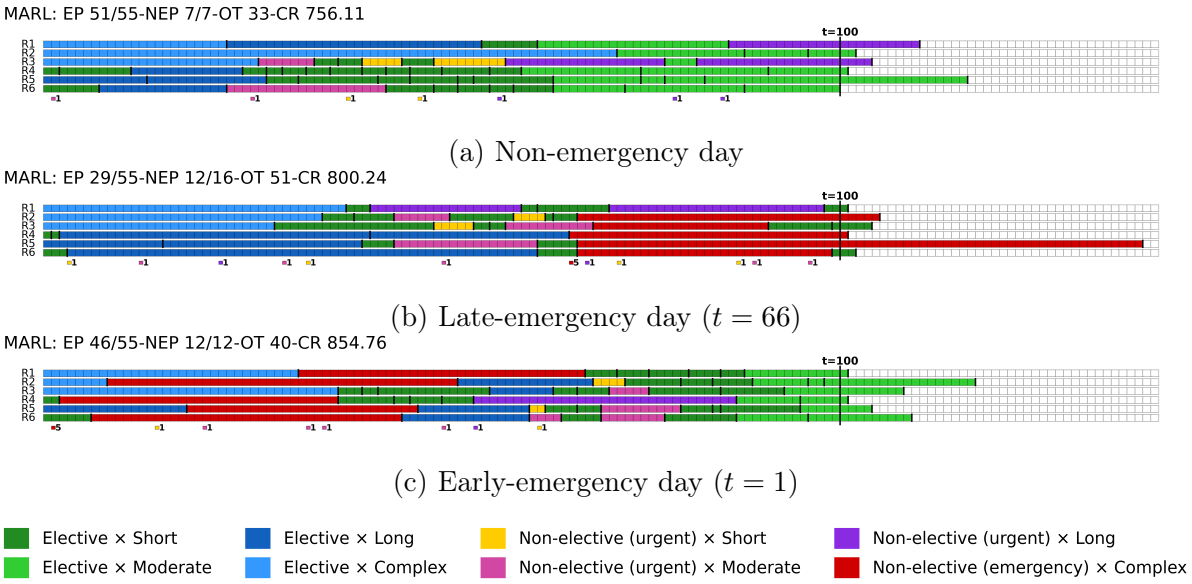
Table 5: Performance comparison of the proposed MARL scheduling framework and rule-based scheduling heuristics under all samples and emergency-case samples

	All Days							
	PT(E)	PT(NE)	Unserved(NE)	Overtime	Delay	Revenue	CR	Average Score
SPT_U	0.98	0.85	0.85	0.00	0.00	0.98	0.00	0.52
LPT_U	0.12	0.82	0.82	0.88	0.61	0.62	0.63	0.64
NE_LPT	0.08	0.98	0.98	0.86	0.67	0.61	0.72	0.70
E_LPT	1.00	0.00	0.00	0.26	0.61	1.00	0.88	0.54
NE_SPT	0.91	0.95	0.95	0.16	0.13	0.91	0.17	0.60
Pre-scheduled based	0.00	1.00	1.00	1.00	1.00	0.00	0.14	0.59
MARL	0.46	0.93	0.93	0.77	0.83	0.75	1.00	0.81
	Emergency Days							
	PT(E)	PT(NE)	Unserved(NE)	Overtime	Delay	Revenue	CR	Average Score
SPT_U	0.95	0.80	0.80	0.16	0.00	0.77	0.00	0.49
LPT_U	0.03	0.74	0.74	0.90	0.64	0.18	0.57	0.54
NE_LPT	0.00	0.98	0.98	0.87	0.69	0.20	0.62	0.62
E_LPT	1.00	0.00	0.00	0.00	0.59	1.00	1.00	0.51
NE_SPT	0.84	0.94	0.94	0.56	0.24	0.55	0.23	0.61
Pre-scheduled based	0.24	1.00	1.00	1.00	1.00	0.00	0.59	0.69
MARL	0.32	0.92	0.92	0.88	0.82	0.40	0.84	0.73
	Non-emergency Days							
	PT(E)	PT(NE)	Unserved(NE)	Overtime	Delay	Revenue	CR	Average Score
SPT_U	1.00	0.94	0.94	0.02	0.00	1.00	0.14	0.58
LPT_U	0.45	0.96	0.96	0.88	0.59	0.76	0.66	0.75
NE_LPT	0.42	0.99	0.99	0.87	0.66	0.74	0.75	0.77
E_LPT	0.99	0.00	0.00	0.52	0.62	0.93	0.75	0.54
NE_SPT	0.99	0.96	0.96	0.00	0.07	0.99	0.24	0.61
Pre-scheduled based	0.00	1.00	1.00	1.00	1.00	0.00	0.00	0.57
MARL	0.75	0.97	0.97	0.72	0.84	0.84	1.00	0.87

real-time deployment.

Figure 3 contrasts representative daily schedules generated by the MARL policy (top) and the oracle (bottom) under identical arrival sequences. With perfect foresight, the oracle "looks ahead" and anticipates the moderate-urgency emergency batch, smartly reserving idle capacity in certain operating rooms while maintaining tight coordination across others. This enables it to accommodate all 55 elective and 7 non-elective patients within the scheduling horizon, achieving full coverage and a cumulative reward (CR) of 872.00. By contrast, the MARL policy—operating without advance knowledge—achieves a slightly lower CR of 756.11, primarily due to fewer elective surgeries completed.

This comparison highlights the strategic foresight advantage of the oracle: it can preemptively adjust to future shocks, something no real-time policy can replicate. Nevertheless, the MARL policy's performance remains close to the oracle benchmark, indicating that CTDE learning captures much of the attainable efficiency despite uncertainty and information constraints.



(a) Non-emergency day

(b) Late-emergency day ($t = 66$)

(c) Early-emergency day ($t = 1$)

Figure 2: Representative Gantt charts illustrating learned scheduling behavior under different operating conditions: (a) a regular non-emergency day, (b) a day with an emergency batch arriving late ($t = 66$), and (c) a day with an early emergency batch ($t = 1$) occurring while surgeries are in progress, constraining the MARL policy and leading to longer delays.

5 Discussion

This section synthesizes managerial implications, limitations, and avenues for future research arising from our study. We also reflect on the theory and on the external validity of our simulation-based evaluation.

Managerial takeaways. First, a *unified reward* (Eq. (2))—combining throughput/value u_k, timeliness via $c_k\omega_i^2$, and overtime $C_o \cdot \mathrm{OT}$—is a practical way to operationalize the multi-objective nature of intraday OR control. In our experiments (Section 4), the learned policy consistently trades off elective delay and overtime while preserving urgent access (Fig. 1). Second, *interpretable behaviors* emerge without hand-crafted rules: the policy batches shorter and similar case types to minimize setup transitions, forms an "early-long, late-short" temporal structure that mitigates overtime risk, dynamically prioritizes urgent or high-p_k arrivals, and staggers resource-intensive cases across operating rooms to prevent congestion and enhance overall system robustness. These patterns align with expert intuition, making the approach more acceptable to clinical stakeholders. Third, the *sequential within-epoch assignment* offers a scalable alternative to exhaustive joint optimization—simple to integrate into a control-desk workflow and fast at runtime—while remaining compatible with centralized training.

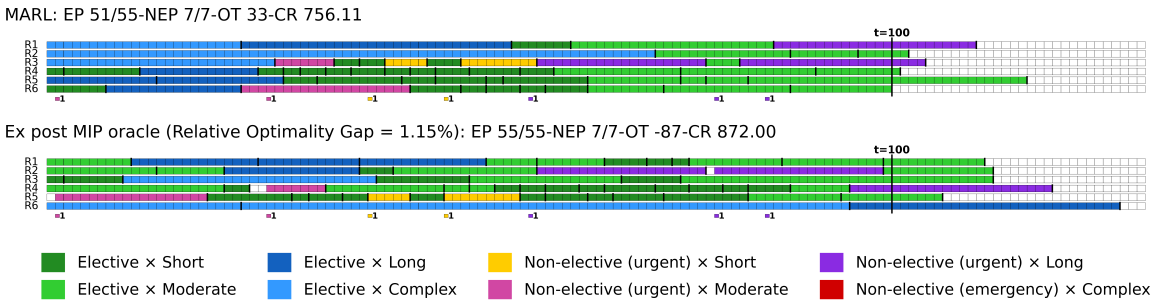


Figure 3: Comparison of daily schedules produced by the MARL (top) and the ex post MIP oracle (bottom) under identical arrival sequences.

What the theory tells us. Our Bellman formulation (Eq. (12)) confirms the problem is a standard finite-horizon MDP; optimal joint planning is, however, computationally prohibitive at realistic scales. Under weak coupling (Lemma 1, assumptions A1–A4), the sequential construction is *equivalent* to joint action selection; outside that regime, the gap is driven by *externalities* across rooms. The regret bounds (Eqs. (15)–(16) and Corollary 4) show that loss relative to a centralized optimum concentrates in two channels: excess waiting of high-penalty patients and terminal overtime. Empirically, these are exactly the components in which heuristics underperform MARL.

Scope and limitations. *(i) Homogeneous ORs and missing staff couplings.* We assume any room can process any type and omit explicit surgeon/anaesthesia/nursing rosters or PACU/ICU capacity. This abstraction matches many day-of-control desks but overlooks cross-resource conflicts, break rules, and shift relief. Section 2 assumes homogeneous operating rooms with identical capabilities. Extending the framework to heterogeneous or coupled resources is a natural direction for future work. *(ii) FIFO within type.* Within-type FIFO simplifies queue discipline; urgency is encoded via $\{u_k, c_k\}$ rather than hard priorities. Hospitals that enforce strict clinical triage can incorporate service-level constraints or minimum-priority rules into the reward or as hard constraints.

(iii) Arrival and duration models. Non-elective demand is exogenous and modeled by Poisson/Bernoulli processes; durations use Gamma/mixtures fitted to empirical moments. These choices are standard and face-valid but imperfect: true processes can be nonstationary (seasonality, special clinics) and exhibit heavier tails or surgeon effects. Deployment should re-calibrate $\{\lambda_k(t)\}$ and $\mathcal{D}_k$ from local data.

(iv) Reward calibration. Weights (u_k, c_k, C_o) bundle clinical value, timeliness, and labor costs (including staff well-being). They should be co-designed with stakeholders and finance to reflect local priorities and compliance targets; different choices can tilt policies toward

access or overtime.

(v) Oracle comparators. We benchmark against ex post oracles solely to quantify regret and headroom; these are *unfair* and not deployable.

(vi) Reproducibility and data access. Due to NDAs, we release a synthetic generator calibrated to empirical moments rather than raw hospital data. We provide full hyperparameters, seeds, and code for the simulator and training in the e-companion to facilitate replication with local data.

Threats to validity. The evaluation covers 500 independent days, but it remains a simulation study. Distribution shift—e.g., calendar effects, surgeon-specific variability, or unplanned staffing outages—can degrade performance. Offline evaluation using retrospective logs and shadow-mode A/B pilots are important intermediate steps before live deployment. Additionally, while PPO with shared actor–critic performed robustly, learning dynamics can be sensitive to network size, entropy regularization, or advantage estimation.

Scalability and compute. The joint action space grows exponentially in J, but our CTDE + sequential protocol scales linearly in the number of available rooms per epoch. Inference is fast (milliseconds per epoch on commodity hardware; Section 4.8); training is offline and can be amortized. Larger suites (e.g., $J > 12$) or finer time discretization primarily increase simulation cost rather than action complexity.

Ethical and operational considerations. Policies that "optimize" overtime may inadvertently shift burdens across staff. Our framework can incorporate explicit *staff well-being* terms (e.g., soft caps on extended shifts) via additional penalties or constraints. Likewise, equity across patient groups (e.g., elective specialities) can be handled via minimum service levels or fairness-aware weights. Interpretability is key for adoption; post-hoc policy explanations should accompany any rollout.

Future work. *(A) Coupled resources and downstream units.* Extend the state to include surgeon/anaesthesia rosters, nursing, and PACU/ICU/bed capacity; consider multi-resource setups and handoffs. *(B) Constrained and risk-sensitive RL.* Enforce service-level constraints (e.g., maximum non-elective wait quantiles), Conditional Value-at-Risk (CVaR) based overtime control, or chance constraints on late finishes. *(C) Stronger theory.* Move beyond weak-coupling equivalence to tighter bounds using submodularity or matroid-intersection structure; analyze convergence of CTDE under sequential action composition. *(D) Generalization and adaptation.* Meta-learning across hospitals, context-conditioned policies, dis-

tributionally robust training to handle nonstationarity, and stochastic disruptions such as last-minute surgery cancellations. *(E) Policy distillation.* Extract compact, human-readable rules (e.g., decision trees) from the learned policy to facilitate audit and governance. *(F) Human-in-the-loop integration.* Embed the policy in an OR dashboard that proposes actions, supports "what-if" simulation, and allows supervisor overrides with learning from corrections.

In sum, MARL with a unified reward, centralized training, and sequential execution offers a pragmatic, data-driven complement to optimization-based control. It delivers balanced performance across competing objectives, scales to realistic suites, and yields interpretable behavior—all prerequisites for impact in intraday OR scheduling. The limitations above are tractable engineering and modeling extensions rather than fundamental barriers, and we view this work as a step toward robust, clinician-aligned decision support at the OR control desk.

6 Conclusion

Intraday operating room scheduling is a high-stakes, multi-objective problem that must balance timely access for urgent patients, elective timeliness, and overtime exposure under substantial uncertainty. We formulated this problem as a Markov game with a *unified reward* (Eq. (2)) that internalizes throughput/value, delay, and overtime into a single, optimizable signal. Building on this foundation, we proposed a CTDE multi-agent reinforcement learning approach that uses a shared actor–critic and a *within-epoch sequential assignment* to produce conflict-free joint actions without enumerating the exponential joint-action space. The pre-day MIP plan provides elective reference times that anchor delay penalties while remaining nonbinding intraday.

On the theory side, we established a dynamic-programming basis (Eq. (12)) and identified conditions under which the sequential construction is equivalent to joint action selection (Lemma 1). Beyond this weak-coupling regime, we derived a bound that decomposes regret into excess waiting and overtime (Eqs. (15)–(16)) and provided a clean gap result in a stylized urgent-arrival case (Corollary 4). These results clarify where performance losses can arise and motivate diagnostics and guardrails.

Empirically, across 500 independent day-episodes, the learned policy consistently delivers the highest unified reward (CR), with robust improvements in urgent access, elective delay, and overtime relative to a suite of transparent heuristics. Oracle analyses—while *unfair* due to ex post information—show modest, interpretable gaps concentrated in the theoretically identified channels. Policy analytics reveal face-valid behaviors (type batching, length-aware scheduling, adaptive prioritization and cross-OR load balancing, supporting managerial ac-

ceptability.

The approach is practical: it scales linearly with the number of available rooms per epoch, runs quickly at inference, and offers interpretable summaries of learned behavior. Limitations—homogeneous ORs, omitted staff couplings, and simplified arrival/duration models—are addressable through extensions discussed in Section 5, including coupled-resource modeling, constrained or risk-sensitive RL, and human-in-the-loop deployment.

Overall, MARL with sequential assignment is a viable, data-driven complement to optimization for real-time OR scheduling. By unifying objectives and learning coordinated policies directly from realistic simulations, it achieves balanced performance difficult to obtain with fixed heuristics, while preserving the transparency and scalability needed for clinical operations. We view this work as a step toward reliable decision support at the OR control desk and a foundation for future advances in coupled-resource scheduling, fairness, and safe learning from real hospital data.

References

S. M. Abate, B. Chekol, B. Basu, R. A. Rios, and M. Gebremariam. Global prevalence and reasons for case cancellation on the intended day of surgery: a systematic review and meta-analysis. *International Journal of Surgery Open*, 26:55–63, 2020.

Mojtaba Arab Momeni, Amirhossein Mostofi, Vipul Jain, and Gunjan Soni. Covid19 epidemic outbreak: operating rooms scheduling, specialty teams timetabling and emergency patients' assignment using the robust optimization approach. *Annals of Operations Research*, pages 1–31, 2022.

Macarena Azar, Rodrigo A. Carrasco, and Susana Mondschein. Dealing with uncertain surgery times in operating room scheduling. *European Journal of Operational Research*, 299(1):377–394, 2022.

Christopher P. Childers and Melinda Maggard-Gibbons. Understanding costs of care in the operating room. *JAMA Surgery*, 153(4):e176233, 2018.

Sangdo Choi and Wilbert E. Wilhelm and. An analysis of sequencing surgeries with durations that follow the lognormal, gamma, or normal distribution. *IIE Transactions on Healthcare Systems Engineering*, 2(2):156–171, 2012.

Brian T. Denton, James Viapiano, and Andrea Vogl. Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health Care Management Science*, 10(1):13–24, 2007.

Brian T. Denton, Andrew J. Miller, Hari J. Balasubramanian, and Todd R. Huschka. Optimal allocation of surgery blocks to operating rooms under uncertainty. *Operations Research*, 58:802–816, 2010.

Michael Fairley, David Scheinker, and Margaret L. Brandeau. Improving the efficiency of the operating room environment with an optimization and machine learning model. *Health Care Management Science*, 22(4):756–767, 2019.

E. Gökalp, N. Gülpınar, and X.V. Doan. Dynamic surgery management under uncertainty. *European Journal of Operational Research*, 309(2):832–844, 2023.

Mehdi Lamiri, Xiaolan Xie, Alexandre Dolgui, and Frédéric Grimaud. A stochastic model for operating room planning with elective and emergency demand for surgery. *European Journal of Operational Research*, 185(3):1026–1037, 2008.

Mohammed Majthoub Almoghrabi and Guillaume Sagnol. Surgery scheduling in flexible operating rooms by using a convex surrogate model of second-stage costs. *European Journal of Operational Research*, 321(1):23–40, 2025.

Daiki Min and Yuehwern Yih. Scheduling elective surgery under uncertainty and downstream capacity constraints. *European Journal of Operational Research*, 206(3):642–652, 2010.

Patrizia Ribino, Claudia Di Napoli, and Luca Serino. A multi-agent rl algorithm for single-day operating room scheduling. *Workshops at the 18th International Conference on Intelligent Environments (IE2022)*, pages 277–287, 2022.

Atle Riise, Carlo Mannino, and Edmund K. Burke. Modelling and solving generalised operational surgery scheduling problems. *Computers & Operations Research*, 66:1–11, 2016.

David H. Rothstein and Mehul V. Raval. Operating room efficiency. *Seminars in Pediatric Surgery*, 27(2):79–85, April 2018.

Hadhemi Saadouli, Badreddine Jerbi, Abdelaziz Dammak, Lotfi Masmoudi, and Abir Bouaziz. A stochastic optimization and simulation approach for scheduling operating rooms and recovery beds in an orthopedic surgery department. *Computers & Industrial Engineering*, 80:72–79, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Karmel S. Shehadeh. Data-driven distributionally robust surgery planning in flexible operating rooms over a Wasserstein ambiguity. *Computers & Operations Research*, 145:105927, 2022.

Karmel S. Shehadeh and Rema Padman. A distributionally robust optimization approach for stochastic elective surgery scheduling with limited intensive care unit capacity. *European Journal of Operational Research*, 290(3):901–913, 2021.

Karmel S. Shehadeh, Man Yiu Tsang, Rema Padman, and Arman Kilic. Operating room-to-downstream elective surgery planning under uncertainty. *European Journal of Operational Research*, 329(1):288–307, 2026.

Thiago A.O. Silva and Mauricio C. de Souza. Surgical scheduling under uncertainty by approximate dynamic programming. *Omega*, 95:102066, 2020.

Jin Wang, Hainan Guo, Monique Bakker, and Kwok-Leung Tsui. An integrated approach for surgery scheduling under uncertainty. *Computers & Industrial Engineering*, 118:1–8, 2018.

Yu Wang, Yu Zhang, Minglong Zhou, and Jiafu Tang. Feature-driven robust surgery scheduling. *Production and Operations Management*, 32(6):1921–1938, 2023.

Lixiang Zhao, Han Zhu, Min Zhang, Jiafu Tang, and Yu Wang and. Large-scale dynamic surgical scheduling under uncertainty by hierarchical reinforcement learning. *International Journal of Production Research*, pages 1–32, 2024.