Multi-Objective Reinforcement Learning for Large-Scale Tote Allocation in Human-Robot Collaborative Fulfillment Centers

Sikata Sengupta [1 *] **Guangyi Liu** [2 *] **Omer Gottesman** [2]
Joseph W Durham [2] **Michael Kearns** [1 2] **Aaron Roth** [1 2] **Michael Caldara** [2]

Abstract

Optimizing the consolidation process in container-based fulfillment centers requires trading off competing objectives such as processing speed, resource usage, and space utilization while adhering to a range of real-world operational constraints. This process involves moving items between containers via a combination of human and robotic workstations to free up space for inbound inventory and increase container utilization. We formulate this problem as a large-scale Multi-Objective Reinforcement Learning (MORL) task with high-dimensional state spaces and dynamic system behavior. Our method builds on recent theoretical advances in solving constrained RL problems via best-response and no-regret dynamics in zero-sum games, enabling principled minimax policy learning. Policy evaluation on realistic warehouse simulations shows that our approach effectively trades off objectives, and we empirically observe that it learns a single policy that simultaneously satisfies all constraints, even if this is not theoretically guaranteed. We further introduce a theoretical framework to handle the problem of error cancellation, where time-averaged solutions display oscillatory behavior. This method returns a single iterate whose Lagrangian value is close to the minimax value of the game. These results demonstrate the promise of MORL in solving complex, high-impact decision-making problems in large-scale industrial systems.

[*]Equal contribution [1]Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA [2]Amazon. Correspondence to: Sikata Sengupta <sikata@seas.upenn.edu>.

ICML 2026 Workshop: RLxF: Reinforcement Learning from World Feedback

1. Introduction and Background

Modern warehouses (Figure 1a) increasingly rely on human-robot collaboration to manage large-scale inventory operations. For example, Amazon's robotic fulfillment system, "Sequoia," leverages these collaborations to store inventory 75% faster and process customer orders 25% faster than its prior generation fulfillment system through the integration of multiple robotic products used to containerize inventory and present it at an ergonomic height for processing at an employee workstation. Key to unlocking these speed-ups is efficient resource management within the storage system. In this work, we focus on a particular space management process called "consolidation," wherein items are moved between containers to improve utilization (Figure 1b) and free up space for new inbound items. Items are stored in containers, known as "totes", which are cyclically inbounded into storage shelves, consolidated to reclaim space, and picked from to fulfill customer orders. Efficient tote selection for consolidation is thus central to optimizing both storage utilization and throughput in this environment, but requires careful management of multiple operational objectives.

For illustration purposes, we assume the following stylized consolidation process as totes can be processed by both humans and robots, each with different manipulation capabilities. We assume that humans are capable of safely manipulating any item, whereas robots are only capable of safely manipulating some items. The task of both operators is to transfer items from a "source" tote to one or more "destination" totes. Source totes are ejected when emptied, and destination totes are ejected when full. A range of factors influence whether a tote is better suited for a human or robot to consolidate including the particular items in the source tote, the destination tote fullness, and the tote properties (e.g., we assume two types of totes "smaller" and "larger" of differing dimensions). Employee workstations typically handle a broader range of item types, but use up processing capacity from other workflow, whereas robotic stations excel at consistency but may be restricted by item properties. As such, consolidation decisions must not only maximize throughput and space utilization, but also *intelligently allocate workloads between humans and robots* to leverage their

respective strengths. Operational goals for consolidation are similarly multi-faceted and often conflicting. Key performance indicators (KPIs) include throughput efficiency, inventory tote type balance, and space utilization. Optimizing across these dimensions requires managing complex trade offs under dynamic conditions and real-world constraints.

Due to the complexity of this decision, we expect that heuristics and single-objective optimization strategies will fail to generalize or scale beyond some small number of trade offs. A common approach to handle such scenarios is *scalarization*, where the different objectives are combined into a single objective function using predefined weights. However, in one-shot settings, these weights must be fixed in advance, making the approach sensitive to weight selection and ill-suited for environments with shifting priorities. Moreover, they typically optimize for one KPI at the expense of others, resulting in suboptimal system-wide performance. In contrast, our work avoids the need for manual weight specification by employing an adaptive procedure that iteratively adjusts the objective weights during training. This enables the learning process to naturally converge toward a minimax solution that balances objectives under dynamic and uncertain conditions, as we elaborate later in the paper.

In this work, we address this problem by framing the tote consolidation and allocation problem as a large-scale *Multi-Objective Reinforcement Learning* (MORL) task. Our formulation captures the dynamic and stochastic nature of warehouse operations while explicitly modeling the trade offs between conflicting KPIs and the heterogeneous capabilities of human and robotic stations. Building on recent theoretical advances in constrained RL, particularly best-response and no-regret learning in zero-sum games, we develop a principled and scalable approach for minimax policy learning in complex multi-objective settings.

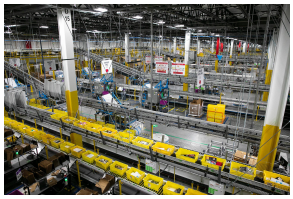
Our contributions are threefold: (i) we propose a novel MORL formulation for real-world consolidation problems in human-robot collaborative fulfillment centers, explicitly modeling heterogeneous station capabilities; (ii) we develop a theoretical framework that reformulates the multi-objective problem as a zero-sum Lagrangian game and prove that we can select a single iterate from the time-averaged approximate minimax mixture whose Lagrangian value is close to the minimax value of the game; and (iii) we demonstrate strong empirical performance on realistic warehouse simulations, where our approach outperforms baselines across KPIs. These results provide compelling evidence that MORL is a viable and impactful approach to optimizing high-dimensional, high-stakes industrial decision making systems.

2. Related Work

We organize our related work mainly into two categories– theoretical works related to multi-objective or constrained RL, and empirical works related to this domain of large-scale industrial applications like inventory management.

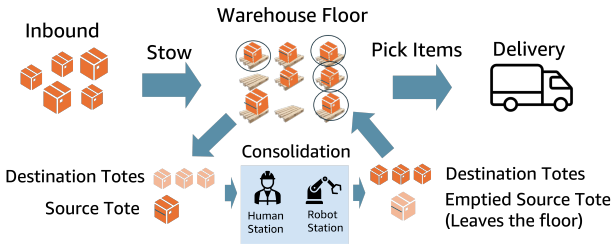
Hayes et al. (2022) provides a comprehensive overview of general multi-objective RL approaches that have been explored. As noted in (Yang et al., 2019), MORL methods can often be divided into two main categories: single-policy and multiple-policy methods. Single policy methods often require knowledge of preferences in advance, so in this paper we take a multiple policy approach. Constrained RL problems can be modeled as a multi-objective problem with constraint violation terms written as additional objectives. Constrained RL is of interest in many domains ranging from safety to fairness, where one wants to learn a policy that optimizes some global reward subject to various trustworthiness constraints. (Calvo-Fullana et al., 2024) utilize primal-dual methods (via optimizing single objective RL for the learner) and utilizing dual descent for the regulator. Especially for zero-sum games, one can relate game-theoretic notions of best-response and no-regret to primal-dual style algorithms. (Miryoosefi et al., 2019) use best-response vs. no-regret dynamics with convex constraints and provide safety guarantees on the time-averaged strategies of each player. (Eaton et al., 2025) provide oracle-efficient algorithms similarly utilizing best-response vs. no-regret dynamics to solve for a minimax solution of the Lagrangian of this problem even when the number of objectives (groups in the context of fairness) is exponentially large for both tabular and large state space MDPs. (Park et al., 2024) and (Byeon et al., 2025) also adopt a minimax perspective on multi-objective RL–they also formulate a repeated game between a learner and regulator to solve a slightly different, but related regularized objective. We draw on ideas from all of these papers to study best-response vs. no-regret dynamics, and build upon it to provide a method for extracting a single deployable iterate rather than relying on a mixture policy, and target scalability to large action spaces in industrial settings. We choose to utilize this repeated game framework because it does not require us to know preferences over objectives in advance and because it does not make strong assumptions on the structure/geometry of multi-objective problem.

On the application side, (El Shar et al., 2023) consider a Multi-Objective RL approach to sustainable supply-chain optimization. They learn policies to try to approximate the Pareto Frontier. They make use of an outer-inner loop procedure where the outer loop makes use of evolutionary strategy updates and the inner loop uses Envelope Multi-Objective Q-Learning (Yang et al., 2019).



(a) Container-based fulfillment center.

(b) Robotic consolidation station.



(c) Human-robot collaborative fulfillment center workflow.

Figure 1. Overview of the fulfillment center environment. (a) A large-scale facility with dense human-robot collaboration. (b) A robotic consolidation station transferring items between totes. (c) Schematic workflow of the consolidation process.

3. Problem Setting and Model

We study the decision-making problem underlying tote consolidation in human-robot collaborative fulfillment centers, illustrated in Figure 1c. Consolidation operates alongside inbound and pick processes and involves transferring items from partially filled source totes into destination totes to reclaim storage capacity. Employee and robot stations differ in their manipulation capabilities: employee stations can handle complex or irregular items, while robot stations are limited in their reach, perception, and handling of deformable and reflective products. Effective consolidation therefore requires allocating work across heterogeneous stations while respecting operational constraints.

At each decision point, the system selects a source tote and a compatible destination tote based on tote-level attributes such as fill level and item complexity, as well as floor-level signals, such as the current distribution of tote types. The objective is to balance multiple, often competing, KPIs, including maintaining high overall processing efficiency, making effective use of available empty capacity during consolidation, and preserving a healthy distribution of tote types due to the warehouse configuration. We model this as a multi-objective sequential decision-making problem and address it using constrained multi-objective reinforcement learning (MORL), which enables the learning of station-aware tote prioritization policies that adapt to dynamic warehouse conditions.

3.1. Preliminaries

We consider an episodic RL setting modeled as a Markov Decision Process (MDP) $\mathcal{M} = \left(\mathcal{X}, \mathcal{A}, \{P_h\}_{h=0}^{H-1}, \{r_{i,h}\}_{i=0,\,h=0}^{m,\,H-1}, \mu, H\right)$, where $\mathcal{X}$ is the state space, $\mathcal{A}$ is the action space, $P_h : \mathcal{X} \times \mathcal{A} \to \Delta(\mathcal{X})$ is the transition function at step h, $r_{i,h} : \mathcal{X} \times \mathcal{A} \to [0, R]$ is the reward function for objective $i \in \{0, \ldots, m\}$ at step h, μ is the initial-state distribution, and H is the episode horizon. Here, $\Delta(\mathcal{X})$ denotes the probability simplex over $\mathcal{X}$. A non-stationary policy is a sequence $\pi = \{\pi_h\}_{h=0}^{H-1}$, where each $\pi_h : \mathcal{X} \to \Delta(\mathcal{A})$ maps states to distributions over actions. We use $[H] := \{0, \ldots, H-1\}$ to denote the set of time indices. Given a policy π, the trajectory $(\mathbf{x}_0, \mathbf{a}_0, \ldots, \mathbf{x}_{H-1}, \mathbf{a}_{H-1})$ is generated by $\mathbf{x}_0 \sim \mu$, $\mathbf{a}_h \sim \pi_h(\mathbf{x}_h)$, and $\mathbf{x}_{h+1} \sim P_h(\cdot \mid \mathbf{x}_h, \mathbf{a}_h)$ for all $h \in [H-1]$. For objective i, the value function under policy π is

$$V_{i,h}^{\pi}(x) \;=\; \mathbb{E}^{\pi}\left[\sum_{t=h}^{H-1} r_{i,t}(\mathbf{x}_t, \pi_t(\mathbf{x}_t)) \,\middle|\, \mathbf{x}_h = x\right].$$

We cast our setting as a constrained RL problem (Paternain et al., 2019; Eaton et al., 2025) of the form

$$\begin{aligned} \max_{D \in \Delta(\Pi)} \quad & \mathbb{E}_{\pi \sim D}\left[\mathbb{E}_{\mathbf{x} \sim \mu}\left[V_{0,0}^{\pi}(\mathbf{x})\right]\right] \\ \text{s.t.} \quad & \mathbb{E}_{\pi \sim D}\left[\mathbb{E}_{\mathbf{x} \sim \mu}\left[V_{i,0}^{\pi}(\mathbf{x})\right]\right] \le \alpha_i, \end{aligned} \tag{1}$$

for all $i \in \{1, \ldots, m\}$, where Π is the policy class and $D \in \Delta(\Pi)$ denotes a distribution over policies. Here, $V_{i,0}^{\pi}(\mathbf{x})$ is the value function at the initial time step for objective i, with $i = 0$ denoting the primary objective and $i \ge 1$ corresponding to constraint objectives.

To solve the constrained optimization problem (1), we introduce its Lagrangian:

$$\begin{aligned} \mathcal{L}(D, \lambda) = \mathbb{E}_{\pi \sim D}\left[\mathbb{E}_{\mathbf{x} \sim \mu}\left[V_{0,0}^{\pi}(\mathbf{x})\right]\right] \\ + \sum_{i=1}^{m} \lambda_i \left(\alpha_i - \mathbb{E}_{\pi \sim D}\left[\mathbb{E}_{\mathbf{x} \sim \mu}\left[V_{i,0}^{\pi}(\mathbf{x})\right]\right]\right), \end{aligned} \tag{2}$$

where $\lambda \in \mathbb{R}_{\ge 0}^{m}$ are the Lagrange multipliers. This formulation can be interpreted as a zero-sum game between: (i) a *Learner* selecting a distribution over policies $D \in \Delta(\Pi)$ to maximize $\mathcal{L}$, and (ii) a *Regulator* selecting nonnegative multipliers to minimize $\mathcal{L}$. The regulator's strategy space is the compact set $\Lambda = \left\{\lambda \in \mathbb{R}_{\ge 0}^{m} : \|\lambda\|_1 \le C\right\}$. Under these conditions, by (Sion, 1958)'s minimax theorem,

$$L^* := \min_{\lambda \in \Lambda} \max_{D \in \Delta(\Pi)} \mathcal{L}(D, \lambda) = \max_{D \in \Delta(\Pi)} \min_{\lambda \in \Lambda} \mathcal{L}(D, \lambda).$$

Note that because the learner's objective is linear in D, a best-response to any fixed λ_t is attained at an extreme point of $\Delta(\Pi)$, and thus each D_t corresponds to a single policy. We can define an approximate minimax equilibrium as follows.

Definition 1 (ν-approximate Minimax Equilibrium)**.** $(\hat{D}, \hat{\lambda})$ is a ν-approximate minimax equilibrium if:

$$\max_{D \in \Delta(\Pi)} \mathcal{L}(D, \hat{\lambda}) - \nu \leq \mathcal{L}(\hat{D}, \hat{\lambda}) \leq \min_{\lambda \in \Lambda} \mathcal{L}(\hat{D}, \lambda) + \nu.$$

To compute an approximate solution (as is shown in Definition 1) to the minimax problem $\min_{\lambda \in \Lambda} \max_{D \in \Delta(\Pi)} \mathcal{L}(D, \lambda)$ in practice, we adopt a repeated game perspective. In this view, the learner and the regulator iteratively update their strategies over T rounds, and the average strategies converge to an approximate equilibrium. Our approach builds on the seminal best-response vs. no-regret result of (Freund & Schapire, 1996):

Theorem 3.1 ((Informal) Best-Response vs. No-Regret (Freund & Schapire, 1996))**.** *Let $(D_1, \ldots, D_T)$ be the sequence of best-response (distributions over) policies and let $(\lambda_1, \ldots, \lambda_T)$ be the corresponding sequence of Lagrangian weights maintained by the learner and regulator respectively. Then $(\bar{D}, \bar{\lambda})$ form an approximate minimax equilibrium of the game defined by $\mathcal{L}$, where $\bar{D} = \frac{1}{T}\sum_{t=1}^{T} D_t$ and $\bar{\lambda} = \frac{1}{T}\sum_{t=1}^{T} \lambda_t$.*

We next quantify the learner's ability to produce an approximate best-response policy, following (Eaton et al., 2025):

Definition 2 (ϵ-approximate best-response)**.** A distribution over policies D^* is an ϵ-best-response if for a given $\lambda \in \Lambda$,

$$\mathcal{L}(D^*, \lambda) \geq \max_{D \in \Delta(\Pi)} \mathcal{L}(D, \lambda) - \epsilon.$$

Given a fixed $\lambda \in \Lambda$ from the regulator, the learner's best-response problem can be expressed as optimizing a scalarized reward function (Eaton et al., 2025):

$$r_\lambda(x, a) = r_0(x, a) + \sum_{i=1}^{m} \lambda_i \left(\frac{\alpha_i}{H} - r_i(x, a) \right).$$

Thus, in practice, the best-response reduces to solving a single-objective RL problem parameterized by λ. In this paper, we implement this step using Deep Q-Learning (DQN) (Mnih et al., 2015), although many other deep RL algorithms (Mnih et al., 2013; Schulman et al., 2017; Haarnoja et al., 2018) could be used in its place.

For the regulator, we define its regret with respect to Λ as follows:

$$\mathrm{Reg}(\lambda_{1:T}) = \sum_{t=1}^{T} \mathcal{L}(D_t, \lambda_t) - \min_{\lambda \in \Lambda} \sum_{t=1}^{T} \mathcal{L}(D_t, \lambda). \quad (3)$$

If the regret of an algorithm is sublinear in T, equivalently, if the average regret tends to 0 as $T \to \infty$, we refer to it as a *no-regret* algorithm throughout this paper. One concrete example is Online Gradient Descent (OGD):

Theorem 3.2 (Online Gradient Descent No-Regret (Zinkevich, 2003))**.** *For a sequence of differentiable functions $f^1, \ldots, f^T : S \to \mathbb{R}$ such that $\|\nabla f^t(x)\| \leq G$ for any $1 \leq t \leq T, x \in S$, i.e., G is an upper bound on the gradient magnitudes. Then, for the following sequence $x^1, \ldots, x^T$ built according to*

$$x^{t+1} = \Pi_S(x^t - \eta \nabla f^t(x^t)),$$

where $\eta = \frac{D}{G\sqrt{T}}$ and D is an upper bound on the diameter of the closed convex set S, the regret is bounded as:

$$\sum_{t=1}^{T} f^t(x^t) \leq \min_{x \in S} \left(\sum_{t=1}^{T} f^t(x) \right) + DG\sqrt{T}.$$

It is important to note that the guarantee in Theorem 3.1 applies to the time-averaged solution $\bar{D}$ in expectation. Individual policies drawn from the support of $\bar{D}$ may violate constraints, as long as such violations are offset on average across the mixture, closeness of the mixture to the minimax value does not imply that any single policy satisfies all constraints. To illustrate, consider a simple MDP with state space $\mathcal{X} = X_{\text{safe}} \cup X_{\text{left}} \cup X_{\text{right}}$, where deviations to either side correspond to entering a danger region. A mixture policy could alternate between left and right, consistently violating constraints yet appearing feasible in expectation due to cancellation. In Appendix D, we build on the framework of Eaton et al. (2025) to reformulate the problem and show that, when the mixture is near-minimax, it is nevertheless possible to probabilistically extract a single iterate whose Lagrangian value is close to the minimax value.

3.2. MDP Model of Warehouse Floor

We begin by defining the state space, followed by the action space, transition dynamics, and reward structure.

The warehouse floor F has capacity for `floor_max` totes, which we assume to be large. Each state $x \in \mathcal{X}$ corresponds to a specific tote slot, which may be occupied or empty, together with aggregate information describing the global state of the floor. We represent x as a feature vector comprising the following task relevant quantities:

$$x_t = \Big(N_{\texttt{large}}, \texttt{ETPH}, L_S^H, L_D^H, L_S^R, L_D^R, O_k, \texttt{LTE}, n_{\texttt{item}}, n_{\texttt{pick}}, \texttt{GCU}, t \Big) \quad (4)$$

We describe each component below:

- $N_{\texttt{large}} \in \mathbb{Z}_+$: Total number of larger totes on the floor.
- $\texttt{ETPH} \in \mathbb{R}_+$: Empty Totes Per Hour, which measures the

number of source totes emptied via consolidation per hour.
• $L_S^H, L_D^H, L_S^R, L_D^R \in \mathbb{Z}_+$: Queue lengths of source (S) and destination (D) totes at human (H) and robotic (R) stations.
• O_k: Occupancy of the tote slot; 0 if empty, 1 if occupied by a larger tote, 2 if occupied by a smaller tote.
• $n_{\texttt{item}} \in \mathbb{Z}_+$: Number of items in the tote.
• $\texttt{LTE} \in (0, 1]$: Likelihood to be successfully emptied by robot stations, approximated by $1/n_{\texttt{item}}$.
• $n_{\texttt{pick}} \in \mathbb{Z}_+$: Number of items in the tote scheduled to be picked that day.
• $\texttt{GCU} \in \mathbb{R}_+$: Gross Cubic Utilization, approximated as the estimated rectangular volume of all items in the tote.
• $t \in [0, H]$: Current time step in the episode.

The action space is given by

$$\mathcal{A} = \left\{ \begin{array}{c} \texttt{ignore} \\ \texttt{not\_ignore} \end{array} \right\} \times \left\{ \begin{array}{c} \texttt{source} \\ \texttt{destination} \end{array} \right\} \times \left\{ \begin{array}{c} \texttt{human} \\ \texttt{robot} \end{array} \right\},$$

which encodes the decision for a given tote slot, to either skip it or assign the tote it contains to a human or robotic station, and to designate it as either a source or destination tote.

State transitions are determined by the environment's update rules for floor and station states. Specifically: (i) the number of larger totes decreases when one is sent as a source tote to a station and increases when new larger totes are stored on the floor; (ii) ETPH evolves according to the aggregate outcomes of consolidation operations across all stations, depending on the assigned station type, tote composition, and other operational factors; (iii) station queue lengths are updated based on the action's station and role assignment. After each action, the next floor slot is selected via linear traversal of the remaining slots. At the end of each day, the stow and pick processes (by which new totes are added to the floor and items from totes are removed to fulfill order demands) are simulated and we shuffle the tote orderings.

The reward functions $\{r_i\}_{i=0}^m$ correspond to the key performance indicators (KPIs) of interest, such as ETPH, $N_{\texttt{large}}$, Source to Destination tote (S/D) ratio, and consolidation station capacity compliance. Some rewards directly capture operational constraints specified in our optimization problem. The primary objective reward is

$$r_0(x_t, a_t, x_{t+1}) = x_{t+1}[\texttt{ETPH}], \tag{5}$$

which captures the throughput efficiency achieved by the current consolidation decision. Since in our setting, the state includes some floor-wide summary statistics , we use ETPH as both part of the state space and the reward signal r_0 we are optimizing over. We use the shorthand $x_t[\texttt{ETPH}]$ to corresponding to the value of that floor-wide statistic at time t in state x_t. The first constraint-based reward is

$$r_1(x_t, a_t, x_{t+1}) = \frac{1}{H}\left(\alpha_1 - \frac{x_{t+1}[N_{\texttt{large}}]}{\texttt{floor\_max}}\right), \tag{6}$$

which penalizes deviations from a desired fraction of larger totes on the floor, reflecting structural constraints imposed by the fulfillment center layout. The second constraint-based reward is

$$r_2(x_t, a_t, x_{t+1}) = \tfrac{1}{H}\left(-\alpha_2 + \tfrac{x_{t+1}[L_S^H] + x_{t+1}[L_S^R]}{1 + x_{t+1}[L_D^H] + x_{t+1}[L_D^R]}\right) \tag{7}$$

which encourages a balanced S/D assignment across stations. By limiting excessive usage of destination totes, this constraint reduces downstream swapping and improves overall consolidation efficiency. The third and fourth constraint-based rewards are:

$$r_3(x_t, a_t, x_{t+1}) = \frac{1}{H}\left(\alpha_3 - \left(x_{t+1}[L_S^H] + x_{t+1}[L_D^H]\right)\right), \tag{8}$$

$$r_4(x_t, a_t, x_{t+1}) = \frac{1}{H}\left(\alpha_4 - \left(x_{t+1}[L_S^R] + x_{t+1}[L_D^R]\right)\right), \tag{9}$$

which enforce capacity constraints at human and robotic stations by penalizing excessive queue lengths, thereby preventing overload and ensuring stable operation.

We aim to maintain the Markov property in our MDP formulation. However, in practice, certain modeling simplifications, which introduced to reduce the dimensionality of the state and action spaces, may introduce mild deviations. We formulate the following constrained RL problem based upon the MDP and problem described above.

$$\begin{aligned} \max_{D \sim \Delta(\Pi)} \ & \mathbb{E}_{\pi \sim D}\Big[\sum_{t=0}^{H-1} \texttt{ETPH}_t(\pi)\Big] \\ \text{s.t. } & \mathbb{E}_{\pi \sim D}\Big[\sum_{t=0}^{H-1} N_{\texttt{large},t}(\pi)\Big] \le \alpha_1, \\ & \mathbb{E}_{\pi \sim D}\Big[-\sum_{t=0}^{H-1} (S/D)_t(\pi)\Big] \le -\alpha_2, \\ & \mathbb{E}_{\pi \sim D}\Big[\sum_{t=0}^{H-1} \left(L_{S,t}^H(\pi) + L_{D,t}^H(\pi)\right)\Big] \le \alpha_3, \\ & \mathbb{E}_{\pi \sim D}\Big[\sum_{t=0}^{H-1} \left(L_{S,t}^R(\pi) + L_{D,t}^R(\pi)\right)\Big] \le \alpha_4. \end{aligned} \tag{10}$$

We can label the bounds in (10), for example, as $\alpha_1 = L$, $\alpha_2 = SD$, $\alpha_3 = B_H$, and $\alpha_4 = B_R$, where L stands for an upper bound on the number of large totes on the floor, SD stands for a lower bound on the source to destination ratio, and B_H, B_R stand for the human and robotic budget capacities. Let $f(\pi) \in \mathbb{R}^m$ denote the vector of constraint

values corresponding to policy π, and let $\alpha \in \mathbb{R}^m$ be the corresponding bound vector, adjusted for the constraint signs. The Lagrangian can then be expressed as

$$\mathcal{L}(D, \lambda) \;=\; \mathbb{E}_{\pi \sim D}\big[\texttt{ETPH}(\pi) + \sum_{i=1}^{m} \lambda_i \big(\alpha_i - f_i(\pi)\big)\big].$$

Remark 3.3 (Constrained RL vs. Multi-Objective RL). The tote-allocation problem is inherently both multi-objective and constrained: ETPH and S/D are system-level objectives, while $N_{\texttt{large}}$ balance and station capacities are operational constraints. We formulate it as a constrained RL problem for clarity, but the two perspectives are closely related via Lagrangian relaxation. In both cases the primal player solves the same scalarized RL problem; the distinction lies in the regulator's role—whether λ drives KPIs toward specified targets (constrained view) or finds robustness to worst-case weightings (multi-objective view). Since we solve a minimax problem in either case, our framework handles both formulations without modification.

3.3. MORL via Best-Response vs. No-Regret Dynamics

To solve the constrained optimization problem in (10), we adopt a repeated game perspective between two players: a *learner* and a *regulator*. This framework follows the best-response versus no-regret paradigm used in prior work on constrained and multi-objective reinforcement learning (Eaton et al., 2025; Calvo-Fullana et al., 2024; Miryoosefi et al., 2019). At each round, the regulator specifies a vector of Lagrange multipliers λ_t, which defines a scalarized reward function. The learner then computes an approximate best-response policy to this reward. The resulting policy is evaluated to estimate constraint satisfaction, and this feedback is used to update the regulator's multipliers via a no-regret procedure (here we use Online Gradient Descent).

This interaction is repeated over multiple rounds, and the time-averaged strategies of the learner and regulator are returned. Under standard no-regret guarantees, this procedure converges to an approximate minimax equilibrium of the underlying Lagrangian game, yielding policies that balance throughput maximization with constraint enforcement. The full pseudocode is given in Algorithm 1 (Appendix A).

4. Experiments

We evaluate our approach using an event-driven simulator developed for this work to capture the key operational dynamics of a large-scale human-robot collaborative fulfillment center. The simulator is calibrated with real operational data so that simulated dynamics reflect realistic operating conditions. The simulator models tote consolidation, station-level capacity constraints, heterogeneous human and robotic capabilities, and evolving floor-level state variables, while abstracting away low-level physical details. This design allows us to isolate the decision-making problem of tote selection and assignment, stress test constraint satisfaction under realistic load conditions, and conduct controlled comparisons across algorithms at scale.

4.1. Single-Objective Optimization

To validate the learning capability of the base RL algorithm, we first evaluate DQN in a single-objective setting using the heuristic ETPH reward. More details on the architecture can be found in Appendix B. Each simulated day consists of $60 \times 24 = 1440$ decision steps, and a single episode spans $N_{\text{days}} = 10$. The agent is trained for 30 episodes per run, corresponding to one learner update round in the repeated game procedure. As shown in Figure 4 (Appendix C), the learned policy exhibits steady performance improvement, with episodic returns increasing substantially over the course of training.

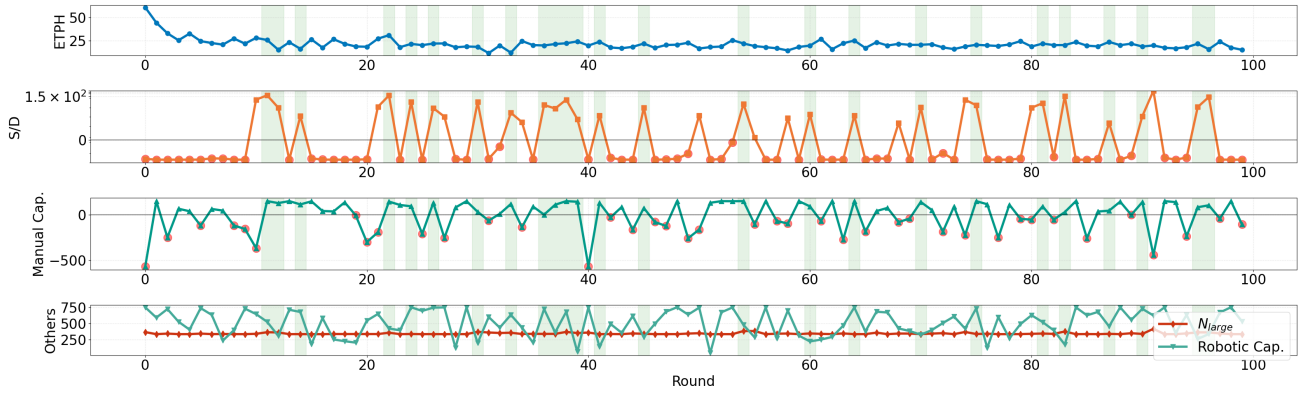
4.2. Multi-Objective Optimization with MORL

As shown in Figure 2a and Figure 5, the regulator steers the learner through the Lagrange multipliers λ_t over repeated rounds, with $C = 20{,}000$. In all constraint plots, values ≥ 0 indicate satisfaction and negative values indicate violation. The $N_{\texttt{large}}$ and robot capacity constraints are largely inactive (multipliers near zero), while enforcing the manual capacity and S/D constraints reduces ETPH, reflecting the inherent throughput–constraint trade-off. The oscillations in λ_t track transitions between satisfaction and violation. Although feasibility is only guaranteed for the time-averaged policy distribution, we empirically observe rounds in which a single policy satisfies all constraints.

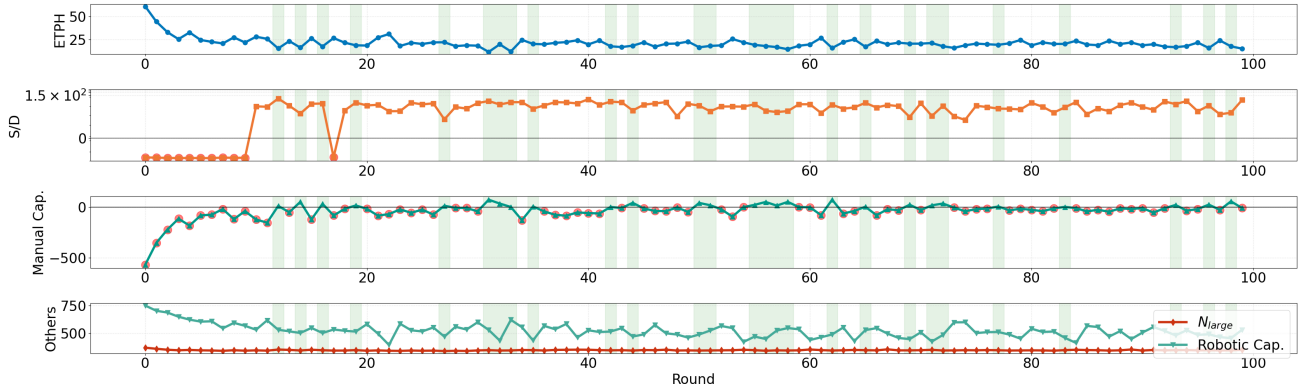
While individual policies may oscillate, the time-averaged (mixture) policies converge toward satisfying all constraints, as shown in Figure 2b and Figure 6. This is consistent with the minimax guarantees in Section 3: the averaged distribution trades some ETPH for improved S/D and manual capacity satisfaction, and Figure 3a confirms that performance under $\bar{\lambda}$ improves as constraint satisfaction stabilizes. If exact feasibility is required, conservative slack in the constraint thresholds can further increase the probability of satisfying the true desired constraints.

4.3. Feasible Stationary Policy via MORL

As noted above, feasibility is theoretically guaranteed only for the time-averaged mixture. However, Figure 3b shows that feasible single policies arise consistently across 20 random seeds, with the cumulative count growing steadily over training rounds even in the worst case. This suggests that the repeated game dynamics naturally guide the learner through

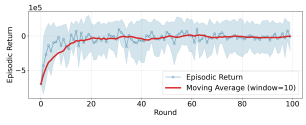


(a) Performance of individual policies at each round.

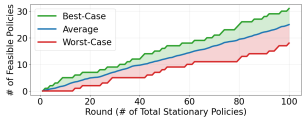


(b) Performance of time averaged policies

Figure 2. Policy performance over repeated game rounds. Each plot shows the global ETPH objective and four constraint values. Black horizontal lines denote constraint thresholds, red markers indicate violations, and green vertical regions indicate rounds where all constraints are simultaneously satisfied.



(a) Average episodic return.



(b) Feasible stationary policies.

Figure 3. (a) Time-averaged policy performance improves as constraint satisfaction stabilizes. (b) Cumulative count of feasible single policies across training rounds.

feasible regions of the policy space, making deployable single policies readily available in practice without requiring mixture execution. Section D provides theoretical support for this observation by showing that a single iterate with near-minimax Lagrangian value can be extracted from the sequence.

To assess the quality of such policies, Table 1 compares a representative feasible policy learned via MORL against several baselines: a rule-based heuristic that routes totes with few items as sources and totes with sufficient empty space as destinations, assigning stations based on tote-level physical features; a fixed-penalty CMDP using pre-selected static penalty vectors; a fixed-weight MORL baseline using static scalarization weights; a primal-dual CMDP inspired by RCPO (Tessler et al., 2019) that updates policy and dual variables online; as well as random actions and an unconstrained policy. Among methods that satisfy all constraints (heuristic, primal-dual CMDP, and ours), our approach achieves substantially higher ETPH (20.52 vs. 9.05 and 3.20), a 2.3× improvement over the best feasible baseline. The fixed-penalty CMDP and fixed-weight MORL baselines both violate the manual capacity constraint, illustrating the difficulty of choosing static weights a priori, while the unconstrained policy attains the highest ETPH at the cost of severe capacity violations. Appendix D provides theoretical support by showing that a single iterate with near-minimax Lagrangian value can be extracted from the learned sequence. In practice, policy feasibility can be assessed prior to full-scale use through controlled pilot testing against the operational KPI and constraint metrics.

Table 1. Comparison of ETPH and constraint satisfaction across methods. Negative slack values (in **red**) indicate constraint violations.

	MORL (Ours)	Heuristic	Fixed-Penalty CMDP	Fixed-Weight MORL	Primal-Dual CMDP	Random Actions	Unconstrained
Satisfies all constraints?	√	√	×	×	√	×	×
ETPH	**20.52**	9.05	7.53	0.00	3.20	9.19	61.81
$N_{\texttt{large}}$ Slack	362.62	362.27	354.76	413.89	340.11	359.15	370.62
S/D Slack	10.81	0.028	14.99	63.46	1.00	0.49	**-0.37**
Manual Cap. Slack	83.21	73.88	**-216.51**	**-495.20**	45.87	**-32.99**	**-563.23**
Robot Cap. Slack	258.23	692.32	732.93	703.94	722.79	571.01	743.68

5. Conclusion

We have explored a Multi-Objective RL framework for large-scale tote allocation and consolidation under realistic operational constraints. Our approach leverages theoretical results on best-response versus no-regret dynamics in zero-sum games, adapting them to the RL setting and introducing efficient implementation strategies to bridge the gap between theory and practice. Using a simulator that captures the core dynamics of fulfillment center operations, we demonstrate that our method is able to learn policies that effectively balance trade offs of KPIs.

This work opens several promising directions for future research based upon improving upon the limitations of this current approach. From a modeling perspective, exploring alternative MDP formulations and state/action space abstractions could yield further gains in scalability and realism. Algorithmically, initializing the learner with policies trained individually on each of the $m + 1$ objectives, rather than retraining from scratch, may improve convergence speed and stability. Additionally, ensembling-based techniques like in (Hussing et al., 2024) or other papers can be used to improve the performance of best-response in each round of the game. Finally, incorporating strategic interactions and incentive structures between human and robotic agents presents an intriguing avenue for understanding and enhancing complementarity in hybrid fulfillment systems.

References

Byeon, W., Park, G., Chae, J., Leshem, A., and Sung, Y. Multi-objective reinforcement learning with max-min criterion: A game-theoretic approach. *arXiv preprint arXiv:2510.20235*, 2025.

Calvo-Fullana, M., Paternain, S., Chamon, L. F. O., and Ribeiro, A. State augmented constrained reinforcement learning: Overcoming the limitations of learning with rewards. *IEEE Transactions on Automatic Control*, 69(7): 4275–4290, 2024. doi: 10.1109/TAC.2023.3319070.

Eaton, E., Hussing, M., Kearns, M., Roth, A., Sengupta, S. B., and Sorrell, J. Intersectional fairness in reinforcement learning with large state and constraint spaces, 2025. URL https://arxiv.org/abs/2502.11828.

El Shar, I., Wang, H., and Gupta, C. Multi-objective reinforcement learning for sustainable supply chain optimization. In *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, pp. 1–7. IEEE, 2023.

Freund, Y. and Schapire, R. E. Game theory, on-line prediction and boosting. In *Proceedings of the ninth annual conference on Computational learning theory*, pp. 325–332. ACM Press, 1996.

Geist, M. et al. Concave utility reinforcement learning, 2021. URL https://arxiv.org/abs/2106.03787.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

Hayes, C. F., Rădulescu, R., Bargiacchi, E., Källström, J., Macfarlane, M., Reymond, M., Verstraeten, T., Zintgraf, L. M., Dazeley, R., Heintz, F., et al. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

Huang, S., Dossa, R. F. J., Ye, C., Braga, J., Chakraborty, D., Mehta, K., and Araújo, J. G. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022.

Hussing, M., Kearns, M., Roth, A., Sengupta, S. B., and Sorrell, J. Oracle-efficient reinforcement learning for max value ensembles. *Advances in Neural Information Processing Systems*, 37:117657–117681, 2024.

Jaggi, M. Revisiting frank–wolfe: Projection-free sparse convex optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2013. URL https://proceedings.mlr.press/v28/jaggi13.html.

Kearns, M., Neel, S., Roth, A., and Wu, Z. S. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2564–2572, Stockholm, Sweden, 10–15 Jul 2018. PMLR. URL https://proceedings.mlr.press/v80/kearns18a.html.

Miryoosefi, S., Brantley, K., Daume III, H., Dudik, M., and Schapire, R. E. Reinforcement learning with convex constraints. *Advances in neural information processing systems*, 32, 2019.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533, 2015.

Park, G., Byeon, W., Kim, S., Havakuk, E., Leshem, A., and Sung, Y. The max-min formulation of multi-objective reinforcement learning: From theory to a model-free algorithm. *arXiv preprint arXiv:2406.07826*, 2024.

Paternain, S., Chamon, L., Calvo-Fullana, M., and Ribeiro, A. Constrained reinforcement learning has zero duality gap. *Advances in Neural Information Processing Systems*, 32, 2019.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sion, M. On general minimax theorems, 1958.

Tessler, C., Mankowitz, D. J., and Mannor, S. Reward constrained policy optimization. In *International Conference on Learning Representations*, 2019.

Yang, R., Sun, X., and Narasimhan, K. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. *Advances in neural information processing systems*, 32, 2019.

Zahavy, T., O'Donoghue, B., Desjardins, G., and Singh, S. Reward is enough for convex MDPs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2106.00661.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pp. 928–935. AAAI Press, 2003. ISBN 1577351894.

A. Algorithm Pseudocode

Algorithm 1 summarizes the repeated game procedure between the learner and the regulator. At each round, the learner computes an approximate best-response policy via DQN given the current Lagrange multipliers, and the regulator updates the multipliers via online gradient descent based on the observed constraint slacks. The time-averaged strategies are returned as the final output.

Algorithm 1 Repeated Game: Learner vs Regulator

Initialize: $\lambda_0 \in \Lambda$, total rounds T, step size η
$\bar{D} \leftarrow 0$, $\bar{\lambda} \leftarrow 0$ {Time averaged strategies}
for $t = 1$ to T **do**
  Learner: $D_t \leftarrow$ DQN_Best_Response(λ_{t-1})
  $g_t \leftarrow \text{EvaluatePolicy}(D_t)$ {Constraint slacks}
  Regulator: $\lambda_t \leftarrow \text{Proj}_{\Lambda}(\lambda_{t-1} - \eta g_t)$ {Online gradient descent}
  $\bar{D} \leftarrow \frac{1}{t}\big((t-1)\bar{D} + D_t\big)$ {Update time average}
  $\bar{\lambda} \leftarrow \frac{1}{t}\big((t-1)\bar{\lambda} + \lambda_t\big)$
end for
Return: $(\bar{D}, \bar{\lambda})$ {Time averaged strategies}

B. Implementation Details

Environment. We use an episodic Gymnasium environment with horizon $H = 1440 \times 10 = 14{,}400$ steps (10 simulated days at minute-level granularity). The state is a 12-dimensional feature vector as defined in (4), flattened with zero-padding/truncation. The action space consists of 8 discrete actions (ignore/not-ignore $\times$ source/destination $\times$ human/robot).

Reward. At each step, the scalarized reward is $r_t = 100 \times (\Delta\texttt{ETPH}_t + \lambda^\top c_t)$, where c_t is the 4-dimensional constraint slack vector: $N_{\texttt{large}}$ balance $\big((\alpha_1 - N_{\texttt{large}})/H\big)$, S/D ratio, human capacity $(L_S^H + L_D^H)$, and robot capacity $(L_S^R + L_D^R)$, with thresholds $\alpha = (950,\ 0.5,\ 150,\ 750)$.

DQN architecture and hyperparameters. The DQN network and training hyperparameters are summarized in Tables 2 and 3. Our implementation builds on CleanRL (Huang et al., 2022).

Table 2. DQN learner architecture.

Layer	Input dim	Output dim	Activation
Linear 1	12	120	ReLU
Linear 2	120	84	ReLU
Linear 3	84	$\lvert\mathcal{A}\rvert$	None

Table 3. DQN training hyperparameters.

Parameter	Value
Optimizer	Adam
Learning rate	10^{-4}
Replay buffer size	10,000
Batch size	128
Discount factor γ	0.99
Target network update	Hard copy every 500 steps
ε-greedy schedule	$1.0 \rightarrow 0.05$, linear over 50% of steps
Learning starts	After 500 steps
Learning frequency	Every 10 steps
Episodes per round	50

Regulator. The Lagrange multipliers $\lambda \geq 0$ are updated via projected online gradient descent (Theorem 3.2) with step size $\eta = C/(G\sqrt{T})$, where $C = 20{,}000$, $G = \max_i |\alpha_i| + 1$, and $T = 20$ rounds. After each gradient step, λ is projected onto $\mathbb{R}^m_+ \cap \{\|\lambda\|_1 \leq C\}$.

Evaluation. Policies are evaluated in two modes: (a) *mixture*: at each episode, a policy Q_t is sampled uniformly from $\{Q_1, \ldots, Q_T\}$ and evaluated under $\bar{\lambda} = \frac{1}{T}\sum_{t=1}^{T} \lambda_t$; (b) *single-policy*: each Q_t is evaluated over 20 episodes, and the best iterate is selected by average return. Reported metrics include episodic return, ETPH, and all four constraint slacks.

C. Additional Experimental Results

This section presents supplementary experimental results that complement the main evaluation in Section 4.3.

Single-objective learning curves. Figure 4 shows the episodic returns for DQN trained in a single-objective setting, optimizing only the ETPH reward. The steady increase in returns over training episodes confirms that the base RL algorithm is capable of learning effective policies in this environment, validating its use as the best-response oracle within the repeated game framework.

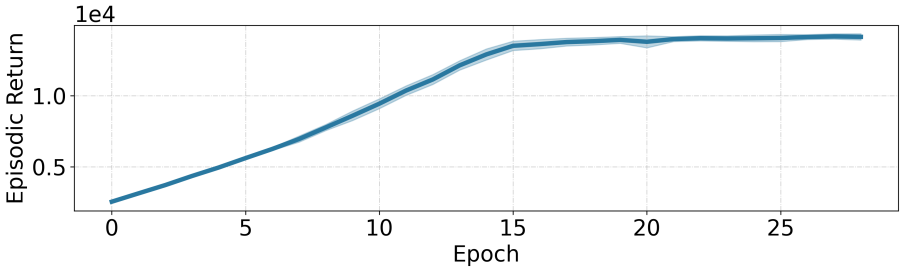


Figure 4. Episodic returns for DQN in the single-objective ETPH setting, showing unnormalized performance of a best-response policy optimizing ETPH.

Lagrange multiplier dynamics. Figure 5 tracks the evolution of the time-averaged Lagrange multipliers $\bar{\lambda}$ across training rounds. The $N_{\texttt{large}}$ and robot capacity multipliers remain near zero throughout, indicating that these constraints are largely inactive. In contrast, the S/D ratio and manual capacity multipliers exhibit significant oscillations, reflecting the regulator's effort to enforce the tighter constraints and the inherent tension between throughput and constraint satisfaction.

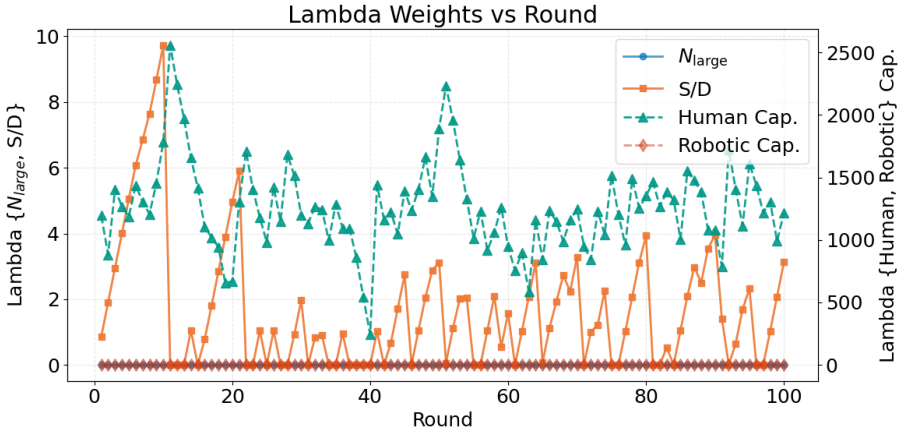


Figure 5. Average Lagrange multipliers $\bar{\lambda}$ over training rounds. The $N_{\texttt{large}}$ and robot capacity multipliers remain at zero, while the remaining constraints exhibit significant oscillations.

Objective–constraint trade-off trajectories. Figure 6 visualizes the trade-off between ETPH and constraint satisfaction as training progresses. Each point corresponds to a time-averaged policy at a given round, with color indicating the training stage. The trajectories show that the repeated game dynamics gradually steer the policy toward regions of improved constraint satisfaction, at the cost of moderate reductions in ETPH, consistent with the minimax convergence guarantees.

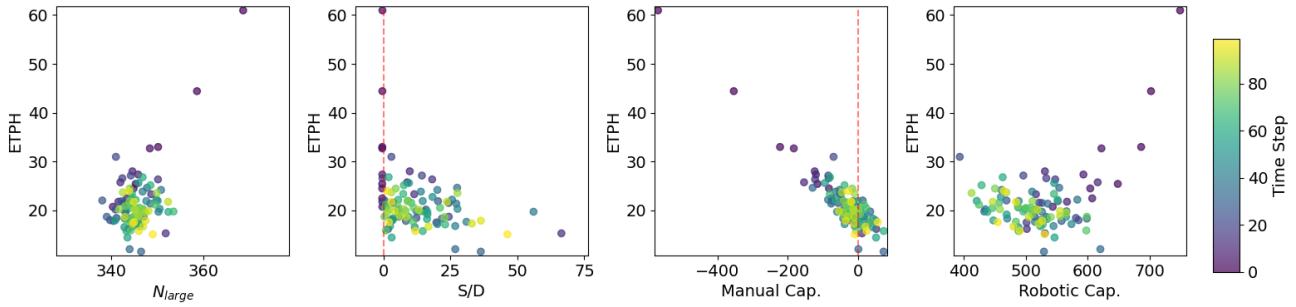


Figure 6. Phase diagram of ETPH versus constraint satisfaction across rounds. Color indicates training rounds, red dashed lines denote constraint thresholds. As training progresses, time averaged policies trade ETPH for improved S/D and manual capacity satisfaction.

D. Error Cancellations

Here, we will argue that if we have convergence of the randomized mixture of policies from the repeated game of no-regret vs. best-response dynamics from the learner and regulator, then there must be at least one policy in that set which we can select that approximately achieves the value of the game for this reformulated optimization problem. The broad intuition of this reformulation is that best-response vs. no-regret guarantees we satisfy the constraints on average for arbitrary convex constraints when feasible. The max of the constraint violations and 0 is still convex if the constraints are individually convex. Now, the constraint value is never negative under this reformulation, so satisfying the constraint on average is only possible if we satisfy on most iterates and hence on some iterate.

D.0.1. Reformulated Optimization Problem

In the original formulation both negative and positive values of the constraint violation were included, which enabled a phenomenon known as error cancellation to occur where on average, the constraints are close to satisfied, but individual policies violate different constraints. Here, we reformulate the constraints using the positive part so that feasibility in expectation cannot be achieved via cancellations across different constraints, and violations of any single constraint are penalized uniformly.

$$
\begin{aligned}
\max_{D\sim\Delta(\Pi)} \ & \mathbb{E}_{\pi\sim D}\left[\sum_{t=0}^{H-1} \mathtt{ETPH}_t(\pi)\right] \\
\text{s.t.} \quad & \max\left\{\mathbb{E}_{\pi\sim D}\left[\sum_{t=0}^{H-1} N_{\mathtt{large},t}(\pi)\right] - \alpha_1, 0\right\} \le 0, \\
& \max\left\{\mathbb{E}_{\pi\sim D}\left[-\sum_{t=0}^{H-1} (S/D)_t(\pi)\right] + \alpha_2, 0\right\} \le 0, \\
& \max\left\{\mathbb{E}_{\pi\sim D}\left[\sum_{t=0}^{H-1} \left(L^H_{S,t}(\pi) + L^H_{D,t}(\pi)\right)\right] - \alpha_3, 0\right\} \le 0, \\
& \max\left\{\mathbb{E}_{\pi\sim D}\left[\sum_{t=0}^{H-1} \left(L^R_{S,t}(\pi) + L^R_{D,t}(\pi)\right)\right] - \alpha_4, 0\right\} \le 0,
\end{aligned}
$$

This can be even further reduced to

$$\max_{D\sim\Delta(\Pi)} \mathbb{E}_{\pi\sim D}\left[\sum_{t=0}^{H-1} \texttt{ETPH}_t(\pi)\right]$$
$$\text{s.t.} \quad \max\left\{\mathbb{E}_{\pi\sim D}\left[\sum_{t=0}^{H-1} N_{\texttt{large},t}(\pi)\right] - \alpha_1, \mathbb{E}_{\pi\sim D}\left[-\sum_{t=0}^{H-1}(S/D)_t(\pi)\right] + \alpha_2,\right.$$
$$\left.\mathbb{E}_{\pi\sim D}\left[\sum_{t=0}^{H-1}\left(L^H_{S,t}(\pi) + L^H_{D,t}(\pi)\right)\right] - \alpha_3, \mathbb{E}_{\pi\sim D}\left[\sum_{t=0}^{H-1}\left(L^R_{S,t}(\pi) + L^R_{D,t}(\pi)\right)\right] - \alpha_4, 0\right\} \leq 0,$$

D.0.2. FINDING THE ESTIMATED BEST ITERATE

We select the best D_t from the algorithm based upon an estimated Lagrangian with respect to $\bar{\lambda}$. Let $L(D,\lambda) = \mathbb{E}_{\pi\sim D}\left[V_0^\pi(\mu)\right] - \lambda\,[\,g(D)\,]_+$, where $g(D) := \max_{i\in[m]}(\mathbb{E}_{\pi\sim D}[V_i^\pi(\mu)] - \alpha_i)$ and $\lambda \geq 0$. In this presentation, it is worth noting that we are assuming that without loss of generality (WLOG) we can formulate our constrained optimization problem to be of this form and that all signs are taken into account in the assignment of α_i. Note that we could modify our reward/value function bounds to take into account possible negative values, but this would not change the analysis our bounds much so we use the following WLOG formulation for clarity. We are interested in finding $\arg\max_{t\in[T]} L(D_t, \bar{\lambda})$. However, we do not have an exact estimate of the Lagrangian. Instead, we compute $\hat{L}$ as follows: suppose we form Monte Carlo estimates from n i.i.d. episodes per iterate: $\widehat{V}_0(D_t) = \frac{1}{n}\sum_{j=1}^n \hat{V}_0^j(\tau_{t,j})$, $\widehat{c}_i(D_t) = \frac{1}{n}\sum_{j=1}^n \hat{V}_i^j(\tau_{t,j})$, $\widehat{g}(D_t) := \max_i(\widehat{c}_i(D_t) - \alpha_i)$, and $\widehat{L}(D_t, \bar{\lambda}) := \widehat{V}_0(D_t) - \bar{\lambda}\,[\widehat{g}(D_t)]_+$.

D.0.3. LEARNER'S BEST-RESPONSE CONVEX RL

Earlier we discussed approximate best-response for the learner in terms of the ability to approximately solve single-objective RL problems. In this section, we discuss this notion when the objective functions are concave in the variable D corresponding to the distribution over policies.

Definition 3 (Episodic Occupancy Measure)**.**

$$q_h^\pi(s,a) = \Pr^{\pi,\mu}(s_h = s, a_h = a)$$

Define $d = \sum_{h=0}^{H-1} q_h^\pi$. In practice, we may have a policy π specified by a $D \in \Delta(\Pi)$, so we may use the shorthand notation μ_D to denote this summed over time occupancy measure over the given policy. In an episodic RL setting especially with fixed rewards over time, we can denote $\mathbb{E}_{\pi\sim D}[V^\pi(\mu)] = \langle r, \mu_D\rangle$ (or $\langle r, d\rangle$ using our notation above) as will be shown below.

$$
\begin{aligned}
V_i^{\pi}(\mu) &= \mathbb{E}_{\pi,\mu}\Big[\sum_{h=0}^{H-1} r(s_h, a_h)\Big] \\
&= \mathbb{E}_{\pi,\mu}\Big[\sum_{h=0}^{H-1}\sum_{s,a} r(s,a) I(s_h = s, a_h = a)\Big] \\
&= \sum_{h=0}^{H-1}\sum_{s,a} r(s,a)\mathbb{E}_{\pi}[I(s_h = s, a_h = a)] \\
&= \sum_{h=0}^{H-1}\sum_{s,a} r(s,a)\Pr^{\pi,\mu}(s_h = s, a_h = a) \\
&= \sum_{h=0}^{H-1}\sum_{s,a} r(s,a) q_h^{\pi}(s,a) \\
&= \sum_{s,a} r(s,a)\sum_{h=0}^{H-1} q_h^{\pi}(s,a) \\
&= \langle r, d\rangle
\end{aligned}
$$

Let $r(s,a)$ denote the per-step *reward*. For each constraint $i = 1, \ldots, m$, let $r_i(s,a)$ denote the per-step *constraint signal*. For a distribution over policies D with occupancy measure μ_D, define

$$
r_i(D) := \langle \mu_D, r_i\rangle = \mathbb{E}_{\pi\sim D}\Big[\sum_{h=0}^{H-1} r_i(s_h, a_h)\Big].
$$

Given thresholds α_i, the *constraint violation* for constraint i is $g_i(D) := r_i(D) - \alpha_i$, and the aggregate violation is

$$
g(D) := \max_{i\in[m]} g_i(D), \qquad [x]_+ := \max\{0, x\}.
$$

The Lagrangian at multiplier $\lambda \geq 0$ is thus

$$
L(D, \lambda) = r(D) - \lambda\,[\,g(D)\,]_+.
$$

For fixed $\lambda_t \geq 0$ and current iterate D_t, write

$$
L(D, \lambda_t) = \langle \mu_D, r\rangle - \lambda_t\,[\max_i\{\langle \mu_D, r_i\rangle - \alpha_i\}]_+.
$$

Because $D \mapsto L(D, \lambda_t)$ is concave, any supergradient at D_t yields an *affine upper bound*.

More directly, for a given $\lambda_t \in \Lambda$, we can write

$$
f(d) = L(d, \lambda_t) = \langle d, r\rangle - \lambda_t\,[\max_i\{\langle d, r_i\rangle - \alpha_i\}]_+.
$$

Let

$$
\partial_d \mathcal{L}(d, \lambda) = \begin{cases} r_0 \quad \text{if } [g(d)]_+ \leq 0 \\ r_0 - \lambda_t \sum_j p_j r_j \text{if } [g(d)]_+ \geq 0 \text{ where } j \in \arg\max g_i(d) \text{ and } p_j \geq 0\ \forall j, \sum_j p_j = 1 \end{cases}
$$

In practice, to evaluate this we would take the given policy specified by D, evaluate the performance of the policy over all the constraints, then compute the gradient based upon the above.

Then with $s_t := r - \lambda_t \sum_i p_{t,i}\, r_i$ where p_t is set according to the supergradient, we have, for all D,

$$
L(D, \lambda_t) \;\leq\; L(D_t, \lambda_t) + \langle \mu_D - \mu_{D_t},\; s_t\rangle.
$$

(Maximizing the linear form $\langle \mu_D, s_t \rangle$ is the standard *linear minimization oracle* / RL call used in Frank–Wolfe and convex RL (Jaggi, 2013; Zahavy et al., 2021; Geist et al., 2021).) Thus the learner's ε-*best-response* can be implemented by solving the single linear problem

$$\max_D \langle \mu_D,\ s_t \rangle \quad \Longleftrightarrow \quad \max_D \mathbb{E}_{\pi \sim D}\Big[\sum_h \underbrace{\big(r(s_h, a_h) - \lambda_t \sum_i p_{t,i} r_i(s_h, a_h)\big)}_{r_{\lambda_t}(s_h, a_h)} \Big]$$

up to additive error ε. Therefore, we have that if $\langle \mu_{D^*}, s_t \rangle - \langle \mu_{D_{t+1}}, s_t \rangle \le \varepsilon$, then

$$\max_D L(D, \lambda_t) - L(D_{t+1}, \lambda_t) \ \le\ \varepsilon.$$

D.0.4. Frank Wolfe Procedure

For every λ selected by the regulator, the learner will have to solve a sequence of RL problems over linearized reward functions in order to approximately best respond to the regulator's strategy. This will be done by computing super-gradients of the Lagrangian. In this setting, though the objective is concave in D and no longer linear, each D_t corresponds to a representative policy based upon the single policies returned from the FW linear oracles.

Algorithm 2 Learner Concave Best-Response

Receive: $\lambda \in \Lambda$, step size α, tolerance ϵ
Initialize occupancy measure: $x^{(1)} \in \mathcal{O}$ {Current iterate (occupancy measure)}
for $w = 1$ to W **do**
  $\alpha_w \leftarrow \frac{2}{1+w}$
  $d^{(w)} \leftarrow \arg\max_{d \in \mathcal{O}} \left\langle \partial \mathcal{L}\big(x^{(w)}\big),\ d \right\rangle$ {DQN with scalarized reward induced by λ}
  $g^{(w)} \leftarrow \left\langle \partial \mathcal{L}\big(x^{(w)}\big),\ d^{(w)} - x^{(w)} \right\rangle$
  if $g^{(w)} \le \epsilon$ **then**
    break
  end if
  $x^{(w+1)} \leftarrow (1 - \alpha_w)\, x^{(w)} + \alpha_w\, d^{(w)}$ {Convex update}
end for
Compute policy: $\pi(s, a) \leftarrow \dfrac{x^{(w_\star)}(s, a)}{\sum_{a' \in \mathcal{A}} x^{(w_\star)}(s, a')}$ {$w_\star$ is final iterate}
Assign $D \in \Delta(\Pi)$ accordingly
Return: D {Learner best-response}

D.0.5. Theoretical Results

We now present our theoretical results on the single round's D_t selected out of this procedure.

Let $V_i^\pi(\mu)$ denote the (episodic) return for objective/constraint i under policy π from initial distribution μ. Define the per-policy worst-constraint violation

$$g(D) \ :=\ \max_{i \in [m]} \Big(\mathbb{E}_{\pi \sim D}[V_i^\pi(\mu)] - \alpha_i \Big), \qquad [x]_+ := \max\{0, x\}.$$

For a distribution $D \in \Delta(\Pi)$ over policies and scalar multiplier $\lambda \in [0, C]$, define the *reformulated* Lagrangian

$$L(D, \lambda) \ :=\ \mathbb{E}_{\pi \sim D}[V_0^\pi(\mu)] \ -\ \lambda\, [g(D)]_+. \tag{11}$$

Let $L^\star := \min_{\lambda \in [0, C]} \max_{D \in \Delta(\Pi)} L(D, \lambda)$ be the minimax value of the reformulated game. Assume the repeated game dynamics produce a time-average pair $(\bar{D}, \bar{\lambda})$ that is a ν-approximate minimax equilibrium in the sense of Definition 1 (applied to (11)). Papers like (Eaton et al., 2025) and (Kearns et al., 2018) have already proven that the time-averaged

solution satisfies this property, so we are building on the basis of those results and this is not truly a separate assumption. We simply abstract away that analysis since it has been done elsewhere. In particular, we will use the learner-side condition

$$L(\bar{D}, \bar{\lambda}) \;\geq\; \max_{D \in \Delta(\Pi)} L(D, \bar{\lambda}) \;-\; \nu \;\geq\; L^\star - \nu. \tag{12}$$

We assume constraint thresholds satisfy $\alpha_i \in [-H, H]$. Since per-step rewards lie in $[0, 1]$ and the horizon is H, this implies $V_i^\pi(\mu) \in [0, H]$ and hence

$$g_i(D) = \mathbb{E}_{\pi \sim D}[V_i^\pi(\mu)] - \alpha_i \in [-H,\, 2H], \qquad [g_i(D)]_+ \in [0,\, 2H].$$

Lemma D.1 (Uniform Concentration for $\widehat{L}$)**.** *Assume per-step rewards and per-step constraint signals are bounded in* $[0, 1]$*, so that for every policy* π*,*

$$0 \leq V_0^\pi(\mu) \leq H, \qquad 0 \leq V_i^\pi(\mu) \leq H \;\; (i \in [m]),$$

and assume $\alpha_i \in [-H, H]$ *for all* i*, so that*

$$0 \leq [g(D)]_+ \leq 2H.$$

Fix $\bar{\lambda} \in [0, C]$ *and iterates* $\{D_t\}_{t=1}^T$*. For each* t*, form* $\widehat{L}(D_t, \bar{\lambda})$ *from* n *i.i.d. rollouts of a policy* $\pi \sim D_t$ *(sampling* π *once per rollout, then generating one episode), using the unbiased Monte Carlo estimates of* $V_0^\pi(\mu)$ *and* $V_i^\pi(\mu)$ *to compute estimates of* $[g(D)]_+$ *and* $V_0^\pi(\mu)$*. Then for any* $\varepsilon, \delta \in (0, 1)$*,*

$$\Pr\left(\max_{t \in [T]} \left|\widehat{L}(D_t, \bar{\lambda}) - L(D_t, \bar{\lambda})\right| \;\leq\; \varepsilon\right) \;\geq\; 1 - \delta$$

whenever

$$n \;\geq\; \frac{(1 + 2\bar{\lambda})^2 H^2}{2\varepsilon^2} \, \ln \frac{2T}{\delta}. \tag{13}$$

Proof. For a single rollout of (π, τ) with $\pi \sim D_t$, define the bounded random variable

$$Z := V_0^\pi(\mu) - \bar{\lambda}\,[g(\pi)]_+ \in [-2\bar{\lambda}H,\, H],$$

so the range is at most $(1 + 2\bar{\lambda})H$. The estimator $\widehat{L}(D_t, \bar{\lambda})$ is the empirical mean of n i.i.d. copies of Z, hence Hoeffding's inequality gives

$$\Pr\left(|\widehat{L}(D_t, \bar{\lambda}) - L(D_t, \bar{\lambda})| > \varepsilon\right) \;\leq\; 2\exp\left(-\frac{2n\varepsilon^2}{(1 + 2\bar{\lambda})^2 H^2}\right).$$

A union bound over $t \in [T]$ yields the claim under (13). □

Define the iterate–average Jensen gap at $\bar{\lambda}$:

$$J_{\text{avg}}(\bar{\lambda}) \;:=\; L(\bar{D}, \bar{\lambda}) \;-\; \frac{1}{T}\sum_{t=1}^{T} L(D_t, \bar{\lambda}) \;\; \geq 0, \tag{14}$$

which is nonnegative because $D \mapsto L(D, \bar{\lambda})$ is concave.

Theorem D.1 (Single Iterate Extraction)**.** *Assume* $(\bar{D}, \bar{\lambda})$ *satisfies* (12) *for the reformulated Lagrangian* (11)*. Let* $t^\star \in \arg\max_{t \in [T]} \widehat{L}(D_t, \bar{\lambda})$*, using* $\widehat{L}$ *as in Lemma D.1. Then with probability at least* $1 - \delta$*,*

$$L(D_{t^\star}, \bar{\lambda}) \;\geq\; L^\star \;-\; (\nu + 2\varepsilon + J_{avg}(\bar{\lambda})). \tag{15}$$

Moreover, there exists a policy $\pi^\star$ *in the support of* $D_{t^\star}$ *such that*

$$L(D_{\pi^\star}, \bar{\lambda}) \;\geq\; L^\star \;-\; (\nu + 2\varepsilon + J_{avg}(\bar{\lambda})). \tag{16}$$

Proof. On the uniform concentration event of Lemma D.1, for all t we have

$$L(D_t, \bar{\lambda}) \;\le\; \widehat{L}(D_t, \bar{\lambda}) + \varepsilon \;\le\; \widehat{L}(D_{t^\star}, \bar{\lambda}) + \varepsilon \;\le\; L(D_{t^\star}, \bar{\lambda}) + 2\varepsilon,$$

so $\max_t L(D_t, \bar{\lambda}) \le L(D_{t^\star}, \bar{\lambda}) + 2\varepsilon$ and hence

$$L(D_{t^\star}, \bar{\lambda}) \;\ge\; \max_{t \in [T]} L(D_t, \bar{\lambda}) \;-\; 2\varepsilon. \tag{17}$$

Because $L(\cdot, \bar{\lambda})$ is concave in D and by definition of the Jensen gap,

$$L(\bar{D}, \bar{\lambda}) \le \; \frac{1}{T} \sum_{t=1}^{T} D_t + J_{\mathrm{avg}}(\bar{\lambda}) \le \max_{t \in [T]} L(D_t, \bar{\lambda}) + J_{\mathrm{avg}}(\bar{\lambda}).$$

Combining with (17) and (12) yields

$$L(D_{t^\star}, \bar{\lambda}) \;\ge\; L(\bar{D}, \bar{\lambda}) - 2\varepsilon - J_{\mathrm{avg}}(\bar{\lambda}) \;\ge\; L^\star - (\nu + 2\varepsilon + J_{\mathrm{avg}}(\bar{\lambda})),$$

which is (15).

Note that the FW oracle is already returning a representative single policy, but another way to see the single policy statement is as follows. For the single-policy statement, expand (11) at $(D_{t^\star}, \bar{\lambda})$:

$$L(D_{t^\star}, \bar{\lambda}) \;=\; \mathbb{E}_{\pi \sim D_{t^\star}} L(\pi, \bar{\lambda}).$$

Therefore there exists $\pi^\star$ in the support of $D_{t^\star}$ with $L(D_{\pi^\star}, \bar{\lambda}) \ge L(D_{t^\star}, \bar{\lambda})$, giving (16). □

While the Jensen gap can be nonzero due to concavity, in our setting the Lagrangian is piecewise linear in D, making the gap empirically smaller than the worst-case. To analyze constraint violations, we can use a bound on the average constraint violation level coupled with non-negativity to reason about the constraint violation level of a single iterate.