Zeroth-Order Implicit Reinforcement Learning for Sequential Decision Making in Distributed Control Systems

Vanshaj Khattar Virginia Tech
Qasim Wani[*] Virginia Tech
Harshal Kaushik Virginia Tech
Zhiyao Zhang[*] Xiangtan University
Ming Jin Virginia Tech

Abstract

Despite being reliable and well-established, convex optimization models, once built, do not adapt to the changing real-world conditions or fast evolving data streams. Such rigidness may lead to higher costs and even unsafe scenarios, rendering traditional model-based approaches inadequate for modern distributed control systems (DCSs). Recently, learning-based techniques have achieved remarkable success in operating systems with unknown dynamics; nevertheless, it is difficult to include safety and assurance during the online process. We propose a novel zeroth-order implicit reinforcement learning framework for adaptive decision making in DCSs. We solve the sequential decision making problem with a policy class based on convex optimization. The reinforcement learning (RL) agent aims to adapt the parameters within the optimization model iteratively to the dynamically changing conditions. This approach enables us to include general constraints within the RL framework. We also improve the convergence of the proposed zeroth-order method by introducing a guidance mechanism. Our theoretical analysis of the non-asymptotic global convergence rate reveals the benefits of the guidance factor. The effectiveness of our proposed method is validated on two real-world applications, including a recent RL competition, showing a significant improvement over existing algorithms.

1 Introduction

Sequential decision making problems appear in numerous domains, such as energy, supply chain management, finance, health, and robotics to name a few. The major challenge in these applications is to make decisions under uncertainty (that may arise in the forms of uncertain demands of electricity, goods, changing prices, and unstable weather), which may be attributed to various factors such as dynamic environments and human interventions. A large class of problems involve distributed control systems, where the objective is to control multiple systems to achieve a desired objective while satisfying domain-specific constraints (Zhou et al., 2020; Wilson et al., 2020).

Reinforcement learning is promising for DCSs for two key reasons. First, it allows the agent to act without the need of a predefined model—a feature of particular interests for large-scale, complex systems, where it is not cost-effective to develop such a model and information sharing is fundamentally limited. Also, by continuously adapting the agent to the environment, RL can systematically account for dynamic uncertainty and environmental drift. Notably, RL agents are rarely deployed in a standalone manner but as part of a larger pipeline with infrastructures and humans in the loop. Thus, it is imperative for RL agents to make real-time decisions that respect constraints on their physical and social ramifications. Recently, (Dulac-Arnold et al., 2021) identified a set of challenges for real-world RL (all pertinent to DCSs). We are poised to address seven out of the nine challenges in the present study (as evidenced in the success of our method in a recent RL competition, detailed in Section 4), including: *1)* the ability to learn on live systems from limited samples; *2)* learning and acting in high-dimensional state and action spaces; *3)* reasoning about system constraints that should never or rarely be violated; *4)* interacting with systems that are partially observable, which can alternatively be viewed as systems that are stochastic; *5)* learning from multiple objectives; *6)* the ability to provide actions quickly; and *7)* providing system operators with explainable policies.

[*]Works completed during the summer internship in Dr. Ming Jin's group as undergraduate researchers.

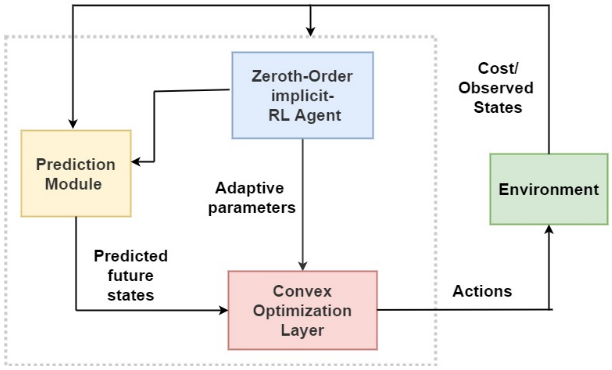

Figure 1: Basic architecture for ZO-iRL.

To address the above challenges, we propose a paradigm shift from traditional representations of policies (e.g., linear functions, neural networks, nonparametric functions) to a new class of policies that combines the synergistic strength of optimization with RL. Indeed, optimization (especially convex optimization) has become the de facto standard in industrial systems with profound theoretical foundations and many well-established formulations for control and planning applications (Boyd et al., 2004). Such approaches can easily encode physical and even social constraints (in the form of nonlinear functions, variational inequalities, or fixed point equations), and can gracefully handle problems with millions of decision variables. Although considered as being reliable and well established, optimization models, once built, cannot adapt to the changing real-world conditions—or rather, current approaches are "rigid."

The crux of our idea is to adapt the optimization models (including parameters in both the objective function and constraints) with RL. In particular, we propose the zeroth-order implicit RL (ZO-iRL), which consists of a convex optimization layer within the RL agent (see Fig. 1 for the basic architecture). Such a method is implicit because the policy actions are implicitly defined with respect to the parameters of the optimization model, the update of which also implicitly depends on the rewards. Zeroth-order optimization algorithm is easy to implement in general, but may potentially suffer from scalability/convergence issues. Nonetheless, we note that in most DCS applications, the parameters of the corresponding optimization model have clear interpretations. Thus, we design a mechanism to use proper guidance on the search of optimal parameters. Under some mild conditions, we show a non-asymptotic convergence rate that reveals the effects of such guidance factors. We further validate the proposed ZO-iRL method in two real-world applications.

Our contributions are as follows:

1. We introduce a novel implicit RL approach for adaptive and assured decision making in DCSs.
2. We propose a guidance mechanism to improve the convergence of the zeroth-order search and analyze the non-asymptotic convergence rate.
3. We validate our approach on two real-world problems, including one recent RL competition.

The rest of the paper is organized as follows. Section 2 covers related work by various research communities on sequential decision making for DCSs. Section 3 presents the research methodology, including problem formulation, algorithm design, and convergence analysis. We validate the proposed algorithm on two real-world applications in Section 4. The paper ends with concluding remarks and discussions of future directions.

2 Related work

Optimal control and stochastic optimal control are well-known approaches to sequential decision makings in DCSs (Nozhati et al., 2020; Khattar, 2021). Convex optimization is another avenue (Agrawal et al., 2020). Most of the existing works assume a known dynamic model of the system, which makes them less applicable in many scenarios. Stochastic programming models have been designed to principally perform sequential decision making under uncertainty. Various large-scale stochastic program models have been proposed in the literature to handle future uncertainty (Carpentier et al., 2014; Jin et al., 2011; Lium et al., 2009). The major drawback is that they can potentially become computationally expensive due to the rapid expansion of scenario trees in multi-stage stochastic programming. Our method is computationally lightweight due to the deterministic approximation of future uncertainty within a convex optimization policy class.

Recently, RL-based methods have gained popularity for controlling systems with unknown dynamics and/or high-dimensional state and action spaces (Ebert et al., 2018; Gu et al., 2017). However, unlike control-theoretic approaches, these methods generally lack the necessary mathematical framework to provide guarantees on correctness, such as safety and constraint satisfaction, causing concerns about trustworthiness (Amodei et al., 2016; Stoica et al., 2017). Some recent attempts to overcome these drawbacks can be categorized into constrained RL (Choi et al., 2020; Ammanabrolu and Hausknecht, 2020; Stooke et al., 2020), robust control approaches (Jin and Lavaei, 2020; Yin et al., 2021; Gu et al., 2021), Lyapunov-based methods (Berkenkamp et al., 2017; Chow et al., 2018; Perkins and Barto, 2002), Hamilton–Jacobi reachability methods (Gillula and Tomlin, 2012; Fisac et al., 2018), control-barrier functions (Ames et al., 2016; Cheng et al., 2019; Dean et al., 2020; Qin et al., 2021), and robust MPC (Hewing et al., 2020). Lately, game-theoretic approaches have emerged to coordinate multiple systems in DCSs (Zhang et al., 2021). Our method is a step forward in the area of Constrained RL. The prime advantage of our method is that it is adaptive to any applications due to the simplicity and ubiquity of convex optimization control policies. Donti et al. (2021) also introduces an optimization layer based on Lyapunov guarantees for safety. However, their method requires model information and is first-order. Our method is model-free and zeroth-order (under guidance).

The present work is closely related to (Agrawal et al., 2020; Ghadimi et al., 2020), which share the line of thinking that uses convex optimization as a policy class to handle uncertainty. In (Agrawal et al., 2020), convex optimization control policies are learned by tuning the parameters within the convex optimization layer. We extend their method to systems with unknown dynamics by incorporating a lookahead model (predictions) in our formulation. The parametric cost function approximation (PCFA) framework in (Ghadimi et al., 2020) is limited to linear or affine PCFAs. We extend their method to parametric convex cost functions approximations inspired by many attractive properties of convex optimization models (Agrawal et al., 2020).

3 Methodology

3.1 Problem formulation

Consider a sequential decision making problem in a DCS with N subsystems, each controlled by an agent. Furthermore, we allow couplings among different subsystems within the DCS, so that the evolution of one subsystem may affect the states of the others. The goal is to learn an optimal policy for each agent that minimizes the overall cost over a period of evaluation.

Formally, we aim to find a set of optimal policies $\Pi = \{\pi_1, \pi_2, ..., \pi_N\}$ for each agent in the DCS that minimizes an objective function defined over a finite time horizon T:

$$\min_{\Pi=\{\pi_1,\dots,\pi_N\}} \mathbb{E}\left[\sum_{t=0}^{T}\sum_{i=1}^{N} C_{t,i}\big(x_t^i, \pi_i(x_t^i), \{x_t^l\}_{l\in L(i)}\big)\right], \tag{1}$$

where $x_t^i \in \mathbb{R}^{n_i}$ is the state of the i-th subsystem at time t; $\pi_i : \mathbb{R}^{n_i} \to \mathbb{R}^{m_i}$ is the policy of agent i, so $\pi_i(x_t^i)$ is the action taken at time t. Let $x_{t+1}^i = f_{M_i}(x_t^i, \pi_i(x_t^i), W_{t+1}^i)$ be the state transition function for the i-th subsystem, which is unknown to us. We use the random variable W to capture the randomness from all possible sources, the realization of which for subsystem i at time $t+1$ is represented by W_{t+1}^i. Note that the expectation is taken over the initial states and the random variable W (which may affect the state transitions). Denote the cost function as $C_{t,i} : \mathbb{R}^{n_i+m_i} \to \mathbb{R}$, so $C_{t,i}\big(x_t^i, \pi_i(x_t^i), \{x_t^l\}_{l\in L(i)}\big)$ represents the cost incurred at time t for subsystem i under policy π_i, where $\{x_t^l\}_{l\in L(i)}$ represents the collection of states of the set of subsystems $L(i)$ coupled with subsystem i.

Problem (1) is not possible to address in practice because of the unknown underlying dynamics and randomness involved. To solve for the optimal policies in (1), we employ a lookahead model formulation (Powell and Meisel, 2015) to account for the impact of present decision on the future outcomes. Thereby, the optimal policy at time t can be computed as

$$\pi_i^\star(x_t^i) = \underset{\pi_i}{\operatorname{argmin}}\Bigg(C_{t,i}\big(x_t^i, \pi_i(x_t^i), \{x_t^l\}_{l\in L(i)}\big) + \\ \mathbb{E}\left[\sum_{t'=t+1}^{T} C_{t',i}\big(x_{t'}^i, \pi_i(x_{t'}^i)\Big| x_t^i, \pi_i(x_t^i)\right]\Bigg) \tag{2}$$

The formulation in (2) can also be seen as a Bellman equation. The main challenge in solving (2) is that the observed costs usually contain noise and are random in nature (true for many real-world applications). Costs of agent i are also dependent on the the actions of the others.

3.2 Zeroth-order implicit RL framework

To solve (2), we construct a surrogate convex cost function $\tilde{C}_{t,i}$, which is parametrized by some parameters $\zeta^i = \{\zeta_1^i, \cdots \zeta_T^i\}$. This allows us to write (2) as:

$$\pi_i^\star(x_t^i|\zeta^i) = \underset{\pi_i^\star}{\operatorname{argmin}}\Bigg(\tilde{C}_{t,i}\big(x_t^i, \pi_i(x_t^i)|\zeta^i\big) \\ + \sum_{t'=t+1}^{T} \tilde{C}_{t,i}\big(\hat{x}_{t'}^i, \pi_i(\hat{x}_{t'}^i)|\zeta^i\big)\Bigg) \tag{3}$$

where $\hat{x}_{t'}^i$ are the predicted future states, $\zeta^i \in \mathbb{R}^{d_i}$ is a d_i-dimensional parameter vector associated with the i-th subsystem, and it considers a decentralized setting where the agent can only access its own state (but not others). Construction of such a surrogate convex cost function should be such that desired behavior is encouraged. This is closely related to the reward design problem in RL (Prakash et al., 2020). Formulation (3) is a convex optimization problem that can be solved, provided that the estimates of the future states are reliable. The optimized policy obtained from (3) can be directly applied to the environment. The overall goal of (1) can now be restated as: *find the best possible parameters for* (3)*, such that optimal policy being deployed in environment achieves minimum cost.* This motivates the ZO-iRL method, where the RL-agent finds the best possible parameters for the surrogate convex cost function $\tilde{C}_{t,i}$. The policy that is being deployed in the environment will be computed after solving the implicit convex optimization problem. The ZO-iRL framework can be formulated (and understood) as a bilevel optimization problem:

$$\zeta^i = \underset{\zeta^i}{argmin}\,\mathbb{E}\left[\sum_{t=0}^{T} C_{t,i}\big(x_t^i, \pi_i^\star(x_i^t), \{x_t^l\}_{l\in L(i)}|\zeta^i\big)\right], \tag{4}$$

where the policy $\pi_i^\star(x_{t'}^i|\zeta^i)$ is obtained after solving an implicit convex optimization problem for the i-th subsystem with a fixed instantiation of parameters ζ^i:

$$\begin{aligned}\pi_i^\star(x_t^i|\zeta^i) = \underset{\pi_i^\star}{\operatorname{argmin}} \Bigg(& \tilde{C}_{t,i}\big(x_t^i, \pi_i(x_t^i)|\zeta^i\big) \\ & + \sum_{t'=t+1}^{T} \tilde{C}_{t,i}\big(\hat{x}_{t'}^i, \pi_i(\hat{x}_{t'}^i)|\zeta^i\big)\Bigg) \quad (5)\\ \text{subject to } \; & g_j(x_t^i, \pi_i(x_t^i)) \leq 0 \;\;;\;\; j \in \mathcal{I} \\ & h_j(\pi_i(x_t^i)) = 0 \;\;\;;\;\; j \in \mathcal{E},\end{aligned}$$

where $g_j : \mathbb{R}^{m_i+n_i} \to \mathbb{R}$ for all $j =\in \mathcal{I}$ are convex in $\pi_i(x_t^i)$ and $h_j : \mathbb{R}^{m_i} \to \mathbb{R}$ for all $j \in \mathcal{E}$ are affine in $\pi_i(x_t^i)$.

This bilevel optimization framework consists of an outer optimization layer (4) and an implicit convex optimization layer (5). The RL agent (outer layer) is represented using (4) with the aim to learn the parameters ζ^i that minimize the observed costs $C_{t,i}$ in the environment. The observed costs $C_{t,i}$ are dependent on the optimal policy computed by the implicit convex optimization layer.

3.3 Guided random search

This section presents the method for the RL agent to learn the optimal parameters within the convex optimization layer, namely, to solve the outer layer optimization problem. The parameters of an optimization model often carry clear physical meanings—in practice, they are typically set by experts based on experience or system specifications. Thus, we propose a guidance mechanism that enables the search algorithm to leverage such domain knowledge for faster and robust convergence. This zeroth-order method, given in Algorithm 1, is partly inspired by the "method of generations" (see (Zhigljavsky, 2012)).

Algorithm 1 ZO-iRL Algorithm

```
Input: $N_c$, $\sigma$, $\rho$, $P_1 = \mathcal{N}(0,\sigma^2)$, $k = 1$
Output: $\zeta^\star$
for k=1,2, $\cdots$ do
    Sample $N_c$ candidates from the distribution $P_k$: $\zeta_1^{(k)}, \zeta_2^{(k)}, \cdots, \zeta_{N_c}^{(k)}$
    for $j = 1, \cdots N_c$ do
        For each agent, adopt the policy (5) parametrized by $\zeta_j^{(k)}$ to control its corresponding subsystem over the horizon $T$
        Observe cost $C(\zeta_j^{(k)})$ provided by sampling from the objective function of (4)
    end for
    Sort the $N_c$ candidates on the basis of costs obtained. Let $\zeta_1^\star$ and $\zeta_2^\star$ be the best parameters.
    Calculate $A_g^{(k)} = mean(\zeta_1^\star, \zeta_2^\star)$
    Compute the guidance for each of the candidate by:
        $$G(\zeta_j^{(k)}) = \zeta_j^{(k)} + \rho(A_g^{(k)} - \zeta_j^{(k)})   \quad (6)$$
    Compute the new probability distribution $P_{k+1}$ for the next:
        $$P_{k+1}(d\zeta) = \sum_{j=1}^{N_c} r_j^{(k)} T\big(\zeta_j^{(k)}, d\zeta\big)   \quad (7)$$
    where:
        $$r_j^{(k)} = \frac{\exp[-C_t(\zeta_j^{(k)})]}{\sum_{j=1}^{N_c} \exp[-C_t(\zeta_j^{(k)})]}   \quad (8)$$
        $$T(\zeta_j^{(k)}, d\zeta) = \mathcal{N}(G(\zeta_j^{(k)}), \sigma^2)\mu(d\zeta)   \quad (9)$$
    and $\mu$ is a fixed probability measure on $(Z, \mathcal{B})$
    Increment $k \to k+1$
```

In a nutshell, at each iteration, the algorithm randomly samples a set of candidates for exploration. The sampling distribution is determined by the noisy costs observed for the previous candidates. In particular, for agent i at time t, we observe a noisy cost sample j:

$$C_{\mathrm{t}}(\zeta_j^{(k)}) = f_{\mathrm{true}}(\zeta_j^{(k)}) + w_k(\zeta_j^{(k)}) \tag{10}$$

where f_{true} is the underlying true cost function that we cannot observe (corresponding to the objective in (4)) and $w_k(\zeta_j^{(k)})$ is the random noise at iteration k. The visualisation of the Algorithm 1 is given in the Fig. 2.

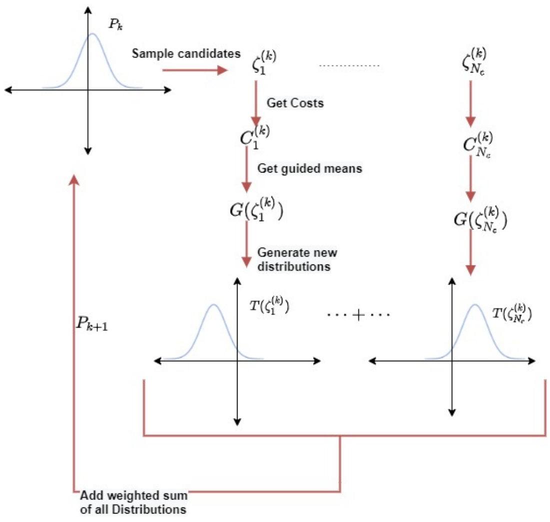

Figure 2: Visualization of the guidance mechanism in Algorithm 1.

3.4 Convergence rate analysis

We now prove that the sequence generated by ZO-iRL is guaranteed to converge to an optimal solution and show the non-asymptotic convergence rate. We adopt the following notations: Z is a compact metric space; $\mathcal{B}$ is the $\sigma-$algebra of Borel-subsets of Z; and $\zeta \in Z^n$ is the collection of parameters to be learned, where n is its ambient dimension.

For the convergence analysis, we explicitly state the following assumptions:

1. The disturbance $w_k(\zeta)$ is a random variable with zero mean distribution $F(\zeta, dw)$ that is concentrated within a finite interval $[-d, d]$.
2. The random variables $w_k(\zeta)$ are independent for all candidate k and parameter $\zeta \in Z$
3. The transition probability function is given by $T(\zeta_j^{(k)}, d\zeta) = \kappa(\zeta_j^{(k)}, \zeta_j^{(k+1)})\mu(d\zeta)$, and is always bounded, with μ a probability measure on $(Z, \mathcal{B})$:
$$\sup_{\zeta_j^{(k)}, \zeta_j^{(k+1)} \in Z} \kappa(\zeta_j^{(k)}, \zeta_j^{(k+1)}) \leq \Lambda < \infty$$
4. There exists a permutation order ν such that the magnitude of difference between the generated value $\zeta_{\nu(j)}^{(k+1)}$ from the guided value of the previous sample ζ_j^k is bounded by twice of the guidance factor ρ using the following equation:
$$|\zeta_{\nu(j)}^{k+1} - G(\zeta_j^k)| \leq 2\rho$$
5. There exists a global optimal solution $\zeta^\star$, and there exists an $\epsilon > 0$, such that the underlying true objective function f_{true} is continuous in the set $B(\zeta^\star, \epsilon)$.

We note that Assumption 5 is a much more relaxed condition than requiring differentiability (Agrawal et al., 2020). The following theorem shows the convergence rate of generated distributions $P_k(d\zeta)$ to a stationary distribution S_{N_c}.

Theorem 3.1. *Under the assumptions listed above:*

1. For any number of candidates N_c, the random elements $a_k = (\zeta_1^{(k)}, \cdots, \zeta_{N_c}^{(k)})$ form a homogeneous Markov chain with a stationary distribution $S_{N_c}(d\zeta_1, \cdots, d\zeta_{N_c})$.

2. $P_k(d\zeta)$ converges to S_{N_c} with a geometric rate that depends on N_c, guidance factor ρ, and variance σ^2 as follows:

$$\sup_{B \in \mathcal{B}_N} \left| P_k(B) - S_{N_c}(B) \right| \leq (1 - c_2)^{k-1} \tag{11}$$

where $c_2 = \left[\frac{N_c c_1 e^{-2\rho^2/\sigma^2}}{c_1 + (N_c - 1)\exp[m_f + d]\sigma\sqrt{2\pi}} \right]^{N_c}$ with m_f denote the maximum value of f_{true} within a compact set of parameters, and $c_1 = \inf \mathbb{E}[C_t(\zeta)]$, and given $0 < c_2 < 1$

Detailed proof of Theorem 3.1 can be found in the supplementary material.

Deciding the value of the guidance factor: For the result of Theorem 3.1 to be valid, we need to have $0 < c_2 < 1$. The introduction of the guidance factor ρ can ensure that c_2 always stays between 0 and 1. Moreover, a good choice of ρ can make c_2 closer to 1, which, by our analysis, may lead to a faster convergence of $P_k(d\zeta)$ to the stationary distribution S_{N_c}.

To keep $0 < c_2 < 1$, we need limit the following ratio::

$$\frac{N_c c_1 e^{-2\rho^2/\sigma^2}}{c_1 + (N_c - 1)exp[m_f + d]\sigma\sqrt{2\pi}} \in (0, 1) \tag{12}$$

We propose that to have a fast convergence rate, a reasonable value for k_1 should be between 0.6 and 1. This implies:

$$0.6 < \frac{e^{-2\rho^2/\sigma^2}}{\sigma\sqrt{2\pi}} k_1 < 1, \tag{13}$$

where

$$k_1 = \frac{N_c c_1}{c_1 + (N - 1)exp[m_f + d]}. \tag{14}$$

Based on (13), a reasonabale range for the guidance factor can be:

$$\frac{\sigma}{\sqrt{2}}\sqrt{ln(k_1) - ln(\sigma\sqrt{2\pi})} < \rho < \frac{\sigma}{\sqrt{2}}\sqrt{ln(k_1) - ln(0.6\sigma\sqrt{2\pi})} \quad (15)$$

While the exact determination of the relevant parameters may be difficult in some situations, the above inequality provides some quantitative relation of a proper value for the guidance factor with respect to the noise variance and other properties of the problem.

4 Experiments

We consider two applications to validate the proposed ZO-iRL algorithm. Source code can be found in the supplementary material, and will be made available to the public upon publication.

4.1 Profit maximization for a network of supply chains

Problem Definition: As presented in (Agrawal et al., 2020), the goal is to find a set of policies for a network of supply chains that maximizes profit. We note that the problem setting is simpler than the present study, because 1) the dynamics and future states are assumed to be known, and 2) only a one-time decision is required, obviating the need of a lookahead model.

A network of supply chains contains n interconnected warehouses, m links over which goods can flow, k links that connect suppliers to warehouses, c links that connect warehouses to customers, with the rest of the $m-k-c$ links represent the internode connections.

Let $h_t \in \mathbb{R}_+^n$ denote the amount of goods held at each node at time t, $p_t \in \mathbb{R}_+^k$ denotes the price at which the warehouses can buy from suppliers at time t, $r \in \mathbb{R}_+^c$ characterize the fixed process for sale to consumers, $d_t \in \mathbb{R}_+^c$ be the uncertain consumer demand at time t. The decision variables include: 1) $b_t \in \mathbb{R}_+^k$ (the quantity to buy from suppliers); 2) $s_t \in \mathbb{R}_+^c$ (the quantity to be sold to to the customers); and 3) $z_t \in \mathbb{R}_+^{m-k-c}$ (the quantity to be shipped across internode links). The holding costs for goods is given by $\alpha^T h_t + \beta^T h_t^2$, where $\alpha, \beta \in \mathbb{R}_+^n$.

The true cost to minimize can be accessed directly without additional noise, and is given by:

$$\frac{1}{T}\sum_{t=0}^{T-1} p_t^T b_t - r^T s_t + \tau^T z_t + \alpha^T h_t + \beta^T (h_t^T . h_t) \quad (16)$$

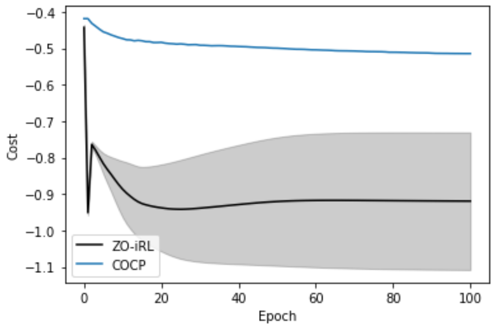

Figure 3: Comparison of learning curves in the setting of noiseless cost evaluation.

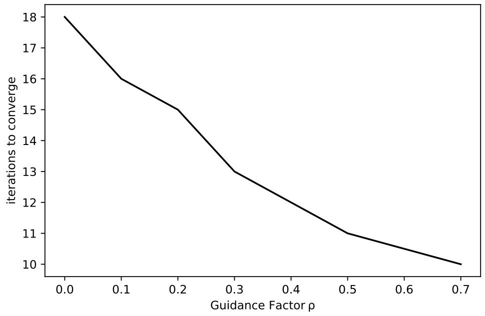

Figure 4: Number of iterations to converge with respect to the guidance factor in the noiseless cost setting.

The overall strategy is to formulate an implicit convex optimization problem, parametrized by parameters $\zeta = (P, q)$, where $P \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^n$. Here P and q are meant to capture the effect of randomness from α and β that appear inside the original cost function. The RL agent is tasked learn the best parameters resulting in the maximum profit. More details on the problem setup can be found in the supplementary material. In the following, we consider a setup with $n = 4$ nodes, $m = 8$ links, $k = 2$ supply links, and $c = 2$ customer links.

Noiseless cost evaluation: We use the same cost function as given in (Agrawal et al., 2020) with constant α and β. We run ZO-iRL for 100 iterations with a guidance factor $\rho = 0.1$, and repeat the experiment 10 times. Results show that our algorithm achieves on average 109% improvement over the baseline cost, significantly better than the 24% improvement achieved by (Agrawal) (see Fig. 3 for the fast convergence).

The effect of the guidance factor is examined in Fig. 4. The relation is clear: a higher guidance factor leads to faster convergence to an optimal solution when no noise is present in the cost function evaluation.

Noisy cost evaluations: We introduced noise in the

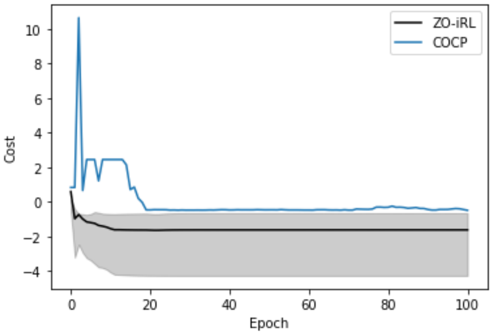

Figure 5: Comparison of learning curves in the setting of noisy cost evaluation.

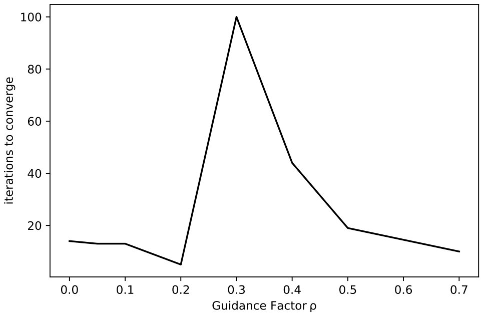

Figure 6: Number of iterations to convergence with respect to the guidance factor in the noisy cost setting.

cost function evaluation by introducing randomness in the parameters α and β to make the setup closer to the real world scenarios. We ran our method again for 100 iterations with a guidance a factor $\rho = 0.5$. This setup was run 10 times and our method not only improved the baseline cost from 0.83 to an average of -2.455 but also exhibits more stable learning as compared to COCP's experiments. (Agrawal et al., 2020) approach was only able to improve the baseline cost only from 0.83 to -0.483, and also exhibits unstable behaviors in the presence of noise.

The effect of guidance is shown in Fig. 6. In this case, there is no simple relation between the guidance factor and convergence rate, consistent with our theoretical analysis. Indeed, as shown in (15), the best range of rho depends on a host of variables. It appears that a reliable choice of the guidance factor should be between 0.5 and 0.7.

4.2 CityLearn Challenge

Problem definition: The CityLearn Challenge is an annual competition aimed to inspire RL solutions to address grand challenges for the control of power grids (Vazquez-Canteli et al., 2020a). This competition is motivated by the pressing needs for climate change mitigation and adaptation by enhancing energy flexibility and resilience in the face of climate-induced demand surge (as already observed in places like California, where rolling blackouts are increasingly frequent during the Summer) and natural adversity (such as the disastrous winter storm in Texas and extreme heat waves in India and Iran this year). It provides a set of benchmarks for evaluating AI-based methods in coordinating distributed energy resources for intelligent energy management.

The competition has an online setup with only one episode of the entire 4 years, when agents will exploit the best policies to optimize the coordination strategy. Notably, CityLearn encompasses all the challenges discussed in the Introduction section, including *1)* the ability to learn on live systems from limited samples—there is no training period; *2)* learning and acting in high-dimensional state and action spaces—the RL agents need to coordinate 9 individual buildings; *3)* dealing with system constraints that should never or rarely be violated—there are strict balancing equations for electricity, heating, and cooling energy; *4)* interacting with systems that are partially observable—it is a distributed control setup; *5)* learning from multiple objectives—the evaluation metrics include ramping cost, peak demands, 1-load factor, carbon emissions; *6)* the ability to provide actions quickly—there is a strict time limit for completing the 4 year evaluation on Google's Colab; and *7)* providing system operators with explainable policies—a necessity to facilitate real-world adoption and deployment. For more information on the simulation setup, we refer to (Vazquez-Canteli et al., 2020a)

To model the implicit convex optimization layer, we first construct a surrogate convex cost function to promote desirable behaviors such as load flattening and smoothing. As a proxy for the competition evaluation metrics, the surrogate cost function at time t is given by:

$$C_{t,i} = |E_t^{\text{grid}} - E_{t-1}^{\text{grid}}| + p_t^{\text{ele}} E_t^{\text{grid}} + \sum_{t'=t+1}^{T} \left(|E_{t'}^{\text{grid}} - E_{t'-1}^{\text{grid}}| + p_{t'}^{\text{ele}} E_{t'}^{\text{grid}} \right) \tag{17}$$

where E_t^{grid} is the electricity demand for a building at time t. The RL agents aim is to learn the parameter $p_{\text{ele}} \in \mathbb{R}^{24}$ (separately for each of the 9 buildings) that represents the virtual electricity cost. The constraints of the optimization model include energy balance equations and technological constraints, and can be found in the supplementary material. We model

Method	Climate Zone	Ramping	1-Load Factor	Avg. Daily Peak	Peak Demand	Net Elec. Consumption	Avg. Score
ZO-iRL	1	0.833	1.010	0.986	0.953	1.002	0.964
	2	0.784	1.025	0.962	0.961	1.001	0.956
	3	0.822	1.048	0.989	0.955	1.001	0.969
	4	**0.743**	0.990	0.974	1.006	1.002	0.953
	5	**0.711**	0.999	0.9691	0.939	1.004	0.924
						Average Score	**0.953**
SAC	1	2.470	1.202	1.354	1.209	1.049	1.390
	2	2.413	1.183	1.349	1.152	1.056	1.369
	3	2.609	1.1185	1.382	1.313	1.056	1.435
	4	2.512	1.168	1.376	1.207	1.057	1.397
	5	1.614	1.115	1.133	1.159	1.015	1.177
						Average Score	**1.353**
Random Agent	1	1.071	1.130	1.168	1.077	0.993	1.073
	2	1.045	1.138	1.151	1.079	0.987	1.066
	3	1.032	1.131	1.158	1.180	0.991	1.081
	4	0.965	1.101	1.114	1.134	0.984	1.048
	5	1.015	1.138	1.116	1.089	0.987	1.057
						Average Score	**1.065**
MARLISA	1	1.02	1.019	1.015	1.0	1.0	1.009
	2	1.008	1.02	1.012	1.0	0.998	1.006
	3	1.002	1.017	1.01	1.0	0.999	1.005
	4	1.002	1.029	1.014	1.0	0.998	1.007
	5	1.39	1.105	1.103	1.205	1.001	1.136
						Average Score	**1.032**

Table 1: Scores for ZO-iRL and comparison methods, including SAC (Kathirgamanathan et al., 2020) and MARLISA (Vazquez-Canteli et al., 2020b). The random agent basically uniformly selects an action within the range at each timestep.

this problem using the lookahead model, where a set of predictors are designed to predict future states based on the past 2 weeks data. The competition evaluates performance of RL agents by computing the ratio of costs with respect to a deterministic rule-based controller (RBC). Therefore, a lower ratio indicates a better performance.

Table 1 shows the scores of our ZO-iRL agent compared to existing baseline. Note that MARLISA is tailor-designed for the competition Vazquez-Canteli et al. (2020b). All agents are validated on data from 5 different climate zones. It can be observed that ZO-iRL has achieved the lowest cost ratios (i.e. the best scores) as compared to baseline methods. In particular, baseline RL methods struggle to learn a reasonable policy within the limited 4-year test period, while ZO-iRL is able to quickly find a good policy within the first few months (see Fig. 7). Furthermore, the learning progress can be intuitively explained by inspecting the evolution of parameter p_{ele} as shown in Fig. 8. In this particular case, the building tends to overcharge its storage in the early morning, which results in unexpected electricity peaks that is undesirable; by increasing the virtual prices during that period, the agent is able to find a better strategy that smoothes the peaks, thus resulting in better performance. More details can be found in the supplementary material.

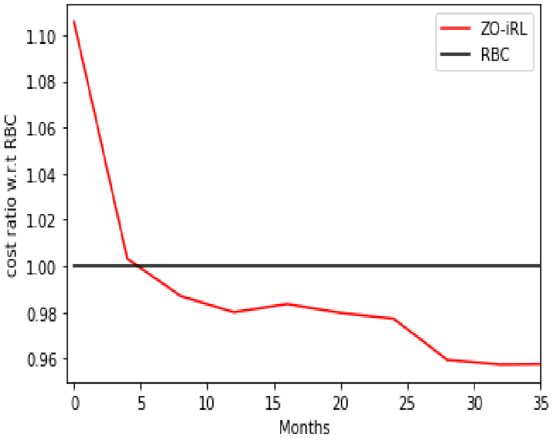

Figure 7: Learning curve of ZO-iRL, where RBC takes a baseline cost of constant 1.

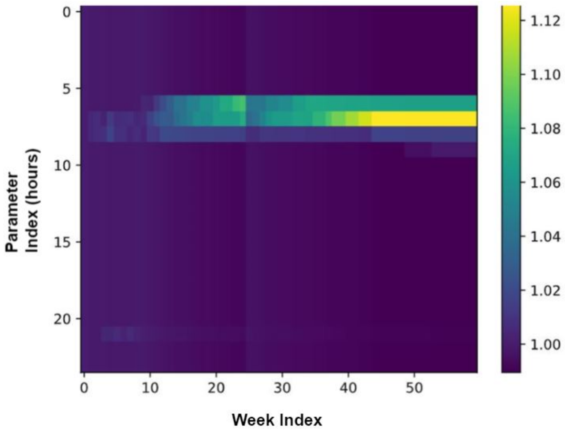

Figure 8: Evolution of the implicit parameters p_{ele} over the test period, where the values are color-coded.

5 Conclusion and future directions

The present work introduces a novel implicit RL framework for model-free decision making in DCSs. By leveraging the synergistic strength of convex optimization and RL, the proposed method is able to simultaneously address a range of challenges for real-world RL. Our work opens up exciting research directions for future works, including the extension of the implicit RL framework to other derivative-free methods such as Bayesian optimization or first-order methods such as actor-critic RL.

Acknowledgements

The authors would like to thank the organizers of the CityLearn Challenge (Dr. José R. Vázquez-Canteli, Dr. Zoltan Nagy, Dr. Gregor Henze, and Sourav Dey) for motivating the proposed approach.

References

Agrawal, A., Barratt, S., Boyd, S., and Stellato, B. (2020). Learning convex optimization control policies. In *Learning for Dynamics and Control*, pages 361–373. PMLR.

Ames, A. D., Xu, X., Grizzle, J. W., and Tabuada, P. (2016). Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876.

Ammanabrolu, P. and Hausknecht, M. (2020). Graph constrained reinforcement learning for natural language action spaces. *arXiv preprint arXiv:2001.08837.*

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. (2016). Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565.*

Berkenkamp, F., Turchetta, M., Schoellig, A. P., and Krause, A. (2017). Safe model-based reinforcement learning with stability guarantees. *arXiv preprint arXiv:1705.08551.*

Boyd, S., Boyd, S. P., and Vandenberghe, L. (2004). *Convex optimization.* Cambridge university press.

Carpentier, P.-L., Gendreau, M., and Bastin, F. (2014). Managing hydroelectric reservoirs over an extended horizon using benders decomposition with a memory loss assumption. *IEEE Transactions on Power Systems*, 30(2):563–572.

Cheng, R., Orosz, G., Murray, R. M., and Burdick, J. W. (2019). End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3387–3395.

Choi, J., Castaneda, F., Tomlin, C. J., and Sreenath, K. (2020). Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions. *arXiv preprint arXiv:2004.07584.*

Chow, Y., Nachum, O., Duenez-Guzman, E., and Ghavamzadeh, M. (2018). A lyapunov-based approach to safe reinforcement learning. *arXiv preprint arXiv:1805.07708.*

Dean, S., Taylor, A. J., Cosner, R. K., Recht, B., and Ames, A. D. (2020). Guaranteeing safety of learned perception modules via measurement-robust control barrier functions. *arXiv preprint arXiv:2010.16001.*

Donti, P. L., Roderick, M., Fazlyab, M., and Kolter, J. Z. (2021). Enforcing robust control guarantees within neural network policies. In *International Conference on Learning Representations.*

Dulac-Arnold, G., Levine, N., Mankowitz, D. J., Li, J., Paduraru, C., Gowal, S., and Hester, T. (2021). Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, pages 1–50.

Ebert, F., Finn, C., Dasari, S., Xie, A., Lee, A., and Levine, S. (2018). Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568.*

Fisac, J. F., Akametalu, A. K., Zeilinger, M. N., Kaynama, S., Gillula, J., and Tomlin, C. J. (2018). A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7):2737–2752.

Ghadimi, S., Perkins, R. T., and Powell, W. B. (2020). Reinforcement learning via parametric cost function approximation for multistage stochastic programming. *arXiv preprint arXiv:2001.00831.*

Gillula, J. H. and Tomlin, C. J. (2012). Guaranteed safe online learning via reachability: tracking a ground target using a quadrotor. In *2012 IEEE International Conference on Robotics and Automation*, pages 2723–2730. IEEE.

Gu, F., Yin, H., Ghaoui, L. E., Arcak, M., Seiler, P., and Jin, M. (2021). Recurrent neural network controllers synthesis with stability guarantees for partially observed systems. *arXiv preprint arXiv:2109.03861.*

Gu, S., Holly, E., Lillicrap, T., and Levine, S. (2017). Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3389–3396. IEEE.

Hewing, L., Wabersich, K. P., Menner, M., and Zeilinger, M. N. (2020). Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:269–296.

Jin, M. and Lavaei, J. (2020). Stability-certified reinforcement learning: A control-theoretic perspective. *IEEE Access*, 8:229086–229100.

Jin, S., Ryan, S. M., Watson, J.-P., and Woodruff, D. L. (2011). Modeling and solving a large-scale generation expansion planning problem under uncertainty. *Energy Systems*, 2(3):209–242.

Kathirgamanathan, A., Twardowski, K., Mangina, E., and Finn, D. P. (2020). A centralised soft actor critic deep reinforcement learning approach to district demand side management through citylearn. In *Proceedings of the 1st International Workshop on Reinforcement Learning for Energy Management in Buildings & Cities*, pages 11–14.

Khattar, V. (2021). Threat assessment and proactive decision-making for crash avoidance in autonomous vehicles.

Lium, A.-G., Crainic, T. G., and Wallace, S. W. (2009). A study of demand stochasticity in service network design. *Transportation Science*, 43(2):144–157.

Nozhati, S., Ellingwood, B. R., and Chong, E. K. (2020). Stochastic optimal control methodologies in risk-informed community resilience planning. *Structural Safety*, 84:101920.

Perkins, T. J. and Barto, A. G. (2002). Lyapunov design for safe reinforcement learning. *Journal of Machine Learning Research*, 3(Dec):803–832.

Powell, W. B. and Meisel, S. (2015). Tutorial on stochastic optimization in energy—part i: Modeling and policies. *IEEE Transactions on Power Systems*, 31(2):1459–1467.

Prakash, B., Waytowich, N., Ganesan, A., Oates, T., and Mohsenin, T. (2020). Guiding safe reinforcement learning policies using structured language constraints. *UMBC Student Collection.*

Qin, Z., Zhang, K., Chen, Y., Chen, J., and Fan, C. (2021). Learning safe multi-agent control with decentralized neural barrier certificates. *arXiv preprint arXiv:2101.05436.*

Stoica, I., Song, D., Popa, R. A., Patterson, D., Mahoney, M. W., Katz, R., Joseph, A. D., Jordan, M., Hellerstein, J. M., Gonzalez, J. E., et al. (2017). A berkeley view of systems challenges for ai. *arXiv preprint arXiv:1712.05855.*

Stooke, A., Achiam, J., and Abbeel, P. (2020). Responsive safety in reinforcement learning by pid lagrangian methods. In *International Conference on Machine Learning*, pages 9133–9143. PMLR.

Vazquez-Canteli, J. R., Dey, S., Henze, G., and Nagy, Z. (2020a). Citylearn: Standardizing research in multi-agent reinforcement learning for demand response and urban energy management. *arXiv preprint arXiv:2012.10504.*

Vazquez-Canteli, J. R., Henze, G., and Nagy, Z. (2020b). Marlisa: Multi-agent reinforcement learning with iterative sequential action selection for load shaping of grid-interactive connected buildings. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 170–179.

Wilson, S., Glotfelter, P., Wang, L., Mayya, S., Notomista, G., Mote, M., and Egerstedt, M. (2020). The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems. *IEEE Control Systems Magazine*, 40(1):26–44.

Yin, H., Seiler, P., Jin, M., and Arcak, M. (2021). Imitation learning with stability and safety guarantees. *IEEE Control Systems Letters.*

Zhang, K., Yang, Z., and Başar, T. (2021). Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384.

Zhigljavsky, A. A. (2012). *Theory of global random search*, volume 65. Springer Science & Business Media.

Zhou, Q., Shahidehpour, M., Paaso, A., Bahramirad, S., Alabdulwahab, A., and Abusorrah, A. (2020). Distributed control and communication strategies in networked microgrids. *IEEE Communications Surveys & Tutorials*, 22(4):2586–2633.

Supplementary Material

1 Convergence analysis

We prove a more general version of Theorem 3.1 in the main text that allows the random noises added to each dimension of the variables to have different standard deviations and inter-parameter correlations. To this end, we consider a transition function $T(\zeta_j^{(k)}, d\zeta) = \mathcal{N}(G(\zeta_j^{(k)}), \sigma^2)\mu(d\zeta)$ with a non-singular covariance matrix $\Sigma \in^{l \times l}$ (equation (9) in Algorithm 1). The theorem is stated as follows.

Theorem 3.1b. *Under the assumptions listed in the main text (Section 3.4):*

1. For any number of candidates N_c, the random elements $a_k = (\zeta_1^{(k)}, \cdots, \zeta_{N_c}^{(k)})$ form a homogeneous Markov chain with a stationary distribution $S_{N_c}(d\zeta_1, \cdots, d\zeta_{N_c})$ with $\zeta_i \in \mathbb{R}^l$.

2. $P_k(d\zeta)$ converges to S_{N_c} with a geometric rate that depends on N_c, guidance factor ρ, and covariance matrix $\Sigma \in \mathbb{R}^{l \times l}$ as follows:

$$\sup_{B \in \mathcal{B}_N} \left| P_k(B) - S_{N_c}(B) \right| \leq (1 - c_2)^{k-1} \tag{1}$$

where $c_2 = \left[\frac{N_c c_1 e^{-2\rho^2/\|\Sigma\|^2}}{c_1 + (N_c - 1)\exp[m_f + d]\sqrt{(2\pi)^l \det(\Sigma)}} \right]^{N_c}$ with m_f denote the maximum value of f_{true} within a compact set of parameters, and $c_1 = \inf\ \mathbb{E}[C_t(\zeta)]$, and given $0 < c_2 < 1$.

Proof. We will first prove that sampling N_c candidate parameters using the ZO-iRL algorithm (see Algorithm 1 in the text) can be considered as sampling from a homogeneous Markov chain in the space Z^n. Denote the samples from Z^n at iteration k by

$$a_k := (\zeta_1^{(k)}, \cdots, \zeta_{N_c}^{(k)}). \tag{2}$$

The initial distribution of the Markov chain is given by

$$R_1(da) = P_1(d\zeta_1) \cdots P_1(d\zeta_{N_c}). \tag{3}$$

The transition probability of the Markov chain for the sampling process is given by

$$Q(a_{k-1}, da_k) = \int_{-d}^{d} \cdots \int_{-d}^{d} F(\zeta_1^{(k-1)}, dw_1) \cdots F(\zeta_{N_c}^{k-1}, dw_N) \prod_{j=1}^{N_c} \sum_{i=1}^{N_c} \frac{\exp[f_{\text{true}}(\zeta_i^{(k-1)}) + w_i]}{\sum_{l=1}^{N_c} \exp[f_{\text{true}}(\zeta_l^{(k-1)}) + w_l]} T(\zeta_i^{(k-1)}, d\zeta_j), \tag{4}$$

where $F(\zeta, dw)$ is the zero-mean distribution for random variable $w_k(\zeta)$ that is concentrated within a finite interval $[-d, d]$. Since the transition probability $T(\zeta_i^{(k-1)}, d\zeta_j)$ is Markovian (by design), the derived transition probability $Q(a_{k-1}, a_k)$ is Markovian. In particular, we can choose the Gaussian distribution:

$$T(\zeta_i^{(k-1)}, d\zeta_j) = \frac{\exp\left[\frac{-1}{2} \left((\zeta^{(k)} - G(\zeta_i^{(k-1)}))^T \Sigma^{-1} (\zeta_{(k)} - G(\zeta_i^{(k-1)}) \right) \right]}{\sqrt{(2\pi)^l \det(\Sigma)}} \tag{5}$$

Now we will show that the distributions $R_k(da)$ recursively defined by

$$R_{k+1}(da_{k+1}) = \int_{Z^n} R_k(da_k) \prod_{i=1}^{N_c} T(\zeta_i^{(k)}, d\zeta) \tag{6}$$

will converge to a stationary distribution as $k \to \infty$.

In our setting, we can only observe noisy cost samples (instead of the true cost values). It follows from (Loeve, 1963, Sec. 29.4) that the random variable for any m defined as

$$\alpha_m := \frac{\sum_{i=1}^m C_t(\zeta_i)}{m} \tag{7}$$

will converge for $m \to \infty$ to a random variable α that is independent of α_m and $C_t(\zeta_i)$, such that

$$\mathbb{E}[\alpha] = \mathbb{E}[C_t(\zeta_i)]. \tag{8}$$

Now, we proceed to lower bound the transition probability:

$$\begin{aligned}
Q(a_{k-1}, da_k) &\geq \prod_{j=1}^{N_c} \sum_{i=1}^{N_c} \frac{c_1}{c_1 + (N_c - 1)(\exp[\max(f_{\text{true}}) + d])} T(\zeta_i^{(k-1)}, d\zeta_j) \\
&\geq \prod_{j=1}^{N_c} \sum_{i=1}^{N_c} \frac{N_c c_1}{c_1 + (N_c - 1)(\exp[m_f + d])} \frac{\exp\left[\frac{-1}{2}\Big((\zeta^{(k)} - G(\zeta_i^{(k-1)}))^T \Sigma^{-1} (\zeta_{(k)} - G(\zeta_i^{(k-1)})\Big)\right]}{\sqrt{(2\pi)^l \det(\Sigma)}} \\
&\geq \prod_{j=1}^{N_c} \frac{N_c c_1}{c_1 + (N_c - 1)(\exp[m_f + d])} \frac{e^{-2\rho^2/\|\Sigma\|^2}}{\sqrt{(2\pi)^l \det(\Sigma)}} \\
&= c_2 \tilde{\mu}(da_k)
\end{aligned}$$

where $\|\Sigma\|$ is the spectral norm of Σ, $c_1 = \inf \mathbb{E}[C_t(\zeta_i)]$, $\tilde{\mu}(da_k) = \mu(d\zeta_1^{(k)}) \cdots \mu(d\zeta_{N_c}^{(k)})$ is the correspondingly defined probability measure on (Z^n, B_{N_c}), m_f denotes the maximum value of f_{true} within a compact set of parameters and

$$c_2 = \left[\frac{N_c c_1 e^{-2\rho^2/\|\Sigma\|^2}}{c_1 + (N_c - 1) exp[m_f + d] \sqrt{(2\pi)^l \det(\Sigma)}}\right]^{N_c}. \tag{9}$$

By the results in (Kendall, 1966, Section V.3), we have the following result for the transition probability of the sampled Markov chain:

$$\Delta_0 = \sup_{a_{k-1}, a_k \in Z^n} \underset{B \in \mathcal{B}_{N c}}{sup} [Q(a_{k-1}, B) - Q(a_k, B)] \leq 1 - c_2. \tag{10}$$

Using the exponential convergence criterion (Loeve, 1963), we can conclude that the distributions $R_k(da_k)$ will converge to a stationary distribution $S_{N_c}(da_k)$ as $k \to \infty$, where S_{N_c} is defined as the unique positive solution of the following equation:

$$S_{N_c}(da_{k-1}) = \int_{Z^n} S_{N_c}(da_k) Q(a_{k-1}, da_k) \tag{11}$$

Moreover, by (Loeve, 1963) we have the following inequalities :

$$\sup_{B \in \mathcal{B}_{N_c}} \left| P_k(B) - S_{N_c}(B) \right| \leq \Delta_0^{k-1} \leq (1 - c_2)^{k-1}. \tag{12}$$

Thus, the proof is completed. Note that if we specify the covariance matrix to be a diagonal matrix with identical diagonal entries, then the result is reduced to Theorem 3.1 in the text.

■

Remarks on theoretical contributions: The proposed algorithm bears resemblance with some existing zeroth-order methods, such as cross entropy method (CEM) and evolutionary algorithms (EA); however, most convergence results for these algorithms only show asymptotic convergence to an optimal solution with high probability. In particular, to the best of the authors' knowledge, the non-asymptotic convergence rate has not been derived for CEM. By contrast, in the present work, we prove that the algorithm yields a convergent sequence with the property that the probability distributions P_k will converge to a stationary distribution S_{N_c} at a geometric rate. Some of the recent works have explored the average rate of convergence for EA (Chen and He, 2021). However, their theoretical analysis is based on a martingale-type argument, which only applies to the case where there is no noise in the cost function evaluations. Applicable to a more general setting, our result is able to deal with the noise in cost function evaluation. Moreover, our convergence result is able to account for the guidance factor.

2 Additional details for experiments

2.1 Profit maximization for a network of supply chains

Denote $h_t \in \mathbb{R}_+^n$ as the amount of goods held at each node at time t, $p_t \in \mathbb{R}_+^k$ as the price at which the warehouses can buy from suppliers at time t, $r \in \mathbb{R}_+^c$ as the fixed process for sales to consumers, $d_t \in \mathbb{R}_+^c$ as the uncertain consumer demand at time t. The decision variables include: 1) $b_t \in \mathbb{R}_+^k$ (the quantity to buy from suppliers); 2) $s_t \in \mathbb{R}_+^c$ (the quantity to be sold to the customers); and 3) $z_t \in \mathbb{R}_+^{m-k-c}$ (the quantity to be shipped across internode links). The holding costs for goods is given by $\alpha^T h_t + \beta^T \|h_t\|^2$, where $\alpha, \beta \in \mathbb{R}_+^n$.

The system dynamics are assumed to be known and given by

$$h_{t+1} = h_t + (A^{\text{in}} - A^{\text{out}}) a_t, \tag{13}$$

where $A^{\text{in}} \in \mathbb{R}^{n \times m}$ and $A^{\text{out}} \in \mathbb{R}^{n \times m}$ are known matrices, and a_t is a concatenation of vector-valued decision variables including b_t, s_t, and z_t.

The following constraints are imposed as part of the supply chain management:

$$0 \leq h_t \leq h_{\max} \tag{14a}$$
$$0 \leq b_t \leq b_{\max} \tag{14b}$$
$$0 \leq s_t \leq s_{\max} \tag{14c}$$
$$0 \leq z_t \leq z_{\max} \tag{14d}$$
$$A^{\text{out}} u_t \leq h_t \tag{14e}$$
$$s_t \leq d_t \tag{14f}$$

The uncertainty in future demand and supplier prices are modeled using the log-normal distribution:

$$\log w_t := (\log p_{t+1}, \log d_{t+1}) \sim \mathcal{N}(\mu, \Sigma). \tag{15}$$

The true cost to minimize is given by

$$\frac{1}{T} \sum_{t=0}^{T-1} p_t^T b_t - r^T s_t + \tau^T z_t + \alpha^T h_t + \beta^T \|h_t\|^2. \tag{16}$$

The implicit convex optimization layer is parametrized by $\zeta = (P, q)$ for some $P \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^n$:

$$\begin{aligned}
\min_{a_t=(b_t,s_t,z_t)} \quad & p_t^T b_t - r^T s_t + \tau^T z_t - ||Ph_t||_2^2 - q^T h_t \\
\text{s.t.} \quad & h_{\mathrm{t+1}} = h_t + (A^{\mathrm{in}} - A^{\mathrm{out}})a_t \\
& 0 \le h_t \le h_{\max} \\
& 0 \le b_t \le b_{\max} \\
& 0 \le s_t \le s_{\max} \\
& 0 \le z_t \le z_{\max}
\end{aligned} \tag{17}$$

Experimental setup: Following (Agrawal et al., 2020), the initial values h_t for the network are sampled from a uniform distribution between 0 and $h_{\max}$, where $h_{\max}$ is set to be 3. The maximum for each decision variable is set to be 2. The mean and covariance for the customer demands and the supplier prices are given by

$$\mu = (0, 0.1, 0, 0.4), \qquad \Sigma = 0.04I \tag{18}$$

In the *noiseless cost evaluation* setup, we treat the storage cost parameters α and β as known constants with values of 0.01. In the *noisy cost evaluation* setup, we consider the storage cost parameters α and β as random variables, with values sampled from a standard normal distribution $\mathcal{N}(0, 1)$.

2.2 CityLearn challenge

We refer the reader to (Vazquez-Canteli et al., 2020a) and the corresponding online documentation[1] for the detailed setup of the competition. We will only focus on our strategy in this document. To build a lookahead model for the implicit convex optimization layer, we learn a set of predictors for solar generation and electricity/thermal demands. Prediction is done on a rolling-horizon basis for the next 24 hours using the past 2 weeks data. Denote the hour index by $r \in \{1, 2, \cdots, T\}$, where $T = 24$. Suppose that we are at the beginning of hour r. Then we need to plan for the action for the upcoming hour r (note that we need to plan for future hours in the process).

The decision variables for the implicit convex optimization layer at hour r include:

1. Net electricity grid import: E_t^{grid} for $T \ge t \ge r$
2. Electricity grid sell: E_t^{sell} for $T \ge t \ge r$
3. Heat pump electricity usage: E_t^{hpC} for $T \ge t \ge r$
4. Electric heater electricity usage: E_t^{ehH} for $T \ge t \ge r$
5. Electric battery state of charge: $\mathrm{SOC}_t^{\mathrm{bat}}$ for $T \ge t \ge r$
6. Electrical storage action: a_t^{bat} for $T \ge t \ge r$
7. Heat storage state of charge: $\mathrm{SOC}_t^{\mathrm{H}}$ for $T \ge t \ge r$
8. Heat storage action: a_t^{Hsto} for $T \ge t \ge r$
9. Cooling storage state of charge: $\mathrm{SOC}_t^{\mathrm{C}}$ for $T \ge t \ge r$
10. Cooling storage action: a_t^{Csto} for $T \ge t \ge r$

The learnable parameter for the implicit convex optimization layer corresponds to the virtual electricity price p_t^{ele} for $1 \le t \le T$. Technology parameters are as follows (we also specify their initializations):

Heat pump technology parameters:

[1] link: https://sites.google.com/view/citylearnchallenge (Accessed: Oct 21, 2021)

1. Technical efficiency $\eta^{\mathrm{hp}}_{\mathrm{tech}} = 0.22$
2. Target cooling temperature $t^{\mathrm{hp}}_{C} = 8$
3. Hourly COP of heat pump $\mathrm{COP}^{C}_{t} = \eta^{\mathrm{hp}}_{\mathrm{tech}} \frac{t^{\mathrm{hp}}_{c}+273.15}{\mathrm{temp}_t - t^{\mathrm{hp}}_{C}}$, where temp_t is the outside temperature for hour t for $1 \leq t \leq T$
4. Heat pump capacity $E^{\mathrm{hpc}}_{\mathrm{max}}$, estimated from data following a simple statistical procedure

Electric heater parameters:

1. Efficiency $\eta_{\mathrm{ehH}} = 0.9$
2. Capacity $E^{\mathrm{ehH}}_{\mathrm{max}}$, estimated based on a simple statistical procedure

Electric battery:

1. Rate of decay $Cf^{\mathrm{bat}} = 0.00001$
2. Capacity Cp^{bat}, estimated based on a simple statistical procedure
3. Efficiency $\eta^{\mathrm{bat}}_{t} = 1$

Heat storage:

1. Rate of decay $Cf^{\mathrm{Hsto}} = 0.0008$
2. Capacity Cp^{Hsto}, estimated based on a simple statistical procedure
3. Efficiency $\eta^{\mathrm{Hsto}}_{t} = 1$

Cooling storage:

1. Rate of decay $Cf^{\mathrm{Csto}} = 0.006$
2. Capacity Cp^{Csto}, estimated based on a simple statistical procedure
3. Efficiency $\eta^{\mathrm{Csto}}_{t} = 1$

The surrogate objective function is given by:

$$C_t(\{E^{\mathrm{grid}}_{t'}\}_{t' \geq t}) = |E^{\mathrm{grid}}_{t} - E^{\mathrm{grid}}_{t-1}| + p^{\mathrm{ele}}_{t} E^{\mathrm{grid}}_{t} + \sum_{t'=t+1}^{T} \left(|E^{\mathrm{grid}}_{t'} - E^{\mathrm{grid}}_{t'-1}| + p^{\mathrm{ele}}_{t'} E^{\mathrm{grid}}_{t'} \right) \tag{19}$$

The constraints include both energy balance constraints and technology constraints.
Energy balance constraints:

- Electricity balance for each hour $t \geq r$:
$E^{\mathrm{PV}}_{t} + E^{\mathrm{grid}}_{t} = E^{\mathrm{NS}}_{t} + E^{\mathrm{hpC}}_{t} + E^{\mathrm{ehH}}_{t} + a^{\mathrm{bat}}_{t} C^{\mathrm{bat}}_{p} + E^{\mathrm{sell}}_{t}$
- Heat balance for each hour $t \geq r$:
$E^{\mathrm{ehH}}_{t} = a^{\mathrm{Hsto}}_{t} C^{\mathrm{Hsto}}_{p} + H^{\mathrm{bd}}_{t}$
- Cooling balance for each hour $t \geq r$:
$E^{\mathrm{hpC}}_{t} \mathrm{COP}^{\mathrm{C}}_{t} = a^{\mathrm{Csto}}_{t} C^{\mathrm{Csto}}_{p} + C^{\mathrm{bd}}_{t}$

Heat pump constraints:

- Maximum cooling for each hour $t \geq r$:
  $E_t^{\text{hpC}} \leq E_{\max}^{\text{hpC}}$
- Minimum cooling for each hour $t \geq r$:
  $E_t^{\text{hpC}} \geq 0$

Electric heater constraints:

- Maximum limit for each hour $t \geq r$:
  $E_t^{\text{ehH}} \leq E_{\max}^{\text{ehH}}$
- Minimum limit for each hour $t \geq r$:
  $E_t^{\text{ehH}} \geq 0$

Electric battery constraints:

- Initial SOC:
  $SOC_r^{\text{bat}} = (1 - C_f^{\text{bat}} SOC_{r-1}^{\text{bat}}) + a_r^{\text{bat}} \eta^{\text{bat}}$
- SOC updates for each hour $t \geq r$:
  $SOC_t^{\text{bat}} = (1 - C_f^{\text{bat}}) SOC_{t-1}^{\text{bat}} + a_t^{\text{bat}} \eta^{\text{bat}}$
- SOC terminal condition (note that we set $c_{\text{bat}}^{\text{end}} = 0.1$):
  $SOC_T^{\text{bat}} = c_{\text{end}}^{\text{bat}}$
- Action limits for each hour $t \geq r$:
  $a_{\text{lb}}^{\text{bat}} \leq a_t^{\text{bat}} \leq a_{\text{ub}}^{\text{bat}}$
- Bounds of battery SOC or each hour $t \geq r$:
  $1 \geq SOC_t^{\text{bat}} \geq 0$

Heat Storage constraints:

- Initial SOC:
  $SOC_r^{\text{H}} = (1 - C_f^{\text{Hsto}} SOC_{r-1}^{\text{H}}) + a_r^{\text{Hsto}} \eta^{\text{Hsto}}$
- SOC updates for each hour $t \geq r$:
  $SOC_t^{\text{H}} = (1 - C_f^{\text{Hsto}}) SOC_{t-1}^{\text{H}} + a_t^{\text{Hsto}} \eta^{\text{Hsto}}$
- Action limits or each hour $t \geq r$:
  $a_{\text{lb}}^{\text{Hsto}} \leq a_t^{\text{Hsto}} \leq a_{\text{ub}}^{\text{Hsto}}$
- Bounds of battery SOC or each hour $t \geq r$:
  $1 \geq SOC_t^{\text{H}} \geq 0$

Cooling Storage constraints:

- Initial SOC:
  $SOC_r^{\text{C}} = (1 - C_f^{\text{Csto}} SOC_{r-1}^{\text{C}}) + a_r^{\text{Csto}} \eta^{\text{Csto}}$
- SOC updates for each hour $t \geq r$:
  $SOC_t^{\text{C}} = (1 - C_f^{\text{Csto}}) SOC_{t-1}^{\text{C}} + a_t^{\text{Csto}} \eta^{\text{Csto}}$
- Action limits or each hour $t \geq r$:
  $a_{\text{lb}}^{\text{Csto}} \leq a_t^{\text{Csto}} \leq a_{\text{ub}}^{\text{Csto}}$
- Bounds of battery SOC or each hour $t \geq r$:
  $1 \geq SOC_t^{\text{C}} \geq 0$

The above optimization can be formulated as a linear program and solved efficiently. For more implementation details, please refer to our code (submitted as supplementary materials).

3 Additional experimental results

3.1 Profit maximization for a network of supply chains

Here we provide the topological diagram for the flow of goods over a network of supply chains. These results are for the same experimental setup as in the main text. Figure 1 shows the flow of goods over the network for the untrained and trained policy for a cost function without noise. Similarly, Figure 2 shows the flow of goods over the network for trained and untrained policies. The colors denote the values of normalized h_t across the internode links.

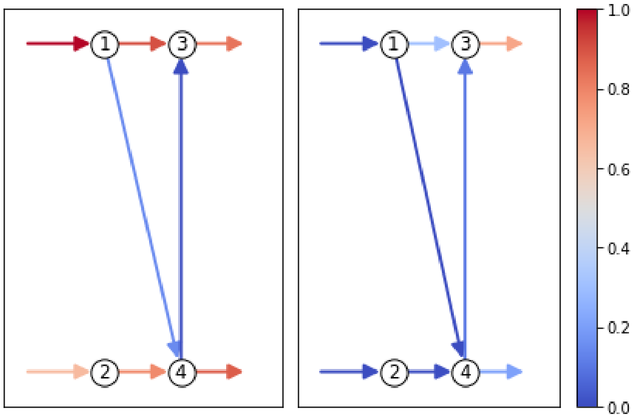

Figure 1: Topology of the supply chain considered in the main text. Left figure shows the initial untuned policy when no noise is considered in the cost function. Right figure shows the tuned policy after 100 iterations. The colors denote the normalized h_t across the internode links.

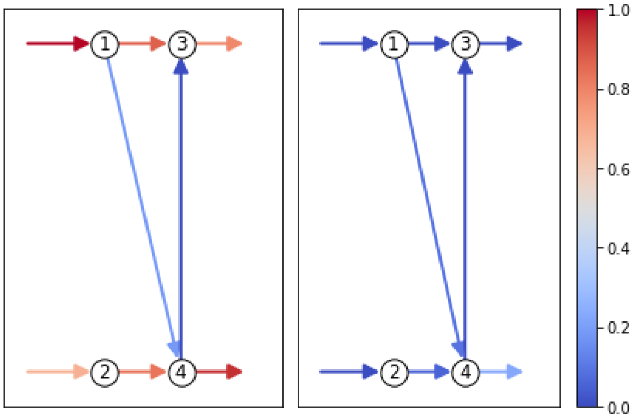

Figure 2: Topology of the supply chain considered in the main text. Left figure shows the initial untuned policy when noise is considered in the cost function. Right figure shows the tuned policy after 100 iterations. The colors denote the normalized h_t across the internode links.

3.2 CityLearn challenge

Here we provide the performance comparison ZO-iRL with different agents for 5 different climate zones. The best performance was observed in the reducing the ramping costs as shown in Figure 3.

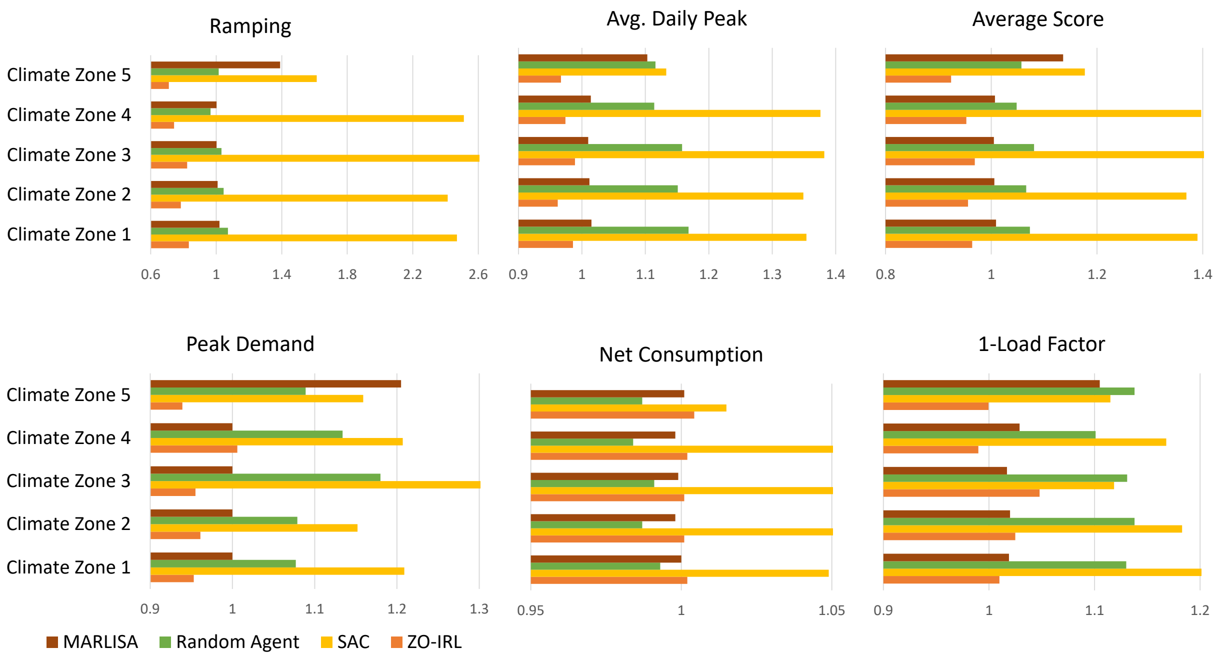

Figure 3: Scores for ZO-iRL and comparison with other methods, including SAC (Kathirgamanathan et al., 2020) and MARLISA (Vazquez-Canteli et al., 2020b) for different climate zones. The random agent basically uniformly selects an action within the range at each timestep.

3.3 Linear Quadratic Regulator (LQR)

As presented in (Agrawal et al., 2020), we consider a classical control LQR problem, where the cost function and dynamics are assumed to be known. The dynamics are given by:

$$f(x, u, w) = Ax + Bu + w, \tag{20}$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}$. The true cost to minimize can be accessed directly without additional noise, and is given by:

$$\frac{1}{T+1} \sum_{t=0}^{T} (x_t^T Q x_t + u_t^T R u_t), \tag{21}$$

where $Q \in \mathbb{R}^{nn}$, $R \in \mathbb{R}^{m \times m}$, and w is assumed to be sampled from a normal distribution: $w \sim \mathcal{N}(0, \Sigma)$.

The following implicit convex optimization layer is built parametrized by $\zeta = P$ for $P \in \mathbb{R}^{n \times n}$:

$$\min_{u} \quad (u^T R u + \|\zeta(Ax + Bu)\|^2) \tag{22}$$

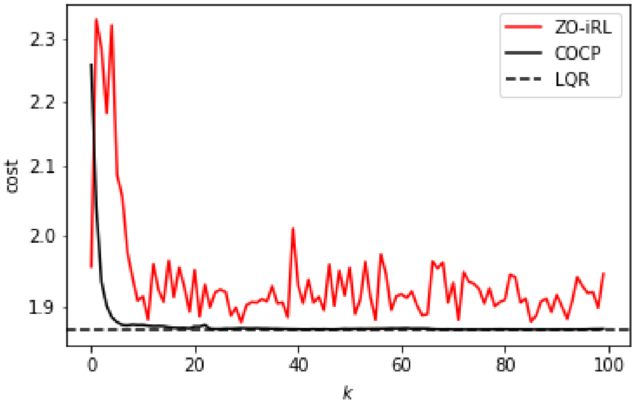

Figure 4: Comparison of learning curves in the setting of noiseless cost evaluation for tuning an LQR control policy

Experimental setup: $n = 4$ states, $m = 2$ inputs are considered with a time horizon $T = 100$. A and B are sampled from a standard normal distribution. The cost matrices Q and R are set to be the identity matrices. The noise covariance matrix $\Sigma = (0.25)I$ is considered. ζ is initialised with an identity matrix. We ran our algorithm and the COCP algorithm (Agrawal et al., 2020) for 100 iterations. We used a guidance factor $\rho = 0.8$ and $N_c = 14$ candidates.

Figure 4 shows the average cost during the learning phase for our method and the COCP versus the optimal LQR policy (i.e. for $T \to \infty$). Our algorithm was able to achieve a consistent cost with an upper and lower bound in 15 iterations, which is a little worse than the COCP method which converged in 10 iterations. This result could be attributed to the gradient-free random search which is usually less sample efficient than the gradient-based methods.

References

Agrawal, A., Barratt, S., Boyd, S., and Stellato, B. (2020). Learning convex optimization control policies. In *Learning for Dynamics and Control*, pages 361–373. PMLR.

Chen, Y. and He, J. (2021). Average convergence rate of evolutionary algorithms in continuous optimization. *Information Sciences*, 562:200–219.

Kathirgamanathan, A., Twardowski, K., Mangina, E., and Finn, D. P. (2020). A centralised soft actor critic deep reinforcement learning approach to district demand side management through citylearn. In *Proceedings of the 1st International Workshop on Reinforcement Learning for Energy Management in Buildings & Cities*, pages 11–14.

Kendall, D. (1966). Bases mathématiques du calcul des probabilités. by j. neveu. pp. xii, 203. 55 francs. 1964.(masson, paris.). *The Mathematical Gazette*, 50(371):82–83.

Loeve, M. (1963). *Probability theory, 3rd*. PhD thesis, ed. D. Van Nostrand, 1963. 2.

Vazquez-Canteli, J. R., Dey, S., Henze, G., and Nagy, Z. (2020a). Citylearn: Standardizing research in multi-agent reinforcement learning for demand response and urban energy management. *arXiv preprint arXiv:2012.10504*.

Vazquez-Canteli, J. R., Henze, G., and Nagy, Z. (2020b). Marlisa: Multi-agent reinforcement learning with iterative sequential action selection for load shaping of grid-interactive connected buildings. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 170–179.