Information Aggregation for Constrained Online Control

TONGXIN LI, California Institute of Technology, United States

YUE CHEN, Tsinghua University, China

BO SUN, Hong Kong University of Science and Technology, China

ADAM WIERMAN, California Institute of Technology, United States

STEVEN LOW, California Institute of Technology, United States

This paper considers an online control problem involving two controllers. A central controller chooses an action from a feasible set that is determined by time-varying and coupling constraints, which depend on all past actions and states. The central controller's goal is to minimize the cumulative cost; however, the controller has access to neither the feasible set nor the dynamics directly, which are determined by a remote local controller. Instead, the central controller receives only an aggregate summary of the feasibility information from the local controller, which does not know the system costs. We show that it is possible for an online algorithm using feasibility information to nearly match the dynamic regret of an online algorithm using perfect information whenever the feasible sets satisfy a causal invariance criterion and there is a sufficiently large prediction window size. To do so, we use a form of feasibility aggregation based on entropic maximization in combination with a novel online algorithm, named Penalized Predictive Control (PPC) and demonstrate that aggregated information can be efficiently learned using reinforcement learning algorithms. The effectiveness of our approach for closed-loop coordination between central and local controllers is validated via an electric vehicle charging application in power systems.

CCS Concepts: • **Theory of computation** → **Online algorithms**; **Regret bounds**; • **Hardware** → *Smart grid*.

Additional Key Words and Phrases: online control; closed-loop control; model predictive control; regret analysis; electric vehicle charging

ACM Reference Format:

Tongxin Li, Yue Chen, Bo Sun, Adam Wierman, and Steven Low. 2021. Information Aggregation for Constrained Online Control. *Proc. ACM Meas. Anal. Comput. Syst.* 5, 2, Article 18 (June 2021), 34 pages. <https://doi.org/10.1145/3460085>

1 INTRODUCTION

The use of online learning methods for controlling dynamical systems has captured increasing attention from both the learning and control communities. Significant effort has been made to design online optimal controllers using tools from machine learning in a variety of contexts in recent years [7, 9, 12, 13, 31, 48, 49]. One general dynamic model of particular interest for many applications is the following, which has *time-varying* and *time-coupling* constraints:

$$x_{t+1} = f_t(x_t, u_t), \quad x_t \in \mathcal{X}_t(\mathbf{x}_{<t}, \mathbf{u}_{<t}), \quad u_t \in \mathcal{U}_t(\mathbf{x}_{<t}, \mathbf{u}_{<t}), \quad (1)$$

Authors' addresses: Tongxin Li, California Institute of Technology, Pasadena, United States, tongxin@caltech.edu; Yue Chen, Tsinghua University, Beijing, China, cy11@tsinghua.org.cn; Bo Sun, Hong Kong University of Science and Technology, Hong Kong, China, bsunaa@connect.ust.hk; Adam Wierman, California Institute of Technology, Pasadena, United States, adamw@caltech.edu; Steven Low, California Institute of Technology, Pasadena, United States, slow@caltech.edu.

This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2021 Copyright held by the owner/author(s).

2476-1249/2021/6-ART18.

<https://doi.org/10.1145/3460085>

where the deterministic function f_t represents the transition of the state x_t , and u_t is the control or action determined by the controller. Crucially, the constraints $\mathcal{X}_t(\mathbf{x}_{<t}, \mathbf{u}_{<t})$ and $\mathcal{U}_t(\mathbf{x}_{<t}, \mathbf{u}_{<t})$ may change over time and they depend on the past history of states $\mathbf{x}_{<t} = (x_1, \dots, x_{t-1})$ and actions $\mathbf{u}_{<t} = (u_1, \dots, u_{t-1})$. At each time t , the action u_t chosen by the online controller incurs a cost $c_t(u_t)$ and the goal is to minimize the cumulative costs without violating the dynamical constraints.

Designing controllers for the general constrained dynamics in (1) is challenging and, as a result, traditional online optimization models often adopt simplified versions of (1), such as unconstrained optimization [11, 22, 38] or time-invariant constraints [26, 27] that are known a priori. More specifically, previous studies in online control and online optimization mostly focus on specific forms of constraints or costs depending on different applications, such as switching costs [17, 33, 35, 43], ramping constraints [4, 43], polytopic constraints [16], time-varying memoryless cumulative constraints [9, 46, 48–50], convex loss functions with memory [1–3, 42] or inventory constraints [30, 35]. Within this literature, the goal is to derive policies with either small regret [9, 15, 33, 35, 49, 51] or competitive ratio [4, 17, 34, 42, 43]. However, there has been little progress deriving policies with small regret/competitive ratio under general forms of constraints, like in (1).

In particular, our work is motivated by settings where transmitting the constraints precisely is hard, either due to complexity or privacy concerns. These issues often arise in a two-controller system where a *local controller* governs large-scale infrastructure consisting of many controllable devices and a *central controller* operates remotely. Therefore, the control system often contains two controllers: a central controller that wishes to optimize costs but is far away from a large fleet of controllable devices and a *local controller* that has direct access to the controllable devices, whose dynamics can be modeled by (1). In many situations, full information about the local controllers' dynamics and constraints is not available to the central controller and the local controller cannot access the system's costs. In such settings, the central and local controllers each have part of the information needed to control the whole dynamical system online. The task of designing controllers is therefore made even more challenging than the single controller case.

A motivating example for our work is the coordination between a system operator (central controller) and an aggregator (local controller) in large-scale electric vehicle (EV) charging systems, as discussed in [19, 29, 32, 39]. In a smart grid, it is desirable to increase the ability of the system to provide flexibility via distributed energy resources (such as electric vehicles). Aggregators have emerged as dominate players for coordinating these resources and they are able to provide coordination among large pools of DERs and then give a single point of contact for independent system operators (ISOs) to call on for obtaining the aggregate flexibility of DERs [32]. This enables ISOs to minimize cost and respond to unexpected fluctuations of renewables. An example of a system operator is the California ISO, that takes charge of managing and balancing various controllable loads and providing auxiliary services, e.g., demand response, energy storage and flexibility reserves to enhance the system stability and quality. In the case of managing an EV charging garage, on the one hand the aggregator cannot solve the problem independently because it does not have cost information (since the costs are often sensitive and only of the operator's interests) from the operator and even if the aggregator could, it may not have enough power to solve an optimization to obtain an action, such as deciding the substation power level of the EV charging garage. On the other hand, the operator has to receive flexibility information from the aggregator in order to act. Well-known methods in learning or control cannot be used for this problem directly. From a learning perspective, the aggregator cannot simply use reinforcement learning and transmit a learned Q-function or an actor-critic model to the operator because the aggregator does not know the costs. From a control perspective, although model predictive control (MPC) is widely used for EV charging scheduling in practical charging systems [28, 29], it requires precise state information of electric vehicle supply equipment (EVSE). Thus, to solve the induced

MPC problem, the system operator or aggregator needs to solve an online optimization at each time step that involves hundreds or even thousands of variables. This not just a complex problem, but the state information of the EVSE is potentially sensitive. This combination makes controlling subsystems using precise information impractical for a future smart grid [32]. Note that there are a wide variety of other situations that face similar challenges, including data center scheduling [49] and fog computing [12].

The complexities illustrated by the examples above highlight the necessity for system operators to use aggregate information about the constraints of the agents being controlled. However, controlling a dynamical system such as (1) using only aggregate information adds to the difficulty of the design of the controller. Specifically, the two-controller system is now faced with two design tasks: (i) *summarize* – design a form of the aggregate signal summarizing the constraints and dynamics of the controllable devices for a local controller, and (ii) *optimize* – design a central controller that uses the aggregate signal to minimize the cumulative cost while satisfying the constraints of the local controller.

Contributions of this paper. In this work, we consider a system formed by a central controller and a local controller and propose a design of aggregate feedback that integrates into a novel predictive control policy for online control of a system with time-varying and time-coupling constraints. Importantly, the aggregate feedback can be efficiently approximated via a data-driven approach. Our main results bound the dynamic regret of our policy and show that it achieves near-optimal performance. In more detail, our main contributions are four-fold.

First, we design an approach for information aggregation, termed maximum entropy feedback (MEF). MEF, introduced in Section 3.3, is defined as a density function on the action space. We introduce a control policy, *penalized predictive control* (PPC) (see Section 4), that incorporates MEF as a penalty term into an MPC-like policy. This technique for incorporating aggregate feedback is a novel design approach that we expect to have applicability beyond the context of this paper. We also introduce a method to approximate the MEF using model-free reinforcement learning.

Second, denoting by d the diameter of the action space $\mathcal{U}$, T the number of total time steps and w the number of predictions available, we show that the dynamic regret of any deterministic policy must satisfy a lower bound on $\text{Regret}(\mathbf{u}) = \Omega(d(T - w))$ for any feasible sequence of actions $\mathbf{u}$ generated by the deterministic policy, even if it has full information of the constraints. Note that it is well-known that, in the worst case, a sub-linear dynamic regret without the use of predictions is impossible (cf. [24]). Therefore, conditions on the constraints and predictions are necessary to obtain a sub-linear dynamic regret.

Third, we introduce a new assumption, termed *causally invariance* (see Definition 5.1) on the set of feasible actions. The condition holds quite generally, including in applications with inventory constraints (Example 1) and tracking problems (Example 2). We show that when the constraints are causally invariant (see Definition 5.1), the *dynamic regret* of PPC is bounded from above by

$$\text{Regret}(\mathbf{u}) = O\left(dT\left(\frac{\delta \log \lambda}{\sqrt{w}} + \frac{\sqrt{\delta}}{w^{1/4}}\right)\right)$$

where $\mathbf{u}$ is the sequence of actions generated by PPC, $\lambda, \delta > 0$ are parameters of the causal invariance assumption, w is the prediction window size, and d is the diameter of the action space. In particular, when $w = \omega(1)$, even with aggregate feedback information, PPC establishes a sub-linear (in T) bound on the dynamic regret. To the best of our knowledge, this is the first bound on dynamic regret for a policy that uses only aggregate feedback information in the context of time-varying and time-coupling constraints.

Finally, to illustrate the performance of our controller in a practical setting, in Section 7 we consider a cost minimization problem for an EV charging system controlled by a system operator. We provide experimental results first showing that in the case when there is no prediction, our approach can reduce the number of variables by 80% while outperforming classical MPC with full information were available by achieving a lower total cost. We demonstrate that the cost can be further reduced with predictions.

Related work. This model falls into the growing literature seeking to understand the role of information and feedback in online decision-making. In this line of work, the online controller does not know (at least) one of the objective function, dynamics or constraints in advance, and therefore, feedback containing information about the objective function, dynamics and constraints becomes necessary.

When the cost functions are unknown, using a semi-definite relaxation for an online controller, [14] shows a $O(\sqrt{T})$ regret where T is the length of the time horizon for quadratic costs and stochastic noise. It is well known that without the curvature assumptions, $O(\sqrt{T})$ regret is tight [23]. Similarly, in [1], $O(\sqrt{T})$ regret is proved for convex costs with adversarial noise. Recently, with the help of online learning, [2] considers a linear dynamical system under adversarially changing strongly convex cost functions, which includes the Kalman filter and the linear quadratic regulator. Assuming the transition dynamics are known, the authors in [2] obtain a logarithmic upper bound $O(\text{poly}(\log T))$ on the regret. When the dynamics are not known, nonlinear control with safety constraints have been considered in [6], wherein an algorithm is introduced to safely learn about the dynamics modeled by a Gaussian process.

When the constraints and costs are not known a priori, results are less general and often hold for specific forms of constraints. To help the online controller be aware of the constraint information, feedback is necessary. Two widely adopted feedback signals are bandit feedback [9, 12], which reveals the values of the objective and constraint functions with certain chosen actions, and gradient feedback [7, 13, 48, 49], which further offers the gradient of the unknown functions. To briefly mention some of the results, in [12], with bandit feedback, the dynamic regret is bounded by $o(T)$. In [9], the dynamic regret is bounded by $O(\sqrt{T\Delta(T)})$ and the constraint violations are bounded by $O(T^{3/4}\Delta(T)^{1/4})$ where $\Delta(T)$ is the “drift” of the sequence of optimal actions. In [31], $O(\sqrt{T})$ regret is proven, and the results are extended to incorporate future information (predictions). In the case of stochastic long-term constraints, the authors in [49] achieve $O(\sqrt{T} \log T)$ regret and constraint violations with high probability. However, both bandit and gradient feedback are not designed to deal with time-coupling constraints and there are no results providing guaranteed performance for the general setting in (1). Indeed, for the case when the offline constraints on the actions $\mathbf{u} = (u_1, \dots, u_T)$ are of the form $\sum_{t=1}^T g_t(u_t) \leq 0$, [37] shows that if the feedback at time t contains only information on the function g_t and a convex cost c_t , it is impossible to achieve both sub-linear regret and constraint violations even if the functions g_t and c_t are linear.

Finally, note that the literature described above typically compares algorithms with the best *fixed* decision in hindsight, with notable exceptions such as [12, 48], which shows sub-linear dynamic regret and constraint violations. This is in contrast to the current paper which compares to dynamic optimal decisions in hindsight.

Notation and Conventions. The (differential) entropy function is denoted by $\mathbb{H}(\cdot)$. Sequences of vectors are written as boldface letters, such as $\mathbf{u} = (u_1, \dots, u_T)$ and u_t is a vector in an Euclidean space. To distinguish random variables and their realizations, we follow the convention to denote the former by capital letters (e.g., U and $\mathbf{U}$) and the latter by lower case letters (e.g., u and $\mathbf{u}$). We fix the base of the logarithms to be the natural number e , unless otherwise stated. The concatenation of two sequences $\mathbf{x}$ and $\mathbf{y}$ is denoted by $(\mathbf{x}, \mathbf{y})$. The ℓ_2 norm of a vector x is written as $\|x\|_2$. Let

Fig. 1. Closed-loop interaction between a central controller and a local controller.

$\mu(S)$ be the Lebesgue measure of a measurable set S . The Hausdorff distance between two sets S_1 and S_2 in an Euclidean space with respect to the ℓ_2 norm is denoted as

$$d_H(S_1, S_2) := \max \left\{ \sup_{x \in S_1} \inf_{y \in S_2} \|x - y\|_2, \sup_{y \in S_2} \inf_{x \in S_1} \|x - y\|_2 \right\}.$$

2 MODEL

We consider the deterministic dynamical system in (1) over a discrete time horizon $[T] := \{1, \dots, T\}$ with time-varying and time-coupling constraints.

The dynamical system is governed by a *local controller* ψ , which manages a large fleet of controllable units. The collection of the states of the units is represented by x_t in a state space $X \subseteq \mathbb{R}^n$.

There is a distant *central controller* π that communicates with the local controller. The central controller selects an action u_t at each time $t \in [T]$. The actions must be selected from a closed and bounded domain $U \subseteq \mathbb{R}^m$. The initial point u_0 is assumed to be the origin without loss of generality.

Both the state and action at each time are confined by *safety constraints* that maybe time-varying and time-coupling, i.e., $x_t \in X_t(x_{<t}, u_{<t})$ and $u_t \in \mathcal{U}_t(x_{<t}, u_{<t})$ for $t \in [T]$. For simplicity, we denote the *safety sets* $\mathcal{U}_t(x_{<t}, u_{<t})$ and $X_t(x_{<t}, u_{<t})$ by $\mathcal{U}_t$ and X_t in future contexts. The central controller receives time-varying cost functions online from an external environment and each $c_t(\cdot) : U \rightarrow \mathbb{R}_+$ only depends on the action u_t chosen by the central controller. We assume that the local controller does not know the costs and has to choose the action given by the central controller and the central controller cannot access the constraints directly, but some information about the constraints, summarized as some feedback p_t , can be transmitted during the control (more details of p_t will be given in Section 3.1). In settings with constraints, predictions are crucial to maintaining feasibility and reducing costs. Thus, in this work, we suppose the online controller has (perfect) predictions of the cost functions and feedback functions of the current and the next w time slots. We formally define the predictions available in Section 3.2. The goal of an online control policy in this setting is to make the local and central controllers jointly minimize a cumulative cost $C_T(u) := \sum_{t=1}^T c_t(u_t)$ while satisfying (1). Our system model is shown in Figure 1.

Throughout this paper, we make the following assumptions on the model.

Assumption 1. The dynamic $f_t(\cdot, \cdot) : X_t \times \mathcal{U}_t \rightarrow X_{t+1}$ is a Borel measurable function for $t \in [T]$.

Assumption 2. The action space $U \subseteq \mathbb{R}^m$ is closed and bounded.

Assumption 3. The safety sets $\{\mathcal{U}_t : t \in [T]\}$ and $\{X_t : t \in [T]\}$ are Borel sets in $\mathbb{R}^m$ and $\mathbb{R}^n$. Furthermore, the safety sets are atoms, i.e., $\mu(X_t) > 0$ and $\mu(\mathcal{U}_t) > 0$ for all $t \in [T]$ if $X_t, \mathcal{U}_t \neq \emptyset$.

Additionally, we adopt the following smoothness condition for cost functions.

Assumption 4. For each $t \in [T]$, the cost function $c_t(\cdot) : \mathcal{U} \rightarrow \mathbb{R}_+$ is Lipschitz continuous. We assume that there exists a Lipschitz constant $L_c > 0$ such that

$$|c_t(u) - c_t(v)| \leq L_c \|u - v\|_2 \quad \text{for all } u, v \in \mathcal{U} \text{ and } t \in [T].$$

2.1 Dynamic regret

The focus of this paper is the analysis of worst-case bounds on the *dynamic regret* [21, 52], which is the difference between the cost of the algorithm and that of the offline optimal decision. Formally, the offline optimal cost with full information of the functions and the safety sets is defined as:

$$C_T^* := \inf_{\mathbf{u}} C_T(\mathbf{u}) \tag{2a}$$

$$\text{subject to (1), } \forall t \in [T], \tag{2b}$$

and the dynamic regret, $\text{Regret}(\mathbf{u})$, is:

$$\text{Regret}(\mathbf{u}) := \sup_{\mathbf{c} \in \mathcal{C}} \sup_{\mathbf{f} \in \mathcal{F}} \sup_{(U, X) \in \mathcal{I}} C_T(\mathbf{u}) - C_T^* \tag{3}$$

where $\mathbf{u}$ is the sequence of actions generated by the online policy π , $\mathbf{f} := (f_1, \dots, f_T)$ denotes a sequence of dynamics chosen from a set of Borel measurable functions $\mathcal{F}$ satisfying Assumption 1, $\mathbf{c} := (c_1, \dots, c_T)$ denotes a sequence of cost functions chosen from the set of all Lipschitz continuous functions $\mathcal{C}$; $\mathcal{U} := (\mathcal{U}_1, \dots, \mathcal{U}_T)$ and $\mathcal{X} := (\mathcal{X}_1, \dots, \mathcal{X}_T)$ are the collections of safety constraints. It is important to note that without any restrictions on $\mathcal{U}$ and $\mathcal{X}$, $\text{Regret}(\mathbf{u})$ can be no better than $\Omega(T)$ for any deterministic online policy π , even with predictions (see Theorem 5.1 for more details). Therefore, the focus of this paper is to find conditions on $(\mathcal{U}, \mathcal{X})$ so that given enough predictions, the regret can be bounded by a sub-linear (in T) function. We denote by $\mathcal{I}$ the domain of safety constraints $(\mathcal{U}, \mathcal{X})$ satisfying certain conditions depending on the contexts and will formally state the conditions (such as the *causal invariance criterion* used in Theorem 5.3) in the theorem statements. In the remainder of the paper, we sometimes write $\text{Regret}(\pi)$, in replacement of $\text{Regret}(\mathbf{u})$ defined in (3), with $u_t = \pi(\mathbf{c}_{t:t+w-1}, \mathbf{p}_{t:t+w-1}(\cdot | \mathbf{u}_{<t}))$. It is worth mentioning that, to the best of our knowledge, there is no existing bound on the dynamic regret defined in (3) in the current setting, nor are there existing results for a similar worst-case metric called *competitive ratio* [11, 34–36, 43].

2.2 Example: Coordination between an operator and an EV charging system

The model described above may seem abstract. We now illustrate it with a concrete example of EV charging that we have briefly mentioned in Section 1. Consider a local controller that manages multiple EV chargers in an EV charging system. The local controller, known as an *aggregator*, interacts with a central controller that operates the electricity grid, known as a *system operator*. The aggregator is able to provide coordination services among large pools of DERs such as batteries and EVs and it serves as a single point of contact for the system operator to call on for demand response. A common objective of the operator is to minimize the cumulative electricity costs, subject to the constraints imposed by each individual DER. This example will also be used in our experiments in Section 7. Figure 2 in Section 7 depicts the closed-loop interaction between the local controller and the central controller, in the context of EV charging.

2.2.1 Local controller constraints. Let N be the total number of accepted EV charging sessions. For each EV $j \in [N] := \{1, \dots, N\}$, it has a length-4 private vector $(a(j), d(j), e(j), r(j)) \in \mathbb{R}^4$ where $a(j)$ denotes its arrival (connecting) time; $d(j)$ denotes its departure (disconnecting) time, normalized according to the time indices in $[T]$; $e(j)$ denotes the total energy to be delivered, and $r(j)$ is its peak charging rate. Fix a set of N users with their private vectors $(a(j), d(j), e(j), r(j))$,

the system state x_t at time $t \in [T]$ is a collection of length-2 vectors $x_t := (d_t(j), e_t(j))$ for each EV that has arrived and has not departed by time t . Here $0 \leq d_t(j) \leq T$ denotes the remaining charging time and $0 \leq e_t(j)$ is the remaining energy demand of user j at time t . The charging system is running a scheduling policy, ϕ , which can be well-known scheduling policies such as earliest-deadline-first, least-laxity-first, etc. For an active EV $j \in [N]$, the charging decision $\phi_t(j) \in \mathbb{R}_+$ at each time $t \in [T]$ updates the remaining energy $e_t(j)$ and charging time $d_t(j)$ (unit 1), assuming there is no power loss:

$$d_t(j) = d_{t-1}(j) - \Delta \quad (4)$$

$$e_t(j) = \begin{cases} [e_{t-1}(j) - \phi_t(j)]_+, & a(j) < t < d(j) \\ e(j), & t = a(j) \\ 0, & t = d(j) \end{cases} \quad (5)$$

where Δ is a fixed time interval; $d_0(j) \equiv d(j)$ and $e_0(j) \equiv e(j)$ and $[x]_+ := \max\{x, 0\}$. The energy demand of each EV needs to be satisfied, leading to the following charging constraints:

$$e_t(j) = 0, \quad \forall d_t(j) = 0, \quad (6)$$

$$e_{t-1}(j) - e_t(j) \leq r(j), \quad \forall t \in [T]. \quad (7)$$

2.2.2 Central controller's objective. The charging system manager needs to communicate with a grid operator, which solves an online optimization. Besides satisfying the charging demands, the charging decisions also produce an action

$$u_t = \sum_{j: a(j) \leq t \leq d(j)} \phi_t(j) \quad (8)$$

representing the aggregate charging energy. The action is chosen from a set of power levels $\mathcal{U}$. The actions affects the loads on the power grid and the operational cost. Let $c_1, \dots, c_T$ be the electricity price at each time $t \in [T]$. The goal of the grid operator is to solve the following online optimization problem:

$$\min_{u \in \mathcal{U}^T} \sum_{t=1}^T c_t u_t \text{ subject to } (4) - (8). \quad (9)$$

Note that new EVs may arrive at each time slot and the constraint at each time $t \in [T]$ depends on past charging decisions so the constraints in (6)-(8) are *time-varying* and *time-coupling*. The action space is the set of substation power levels. It is often much smaller than the number of states, which is the number of chargers in this example. In Section 7, using this EV charging example we provide experimental results and exemplify concrete choices of system parameters for our proposed policy.

In this work we introduce the design of a predictive controller that achieves near optimal dynamic regret via aggregate feedback with a sufficiently large prediction window size. It is worth emphasizing that it is difficult to achieve optimal dynamic regret even when full information is available (see Theorem 5.1). So it is a non-trivial problem to figure out what information of the constraints is the most useful and what conditions on the constraints can help. We answer the first problem in the next section by formally defining a feedback function p_t (and its predictions) and handle the second issue later by presenting our main results in Section 5. The two key aspects of the design are (i) how to design the aggregate feedback signal and (ii) how to use the combination of predictions and aggregate feedback to determine control actions in an online setting. We describe each in turn in Section 3 and Section 4.

3 INFORMATION AGGREGATION

We first present the design of feedback in the two-controller system.

3.1 Limited Feedback Information

A distinctive feature of this paper is the question of how to design online controllers when limited information is available. Limitations on the information available may occur because (1). the central controller is distant from the local controller and sending the full information about the dynamical system renders communication issues, (2). the central controller may have other tasks running in parallel and its computational power is limited and (3). the size n of the state space is much larger than the size m of the action space, preventing the central controller from accessing the full state $\{x_t \in \mathbb{R}^n : t \in [T]\}$ and the safety sets $\{\mathcal{U}_t : t \in [T]\}$ and $\{\mathcal{X}_t : t \in [T]\}$ from the local controller. For example, consider an electric vehicle charging station described earlier in Section 2.2, there may be hundreds or even thousands of electric vehicle chargers. Each of the chargers has a state and the overall state vector has a high dimension. However, there are often only a few choices of the sub-station power levels to be chosen by a remote system operator (see Section 7 for more details about a realistic EV charging setting). (4). Alternatively, limitations may result because of privacy concerns, e.g., EV owners do not want to directly share their charging session information with a third-party, making transmission of the exact constraints to the central controller undesirable.

To capture such limitations, we consider a setting where the central controller receives a simplified feedback signal summarizing the state and safety sets. These signals have domains over the action space $\mathcal{U}$. The regime of interest is when the action space is much smaller than the state space, so sending the density functions requires much less communication resources than sending the states and constraints. While this definition is abstract, recall that we give a concrete example in the case of power systems (see Section 2.2). In the following, we formally state the form of the feedback and make the following model assumption on the feedback sent from the local controller to the central controller.

Assumption 5. *At each time $t \in [T]$, the local controller is allowed to send a feedback (density) function whose domain is the action space $\mathcal{U}$, denoted by $p_t(\cdot) : \mathcal{U} \rightarrow [0, 1]$ to the central controller.*

Remark 1. At first glance transmitting a density function from a local controller to a central controller seems challenging. In practice, given the regime that we are interested in when the action space dimension $|\mathcal{U}|$ is much smaller than the state space dimension X , sending a quantized density function simplifies the communication compared to sending the whole state. Further, in certain applications the action space is discrete, e.g., the EV charging application described in Section 2.2, so p_t becomes a probability vector whose length is much smaller than the state vector (see the case study in Section 7 for more details). Moreover, later in Section 7 we demonstrate that an data-driven feedback generator can be pre-trained and used ad hoc for the aggregator as a black box. So the aggregator does not need as much computational power as required for solving optimization problems. More discussion about learning the feedback function can be found in Section 6. Note that the online control cannot be directly implemented in a local controller, because the central controller may not want the local controller to know the exact process of how the actions are generated. When generalizing to a multi-controller system, the central controller needs to take into account of information from other aggregators as well. The discussion of this setup is beyond the scope of this article and would be an interesting extension.

Since the dynamic system in (1) has memory, the feedback function also depends on previous actions. To be more precise, at each time $t \in [T]$, a (conditional) density function denoted by $p_t(\cdot | \mathbf{u}_{<t}) : \mathcal{U} \rightarrow [0, 1]$, given past actions $\mathbf{u}_{<t}$ is sent from the local controller to the central

controller. The density function contains only information about the constraints. A natural question is how to design such feedback, and we discuss this question in Section 3.3.

3.2 Predictions

For general dynamics such as (1), it is well-known that, in the worst case, a sub-linear dynamic regret without the use of predictions is impossible (cf. [24]). In many applications, while the cost functions are not known a priori, predictions of future cost functions are available. The question of how to make use of such predictions for online control and optimization has received considerable attention in recent years, e.g., [11, 33, 34, 36]. Formally, at time $t \in [T]$, the cost functions $c_t, \dots, c_{t+w-1}$ are given to the central controller from an oracle and the joint feedback function $p_{t:t+w-1}(\cdot | \mathbf{u}_{<t}) : \mathcal{U}^w \rightarrow [0, 1]$, as a density function on the subsequence of actions $\mathbf{u}_{t:t+w-1}$, are sent to the central controller from the local controller where $w > 0$ is an integer denoting the *prediction window size*. Let $\mathcal{C}_{<t+w}$ and $\mathcal{P}_{<t+w}$ be the sets of cost functions and feedback functions received by the central controller at time $t \in [T]$. The goal of the central controller is to design an *online policy* $\pi : \mathcal{C}_{<t+w} \times \mathcal{P}_{<t+w} \rightarrow \mathcal{U}$ that generates actions. This model for predictions that we consider is standard, and has been adopted in [11, 33, 36]. While the assumption that predictions are perfect is overly optimistic, the insights derived typically extend to settings with inexact predictions, albeit with considerable technical effort, e.g., [8, 10, 18]. We denote by $\psi : \mathcal{X} \rightarrow \mathcal{P}_{t:t+w-1}$ an *information aggregation function* that outputs a sequence of feedback predictions and $\mathcal{P}_{t:t+w-1}$ is the set of feedback functions $p_t, \dots, p_{t+w-1}$.

3.3 Aggregation Feedback: Maximum entropy feedback (MEF)

A key feature of our model is the aggregate feedback signal p_t provided to the controller. Such an aggregate signal p_t is needed in many situations for a variety of reasons. One prominent motivation is that it can be hard to transmit complicated constraints precisely from large-scale controllable units to the agent as a result of communication limitations. Another common motivation is that the information structure among the agent and units is asymmetric, and the dynamical operating parameters of each controllable unit are private. As a result, it is undesirable and impractical to ask for exact constraints and system states. To ensure the feasibility of the chosen action profile while avoiding the leakage of private parameters, one compromise is to aggregate the necessary information via a feedback signal p_t , defined earlier in Assumption 5. In the following we consider a special design of p_t .

Note that the design of an aggregate feedback signal must balance possibly competing goals. First, it must contain the information about *what actions are feasible*. Second, it must contain information about the *future impact* of the actions to be selected on the system's feasibility. Finally, it must be compact, leaking as little individually identifiable information as possible.

Before we proceed to the details of our proposed design, we first introduce some useful notation.

3.3.1 Set of all feasible actions $\mathcal{S}$. We begin with defining a set $f^{-1}(\mathcal{X}_t)$ denoting the inverse image of the safety set $\mathcal{X}_t$ for actions: $f^{-1}(\mathcal{X}_t)(\mathbf{u}_{<t}) := \{u \in \mathbb{R}^m : f(x_{t-1}, u) \in \mathcal{X}_t\}$. The inverse image $f^{-1}(\mathcal{X}_t)$ depends only on the past actions $\mathbf{u}_{<t}$ since the states $\mathbf{x}_{<t}$ are determined by $\mathbf{u}_{<t}$ via the dynamics in (1). Note that $\mathcal{X}_t$ and the dynamic f are Borel measurable by Assumption 1 and 3. Therefore the inverse image $f^{-1}(\mathcal{X}_t)$ is also a Borel set, implying that the intersection $\mathcal{U}_t \cap f^{-1}(\mathcal{X}_t)$ is also Borel measurable. Denote by

$$\mathcal{S} := \left\{ \mathbf{u} \in \mathcal{U}^T : u_t \in \mathcal{U}_t \cap f^{-1}(\mathcal{X}_t), \forall t \in [T] \right\}$$

the non-empty Borel measurable set of all *feasible* sequences of actions satisfying the constraints in (11b). Since the actions space $\mathcal{U}$ is bounded, the set $\mathcal{S} \subseteq \mathcal{U}^T$ is also bounded.

3.3.2 *Set of all subsequent feasible actions* S_k . We denote the set of subsequent feasible trajectories by:

$$S_k(\mathbf{u}_{\leq t}) := \left\{ \mathbf{v}_{t+1:t+k} \in \mathcal{U}^k : \mathbf{v}_{\leq t} \equiv \mathbf{u}_{\leq t}; \mathbf{v} \in S \right\}, \quad (10)$$

which consists of all feasible k actions at time $t + 1, \dots, \min\{t + k, T\} \in [T]$, given the past actions $\mathbf{u}_{\leq t}$. For the case when all future actions are considered, we simplify $S_k(\mathbf{u}_{\leq t})$ as $S(\mathbf{u}_{\leq t})$.

The core of our design is the following *maximum entropy feedback* (MEF) proposed by [32] in a discrete form, which provides a way for summarizing the time-varying and coupling safety sets $\{\mathcal{U}_t \cap f^{-1}(X_t) \subseteq \mathbb{R}^m : t \in [T]\}$. The intuition behind our definition is that the conditional density function $p_t(x_t | \mathbf{x}_{<t})$ encapsulates the resulting future flexibility of the constraints if the agent chooses x_t as the action at time t , given the previous actions up to time $t - 1$. The sum of the conditional entropy of p_t is thus a measure of the information in p_t . This suggests choosing a conditional density function p_t that maximizes its conditional entropy.

Definition and properties. Consider the stochastic optimization problem:

$$\sup_{p_1, \dots, p_T} \sum_{t=1}^T \mathbb{H}(U_t | \mathbf{U}_{<t}) \quad (11a)$$

$$\text{subject to } U_t \in \mathcal{U}_t \cap f^{-1}(X_t), \forall t \in [T] \quad (11b)$$

where the variables are density functions $p_t := p_t(\cdot | \cdot) := \mathbb{P}_{U_t | \mathbf{U}_{<t}}(\cdot | \cdot)$; $\mathbf{U}$ is a random variable distributed according to the joint distribution $\prod_{t=1}^T p_t$; and $\mathbb{H}(U_t | \mathbf{U}_{<t})$ is the differential conditional entropy of $p_t(\cdot | \cdot)$.

By definition, a quantity conditioned on “ $u_{<1}$ ” means an unconditional quantity, so in the above, $\mathbb{H}(U_1 | \mathbf{U}_{<1}) := \mathbb{H}(U_1) := \mathbb{H}(p_1)$. The chain rule shows that $\sum_{t=1}^T \mathbb{H}(U_t | \mathbf{U}_{<t}) = \mathbb{H}(\mathbf{U})$. Hence (11) can be interpreted as maximizing the entropy $\mathbb{H}(\mathbf{U})$ of a random trajectory $\mathbf{U}$ sampled according to the joint distribution $\prod_{t=1}^T p_t$, conditioned on $\mathbf{U}$ satisfying (11b), where the maximization is over the collection of conditional distributions $(p_1, \dots, p_T)$.

Definition 3.1 (Maximum entropy feedback). *The conditional density function p_t^* for $t \in [T]$ is called the maximum entropy feedback (MEF) if $(p_1^*, \dots, p_T^*)$ is the unique optimal solution of (11).*

Even though the optimization problem (11) involves variables p_t for the entire time horizon $[T]$, the individual variable p_t in (11) is a conditional density function that depends only on information available at time t . Therefore the maximum entropy feedback p_t^* in Definition 3.1 is indeed causal. The existence and uniqueness of p_t^* are guaranteed by Lemma 1 below.

The MEF possesses some nice properties summarized below.

PROPOSITION 1. *Let p_t^* be the the maximum entropy feedback given by (11a)-(11b) at each time $t \in [T]$.*

- (1) (Feasibility). *For any sequence of actions $\mathbf{u} = (u_1, \dots, u_T)$, $p_t^*(u_t | \mathbf{u}_{<t}) > 0$ for all $t \in [T]$ if and only if $\mathbf{u} \in S$.*
- (2) (Interpretability). *For all $u_t, u'_t \in \mathcal{U}$ at each time $t \in [T]$, if $p_t^*(u_t | \mathbf{u}_{<t}) \geq p_t^*(u'_t | \mathbf{u}_{<t})$, then $\mu(S(\mathbf{u}_{<t}, u_t)) \geq \mu(S(\mathbf{u}_{<t}, u'_t))$ where $\mu(\cdot)$ denotes the Lebesgue measure.*

The proof of Proposition 1 can be found in the supplementary materials (see Appendix B.2). The key part for the proof relies on the following explicit expression of the MEF.

LEMMA 1. Let $\mu(\cdot)$ denote the Lebesgue measure. The MEF as optimal solutions of the maximization in (11a)-(11b) are given by¹

$$p_t^*(u|u_{<t}) \equiv \frac{\mu(S((u_{<t}, u)))}{\mu(S(u_{<t}))}, \quad \forall u \in U \text{ and } u_{<t} \in U^{t-1}. \quad (12)$$

Moreover, the optimal value of (11a)-(11b) is equal to $\log \mu(S)$.

The proof of Lemma 1 is given in Appendix B.1. Proposition 1 says that the density function p_t^* provides the right information for the online agent to choose its action u_t at time $t \in [T]$. Specifically, the first statement of the proposition (feasibility) says that if the operator always chooses an action u_t with a positive value $p_t^*(u_t|u_{<t}) > 0$ for each time t , then the resulting sequence of actions is guaranteed to be feasible, $u \in S$, i.e., the system will remain feasible at every time t along the way. Further, according to the second statement of the proposition, if the online agent chooses an action u_t with a larger $p_t^*(u_t)$ value at time t , then the agent will be more flexible going forward than if it had chosen another signal u'_t with a smaller $p_t^*(u'_t)$ value, in the sense that there are more feasible trajectories in $S((u_{<t}, u_t))$ going forward.

It is remarkable that the MEF only depends on the constraints, but not the costs. Further, the theoretical definition we present in (11a)-(11b) involves the offline information of S , the set of all feasible sequences of actions. This leads to computational difficulties; however learning techniques can be used to generate the MEF, as we describe in Section 6.

4 PENALIZED PREDICTIVE CONTROL VIA AGGREGATE FEEDBACK

In this section, with the two-controller system described in Section 2 and the feedback and predictions defined in Section 3, we are ready to formally present a closed-loop control framework and the associated online algorithm.

4.1 Key Idea: Maximum entropy feedback is a penalty term for the offline optimal

When first seeing the definition of MEF it is not immediately clear why it is useful in the context of predictive control. To make that clear, in this section we highlight the key idea in the design of our controller – MEF can act as an effective penalty term in the offline optimization problem. More specifically, there is in general a trade-off between ensuring future flexibility and minimizing the current system cost in predictive control. The action u_t guaranteeing the maximal future flexibility, i.e., having the largest $p_t^*(u_t|u_{<t})$ may not be the one that minimizes the current cost function c_t and vice versa. Therefore, we need to design an online algorithm for the central controller that balances the MEF and the cost functions.

To further illustrate this point, note that Proposition 1 guarantees that the online agent can always find a feasible action $u \in S$. Indeed, knowing the MEF p_t^* for every $t \in [T]$ is equivalent to knowing the set of all admissible sequences of actions S . Using this observation, the constraints (2b) in the offline optimization can be rewritten as a penalty in the objective of (2a). Formally, the offline optimization can be recast as the following.

LEMMA 2. The offline optimization (2a)-(2b) is equivalent to the following unconstrained minimization for any $\beta > 0$:

$$\inf_{u \in U^T} \sum_{t=1}^T (c_t(u_t) - \beta \log p_t^*(u_t|u_{<t})) \quad (13)$$

¹When the denominator $\mu(S(u_{<t}))$ is zero, the numerator $\mu(S((u_{<t}, u)))$ has also to be zero. For this case, we set $p_t^*(u|u_{<t}) = 0$ and this does not affect the optimality of (11a)-(11b).

This draws a clear connection between MEF and the offline optimal, which we exploit in the design of an online controller in the next section.

PROOF. The offline optimization (2a)-(2b) is equivalent to

$$\inf_{\mathbf{u} \in \mathcal{U}^T} \sum_{t=1}^T c_t(u_t) - \beta \log q(\mathbf{u}) \quad (14)$$

for any $\beta > 0$ and $q(\mathbf{u})$ is a uniform distribution on $\mathcal{S}$:

$$q(\mathbf{u}) := \begin{cases} 1/\mu(\mathcal{S}) & \text{if } \mathbf{u} \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases}. \quad (15)$$

Further, decomposing the joint distribution $q(\mathbf{u}) = \prod_{t=1}^T p_t^*(u_t | \mathbf{u}_{<t})$ into the conditional distributions given by (11a)-(11b), the objective function (14) becomes

$$\sum_{t=1}^T c_t(u_t) - \beta \log \left(\prod_{t=1}^T p_t^*(u_t | \mathbf{u}_{<t}) \right) = \sum_{t=1}^T (c_t(u_t) - \beta \log p_t^*(u_t | \mathbf{u}_{<t})). \quad (16)$$

□

4.2 Algorithm: Penalized Predictive Control via maximum entropy feedback

Our proposed design, termed Penalized Predictive Control (PPC), is a combination of Model Predictive Control (MPC), which is a competitive policy for online optimization with predictions, and the idea of using MEF as a penalty term. This design makes a connection between the MEF and the well-known MPC scheme. The MEF as a feedback function, only contains limited information about the dynamical system in the local controller's side. (It contains only the feasibility information of the current and future time slots, as explained in Section 3.3). The PPC scheme therefore is itself a novel contribution since it shows that, even if *only* feasibility information is available, it is still possible to incorporate the limited information to MPC as a *penalty term*. Moreover, this innovation allows PPC to achieve nearly optimal dynamic regret despite having only aggregate feasibility information about constraints and dynamics, a setting where no prior algorithms have any provable guarantees.

Algorithm 1: Penalized Predictive Control (PPC)

Input : Sequential cost functions and MEF with a prediction window size w

Output: Actions $\mathbf{u} = (u_1, \dots, u_T)$

for $t = 1, \dots, T$ **do**

if $t \in \mathcal{I}$ **then**

 Choose an action u_t by minimizing over the next w time slots using (17)-(18)

end

end

Return $\mathbf{u}$;

We present PPC in Algorithm 1, where we use the following notation. Let $t' := \min\{t + w - 1, T\}$. Define a set of time indices $\mathcal{I} := \{t \in [T] : t \equiv 1 \pmod{w}\}$. Consider for any $t \in \mathcal{I}$:

$$\mathbf{u}_{t:t'} = \arg \inf_{\mathbf{u}_{t:t'}} \sum_{\tau=t}^{t'} (c_\tau(u_\tau) - \beta \log p_\tau^*(u_\tau | \mathbf{u}_{<\tau})) \quad (17)$$

$$\text{subject to } \mathbf{u}_{t:t'} \in \mathcal{U}^{t'-t+1} \quad (18)$$

where in above $\beta > 0$ is a *tuning parameter* in predictive control to trade-off the flexibility in the future and minimization of the current system cost. In the remainder of the paper, we denote by π_{PPC} the online policy that uses the PPC scheme.

4.3 Framework: Closed-loop control between local and central controllers

Algorithm 2: Closed-loop online control framework

for $t = 1, \dots, T$ **do**

end

Return Total cost C_T ;

Given the PPC scheme described in Section 4, we can now formally present our online control framework for the distant central controller and local controller (defined in Section 2). An overview is given in Algorithm 2, where π_{PPC} denotes a *PPC online policy* and ψ_{IA} a specific *information aggregation function* (see Section 6 for an example of learning-based ψ_{IA}). To the best of our knowledge, this paper is the first to consider such a closed-loop control framework with limited information communicated in real-time between two geographically separate controllers seeking to solve an online control problem. We present the framework below.

At each time $t \in [T]$, the local controller first efficiently generates an estimated MEF sequence $\mathbf{p}_{t:t+w-1} := (p_t, \dots, p_{t+w-1}) \in \mathcal{P}$ using an information aggregation function (for example, in Section 7 we use a reinforcement learning algorithm to train an information aggregation function). After receiving the MEF $\mathbf{p}_{t:t+w-1}$ and cost functions with predictions $\mathbf{c}_{t:t+w-1} = (c_t, \dots, c_{t+w-1})$, the central controller uses the PPC scheme in Algorithm 1 to generate an action $u_t \in \mathcal{U}$ and sends it back to the local controller. The local controller then updates its state $x_t \in \mathcal{X}$ to a new state x_{t+1} based on the system dynamic in (1) and repeats this procedure again. Later in Section 5.1, we show that if the generated MEF is exact, i.e., $p_t = p_t^*$, for every $t \in [T]$, then feasibility can be ensured, i.e., $x_t \in \mathcal{X}_t$ and $u_t \in \mathcal{U}_t$ for every $t \in [T]$ and with assumptions, a sub-linear (in T) bound on the dynamic regret is possible with $w = \omega(1)$ using our online control scheme presented above.

5 RESULTS

In this section we state our main results, which guarantee feasibility and bound the dynamic regret of the PPC controller. Additionally, we present a lower bound for any online deterministic policy that have full information about safety constraints. This lower bound highlights the near optimality of PPC.

5.1 Feasibility

To begin, we highlight that PPC always yields a feasible trajectory of actions. To see this, recall that Proposition 1 shows that the MEF can guarantee feasibility if it is used appropriately. It implies that the feedback measures the volume of the set consisting of all sequences of actions that are feasible, conditional on the past actions taken by the agent. Therefore, as long as it is non-zero, the feasible set remains non-empty and there is always a feasible sequence of actions the agent can choose. The feedback is used as a penalty term in (17), therefore for any tuning parameter $\beta > 0$, the action $u_t \in \mathbb{R}^m$ selected by PPC always satisfies that $p_t^*(u_t | \mathbf{u}_{<t}) > 0$ and hence feasibility is guaranteed. Otherwise, suppose $p_t^*(u_t | \mathbf{u}_{<t}) = 0$ for the chosen action u_t , then the objective in (17) would blow up. This is summarized in the following corollary, which shows that the actions determined by PPC are feasible, subject to the constraints in (1).

COROLLARY 5.1. *For any predication window size $w \geq 1$, the sequence of actions $\mathbf{u} = (u_1, \dots, u_T)$ generated by the PPC in (17) always satisfies $\mathbf{u} \in S$.*

5.2 A fundamental limit

Before proceeding to the analysis of the dynamic regret of the PPC, we first consider a lower bound on $\text{Regret}(\mathbf{u})$, for any sequence of actions $\mathbf{u}$ generated by a deterministic online policy.

THEOREM 5.1 (FUNDAMENTAL LIMIT). *For any sequence of actions $\mathbf{u} \in S$ generated by a deterministic online policy that can access the safety sets $\{\mathcal{U}_t : t \in [T]\}$ and $\{\mathcal{X}_t : t \in [T]\}$ satisfying Assumption 2, for any $w \geq 1$, $\text{Regret}(\mathbf{u}) = \Omega(d(T - w))$ where $d := \text{diam}(\mathcal{U}) := \sup\{\|u - v\|_2 : u, v \in \mathcal{U}\}$ is the diameter of the action space $\mathcal{U}$, w is the prediction window size and T is the total number of time slots.*

The proof can found in Appendix This result highlights that additional assumptions are needed if one hopes to obtain a positive result. The next section considers such an assumption. Note that the proof shows that it suffices to have a memory size of one, i.e., the safety sets $\mathcal{U}_t$ and $\mathcal{X}_t$ only depend on the action u_{t-1} and state x_{t-1} .

5.3 Causally invariant safety constraints

Motivated by the lower bound in the previous section, we now introduce a particular class of safety constraints where it is possible to have better performance. The class is defined by a form of causal invariance that is intuitive and general. Specifically, we state a condition under which the sets of subsequent feasible actions do not change too much if the measures of the sets are close.

We define the following specific sequences of actions. Let $\bar{\mathbf{u}}_{\leq t} = (\bar{u}_1, \dots, \bar{u}_t)$ be a subsequence of optimal actions that maximizes the volume of the set of feasible actions, defined as $\bar{\mathbf{u}}_{\leq t} := \arg \sup_{\mathbf{u} \in \mathcal{U}^t} \mu(S(\mathbf{u}))$. With a slight abuse of notation, given $\mathbf{u}_{\leq t}$, define the length- k maximizing subsequence of actions as $\bar{\mathbf{u}}_{t+1:t+k} := \arg \sup_{\mathbf{u} \in \mathcal{U}^k} \mu(S_k(\mathbf{u}_{\leq t}, \mathbf{u}))$.

Definition 5.1 ((k, δ, λ) -causal invariance). *The safety sets are (k, δ, λ) -causally invariant if there exist constants $\delta, \lambda > 0$ such that the following holds:*

- (1) *For all $t \in [T]$ and sequences of actions $\mathbf{u}_{\leq t}$ and $\mathbf{v}_{\leq t}$,*

$$d_H(S_k(\mathbf{u}_{\leq t}), S_k(\mathbf{v}_{\leq t})) \leq \delta \left(\frac{|\mu(S(\mathbf{u}_{\leq t})) - \mu(S(\mathbf{v}_{\leq t}))|}{\mu(\mathcal{B})} \right)^{1/((T-t)m)} \quad (19)$$

where $\mathcal{B}$ denotes the unit ball in $\mathbb{R}^{m \times (T-t)}$.

- (2) *For all $t \in [T]$ and sequences of actions $\mathbf{u}_{\leq t}$,*

$$\frac{\mu(S(\mathbf{u}_{\leq t}))}{\mu(S(\bar{\mathbf{u}}_{\leq t}))} \leq \lambda \left(\frac{\mu(S(\mathbf{u}_{\leq t}, \bar{\mathbf{u}}_{t+1:t+k}))}{\mu(S(\bar{\mathbf{u}}_{\leq t+k}))} \right)^{\frac{T-t}{T-t-k}}. \quad (20)$$

Note that the definition of causal invariance is independent of the costs. Condition (1) is an inverse of the isodiametric inequality. It says that if the two subsequences of actions $\mathbf{u}_{\leq t}$ and $\mathbf{v}_{\leq t}$ do not affect the measure of the space of feasible sequences of actions too much, then the Hausdorff distance between the two sets $S_k(\mathbf{u}_{\leq t})$ and $S_k(\mathbf{v}_{\leq t})$ is also small. Condition (2) states that the ratio of the measure of a space of feasible sequences of actions, given any previous actions $\mathbf{u}_{\leq t}$, does not differ too much from the ratio of the measure of another space by fixing the next k actions to be those preserving the most future flexibility.

Definition 5.1 is general and practically applicable, as the following examples show.

EXAMPLE 1 (INVENTORY CONSTRAINTS). Consider the following set of inventory constraints, which is a common form of constraints in energy storage and demand response problems [30, 35, 45]. The summation of the squares of the ℓ_2 norms of the actions is bounded from above by $\gamma > 0$, representing the limited resources of the system: $\sum_{t=1}^T \|u_t\|_2^2 \leq \gamma$ where $u_t \in \mathbb{R}^m$. These inventory constraints are $(k, 1, 1)$ -causally invariant for all $1 \leq k \leq T$.

EXAMPLE 2 (TRACKING CONSTRAINTS). The following form of tracking constraints are common in situations where the system has a target "optimal" resource configuration that varies depending on the environment [21, 33, 40]. The agent seeks to track a fixed sequence of nominal actions $\mathbf{y} := (y_1, \dots, y_T)$. The constraints are $\sum_{t=1}^T |u_t - y_t|^p \leq \sigma$ with $p \geq 2$ where $u_t, y_t \in \mathbb{R}$ and $\sigma > 0$ represents the system's adjusting ability. These tracking constraints are $(k, \frac{2}{\sqrt{p}}, 1)$ -causally invariant for all $1 \leq k \leq T$.

Additionally, to highlight the structures that are *not* causally invariant, we return to the construction of constraints used in the proof of the lower bound in Theorem 5.1.

EXAMPLE 3 (CONSTRAINTS IN THE LOWER BOUND (THEOREM 5.1)). Consider the following set of constraints defined in the proof of Theorem 5.1 for obtaining a lower bound $\Omega(d(T - w))$ on the dynamic regret: $S := \{\mathbf{u} \in \mathcal{U}^T : u_t \in \mathcal{U}_t, \forall t \in [T]\}$ where

$$\mathcal{U}_t := \begin{cases} \mathcal{B}(a/2), & \text{if } \|u_{t-1} - v_{t-1}\|_2 \leq a \\ \mathcal{U} \setminus \mathcal{B}(a), & \text{if } \|u_{t-1} - v_{t-1}\|_2 > a \end{cases} \quad (21)$$

for some $a > 0$. Suppose the action space $\mathcal{U} \subseteq \mathbb{R}^m$ is closed and bounded. These constraints are not causally invariant for all $k = \Omega(T)$. The reason why the causal invariance criterion is violated for the safety constraints in (21) is that, in order to satisfy Condition (1) in (19), it is necessary to have $\delta = \Omega(k) = \Omega(T)$, which is not a constant.

Motivated by the example above, and mimicking the adversarial construction in Theorem 5.1, we can derive the following theorem and corollary, whose proofs are in Appendix B.4.

THEOREM 5.2 (LOWER BOUND ON Regret). If there exist a sequence of actions $\mathbf{v}_{\leq s} \in \mathcal{U}^s$, $1 \leq s \leq T - k$, $k \geq w$ with $k = \Omega(T)$ and some constant $\alpha > 0$ such that $d_H(S_k(\mathbf{u}_{\leq s}), S_k(\mathbf{v}_{\leq s})) \geq \alpha dk$ for any $\mathbf{u}_{\leq s} \in \mathcal{U}^s$, then for any sequence of actions $\mathbf{u} \in S$ generated by a deterministic online policy that can access the safety sets $\{\mathcal{U}_t : t \in [T]\}$ and $\{\mathcal{X}_t : t \in [T]\}$, $\text{Regret}(\mathbf{u}) = \Omega(d(T - w))$ where $\lambda(T)$ is a parameter that may depend on T ; d is the diameter of the action space $\mathcal{U}$; w is the prediction window size and T is the total number of time slots.

The theorem above states that, for any online policy knowing the safety sets $\{\mathcal{U}_t : t \in [T]\}$ and $\{\mathcal{X}_t : t \in [T]\}$ in advance, if there are two sets of length- k subsequences of actions $S_k(\mathbf{u}_{\leq t})$ and $S_k(\mathbf{v}_{\leq t})$ that are far from each other in terms of the Hausdorff distance, then a sub-linear regret is impossible. This highlights the necessity of the causal invariance condition. We make this explicit by further restricting the power of the online policy and assuming that it can only access the MEF $p_t, \dots, p_{t+k-1}$ at time t , which yields the following impossibility result as a corollary of Theorem 5.2.

COROLLARY 5.2 (LOWER BOUND ON Regret). *If there exist a sequence of actions $\mathbf{v}_{\leq t} \in \mathcal{U}^t$, $1 \leq s \leq T - k$, $k \geq w$ with $k = \Omega(T)$, a constant $\xi > 0$ and $\alpha = \Omega(T)$ such that*

$$\begin{aligned} \mu(S(\mathbf{u}_{\leq s})) &> \xi \mu(S(\bar{\mathbf{u}}_{\leq s})), \\ d_H(S_k(\mathbf{u}_{\leq s}), S_k(\mathbf{v}_{\leq s})) &\geq \alpha \left(\frac{|\mu(S(\mathbf{u}_{\leq s})) - \xi \mu(S(\mathbf{v}_{\leq s}))|}{\mu(\mathcal{B})} \right)^{1/((T-s)m)} \end{aligned}$$

for any $\mathbf{u}_{\leq s} \in \mathcal{U}^s$, then for any sequence of actions $\mathbf{u} \in \mathcal{S}$ generated by a deterministic online policy that can access the MEF $p_t, \dots, p_{t+w-1}$ at each time $t \in [T]$, $\text{Regret}(\mathbf{u}) = \Omega(d(T-w))$ where $\lambda(T)$ is a parameter that may depend on T ; d is the diameter of the action space $\mathcal{U}$; w is the prediction window size and T is the total number of time slots.

The corollary above indicates that it is necessary to have an upper bound on the Hausdorff distance between $S_k(\mathbf{u}_{\leq t})$ and $S_k(\mathbf{v}_{\leq t})$ (such as Condition (1) and (2) in the causal invariance criterion) in order to make the dynamic regret sub-linear. In the next section, we show that, however, if the causal invariance criterion holds such that both δ and λ are constants, then the dynamic regret can be made sub-linear with sufficiently many predictions, even for arbitrary Lipschitz continuous cost functions.

5.4 Bounding the dynamic regret of PPC

We are now ready to present our main result, which bounds the dynamic regret by a decreasing function of the prediction window size under the assumption that the safety sets are causally invariant.

THEOREM 5.3 (UPPER BOUND ON Regret). *Suppose the safety sets are (w, δ, λ) -causally invariant. The dynamic regret for the sequence of actions $\mathbf{u}$ given by PPC is bounded from above by*

$$\text{Regret}(\mathbf{u}) = O \left(dT \left(\frac{\delta \log \lambda}{\sqrt{w}} + \frac{\sqrt{\delta}}{w^{1/4}} \right) \right)$$

where d denotes the diameter of the action space $\mathcal{U}$, w is the prediction window size and T is the total number of time slots.

This theorem implies that, with additional assumptions on the safety constraints, a *sub-linear* (in T) dynamic regret is achievable, given a sufficiently large prediction window size $w = \omega(1)$ (in T). Notably, Theorem 5.1 implies that for the worst-case costs and constraints, a deterministic online controller that has full information of the constraints suffers from a linear regret. As a comparison, Theorem 5.3 shows that, under additional assumptions, even if only aggregated information is available, PPC achieves a sub-linear regret. This does not contradict to the lower bound on the dynamic regret in Theorem 5.1, since if there is no regulation assumptions on the safety sets, in the worst-case, the Hausdorff distance $d_H(S_w(\mathbf{u}_{\leq t}), S_w(\mathbf{v}_{\leq t})) = O(dw)$. This implies that a trivial upper bound of $O(dT)$ holds. Additionally, note that there is a trade-off between *flexibility* and *optimality* when selecting the tuning parameter $\beta > 0$. On the one hand, if the tuning parameter is too small, the algorithm is greedy and may suffer losses in the future; on the other hand, if the tuning parameter is too large, the algorithm is penalized by the feedback and therefore is far from being optimal.

A proof is presented in Appendix B.5. Briefly, Theorem 5.1 is proven by optimizing β in an upper bound $\text{Regret}(\mathbf{u}) = O(Td(\delta \log \lambda/w + d\delta\sqrt{w}/\beta + \beta/dw))$, which holds for any sequence of actions $\mathbf{u}$ given by the PCC with any tuning parameter $\beta > 0$.

6 LEARNING MAXIMUM ENTROPY FEEDBACK AND PREDICTIONS

In our analysis to this point, we assume the exact MEF is given at the local controller. However, for practical applications wherein the dynamics are non-stationary, computing the MEF exactly is computationally intensive. Thus, instead of computing it precisely, it is beneficial to learn the MEF using a data-driven approach. In this section, we discuss the use of model-free reinforcement learning (RL) to generate an *aggregator functions* ψ . To be more precise, the learned aggregator function $\psi : X \rightarrow P_{t:t+w-1}$ outputs an estimate of the MEF given the state x_t , where X is the state space and P is the set of all possible MEF $p_t, \dots, p_{t+w-1}$.

A wide variety of model-free deep reinforcement learning (RL) algorithms have been applied to a variety of tasks. In this section, we demonstrate that RL can be used to train a generator that outputs approximate MEF, given the state of the system. Note that the aggregator does not know the cost functions, so it cannot directly use an RL algorithm and transmit the learned Q-function or actor-critic model to the operator. It is also worth noting that a number of effective heuristics may be available such as a greedy approximation in [32] and other gradient-based or density estimation [44] methods. We leave to future work the question of finding an optimal approximation algorithm. In the sequel, we restrict our discussion to the case when the action space U is discrete.

Approximation of discrete MEF

Suppose U is discrete. To approximate the MEF sequences $p_t, \dots, p_{t+w-1}$ for the purpose of using PPC, we approximate a joint distribution $p_{t:t+w-1} = \prod_{\tau=t}^{t+w-1} p_\tau$ as a probability vector in $[0, 1]^{|U|^w}$ since $\log \prod_{\tau=t}^{t+w-1} p_\tau = \sum_{\tau=t}^{t+w-1} \log p_\tau$, which is the penalty term used in (17). For estimating MEF $p_{t:t+w-1}$ at each time $t \in [T]$, we use actor-critic architectures with separate policy and value function networks which enable the learning of a policy ψ that outputs $p_{t:t+w-1}$ given x_t on continuous RL action space X and state space $[0, 1]^{|U|^w}$.² Among the actor-critic algorithms [5], soft actor-critic (SAC) [20] is an off-policy maximum entropy deep RL algorithm that maximizes both the expected return and the expected entropy of the policy ψ :

$$J(\psi) = \sum_{t=1}^T \mathbb{E}_{(y_{t:t+w-1}, p_{t:t+w-1}) \sim \rho_\psi} [r(y_{t:t+w-1}, p_{t:t+w-1}) + \kappa \mathbb{H}(\psi(\cdot|x_t))] \quad (22)$$

where $y_{t:t+w-1} := (x_t, \dots, x_{t+w-1}, u_t, \dots, u_{t+w-1})$; $p_{t:t+w-1} := (p_t, \dots, p_{t+w-1})$; $\kappa > 0$ and ρ_ψ denotes the state-action marginals of the trajectory distribution induced by a policy ψ . To estimate MEF, we need to determine a reward function $r(y_{t:t+w-1}, p_{t:t+w-1})$ in (22) that considers not only the current state-action pair, but also the states and actions in the future $w - 1$ steps. We adopt the following reward function that incorporates the constraints and the definition of MEF:

$$\begin{aligned} r(y_{t:t+w-1}, p_{t:t+w-1}) &= \mathbb{H}(p_{t:t+w-1}) + \sigma g(y_{t:t+w-1}; X_t, \dots, X_{t+w-1}, U_t, \dots, U_{t+w-1}) \\ &= \sum_{\tau=t}^{t+w-1} \mathbb{H}(p_\tau) + \sigma g(y_{t:t+w-1}; X_t, \dots, X_{t+w-1}, U_t, \dots, U_{t+w-1}) \end{aligned} \quad (23)$$

where the first term maximizes the entropy of the probability distribution p_t , based on the definition of the MEF in Definition 3.1 and $g(y_{t:t+w-1}) = g(x_t, \dots, x_{t+w-1}, u_t, \dots, u_{t+w-1})$ is a function that rewards the state and action if they satisfy the constraints $x_\tau \in X_\tau$ and $u_\tau \in U_\tau$ for any $\tau = t, \dots, t + w - 1$. The reward function is independent of the price functions. A concrete example of $g(y_{t:t+w-1})$ is given in the next section using an EV application and a typical training curve of the reward is shown in Figure 3 (shadow region represents variance). For RL algorithms, because of

²Note that the RL state space (consisting of x_t) and action space (consisting of $p_{t:t+w-1}$) referred here are the standard definitions in the context of RL and they are different from the “action space” U and “state space” X defined in Section 2.

Fig. 2. Example of a local controller that charges EVs and a central controller that operates a power grid.

scalability issues, when the prediction window size w increases, the dimension of the RL action space for $\mathbf{p}_{t:t+w-1}$ and RL state space for $\mathbf{y}_{t:t+w-1}$ grows exponentially in w . Thus, there is a trade-off between the scalability of determining the feedback from predictions accurately and the benefit that predictions provide using RL algorithms. In the next section, we demonstrate that $w = 4$ is a reasonable choice for RL algorithms. Recall that Theorem 5.3 indicates that when $w = \omega(1)$, a sub-linear regret is possible, which provides motivation for longer prediction windows. Therefore, improving the current learning model to reduce the dimension of the action and state space is an important future direction.

7 CASE STUDY

In this section, we demonstrate the use of PPC in a practical setting. Specifically, we study the EV charging application described in Section 2.2. The steps in the closed-loop coordination between a system operator and an EV charging aggregator are summarized in Algorithm 3, which is a concrete realization of the closed-loop control framework presented in Algorithm 2.

To understand why the use of aggregate information is crucial in this case, note that the dimension of the system states grows linearly with the number of chargers at time t . For a system operator that controls a large-scale EV charging station, solving the optimization in (9) requires transmitting and processing a huge amount of data including the high-dimensional states and constraints. In practice, the aggregate power rate at the substation is chosen from a small discrete set of power levels. Therefore, for the online control of an EV charging system shown in Figure 2, it is desirable to aggregate the constraints using the MEF defined in Definition 3.1 to both reduce the amount of data to be transmitted and prevent the leakage of exact constraints by aggregating the constraints.

7.1 Setting

In our experiments, we use ACN-Data [29], which is a dataset collected from adaptive EV charging networks (ACNs). The dataset represents a real-world charging system that operates 52 EV chargers. The notation used in this section, if not defined, is introduced and explained comprehensively in an example in Section 2.2 and Table 1 in the supplementary materials.

States. The system state x_t is a 104-dimensional vector where a state is $x_t = \{(e_t(j), d_t(j)) : j = 1, \dots, 52\}$, i.e., the remaining energy to be charged and the remaining charging time if it is being used; otherwise the vector is an all-zero vector.

Actions. The actions represent the power levels for each charger. We suppose that there are 10 operating power levels for the first two experiments and 3 for the last test. The values range between 0 kWh and 150 kWh, uniformly.

Fig. 3. Average rewards (defined in (24)) in the training stage with tuning parameter $\beta = 6000$.

Feedback. The feedback, i.e., MEF defined in Definition 3.1, is a length-10 (or length-3) probability vector, whose dimension is much less than the dimension of x_t , which can be as large as 104. The space of all MEF functions is the 10-dimensional (or 3-dimensional) probability simplex.

Information aggregation. We adopt soft actor-critic (SAC) [20], an off-policy maximum entropy deep reinforcement learning algorithm to learn the MEF. The policy in our experiments is fixed to be a parameterized family of Gaussian distributions. We train an agent $\psi_{\text{SAC}} : \mathbb{R}_+^{104} \rightarrow \mathcal{P}$ using SAC whose input at time $t \in [T]$ is a state x_t that encodes the remaining energy to be delivered and the remaining charging time for the EV being charged at each time $t \in [T]$. We use the following reward function, as a special case of the general reward function in (23) for our EV charging scenario to penalize the violations of constraints and maximize the entropy of $\mathbf{p}_{t:t+w-1} = (p_t, \dots, p_{t+w-1})$ as an empirical sub-optimal solution for (11a)-(11b):

$$r_{\text{EV}}(\mathbf{y}_{t:t+w-1}, \mathbf{p}_{t:t+w-1}) = \sum_{\tau=t}^{t+w-1} (\mathbb{H}(p_\tau) + \sigma_1 \|\phi_\tau(x_\tau, u_\tau)\|_2) - \sigma_2 \mathbb{I}(t < d_t(j) \leq t+1) \left[e(j) - e_t(j) \right]_+ - \sigma_3 \left| \pi_{\text{PPC}}(\mathbf{p}_{t:t+w-1}) - \sum_{j=1}^n \phi_t(j) \right| \quad (24)$$

where $\mathbb{I}(\cdot)$ denotes the indicator function and σ_1, σ_2 and σ_3 are positive constants. The second term is to enhance charging performance. With the settings described above, in Figure 3 we show a typical training curve of the reward function in (24).

Scheduling. Note that we do not directly learn the disaggregation of u_t , which would have too large an action space. Instead, we fix a specific scheduling algorithm ϕ and learn the feedback vectors directly. Being able to work consistently with existing scheduling policies is another benefit of the MEF. In our experiments we adopt Least Laxity First, but the results are consistent with other algorithms.

Online control algorithm. We implement the PPC scheme described in Section 4 as our online control algorithm. We test different choices of tuning parameters β in Section 7.2. The hyper-parameters used in our experiments are summarized in Table 1 and the closed-loop control framework is presented in Algorithm 3.

7.2 Results

In the following, we show our experimental results for online cost minimization, in the context of the ACN-Data [29]. We first introduce two error metrics to measure the constraint violations. To better approximate real-world cases, we consider an additional *operational constraints* for the operator (central controller) and require that $u_t \leq \gamma$ (kWh) for every $t \in [T]$. The total number of tests is L and the total number of charging sessions is N . To measure the error induced by undelivered energy (corresponding to constraint (6)), we define the following mean percentage error (MPE):

$$\text{MPE} := \sum_{k=1}^L \sum_{t=1}^T \sum_{j=1}^N \phi_t^{(k)}(j) / \left((L \times T) \cdot \sum_{j=1}^N e(j) \right) \times \%, \quad (25)$$

where $e(j)$ is the energy request for each load $j \in [N]$ and $\phi_t^{(k)}(j)$ is the energy scheduled to the j -th charging session at time t for the k -th test.

Experiment A. Trade-offs of MPE and Costs. First, in Figure 4 we show the changes of the cumulative costs by varying the MPE (defined (25)). We use $w = 1$ for this experiment. There are $K = 14$ episodes tested for days selected from Dec. 2, 2019 to Jan. 1, 2020 (days with less than 30 charging sessions are removed, i.e. we require, $N \geq 30$). Note that $0 \leq \text{MPE} \leq 1$. We allow constraint violations and relax the total energy constraint according to fixed MPE values. For the PPC, we vary the tuning parameter β to obtain the corresponding costs and MPE. For the MPC in our tests, we solve the following optimization at each time for obtaining the charging decisions $s_t = (s_t(1), \dots, s_t(n'))$:

$$\begin{aligned} s_t = \arg \min_{s_t} \quad & \sum_{\tau=t}^{t'} c_\tau \left(\sum_{i=1}^{n'} s_\tau(i) \right) \text{ subject to :} \\ & s_\tau(i) = 0, \quad \tau < a(i), \quad i = 1, \dots, n', \\ & s_\tau(i) = 0, \quad \tau > d(i), \quad i = 1, \dots, n', \\ & \sum_{i=1}^{n'} s_\tau(i) = u_t, \quad \tau = t, \dots, t', \\ & \sum_{\tau=1}^T s_\tau(i) = \zeta \cdot e(i), \quad i = 1, \dots, n', \\ & 0 \leq s_\tau(i) \leq r(i), \quad \tau = t, \dots, t' \end{aligned} \quad (26)$$

where at time t , the integer n' denotes the number of EVs being charged at the charging station and the time horizon of the online optimization is from $\tau = t$ to t' , which is the latest departure time of the present charging sessions; $a(i)$ and $d(i)$ are the arrival time and departure time of the i -th session; $\zeta > 0$ relaxes the energy demand constraints and therefore changes the MPE region for MPC. The offline cost-energy curve is obtained by varying the energy demand constraints in (5) in a similar way. We assume there is no admission control and an arriving EV will take a charger whenever it is idle for both MPC and PPC. Note that this MPC framework in (26) is widely studied [41] and used in EV charging applications [28]. It requires the precise knowledge of a 104-dimensional state vector of 52 chargers at each time step. We observe that with *only* feasibility information, PPC outperforms MPC for all $0 \leq \text{MPE} \leq 1$.

The results in the figure show a trend that, as β grows larger, the operator running the PPC scheme favors a signal that preserves future feasibility; therefore improving the charging performance (more energy is charged). This is reflected in the decreases of the mean percentage error $\text{MPE}(\phi, \mathbf{x})$

Fig. 4. Cost-energy curves for the offline optimization in (2a)–(2b), MPC and PPC (introduced in Section 4).

Fig. 5. Pilot control signals and real energy allocated to EVs controlled by PPC for selected days (with no less than 30 charging sessions, i.e., $N \geq 30$) from Dec. 2, 2019 to Jan. 1, 2020.

for charging. As a trade-off, the total costs increase. The costs for $\beta = 1 \times 10^3$, 2×10^3 and 4×10^3 are smaller than the offline costs because the EVs' energy demands are not fully satisfied as in the offline case (equivalently, MPE is positive).

Experiment B. Charging curves & performance. In Figure 5, pilot control and real energy signals are shown. The agent is trained on data collected from Nov. 1, 2018 to Dec. 1, 2019 with linear price functions $c_t = 1 - t/24$, where $t \in [0, 24]$ is the time index and tested on Dec. 18, 2019 with average LMPs on the CAISO day-ahead market in 2016, shown on the bottom. We use a tuning parameter $\beta = 6 \times 10^3$. The figure highlights that, with a suitable choice of the tuning parameter, the operator is able to schedule charging at time slots where prices are lower and avoid charging at

Fig. 6. Cumulative costs for different prediction window sizes: $w = 1$ to 4.

the peak of prices, as desired, for most of the tested days. (For the curve shown in the bottom center, the operator does not shift the energy demands to avoid peak prices, because there is not enough flexibility to do so.) The true substation energy (real charging signal) also accurately matches the actions given by the operator (pilot control signal). To make our experiments closer to real-world cases, we consider an additional *operational constraints* for the operator (central controller) and require that $u_t \leq 150$ (kWh) for every $t \in [T]$ and the learned MEF is able to automatically flatten the charging curve within this range, without explicitly knowing it.

Experiment C. Cumulative costs & predictions. Finally, in Figure 6 we show that the cumulative costs decrease as the prediction window size w increases, up to a point. Because it is challenging to obtain the exact MEF when predictions are used, we consider the MEF defined over the action space U^w and use SAC to learn this MEF with estimated predictions. To avoid an overly large action space, we consider a smaller set of power levels that has two distinct choices spaced equally in $[0, 150]$ (kWh). The training and testing datasets are the same as Experiment A. and B. We use a tuning parameter $\beta = 8 \times 10^3$.

The results highlight a decrease in the cumulative costs as the prediction window size w increases from 1 to 4. It is worth noting that when the prediction window size goes beyond 4, because of scalability issues, the reward given by SAC does not converge very well in the training stage. This results in inaccurate aggregate feedback and significantly worse performance. This exemplifies an important trade-off between the scalability of determining the feedback from predictions accurately and the benefit that predictions provide using RL algorithms. In this example, $w = 4$ is the sweet spot. However, as more scalable approaches for RL are developed in this context, more predictions can be included, which our theory indicates will further reduce costs.

8 CONCLUDING REMARKS AND FUTURE DIRECTIONS

This paper studies and analyzes the closed-loop control framework created by the interaction between a central controller and a local controller. Our analysis shows that it is possible to combine model predictive control with limited aggregate feedback about feasibility to, under certain model assumptions, achieve a sub-linear dynamic regret. We also illustrate the application of the proposed scheme to an EV charging application.

Though we focus on the application of EV charging, the results in this paper are applicable well beyond EV charging. For example, a similar scheme may be used in cloud computing and data center scheduling [12, 49] to make the network more sustainable. Further, the same design applies to the networks formed by other distributed energy resources (DERs) such as HVAC (heating,

ventilation, and air conditioning) systems, rooftop solar PV units, energy storage systems and inverters. It will be interesting for future work to explore these applications.

On the analytic side, this paper presents the first analysis of a two-controller system where a local controller governs a complicated time-varying and coupling dynamical system and a central controller minimizes costs. To achieve our analytic guarantees, we make several modeling assumptions and the relaxation of these assumptions is an important task for future work. First, our penalized predictive control scheme is designed for 1-1 coordination between two controllers. It would be interesting to extend the control scheme to allow for the interaction between a central controller and multiple local participants. Second, our analysis assumes perfect maximum entropy feedback (MEF) but we see empirically that approximations of the MEF can still guarantee feasibility. Generalizing our current results to approximations of MEF is an challenging and important direction. Third, we have used reinforcement learning to approximate the MEF in this paper, but the approach is general and it is important to explore additional deep learning techniques for the estimation of the MEF.

ACKNOWLEDGMENTS

Tongxin Li and Steven Low acknowledge the support received from National Science Foundation (NSF) through grants CCF 1637598, ECCS 1931662 and CPS ECCS 1932611. Bo Sun is supported by Hong Kong Research Grant Council (RGC) General Research Fund (Project 16207318). Adam Wierman's research is funded by NSF (AitF-1637598 and CNS-1518941), PIMCO, and Amazon AWS.

REFERENCES

- [1] Naman Agarwal, Brian Bullins, Elad Hazan, Sham Kakade, and Karan Singh. Online control with adversarial disturbances. In *International Conference on Machine Learning*, pages 111–119, 2019.
- [2] Naman Agarwal, Elad Hazan, and Karan Singh. Logarithmic regret for online control. In *Advances in Neural Information Processing Systems*, pages 10175–10184, 2019.
- [3] Oren Anava, Elad Hazan, and Shie Mannor. Online learning for adversaries with memory: price of past mistakes. In *Advances in Neural Information Processing Systems*, pages 784–792, 2015.
- [4] Masoud Badieli, Na Li, and Adam Wierman. Online convex optimization with ramp constraints. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 6730–6736. IEEE, 2015.
- [5] Andrew G Barto, Richard S Sutton, and Charles W Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, (5):834–846, 1983.
- [6] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in neural information processing systems*, pages 908–918, 2017.
- [7] Andrey Bernstein, Emiliano Dall’Anese, and Andrea Simonetto. Online primal-dual methods with measurement feedback for time-varying convex optimization. *IEEE Transactions on Signal Processing*, 67(8):1978–1991, 2019.
- [8] Aditya Bhaskara, Ashok Cutkosky, Ravi Kumar, and Manish Purohit. Online learning with imperfect hints. *arXiv preprint arXiv:2002.04726*, 2020.
- [9] Xuanyu Cao and KJ Ray Liu. Online convex optimization with time-varying constraints and bandit feedback. *IEEE Transactions on Automatic Control*, 64(7):2665–2680, 2018.
- [10] Niangjun Chen, Anish Agarwal, Adam Wierman, Siddharth Barman, and Lachlan LH Andrew. Online convex optimization using predictions. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 191–204, 2015.
- [11] Niangjun Chen, Joshua Comden, Zhenhua Liu, Anshul Gandhi, and Adam Wierman. Using predictions in online optimization: Looking forward with an eye on the past. *ACM SIGMETRICS Performance Evaluation Review*, 44(1):193–206, 2016.
- [12] Tianyi Chen and Georgios B Giannakis. Bandit convex optimization for scalable and dynamic iot management. *IEEE Internet of Things Journal*, 6(1):1276–1286, 2018.
- [13] Tianyi Chen, Qing Ling, and Georgios B Giannakis. An online convex optimization approach to proactive network resource allocation. *IEEE Transactions on Signal Processing*, 65(24):6350–6364, 2017.
- [14] Alon Cohen, Avinatan Hassidim, Tomer Koren, Nevena Lazic, Yishay Mansour, and Kunal Talwar. Online linear quadratic control. *arXiv preprint arXiv:1806.07104*, 2018.
- [15] Alon Cohen, Tomer Koren, and Yishay Mansour. Learning linear-quadratic regulators efficiently with only $\sqrt{T}$ regret. In *International Conference on Machine Learning*, pages 1300–1309, 2019.
- [16] Sarah Dean, Stephen Tu, Nikolai Matni, and Benjamin Recht. Safely learning to control the constrained linear quadratic regulator. In *2019 American Control Conference (ACC)*, pages 5582–5588. IEEE, 2019.
- [17] Gautam Goel and Adam Wierman. An online algorithm for smoothed regression and lqr control. *Proceedings of Machine Learning Research*, 89:2504–2513, 2019.
- [18] Lars Grüne and Simon Pirkelmann. Economic model predictive control for time-varying system: Performance and stability results. *Optimal Control Applications and Methods*, 41(1):42–64, 2020.
- [19] Linqi Guo, Karl F Erliksson, and Steven H Low. Optimal online adaptive electric vehicle charging. In *2017 IEEE Power & Energy Society General Meeting*, pages 1–5. IEEE, 2017.
- [20] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870, 2018.
- [21] Eric Hall and Rebecca Willett. Dynamical models and tracking regret in online convex programming. In *International Conference on Machine Learning*, pages 579–587, 2013.
- [22] Eric C Hall and Rebecca M Willett. Online convex optimization in dynamic environments. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):647–662, 2015.
- [23] Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.
- [24] Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization: Competing with dynamic comparators. In *Artificial Intelligence and Statistics*, pages 398–406, 2015.
- [25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [26] Alec Koppel, Felicia Y Jakubiec, and Alejandro Ribeiro. A saddle point algorithm for networked online convex optimization. *IEEE Transactions on Signal Processing*, 63(19):5149–5164, 2015.
- [27] Alec Koppel, Brian M Sadler, and Alejandro Ribeiro. Proximity without consensus in online multiagent optimization. *IEEE Transactions on Signal Processing*, 65(12):3062–3077, 2017.

- [28] Zachary J Lee, Daniel Chang, Cheng Jin, George S Lee, Rand Lee, Ted Lee, and Steven H Low. Large-scale adaptive electric vehicle charging. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–7. IEEE, 2018.
- [29] Zachary J Lee, Tongxin Li, and Steven H Low. Acn-data: Analysis and applications of an open ev charging dataset. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, pages 139–149, 2019.
- [30] Yanzhe Murray Lei, Stefanus Jasin, and Amitabh Sinha. Near-optimal bisection search for nonparametric dynamic pricing with inventory constraint. *Ross School of Business Paper*, (1252), 2014.
- [31] Antoine Lesage-Landry, Iman Shames, and Joshua A Taylor. Predictive online convex optimization. *Automatica*, 113:108771, 2020.
- [32] Tongxin Li, Steven H Low, and Adam Wierman. Real-time flexibility feedback for closed-loop aggregator and system operator coordination. In *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, pages 279–292, 2020.
- [33] Yingying Li, Guannan Qu, and Na Li. Using predictions in online optimization with switching costs: A fast algorithm and a fundamental limit. In *2018 Annual American Control Conference (ACC)*, pages 3008–3013. IEEE, 2018.
- [34] Minghong Lin, Zhenhua Liu, Adam Wierman, and Lachlan LH Andrew. Online algorithms for geographical load balancing. In *2012 international green computing conference (IGCC)*, pages 1–10. IEEE, 2012.
- [35] Qiulin Lin, Hanling Yi, John Pang, Minghua Chen, Adam Wierman, Michael Honig, and Yuanzhang Xiao. Competitive online optimization under inventory constraints. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(1):1–28, 2019.
- [36] Yiheng Lin, Gautam Goel, and Adam Wierman. Online optimization with predictions and non-convex losses. *arXiv preprint arXiv:1911.03827*, 2019.
- [37] Shie Mannor, John N Tsitsiklis, and Jia Yuan Yu. Online learning with sample path constraints. *Journal of Machine Learning Research*, 10(Mar):569–590, 2009.
- [38] Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 7195–7201. IEEE, 2016.
- [39] Miguel A Ortega-Vazquez, François Bouffard, and Vera Silva. Electric vehicle aggregator/system operator coordination for charging scheduling and services procurement. *IEEE Transactions on Power Systems*, 28(2):1806–1815, 2012.
- [40] Romer Rosales and Stan Sclaroff. Improved tracking of multiple humans with trajectory prediction and occlusion modeling. Technical report, Boston University Computer Science Department, 1998.
- [41] Ugo Rosolia, Xiaojing Zhang, and Francesco Borrelli. Data-driven predictive control for autonomous systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:259–286, 2018.
- [42] Guanya Shi, Yiheng Lin, Soon-Jo Chung, Yisong Yue, and Adam Wierman. Beyond no-regret: Competitive control via online optimization with memory. *arXiv preprint arXiv:2002.05318*, 2020.
- [43] Ming Shi, Xiaojun Lin, Sonia Fahmy, and Dong-Hoon Shin. Competitive online convex optimization with switching costs and ramp constraints. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 1835–1843. IEEE, 2018.
- [44] Shashank Singh, Ananya Uppal, Boyue Li, Chun-Liang Li, Manzil Zaheer, and Barnabás Póczos. Nonparametric density estimation under adversarial losses. In *Advances in Neural Information Processing Systems*, pages 10225–10236, 2018.
- [45] Bo Sun, Ali Zeynali, Tongxin Li, Mohammad Hajiesmaili, Adam Wierman, and Danny HK Tsang. Competitive algorithms for the online multiple knapsack problem with application to electric vehicle charging. *arXiv preprint arXiv:2010.00412*, 2020.
- [46] Wen Sun, Debadeepta Dey, and Ashish Kapoor. Safety-aware algorithms for adversarial contextual bandit. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3280–3288. JMLR. org, 2017.
- [47] Xianfu Wang. Volumes of generalized unit balls. *Mathematics Magazine*, 78(5):390–395, 2005.
- [48] Xinlei Yi, Xiuxian Li, Lihua Xie, and Karl H Johansson. Distributed online convex optimization with time-varying coupled inequality constraints. *IEEE Transactions on Signal Processing*, 68:731–746, 2020.
- [49] Hao Yu, Michael Neely, and Xiaohan Wei. Online convex optimization with stochastic constraints. In *Advances in Neural Information Processing Systems*, pages 1428–1438, 2017.
- [50] Jianjun Yuan and Andrew Lamperski. Online convex optimization for cumulative constraints. In *Advances in Neural Information Processing Systems*, pages 6137–6146, 2018.
- [51] Lijun Zhang, Tianbao Yang, Zhi-Hua Zhou, et al. Dynamic regret of strongly adaptive methods. In *International Conference on Machine Learning*, pages 5882–5891, 2018.
- [52] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.

APPENDIX

Table 1. Hyper-parameters in the experiments.

Parameter	Value
<i>Central Controller (System Operator)</i>	
Number of power levels $ U $	10
Cost functions $c_1, \dots, c_T$	Average LMPs
Operator function π	PPC (this work)
Tuning parameter β	$1 \times 10^3 - 1 \times 10^6$
<i>Local Controller (EV Charging Aggregator)</i>	
Number of Chargers W	52
State space X	$\mathbb{R}_+^{104}$
Action space	$[0, 1]^{10}$
Time interval Δ	12 minutes
Private vector $(a(j), d(j), e(j), r(j))$	ACN-Data [29]
Power rating	150 kW
Scheduling algorithm ϕ	LLF
Laxity	$d_t(j) - e_t(j)/r(j)$
RL algorithm	SAC [20]
Optimizer	Adam [25]
Learning rate	$3 \cdot 10^{-4}$
Discount factor	0.5
Relay buffer size	10^6
Number of hidden layers	2
Number of hidden units per layer	256
Number of samples per minibatch	256
Non-linearity	ReLU
Reward function	$\sigma_1 = 0.1, \sigma_2 = 0.2, \sigma_3 = 2$
Temperature parameter	0.5

EV charging Experiments in Section 7

Algorithm 3: Closed-loop coordination between an operator and an EV charging aggregator in Section 7.2 (setting prediction window size $w = 1$ for simplicity).

Data: Electricity prices $c_1, \dots, c_T$.

Total number of time slots, T .

Total number of charging sessions, N .

Total number of EV chargers, W .

Sequential charging session data $\{a(j), d(j), e(j), r(j)) : j \in [N]\}$.

Result: Total electricity cost $C_T = \sum_{t=1}^T c_t u_t$.

Initialize: Set of occupied EV chargers $\mathcal{V} = \emptyset$. States $x_0^{(i)} = (0, 0)$, $i \in [W]$.

for $t \in [T]$ **do**

System operator

Generate substation power (action u_t):

$$u_t = \pi_{\text{PPC}}(c_t, p_t)$$

Update electricity cost:

$$C_t = C_{t-1} + c_t u_t$$

EV aggregator

Update EV chargers (state x_t) for new arrivals:

for $j \in [N]$ **do**

 Check if there is a vacancy:

if $t < a(j) \leq t + \Delta$ and $\exists i \in [W] \setminus \mathcal{V}$ **then**

$$x_t^{(i)} = (e(j), d(j))$$

$$\mathcal{V} \leftarrow \mathcal{V} \cup \{i\}$$

end

end

Schedule substation power to chargers: $\phi_t = \phi(x_t, u_t)$

Update EV chargers (state x_t):

for $i \in \mathcal{V}$ **do**

 Check if a session ends:

if $t < x_t^{(i)}(2) \leq t + \Delta$ **then**

$$x_{t+1}^{(i)} = (0, 0)$$

$$\mathcal{V} \leftarrow \mathcal{V} \setminus \{i\}$$

end

else

$$x_{t+1}^{(i)} = x_t^{(i)} - (\phi_t(i), \Delta)$$

end

end

Generate estimated MEF:

$$p_{t+1} = \psi_{\text{SAC}}(x_{t+1})$$

end

return Total cost C_T

A EXPLANATION OF DEFINITION 5.1 AND TWO EXAMPLES

Fig. 7. Graphical illustration of (19) in Definition 5.1.

Example 1: Inventory constraints.

Recall that the feasible set of actions for the inventory constraints is

$$S = \left\{ \mathbf{u} \in \mathbb{R}^{T \times m} : \sum_{t=1}^T \|\mathbf{u}_t\|_2^2 \leq \gamma \right\}.$$

The sequence of actions $\bar{\mathbf{u}}_{\leq t}$ maximizing the size of the set of admissible actions is the all-zero vector. Hence,

$$\left(\frac{\mu(S(\mathbf{u}_{\leq t}))}{\mu(S(\bar{\mathbf{u}}_{\leq t}))} \right)^{\frac{1}{T-t}} = \left(1 - \frac{\sum_{\tau=1}^t \|\mathbf{u}_\tau\|_2^2}{\gamma} \right)^{\frac{m}{2}} = \left(\frac{\mu(S((\mathbf{u}_{\leq t}, \bar{\mathbf{u}}_{t+1:t+k})))}{\mu(S(\bar{\mathbf{u}}_{\leq t+k}))} \right)^{\frac{1}{T-t-k}}.$$

Therefore setting $\lambda = 1$, (20) is satisfied. It remains to check the bound on the Hausdorff distance. Figure 7 shows the idea behind the definition. If the two sets $S(\mathbf{v}_{\leq t})$ and $S(\mathbf{u}_{\leq t})$ are close to each other, the Hausdorff distance of the projected sub-spaces $S_k(\mathbf{u}_{\leq t})$ and $S_k(\mathbf{v}_{\leq t})$ can also be bounded. For inventory constraints, this is indeed the case. For all $1 \leq k \leq T$, sequences of actions $\mathbf{u}_{\leq t}$ and $\mathbf{v}_{\leq t}$,

$$d_H(S_k(\mathbf{u}_{\leq t}), S_k(\mathbf{v}_{\leq t})) = \left(\left| \sum_{\tau=1}^t \|\mathbf{u}_\tau\|_2^2 - \sum_{\tau=1}^t \|\mathbf{v}_\tau\|_2^2 \right| \right)^{1/2} \quad (\text{A.1})$$

and

$$\begin{aligned} \mu(\mathcal{B})^{\frac{2}{(T-t)m}} \left| \sum_{\tau=1}^t \|\mathbf{u}_\tau\|_2^2 - \sum_{\tau=1}^t \|\mathbf{v}_\tau\|_2^2 \right| &= \left| \mu(S(\mathbf{v}_{\leq t}))^{\frac{2}{(T-t)m}} - \mu(S(\mathbf{u}_{\leq t}))^{\frac{2}{(T-t)m}} \right| \\ &\leq |\mu(S(\mathbf{v}_{\leq t})) - \mu(S(\mathbf{u}_{\leq t}))|^{\frac{2}{(T-t)m}}. \end{aligned} \quad (\text{A.2})$$

Therefore setting $\delta = 1$, (A.1) and (A.2) imply (19). We validate that the inventory constraints in Example 1 are $(k, 1, 1)$ -causally invariant for all $1 \leq k \leq T$.

Example 2: Tracking constraints.

Recall that the feasible set of actions for the tracking constraints is (with $p \geq 2$):

$$S = \{ \mathbf{u} \in \mathbb{R}^T : \|\mathbf{u} - \mathbf{y}\|_p^p \leq \sigma \}.$$

The sequence of actions maximizing the size of the set of admissible actions is $\bar{\mathbf{u}}_{\leq t} = \mathbf{y}_{\leq t}$. Similar to Example 1, according to the formula of the volume of an ℓ_p -ball [47], we have

$$\left(\frac{\mu(S(\mathbf{u}_{\leq t}))}{\mu(S(\bar{\mathbf{u}}_{\leq t}))} \right)^{\frac{1}{T-t}} = \left(1 - \frac{\|\mathbf{u}_{\leq t} - \mathbf{y}_{\leq t}\|_p^p}{\sigma} \right)^{1/p} = \left(\frac{\mu(S((\mathbf{u}_{\leq t}, \bar{\mathbf{u}}_{t+1:t+k})))}{\mu(S(\bar{\mathbf{u}}_{\leq t+k}))} \right)^{\frac{1}{T-t-k}}.$$

Therefore setting $\lambda = 1$, (20) is satisfied. Next, we give a bound on the Hausdorff distance. For all $1 \leq k \leq T$, sequences of actions $\mathbf{u}_{\leq t}$ and $\mathbf{v}_{\leq t}$,

$$\begin{aligned} & \left(\frac{(2\Gamma(1+1/p))^{T-t}}{\Gamma((T-t)/p+1)} \right)^{\frac{1}{T-t}} d_H(S_k(\mathbf{u}_{\leq t}), S_k(\mathbf{v}_{\leq t})) \\ & \leq \frac{2\Gamma(1+1/p)}{\sqrt{\pi}} \left(\frac{\pi^{(T-t)/2}}{\Gamma((T-t)/2+1)} \right)^{\frac{1}{T-t}} \left| \sum_{\tau=1}^t \|u_\tau - y_\tau\|_p - \sum_{\tau=1}^t \|v_\tau - y_\tau\|_p \right| \\ & = \frac{2\Gamma(1+1/p)}{\sqrt{\pi}} \mu(\mathcal{B})^{\frac{1}{T-t}} \left| \sum_{\tau=1}^t \|u_\tau - y_\tau\|_p - \sum_{\tau=1}^t \|v_\tau - y_\tau\|_p \right| \\ & \leq \frac{2\Gamma(1+1/p)}{\sqrt{\pi}} |\mu(S(\mathbf{v}_{\leq t})) - \mu(S(\mathbf{u}_{\leq t}))|^{\frac{1}{T-t}} \end{aligned}$$

where $\Gamma(\cdot)$ is Euler's gamma function. Therefore setting $\delta = \frac{2\Gamma(1+1/p)}{\sqrt{\pi}} \leq \frac{2}{\sqrt{\pi}}$ for all $p \geq 2$, (19) holds.

The tracking constraints in Example 2 are $(k, 2/\sqrt{\pi}, 1)$ -causally invariant for all $1 \leq k \leq T$.

Example 3: Constraints in the Proof of Theorem 5.1.

Denote by $A_1 := \mathcal{B}(a/2)$ and $A_2 := \mathcal{U} \setminus \mathcal{B}(a/2)$. The feasible length- k sub-sequences of actions are either in the Cartesian product of sets $A_1^k := A_1 \times \cdots \times A_1$ or $A_2^k := A_2 \times \cdots \times A_2$. If the two sequences $\mathbf{u}_{\leq t}$ and $\mathbf{v}_{\leq t}$ are in the same set, then $\delta = \lambda = 1$; otherwise, the RHS of (20) becomes a constant term. Assuming $t+k \leq T$, the Hausdorff distance between $S_k(\mathbf{u}_{\leq t}) = A_1^k$ and $S_k(\mathbf{v}_{\leq t}) = A_2^k$ is $\Omega(k)$. Therefore, a non-scalar parameter $\delta = \Omega(k)$ is necessary for (20) to hold.

B PROOFS

B.1 Proof of Lemma 1

We prove the statement by induction. It is straightforward to verify the results hold when $T = 1$. We suppose the lemma is true when $T = m$. Suppose $T = m + 1$. Let

$$F(u) := \max_{p_2, \dots, p_T} \sum_{t=2}^T \mathbb{H}(U_t | \mathbf{U}_{2:t-1}; U_1 = u)$$

denote the optimal value corresponding to the time horizon $t \in [T]$, given the first action $U_1 = u$. By the definition of conditional entropy, we have

$$F = \max_{p_1} \int_{u \in \mathcal{U}} p_1(u) F(u) du + \mathbb{H}(p_1).$$

By the induction hypothesis, $F(u) = \mu(S(u))$. Therefore,

$$\begin{aligned} F &= \max_{p_1} \int_{u \in \mathcal{U}} p_1(u) \log \mu(S(u)) du + \mathbb{H}(p_1) \\ &= \max_{p_1} \int_{u \in \mathcal{U}} p_1(u) \log \left(\frac{\mu(S(u))}{p_1(u)} \right) du \end{aligned}$$

whose optimizer p_1^* satisfies (12) and we get $F = \mu(S)$. The lemma follows by finding the optimal conditional distributions $p_1^*, \dots, p_T^*$ inductively.

B.2 Proofs of Proposition 1 and Corollary 5.1

PROOF OF PROPOSITION 1. Lemma 1 shows that the value of the density function corresponding to choosing $u_t = u$ in the maximum entropy feedback is proportional to the volume of $S((\mathbf{u}_{<t}, u))$, completing the proof of interpretability. According to the explicit expression in (12) of the maximum entropy feedback, the selected action u always ensures that $\mu(S((\mathbf{u}_{<t}, u)) > 0$ and therefore $S((\mathbf{u}_{<t}, u)$ is non-empty. This guarantees that the generated sequence $\mathbf{u}$ is always in S . $\square$

PROOF OF COROLLARY 5.1. The explicit expression in Lemma 1 ensures that whenever $p_t^*(u_t|\mathbf{u}_{<t}) > 0$, then there is always a feasible sequence of actions in $S(\mathbf{u}_{<t})$. Now, if the tuning parameter $\beta > 0$, then the optimization (17) guarantees that $p_t^*(u_t|\mathbf{u}_{<t}) > 0$ for all $t \in [T]$; otherwise, the objective value in (17) is unbounded. Proposition 1 guarantees that for any sequence of actions $\mathbf{u} = (u_1, \dots, u_T)$, if $p_t^*(u_t|\mathbf{u}_{<t}) > 0$ for all $t \in [T]$, then $\mathbf{u} \in S$. Therefore, the sequence of actions $\mathbf{u}$ given by the PPC is always feasible. $\square$

B.3 Proof of Theorem 5.1

The actions space U is closed and bounded by Assumption 1. Therefore, there exists a closed ball $\mathcal{B}(a)$ of radius $a > 0$ centered at $\mathbf{v} = (v_1, \dots, v_T) \in U$ such that $\mathcal{B}(a) \subseteq U$. Consider the following safety constraints for all $t \in [T] \setminus \{1\}$ with memory size one:

$$\mathcal{U}_t := \begin{cases} \mathcal{B}(a/2), & \text{if } \|u_{t-1} - v_{t-1}\|_2 \leq a \\ U \setminus \mathcal{B}(a), & \text{if } \|u_{t-1} - v_{t-1}\|_2 > a \end{cases} \quad (\text{B.1})$$

where $\mathcal{B}(a/2)$ is a closed balls around the same center as $\mathcal{B}(a)$ with radius $a/2$. Let the state safety set $\mathcal{X}_t = \mathbb{R}^n$ for all $t \in [T]$ (i.e., no constraints on states). Whenever the first action u_1 at time $t = 1$ is taken, the future actions have to stay in the ball $\mathcal{B}(a/2)$, or stay outside $\mathcal{B}(a)$. Any deterministic policy at time $t \in [T] \setminus \{1\}$ has to take either $\|u_t - v_t\|_2 \leq a/2$ or $\|u_t - v_t\|_2 > a$.

Consider the following Lipschitz continuous cost functions that can be chosen adversarially:

$$c_t(u_t) = \begin{cases} 0, & \text{if } t \leq w \\ \|u_t - v_t\|_2, & \text{if } \|u_{t-1} - v_{t-1}\|_2 > a, t > w \\ (M - \|u_t - v_t\|_2), & \text{if } \|u_{t-1} - v_{t-1}\|_2 \leq a/2, t > w \end{cases}$$

where $M := \sup_{u \in U} \|u\|_2$. The construction of $\mathbf{c} = (c_1, \dots, c_T)$ guarantees that the dynamic regret for any sequence of actions $\mathbf{u}$ given by a deterministic online policy is bounded from below by

$$\text{Regret}(\mathbf{u}) \geq (T - w) \min \left\{ a, \left(M - \frac{a}{2} \right) \right\}.$$

Take $a = M/2$ and note that $M = \Omega(d)$ where $d := \text{diam}(U) := \sup\{\|u - v\|_2 : u, v \in U\}$ is the diameter of the action space U . Therefore the dynamic regret is bounded from below as

$$\text{Regret}(\mathbf{u}) = \Omega(d(T - w))$$

for any sequence of actions $\mathbf{u}$ given by a deterministic online policy.

B.4 Proofs of Theorem 5.2 and Corollary 5.2

PROOF OF THEOREM 5.2. Suppose there exists a sequence of actions $\mathbf{v}_{\leq s} \in U^s$, $1 \leq s \leq T - k$, $k \geq w$ with $k = \Omega(T)$ and some constant $\alpha > 0$ such that

$$d_H(S_k(\mathbf{u}_{\leq s}), S_k(\mathbf{v}_{\leq s})) \geq \alpha dk \quad (\text{B.2})$$

for any $\mathbf{u}_{\leq s} \in U^s$. For any safety sets satisfying (B.2), whenever the first s actions $\mathbf{u}_{\leq s}$ are taken, the future actions have to stay in $S_k(\mathbf{u}_{\leq s})$. The same argument also holds for choosing $\mathbf{v}_{\leq s}$. This

means that there exist two subsequences $\mathbf{u}_{s+1:s+k}$ and $\mathbf{v}_{s+1:s+k}$ such that

$$\begin{aligned} \|\mathbf{u}_{s+w:s+k} - \mathbf{v}_{s+w:s+k}\|_2 &\geq \|\mathbf{u}_{s+1:s+k} - \mathbf{v}_{s+1:s+k}\|_2 - \|\mathbf{u}_{s+1:s+w-1} - \mathbf{v}_{s+1:s+k}\|_2 \\ &\geq \|\mathbf{u}_{s+1:s+k}\|_2 - \sum_{t=s+1}^{s+w-1} \|u_t - v_t\|_2 \\ &\geq \alpha dk - d(w-1) \\ &= d(\alpha k - w + 1) \end{aligned}$$

Then the adversary can construct Lipschitz continuous cost functions similar to the one we used in the proof of Theorem 5.1 so that the costs $c_{s+w}, \dots, c_{s+k}$ satisfy $\sum_{t=s+w}^{s+k} c(a_t) = \Omega((k-w+1)d) = \Omega(d(T-w))$ (since $k = \Omega(T)$) for the case when $a_t = v_t$ for $s < t \leq s+k$ and $a_t = u_t$ for $t \leq s$; but $c_{s+w} = \dots = c_{s+k} = 0$ for the case when $a_t = u_t$ for $s < t \leq s+k$ and $a_t = v_t$ for $t \leq s$ and vice versa. This “switching pattern” attack is possible, because the online agent knows nothing about the costs $c_{s+w}, \dots, c_{s+k}$ at time s and the adversary can design costs freely as long as they satisfy Assumption 1. The bound on the dynamic regret follows since for the optimal actions $\mathbf{u}^* = (u_1, \dots, u_T)$, $\sum_{t=s+w}^{s+k} c(u_t^*) = 0$ but $\sum_{t=s+w}^{s+k} c(a_t) = \Omega(d(T-w))$ for any actions $\mathbf{a}$ generated by a deterministic online policy. $\square$

PROOF OF COROLLARY 5.2. Applying Theorem 5.2 and noting that $\alpha = \Omega(T)$ and at each time t the received MEF functions $p_t^*, \dots, p_{t+w-1}^*$ equivalently form a joint density function on $\mathbf{u}_{t+1:t+w}$, which has less information than an online policy that knows the offline safety sets. Since $\bar{\mathbf{u}}$ maximizes $\mu(S(\cdot))$,

$$\frac{\mu(S(\mathbf{u}_{\leq s}))}{\mu(S(\mathbf{v}_{\leq s}))} \geq \frac{\mu(S(\mathbf{u}_{\leq s}))}{\mu(S(\bar{\mathbf{u}}_{\leq s}))} > \xi,$$

it follows that

$$\left(\frac{|\mu(S(\mathbf{u}_{\leq s})) - \xi \mu(S(\mathbf{v}_{\leq s}))|}{\mu(\mathcal{B})} \right)^{1/((T-s)m)} = \Omega(1).$$

Hence, the same argument in the proof of Theorem 5.2 follows and we obtain the desired regret lower bound. $\square$

B.5 Proof of Theorem 5.3

In this appendix, we prove Theorem 5.3 in four steps. First, in Lemma 3, we bound the deviation of the density function (feedback) evaluated at the actions selected by the PPC from the largest density function value, given the previous actions selected by the PPC. The deviation decreases with the tuning parameter $\beta > 0$. Next, using the bound obtained from the first step, in Lemma 4 we show that the ratio of the volume of the set of feasible actions, given the previous actions selected by the PPC and the volume of the largest feasible set is bounded from above by an exponential function, that is decreasing with β as well. Next, we bound the partial cost difference, of the cost induced by a subsequence of feasible actions and the offline optimal cost. Finally, combining the partial costs, we bound the total cost and this leads to an upper bound on the dynamic regret.

Step 1. Bound the feedback deviation

Recall that $t' := \min\{t+w-1, T\}$. For every $t \in \mathcal{I}$, let $\bar{\mathbf{u}}_{t:t'} = \bar{u}_t, \dots, \bar{u}_{t'}$ be a subsequence of optimal actions that maximizes the penalty term:

$$\bar{\mathbf{u}}_{t:t'} := \arg \sup_{\mathbf{u} \in \mathcal{U}^{t'-t+1}} p_{t:t'}^*(\mathbf{u} | \mathbf{u}_{<t}) = \arg \sup_{\mathbf{u} \in \mathcal{U}^{t'-t+1}} \mu(S((\mathbf{u}_{<t}, \mathbf{u}))).$$

Before proceeding to the proof of Theorem 5.3, we first show the following lemma, which gives a lower bound on the feedback given the sequence of actions selected by the PPC. Note that when $t - 1 < 0$, the density functions become unconditional.

LEMMA 3. For any $t \in \mathcal{I}$, the sequence of actions $\mathbf{u} = (\mathbf{u}_{\leq t'})$ selected by the PPC satisfies

$$\frac{p_{t:t'}^*(\mathbf{u}_{t:t'}|\mathbf{u}_{<t})}{p_{t:t'}^*(\bar{\mathbf{u}}_{t:t'}|\mathbf{u}_{<t})} \geq \exp(-L_c w d / \beta)$$

where L_c is the Lipschitz constant, $d := \text{diam}(\mathcal{U}) = \sup\{\|u - v\|_2 : u, v \in \mathcal{U}\}$ is the diameter of the action space $\mathcal{U}$.

PROOF OF LEMMA 3. First, we note that the actions $\mathbf{u}_{t:t'}$ chosen by the PPC must satisfy

$$\sum_{\tau=t}^{t'} \beta (\log p_{\tau}^*(\bar{\mathbf{u}}_{\tau}|\mathbf{u}_{<\tau}) - \log p_{\tau}^*(u_{\tau}|\mathbf{u}_{<\tau})) \leq \left| \sum_{\tau=t}^{t'} (c_{\tau}(\bar{\mathbf{u}}_{\tau}) - c_{\tau}(u_{\tau})) \right|. \quad (\text{B.3})$$

To see this, suppose (B.3) does not hold. Then choosing $\bar{\mathbf{u}}_{t:t'}$ gives a smaller objective value in (17), which violates the definition of PPC-generated actions $\mathbf{u}_{t:t'}$. Using the chain rule, (B.3) becomes

$$\log p_{t:t'}^*(\bar{\mathbf{u}}_{t:t'}|\mathbf{u}_{<t}) \leq \log p_{t:t'}^*(\mathbf{u}_{t:t'}|\mathbf{u}_{<t}) + \frac{1}{\beta} \left| \sum_{\tau=t}^{t'} (c_{\tau}(\bar{\mathbf{u}}_{\tau}) - c_{\tau}(u_{\tau})) \right|.$$

Therefore, since the cost functions are Lipschitz continuous,

$$\begin{aligned} p_{t:t'}^*(\bar{\mathbf{u}}_{t:t'}|\mathbf{u}_{<t}) &\leq \exp \left(\frac{1}{\beta} \left| \sum_{\tau=t}^{t'} (c_{\tau}(\bar{\mathbf{u}}_{\tau}) - c_{\tau}(u_{\tau})) \right| \right) p_{t:t'}^*(\mathbf{u}_{t:t'}|\mathbf{u}_{<t}) \\ &\leq \exp \left(L_c \frac{w d}{\beta} \right) p_{t:t'}^*(\mathbf{u}_{t:t'}|\mathbf{u}_{<t}). \end{aligned}$$

□

Step 2. Bound the Lebesgue measure deviation

Based on Lemma 3, the following lemma holds. For every $t \in \mathcal{I}$, let $\bar{\mathbf{u}}_{<t'} = u_1, \dots, u_{t'}$ be a subsequence of optimal actions maximizing the volume of the set of feasible actions:

$$\bar{\mathbf{u}}_{<t'} := \arg \sup_{\mathbf{u} \in \mathcal{U}^{t'-1}} \mu(S(\mathbf{u})).$$

LEMMA 4. Suppose the safety sets are (w, δ, λ) -causally invariant. For any $t \in \mathcal{I}$, the actions selected by the PPC satisfy that

$$\frac{\mu(S(\mathbf{u}_{\leq t'}))}{\mu(S(\bar{\mathbf{u}}_{<t'}))} \geq \exp(-L_c t d / \beta) \lambda^{-\lceil t/w \rceil}$$

where $L_c > 0$ is the Lipschitz constant, $d := \sup\{\|u - v\|_2 : u, v \in \mathcal{U}\}$ is the diameter of the action space $\mathcal{U}$.

PROOF. For any $t \in \mathcal{I}$, since the safety sets are (w, δ, λ) -causally invariant and $\mu(S(\mathbf{u}_{<t'})) = \mu(S(\mathbf{u}_{\leq t'}))$,

$$\frac{\mu(S(\mathbf{u}_{\leq t'}))}{\mu(S(\bar{\mathbf{u}}_{<t'}))} \geq \frac{1}{\lambda} \left(\frac{\mu(S(\mathbf{u}_{<t}))}{\mu(S(\bar{\mathbf{u}}_{<t}))} \right)^{\frac{T-t'+1}{T-t+1}} \cdot \frac{\mu(S(\mathbf{u}_{\leq t'}))}{\mu(S((\mathbf{u}_{<t}, \bar{\mathbf{u}}_{t:t'}))}. \quad (\text{B.4})$$

Applying the explicit expression in Lemma 1, Lemma 3 implies that

$$\frac{\mu(S(\mathbf{u}_{\leq t'}))}{\mu(S((\mathbf{u}_{<t}, \bar{\mathbf{u}}_{t:t'}))} = \frac{p_{t:t'}^*(\mathbf{u}_{t:t'}|\mathbf{u}_{<t})}{p_{t:t'}^*(\bar{\mathbf{u}}_{t:t'}|\mathbf{u}_{<t})} \geq \exp(-w L_c d / \beta).$$

Therefore, applying (B.4) recursively leads to that for any $t \in \mathcal{I}$,

$$\frac{\mu(S(\mathbf{u}_{\leq t'}))}{\mu(S(\bar{\mathbf{u}}_{< t'})} \geq \exp(-L_c t d / \beta) \lambda^{-\lceil t/w \rceil}.$$

□

Step 3. Bound the partial cost deviation

Lemma 4 implies that for the offline optimal solution $\mathbf{u}_{t':t'+w}^* := (u_{t'}^*, \dots, u_{t'+w}^*)$,

$$\begin{aligned} \mu(S(\mathbf{u}_{\leq t'})) &\geq \exp(-L_c t d / \beta) \lambda^{-2t/w} \mu(S(\bar{\mathbf{u}}_{< t'})) \\ &\geq \exp(-L_c t d / \beta) \lambda^{-2t/w} \mu(S(\mathbf{u}_{< t'}^*)), \end{aligned}$$

which leads to

$$\begin{aligned} \mu(S(\mathbf{u}_{< t'}^*)) - \mu(S(\mathbf{u}_{\leq t'})) &\leq \left(1 + \exp\left(\frac{2t}{w} \log \lambda - L_c \frac{td}{\beta}\right)\right) \mu(S(\mathbf{u}_{< t'}^*)) \\ &\leq \left(L_c \frac{td}{\beta} + \frac{2t}{w} \log \lambda\right) \mu(S(\mathbf{u}_{< t'}^*)) \\ &\leq \left(L_c \frac{td}{\beta} + \frac{2t}{w} \log \lambda\right) \mu(\mathcal{B}) \left(\frac{d}{2}\right)^{(T-t'+1)m} \end{aligned}$$

where we have used the inequality $e^x \geq 1 + x$ for all $x \in \mathbb{R}$. The last inequality follows from the isodiametric inequality, with $\mathcal{B}$ denoting the unit ball in $\mathbb{R}^{(T-t'+1)m}$.

Since the safety sets are (w, δ, λ) -causally invariant, it follows that for any $t \in \mathcal{I}$,

$$\|\mathbf{u}_{t:t'}^* - \hat{\mathbf{u}}_{t:t'}\|_2 \leq d_H(S_w(\mathbf{u}_{< t}^*), S_w(\mathbf{u}_{< t})) \leq \delta d \left(L_c \frac{td}{\beta} + \frac{2t}{w} \log \lambda\right)^{\frac{1}{(T-t+1)m}}$$

for some $\hat{\mathbf{u}}_{t:t'} \in S_w(\mathbf{u}_{< t})$ satisfying $p_{t:t'}^*(\hat{\mathbf{u}}_{t:t'} | \mathbf{u}_{< t}) > 0$.

Next, we consider the objective for the PPC. For any $t \in \mathcal{I}$, denote by $C(\mathbf{u}_{t:t'}) := \sum_{\tau=t}^{t'} c_\tau(u_\tau)$ the cost for the times slots between t and $t' := \min\{t+w, T\}$. Since the costs are Lipschitz continuous,

$$\begin{aligned} |C(\hat{\mathbf{u}}_{t:t'}) - C(\mathbf{u}_{t:t'}^*)| &\leq \sum_{\tau=t}^{t'} |c_\tau(u_\tau) - c_\tau(u_\tau^*)| \\ &\leq \sum_{\tau=t}^{t'} L_c \|u_\tau - u_\tau^*\|_2 \\ &\leq \sqrt{w} L_c \|\mathbf{u}_{t:t'}^* - \hat{\mathbf{u}}_{t:t'}\|_2 \\ &\leq d L_c \left(L_c \frac{td\sqrt{w}}{\beta} + \frac{2t}{\sqrt{w}} \log \lambda\right)^{\frac{1}{(T-t+1)m}}. \end{aligned} \tag{B.5}$$

Since $\mathbf{u}_{t:t'}$ is a minimizer of (17), we obtain

$$C(\mathbf{u}_{t:t'}) - \sum_{\tau=t}^{t'} \beta \log p_\tau(u_\tau | \mathbf{u}_{< \tau}) \leq C(\hat{\mathbf{u}}_{t:t'}) - \sum_{\tau=t}^{t'} \beta \log p_\tau(\hat{u}_\tau | \mathbf{u}_{< \tau}),$$

which implies

$$C(\mathbf{u}_{t:t'}) \leq C(\hat{\mathbf{u}}_{t:t'}) + \beta \log \frac{p_{t:t'}^*(\mathbf{u}_{t:t'} | \mathbf{u}_{< t})}{p_{t:t'}^*(\hat{\mathbf{u}}_{t:t'} | \mathbf{u}_{< t})}. \tag{B.6}$$

Step 4. Bound the dynamic regret $\text{Regret}(\mathbf{u})$

PROOF OF THEOREM 5.3. For the total cost, it follows that

$$\begin{aligned} C_T(\mathbf{u}) &= \sum_{t \in I} c_t(\mathbf{u}_{t:t'}) \leq \sum_{t \in I} \left[C(\widehat{\mathbf{u}}_{t:t'}) + \beta \log \frac{p_{t:t'}^*(\mathbf{u}_{t:t'} | \mathbf{u}_{<t})}{p_{t:t'}^*(\widehat{\mathbf{u}}_{t:t'} | \mathbf{u}_{<t})} \right] \\ &\leq \underbrace{\sum_{t \in I} C(\widehat{\mathbf{u}}_{t:t'})}_{:= (a)} + \beta \underbrace{\sum_{t \in I} \log \frac{\mu(S(\mathbf{u}_{<t}, \mathbf{u}_{t:t'}))}{\mu(S(\mathbf{u}_{<t}, \widehat{\mathbf{u}}_{t:t'}))}}_{:= (b)}. \end{aligned}$$

For (a), plugging in (B.5), the total cost for the sequence $\widehat{\mathbf{u}}$ is bounded from above by

$$\begin{aligned} \sum_{t \in I} C(\widehat{\mathbf{u}}_{t:t'}) &\leq C_T^* + \sum_{t \in I} |C(\widehat{\mathbf{u}}_{t:t'}) - C(\mathbf{u}_{t:t'}^*)| \\ &\leq C_T^* + \sum_{k=1}^{2T/w} d\delta L_c \left(L_c \frac{kd\sqrt{w}}{\beta} + \frac{k}{\sqrt{w}} \log \lambda \right)^{\frac{1}{(T-kw+1)m}} \\ &= C_T^* + O \left(\frac{T}{\sqrt{w}} d\delta \log \lambda + \frac{d^2\delta}{\beta} T \sqrt{w} \right) \end{aligned}$$

where $C_T^* := \sum_{t=1}^T c_t^*(\mathbf{u}_t^*)$ is the optimal total cost. Finally, (b) becomes $O(T\beta/w)$ since the safety sets are atomic and $\mathbf{U}$ is bounded by Assumption 2 and 3. Rearranging the terms,

$$\text{Regret}(\mathbf{u}) = O \left(Td \left(\frac{\delta \log \lambda}{\sqrt{w}} + \frac{d\delta\sqrt{w}}{\beta} + \frac{\beta}{dw} \right) \right).$$

Setting $\beta = d\sqrt{\delta}w^{3/4}$ immediately implies $\text{Regret}(\mathbf{u}) = O \left(Td \left((\delta \log \lambda)/\sqrt{w} + \sqrt{\delta}/w^{1/4} \right) \right)$. $\square$

Received February 2021; revised April 2021; accepted April 2021