# Winning the CityLearn Challenge: Adaptive Optimization with Evolutionary Search under Trajectory-based Guidance

**Vanshaj Khattar and Ming Jin**

Electrical and Computer Engineering, Virginia Tech

## Abstract

Modern power systems will have to face difficult challenges in the years to come: frequent blackouts in urban areas caused by high power demand peaks, grid instability exacerbated by intermittent renewable generation, and global climate change amplified by rising carbon emissions. While current practices are growingly inadequate, the path to widespread adoption of artificial intelligence (AI) methods is hindered by missing aspects of trustworthiness. The CityLearn Challenge is an exemplary opportunity for researchers from multiple disciplines to investigate the potential of AI to tackle these pressing issues in 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 function of optimization as policies to compute actions for sequential decision-making, while notably adapting the parameters of the optimization model from online observations. Algorithmically, this is achieved by an evolutionary algorithm under a novel trajectory-based guidance scheme. Formally, the global convergence property is established. Our agent ranked first in the latest 2021 CityLearn Challenge, being able to achieve superior performance in almost all metrics while maintaining some key aspects of interpretability.

## 1 Introduction

Rapid urbanization in the past decades has led to a 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 (Rolnick et al. 2022). The cornerstone of addressing emerging issues is the deployment of proper control and coordination strategies, which have a potential impact on enhancing energy flexibility and resilience in the face of a surge in climate-induced demand (as already seen in places like California, where rolling blackouts are increasingly frequent during the summer) (Vazquez-Canteli et al. 2020). Current industry practice is heavily based on optimization models, such as energy dispatch and unit commitment, where parameters (e.g., technological and physical constraints) are fixed throughout the lifecycle; however, such an approach is increasingly confronted by environmental uncertainty, renewable generation stochasticity, and the ever-increasing complexity of the distribution grid (Abedi, Gaudard, and Romerio 2019). On the other hand, there has been a surge in machine learning research, notably RL, because it allows the agent to act without the need to access the true model—a feature of particular interest for large-scale, complex systems, where it is not cost-effective to develop models of such high fidelity. Despite recent progress, real-world RL is still in its infancy (Dulac-Arnold et al. 2021).

Against this backdrop, the CityLearn Challenge aims to spur RL solutions for 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 with exploring and exploiting the best control strategy for energy storage distributed in a community of buildings. Performance is evaluated against standard metrics such as ramping costs, peak demands, and carbon emissions. The CityLearn encapsulates 4 of the 9 real-world RL challenges 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 quick action—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 describe our winning solution for the 2021 CityLearn Challenge based on the idea broadly categorized as *adaptive optimization*. Indeed, optimization (especially convex optimization) has become the de facto standard in industrial systems with profound theoretical foundations and various 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), 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 real-world conditions, rendering current approaches rather "rigid."

To address this fundamental limitation, we exploit that the

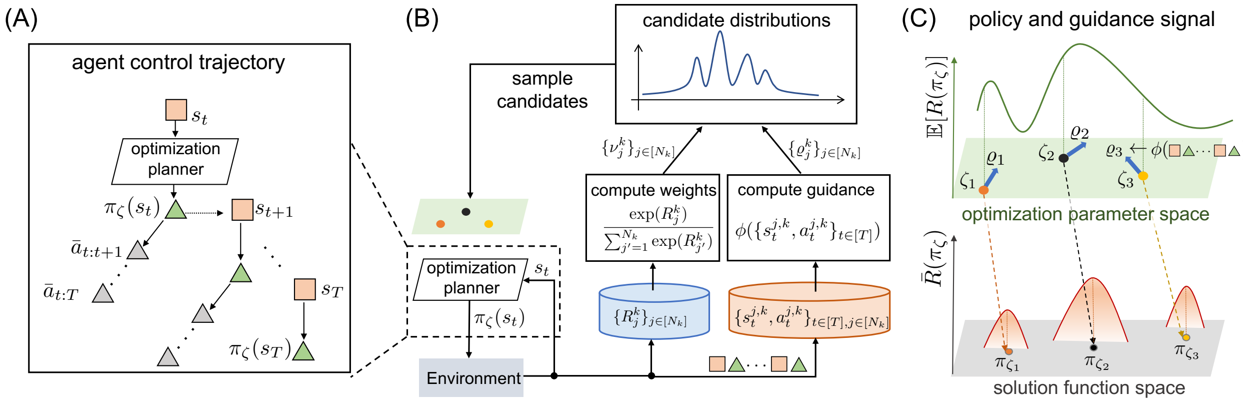


Figure 1: **Overview.** (A) Agent trajectory of observed states (square) and actual (green triangle)/planned (grey triangle) actions. At each step, the agent solves optimization (2) to plan ahead but only executes the most immediate action; $\bar{a}_{t:t'}$ represents the action planned at time $t$ for time $t' \geq t$. (B) Illustration of the system, including interaction with the environment and adaptation of optimization parameters. Each iteration of $k$ consists of evaluating $N_k$ (three in this case) policies sampled from the candidate distribution. The observed rewards $\{R_j^k\}_{j\in[N_k]}$ and trajectories $\{s_t^{j,k}, a_t^{j,k}\}_{j\in[N_k]}$ are stored in some buffers, which are then used to compute weights $\{\nu_j^k\}_{j\in[N_k]}$ and guidance signals $\{\varrho_j^k\}_{j\in[N_k]}$ to update the distribution as in (5). (C) The two-level structure indicates the correspondence between an optimization parameter $\zeta$ and the solution function $\pi_\zeta$ (used as policy). The upper level objective is the expected episodic rewards (1) and the lower level objective is the objective function of (2), the extremum of which is the policy action (see (3)). The guidance signal $\varrho$ is computed based on the trajectory of each candidate and applied on top of the original parameter during the update.

solution of 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, while extracting insights from trajectory data to design guidance signals (see Fig. 1 for a detailed illustration). Key differences between our method and well-established optimization techniques (e.g., stochastic optimization (Powell 2020), bi-level optimization (Dempe and Zemkoho 2020)) include:

*1)* we only allow access to the environment through interactive samples (reward, states, etc.) but not the true dynamics or reward function;

*2)* we make full use of the control trajectory of a Markov decision process (MDP) to obtain a guidance signal.

While *1)* is similar to the RL setup, *2)* can be viewed as an *augmentation of zeroth-order search methods with insights extracted from control data*, which, to the best of our knowledge, is the *first of its kind.* In general, zeroth-order algorithms, such as simultaneous perturbation (Spall 2005; Mania, Guy, and Recht 2018), evolutionary algorithms (Salimans et al. 2017; Zhou, Yu, and Qian 2019), and Bayesian optimization (Snoek, Larochelle, and Adams 2012; Frazier 2018) are natural candidates for RL and easy to implement, but can potentially suffer scalability problems (Ghadimi and Lan 2013). Nevertheless, the parameters of an optimization model (i.e., variables to be learned) usually have clear interpretations. Thus, we design a mechanism to leverage domain knowledge for the design of appropriate guidance during the search for these parameters; such guidance can be specified as a general function of trajectory data (including observed states and actions of an MDP). The method works well in an online environment without an extensive training period, which is particularly advantageous in a real-world setting where an 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. To demonstrate effectiveness against existing techniques, we further validate the method by comparing it with a range of baselines. Key contributions are as follows:

- A framework of adaptive optimization for online control, winning the **1st place in the 2021 CityLearn Challenge**
- A novel evolutionary search (ES) algorithm with a guidance function based on state-action-trajectory data
- Theoretical analysis of asymptotic convergence to global optima with noisy function evaluations
- Empirical comparison against a range of baselines

## 1.1 Related work

Optimal control and stochastic optimal control are well-known approaches to sequential decision-making problems (Bertsekas 2019). Convex optimization is another avenue (Agrawal et al. 2020). Most existing works assume a known dynamic model of the system. Various large-scale stochastic programs have been proposed in the literature to deal with future uncertainty (Prékopa 2013; Powell 2020). The major drawback is the computational burden of rapidly expanding scenario trees in multi-stage stochastic programming. Our method relieves computation by using plug-in estimators, a.k.a., deterministic approximation of future uncertainty within convex optimization.

Recently, RL has gained popularity in various domains (Chen et al. 2022; Haydari and Yilmaz 2020; Nian, Liu, and Huang 2020). To contextualize the present approach, we make a few remarks about the relation to model-based RL (MBRL). In particular *implicit* MBRL, where the entire procedure (e.g., model learning and planning) is optimized for optimal policy computation (Moerland, Broekens, and Jonker 2020). However, unlike existing works (e.g., (Tamar et al. 2016; Karkus, Hsu, and Lee 2017; Racanière et al. 2017; Guez et al. 2018; Schrittwieser et al. 2020) that build a model based on (recurrent/convolutional) neural networks (NNs) with primary restrictions to discrete state and action space, our method learns how to plan by solving optimization and adapting its parameters; hence, it is amenable to a wide range of applications with continuous states and actions. The present work is closely related to (Ghadimi, Perkins, and Powell 2020; Agrawal et al. 2020), which also use convex optimization as a policy class to handle uncertainty. In particular, convex optimization control policies are learned in (Agrawal et al. 2020) with implicit differentiation (Agrawal et al. 2019). We extend their method to the RL setting and propose a novel ES algorithm for learning.

Differentiated from existing ES-based strategies (Szita and Lörincz 2006; Salimans et al. 2017; Khadka and Tumer 2018; Gangwani and Peng 2018; Conti et al. 2018) or zeroth-order optimization (Snoek, Larochelle, and Adams 2012; Liu et al. 2020), our ES is augmented with a guidance function that depends on the data collected as the policy interacts with the environment; hence, it can be viewed as a type of *MDP-augmented ES*. The guidance mechanism is also flexible enough to allow the effective incorporation of domain knowledge as demonstrated in CityLearn. We further provide theoretical justification for this rather complex scheme.

# 2 Preliminaries

## 2.1 Problem setup

Consider an MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r)$, where $\mathcal{S}$ is the (possibly 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 $r(s, a)$ gives the corresponding immediate reward (can be time-dependent). The goal in episodic RL is to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes cumulative rewards over a finite time horizon:

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

where $R(\pi) := \sum_{t=0}^{T} r_t\big(s_t, \pi(s_t)\big)$ 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 horizon. The expectation is taken over initial state distributions and transition dynamics (under deterministic policy $\pi$). We require access to a random sample $R(\pi)$ of the episodic reward as well as trajectory data $\{(s_t, a_t)\}_{t\in[T]}$ to be used to later compute the guidance signal. Here, we use the shorthand $[T] = \{1, ..., T\}$.

## 2.2 Canonical approaches and solution functions

The proposed method is based on canonical stochastic programming methods (Powell 2020). For example, in multi-stage stochastic programming (Pflug and Pichler 2014), the action at state $s_t$ is computed as

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

where $\tilde{r}_t : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the surrogate reward function and $\tilde{\mathbb{E}}[\cdot]$ is the surrogate expectation operator (e.g., model-based scenario trees) designed to approximate the true environment. The above formulation can also be seen as finding the solution to a Bellman equation in dynamic programming. The main limitations, nevertheless, are the high computational cost of evaluating the expectation operator and the potential model mismatch due to approximations.

A simpler yet more practically appealing method, widely adopted in today's industries, 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): $\pi_\zeta(s_t)$ is given by

$$\begin{aligned} \arg\max_{\bar{a}_t \in \mathcal{A}} \quad \max_{\{\bar{s}_{t'}, \bar{a}_{t'}\}_{t'=t+1}^{T}} \quad & \bar{R}(s_t, \bar{a}_t, \{\bar{s}_{t'}, \bar{a}_{t'}\}_{t'=t+1}^{T}; \zeta) \\ \text{s. t.} \quad & g_i(s_t, \bar{a}_t, \{\bar{s}_{t'}, \bar{a}_{t'}\}_{t'=t+1}^{T}; \zeta) \le 0 \;\; ; \quad \forall i \in \mathcal{I} \\ & h_i(s_t, \bar{a}_t, \{\bar{s}_{t'}, \bar{a}_{t'}\}_{t'=t+1}^{T}; \zeta) = 0 \;\; ; \quad \forall i \in \mathcal{E}, \end{aligned} \tag{2}$$

with the objective $\bar{R}(s_t, \bar{a}_t, \{\bar{s}_{t'}, \bar{a}_{t'}\}_{t'=t+1}^{T}; \zeta)$ defined as

$$\bar{r}_t\big(s_t, \bar{a}_t; \zeta\big) + \sum\nolimits_{t'=t+1}^{T} \bar{r}_{t'}\big(\bar{s}_{t'}, \bar{a}_{t'}; \zeta\big),$$

where $\{\bar{s}_{t'}\}_{t'=t+1}^{T}$ and $\{\bar{a}_{t'}\}_{t'=t}^{T}$ are optimization variables corresponding to the planned states and actions, $\bar{r}_t$ is surrogate reward, and the feasible set is defined by $\{g_i\}_{i\in\mathcal{I}}$ and $\{h_i\}_{i\in\mathcal{E}}$. 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 (2) as part of the lookahead model. Many examples can be found in, e.g., (Borrelli, Bemporad, and Morari 2017). We remark that some optimization parameters may be provided by predictors based on the current state $s_t$, the parameter of which is also collected by $\zeta$ in this case.

The policy $\pi_\zeta(s_t)$, a.k.a., the solution function (Dontchev and Rockafellar 2009), provides the action at the current state $s_t$ as the optimal solution to (2). Since this function is generally set-valued (Dontchev and Rockafellar 2009), we make the following assumption.

**Assumption 1.** *For each $\zeta \in \mathcal{Z}$ and $s_t \in \mathcal{S}$:* a) *the objective function in* (2) *is continuous, strictly convex, $g_i$ is continuous and convex for each $i \in \mathcal{I}$, and $h_i$ is affine for each $i \in \mathcal{E}$;* b) *the feasible set of* (2) *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 (2). Note that in our approach, we make no convexity assumption about the true dynamics or rewards of the environment, which can be seen as a blackbox. The convexity condition is only stipulated for the surrogate model for computational efficiency. Our goal is simply to learn the parameters of the optimization model in order to have a good decision-making capability.[1] An immediate consequence of the above assumption is that the solution to (2) is unique; furthermore, it implies continuity with respect to parameters.

**Lemma 1.** *The solution function $\pi_\zeta(s_t)$ defined in* (2) *is continuous with respect to parameter $\zeta$ for each $s_t \in \mathcal{S}$.*

The proof is a direct application of the Berge maximum theorem (Berge 1997). To conclude this section, let us make some comments on the construction of the surrogate model in (2). By analogy to reward design (Prakash et al. 2020), the objective function should be chosen to promote desirable behaviors. The set of constraints introduces 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 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 (2) to provide actions quickly—hence the choice of convex programs.

## 3 Policy adaptation with ES under guidance

The potential mismatch between the surrogate model and the real environment and errors due to predictions may adversely affect the decision quality of (2). 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:

$$\Upsilon := \arg\max_{\zeta \in \mathcal{Z}} \mathbb{E}\left[\sum_{t=0}^{T} r_t\big(s_t, \pi_\zeta(s_t)\big)\right], \tag{3}$$

where $\Upsilon$ is the set of global optima for policy parameters. Note that $\zeta$ is not 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 (2). Since $\pi_\zeta(s_t)$ is given by an optimization (2), (3) can also be viewed as a bi-level problem (c.f., (Dempe and Zemkoho 2020)): the outer level aims to learn parameters to maximize rewards, while the inner level defines policy action as a solution to (2). The key challenge in solving (3) as a bi-level problem, however, is that the outer level objective is not analytically revealed (thus prohibiting any direct differentiation approach) and can be nonconvex with respect to $\zeta$.

### 3.1 Guided evolutionary search

**Algorithm 1:** Evolutionary search under guidance

```
Input: Hyperparameters $\{N_k\}$, uniform distribution
       $\mu$, initial point $z_0$
Initialize $P_1(d\zeta) \sim \exp(\|\zeta - z_0\|)\mu(d\zeta)$
for $k = 1, 2, \dots$ do
    Sample $N_k$ candidates from the distribution $p_k$:
    $\zeta_1^k, \zeta_2^k, \cdots, \zeta_{N_k}^k \overset{\text{iid}}{\sim} p_k$
    for $j = 1, \dots, N_k$ do
        Deploy policy $\pi_{\zeta_j^k}$ for one episode and observe an
        episodic reward $R_j^k \leftarrow R(\pi_{\zeta_j^k})$
        Compute the guidance signal $\varrho_j^k$ by (4)
    end for
    Update the distribution $p_{k+1}$ for the next iteration
    according to (5).
end for
```

This section discusses the proposed ES algorithm (Algorithm 1), inspired by the method of generations (Zhigljavsky 2012). 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 $j \in [N_k]$, we evaluate the corresponding policy in the environment and observe an episodic reward $R_j^k \sim \mathcal{R}(\pi_{\zeta_j^k})$, where $\mathcal{R}(\pi_{\zeta_j^k})$ denotes the distribution of episodic reward for policy $\pi_{\zeta_j^k}$, as well as all the state and action pairs $\{s_t^{j,k}, a_t^{j,k}\}_{t \in [T]}$ in the past episode. Based on trajectory data, we compute a guidance signal

$$\varrho_j^k = \phi(\{s_t^{j,k}, a_t^{j,k}\}_{t \in [T]}), \tag{4}$$

where $\phi : (\mathcal{S} \times \mathcal{A})^T \to \mathcal{Z}'$ is a function that may have complicated dependence on past states and actions with $\mathcal{Z}'$ as the range. The design of such a guidance function is often based on domain knowledge (to be discussed later in CityLearn). Then, we update the distribution for the next iteration as

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

where

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

are the weights obtained by taking the softmax over candidate rewards. The probability measure $Q_k(\zeta_j^k, \varrho_j^k, d\zeta)$ is the transition probability given candidate $\zeta_j^k$ and guidance signal $\varrho_j^k$. 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 $\{\nu_j^k\}$, followed by sampling from $Q_k(\zeta_j^k, \varrho_j^k, d\zeta)$. For example, in our algorithm for CityLearn,

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

where $\|\cdot\|$ is the Euclidean norm, $\mu(d\zeta)$ is a uniform measure over $\mathcal{Z}$, and $\iota_k > 0$ and $\alpha_k \geq 0$ are such that their sum over time is bounded. Other candidates are possible and can still ensure convergence to global optimal, as long as certain conditions are met; intuitively, we require that the span of $Q_k$ decreases over time but not so rapidly that it fails to reach a global optimum. Note that to simplify the presentation, in

[1] Perhaps surprisingly, despite the fact that (2) is convex, the policy (given by the solution function) can be highly nonconvex with high representational capacity. In particular, the contemporary work by (Jin et al. 2023) shows a "universal approximation" property of the solution functions of linear programs (LPs).

the above algorithm, we assume that each candidate policy is evaluated only on one episode; extending this to the case of multiple episodes is straightforward (e.g., we would instead take the average of the evaluations among the episodes in the computation of weights (6)).

## 4 Theoretical analysis

We now analyze the convergence property of the sequence generated by Algorithm 1. The following notations are used: $f(\zeta) = \mathbb{E}[R(\pi_\zeta)]$ is the expected episodic reward of policy $\pi_\zeta$, $\Upsilon = \arg\max_{\zeta \in \mathcal{Z}} f(\zeta)$ 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 $\Upsilon$; $\mathbb{B}(\zeta, \epsilon) = \{\zeta' \in \mathcal{Z} : \|\zeta' - \zeta\| \leq \epsilon\}$ is a ball centered at $\zeta$ with radius $\epsilon$, $\mathbb{B}^*(\epsilon) = \{\zeta \in \mathcal{Z} : \min_{\zeta' \in \Upsilon} \|\zeta' - \zeta\| \leq \epsilon\}$ is a set of points that are $\epsilon$ away from the optimal solution set $\Upsilon$; $\delta_\zeta(dz)$ is the probability measure concentrated at the point $\zeta$. We use $\Rightarrow$ to denote the weak convergence of measures. We can consider $\|\cdot\|$ as any norm (e.g., Euclidean norm).

The measures $p_{k+1}(d\zeta)$, $k \in \mathbb{N}$ defined in (5) are distributions of random points $\zeta_j^{k+1}$, for any $j \in [N_{k+1}]$, conditional on the results of preceding evaluations of $\{R_j^k\}_{j \in [N_k]}$ and realizations of $\{\zeta_j^k, \varrho_j^k, \xi_j^k\}_{j \in [N_k]}$. Let $P_k(d\zeta_1, ..., d\zeta_{N_k})$ represent their unconditional joint distributions at iteration $k$, and

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

is the unconditional marginal distribution (note that we introduce $z$ for $\zeta$ as the need arises in integration).

The formalism of the guidance signal requires some basics of random process and measure theory (details can be found in the appendix). Essentially, the guidance signal $\varrho_j^k \in \mathcal{Z}'$ is a random variable (adapted to the $\sigma$-algebra generated by the trajectory within an episode) with probability measure $M_k(\zeta_j^k, d\varrho)$. Note that $M_k(\zeta_j^k, d\varrho)$ is dependent on $\zeta_j^k$ because the stochastic process that generates the trajectory depends on policy $\pi_{\zeta_j^k}$, but $M_k(\zeta_j^k, d\varrho)$ is conditionally independent of all other candidates $\{\zeta_{j'}^k\}_{j' \neq j}$. For analysis, 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 with distribution* $F_k(d\xi)$ *bounded on a finite interval* $[-c_\xi, c_\xi]$ *and* $\mathbb{E}\exp(\xi_j^k) = 1$*;*

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

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

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

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

*(f)* $\sup_{z,\varrho} M_k(z, d\varrho) < \infty$ *for all* $k \in \mathbb{N}$*;*

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

*(h) 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 noise be independent and bounded; the expectation requirement can be satisfied for truncated log-normal distributions (Thompson 1950). The iid requirement can be relaxed to mixing processes at the cost of more complex analysis (Doukhan 2012); the boundedness condition, on the other hand, seems necessary to keep iterates in the vicinity of global maximum if they are already there. Condition $(b)$ is non-restrictive for practical problems. Condition $(c)$ is natural since $\pi_\zeta$ is continuous by Lemma 1, and can be satisfied if the true reward functions $r_t$ are continuous for all $t \in [T]$. Assumptions $(d)$, $(e)$, $(f)$, $(g)$, and $(h)$ formulate necessary requirements on the parameters of the algorithm. Intuitively, conditions $(d)$, $(e)$, and $(f)$ stipulate that the search becomes more "focused" over time in order to concentrate on the global optima; however, conditions $(g)$ and $(h)$ indicates that the decrease of span cannot be too fast in order not to miss the global optima. Condition $(f)$ on the guidance function $\phi$ can be met by proper smoothing if needed. Condition $(e)$ can be relaxed to $N_k = N$ for some finite integer $N$ for all $k \in \mathbb{N}$, but the convergence will only be towards the vicinity of $\Lambda$ 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 $(c)$, $(d)$, $(e)$, $(f)$, and $(g)$, condition $(h)$ is not constructive; hence, we provide some verifiable conditions sufficient for $(h)$ to hold in Corollary 1 (also see Corollary 2 in the appendix). Next, we analyze the update rule of Algorithm 1.

**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:*

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

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

$$\omega_{N_k} = \{z_1, ..., z_{N_k}, \varrho_1, ..., \varrho_{N_k}, \xi_1, ..., \xi_{N_k}\} \in \Omega^{N_k},$$

$$\chi_k(d\omega_{N_k}) = P_k(dz_1, ..., dz_{N_k}) \prod_{j=1}^{N_k} F_k(d\xi_j) M_k(z_j, d\varrho_j),$$

$$\beta(\omega_{N_k}) = \frac{1}{\sum_{j=1}^{N_k} \exp(f(z_j) + \xi_j)}, \quad \text{and}$$

$$\Lambda(z, \varrho, \xi, d\zeta) = \exp(f(z) + \xi) Q_k(z, \varrho, d\zeta).$$

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

Now, we provide the main result on the convergence of $\tilde{P}_k(d\zeta)$ to some distribution $\lambda(d\zeta)$ over the global optima.

**Theorem 1.** *Suppose that Assumption 2 holds true, and let* $\{\tilde{P}_k\}$ *be the sequence of unconditional marginal distributions determined by Algorithm 1. Then, the distribution sequence* *weakly converges to some measure $\lambda$ over the optimal set, i.e., $\tilde{P}_k \Rightarrow \lambda$ as $k \to \infty$.*

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

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

for some measure $\kappa$, where $m$ is the index of the subsequence. The above distribution is effectively a softmax function over function values and converges to the extrema as $m \to \infty$.

All the conditions in Assumption 2 are natural with the exception of $(h)$, which requires some further justification. In the following, we present a sufficient condition for $(h)$ with a proper design of $Q_k(z, \varrho, d\zeta)$, which applies to the case of noisy function evaluations; see the appendix for another example in the case of noiseless function evaluations.

**Corollary 1.** *Under Assumption 2 (except $(h)$), and let the transition probability $Q_k(z, \varrho, d\zeta)$ be*

$$Q_k(z, \varrho, d\zeta) = c_k(z, \varrho)\psi((\zeta - z - \alpha_k \varrho)/\iota_k)\mu(d\zeta), \quad (9)$$

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

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

*Then, there exists a sequence of natural numbers $\{N_k\}$ such that $\{\tilde{P}_k\}$ weakly converges to $\lambda$.*

Our analysis accounts for the effect of trajectory-based guidance, which is a novel contribution to the ES literature. By examining the proof, we can relax the condition that $\sum_{k=1}^{\infty} \alpha_k < \infty$, i.e., continue applying the guidance without the need to diminish its impact in the long run, as long as the guidance signal "approximately" points to the global optima in the proximity (see the appendix for exact conditions). However, such guidance can be difficult to design or even verify in practice; thus, it is still advisable to relinquish human knowledge and let data drive the decision, eventually.

## 5 Results from the CityLearn Challenge

*Challenge overview.* The competition has an online setup with a simulation period of 1 or 4 years, where agents exploit the best policies to optimize the coordination strategy. The goal of each agent is to minimize environmental costs, such as ramping costs, peak demands, 1-load factor, and carbon emissions. The state space contains information such as daylight hours, outdoor temperature, storage device state of charge (SOC), net electricity consumption of the building, carbon intensity of the power grid, among a total of 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 (i.e., 3 continuous actions per building). The environment is seen as a blackbox to the agent as a standard RL setup, where the transition dynamics depend on the responses of various devices (e.g., air-to-water heat pumps, electric heaters) as well as the energy loads of buildings, which include space cooling, dehumidification, DHW demand, and solar generation.

| | **ZO-iRL** (ours) | **ICD-CA** | **IDLab** | **Breakfast Club** |
|---|---|---|---|---|
| **Total score** | **0.944** | 1.070 | 1.070 | 1.130 |
| **Coord. score** | **0.915** | 1.107 | 1.098 | 1.095 |

Table 1: Total and coordination scores of the top 4 teams in 2021 CityLearn Challenge.

*Evaluation.* The submission of each team is evaluated on a set of metrics, including: *(1)* ramping: $\sum |e_t - e_{t-1}|$, where $e$ is the net electricity consumption at each time step; *(2)* 1-load factor: average net electricity load divided by maximum electricity load; *(3)* average daily peak demand; *(4)* maximum peak electricity demand; *(5)* total electricity consumed; *(6)* carbon emissions. The competition evaluates performance by computing the ratio of costs with respect to a rule-based controller (RBC)—lower ratios indicate better performances.[2] The average of the above metrics for the full simulated period is the *total score*, while the average of the metrics *(1)-(4)* is *coordination score*. The performance of the top 4 teams is listed in Table 1. Refer to (Vazquez-Canteli et al. 2020) for more details on the contest.

**ZO-iRL: zeroth-order implicit RL.** [3] As our method is designed for single-agent episodic RL, we first reduce the original task that consists of a single period of 1 or 4 years into episodes of 24 hours. 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 favours consumption patterns that are smoothly averaged without demand peaks, aligned with multiple metrics used in the evaluation, such as the 1-load factor and peak electricity demand. Another reduction is from multi-agent RL to single-agent RL, where each building's policy is updated independently, reducing the problem to decentralized control with additive rewards; such a reduction is computationally efficient for large-scale problems (De Nijs et al. 2021). We omit the notational dependence on candidate $j$ and iteration $k$ when presenting the method.

*Optimization planner.* We instantiate the optimization in (2) as follows. The planned states $\bar{s}_t$ consist of state variables such as net electricity consumption and SOCs of storage devices; the action $\bar{a}_t \in \mathcal{A}$ is the action of the MDP; the surrogate reward

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

is a combination of the negated ramping cost and the "virtual" electricity cost, where $\zeta = \{\theta_t \in [0, 5]\}_{t \in [24]}$ can be viewed as virtual electricity prices to be learned to encourage desirable consumption patterns (e.g., load flattening

[2] Note that the RBC controller is ubiquitous in traditional building control systems and is a simple form of "take action $a_h$ in hour $h$," where $a_h$ is a constant independent of current states except for the hour of the day ($h \in [24]$).

[3] We name our method ZO-iRL because the policy is implicitly determined by solving an optimization problem and the learning algorithm is zeroth-order in an RL setting.

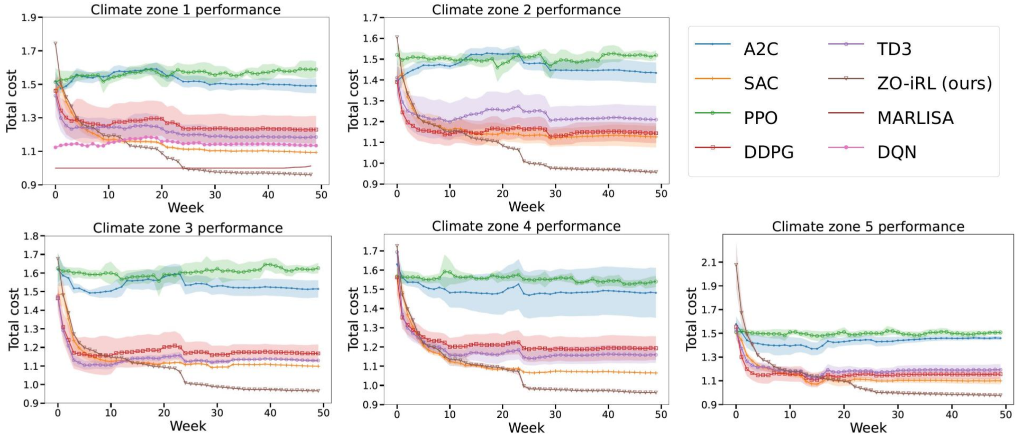


Figure 2: Learning curves of ZO-iRL and baselines. We perform 10 runs on each baseline to obtain performance plots with standard deviations for Climate Zones 1 to 5.

| | SAC | A2C | DDPG | DQN | PPO | TD3 | MARLISA | ZO-iRL |
|---|---|---|---|---|---|---|---|---|
| **ramping cost** | 1.145 (0.015) | 1.189 (0.002) | 1.174 (0.029) | 1.302 (0.008) | 1.638 (0.005) | 1.178 (0.026) | 1.022 (0.010) | **0.781** (0.005) |
| **1-load factor** | 1.158 (0.002) | 1.146 (0.002) | 1.143 (0.007) | 1.159 (0.003) | 1.168 (0.003) | 1.142 (0.004) | 1.026 (0.006) | **1.010** (0.008) |
| **avg. daily peak** | 1.180 (0.002) | 1.184 (0.008) | 1.195 (0.011) | 1.212 (0.002) | 1.242 (0.001) | 1.193 (0.006) | 1.015 (0.001) | **0.996** (0.001) |
| **peak demand** | 1.077 (0.008) | 1.088 (0.007) | 1.100 (0.010) | 1.115 (0.009) | 1.132 (0.006) | 1.098 (0.013) | 1.000 (6e-5) | **0.962** (0.005) |
| **net electric. peak** | 0.995 (0.001) | **0.994** (0.002) | 0.997 (8e-4) | 0.997 (1e-4) | 1.003 (7e-5) | 0.997 (7e-4) | 1.000 (5e-4) | 1.006 (2e-4) |
| **carbon emissions** | **1.000** (0.001) | 1.000 (0.002) | 1.003 (7e-4) | 1.005 (1e-4) | 1.009 (7e-5) | 1.004 (7e-4) | 1.001 (5e-4) | 1.007 (2e-4) |
| **total score** | 1.092 (0.003) | 1.101 (0.003) | 1.102 (0.008) | 1.132 (0.002) | 1.199 (0.001) | 1.102 (0.006) | 1.011 (0.002) | **0.962** (0.001) |

Table 2: Comparison with baselines: SAC (Kathirgamanathan et al. 2020) and MARLISA (Vazquez-Canteli, Henze, and Nagy 2020) have been officially implemented for CityLearn; other baselines are implemented by (Raffin et al. 2019). The reported values are the average and standard deviation (in brackets) across 10 independent runs on Climate Zone 1 data.

and smoothing). Intuitively, a higher value of $\theta_t$ discourages planned electricity consumption in the corresponding hour $t$.

The inequalities are grouped into technological constraints (e.g., maximum/minimum cooling power) and constraints on states and actions. The equalities are grouped into physics accounting for energy balances (i.e., consumption equal to supply) and technology (e.g., SOC update rules). Further details are provided in the appendix. Note that to set up the optimization (2), we also need to provide predictions of energy demands and solar generation. For simplicity, our predictors are based on a simple averaging scheme that takes the average in the corresponding hours of the last 2 weeks of data; thus, there are no specific parameters to learn.

*Transition and guidance.* We use (7) as the transition probability, with variance $\iota_k = 0.4/k^2$ that is initialized to 0.4 and decreases by $k^2$ in each iteration. The guidance signal $\varrho$ is computed as follows. By the end of each episode, we examine the net electricity usage in the past 24 hours, $e_t$ for $t \in [24]$ and find the top 2 hours with the most electricity usage, denoted by $t_1$ and $t_2$. Then, the guidance signal $\varrho_t$ is $0.02$ if $t \in \{t_1, t_2\}$ and $-0.04/22$ otherwise. Note that we have centered the signal ($\sum_t \varrho_t = 0$) by assigning negative values for hours other than peaks. We choose $\alpha_k = 1$ for all $k$ over the entire 4-year period, as there is no training phase in the CityLearn Challenge and we prefer to adapt quickly during the test phase; this is not a violation of our theory, as we can choose to diminish $\alpha_k$ after a while to still satisfy the condition $\sum_{k=1}^{\infty} \alpha_k < \infty$.

**Results.** For baselines, we use the implementation of (Raffin et al. 2019) with the default ADAM optimizer, where the policy is an NN architecture with tanh activation and two layers of 256 units each. From Table 2, we see that ZO-iRL has achieved the lowest cost ratios (i.e., best scores) of all, which is consistent with the official result of the competition (Table 1). In particular, as shown in Fig. 3, ZO-iRL is able to find a good policy in the first few months, while baselines seem to struggle; we speculate that more samples would eventually improve the performance of baselines, and all methods may benefit from schemes to handle the potentially nonstationary environment due to seasonal patterns.

## 6 Conclusion and future directions

We presented a novel adaptive optimization framework that has been shown to be very effective for energy storage management. Using solution functions as policies offers a promising way to introduce data-driven algorithms into the real world where convex optimization has been widely adopted. To adapt the optimization parameters, we developed an evolutionary search algorithm that can incorporate insights from control trajectory data as guidance for parameter updates. The method outperforms several baselines and ranked first in the latest 2021 CityLearn Challenge. Some potential future directions could be to extend the proposed framework to other methods such as Bayesian optimization or first-order methods such as actor-critic.

## 7 Acknowledgments

We would like to thank all the ZORL/ROLEVT team members at the CityLearn Challenge 2021: Qasim Wani, Zhiyao Zhang, and Mingyu Kim. We would also like to thank the organizers of the CityLearn Challenge: Dr. José R. Vázquez-Canteli, Dr. Zoltan Nagy, Dr. Gregor Henze, and Sourav Dey for organizing this competition and inspiring novel RL-based approaches to the contemporary challenges facing the power and energy sector. This project is funded by NSF, the Commonwealth Cyber Initiative (CCI), C3.ai Digital Transformation Institute, and the U.S. Department of Energy.

## References

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, A.; Amos, B.; Barratt, S.; Boyd, S.; Diamond, S.; and Kolter, J. Z. 2019. Differentiable convex optimization layers. *Advances in neural information processing systems*, 32.

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

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

Bertsekas, D. 2019. *Reinforcement learning and optimal control*. Athena Scientific.

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

Blum, J.; Chernoff, H.; Rosenblatt, M.; and Teicher, H. 1958. Central limit theorems for interchangeable processes. *Canadian Journal of Mathematics*, 10: 222–229.

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

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

Chen, X.; Qu, G.; Tang, Y.; Low, S.; and Li, N. 2022. Reinforcement learning for selective key applications in power systems: Recent advances and future challenges. *IEEE Transactions on Smart Grid*.

Conti, E.; Madhavan, V.; Petroski Such, F.; Lehman, J.; Stanley, K.; and Clune, J. 2018. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. *Advances in neural information processing systems*, 31.

De Nijs, F.; Walraven, E.; De Weerdt, M.; and Spaan, M. 2021. Constrained multiagent Markov decision processes: A taxonomy of problems and algorithms. *Journal of Artificial Intelligence Research*, 70: 955–1001.

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

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

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

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.

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

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

Gangwani, T.; and Peng, J. 2018. Policy Optimization by Genetic Distillation. In *International Conference on Learning Representations*.

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

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*.

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.

Hajek, B. 2015. *Random processes for engineers*. Cambridge university press.

Haydari, A.; and Yilmaz, Y. 2020. Deep reinforcement learning for intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*.

Jin, M.; Khattar, V.; Kaushik, H.; Sel, B.; and Jia, R. 2023. On Solution Functions of Optimization: Universal Approximation and Covering Number Bounds. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Karkus, P.; Hsu, D.; and Lee, W. S. 2017. Qmdp-net: Deep learning for planning under partial observability. *Advances in neural information processing systems*, 30.

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.

Khadka, S.; and Tumer, K. 2018. Evolution-guided policy gradient in reinforcement learning. *Advances in Neural Information Processing Systems*, 31.

Liu, S.; Chen, P.-Y.; Kailkhura, B.; Zhang, G.; Hero III, A. O.; and Varshney, P. K. 2020. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5): 43–54.

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.

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

Nagy, Z.; Vázquez-Canteli, J. R.; Dey, S.; and Henze, G. 2021. The CityLearn Challenge 2021. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, 218–219.

Nian, R.; Liu, J.; and Huang, B. 2020. A review on reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering*, 139: 106886.

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

Powell, W. 2020. Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions. *Princeton NJ*.

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

Prékopa, A. 2013. *Stochastic programming*, volume 324. Springer Science & Business Media.

Racanière, S.; Weber, T.; Reichert, D.; Buesing, L.; Guez, A.; Jimenez Rezende, D.; Puigdomènech Badia, A.; Vinyals, O.; Heess, N.; Li, Y.; et al. 2017. Imagination-augmented agents for deep reinforcement learning. *Advances in neural information processing systems*, 30.

Raffin, A.; Hill, A.; Ernestus, M.; Gleave, A.; Kanervisto, A.; and Dormann, N. 2019. Stable baselines3.

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

Rolnick, D.; Donti, P. L.; Kaack, L. H.; Kochanski, K.; Lacoste, A.; Sankaran, K.; Ross, A. S.; Milojevic-Dupont, N.; Jaques, N.; Waldman-Brown, A.; et al. 2022. Tackling climate change with machine learning. *ACM Computing Surveys (CSUR)*, 55(2): 1–96.

Salimans, T.; Ho, J.; Chen, X.; Sidor, S.; and Sutskever, I. 2017. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*.

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.

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

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

Szita, I.; and Lörincz, A. 2006. Learning Tetris using the noisy cross-entropy method. *Neural computation*, 18(12): 2936–2941.

Tamar, A.; Wu, Y.; Thomas, G.; Levine, S.; and Abbeel, P. 2016. Value iteration networks. *Advances in neural information processing systems*, 29.

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

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, 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.

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

Zhou, Z.-H.; Yu, Y.; and Qian, C. 2019. *Evolutionary learning: Advances in theories and algorithms*. Springer.

# A Proof of results in the main paper

## A.1 Formalism of the guidance signal

The formalism of the guidance signal requires some basics from random process and measure theory (interested readers are referred to (Hajek 2015)). We keep our presentation minimal but sufficient enough to carry out the analysis. Consider a stochastic process $(S_t^{j,k}, A_t^{j,k})_{t\in[T]}$ defined by policy $\pi_{\zeta_j^k}$ interacting with the MDP environment, where $S_t^{j,k}$ and $A_t^{j,k}$ are random variables representing the state and action at time $t$. Let $\mathcal{I}_t = (\mathcal{S}\times\mathcal{A})^t$ be the set of possible histories up to time step $t$ within an episode, and $I_t^{j,k} := (S_1^{j,k}, A_1^{j,k}, ..., S_t^{j,k}, A_t^{j,k}) \in \mathcal{I}_t$ is a random vector taking values in $\mathcal{I}_t$ containing all state-action pairs observed up to step $t$. Denote by $\mathcal{F}_t^{j,k}$ a non-decreasing sequence of $\sigma$-algebra (a *filtration*) generated by $I_t^{j,k}$. Then, the guidance signal $\varrho_j^k \in \mathcal{Z}'$ is a random variable adapted to the filtration $\mathcal{F}_T^{j,k}$, i.e., $\varrho_j^k$ is $\mathcal{F}_T^{j,k}$-measurable, with probability measure $M_k(\zeta_j^k, d\varrho)$ associated with a properly defined probability space, the existence of which is ensured by the Ionescu-Tulcea theorem. Note that the distribution of $\varrho_j^k$ depends on $\zeta_j^k$, since the stochastic process $I_t^{j,k}$ is determined by the policy $\pi_{\zeta_j^k}$, but is conditionally independent of all other candidates $\zeta_{j'}^k$ for $j' \neq j$.

## A.2 Proof of Lemma 1

Let $\Phi(s_t, \zeta)$ represent the feasible set of (2). By Assumption 1, $\Phi(s_t, \zeta)$ is convex for fixed $s_t$ and $\zeta$ and has a nonempty interior. This implies that $\Phi(s_t, \zeta)$ is continuous in $s_t$ and $\zeta$ (Rockafellar and Wets 2009, example 5.10). Hence, by Berge maximum theorem (Berge 1997), $\pi_\zeta(s_t)$ is upper hemicontinuous in $\zeta$ for fixed $s_t \in \mathcal{S}$. However, we know that $\pi_\zeta(s_t)$ contains a single point due to the 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 its parameter $\zeta$.

## A.3 Proof of Theorem 1

Select 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

$$\tilde{P}_{k+1}(d\zeta) = \left(\int \tilde{P}_k(dz)\exp(f(z))\right)^{-1} \int \tilde{P}_k(dz)\exp(f(z))\Big(Q_k(z,\varrho,d\zeta)M_k(z,d\varrho) + \Delta_{N_k}(d\zeta)\Big). \tag{10}$$

By Assumption 2 $(d)$ and $(h)$, 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

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

which, by Lemma 3, converges to $\lambda$. Thus, by the standard diagonalization argument (Billingsley 2013), we can show that there exists a subsequence $\{\tilde{P}_{k_j}\}$ that weakly converges to $\lambda$. 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.

## A.4 Proof of Corollary 1

Under Assumption 2 (except $(h)$), the distributions (16) have continuous densities with respect to the Lebesgue measure. Let $A(\epsilon) = \{\zeta \in \mathcal{Z} : f(\zeta) \geq f^* - \epsilon\}$. By (9) 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 $k \geq k_n$ and $\epsilon > 0$, the following holds

$$\tilde{P}_{k+1}(A(\epsilon+\epsilon_k)) \geq (1-\delta_k)\tilde{P}_k(A(\epsilon)), \tag{11}$$

where

$$0 < \delta_k < 1 \quad \text{for } k \in \mathbb{N}, \qquad \sum_{k\in\mathbb{N}} \delta_k < \infty, \tag{12}$$

and $\epsilon_k \geq 0$ are determined in terms of $\iota_k$, $\alpha_k$, and the sizes of the support of density $\psi$,

$$\sum_{k=1}^{\infty} \epsilon_k \leq \text{constant} \sum_{k=1}^{\infty} \iota_k < \infty. \tag{13}$$

Such sequence of $\{N_k\}$ and $k_n$ exist by Lemma 2, the finiteness of $\psi$, and the condition that $\sum_{k=1}^{\infty} \alpha_k < \infty$. Next, select $k_o \geq k_n$ 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 (12). The proof is complete.

**Remarks on the guidance signal.** From the proof of Corollary 1, it can be observed that we can relax the condition that $\sum_{k=1}^{\infty} \alpha_k < \infty$ as long as the guidance signal $\alpha_k \varrho^k$ is chosen in such a way that (11), (12), and (13) are satisfied. This means that we can continue applying the guidance signal without the need to diminish its impact in the long run. However, (11) is difficult to ensure, as it requires designing a guidance that always points to the global optimal. Therefore, in practice, it is recommended to diminish the effect of guidance and eventually let the data drive the decision.

## A.5 An example of transition probability with noiseless function evaluations

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

$$\begin{aligned}Q_k(z, \varrho, A) = &\int 1_{\{\zeta \in A, f(z) \leq f(\zeta)\}} T_k(z, \varrho, d\zeta) \\ &+ 1_{\{z \in A\}} \int 1_{\{f(\zeta) < f(z)\}} T_k(z, \varrho, d\zeta),\end{aligned} \tag{14}$$

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

*Proof.* By Assumption 2 $(c)$ and $(g)$, we have that $\tilde{P}_1(\mathbb{B}^*(\epsilon)) > 0$ for any $\epsilon > 0$. By (14), 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 $(h)$ is satisfied. By Theorem 1, the claim is proved. □

**Remarks.** To implement the transition of (14), one first needs to sample a variable $\zeta$ according to $T_k(z, \varrho, d\zeta)$ and observe its reward value $f(\zeta)$; then, the output is $\zeta$ if $f(\zeta) \geq f(z)$ and $z$ otherwise. Such a scheme depends crucially on a reliable way of comparing candidates (e.g., noiseless evaluation).

## A.6 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

$$\lim_{k \to \infty} c_k \int \exp(kf(\zeta))\Psi(\zeta)\mu(d\zeta) = \int \Psi(\zeta)\lambda(d\zeta), \tag{15}$$

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 $\epsilon_0, \epsilon_1$, and $\epsilon_2$ to be determined. For any $\delta > 0$, by continuity of $\Psi$, 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 $\epsilon_1$ (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 $\epsilon_2$ 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*

$$\tilde{P}_{k+1}(d\zeta) = \left(\int \tilde{P}_k(dz) \exp(f(z))\right)^{-1} \int \tilde{P}_k(dz) \exp(f(z)) Q_k(z, \varrho, d\zeta) M_k(z, d\varrho) + \Delta_{N_k}(d\zeta), \qquad (16)$$

*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:

$$\begin{aligned}
\tilde{P}_{k+1}(d\zeta) &= \int_{\Omega^N} \chi_k(d\omega_N) \left\{ \beta(\omega_N) \sum_{i=1}^{N} \Lambda(\zeta_i, \varrho_i, \xi_i, d\zeta) \right\} \\
&= \sum_{i=1}^{N} \int_{\Omega^N} \chi_k(d\omega_N) \beta(\omega_N) \Lambda(\zeta_i, \varrho_i, \xi_i, d\zeta) \\
&= \int_{\Omega^N} \chi_k(d\omega_N) \left\{ N\beta(\omega_N) \right\} \Lambda(\zeta_1, \varrho_1, \xi_1, d\zeta).
\end{aligned}$$

which can be represented in the form of (16) with

$$\begin{aligned}
\Delta_N(d\zeta) &= \int_{\Omega^N} \chi_k(d\omega_N) \Lambda(\zeta_1, \varrho_1, \xi_1, d\zeta) \left\{ N\beta(\omega_N) - \left(\int \tilde{P}_k(dz) \exp(f(z))\right)^{-1} \right\} \\
&\quad + \left(\int \tilde{P}_k(dz) \exp(f(z))\right)^{-1} \left\{ \int_{\Omega^N} \chi_k(d\omega_N) \Lambda(\zeta_1, \varrho_1, \xi_1, d\zeta) - \int_{\Omega} \tilde{P}_k(dz) \exp(f(z)) Q_k(z, \varrho, d\zeta) M_k(z, d\varrho) \right\} \\
&= (i) + (ii)
\end{aligned}$$

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

$$v_N(z) = \int_{\Omega^N} \chi_k(d\omega_N) \exp(f(\zeta_1) + \xi_1) q_k(\zeta_1, \varrho_1, z) \left\{ N\beta(\omega_N) - \left(\int \tilde{P}_k(dz) \exp(f(z))\right)^{-1} \right\}.$$

To proceed, let $\gamma_N = \frac{1}{N} \sum_{i=1}^{N} \exp(f(\zeta_i) + \xi_i)$ and $\psi(z) = \exp(f(\zeta_1) + \xi_1) q_k(\zeta_1, \varrho_1, z)$. Due to the symmetrical dependence of random elements $\zeta_1, ..., \zeta_N$ and $\varrho_1, ..., \varrho_N$, as well as the independence of $\xi_1, ..., \xi_N$, the random variables $\gamma_N$ converge in mean for $N \to \infty$ to some random variable $\gamma$ 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,

$$
\begin{aligned}
|v_N(z)| &= \left| \mathbb{E}\left(\frac{\psi(z)}{\gamma_N}\right) - \frac{\mathbb{E}\psi(z)}{\mathbb{E}\gamma} \right| && (17)\\
&= \frac{1}{\mathbb{E}\gamma} \left| \mathbb{E}\left(\frac{\psi(z)\gamma}{\gamma_N}\right) - \mathbb{E}\psi(z) \right| && (18)\\
&\leq \exp(c_f) \left| \mathbb{E}\left(\frac{\psi(z)|\gamma - \gamma_N|}{\gamma_N}\right) \right| && (19)\\
&\leq \exp(2c_f)\|\psi\|_\infty \mathbb{E}|\gamma - \gamma_N| && (20)\\
&\leq L_k \exp(3c_f + c_\xi)\mathbb{E}|\gamma - \gamma_N|, && (21)
\end{aligned}
$$

where the second equality is due to the independence of $\gamma$ from $\gamma_N$ and $\psi$, 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 (21), 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 (Blum et al. 1958)), it follows that $\mathbb{E}|\gamma - \gamma_N| = \mathcal{O}(N^{-1/2})$. Consequently, we have shown that $\int |v_N(\zeta)|\mu(d\zeta) = \mathcal{O}(N^{-1/2})$.

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

$$
\begin{aligned}
&\int_{\Omega^N} \chi_k(d\omega_N)\Lambda(\zeta_1, \varrho_1, \xi_1, d\zeta) - \int_{\Omega} \tilde{P}_k(dz)\exp(f(z))Q_k(z, \varrho, d\zeta)M_k(z, d\varrho)\\
&= \int_{\mathcal{Z}} \tilde{P}_k(dz)\exp(f(z))Q_k(z, \varrho, d\zeta)M_k(z, d\varrho)\left\{\int \exp(\xi) F_k(d\xi) - 1\right\},
\end{aligned}
$$

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

# B Additional details for the CityLearn Challenge

## B.1 Details of optimization model

We refer the reader to (Vazquez-Canteli et al. 2020) and the corresponding online documentation[4] for the detailed setup of the contest. We will focus only on our strategy in this document. In particular, we provide details on the construction of the optimization model in 2. Denote the hourly index by $r \in \{1, 2, \cdots, T\}$, where $T = 24$. Suppose we are at the beginning of the hour $r$. Then we need to plan for the actions for the future hours until the end of the day and execute the plan for the next hour $r$, a.k.a., rolling-horizon planning. Next, we describe hyperparameters, variables, objective, and constraints in 2.

**Hyperparameters.** 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 $\eta_{\text{ehH}}$, nominal power $E_{\max}^{\text{ehH}}$; *(2)* heat pump: technical efficiency $\eta_{\text{tech}}^{\text{hp}}$, target cooling temperature $t_c^{\text{hp}}$, nominal power $E_{\max}^{\text{hpc}}$; *(3)* electric battery: rate of decay $Cf^{\text{bat}}$, capacity $Cp^{\text{bat}}$, efficiency $\eta_t^{\text{bat}}$; *(4)* heat storage: rate of decay $Cf^{\text{Hsto}}$, capacity $Cp^{\text{Hsto}}$, efficiency $\eta_t^{\text{Hsto}}$; *(5)* cooling storage: rate of decay $Cf^{\text{Csto}}$, capacity $Cp^{\text{Csto}}$, efficiency $\eta_t^{\text{Csto}}$.
- The hyperparmeters provided by the 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 $\text{temp}_t$ 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 above predictions are required for hour $r \leq t \leq T$. In our algorithm, predictions are provided by simply averaging the last 2 weeks of data in the corresponding hour.
- The hyperparameters to be learned by Algorithm 1 are the virtual electricity price $\{\theta_t\}_{t=1,\ldots,24}$ 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] link: https://sites.google.com/view/citylearnchallenge

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 $a_r^{\text{bat}}$, $a_r^{\text{Hsto}}$, and $a_r^{\text{Csto}}$. The remaining variables are considered auxiliary variables for planning.

**Objective function.** The objective function is given by:

$$|E_t^{\text{grid}} - E_{t-1}^{\text{grid}}| + \theta_t E_t^{\text{grid}} + \sum_{t'=t+1}^{T} \Big( |E_{t'}^{\text{grid}} - E_{t'-1}^{\text{grid}}| + \theta_{t'} E_{t'}^{\text{grid}} \Big). \tag{22}$$

Note that we use $e_t$ for $E_t^{\text{grid}}$ in the main text. Also, the above objective is used in a standard minimization problem; to make it consistent with the maximization problem in (2), 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 $t \geq r$:
  $0 \leq SOC_t^{\mathrm{H}} \leq 1$

*Cooling storage technology constraints:*

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

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).

# C Additional experimental results

## C.1 Official results for the 2021 CityLearn Challenge

| | **ZO-iRL** | **ICD-CA** | **IDLab** | **Breakfast Club** |
|---|---|---|---|---|
| **total score** | **0.944** | 1.070 | 1.070 | 1.130 |
| **total last year** | **0.942** | 1.052 | 1.077 | 1.067 |
| **coord. score** | **0.915** | 1.107 | 1.094 | 1.195 |
| **coord. score last year** | **0.918** | 1.074 | 1.098 | 1.095 |
| **carbon emissions** | 1.003 | **1.000** | 1.028 | 1.003 |

Table 3: Official results for the 2021 CityLearn Challenge (Nagy et al. 2021). Here, the total score is the average of all 6 cost metrics considered in the competition. The coordination score is the average of the first 4 metrics (see the main paper for these metrics). Last year scores are calculated based on the performance of the last year within the total 4-year simulation period.

## C.2 Hyperparameters of ZO-iRL and baselines

| **Parameter** | **Value** |
|---|---|
| # of parameter candidates $N_k$ | 3 |
| Initial variance $\iota_1$ | 0.4 |
| Guidance signal $\rho$ | specified in the main text |
| Guidance rate $\alpha_k$ | 1 |
| Duration of one episode (hours) | 24 |
| Range of virtual electricity price | $[0, 5]$ |
| State-action trajectory buffer size (days) | 7 |

Table 4: Hyperparameters for ZO-iRL.

| **Parameter** | **Value** | | | | | |
|---|---|---|---|---|---|---|
| | DDPG | DQN | PPO | TD3 | A2C | SAC |
| Learning rate | 1e-3 | 1e-4 | 3e-4 | 1e-3 | 7e-4 | 3e-4 |
| # of epochs | NA | NA | 10 | NA | 5 | NA |
| Buffer size | 1e6 | 1e6 | NA | 1e6 | NA | 1e6 |
| Batch size | 100 | 32 | 64 | 100 | NA | 256 |
| Discount factor | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 |

Table 5: Parameter values used for RL baselines. NA means not applicable. ADAM optimizer is used for each baseline, where the policy is given by the NN architecture with a $\tanh$ activation function and two layers of 256 units each.

### C.3 Results for all climate zones

Here, we provide results for all climate zones. Note that ZO-iRL performs some random parameter exploration in the first few weeks, which results in worse performance. However, over time, performance improves due to the guided ES, as shown in Fig. 3.

| Method | SAC | A2C | DDPG | PPO | TD3 | ZO-iRL |
|---|---|---|---|---|---|---|
| **ramping cost** | 1.244 (0.196) | 2.714 (0.186) | 1.327 (0.181) | 2.718 (0.079) | 1.500 (0.228) | 0.750 (0.010) |
| **1-load factor** | 1.162 (0.026) | 1.265 (0.012) | 1.192 (0.034) | 1.273 (0.022) | 1.279 (0.120) | 1.028 (0.001) |
| **avg. daily peak** | 1.195 (0.047) | 1.413 (0.024) | 1.218 (0.033) | 1.371 (0.018) | 1.351 (0.121) | 0.994 (0.003) |
| **peak demand** | 1.081 (0.001) | 1.149 (0.045) | 1.119 (0.041) | 1.419 (0.082) | 1.195 (0.161) | 0.950 (0.020) |
| **net electric. peak** | 0.992 (0.001) | 1.011 (0.002) | 0.993 (0.004) | 1.013 (0.001) | 1.099 (0.2066) | 1.008 (2e-4) |
| **carbon emissions** | 0.998 (0.001) | 1.017 (0.002) | 0.999 (0.004) | 1.023 (0.002) | 1.104 (0.203) | 1.010 (1e-4) |
| **total score** | 1.112 (0.044) | 1.428 (0.042) | 1.141 (0.047) | 1.469(0.026) | 1.255 (0.141) | 0.957 (0.004) |

Table 6: Comparison of ZO-iRL and baselines for **Climate Zone 2**. The reported values are the average and standard deviation (in brackets) across 10 independent runs.

| Method | SAC | A2C | DDPG | PPO | TD3 | ZO-iRL |
|---|---|---|---|---|---|---|
| **ramping cost** | 1.098 (0.011) | 2.986 (0.223) | 1.367 (0.182) | 3.154 (0.066) | 1.263 (0.127) | 0.775 (0.004) |
| **1-load factor** | 1.140 (0.002) | 1.272 (0.009) | 1.188 (0.040) | 1.375 (0.038) | 1.240 (0.136) | 1.043 (0.001) |
| **avg. daily peak** | 1.169 (0.002) | 1.441 (0.024) | 1.242 (0.063) | 1.439 (0.025) | 1.277 (0.177) | 1.010 (0.003) |
| **peak demand** | 1.182 (0.001) | 1.272 (0.043) | 1.199 (0.022) | 1.472 (0.084) | 1.259 (0.127) | 0.952 (0.014) |
| **net electric. peak** | 0.994 (0.002) | 1.013 (0.003) | 0.996 (0.004) | 1.010 (0.001) | 1.099 (0.206) | 1.006 (3e-4) |
| **carbon emissions** | 1.000 (0.002) | 1.019 (0.003) | 1.002 (0.004) | 1.020 (0.001) | 1.104 (0.203) | 1.007 (1e-4) |
| **total score** | 1.097 (0.002) | 1.501 (0.047) | 1.166 (0.048) | 1.578(0.025) | 1.207 (0.152) | 0.966 (0.003) |

Table 7: Comparison of ZO-iRL and baselines for **Climate Zone 3**. The reported values are the average and standard deviation (in brackets) across 10 independent runs.

| Method | SAC | A2C | DDPG | PPO | TD3 | ZO-iRL |
|---|---|---|---|---|---|---|
| **ramping cost** | 1.024 (0.006) | 2.814 (0.468) | 1.566 (0.230) | 3.070 (0.108) | 1.310 (0.072) | 0.739 (0.003) |
| **1-load factor** | 1.117 (0.003) | 1.229 (0.024) | 1.185 (0.036) | 1.449 (0.019) | 1.187 (0.038) | 1.013 (0.006) |
| **avg. daily peak** | 1.126 (0.002) | 1.411 (0.072) | 1.249 (0.070) | 1.429 (0.015) | 1.262 (0.051) | 1.003 (0.001) |
| **peak demand** | 1.134 (1e-4) | 1.238 (0.069) | 1.155 (0.036) | 1.444 (0.061) | 1.205 (0.099) | 0.999 (0.026) |
| **net electric. peak** | 0.987 (0.002) | 1.010 (0.006) | 0.995 (0.004) | 1.007 (0.002) | 0.990 (0.003) | 1.007 (4e-4) |
| **carbon emissions** | 0.994 (0.002) | 1.017 (0.006) | 1.000 (0.004) | 1.015 (0.001) | 0.996 (0.004) | 1.009 (8e-4) |
| **total score** | 1.064 (0.001) | 1.453 (0.103) | 1.192 (0.060) | 1.569(0.032) | 1.158 (0.042) | 0.962 (0.003) |

Table 8: Comparison of ZO-iRL and baselines for **Climate Zone 4**. The reported values are the average and standard deviation (in brackets) across 10 independent runs.

| Method | SAC | A2C | DDPG | PPO | TD3 | ZO-iRL |
|---|---|---|---|---|---|---|
| **ramping cost** | 1.245 (0.162) | 2.603 (0.158) | 1.320 (0.161) | 2.673 (0.047) | 1.385 (0.144) | 0.789 (0.005) |
| **1-load factor** | 1.241 (0.055) | 1.314 (0.011) | 1.212 (0.033) | 1.332 (0.032) | 1.253 (0.043) | 1.045 (0.011) |
| **avg. daily peak** | 1.172 (0.049) | 1.347 (0.025) | 1.197 (0.043) | 1.346 (0.021) | 1.221 (0.044) | 1.004 (0.002) |
| **peak demand** | 1.285 (0.097) | 1.356 (0.021) | 1.203 (0.035) | 1.535 (0.184) | 1.231 (0.066) | 1.014 (0.022) |
| **net electric. peak** | 0.989 (0.002) | 1.003 (0.002) | 0.990 (0.006) | 1.003 (8e-4) | 0.993 (0.003) | 1.004 (0.001) |
| **carbon emissions** | 0.995 (0.002) | 1.009 (0.002) | 0.997 (0.007) | 1.015 (7e-4) | 0.999 (0.002) | 1.005 (0.001) |
| **total score** | 1.154 (0.057) | 1.438 (0.032) | 1.153 (0.041) | 1.484(0.042) | 1.180 (0.049) | 0.977 (0.005) |

Table 9: Comparison of ZO-iRL and baselines for **Climate Zone 5**. The reported values are the average and standard deviation (in brackets) across 10 independent runs.

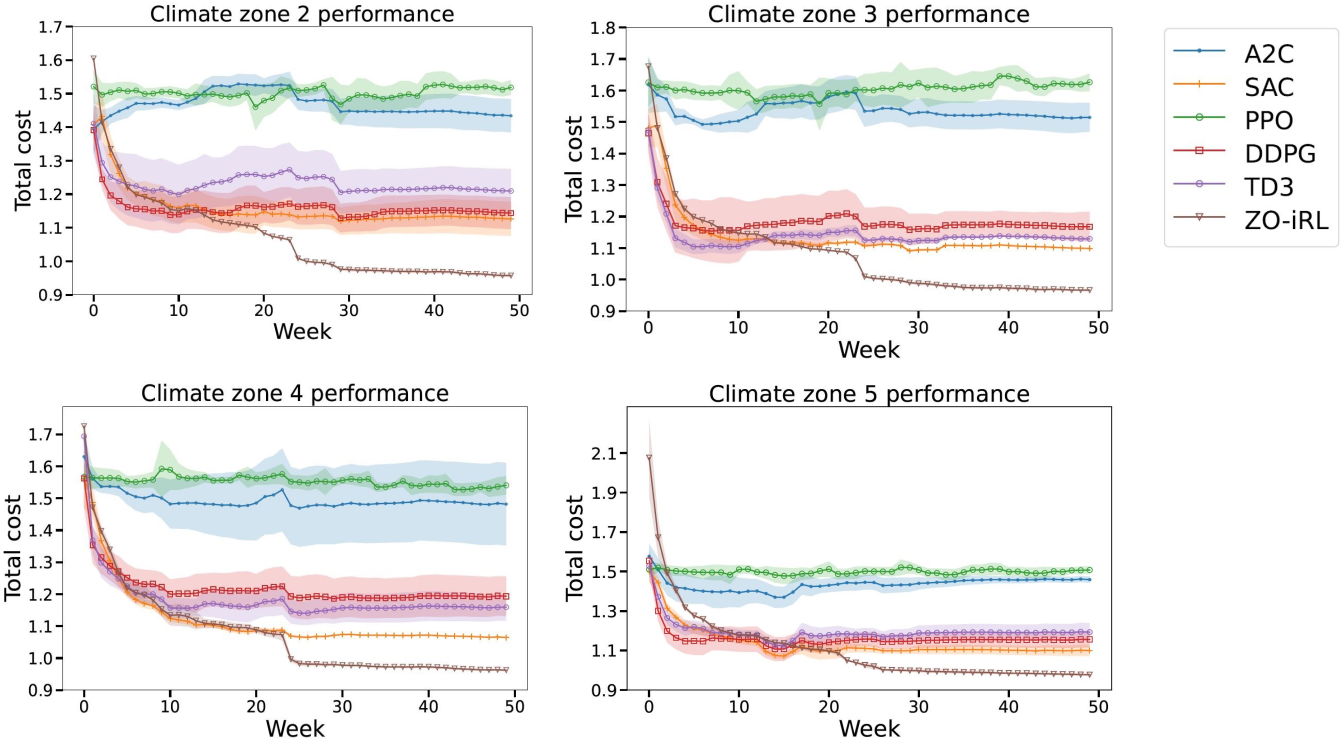
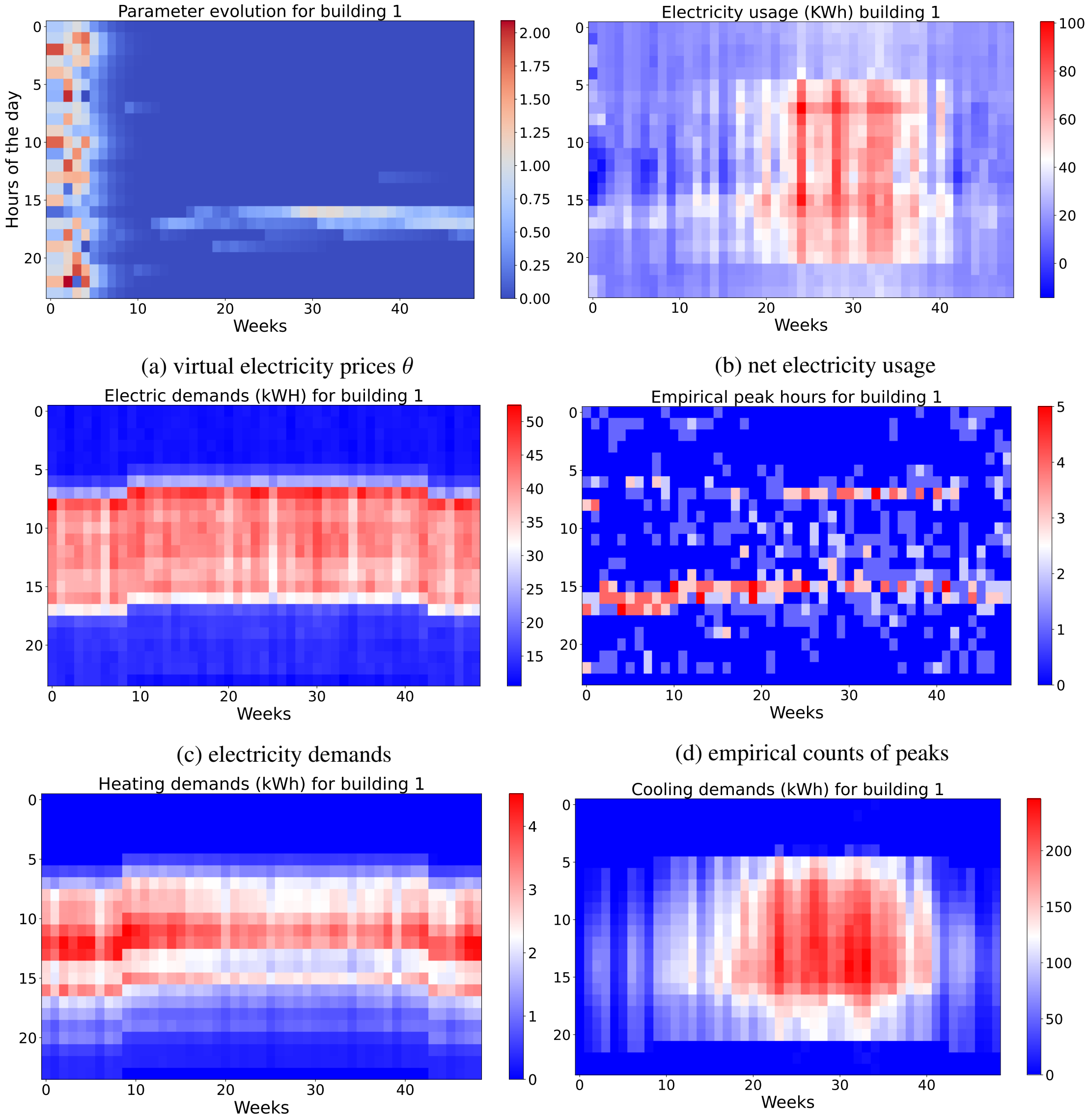


Figure 3: Learning curves of ZO-iRL and baselines for Climate Zones 2–5. Note that ZO-iRL is the only method that consistently achieves a cost below 1 across different runs in different climate zones.

## C.4 Visualization of parameter evolution

In this section, we visualize the evolution of parameters under ZO-iRL for some buildings. We also juxtapose the corresponding patterns of empirical peak counts, net electricity use, electricity demand, heating demand, and cooling demand. The empirical count is calculated for each hour as the number of times the corresponding hour has the top 2 net electricity usage in a week. The higher the empirical counts, the more frequent the corresponding hour has peak usage. We also note that electricity usage is higher than electricity demand due to the additional energy demand for heating and cooling.



(a) virtual electricity prices $\theta$

(b) net electricity usage

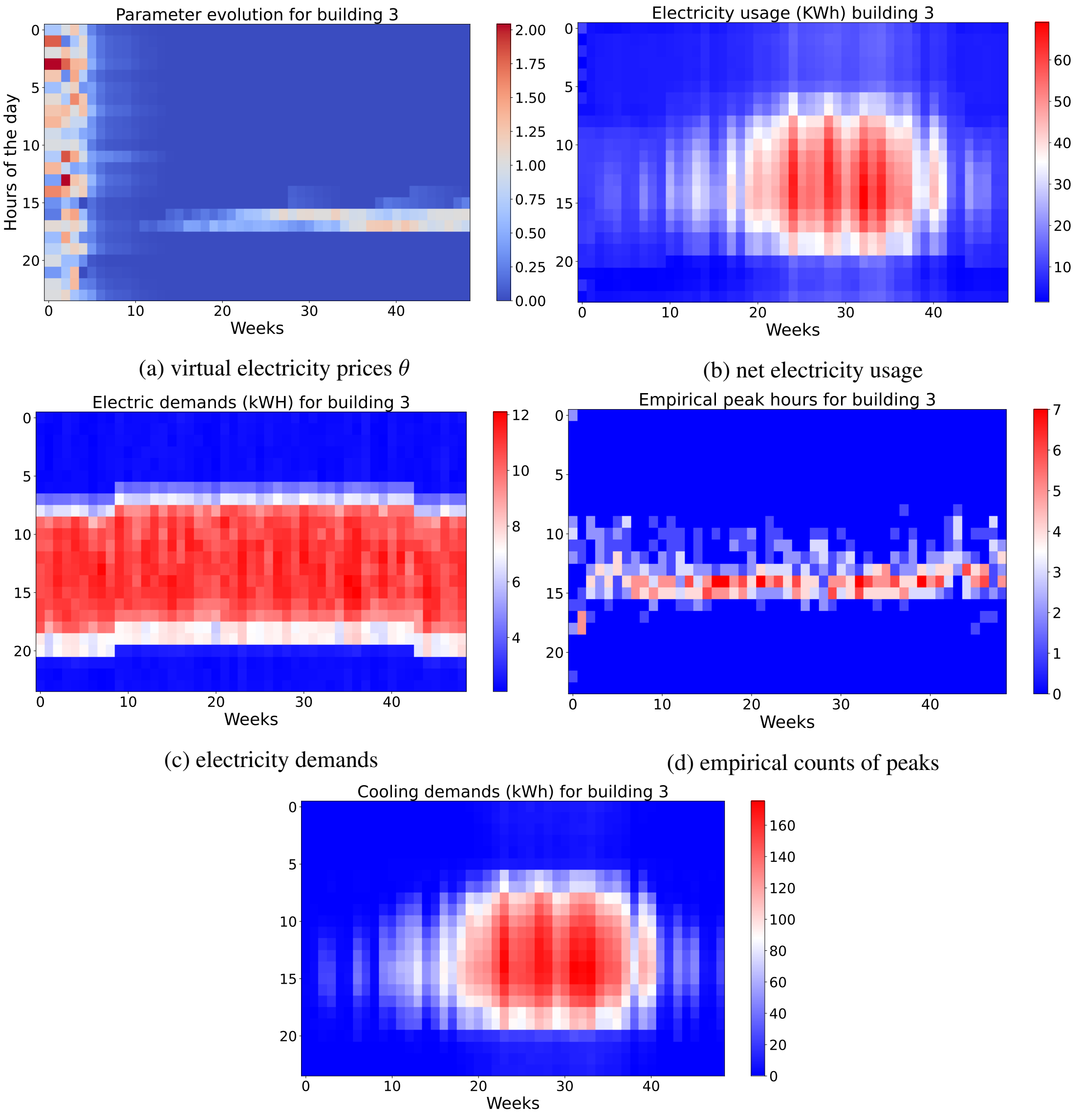
(c) electricity demands

(d) empirical counts of peaks

(e) heating demands

(f) cooling demands

Figure 4: Visualization of (a) parameter learning, (b) net electricity use (c) electric loads, (d) empirical counts of peaks, (e) heating demand, and (d) cooling demand for Building 1. It can be observed that Building 1 continues to increase the virtual electricity prices for hours around 16–18 in response to consistently observed peaks in those hours. Due to storage controls, the net electricity usage pattern is smoother (spreading throughout the day) than the demand patterns, as observed in all buildings.



(a) virtual electricity prices $\theta$

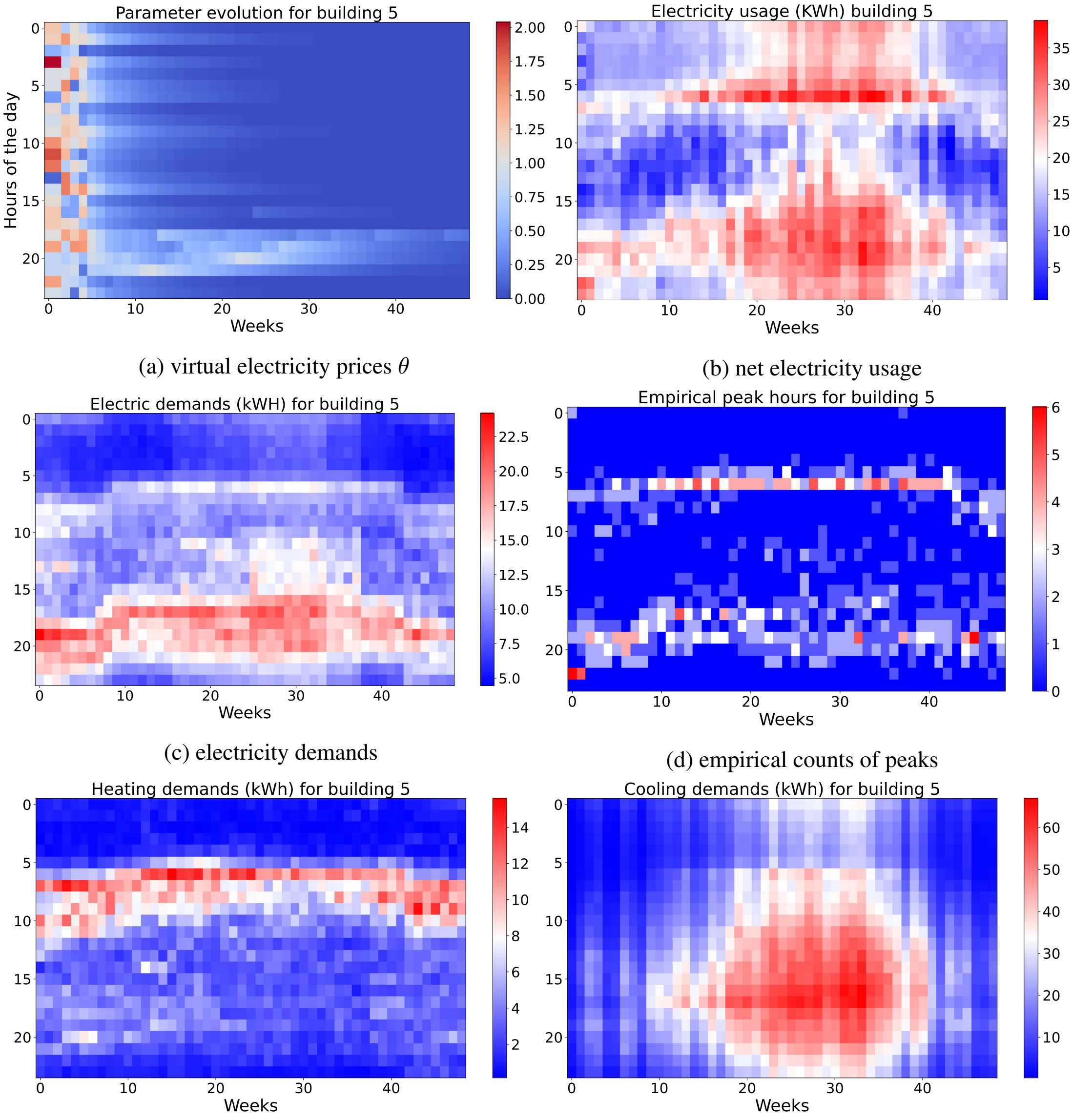
(b) net electricity usage

(c) electricity demands

(d) empirical counts of peaks

(e) cooling demands

Figure 5: Similar patterns can be observed for Building 3. Note that for this building, there is no heating demands.



(a) virtual electricity prices $\theta$

(b) net electricity usage

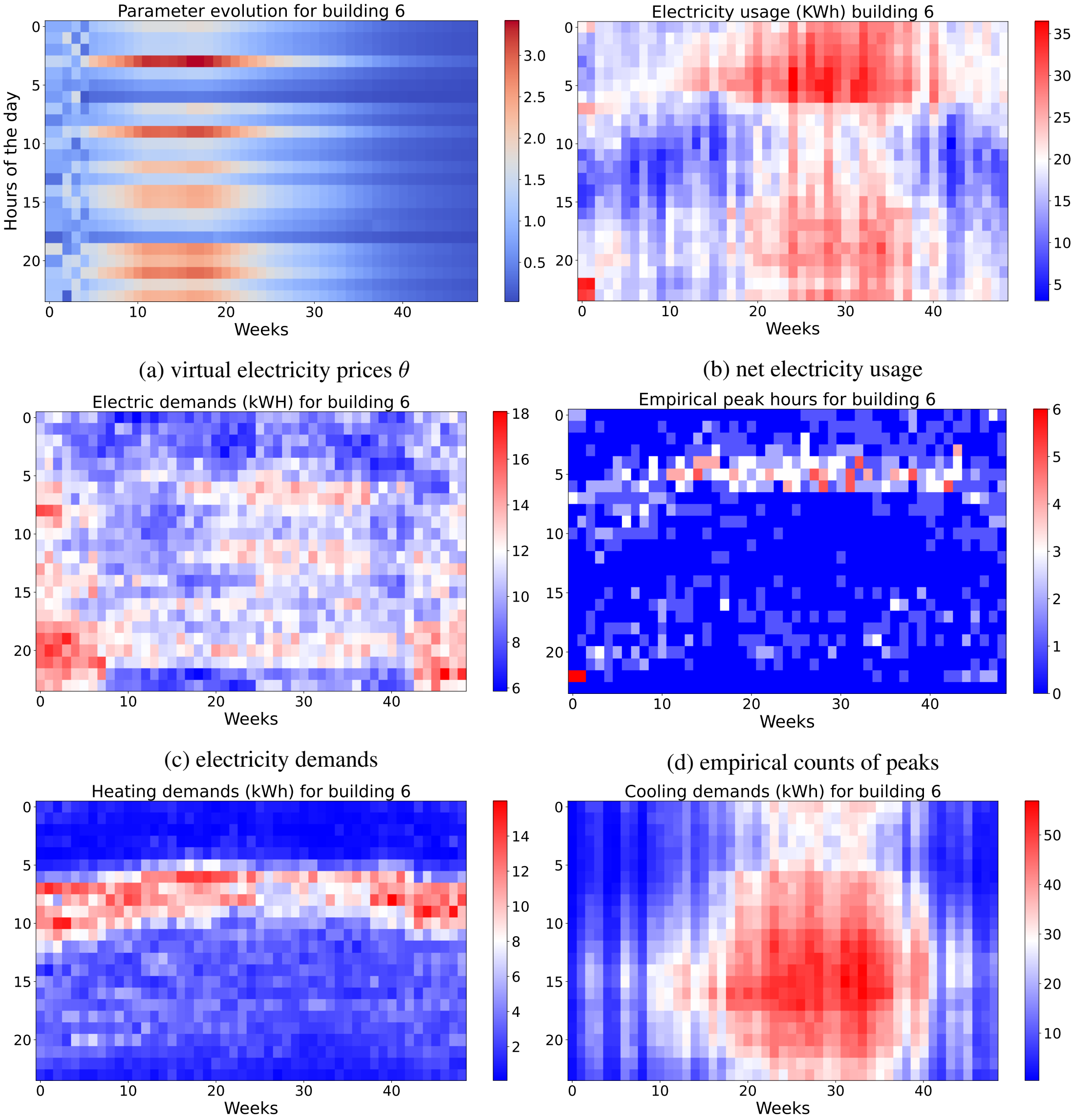
(c) electricity demands

(d) empirical counts of peaks

(e) heating demands

(f) cooling demands

Figure 6: Visualization of (a) parameter learning, (b) net electricity demand, (c) electric loads, (d) empirical counts of peaks, (e) heating demand, and (f) cooling demand for Building 8. It can be observed that Building 8 increases the virtual electricity price during hours 17–23 in response to high electricity peaks. As peak issues are mitigated, virtual electricity prices eventually decline, as can be seen after week 30.



(a) virtual electricity prices $\theta$

(b) net electricity usage

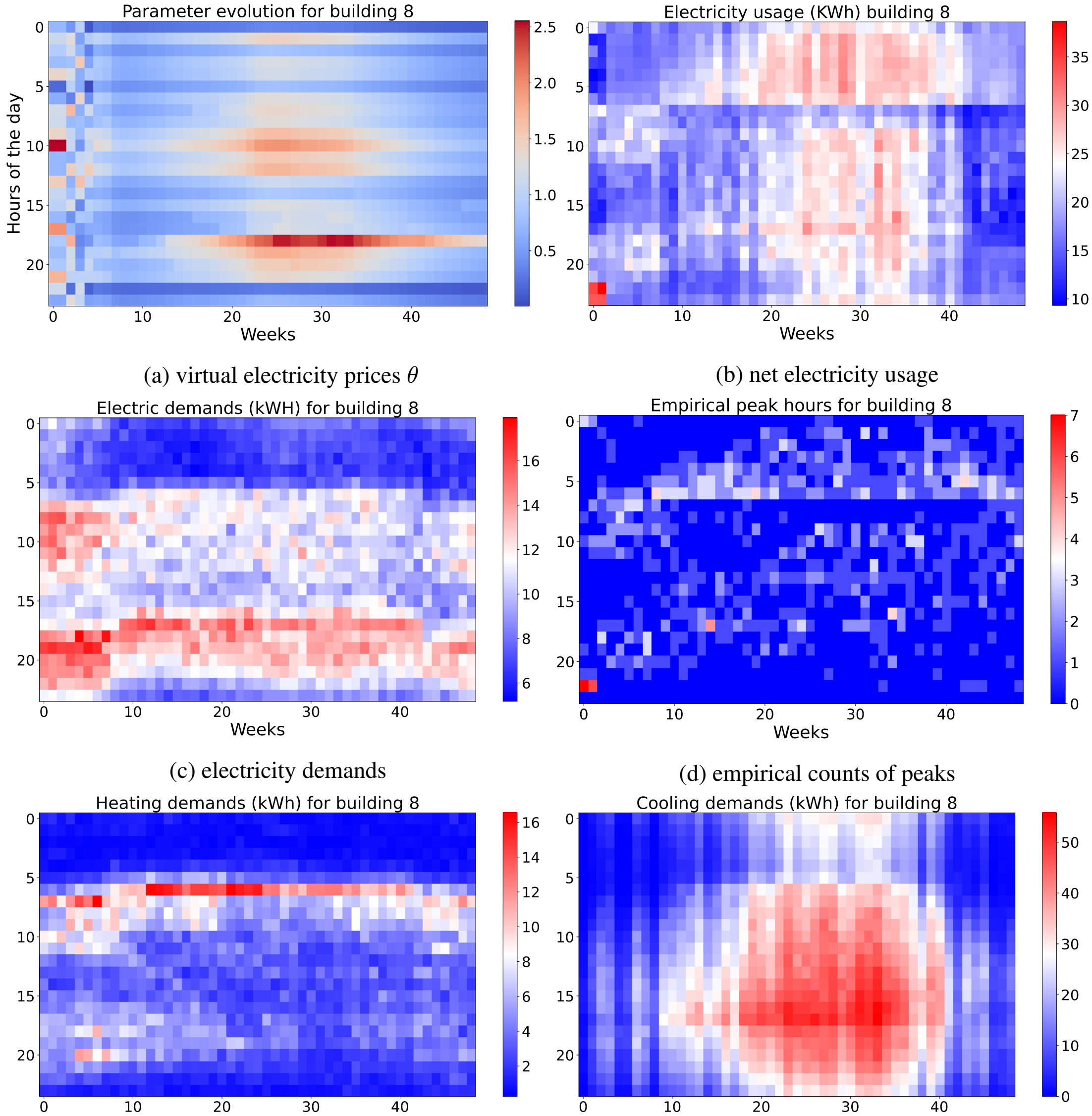
(c) electricity demands

(d) empirical counts of peaks

(e) heating demands

(f) cooling demands

Figure 7: Similar patterns can be observed for Building 6, where virtual electricity prices rise in response to electricity peaks. For this building, the peaks are more dispersed throughout the day.



(a) virtual electricity prices $\theta$

(b) net electricity usage

(c) electricity demands

(d) empirical counts of peaks

(e) heating demands

(f) cooling demands

Figure 8: Similar patterns are observed for Building 8.

### C.5 Sensitivity analysis

In this section, we perform sensitivity analysis of ZO-iRL for different parameters/design choices.

**Initial variance of the transition distribution.** Initial variance $\iota_1$ determines how much randomness we inject during each evolutionary update. Note that in each iteration, we reduce the variance by $1/k^2$. We can observe from Table 10 that the ZO-iRL algorithm is robust to the initial variance of the transition probability.

| | **initial variance** | | |
|---|---|---|---|
| | 0.1 | 0.3 | 0.5 |
| **total score** | 0.960 (5e-4) | 0.961 (7e-4) | 0.962 (0.001) |

Table 10: Performance of ZO-iRL for different initial variance values.

**Number of parameter candidates.** Here, we examine the number of candidates sampled for each update, $N_k \in \{3, 5, 7\}$. In general, the performance on ZO-iRL is robust to this parameter. There is a trade-off between the number of candidates used in each update and the frequency of updates; as we increase $N_k$, we can expect to find a better candidate in the larger pool; however, we may also decrease the frequency of updates as the evaluation of each candidate takes one week in an online setting. As a result, it appears that increasing the number of candidates sampled does not help improve performance.

| | **number of candidates** | | |
|---|---|---|---|
| | 3 | 5 | 7 |
| **total score** | 0.959 (0.001) | 0.964 (0.002) | 0.965 (0.002) |

Table 11: Performance of ZO-iRL for different numbers of sampled candidates per update.

**Guidance signal.** We report the sensitivity of the ZO-iRL algorithm to guidance signal parameters. We consider variants of the guidance signal with respect to 1) number of hours of top electricity use in the past day; 2) incremental value. We keep the guidance learning rate fixed at $\alpha_k = 1$. In the main text, we consider the top-2 hours of electricity usage to be assigned a value of 0.02 (the rest hours are adjusted accordingly so that the sum over all hours of the guidance signal is 0). Here, we consider the following variants: *(a)* top-1 electricity hour to be assigned values of 0.02; *(b)* top-2 electricity hour to be assigned values of 0.04; *(c)* top-3 electricity hour to be assigned values of 0.04; *(d)* top-6 electricity hour to be assigned values of 0.02. We can see from Table 12 that the proposed algorithm is robust to these variants.

| | **guidance parameters** | | | |
|---|---|---|---|---|
| | top-1 | top-2 | top-3 | top-6 |
| **total score** | 0.962 (2e-4) | 0.963 (0.003) | 0.963(0.002) | 0.975 (0.002) |

Table 12: Performance of ZO-iRL for variants of guidance signals.