Winning the CityLearn Challenge: Adaptive Optimization with Evolutionary Search

Vanshaj Khattar and Ming Jin

Abstract

Modern power systems will have to face difficult challenges in the years to come: frequent blackouts in urban areas caused by high peaks of electricity demand, grid instability exacerbated by the intermittency of renewable generation, and climate change on a global scale amplified by increasing carbon emissions. While current practices are growingly inadequate, the pathway of artificial intelligence (AI)-based methods to widespread adoption is hindered by missing aspects of trustworthiness. The CityLearn Challenge is an exemplary opportunity for researchers from multi-disciplinary fields to investigate the potential of AI to tackle these pressing issues within the energy domain, collectively modeled as a reinforcement learning (RL) task. Multiple real-world challenges faced by contemporary RL techniques are embodied in the problem formulation. In this paper, we present a novel method using the solution functions of optimization as policies to compute the actions for sequential decision making, while notably adapting the parameters of the optimization model from noisy observations. Algorithmically, this is achieved by an evolutionary scheme. Formally, the global convergence property is established. Our agent ranked the first place in the latest 2021 CityLearn Challenge, being able to achieve superior performance in almost all metrics while maintaining some key aspects of interpretability.

Introduction

Rapid urbanization in the past decades has led to substantial increase in energy use that puts stress on the grid assets, while the integration of additional renewable generation and energy storage at the distribution level introduces both opportunities and new challenges (Vazquez-Canteli et al. 2020). The cornerstone to handle emerging issues is the deployment of proper control and coordination strategies, which has potential impact on 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) (DiCamillo 2019).

The current industry practice is heavily based on optimization models, such as energy dispatch (ED) and unit commitment (UC), where the parameters (e.g., technological and physical constraints) are fixed throughout the lifecycle; however, such an approach is increasingly confronted by the uncertainty of environment, stochasticity of renewable generation, and ever-increasing complexity of the distribution grid (Abedi, Gaudard, and Romerio 2019). On the other hand, there has been a surge of machine learning (ML) research, notably RL, since it allows the agent to act without the need to access the true model—a feature of particular interests for large-scale, complex systems, where it is not cost-effective to develop models of such high fidelity. However, unlike optimization-theoretic approaches, ML-based techniques lack the necessary mathematical framework to provide guarantees on correctness, such as physical constraint satisfaction (e.g., energy balance, thermal limits), causing concerns about trustworthiness (Amodei et al. 2016). Such concerns are further aggravated by the opaqueness, vulnerability, and fragility of ML systems (Stoica et al. 2017). Despite recent progress towards addressing trustworthiness, real-world RL is still in its infancy (Dulac-Arnold et al. 2021).

Against this backdrop, the CityLearn Challenge aims to spur RL solutions to the control of modern energy systems by providing a set of benchmarks for urban energy management, load shaping, and demand response in a range of climate zones (Vazquez-Canteli et al. 2020). The agent is tasked to explore and exploit the best coordination strategy of energy storages distributed in a community of buildings. The performance is evaluated against standard metrics such as ramping cost, peak demands, and carbon emissions. The CityLearn encapsulates 4 out of the 9 challenges for the real-world RL identified by (Dulac-Arnold et al. 2021), including *1)* the ability to learn on live systems from limited samples—there is no training period; *2)* dealing with system constraints that should never or rarely be violated—there are strict balancing equations for electricity, heating, and cooling energy; *3)* the ability to provide actions quickly—there is a strict time limit for completing the 4 year evaluation on Google's Colab; and *4)* providing system operators with explainable policies—a necessity to facilitate real-world adoption and deployment.

In this paper, we provide a technical note of our winning solution for the 2021 CityLearn challenge based on **adaptive optimization**.[1] Indeed, optimization (especially convex

[1] Our strategy is also applicable to the ongoing 2022 CityLearn Challenge (NeurIPS).

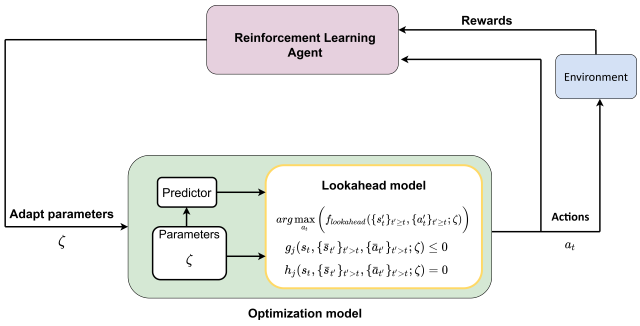

Figure 1: Basic architecture for reinforcement learning with optimization as a policy.

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, Boyd, and Vandenberghe 2004). Such approaches can easily encode domain-specific constraints (in the form of nonlinear functions, variational inequalities, or fixed point equations), and can gracefully handle problems with millions of decision variables (Facchinei and Pang 2007). Although well established, optimization models, once built, typically do not adapt to the real-world conditions, rendering current approaches rather "rigid." Fundamentally, the solution of an optimization lies on a manifold implicitly defined by a general equation (Dontchev and Rockafellar 2009). The crux of our idea is to shape this manifold by adapting the parameters of the optimization model (i.e., objective function and constraints) (see Fig. 1 for an illustration).

The key difference between an RL problem (our setting) with well-studied problems in optimization (e.g., stochastic optimization (Powell 2020), bi-level optimization (Dempe and Zemkoho 2020)) is that RL only allows access to the environment through interactive samples (reward, states, etc.) but not the true dynamics or reward function. Zeroth-order algorithms, such as simultaneous perturbation (Spall 2005; Spall 1998) and Bayesian optimization (Snoek, Larochelle, and Adams 2012; Frazier 2018) are natural candidate for RL and easy to implement in general (see, e.g., (Mania, Guy, and Recht 2018)), but may potentially suffer from scalability issues (Ghadimi and Lan 2013). Nonetheless, the parameters of an optimization model (i.e., variables to be learned) usually have clear interpretations. Thus, we design a mechanism to use proper guidance on the initialization and search of these parameters. Under some mild, verifiable conditions, we prove the asymptotic convergence to the set of global optima of the potentially nonconvex optimization. Through some simple reductions, the method works well in an online environment without an extensive training period, which is especially advantageous in a real-world setting where offline environment for model training is usually not available. According to independent evaluations, the proposed method achieved the highest performance in the recent 2021 CityLearn Challenge (CLC) competition. To demonstrate effectiveness against existing techniques, we further validate the method by comparing with a range of baselines.

Related work

Optimal control and stochastic optimal control are well-known approaches to sequential decision makings problems. (Nozhati, Ellingwood, and Chong 2020). 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 RL. Various large-scale stochastic program models have been proposed in the literature to handle future uncertainty (Carpentier, Gendreau, and Bastin 2014; Jin et al. 2011; Lium, Crainic, and Wallace 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 has 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, there have been emerging concerns about trustworthiness (Amodei et al. 2016; Stoica et al. 2017). The prime advantage of our method is that it is adaptive to any applications due to the simplicity and ubiquity of convex optimization policies.

To contextualize the present approach, we make a few remarks regarding the relation to model-based RL. In particular *implicit* MBRL, where the entire MBRL procedure (e.g., model learning and planning) is optimized for optimal policy computation (Moerland, Broekens, and Jonker 2020). However, unlike existing works (e.g. MuZero (Schrittwieser et al. 2020), Value Prediction Networks (Oh, Singh, and Lee 2017), Predictron (Silver et al. 2017), MCTSNets (Guez et al. 2018), Universal Planning Networks (Srinivas et al. 2018)), which build a model based on (recurrent) neural nets and plan by unrolling the model, our method learns how to plan by solving and adapting the parameters of an optimization problem. The present work is closely related to (Agrawal et al. 2020; **?**), which share the line of thinking that uses convex optimization as a policy class to handle uncertainty. In particular, convex optimization control policies are learned in (Agrawal et al. 2020) by tuning the parameters within the convex optimization layer. We extend their method to the RL setting.

Preliminaries

Problem setup

Consider an MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r)$, where $\mathcal{S}$ is the (possible infinite) state space, $\mathcal{A}$ is the set of actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{M}(\mathcal{S})$ is the transition probability kernel with $\mathcal{M}(\mathcal{S})$ denoting the set of all probability measures over $\mathcal{S}$ and $\mathcal{P}(\cdot|s, a)$ defining the next-state distribution upon taking action a from state s, and $r(s, a)$ gives the corresponding immediate reward (can be time dependent). The goal in RL is to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the cumulative rewards over a finite time horizon:

$$\max_{\pi \in \Pi} \quad \mathbb{E}\left[R(\pi)\right], \tag{1}$$

where $R(\pi) \coloneqq \sum_{t=0}^{T} r_t\big(s_t, \pi(s_t)\big)$ is the episodic reward, with $s_t \in \mathcal{S} \subseteq \mathbb{R}^{n_s}$ denoting the state at time t and

T as the length of an episode. The expectation is taken over the initial state distribution and transition dynamics (under the policy π). To make our method general, for any policy π, we only require access to a random sample $R(\pi)$ of the episodic reward (instead of the per-step reward $r_t(s_t, \pi(s_t))$ within the episode). We refer to the above setting as stochastic zeroth-order oracle, in alignment of the optimization literature (Dasgupta and Michalewicz 2013; Ghadimi and Lan 2013). In the following, we specify the set of policies Π as the solution functions of an optimization (Dontchev and Rockafellar 2009).

Canonical approaches and solution functions

The proposed method is motivated by canonical methods in stochastic programming, which address the above problem (1) by formulating a stochastic optimization problem with a lookahead model to account for the impact of present decisions on future outcomes (Powell 2020). For example, in multi-stage stochastic programming (Pflug and Pichler 2014), the action at state s_t is computed as

$$\arg\max_{a_t \in \mathcal{A}} \Big(\tilde{r}_t\big(s_t, a_t\big) + \max_{\pi \in \Pi} \tilde{\mathbb{E}} \Big[\sum_{t'=t+1}^{T} \tilde{r}_{t'}\big(s_{t'}, \pi(s_{t'})\big) \Big| s_t, a_t \Big] \Big), \tag{2}$$

where $\tilde{r}_t : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and $\tilde{\mathbb{E}}[\cdot]$ are the *surrogate* reward function and *surrogate* expectation operator (e.g., model-based scenario trees) designed to *approximate the true reward and environment*. Formulation (2) can be also viewed as finding the solution to a Bellman equation in dynamic programming. While stochastic programming is widely used in practice, the main drawbacks are the high computational cost to evaluate the expectation operator and the potential model mismatch due to approximations.

A simpler yet more practically appealing method, widely adopted in the industries nowadays, is to use deterministic approximations of the future and capture the dependence of future states on prior decisions through constraints as part of the lookahead model (Powell 2020) (also commonly referred to as model-predictive control (MPC) (Borrelli, Bemporad, and Morari 2017)):

$$\begin{aligned} \pi_\zeta(s_t) = \quad & \arg\max_{\bar{a}_t \in \mathcal{A}} \max_{\bar{s}_{t'} \in \mathcal{S}, \bar{a}_{t'} \in \mathcal{A}, t'=t+1,\dots,T} \Bigg(\bar{r}_t\big(s_t, \bar{a}_t; \zeta\big) + \\ & \sum_{t'=t+1}^{T} \bar{r}_{t'}\big(\bar{s}_{t'}, \bar{a}_{t'}; \zeta\big) \Bigg) \\ \text{s. t.} \quad & g_j(s_t, \{\bar{s}_{t'}\}_{t'=t+1}^{T}, \{\bar{a}_{t'}\}_{t'=t}^{T}; \zeta) \le 0 \ ; \quad j \in \mathcal{I} \\ & h_i(s_t, \{\bar{s}_{t'}\}_{t'=t+1}^{T}, \{\bar{a}_{t'}\}_{t'=t}^{T}; \zeta) = 0 \ ; \quad i \in \mathcal{E} \end{aligned} \tag{3}$$

where $\{\bar{s}_{t'}\}_{t'=t+1}^{T}$ and $\{\bar{a}_{t'}\}_{t'=t}^{T}$ are the optimization variables corresponding to the planned states and actions, and the feasible set is defined by g_j for $j \in \mathcal{I}$ and h_i for $i \in \mathcal{E}$. We denote the parameters of the objective function and the constraints collectively by $\zeta \in \mathcal{Z} \subset \mathbb{R}^d$. The dependencies of future states on current and *planned* states/actions are encoded as constraints in (3) as a part of the lookahead model. Many examples can be found in the MPC literature (Borrelli, Bemporad, and Morari 2017).

The policy $\pi_\zeta(s_t)$ provides the action at the current state s_t as the optimal solution to (3), which is also known as the solution function (Dontchev and Rockafellar 2009). As this function is in general set-valued (Dontchev and Rockafellar 2009), we make the following assumption.

Assumption 1. *For each $\zeta \in \mathbb{R}^d$ and $s_t \in \mathcal{S}$:*

- *The objective function in* (3) *is continuous, strictly convex, g_j is continuous and convex for each $j \in \mathcal{I}$, and h_i is affine for each $i \in \mathcal{E}$.*
- *The feasible set of* (3) *is closed, absolutely bounded, and has a nonempty interior.*

The above assumption can be satisfied by imposing proper conditions on the design of the surrogate model, i.e., objective and constraints in (3). Note that in our approach, we make no convexity assumptions about the true dynamics or rewards of the environment, which can be viewed as a black-box. The convexity condition is only stipulated for the "surrogate model" for computational efficiency. Our objective is simply to learn the parameters of the optimization model to have good decision-making capability. Perhaps surprisingly, despite that (3) is convex, the policy (as the solution function) can be highly nonconvex with high representational capacity. We provide some preliminary results showing the "universal approximation" property of the solution functions of linear programs (LPs). Due to the page limit and the fact that it only serves as a justification of the choice of solution functions as policies, we provide such a discussion in the appendix, along with all proofs of the results.

An immediate consequence of the above assumption is that the solution to (3) is unique; furthermore, it implies the continuity with respect to parameters.

Lemma 1. *The solution function $\pi_\zeta(s_t)$ defined in* (3) *is continuous with respect to parameter ζ for each $s_t \in \mathcal{S}$.*

The proof is a direct application of the Berge maximum theorem (Berge 1997). To end this section, let us make some remarks regarding the construction of the surrogate model of (3). In analogy to reward design (Prakash et al. 2020), the objective function should be chosen to promote desirable behaviors. The set of constraints introduce inductive bias on the transition dynamics of the environment. It is beneficial, though oftentimes unlikely and non-essential, that the surrogate model matches the functional forms of the true reward or dynamics, an idea shared in model-based RL (Moerland, Broekens, and Jonker 2020). It is, nevertheless, desirable to ensure the computational efficiency of (3) to provide actions quickly—hence the choice of convex programs.

Policy adaptation with evolutionary search

The potential mismatch between the surrogate model and the real environment and errors due to predictions may adversely affect the decision quality of (3). Thereby, we aim to adapt the parameters of the surrogate model to shape the solution function. The task of finding the optimal parameter within the set of solution functions $\Pi = \{\pi_\zeta : \zeta \in \mathcal{Z}\}$ can be

compactly written as:

$$\zeta^\star = \arg\max_{\zeta\in\mathcal{Z}} \mathbb{E}\left[\sum_{t=0}^{T} r_t\big(s_t, \pi_\zeta(s_t)\big)\right]. \tag{4}$$

Note that ζ is not a part of the true reward (which remains unknown to the agent), but only the parameters of the surrogate model that implicitly defines the policy in (3). Since $\pi_\zeta(s_t)$ is given by an optimization (3), (4) can be also viewed as a bi-level problem (c.f., (Dempe and Zemkoho 2020)): the outer level aims at learning the parameters to maximize rewards, while the inner level defines the policy action in each state as a solution to (3). The key challenge to solve (4) as a bi-level problem, nevertheless, is that the outer level objective can be only accessed through the stochastic zeroth-order oracle and can be nonconvex with respect to the variable ζ.

Guided evolutionary search

This section discusses the proposed evolutionary algorithm, inspired by the method of generations (Zhigljavsky 2012) (as detailed in Algorithm 1). In a nutshell, at each iteration k, the algorithm randomly samples a set of N_k parameter candidates, $\zeta_1^k, \cdots, \zeta_{N_k}^k \overset{iid}{\sim} p_k$. For each candidate $j \in \{1, ..., N_k\}$, we evaluate the corresponding policy in the environment and observe an episodic reward $R_j^k \sim R(\pi_{\zeta_j^k})$ (here, we slightly overload the notation $R(\pi_{\zeta_j^k})$ to denote the distribution of episodic reward for policy $\pi_{\zeta_j^k}$). Then, we update the distribution for the next iteration as

$$p_{k+1}(d\zeta) = \sum_{j=1}^{N_k} r_j^k Q_k(\zeta_j^k, d\zeta), \tag{5}$$

where

$$r_j^k = \frac{\exp(R_j^k)}{\sum_{j=1}^{N_k}\exp(R_j^k)} \tag{6}$$

are the weights obtained by the softmax function, which promotes candidates with higher rewards. The probability measure $Q_k(\zeta_j^k, d\zeta)$ is the transition probability given candidate ζ_j^k. Hence, $p_{k+1}(d\zeta)$ is a mixture of distributions weighted by observed rewards in the current iteration k, which can be sampled by the standard superposition method: at first the index j is sampled from the discrete distribution $\{r_j^k\}$, and then the distribution $Q_k(\zeta_j^k, d\zeta)$ is sampled given the realization of ζ_j^k. The transition probability used in our algorithm is given by:

$$Q_k(z, d\zeta) \sim \exp(\|\zeta - z\|/\iota_k)\mu(d\zeta), \tag{7}$$

where $\mu(d\zeta)$ is a uniform measure over $\mathcal{Z}$ and $\iota_k \sim \frac{1}{k^2}$ decays at the rate of $\frac{1}{k^2}$. Other candidates are possible and can still ensure convergence to global optimal, as long as certain conditions are met, such as the span of Q decreases over time but does not decrease too quickly that it fails to hit a global optimum. More rigorous analysis is left for the next section. The overall visualisation of the algorithm 1 is presented in Figure 2.

Algorithm 1: Evolutionary search algorithm

Input: Hyperparameters $\{N_k\}$, uniform distribution μ

1: Initialize $P_1 \sim \exp(\|\zeta - z\|)\mu(d\zeta)$,
2: **for** $k = 1, 2, \ldots$ **do**
3: Sample N_k candidates from the distribution p_k: $\zeta_1^k, \zeta_2^k, \cdots, \zeta_{N_k}^k \overset{\text{iid}}{\sim} p_k$
4: **for** $j = 1, \ldots, N_k$ **do**
5: Deploy policy $\pi_{\zeta_j^k}$ for one episode and observe an episodic reward $R_j^k \leftarrow R(\pi_{\zeta_j^k})$
6: **end for**
7: Update the distribution p_{k+1} for the next iteration according to (5).
8: **end for**

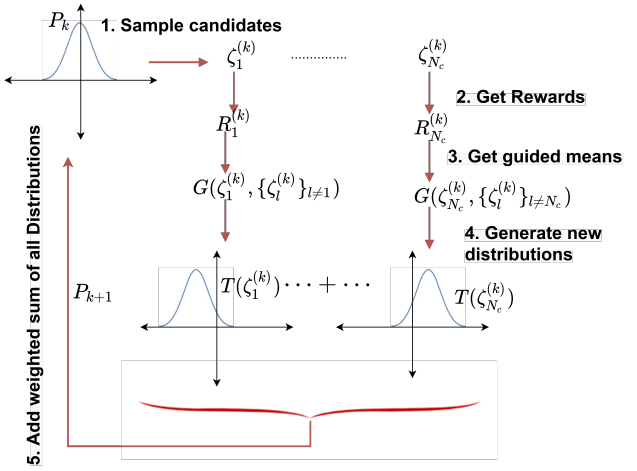

Figure 2: Visualization of the guidance mechanism in Algorithm 1.

Theoretical analysis

We now analyze the convergence property of the sequence generated by Algorithm 1. We adopt the following notations: $f(\zeta) = \mathbb{E}[R(\pi_\zeta)]$ is the expected episodic reward of policy π_ζ, while $R_j^k \sim R(\pi_{\zeta_j^k})$ is a sampled value for candidate j at iteration k; $\Lambda = \arg\max_{\zeta\in\mathcal{Z}} f(\zeta)$ is the set of global maximizers (may not be unique), $f^* = \max_{\zeta\in\mathcal{Z}} f(\zeta)$ is the global maximum, and $\lambda(d\zeta)$ is some measure over Λ; $\mathbb{B}(\zeta, \epsilon) = \{\zeta' \in \mathcal{Z} : \|\zeta' - \zeta\| \le \epsilon\}$ is a ball centered at ζ with radius ϵ, $\mathbb{B}^*(\epsilon) = \{\zeta \in \mathcal{Z} : \min_{\zeta'\in\Lambda} \|\zeta' - \zeta\| \le \epsilon\}$ is a set of points that are ϵ away from the optimal solution set Λ; $\delta_\zeta(dz)$ is the probability measure concentrated at the point ζ. We also use $\Rightarrow$ to denote weak convergence of measures.

For the analysis of this section, we make the following assumptions.

Assumption 2. *The followings hold:*

(a) $R_j^k = f(\zeta_j^k) + \xi_j^k$, *where* $\xi_j^k \overset{iid}{\sim} F_k(d\xi)$ *for any* $k \in \mathbb{N}$ *are independent random variables with distribution* $F_k(d\xi)$ *bounded on a finite internal* $[-c_\xi, c_\xi]$ *and* $\mathbb{E}\exp(\xi_j^k) = 1$;

(b) $|f(\zeta)| \leq c_f$ *for all* $\zeta \in \mathcal{Z}$*;*

(c) there exists $\epsilon > 0$ *such that* f *is continuous on* $\mathbb{B}^*(\epsilon)$*;*

(d) $Q_k(z, d\zeta) = q_k(z, \zeta)\mu(d\zeta)$*, with* $\sup_{z,\zeta\in\mathcal{Z}} q_k(z, \zeta) \leq L_k < \infty$ *for all* $k = 1, 2, ...$*, where* μ *is a probability measure such that* $\mu(\mathbb{B}^*(\epsilon)) > 0$ *for any* $\epsilon > 0$*; for any* $\zeta \in \mathcal{Z}$ *and* $k \to \infty$*, the sequence of probability measures* $Q_k(\zeta, dz)$ *weakly converges to* $\delta_\zeta(dz)$*;*

(e) $\{N_k\}$ *is a sequence of natural numbers* N_k *such that* $N_k \to \infty$ *for* $k \to \infty$*;*

(f) $\tilde{P}_1(\mathbb{B}(\zeta, \epsilon)) > 0$ *for all* $\epsilon > 0$*,* $\zeta \in \mathcal{Z}$*;*

(g) for any $\epsilon > 0$*, there are* $\delta > 0$ *and a natural* $\bar{k}$ *such that* $\tilde{P}_k(\mathbb{B}^*(\epsilon)) \geq \delta$ *for all* $k \geq \bar{k}$*.*

Let us comment on the assumptions above. Condition (a) requires that the evaluation noises be independent and bounded; the requirement on the expectation can be satisfied for truncated log-normal distributions (Thompson 1950); the iid requirement can be relaxed to mixing processes at the cost of more complicated analysis (Doukhan 2012); the boundedness condition, on the other hand, seems necessary to keep the iterates in the vicinity of global maximum if they are already there. Condition (b) is natural on practical problems. Condition (c) is natural given that $\pi(\zeta)$ is continuous by Lemma 1, and can be met if the true reward functions r_t are continuous for all $t = 0, 1, ..., T$. Assumptions $(d), (e), (f)$ and (g) formulate necessary requirements on the parameters of the algorithm that need to be satisfied. Intuitively, conditions (d) and (e) stipulate that the search becomes more "focused" over time in order to concentrate on the global optima; however, conditions (f) and (g) indicates that the decrease of span cannot be too fast in order not to miss the global optima. Condition (e) can be relaxed to $N_k = N$ for some finite integer N for all $k = 1, ...$, but the convergence will only be towards to the vicinity of Λ due to the finite sample effect (see Lemma 4 in the appendix, which states the rate to be on the order $N^{-1/2}$). Unlike $(c), (d)$, and (e), (g) is not constructive; however, we provide some verifiable conditions sufficient for (g) to hold in Corollaries 1 and 2.

Measures $p_{k+1}(d\zeta)$, $k = 1, 2, ...$ defined in (5) are distributions of random points ζ_j^{k+1} conditioned on the results of preceding evaluations of f with respect to realizations of ξ_j^k and ζ_j^k, for $j = 1, ..., N_k$. Let $P_k(d\zeta_1, ..., d\zeta_{N_k})$ denote their unconditional joint distributions at iteration k. Next, we provide the update rule according to Algorithm 1 (note that we introduce z for ζ as integration variables).

Lemma 2. *The probability distribution* $P_{k+1}(d\zeta_1, ..., d\zeta_{N_{k+1}})$ *can be written in terms of the distribution* $P_k(d\zeta_1, ..., d\zeta_{N_k})$ *as:*

$$\int_{\Omega^{N_k}} \chi_k(d\omega_{N_k}) \prod_{j=1}^{N_{k+1}} \left\{ \beta(\omega_{N_k}) \sum_{i=1}^{N_k} \Lambda(z_i, \xi_i, d\zeta_j) \right\}, \quad (8)$$

where $\Omega = \mathcal{Z} \times [-c_\xi, c_\xi]$,

$$\begin{aligned}
\omega_{N_k} &= \{z_1, ..., z_{N_k}, \xi_1, ..., \xi_{N_k}\} \in \Omega^{N_k}, \\
\chi_k(d\omega_{N_k}) &= P_k(dz_1, ..., dz_{N_k}) F_k(d\xi_1) \cdots F_k(d\xi_{N_k}), \\
\beta(\omega_{N_k}) &= \frac{1}{\sum_{j=1}^{N_k} \exp(f(z_j) + \xi_j)}, \quad \textit{and} \\
\Lambda(z, \xi, d\zeta) &= \exp(f(z) + \xi) Q_k(z, d\zeta).
\end{aligned}$$

The proof is immediate by recognizing that the term in the bracket in (8) is the conditional distribution $p_{k+1}(d\zeta_j)$ defined in (5), and the integration is overall the distribution of random variables from the preceding iteration. Note that we take the produce over N_{k+1} candidates since they are drawn iid from p_{k+1}.

Now, we provide the main result on the convergence of the unconditional marginal distribution

$$\tilde{P}_k(d\zeta) = \int_{\mathcal{Z}^{N_k - 1}} P_k(d\zeta, dz_2, ..., dz_{N_k})$$

to some distribution $\lambda(d\zeta)$ over the global optimal set.

Theorem 1. *Suppose that Assumption 2 holds true, and let* $\{\tilde{P}_k\}$ *be the sequence of distributions, where* $\tilde{P}_k(d\zeta)$ *is the unconditional marginal distribution of* ζ *at iteration* k *determined by Algorithm 1. Then, the distribution sequence weakly converges to* $\lambda(d\zeta)$*, i.e.,* $\tilde{P}_k \Rightarrow \lambda$*, for* $k \to \infty$*.*

The key stage of the proof is to show that there exists a subsequence in $\{\tilde{P}_k\}$ that weakly converges to the distribution

$$\vartheta_m(d\zeta) = \frac{\exp(mf(\zeta))\mu(d\zeta)}{\int \exp(mf(z))\mu(dz)},$$

where m is the index of the subsequence. The above distribution is effectively a softmax function over the function, and converges to the max function as m goes to infinity. The rigorous proof can be found in the appendix.

All the conditions in Assumption 2 are natural and non-restrictive with the exception of (g), which requires some further justification. In the following, we present sufficient conditions for (g) for two important ways of desining $Q_k(z, d\zeta)$.

Corollary 1. *Under Assumption 2 (except for* (g)*), and further assume that* f *can be evaluated without noise (i.e.,* $\xi = 0$*). Let the transition probability* $Q_k(z, A)$ *be defined by*

$$\begin{aligned}
Q_k(z, A) = &\int \mathbb{1}_{\{\zeta \in A, f(z) \leq f(\zeta)\}} T_k(z, d\zeta) \\
&+ \mathbb{1}_{\{x \in A\}} \int \mathbb{1}_{\{f(\zeta) < f(z)\}} T_k(z, d\zeta), \quad (9)
\end{aligned}$$

where $\{T_k(z, d\zeta)\}$ *weakly converges to* $\delta_z(d\zeta)$ *for all* $z \in \mathcal{Z}$*. Then, there exists a sequence of natural numbers* N_k *such that the sequence of distributions* $\{\tilde{P}_k\}$ *weakly converges to* λ *for* $k \to \infty$*.*

To implement the transition of (9), one first needs to sample a variable ζ according to $T_k(z, d\zeta)$ and observe its reward value $f(\zeta)$; then, the output is ζ if $f(\zeta) \geq f(z)$ and z otherwise. Such scheme crucially depends on a reliable way of comparing candidates (e.g., noiseless evaluation). The next result applies more generally in the presence of random noise.

Corollary 2. *Under Assumption 2 (except for (g)), and suppose the transition probability $Q_k(z, d\zeta)$ is defined by*

$$Q_k(z, d\zeta) = c_k(z)\psi((\zeta - z)/\iota_k)\mu(d\zeta), \qquad (10)$$

where $c_k(z) = (\int \psi((\zeta - z)/\iota_k)\mu(d\zeta))^{-1}$ is the normalization term, ψ is a continuous symmetrical finite density on $\mathcal{Z}$, and

$$\iota_k > 0, \qquad \sum_{k=1}^{\infty} \iota_k < \infty.$$

Then, there exists a sequence of natural numbers N_k such that $\{\tilde{P}_k\}$ weakly converges to λ for $k \to \infty$.

A special case of the above transition is implemented in our experiment, where μ is the uniform distribution and $\psi = \exp(\|\zeta - z\|/\iota_k)$ is the (unnormalized) Gaussian distribution. Note that some recent works have explored the average rate of convergence for evolutionary algorithms (EAs) (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.

Results from the CityLearn Challenge

Challenge overview. 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. The goal of each agent is to minimize the costs from the environment, such as ramping cost, peak demands, 1-load factor, and carbon emissions. The state space contains information such as hour of day, outdoor temperature/relative humidity/-solar radiation (and 6/12/24 hour-ahead predictions), electricity currently consumed by electrical appliances, state of the charges (SOCs) of the storage devices, current net electricity consumption of the building, current carbon intensity of the power grid, among the total 30 continuous states. The agent is allowed to control the charging/discharging actions of storage devices for domestic hot water (DHW), chilled water, and electricity (in total of 3 continuous actions per building). The environment is viewed as a blackbox to the agent as standard RL setup, where the transition dynamics depend on various energy models (e.g., air-to-water heat pumps, electric heaters) as well as energy loads of the buildings, which include space cooling, dehumidification, appliances, DHW, and solar generation.

Evaluation. The submission of every team will be evaluated on a set of metrics, including: *(1)* ramping: $\sum |e_t - e_{t-1}|$, where e is the net electricity consumption at every time-step; *(2)* 1-load factor: the average net electricity load divided by the maximum electricity load; *(3)* average daily peak net demand; *(4)* maximum peak electricity demand; *(5)* total amount of electricity consumed; *(6)* total amount of carbon emissions. The competition evaluates the performance by computing the ratio of costs with respect to a rule-based controller (RBC)—lower ratios indicate better performances. Note that the RBC controller is ubiquitous in traditional building control systems, and is simply of the form "take action a_h in hour h," where a_h is a constant independent of current states except the hour of the day ($h \in \{1, ..., 24\}$). We refer the readers to the online documents[2] and publication (Vazquez-Canteli et al. 2020) for detailed description of the competition. We will only focus on our strategy in this document.

ZO-iRL (zeroth-order implicit RL) strategy. We name our method ZO-iRL since the policy action is implicitly determined by solving an optimization problem and the learning algorithm is zeroth-order in an RL setting. As our method is designed for single-agent episodic RL, we first reduce the original task that consists of a single episode with the length of 4 years to multiple episodes with the length of a day. We use the per-step reward $-\max(0, e_t)^3$ as recommended by (Vazquez-Canteli et al. 2020), where e_t is the net electricity consumption (or generation if $e_t < 0$). This reward favors consumption patters that are close to average throughout the day rather than peaked in a few hours, which are also aligned with the actual metrics used in the evaluation, such as 1-load factor and maximum peak electricity demand. Another reduction performed is from multi-agent RL to single-agent RL, where each building is controlled and its policy updated separately in a decentralized control fashion.

We instantiate the optimization in (3) as follows:

$$\begin{aligned} &\arg\max_{\bar{a}_t \in \mathcal{A}} \max_{\bar{s}_{t'} \in \mathcal{S}, \bar{a}_{t'} \in \mathcal{A}, t'=t+1,...,T} \sum_{t'=t+1}^{T} \bar{r}_{t'}(\bar{s}_{t'}; \zeta) \\ &\text{s. t. } g_j(s_t, \{\bar{s}_{t'}\}_{t'=t+1}^{T}, \{\bar{a}_{t'}\}_{t'=t}^{T}) \le 0\,; \quad j \in \mathcal{I} \\ &\qquad h_i(s_t, \{\bar{s}_{t'}\}_{t'=t+1}^{T}, \{\bar{a}_{t'}\}_{t'=t}^{T}) = 0\,; \quad i \in \mathcal{E} \end{aligned} \qquad (11)$$

where s_t consists of a subset of state variables, such as the net electricity consumption and SOCs of storage devices, and the surrogate reward

$$\bar{r}_t(s_t; \zeta)) = -|e_t - e_{t-1}| - \theta_t e_t$$

is a combination of the ramping cost and the "virtual" electricity cost, where $\theta_t \in [0, 10]$ can be viewed as the virtual electricity price to be learned in order to encourage desirable consumption patterns (e.g., load flattening and smoothing). For instance, a higher value of θ_t discourages consumption in the corresponding hour t. The inequalities can be grouped into technology constraints (e.g., maximum/minimum cooling power) and bounds on states and actions. The equalities can be grouped into physics accounting for energy balances (i.e., consumption is equal to supply) and technology (e.g., rules of updating SOCs). As these are standard in energy modeling (see, e.g., **??**), we provide details in the appendix.

Let $\zeta = \{\theta_t\}_{t=1,...,24}$ denote the set of policy parameters. The aim of the agent is to learn $\zeta \in \mathbb{R}^{24}$ that represents the virtual electricity costs. Note that the optimization (11) also depends on predictions of energy demands and solar generation in the future. For simplicity, our predictors are based on a simple averaging scheme that takes the average of the value in the corresponding hour among the last 2 weeks data; thus, there are no specific needs to tune parameters.

It can be observed from Fig. 4 that ZO-iRL has achieved the lowest cost ratios (i.e. the best scores) as compared to baseline methods. In particular, baseline RL methods, that

[2] https://sites.google.com/view/citylearnchallenge/

	ZOiRL	ICD-CA	IDLab-EMIB
Total score	**0.944**	1.0705	1.0702
Total last year	**0.942**	1.052	1.077
Coordination score	**0.915**	1.107	1.094
Coordination score last yr.	**0.918**	1.074	1.098
Carbon emissions	1.003	**1.000**	1.028

Table 1: Performance of top 3 teams on the CityLearn challenge 2021. The green represents the best score for a category. (ZOiRL - winning entry)

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 2: Scores for ZO-iRL and comparison methods, including SAC kathirgamanathan2020centralised and MARLISA vazquez2020marlisa. The random agent basically uniformly selects an action within the range at each timestep.

have otherwise shown empirical successes in game playing and various control tasks, 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. 4). Furthermore, the learning progress is more interpretable as we can inspect the evolution of parameter p_{ele} as shown in Fig. 3. 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 smooth-es the peaks, thus resulting in better performance. More details can be found in the supplementary material.

Conclusion and future directions

The present work introduces a novel framework for implicit model-based RL. By exploiting the strength of convex optimization, the proposed method is able to simultaneously address a range of challenges for real-world RL. Using optimization solution-functions as policies offers a promising way to introduce data-driven algorithms into the real world with interpretability. Such method can be potentially extended to a wide range of problems where optimization models exist. Our work opens up exciting research directions for future works, including the extension of the proposed framework to other derivative-free methods such as Bayesian optimization or first-order methods such as actor-critic method.

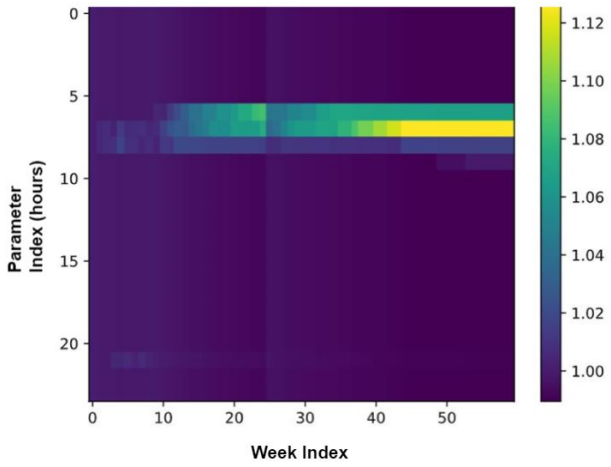

Figure 3: Evolution of the implicit parameters p_{ele} over the test period, where the values are color-coded.

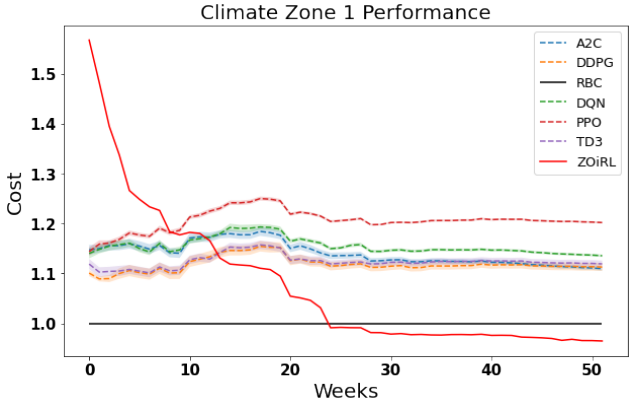

Figure 4: Learning curve of ZO-iRL compared with other RL-baselines, where Rule Based Controller (RBC) takes a baseline cost of constant 1.

References

[Abedi, Gaudard, and Romerio 2019] Abedi, A.; Gaudard, L.; and Romerio, F. 2019. Review of major approaches to analyze vulnerability in power system. *Reliability Engineering & System Safety* 183:153–172.

[Agrawal et al. 2020] Agrawal, A.; Barratt, S.; Boyd, S.; and Stellato, B. 2020. Learning convex optimization control

policies. In *Learning for Dynamics and Control*, 361–373. PMLR.

[Amodei et al. 2016] Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; and Mané, D. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

[Bačák and Borwein 2011] Bačák, M., and Borwein, J. M. 2011. On difference convexity of locally lipschitz functions. *Optimization* 60(8-9):961–978.

[Berge 1997] Berge, C. 1997. *Topological Spaces: including a treatment of multi-valued functions, vector spaces, and convexity*. Courier Corporation.

[Billingsley 2013] Billingsley, P. 2013. *Convergence of probability measures*. John Wiley & Sons.

[Borrelli, Bemporad, and Morari 2017] Borrelli, F.; Bemporad, A.; and Morari, M. 2017. *Predictive control for linear and hybrid systems*. Cambridge University Press.

[Boyd, Boyd, and Vandenberghe 2004] Boyd, S.; Boyd, S. P.; and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

[Carpentier, Gendreau, and Bastin 2014] 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.

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

[Dasgupta and Michalewicz 2013] Dasgupta, D., and Michalewicz, Z. 2013. *Evolutionary algorithms in engineering applications*. Springer Science & Business Media.

[Dempe and Zemkoho 2020] Dempe, S., and Zemkoho, A. 2020. *Bilevel optimization*. Springer.

[DeVore and Lorentz 1993] DeVore, R. A., and Lorentz, G. G. 1993. *Constructive approximation*, volume 303. Springer Science & Business Media.

[DiCamillo 2019] DiCamillo, M. 2019. Tabulations from a late november 2019 survey of california voters about recent power blackouts in california and the problems facing the pacific gas and electric company.

[Dontchev and Rockafellar 2009] Dontchev, A. L., and Rockafellar, R. T. 2009. *Implicit functions and solution mappings*, volume 543. Springer.

[Doukhan 2012] Doukhan, P. 2012. *Mixing: properties and examples*, volume 85. Springer Science & Business Media.

[Dulac-Arnold et al. 2021] 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* 110(9):2419–2468.

[Ebert et al. 2018] 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*.

[Facchinei and Pang 2007] Facchinei, F., and Pang, J.-S. 2007. *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media.

[Frazier 2018] Frazier, P. I. 2018. Bayesian optimization. In *Recent Advances in Optimization and Modeling of Contemporary Problems*. INFORMS. 255–278.

[Ghadimi and Lan 2013] Ghadimi, S., and Lan, G. 2013. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization* 23(4):2341–2368.

[Gu et al. 2017] 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)*, 3389–3396. IEEE.

[Guez et al. 2018] Guez, A.; Weber, T.; Antonoglou, I.; Simonyan, K.; Vinyals, O.; Wierstra, D.; Munos, R.; and Silver, D. 2018. Learning to search with mctsnets. In *International conference on machine learning*, 1822–1831. PMLR.

[Hornik, Stinchcombe, and White 1989] Hornik, K.; Stinchcombe, M.; and White, H. 1989. Multilayer feedforward networks are universal approximators. *Neural networks* 2(5):359–366.

[Jin et al. 2011] 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 et al. 2020] 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*, 11–14.

[Lium, Crainic, and Wallace 2009] 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.

[Mania, Guy, and Recht 2018] Mania, H.; Guy, A.; and Recht, B. 2018. Simple random search of static linear policies is competitive for reinforcement learning. *Advances in Neural Information Processing Systems* 31.

[Mhaskar 1996] Mhaskar, H. N. 1996. Neural networks for optimal approximation of smooth and analytic functions. *Neural Computation* 8(1):164–177.

[Moerland, Broekens, and Jonker 2020] Moerland, T. M.; Broekens, J.; and Jonker, C. M. 2020. Model-based reinforcement learning: A survey. *arXiv preprint arXiv:2006.16712*.

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

[Oh, Singh, and Lee 2017] Oh, J.; Singh, S.; and Lee, H. 2017. Value prediction network. *Advances in neural information processing systems* 30.

[Pflug and Pichler 2014] Pflug, G. C., and Pichler, A. 2014. *Multistage stochastic optimization*, volume 1104. Springer.

[Powell 2020] Powell, W. 2020. Reinforcement learning and stochastic optimization: A unified framework for sequential decisions. *Princeton NJ*.

[Prakash et al. 2020] Prakash, B.; Waytowich, N.; Ganesan, A.; Oates, T.; and Mohsenin, T. 2020. Guiding safe reinforcement learning policies using structured language constraints. *UMBC Student Collection*.

[Rockafellar and Wets 2009] Rockafellar, R. T., and Wets, R. J.-B. 2009. *Variational analysis*, volume 317. Springer Science & Business Media.

[Schrittwieser et al. 2020] Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. 2020. Mastering atari, go, chess and shogi by planning with a learned model. *Nature* 588(7839):604–609.

[Silver et al. 2017] Silver, D.; Hasselt, H.; Hessel, M.; Schaul, T.; Guez, A.; Harley, T.; Dulac-Arnold, G.; Reichert, D.; Rabinowitz, N.; Barreto, A.; et al. 2017. The predictron: End-to-end learning and planning. In *International conference on machine learning*, 3191–3199. PMLR.

[Snoek, Larochelle, and Adams 2012] Snoek, J.; Larochelle, H.; and Adams, R. P. 2012. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems* 25.

[Spall 1998] Spall, J. C. 1998. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on aerospace and electronic systems* 34(3):817–823.

[Spall 2005] Spall, J. C. 2005. *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. John Wiley & Sons.

[Srinivas et al. 2018] Srinivas, A.; Jabri, A.; Abbeel, P.; Levine, S.; and Finn, C. 2018. Universal planning networks: Learning generalizable representations for visuomotor control. In *International Conference on Machine Learning*, 4732–4741. PMLR.

[Stoica et al. 2017] 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*.

[Thompson 1950] Thompson, H. 1950. Truncated normal distributions. *Nature* 165(4194):444–445.

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

[Vazquez-Canteli, Henze, and Nagy 2020] Vazquez-Canteli, J. R.; Henze, G.; and Nagy, Z. 2020. 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*, 170–179.

[Wainwright 2019] Wainwright, M. J. 2019. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press.

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

Proof of results in the main paper

Proof of Lemma 1

Let $\Phi(s_t, \zeta)$ denote the feasible set of (3). By Assumption 1, $\Phi(s_t, \zeta)$ is convex for fixed s_t and ζ and has a nonempty interior. This implies that $\Phi(s_t, \zeta)$ is continuous in s_t and ζ (Rockafellar and Wets 2009, example 5.10). Hence, by Berge maximum theorem (Berge 1997), $\pi_\zeta(s_t)$ is upper hemicontinuous in ζ for fixed $s_t \in \mathcal{S}$. However, we know that $\pi_\zeta(s_t)$ contains a single point due to strict convexity of the objective function. Thus, for fixed $s_t \in \mathcal{S}$, $\pi_\zeta(s_t)$ is a single-valued function continuous in ζ.

Proof of Theorem 1

Choose from $\{\tilde{P}_k\}$ a weakly convergent subsequence $\{\tilde{P}_{k_i}\}$, which is possible due to Prohorov's theorem (Billingsley 2013, Ch. 6), and denote the limit by $\kappa(d\zeta)$. By Lemma 4, we have that

$$\tilde{P}_{k+1}(d\zeta) = \left(\int \tilde{P}_k(dz)\exp(f(z))\right)^{-1} \int \tilde{P}_k(dz)\exp(f(z))\Big(Q_k(z, d\zeta) + \Delta_{N_k}(d\zeta)\Big). \tag{12}$$

It follows that the subsequence $\{\tilde{P}_{k_i+1}\}$ weakly converges to the distribution $\vartheta_1(d\zeta) = c_1 \exp(f(\zeta))\kappa(d\zeta)$, where c_1 is the normalization constant. Similarly, the subsequence $\{\tilde{P}_{k_i+m}\}$ weakly converges to the distribution

$$\vartheta_m(d\zeta) = \frac{\exp(mf(\zeta))\mu(d\zeta)}{\int \exp(mf(z))\mu(dz)},$$

which, by Lemma 3, converges to λ. Thus, by standard argument of diagonalization (Billingsley 2013), one can show that there exists a subsequence $\{\tilde{P}_{k_j}\}$ that weakly converges to λ. Applying Lemma 4 again yields that $\{\tilde{P}_{k_j+1}\}$ converges to the same limit. Thus, any subsequence of $\{\tilde{P}_k\}$ converges to this limit, and the same holds for the sequence itself.

Proof of Corollary 1

By Assumption 2 (c) and (f), we have that $\tilde{P}_1(\mathbb{B}^*(\epsilon)) > 0$ for any $\epsilon > 0$. By (9), we have that

$$\tilde{P}_k(\mathbb{B}^*(\epsilon)) \geq \cdots \geq \tilde{P}_1(\mathbb{B}^*(\epsilon)) > 0$$

for all $k \in \mathbb{N}$. Hence, Assumption 2 (g) is satisfied. By Theorem 1, the claim is proved.

Proof of Corollary 2

Under Assumption 2 (except for (g)), the distributions (15) have continuous densities with respect to the Lebesgue measure. Let $A(\epsilon) = \{\zeta \in \mathcal{Z} : f(\zeta) \geq f^* - \epsilon\}$. By (10) and Lemma 2, we have that $\tilde{P}_k(d\zeta) > 0$ for any $k \in \mathbb{N}$. Fix an arbitrary $\delta > 0$. We shall choose $\{N_k\}$ such that for any k and $\epsilon > 0$, the following holds

$$\tilde{P}_{k+1}(A(\epsilon + \epsilon_k)) \geq (1 - \delta_k)\tilde{P}_k(A(\epsilon)),$$

where

$$0 < \delta_k < 1 \quad \text{for } k \in \mathbb{N}, \qquad \sum_{k\in\mathbb{N}} \delta_k < \infty, \tag{13}$$

and $\epsilon_k \geq 0$ are determined in terms of ι_k and the sizes of the support of density ψ,

$$\sum_{k=1}^{\infty} \epsilon_k = \text{constant} \sum_{k=1}^{\infty} \iota_k < \infty.$$

Such sequence of $\{N_k\}$ exists by Lemma 2 and the finiteness of ψ. Next, choose k_o such that

$$\sum_{k=k_o}^{\infty} \epsilon_k < \frac{1}{2}\delta,$$

and let $\delta_1 = \tilde{P}_{k_o}(A(\delta/2))$. Then, for any $k \geq k_o$, we have

$$\begin{aligned}
\tilde{P}_{k+1}(A(\delta)) &\geq \tilde{P}_{k_o}(A(\delta/2 + \sum_{i=k_o}^{k} \delta_i)) \prod_{i=k_o}^{k} (1 - \delta_i) \\
&\geq \delta_1 \prod_{i=k_o}^{\infty} (1 - \delta_i) \\
&> 0
\end{aligned}$$

where the last inequality is implied by (13). The proof is complete.

Supporting lemmas

Lemma 3. *Under Assumption 2 $(b), (c)$, and (d), the sequence of distributions*

$$\frac{\exp(kf(\zeta))\mu(d\zeta)}{\int \exp(kf(z))\mu(dz)} \Rightarrow \lambda(d\zeta),$$

i.e., weakly converges to $\lambda(d\zeta)$ for $k \to \infty$.

Proof. By the definition of weak convergence, it suffices to show that for any function $\Psi(\zeta)$ continuous on $\mathcal{Z}$, it holds that

$$\lim_{k\to\infty} c_k \int \exp(kf(\zeta))\Psi(\zeta)\mu(d\zeta) = \int \Psi(\zeta)\lambda(d\zeta), \tag{14}$$

where $c_k = 1/\int \exp(kf(z))\mu(dz)$. To proceed, Let $\mathbb{B}_i = \mathbb{B}(\epsilon_i) = \{\zeta \in \mathcal{Z} : \min_{\zeta'\in\Lambda} \|\zeta' - \zeta\| \leq \epsilon_i\}$ and $\mathbb{D}_i = \{\zeta \in \mathcal{Z} : \min_{\zeta'\in\Lambda} \|\zeta' - \zeta\| \geq \epsilon_i\}$, for $i = 0, 1, 2$ and some ϵ_0, ϵ_1, and ϵ_2 to be determined. For any $\delta > 0$, by continuity of Ψ, there exists $\epsilon_0 > 0$ such that $|\Psi(z) - \int \Psi(\zeta)\lambda(d\zeta)| \leq \delta$ for all $z \in \mathbb{B}_0$. Choose some $\epsilon_1 > 0$ such that $\epsilon_1 < \epsilon_0$. Then, we have

$$\begin{aligned}
&\left| c_k \int \exp(kf(\zeta))\Psi(\zeta)\mu(d\zeta) - \int \Psi(\zeta)\lambda(d\zeta) \right| \\
&\leq c_k \int_{\mathbb{B}_1} \exp(kf(z)) \left| \Psi(z) - \int \Psi(\zeta)\lambda(d\zeta) \right| \mu(dz) + c_k \int_{\mathbb{D}_1} \exp(kf(z)) \left| \Psi(z) - \int \Psi(\zeta)\lambda(d\zeta) \right| \mu(dz) \\
&\leq \delta \underbrace{c_k \int_{\mathbb{B}_1} \exp(kf(z))\mu(dz)}_{(i)} + 2\|\Psi\|_\infty \underbrace{c_k \int_{\mathbb{D}_1} \exp(kf(z))\mu(dz)}_{(ii)},
\end{aligned}$$

where the first inequality is due to triangle inequality, and the second inequality is due to the choice of ϵ_1 (also, recall that $\|\Psi\|_\infty = \sup|\Psi(z)|$). Hence, the lemma is proved if we can show that $(i) \to 1$ and $(ii) \to 0$ as $k \to \infty$.

To this end, let $C_1 = \sup_{\zeta\in\mathbb{D}_1} f(\zeta)$. By Assumption 2 (c), there exists ϵ_2 such that $0 < \epsilon_2 < \epsilon_1$, and

$$C_2 = \inf_{\zeta\in\mathbb{B}_2} f(\zeta) > C_1.$$

For any $k > 0$, we have

$$\int_{\mathbb{B}_1} \exp(kf(z) - kC_1)\mu(dz) > \int_{\mathbb{B}_2} \exp(kf(z) - kC_1)\mu(dz) \geq \int_{\mathbb{B}_2} \exp(k(C_2 - C_1))\mu(dz).$$

Thus,

$$\frac{\int_{\mathbb{D}_1} \mu(dz)}{\int_{\mathbb{B}_2} \exp(k(C_2 - C_1))\mu(dz)} \geq \underbrace{\frac{\int_{\mathbb{D}_1} \exp(kf(z))\mu(dz)}{\int_{\mathbb{B}_1} \exp(kf(z))\mu(dz)}}_{(iii)} \geq 0.$$

By driving $k \to \infty$ to the limit and using the sandwich theorem, we have that $(iii) \to 0$. This immediately implies that $(i) \to 1$ and $(ii) \to 0$ as $k \to \infty$, hence concluding the proof. □

Lemma 4. *Let Assumption 2 $(a), (b)$, and (d) be fulfilled. Then, the marginal distributions can be written as*

$$\tilde{P}_{k+1}(d\zeta) = \left(\int \tilde{P}_k(dz) \exp(f(z)) \right)^{-1} \int \tilde{P}_k(dz) \exp(f(z)) Q_k(z, d\zeta) + \Delta_{N_k}(d\zeta), \tag{15}$$

where the signed measures $\Delta_{N_k}(d\zeta)$ converge to zero in variation for $N_k \to \infty$ with the rate ${N_k}^{-1/2}$.

Proof. For notational simplicity, we use N for N_k throughout the proof. By Assumption 2 (a) and Lemma 2, the marginal distribution $\tilde{P}_{k+1}(d\zeta)$ is given by:

$$\begin{aligned}
\tilde{P}_{k+1}(d\zeta) &= \int_{\Omega^N} \chi_k(d\omega_N) \left\{ \beta(\omega_N) \sum_{i=1}^N \Lambda(\zeta_i, \xi_i, d\zeta) \right\} \\
&= \sum_{i=1}^N \int_{\Omega^N} \chi_k(d\omega_N)\beta(\omega_N)\Lambda(\zeta_i, \xi_i, d\zeta) \\
&= \int_{\Omega^N} \chi_k(d\omega_N) \left\{ N\beta(\omega_N) \right\} \Lambda(\zeta_1, \xi_1, d\zeta).
\end{aligned}$$

which can be represented in the form of (15) with

$$\begin{aligned}\Delta_N(d\zeta) &= \int_{\Omega^N} \chi_k(d\omega_N)\Lambda(\zeta_1,\xi_1,d\zeta)\left\{N\beta(\omega_N) - \left(\int \tilde{P}_k(dz)\exp(f(z))\right)^{-1}\right\} \\ &\quad + \left(\int \tilde{P}_k(dz)\exp(f(z))\right)^{-1}\left\{\int_{\Omega^N}\chi_k(d\omega_N)\Lambda(\zeta_1,\xi_1,d\zeta) - \int_{\Omega}\tilde{P}_k(dz)\exp(f(z))Q(z,d\zeta)\right\} \\ &= (i)+(ii)\end{aligned}$$

We shall show that $(i) \to 0$ in variation for $N \to \infty$ and $(ii) = 0$. Due to Assumption 2 (d), the convergence of (i) is equivalent to the fact that $\int |v_N(\zeta)|\mu(d\zeta) \to 0$, where

$$v_N(z) = \int_{\Omega^N}\chi_k(d\omega_N)\exp(f(\zeta_1)+\xi_1)q_k(\zeta_1,z)\left\{N\beta(\omega_N) - \left(\int \tilde{P}_k(dz)\exp(f(z))\right)^{-1}\right\}.$$

To proceed, let $\gamma_N = \frac{1}{N}\sum_{i=1}^N \exp(f(\zeta_i)+\xi_i)$ and $\psi(z) = \exp(f(\zeta_1)+\xi_1)q_k(\zeta_1,z)$. Due to the symmetrical dependence of random elements $\zeta_1,...,\zeta_N$ and the independence of $\xi_1,...,\xi_N$, the random variables γ_N converge in mean for $N\to\infty$ to some random variable γ in dependent of all $\gamma_i(\omega_i)$, $y_i = f(\zeta_i)+\xi_i$, for $i\in\mathbb{N}$, and

$$\mathbb{E}\gamma = \mathbb{E}\exp(y_i) = \int \exp(f(\zeta)+\xi)\tilde{P}_k(d\zeta)F_k(d\xi).$$

Equivalently, for any $\delta_1 > 0$, there exists $N_\gamma(\delta_1) \geq 1$ such that $\mathbb{E}|\gamma_N - \gamma| < \delta_1$ for all $N \geq N_\gamma(\delta_1)$. Then,

$$\begin{aligned}|v_N(z)| &= \left|\mathbb{E}\left(\frac{\psi(z)}{\gamma_N}\right) - \frac{\mathbb{E}\psi(z)}{\mathbb{E}\gamma}\right| &(16)\\ &= \frac{1}{\mathbb{E}\gamma}\left|\mathbb{E}\left(\frac{\psi(z)\gamma}{\gamma_N}\right) - \mathbb{E}\psi(z)\right| &(17)\\ &\leq \exp(c_f)\left|\mathbb{E}\left(\frac{\psi(z)|\gamma-\gamma_N|}{\gamma_N}\right)\right| &(18)\\ &\leq \exp(2c_f)\|\psi\|_\infty \mathbb{E}|\gamma-\gamma_N| &(19)\\ &\leq L_k\exp(3c_f+c_\xi)\mathbb{E}|\gamma-\gamma_N|, &(20)\end{aligned}$$

where the second equality is due to the independence of γ from γ_N and ψ, the first and second inequalities are due to $\gamma,\gamma_N \geq \exp(-c_f)$ (by Assumption 2 (b)), and the last relation is due to $\|\psi\|_\infty \leq \exp(f(\zeta)+\xi))L_k \leq L_k\exp(c_f+c_\xi)$. In order to show that $\int |v_N(z)|\mu(dz) \to 0$, we need to prove that for any $\delta > 0$ and $z\in\mathcal{Z}$, there exists $N^\star(\delta,z)$ such that for $N \geq N^\star(\delta,z)$, there holds $|v_N(z)| \leq \delta$. This can hold if one takes $\delta_1 = \delta L_k^{-1}\exp(-3c_f-c_\xi)$ and $N^\star(\delta,z) = N_\gamma(\delta_1)$.

Now, by (20), we have that $\int |v_N(\zeta)|\mu(d\zeta) \leq L_k\exp(3c_f+c_\xi)\mathbb{E}|\gamma-\gamma_N|$. From the central limit theorem for symmetrically dependent random variables (see **??**), it follows that $\mathbb{E}|\gamma-\gamma_N| = \mathcal{O}(N^{-1/2})$. Consequently, $\int |v_N(\zeta)|\mu(d\zeta) = \mathcal{O}(N^{-1/2})$.

To show that $(ii) = 0$, note that

$$\begin{aligned}&\int_{\Omega^N}\chi_k(d\omega_N)\Lambda(\zeta_1,\xi_1,d\zeta) - \int_\Omega \tilde{P}_k(dz)\exp(f(z))Q(z,d\zeta) \\ &= \int_{\mathcal{Z}}\tilde{P}_k(dz)\exp(f(z))Q(z,d\zeta)\left\{\int \exp(\xi)\,F_k(d\xi) - 1\right\},\end{aligned}$$

which is 0 by Assumption 2 (a). Hence, we have concluded the proof. □

Approximation power of solution functions

To solve (1), a structural assumption is made in (4) to search within $\Pi = \{\pi_\zeta : \zeta\in\mathbb{R}^d\}$, i.e., solution functions of a convex optimization problem. The quality of the solution depends on the approximation error of the solution function class as well as the optimization error in solving (4). While the objective of our work is primarily focused on developing an evolutionary algorithm, which has been detailed in the main paper, we also provide some preliminary study on the expressivity of this important class of functions from an approximation theory viewpoint (DeVore and Lorentz 1993).

To provide meaningful discussions of the approximation rate, we state the assumptions on the function class (commonly referred to as model class assumptions). Consider the class C^2 as the set of functions that are continuously differentiable up to order 2. Without loss of generality, we assume the input space $\mathcal{S} \coloneqq [0,1]^{n_s}$, and we only consider scalar valued functions, since vector-valued function can be treated as concatenation of scalar-valued counterparts. To measure the distortion rate, we

consider the uniform error as $\|f - \tilde{f}\|_\infty = \max_{s\in\mathcal{S}} \left|f(s) - \tilde{f}(s)\right|$. For the purpose of demonstrating the expressivity of this class of functions, we restrict the function class Π to be the set of solution functions of linear programming (LP) without fixing the number of constraints or variables.

Theorem 2 (Universal approximation of C^2 functions)**.** *For any target function $f \in C^2$, there is a solution function $\pi \in \Pi$ of an LP with $\mathcal{O}\Big(\left(\frac{n_s}{\epsilon}\right)^{\frac{n_s}{2}}\Big)$ constraints and $n_s + 1$ variables such that $\|f - \pi\|_\infty \le \epsilon$.*

The above result establishes a universal approximation theory of the solution functions of LPs with a constructive proof. The complexity of the construction is analyzed and compared in terms of the total number of variables and constraints to obtain an ϵ accuracy. It is well known that neural networks have this universal approximation capacity (Hornik, Stinchcombe, and White 1989); for instance, to achieve ϵ approximation accuracy to a C^2 smooth function on n_s dimension, one needs $\mathcal{O}(\epsilon^{n_s/2})$ number of neurons (Mhaskar 1996; **?**). Theorem 2 puts the class of solution functions on the same grounding of neural networks in terms of approximation capability to justify its consideration as policy functions in complex decision-making tasks.

We begin with some notations. Let $\mathcal{C}_{\mathcal{S},B,L}$ denote the class of convex, bounded, subdifferentiable, and uniformly Lipschitz functions on the set $\mathcal{S}$:

$$\mathcal{C}_{\mathcal{S},B,L} \coloneqq \left\{ f : \mathcal{S} \to \mathbb{R} \,\middle|\, f \text{ is convex, } \|f\|_\infty \le B, \text{and } \|v\|_\infty \le L,\ \forall v \in \partial f(s) \right\}, \tag{21}$$

with positive scalars $B, L > 0$. We also introduce the class of max-affine functions that are uniformly bounded and uniformly Lipschitz with at most $K \in \mathbb{N}$ hyperplanes:

$$\mathcal{M}^K_{\mathcal{S},B,L} \coloneqq \left\{ h : \mathcal{S} \to \mathbb{R} \,\middle|\, h(s) = \max_{k=1,\dots,K} p_k^\top s + q_k,\ \|p_k\|_\infty \le L,\ h(s) \in [-B_d, B],\ \forall s \in \mathcal{S} \right\}, \tag{22}$$

where $B_d \coloneqq B + n_s L$. We also denote $\operatorname{diam}(\mathcal{S}') \coloneqq \max_{s,s'\in\mathcal{S}'} \|s - s'\|_\infty$ as the diameter of the set $\mathcal{S}$. Recall that $\operatorname{diam}(\mathcal{S}) \coloneqq 1$ by the assumption that $\mathcal{S} \coloneqq [0,1]^{n_s}$.

Lemma 5. *Any function $h \in \mathcal{M}^K_{\mathcal{S},B,L}$ is equivalent to a solution function of an LP with K constraints and $n_s + 1$ variables. Furthermore, for any function of the form $f = h_1 - h_2$, where $h_1, h_2 \in \mathcal{M}^K_{\mathcal{S},B,L}$ is equivalent to a solution function of an LP with $2K + 1$ constraints and $n_s + 3$ variables.*

Proof. Suppose $h(s) = \max_{k=1,\dots,K} p_k^\top s + q_k$. It can be seen that by introducing an extra variable t and K constraints in the form of $p_k^\top s + q_k \le t$ and change the maximum to minimum, we have constructed an equivalent optimization with solution equal to $h(s)$. This is well-known as the epigraph formulation of the optimization. The construction for $f = h_1 - h_2$ is performed by introducing t_i and K constraints for each h_i, $i = 1, 2$, in addition to an extra variable t_3 and an extra constraint $t_1 + t_2 \le t_3$, with the objective to minimizes over t_3. □

By (Bačák and Borwein 2011), any function $f \in C^2$ can be written as a DC function: $f = \phi_1 - \phi_2$, where $\phi_i \in C_{\mathcal{S},B,L}$. For any $s \in \mathcal{S}$ and convex function ϕ, let $\nabla\phi(s) \in \partial\phi(s)$ be an arbitrary fixed subgradient of ϕ at s. For any $t > 0$ and $i = 1, 2$, define $R_t \coloneqq 1 + 2tL$, $\nu_i(s) \coloneqq s + t\nabla\phi_i(s)$ combines the point s and $\nabla\phi_i(s)$ weighted by t, and define $\mathcal{K}_i \coloneqq \{\nu_i(s) : s \in \mathcal{S}\} \subset \mathbb{R}^{n_s}$ as an expanded set of $\mathcal{S}$ along the direction $\nabla\phi_i$. Note that since the subgradient of a convex function is monotone, $\nu_i(\cdot)$ is strictly monotone, and $\nu_i(s) \ne \nu_i(y)$ for any $s \ne y$. This also implies that $\nu_i(\cdot)$ is a bijection and its inversion is well-defined. Let $\mathcal{K}_{\epsilon,i} \subseteq \mathcal{K}$ be an $\sqrt{\epsilon}$-net of set $\mathcal{K}_i$ with respect to Euclidean norm $\|\cdot\|$, and $\mathcal{S}_{\epsilon,i} \triangleq \left\{\nu_i^{-1}(z) \in \mathcal{S} : z \in \mathcal{K}_{\epsilon,i}\right\}$ be its preimage corresponding to the mapping ν_i for $i = 1, 2$. Since $R_t \ge \operatorname{diam}(\mathcal{K}_i)$, by standard covering number argument (Wainwright 2019) and the fact that $\|s\| \le \sqrt{n_s}\|s\|_\infty$ for any s, $|\mathcal{K}_{\epsilon,i}| = |\mathcal{S}_{\epsilon,i}| \le \left(9n_s R_t^2/\epsilon\right)^{n_s/2}$ for all $\epsilon \in (0, 9n_s R_t^2]$. Note that since $\mathcal{K}_{\epsilon,i}$ is an $\sqrt{\epsilon}$-net of set $\mathcal{K}_i$, by definition, for any $s \in \mathcal{S}$, there exists $\hat{s}_i \in \mathcal{S}_{\epsilon,i}$ such that $\|\nu_i(s) - \nu_i(\hat{s}_i)\| \le \sqrt{\epsilon}$. Hence,

$$\begin{aligned}
&\|s - \hat{s}_i\|^2 + t^2\|\nabla\phi_i(s) - \nabla\phi_i(\hat{s}_i)\|^2 \\
&\le \|s - \hat{s}_i\|^2 + 2t(s - \hat{s}_i)^\top(\nabla\phi_i(s) - \nabla\phi_i(\hat{s}_i)) + t^2\|\nabla\phi_i(s) - \nabla\phi_i(\hat{s}_i)\|^2 \\
&= \|\nu_i(s) - \nu_i(\hat{s}_i)\|^2 \\
&\le \epsilon,
\end{aligned}$$

where the first inequality is due to the convexity of ϕ_i. This implies that for any $s \in \mathcal{S}$, there exists $\hat{s}_i \in \mathcal{S}_{\epsilon,i}$ such that $\|s - \hat{s}_i\|$ is controlled by $\sqrt{\epsilon}$ and $\|\nabla\phi_i(s) - \nabla\phi_i(\hat{s}_i)\|$ is bounded by $\sqrt{\epsilon}/t$.

Now, consider $K \coloneqq \left(18n_s R_t^2/\epsilon\right)^{n_s/2}$ and set $\mathcal{S}_{K,i} \triangleq \left\{\hat{x}_1^{(i)}, \dots, \hat{x}_K^{(i)}\right\} \subseteq \mathcal{S}$ such that $\mathcal{S}_{\epsilon/2,i} \subseteq \mathcal{S}_{K,i}$. Then, we introduce the following piecewise affine function $h : \mathcal{S} \to \mathbb{R}$ as

$$h(s) = \max_{k=1,\dots,K} \left\{\phi_1(\hat{s}_k^{(1)}) + \nabla\phi_1(\hat{s}_k^{(1)})^\top(s - \hat{s}_k^{(1)})\right\} - \max_{k=1,\dots,K} \left\{\phi_2(\hat{s}_k^{(2)}) + \nabla\phi_2(\hat{s}_k^{(2)})^\top(s - \hat{s}_k^{(2)})\right\}.$$

Hence, for any $s \in \mathcal{S}$, we have that

$$\begin{aligned}|f(s) - h(s)| &\leq \left|\phi_1(s) - \max_{k=1,\dots,K}\left\{\phi_1(\hat{s}_k^{(1)}) + \nabla\phi_1(\hat{s}_k^{(1)})^\top(s - \hat{s}_k^{(1)})\right\}\right| \\ &\quad + \left|\phi_2(s) - \max_{k=1,\dots,K}\left\{\phi_2(\hat{s}_k^{(2)}) + \nabla\phi_2(\hat{s}_k^{(2)})^\top(s - \hat{s}_k^{(2)})\right\}\right| \\ &= \sum_{i=1,2}\phi_i(s) - \max_{k=1,\dots,K}\left\{\phi_i(\hat{s}_k^{(i)}) + \nabla\phi_i(\hat{s}_k^{(i)})^\top(s - \hat{s}_k^{(i)})\right\},\end{aligned}$$

where the last equality is because the function $\max_{k=1,\dots,K}\left\{\phi_i(\hat{s}_k^{(i)}) + \nabla\phi_i(\hat{s}_k^{(i)})^\top(s - \hat{s}_k^{(i)})\right\}$ is a uniform lower bound of ϕ_i by convexity. Let us define the selective function

$$k_i(s) = \operatorname*{argmin}_{k=1,\dots,K}\left\|\nu_i^{-1}(s) - \nu_i^{-1}(\hat{s}_k^{(i)})\right\|,$$

which selects the index of the point in $\mathcal{S}_{K,i}$ such that $s_k^{(i)}$ is closest to s as measured by ν_i^{-1}. Since $\mathcal{S}_{K,i}$ is the preimage of $\mathcal{K}_{\epsilon/2,i}$, which is, by definition, an $\epsilon/2$-cover of $\mathcal{K}_i$, we have that Then, since $\mathcal{S}_{K,i}$ is an ϵ

$$\begin{aligned}|f(s) - h(s)| &\leq \sum_{i=1,2}\phi_i(s) - \phi_i(\hat{s}_{k_i(s)}^{(i)}) + \nabla\phi_i(\hat{s}_{k_i(s)}^{(i)})^\top(s - \hat{s}_{k_i(s)}^{(i)}) \\ &\leq \sum_{i=1,2}\|\nabla\phi_i(s) - \nabla\phi_i(\hat{s}_{k_i}^{(i)})\|\|s - \hat{s}_{k_i}^{(i)}\| \\ &\leq \frac{\epsilon}{t},\end{aligned}$$

where the first inequality is by plugging in $k_i(s)$ to into the maximum operator, the second inequality is due to Cauchy-Schwarz, and the last inequality follows from the fact that $\|s - \hat{s}_i\|$ is controlled by $\sqrt{\epsilon/2}$ and $\|\nabla\phi_i(s) - \nabla\phi_i(\hat{s}_i)\|$ is bounded by $\sqrt{\epsilon/2}/t$ by the aforementioned reasoning. Therefore, we have shown that h can uniformly approximate f by accuracy ϵ/t.

From $K := \left(18n_sR_t^2/\epsilon\right)^{n_s/2}$, we have $\epsilon = 18n_sR_t^2K^{-2/n_s}$. Therefore,

$$\|f - h\|_\infty \leq \frac{\epsilon}{t} = \frac{18n_sR_t^2}{t}K^{-2/n_s}.$$

Optimizing over t optimal, we obtain $t^* = \frac{1}{2L}$. Therefore, by choosing $K^* = \left(\frac{\epsilon}{144n_sL}\right)^{\frac{-n_s}{2}}$, we have that $\|f - h\|_\infty \leq \epsilon$. The proof is concluded by recalling Lemma 5.

Additional details for CityLearn

Details of optimization model

We refer the reader to (Vazquez-Canteli et al. 2020) and the corresponding online documentation[3] for the detailed setup of the competition. We will only focus on our strategy in this document. In particular, we provide details regarding the construction of the optimization model in 3. 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 actions for the future hours up to the end of the day and execute the plan for the upcoming hour r, a.k.a., rolling-horizon planning. Next, we describe the hyperparameters, variables, objective, and constraints in 3.

Hyperparameters. The hyperparameters are required to instantiate an optimization and are not part of the optimization variables to be solved by an optimization algorithm.

- The hyperparameters to be set by prior knowledge include: *(1)* electric heater: efficiency η_{ehH}, nominal power $E_{\text{max}}^{\text{ehH}}$; *(2)* heat pump: technical efficiency $\eta_{\text{tech}}^{\text{hp}}$, target cooling temperature t_c^{hp}, nominal power $E_{\text{max}}^{\text{hpc}}$; *(3)* electric battery: rate of decay Cf^{bat}, capacity Cp^{bat}, efficiency η_t^{bat}; *(4)* heat storage: rate of decay Cf^{Hsto}, capacity Cp^{Hsto}, efficiency η_t^{Hsto}; *(5)* cooling storage: rate of decay Cf^{Csto}, capacity Cp^{Csto}, efficiency η_t^{Csto}.
- The hyperparmeters provided by predictors include: *(1)* hourly coefficient of performance (COP) of heat pump $\text{COP}_t^C = \eta_{\text{tech}}^{\text{hp}}\frac{t_c^{\text{hp}}+273.15}{\text{temp}_t - t_c^{\text{hp}}}$, where temp_t is the predicted outside temperature for hour t; *(2)* solar generation E_t^{PV}; *(3)* electricity non-shiftable load E_t^{NS}; *(4)* heating demand H_t^{bd}; and *(5)* cooling demand C_t^{bd}. At hour r, the predictions of the above are required for hour $r \leq t \leq T$. In our algorithm, the predictions are provided by simple averaging of past 2 weeks data in the corresponding hour.

[3]link: `https://sites.google.com/view/citylearnchallenge`

- The hyperparameters to be learned by Algorithm 1 are the virtual electricity price $\{\theta_t\}_{t=1,\dots,24}$ for 24 hours. These values are bounded between $[0, 10]$.

Optimization variables. The variables for the optimization at hour r include:

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

The actions of the policy at hour r are a_r^{bat}, a_r^{Hsto}, and a_r^{Csto}. The rest of the variables are considered as auxiliary variables for planning.

Objective function. The objective function is given by:

$$|E_t^{\text{grid}} - E_{t-1}^{\text{grid}}| + \theta_t E_t^{\text{grid}} + \sum_{t'=t+1}^{T} \Big(|E_{t'}^{\text{grid}} - E_{t'-1}^{\text{grid}}| + \theta_{t'} E_{t'}^{\text{grid}} \Big). \tag{23}$$

Note that we use e_t for E_t^{grid} in the main text. Also, the above objective is used in a standard minimization problem; to make it consistent with the maximization problem in (3), we can take the negation of the value.

Constraints. The constraints include both energy balance constraints and technology constraints.

Energy balance constraints:

- Electricity balance for each hour $t \geq r$:
  $E_t^{\text{PV}} + E_t^{\text{grid}} = E_t^{\text{NS}} + E_t^{\text{hpC}} + E_t^{\text{ehH}} + a_t^{\text{bat}} C_p^{\text{bat}}$
- Heat balance for each hour $t \geq r$:
  $E_t^{\text{ehH}} = a_t^{\text{Hsto}} C_p^{\text{Hsto}} + H_t^{\text{bd}}$
- Cooling balance for each hour $t \geq r$:
  $E_t^{\text{hpC}} \text{COP}_t^{\text{C}} = a_t^{\text{Csto}} C_p^{\text{Csto}} + C_t^{\text{bd}}$

Heat pump technology constraints:

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

Electric heater technology constraints:

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

Electric battery technology 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 $t \geq r$:
  $SOC_t^{\text{bat}} = (1 - C_f^{\text{bat}}) SOC_{t-1}^{\text{bat}} + a_t^{\text{bat}} \eta^{\text{bat}}$
- Action limits for each hour $t \geq r$:
  $-1 \leq a_t^{\text{bat}} \leq 1$
- Bounds of SOC or each hour $t \geq r$:
  $0 \leq SOC_t^{\text{bat}} \leq 1$

Heat storage technology 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 $t \geq r$:
  $-1 \leq a_t^{\text{Hsto}} \leq 1$
- Bounds of SOC or each hour $t \geq r$:
  $0 \leq SOC_t^{\text{H}} \leq 1$

Cooling storage technology 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 $t \geq r$:
  $-1 \leq a_t^{\text{Csto}} \leq 1$
- Bounds of SOC or each hour $t \geq r$:
  $0 \leq SOC_t^{\text{C}} \leq 1$

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).

Additional experimental results

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 5.

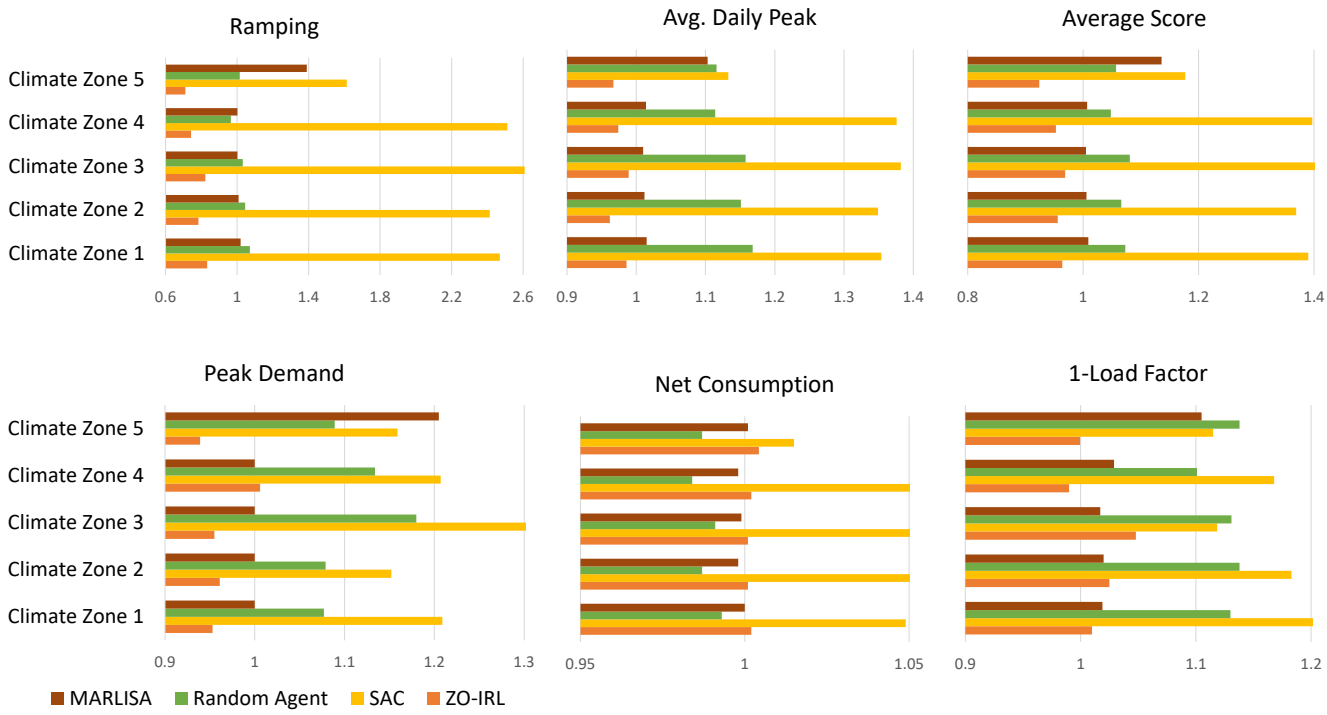

Figure 5: Scores for ZO-iRL and comparison with other methods, including SAC (Kathirgamanathan et al. 2020) and MARLISA (Vazquez-Canteli, Henze, and Nagy 2020) for different climate zones. The random agent basically uniformly selects an action within the range at each timestep.