Real-time Aggregate Flexibility via Reinforcement Learning

Tongxin Li, *Student Member, IEEE,* Bo Sun, *Member, IEEE,* Yue Chen, *Member, IEEE,* Zixin Ye, *Student Member, IEEE,* Steven H. Low, *Fellow, IEEE,* and Adam Wierman, *Member, IEEE*

***Abstract*—Aggregators have emerged as crucial tools for the coordination of distributed, controllable loads. To be used effectively, an aggregator must be able to communicate the available flexibility of the loads they control, as known as the *aggregate flexibility* to a system operator. However, most of existing aggregate flexibility measures often are slow-timescale estimations and much less attention has been paid to real-time coordination between an aggregator and an operator as a closed-loop control system. In this paper, we present a design of *real-time* aggregate flexibility feedback based on maximization of entropy, termed the *maximum entropy feedback* (MEF). The design provides a concise and informative signal that can be used by the system operator to perform online cost minimization, while provably satisfying the constraints of the loads. In addition to deriving analytic properties of the MEF, we show that it can be generated efficiently using reinforcement learning and used as a penalty term in model predictive control (MPC), which gives a novel algorithm – the *penalized predictive control* (PPC). The benefits of the PPC are (1). *Efficient Communication*. An operator running PPC does not need to know the exact states and constraints of the loads on the aggregator's side, but only the MEF sent by the aggregator. (2). *Fast Computation*. The PPC is an unconstrained online optimization and it often has much less number of variables than the optimization formulation of an MPC. (3). *Lower Costs* We illustrate the efficacy of the PPC using a dataset from an adaptive electric vehicle charging network and show that PPC outperforms classical MPC (with no terminal cost) by achieving lower operational costs.**

***Index Terms*—Closed-loop systems, aggregate flexibility, model predictive control, reinforcement learning, electric vehicle charging**

NOMENCLATURE

A. System Operator (Centralized Controller)

T	Total number of time slots.
t	Time index.
u_t	Operator action.
c_t	Cost function.
C_T	Cumulative costs.
ψ_t	Operator function.

Li, Low and Wierman are with the Computing + Mathematical Sciences Department, California Institute of Technology, Pasadena, CA 91125 USA (e-mails: {tongxin, slow, adamw}@caltech.edu)

Sun is with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong (e-mail: bsunaa@connect.ust.hk)

Chen is with the State Key Laboratory of Power Systems, Department of Electrical Engineering, Tsinghua University, Beijing (e-mail: cy11@tsinghua.org.cn)

Ye is with the Electrical Engineering Department, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: zyye@caltech.edu)

B. Aggregator (Local Controller)

x_t	Aggregator state.
p_t	Real-time aggregate flexibility feedback.
X_t	Set of feasible states.
U_t	Set of feasible actions.
S	Set of feasible action trajectories.
f_t	State transition function.
P	Set of flexibility feedback.
ϕ_t	Aggregator function.

C. EV Charging Example

n	Total number of accepted charging sessions.
j	Index of charging sessions.
u_t	Aggregate substation power level.
s_t	Charging decision vector.
$s_t(j)$	Scheduled energy.
$a(j)$	Arrival time.
$d(j)$	Departure time.
$e(j)$	Total energy to be delivered.
$r(j)$	Peak charging rate.
$d_t(j)$	Remaining charging time.
$e_t(j)$	Remaining energy demand.
Δ	Time unit.

I. INTRODUCTION

The need to manage the uncertainty and volatility caused by the growing penetration of renewable sources such as wind and solar power has created a desire to increase the ability of the system to provide flexibility via distributed energy resources (DERs) and aggregators have emerged as dominate players for coordinating these loads [1], [2]. The power of aggregators is that 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 flexibility. This enables ISOs to minimize cost, respond to unexpected fluctuations of renewables, and even mitigate failures quickly and reliably.

To realize the potential benefits of aggregators, ISOs need to be able to call on the aggregator via a time-varying signal, e.g., a desired power profile, that satisfies the DER constraints and optimizes ISO objectives. The signal is then disaggregated by the aggegator in order to determine the behavior of the loads under its control. However, the loads have private constraints on their operation (e.g., satisfying energy demands of electric vehicles before their deadlines). These constraints limit the flexibility available to the aggregator so the aggregator must also communicate with the ISO by providing a signal that

quantifies its available flexibility. This signal is of crucial importance for the ISO when determining the signal it sends to the aggregator, and thus the aggregator and the ISO form a closed-loop control system.

This paper focuses on the design of this closed-loop system and, in particular, the design of the signal quantifying the available flexibility sent from the aggregator to the ISO. The question of how to design the signal providing information on aggregate flexibility of the aggregator to the operator, namely *the aggregate flexibility feedback* signal, is complex and has been the subject of significant research over the last decade, e.g., [3], [4], [5], [6], [7], [8], [9], [10], [11]. Any feedback design must balance between a variety of conflicting goals. Given the scale of aggregators and the complexity of the constraints of loads, it is impossible to communicate precise information about every load. Instead, aggregate flexibility feedback must be a concise summary of a system's constraints. Even if it was possible, providing exact information about the constraints of each load governed by the aggregator would not be desirable because the load constraints are typically private. Information conveyed to the ISO must limit the leakage about specific load constraints. On the other hand, the feedback sent by an aggregator needs to be informative enough that it allows the ISO to achieve operational objectives, e.g., minimize cost, and, most importantly, guarantee the feasibility of the whole system with respect to the private load constraints. Moreover, a design for a flexibility feedback signal must be general enough to be applicable for a wide variety of controllable loads, e.g., electric vehicles (EVs), heating, ventilation, and air conditioning (HVAC) systems, energy storage units, thermostatically controlled loads, residential loads, and pool pumps. It is impractical to imagine different feedback signals for each load, so the same design must work for all DERs.

The challenge and importance of the design of flexibility feedback signals has led to the emergence of a rich literature. In many cases, the literature focuses on specific classes of controllable loads, such as EVs [12], heating, ventilation, and air conditioning (HVAC) systems [13], energy storage units [10], thermostatically controlled loads [4] or residential loads and pool pumps [5], [14]. In the context of these applications, there have been a variety of approaches suggested, e.g., convex geometric approximations [4], [6], [10], [8], [8], scheduling based aggregation [15], [16], [17], and probability-based characterization [5], [14]. These approaches have all yielded some success, especially in terms of quantifying the aggregate flexibility available (we go into more detail about these approaches in the related work section below. See Section I-B).

However, to this point there are no real-time designs of the coordination between an aggregator and a system operator that achieve the goals laid out above except for some preliminary results in [18], wherein the model considered is less general and no comparison with classical algorithms such as model predictive control (MPC) is given. In particular, the goal of providing *real-time* aggregate flexibility feedback has seemed unapproachable and nearly all prior work has focused on slower-timescale estimations. Given the fast-changing environment and the uncertainties of the DERs caused by the penetration of renewable resources and human behaviors, it is desirable to upgrade the network to incorporate real-time flexibility feedback. For example, in an EV charging station, it is notoriously challenging to predict future EV arrivals and their battery capacities and with roof-top solar panel installed, the aggregator's dynamic system can be time-varying and non-stationary so it is crucial that real-time feedback can be defined and approximated for it to be used in online feedback-based applications. Reinforcement learning (RL), especially, deep RL, has been used widely as approximation tools in smart grid applications. Joint pricing and EV charging scheduling for a single EV charger is considered in [19] using state–action–reward–state–action (SARSA). But it is unclear how the proposed method in [19] can be extended to allow multiple chargers. Q-learning is used to estimate the residual energy in an energy storage system at the end of each day in [20] and determine the aggregate action for thermostatically controlled loads (TCLs) [21]. The authors in [22] combine evolution strategies and model predictive control (MPC) to coordinate heterogeneous TCLs. Most existing studies, including the aforementioned works typically use RL for a "centralized controller" (which is an operator in our context) and it is not straightforward based on the literature that how RL can be used to learn flexibility representations.

A. Contributions.

To complement previous research, in this paper we consider a closed-loop control model formed by a system operator (centralized controller) and an aggregator (local controller) and propose a novel design of *real-time aggregate flexibility* feedback, called the *maximum entropy feedback* (MEF) that quantifies the flexibility available to an aggregator. Based on the definition of MEF, we design a reward function, which allows MEF to be efficiently learned by model-free RL algorithms. Our main contributions are summarized below.

1) We introduce a model of the real-time closed-loop control system formed by a system operator and an aggregator. This work is the first to close the loop and both define a concise measure of aggregate flexibility and show how it can be used by the system operator in an online manner to optimize system objectives while respecting the constraints of the aggregator's loads.
2) Within this model we define the "optimal" real-time flexibility feedback as the solution to an optimization problem that maximizes the entropy of the feedback vector. The use of entropy in this context is novel and to the best of our knowledge, this article is among the first to rigorously define a notion for *real-time* aggregate flexibility with provable properties. In particular we show that the MEF allows the system operator to maintain feasibility and enhance flexibility in real time in an online setting.
3) Furthermore, we propose a novel combination of model predictive control (MPC) and the defined MEF. Using the MEF as a penalty term, we introduce an algorithm called the *penalized predictive control* (PPC), which only requires the system operator to receive the MEF at each

time, *without* knowing the states and dynamics of the aggregator. Motivated by the mathematical definition of the MEF, we design a reward function and use an off-policy RL-based approach to efficiently approximate the MEF.

4) Finally, we demonstrate the efficacy of the proposed scheme using real EV charging data from Caltech's ACN-Data dataset [23]. Our experiments show that by sending simple action signals generated by the PPC, a system operator is able to coordinate with an EV charging aggregator to satisfy almost all the EV's charging demands, while only knowing the MEF learned by a model-free off-policy RL algorithm. The PPC is also showed to achieve lower cost than MPC, which in addition needs to have access to all the states of the loads.

B. Related literature.

The growing importance of aggregators for the integration of controllable loads and the challenge of defining and quantifying the flexibility provided by aggregators means that a rich literature on the topic has emerged. Broadly, this work can be separated into three approaches.

Convex geometric approximation. The idea of representing the set of aggregate loads as a virtual battery model dates back to [3], [4]. In [6], flexibility of an aggregation of thermostatically controlled loads (TCLs) was defined as the Minkowski sum of individual polytopes, which is approximated by the homothets of a virtual battery model using linear programming. The recent paper [8] takes a different approach and defines the aggregate flexibility as upper and lower bounds so that each trajectory to be tracked between the bounds is disaggregatable and thus feasible. However, convex geometric approaches cannot be extended to generate real-time flexibility signals because the approximated sets cannot be decomposed along the time axis. In [11], a belief function of setpoints is introduced for real-time control. However, feasibility can only be guaranteed when each setpoint is in the belief set and this may not be the case for systems with memory.

Scheduling algorithm-driven analysis. Scheduling algorithms that enable the aggregation of loads have been studied in depth over the past decade. The authors of [24] introduced a decentralized algorithm with a real-time implementation for EV charging to track a given load profile. The authors of [15] considered the feasibility of matching a given power trajectory and show that causal optimal policies do not exist. In this work, aggregate flexibility was implicitly considered as the set of all feasible power trajectories. Three heuristic causal scheduling policies were compared and the results were extended to aggregation of deferrable loads and storage in [16]. Furthermore, decentralized participation of flexible demand from heat pumps and EVs was addressed in [17]. Notably, the flexibility signals that have emerged from this literature are not general, i.e., the apply to specific policies and DERs.

Probability-based characterization. There is much less work on probabilistic methods. The aggregate flexibility of residential loads was defined based on positive and negative pattern variations by analyzing collective behaviour of aggregate users [5]. A randomized and decentralized control architecture for systems of deferrable loads was proposed in [14], with a linear time-invariant system approximation of the derived aggregate nonlinear model. Flexibility in this work was defined as an estimate of the proportion of loads that are operating. Our work falls into this category, but differs from previous papers in that entropy maximization for a closed-loop control system yield an interpretable signal that can be informative for operator objectives in real-time, as well as guarantee feasibility of the private constraints of loads.

Other approaches. Beyond the works described above, there are many other suggestions for metrics of aggregate flexibility, e.g., graphical-based measures [25] and data-driven approaches [25]. Most of these, and the approaches described above are evaluated at the aggregator level however, and much less attention has been paid to the question of real-time coordination between an ISO and an aggregator that controls decentralized loads.

The assessment and enhancement of aggregate flexibility are often considered independent of the operational objectives today. For instance, the notion of aggregated flexibility is reported to an ISO participating in a reserve market a day ahead and the scheduling is then conducted the next day after receiving the flexibility representation as defined in [3], [26], [8], [7], with notable exceptions, such as [12], which considered charging and discharging of EV fleets batteries for tracking a sequence of automatic generation control (AGC) signals. However, this approach has several limitations. First, in large-scale systems, knowing the exact states of each load is not realistic. Second, classical flexibility representations often rely on a precise state-transition model on the aggregator's side. Third, traditional ISO market designs, such as a day-ahead energy market, often make use of ex ante estimates of future system states. The forecasts of the future states can sometime be far from reality, because of either an inaccurate model is used, or an uncertain event occurs. In contrast, a real-time energy market [27], [28] provides more robust system control when facing uncertainty in the environment, e.g., from fast-changing renewable resources or human behavioral parameters. This further highlights the need for real-time flexibility feedback, and serves to differentiate the approach in our paper.

Notation and Conventions. We use $\mathbb{P}(\cdot)$ and $\mathbb{E}(\cdot)$ to denote the probability distribution and expectation of random variables. The (discrete) entropy function is denoted by $\mathbb{H}(\cdot)$. To distinguish random variables and their realizations, we follow the convention to denote the former by capital letters (e.g., U) and the latter by lower case letters (e.g., u). Furthermore, we denote the length-t prefix of a vector u by $u_{\leq t} := (u_1, \dots, u_t)$. Similarly, $u_{<t} := (u_1, \dots, u_{t-1})$ and $u_{a \to b} := (u_a, \dots, u_b)$. The concatenation of two vectors u and v is denoted by (u, v). Given two vectors $u, v \in \mathbb{R}^n$, we write $u \preceq v$ if $u_i \leq v_i$ for all $i = 1, \dots, n$. For $x \in \mathbb{R}$, denote $[x]_+ := \max\{0, x\}$.

The rest of the paper is organized as follows. We present our closed-loop control model in Section II. We define real-time aggregate flexibility, called the MEF, and prove its properties in Section IV. An RL-based approach for estimating

the MEF is provided in Section V. Combining MEF and model MPC, we propose an algorithm, termed the PPC in Section VI-B. Numerical results are given in Section VII. Finally, we conclude this paper in Section VIII.

II. Problem Formulation

In this paper, we consider a real-time control problem involving two parties – a *load aggregator* and an independent system operator (*ISO*), or simply called an *operator* that interact over a discrete time horizon $[T] := \{1, \dots, T\}$.

A. Load aggregator

A *load aggregator* is a device, often considered as a local controller that controls a fleet of controllable loads. In this part, we formally state the model of an aggregator and its objective. Let x_t denote the *aggregator state* at time t that takes value in a certain set X. To this end, the aggregator receives an *action* $u_t \in \mathsf{U}$ where U denotes a discrete set [1] of actions at each time t from a *system operator*, which will be formally defined in Section II-B. The action space U and state space X are prefixed and known as common knowledge to both the aggregator and the system operator. The goal of the aggregator is to accomplish a certain task over the horizon $[T]$, e.g., delivering energy to a set of EVs by their deadlines while minimizing the costs, subject to system constraints. Mathematically, the constraints are represented by two collections of *time-varying* and *time-coupling* sets $\{\mathsf{X}_t(x_{<t}, u_{<t}) \subseteq \mathsf{X} : t \in [T]\}$ and $\{\mathsf{U}_t(x_{<t}, u_{<t})_t \subseteq \mathsf{U} : t \in [T]\}$. For notational simplicity, we denote $\mathsf{X}_t(x_{<t}, u_{<t})$ by X_t and $\mathsf{U}_t(x_{<t}, u_{<t})$ by U_t in the remaining contexts. The states and actions must satisfy $x_t \in \mathsf{X}_t$ and $u_t \in \mathsf{U}_t$ for all $t \in [T]$. The decision changes the aggregator state x_t according to a *state transition function* f_t:

$$x_{t+1} = f_t(x_t, u_t), \;\; x_t \in \mathsf{X}_t, \; u_t \in \mathsf{U}_t, \tag{1}$$

where f_t represents the transition of the state x_t. The aggregator state x_t and decision u_t need to be chosen from two time-varying sets X_t and U_t.

The aggregator has flexibility in its actions u_t for accomplishing its task and, we assume for this paper, is indifferent to these decisions as long as the task is accomplished by time T. At each time t, based on its current state x_t, the aggregator needs to send *flexibility feedback*, p_t, from a collection of feedback signals P, to the system operator, which describes the flexibility of the aggregator for accepting different actions u_t. Designing p_t is one of the central problems considered in this paper. Below we state the aggregator's goal in the real-time control system.

Aggregator's Objective. *The goal of the aggregator is two-fold: (1). Maintain the feasibility of the system and guarantee that $x_t \in \mathsf{X}_t$ and $u_t \in \mathsf{U}_t$ for all $t \in [T]$. (2). Generate flexibility feedback p_t and send it to the operator at time $t \in [T]$.*

[1] We assume that U is discrete only for simplicity of presentation. Our results, for example, the definition of maximum entropy feedback (Definition IV.1), Theorem IV.1 can be extended to continuous space using a density function as the flexibility feedback, changing the summations to integrals, replacing the discrete entropy functions by differential entropy functions.

B. System operator

A system operator is a centralized controller that operates the power grid. Knowing the flexibility feedback p_t from the aggregator, the operator sends an action u_t, chosen from U to the aggregator at each time $t \in [T]$. Each action is associated with a cost function $c_t(\cdot) : \mathsf{U} \to \mathbb{R}_+$, e.g., the aggregate EV charging rate increases load on the electricity grid. The system's objective is stated as follows.

Operator's Objective. *The goal of the system operator is to provide an action $u_t \in \mathsf{U}$ at time $t \in [T]$ to the aggregator so as to minimize the cumulative system costs given by $C_T(u_1, \dots, u_T) := \sum_{t=1}^{T} c_t(u_t)$.*

C. Real-time operator-aggregator coordination

Overall, considering the aggregator and operator's objectives, the goal of the closed-loop system is to solve the following problem in *real-time*, by coordinating the operator and aggregator via $\{p_t : t \in [T]\}$ and $\{u_t : t \in [T]\}$:

$$\min_{u_1, \dots, u_T} C_T(u_1, \dots, u_T) \tag{2a}$$

$$\text{subject to } \forall t = 1, \dots, T :$$

$$x_{t+1} = f_t(x_t, u_t) \tag{2b}$$

$$x_t \in \mathsf{X}_t, \tag{2c}$$

$$u_t \in \mathsf{U}_t \tag{2d}$$

i.e., the operator aims to minimize its cost C_T in (2a) while the load aggregator needs to fulfill its obligations in the form of constraints (2b)-(2d). This is an offline problem that involves global information at all times $t \in [T]$. The challenges are (i). the aggregator and operator need to solve the online version of (2) and (ii). the constraints (2b)-(2d) are private to the operator. It is impractical for the aggregator to communicate the constraints to the operator because of privacy concerns or computational effort. Moreover, in an online setting, even the aggregator will not know all the constraints at each time $t \in [T]$ that involve future information, e.g., future EV arrivals in an EV charging station. In this work, We explore a solution where the system operator and the aggregator jointly solve an online version of (2) in a closed loop in real-time, as illustrated in Figure 1.

***Remark* 1.** For simplicity, we describe our model in an offline setting where the cost and the constraints in the optimization problem (2) are expressed in terms of the entire trajectories of states and actions. The goal of the closed-loop control system is, however, to solve an online optimization via operator-aggregator coordination.

The real-time operator-aggregator coordination illustrated in Figure 1 does not require the aggregator to know the system operator's objective in (2a), but only the action u_t at each time t from the operator. In addition, it does not require the system operator to know the aggregator constraints in (2b), but only a feedback signal p_t (to be designed) from the aggregator. After receiving flexibility feedback p_t, the system operator generates its action u_t using a causal *operator function* $\phi_t(\cdot) : \mathsf{P} \to \mathsf{U}$. Knowing the state x_t, the aggregator generates its feedback

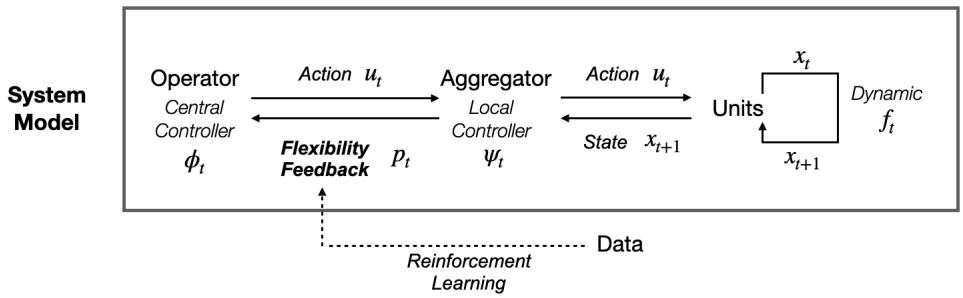

Fig. 1. System model: A feedback control approach for solving an online version of (2).

p_t using a causal *aggregator function* function $\psi_t(\cdot) : \mathsf{X} \to \mathsf{P}$ where P denotes the domain of flexibility feedback that will be formally defined in Section IV-A. By an "online feedback" solution, we mean that these functions (ϕ_t, ψ_t) use only information available locally at time $t \in [T]$.

As a summary, the closed-loop control system in our model proceeds as follows. At each time t, the aggregator computes a length-$|\mathsf{U}|$ vector p_t based on previously received action trajectory $u_{<t} = (u_1, \ldots, u_{t-1})$, and sends it to the system operator.[2] The system operator then computes a (possibly random) action $u_t = \phi_t(p_t)$ based on the flexibility feedback p_t and sends it to the aggregator. The operator chooses its signal u_t in order to solve the time-t problem in an online version of (2), so the function ϕ_t denotes the mapping from the flexibility feedback p_t to an optimal solution of the time-t problem. See VI-B for examples. The aggregator then computes the next feedback p_{t+1} and the cycle repeats; see Algorithm 1. The goal of this paper is to provide concrete constructions of an aggregator function ψ (as an MEF generator; see Section IV) and an operator function ϕ (via the PPC scheme; see Section VI-B).

In the sequel, we demonstrate our system model using an EV charging application, as an example of the problem stated in (2).

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

Operator (Centralized Controller)

Generate actions using the PPC:

$$u_t = \phi_t(p_t)$$
$$C_t = C_{t-1} + c_t(u_t)$$

Aggregator (Local Controller)

Update system state:

$$x_{t+1} = f_t(x_t, u_t)$$

Compute estimated MEF:

$$p_{t+1} = \psi(x_{t+1})$$

end

Return Total cost C_T;

Algorithm 1: Closed-loop online control framework of a system operator (centralized controller) and an aggregator (local controller).

III. An EV Charging Example

Consider an aggregator that is an EV charging facility with n accepted users. Each user j has a 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 *aggregator state* x_t at time $t \in [T]$ is a collection of length-2 vectors $(d_t(j), e_t(j) : a(j) \le t \le d(j))$ for each EV that has arrived and has not departed by time t. Here $e_t(j)$ is the remaining energy demand of user j at time t and $d_t(j)$ is the remaining charging time. The decision $s_t(j)$ is the energy delivered to each user j at time t, determined by a *scheduling policy* π_t such as the well-known earliest-deadline-first, least-laxity-first, etc. Let $s_t := (s_t(1), \ldots, s_t(n))$ and we have $s_t = \pi_t(u_t)$ where u_t in this example is the aggregate substation power level, chosen from a small discrete set U. The aggregator decision $s_t(j) \in \mathbb{R}_+$ at each time t updates the state, in particular $e_t(j)$ such that

$$e_t(j) = e_{t-1}(j) - s_t(j) \tag{3a}$$
$$d_t(j) = d_{t-1}(j) - \Delta \tag{3b}$$

where Δ denotes the time unit and we assume that there is no energy loss. The laws (3a)-(3b) are examples of the generic transition functions $f_1, \ldots, f_T$ in (1).

Suppose, in the context of demand response, the system operator (a local utility company, or a building management) sends a signal u_t that is the aggregate energy that can be allocated to EV charging. The aggregator makes charging decisions $s_t(j)$ to track the signal u_t received from the system operator as long as they will meet the energy demands of all users before their deadlines. Then the constraints in (2b)-(2d) are the following constraints on the charging decisions s_t, as a function of u_t:

$$s_t(j) = 0 \ , \ t < a(j), \ j = 1, \ldots, n, \tag{4a}$$
$$s_t(j) = 0 \ , \ t > d(j), \ j = 1, \ldots, n,$$

[2] We will omit $u_{<t}$ in the notation when it is not essential to our discussion and simplify the probability vector as p_t. Note that in (6c) we slightly abuse the notation and use p_t to denote a conditional distribution. This is only for computational purposes and the information sent from an aggregator to an operator at time $t \in [T]$ is still a length-$|\mathsf{U}|$ probability vector, conditioned on a fixed $u_{<t}$.

$$\sum_{j=1}^{n} s_t(j) = u_t,\ t = 1,\dots,T, \tag{4b}$$

$$\sum_{t=1}^{T} s_t(j) = e(j),\ j = 1,\dots,n, \tag{4c}$$

$$0 \le s_t(j) \le r(j),\ t = 1,\dots,T \tag{4d}$$

In above, constraint (4b) ensures that the aggregator decision s_t tracks the signal u_t at each time $t \in [T]$, the constraint (4c) guarantees that EV j's energy demand is satisfied, and the other constraints say that the aggregator cannot charge an EV before its arrival, after its departure, or at a rate that exceeds its limit. Inequalities (4a)-(4d) above are examples of the constraints in (1). Together, for this EV charging application, (3a)-(3b) and (4a)-(4d) exemplify the dynamic system in (1).

The system operator's objective to minimize the cumulative costs $C_T(u) = \sum_{t=1}^{T} c_t(u_t)$ where $u = (u_1,\dots,u_T)$, as outlined in Section II-B. The cost function c_t depends on multiple factors such as the electricity prices and injections from an installed rooftop solar panel.

IV. Definitions of Real-time Aggregate Flexibility: Maximum Entropy Feedback

In this section we propose a specific function ψ_t in the class defined by (5) for computing flexibility feedback to quantify its future flexibility. We will justify our proposal by showing that the proposed ψ_t has several desirable properties for solving an online version of (2) using the real-time feedback-based approach described in Section II.

A. Definition of Flexibility Feedback p_t

A major challenge in our problem is that the operator has access to neither the feasible set nor the dynamics directly. Therefore, a notion termed *aggregate flexibility* has to be designed. It is often a "simplified" summary of the constraints in (2b)-(2d), as we reviewed in Section I-B. Notably, existing aggregate flexibility definitions (for instance, in [3], [4], [5], [6], [7], [8], [9], [10]) all focus on the offline version of (2). It remains unclear that first, *what is the right notion of real-time aggregate flexibility? i.e., what is the right form of the flexibility feedback p_t?* Second, *how can this p_t be used by an operator?*

In the following, we propose a design of the flexibility feedback p_t that quantifies its future flexibility that will be enabled by an operator action u_t. The feedback p_t therefore is a surrogate for the aggregator constraints (2b) to guide the operator's decision. Let $u := (u_1,\dots,u_T)$. Specifically, define the set of all *feasible action trajectories* for the aggregator as:

$$\mathsf{S} := \left\{ u \in \mathsf{U}^T : u \text{ satisfies (2b)} - \text{(2d)} \right\}.$$

Existing aggregate flexibility definitions focus on approximating S such as finding its convex approximation (see Section I-B for more details). Our problem formulation needs a *real-time* approximation of this set S, i.e., decompose S along the time axis $t = 1,\dots,T$. Throughout, we assume that S is non-empty.

Next, we define the space of flexibility feedback p_t. Formally, let P denote the probability simplex:

$$\mathsf{P} := \left\{ p \in \mathbb{R}^{|\mathsf{U}|} : p(u) \ge 0, u \in \mathsf{U}; \sum_{u \in \mathsf{U}} p(u) = 1 \right\}.$$

Fix x_t at time $t \in [T]$. The aggregator function $\psi_t(\cdot) : \mathsf{X} \to \mathsf{P}$ at each time t is:

$$\psi_t(x_t) = p_t(\cdot | u_{<t}) \tag{5}$$

such that $p_t(\cdot|u_{<t}) : \mathsf{U} \to [0,1]$ is a (conditional) probability distribution in P. We refer to p_t as *flexibility feedback* sent at time $t \in [T]$ from the aggregator to the system operator. In this sense, (5) does not specify a specific aggregator function ψ_t, but a class of possible functions ψ_t. Every function in this collection is *causal* in that it depends only on information available to the aggregator at time t. In contrast to most aggregate flexibility notions in the literature [3], [4], [5], [6], [7], [8], [9], [10], the flexibility feedback here is specifically designed for an online feedback control setting.

B. Maximum entropy feedback

The intuition behind our proposal is using the conditional probability $p_t(u_t|u_{<t})$ to measure the resulting future flexibility of the aggregator if the system operator chooses u_t as the signal at time t, given the action trajectory up to time $t-1$. The sum of the conditional entropy of p_t thus is a measure of how informative the overall feedback is. This suggests choosing a conditional distribution p_t that maximizes its conditional entropy. Consider the optimization problem:

$$F \ := \max_{p_1,\dots,p_T} \sum_{t=1}^{T} \mathbb{H}(U_t|U_{<t}) \text{ subject to } U \in \mathsf{S} \tag{6a}$$

where the variables are conditional distributions:

$$p_t \ := \ p_t(\cdot|\cdot) := \mathbb{P}_{U_t|U_{<t}}(\cdot|\cdot), \qquad t \in [T], \tag{6b}$$

$U \in \mathsf{U}$ is a random variable distributed according to the joint distribution $\prod_{t=1}^{T} p_t$ and $\mathbb{H}(U_t|U_{<t})$ is the conditional entropy of p_t defined as:

$$\mathbb{H}(U_t|U_{<t}) := \sum_{u_1,\dots,u_t \in \mathsf{U}} \Big(-\prod_{\ell=1}^{t} p_\ell(u_\ell|u_{<\ell}) \Big) \log p_t(u_t|u_{<t}). \tag{6c}$$

By definition, a quantity conditioned on "$u_{<1}$" means an unconditional quantity, so in the above, $\mathbb{H}(U_1|U_{<1}) := \mathbb{H}(U_1) := \mathbb{H}(p_1)$.

The chain rule shows that $\sum_{t=1}^{T} \mathbb{H}(U_t|U_{<t}) = \mathbb{H}(U)$. Hence (6) can be interpreted as maximizing the entropy $\mathbb{H}(U)$ of a random trajectory U sampled according to the joint distribution $\prod_{t=1}^{T} p_t$, conditioned on U satisfying $U \in \mathsf{S}$, where the maximization is over the collection of conditional distributions $(p_1,\dots,p_T)$. We provide in Appendix IV-D an axiomatic justification of maximizing the entropy $\mathbb{H}(U)$ of the action trajectory U in (6a).

***Definition* IV.1** (Maximum entropy feedback)**.** *The flexibility feedback $p_t^* = \psi_t^*(u_{<t})$ for $t \in [T]$ is called the maximum*

entropy feedback (MEF) if $(p_1^*,\dots,p_T^*)$ *is the unique optimal solution of* (6).

***Remark* 2.** Even though the optimization problem (6) involves variables p_t for the entire time horizon $[T]$, the individual variables p_t in (6c) are conditional probabilities that depend only on information available to the aggregator at times t. Therefore the maximum entropy feedback p_t^* in Definition IV.1 is indeed causal and in the class of p_t^* defined in (5). The existence and uniqueness of p_t^* is guaranteed by Theorem IV.1 below, which also implies that p_t^* is unique. □

We demonstrate Definition IV.1 using a toy example.

***Example* IV.1** (Maximum entropy feedback p^*)**.** *Consider the following instance of the EV charging example in Section III. Suppose the number of charging time slots is* $T=3$ *and there is one customer, whose private vector is* $(1,3,1,1)$ *and possible energy levels are* 0 *(kWh) and* 1 *(kWh), i.e.,* $\mathsf{U}\equiv\{0,1\}$*. Since there is only one EV, the scheduling algorithm u (disaggregation policy) assigns all power to this single EV. For this particular choices of x and u, the set of feasible trajectories is* $\mathsf{S}=\{(0,0,1),(0,1,0),(1,0,0)\}$*, shown in Figure 2 with the corresponding optimal conditional distributions given by* (6).

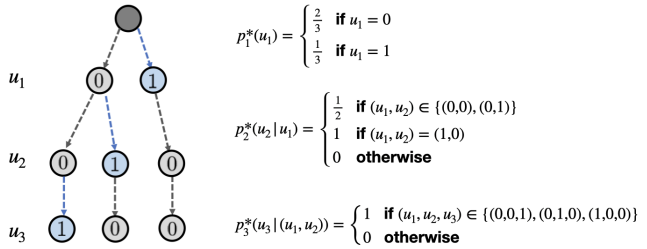

Fig. 2. Feasible trajectories of power signals and the computed maximum entropy feedback in Example IV.1.

C. Properties of p_t^*

We now show that the proposed maximum entropy feedback p_t^* has several desirable properties. We start by computing p_t^* explicitly. Given any action trajectory $u_{\le t}$, define the set of *subsequent* feasible trajectories as:

$$\mathsf{S}(u_{\le t}) := \Big\{ v_{>t}\in \mathsf{U}^{T-t} : v \text{ satisfies (2b)}-\text{(2d)}, v_{\le t}=u_{\le t}\Big\}.$$

The size $|\mathsf{S}(u_{\le t})|$ of the set of subsequent feasible trajectories is a measure of future flexibility, conditioned on $u_{\le t}$. Our first result justifies our calling p_t^* the optimal flexibility feedback: p_t^* is a measure of the future flexibility that will be enabled by the operator's action u_t and it attains a measure of system capacity for flexibility (see Remark 3 below). By definition, $p_1^*(u_1|u_{<1}) := p_1^*(u_1)$.

***Theorem* IV.1.** *The optimal flexibility feedback* p_t^* *is given by*

$$p_t^*(u_t|u_{<t}) \;=\; \frac{|\mathsf{S}((u_{<t},u_t))|}{|\mathsf{S}(u_{<t})|}, \quad \forall (u_{<t},u_t)\in\mathsf{U}^t. \tag{7}$$

for $t\in[T]$*. Moreover, the optimal value* F *of* (6) *is equal to* $\log|\mathsf{S}|$.

The proof can be found in Appendix A. Given the unique maximum entropy feedback $(p_1^*,\dots,p_T^*)$ guaranteed by Theorem IV.1, let $q^*(u)=\prod_{t=1}^T p_t^*(u_t|u_{<t})$ denote the joint distribution of the action trajectory u. Then (7) implies that the joint distribution q^* is the uniform distribution over the set S of all feasible trajectories:

$$q^*(u) := \begin{cases} 1/|\mathsf{S}| & \text{if } u\in\mathsf{S}\\ 0 & \text{otherwise}\end{cases}. \tag{8}$$

***Remark* 3** (System capacity F)**.** The size $|\mathsf{S}|$ is a measure of flexibility inherent in the aggregator. We will hence call $\log|\mathsf{S}|$ the *system capacity*. Theorem IV.1 then says that the optimal value of (6) is the system capacity, $F=\log|\mathsf{S}|$. Moreover the maximum entropy feedback $(p_1^*,\dots,p_T^*)$ is the unique collection of conditional distributions that attains the system capacity in (6). This is intuitive since the entropy of a random trajectory x in S is maximized by the uniform distribution q^* in (15) induced by the conditional distributions $(p_1^*,\dots,p_T^*)$. □

Theorem IV.1 implies the following important properties of the maximum entropy feedback.

***Corollary* IV.1** (Feasibility and flexibility)**.** *Let* $p_t^*=p_t^*(\cdot|u_{<t})$ *be the maximum entropy feedback at each time* $t\in[T]$.

1) *For any action trajectory* $u=(u_1,\dots,u_T)$*, if*
$$p_t^*(u_t|u_{<t}) \;>\; 0 \quad \textit{for all } t\in[T]$$
then $u\in\mathsf{S}$.
2) *For all* $u_t,u_t'\in\mathsf{U}$ *at each time* $t\in[T]$*, if*
$$p_t^*(u_t|u_{<t}) \;\ge\; p_t^*(u_t'|u_{<t})$$
then $|\mathsf{S}((u_{<t},u_t))|\ge|\mathsf{S}((u_{<t},u_t'))|$.

Proof of Corollary IV.1. Theorem IV.1 shows that the value of the probability distribution corresponding to choosing $u_t=u$ in the MEF is proportional to the size of $\mathsf{S}((u_{<t},u))$, completing the proof of interpretability. According to the explicit expression in (7) of the MEF, the selected action u always ensures that $|(\mathsf{S}((u_{<t},u))|>0$ and therefore the set $\mathsf{S}((u_{<t},u)$ is non-empty. This guarantees that the generated sequence u is always in S.

□

We elaborate on the implication of Corollary IV.1 on our online feedback-based solution approach.

***Remark* 4** (Feasibility and flexibility)**.** Corollary IV.1 says that the proposed optimal flexibility feedback p_t^* provides the right information for the system operator to choose its action u_t at time t. Specifically, the first statement of the corollary says that if the operator always chooses an action u_t with positive conditional probability $p_t^*(u_t)>0$ for each time t, then the resulting action trajectory is guaranteed to be feasible, $u\in\mathsf{S}$, i.e., the system will remain feasible at *every* time $t\in[T]$ along the way.

Moreover, according to the second statement of the corollary, if the system operator chooses an action u_t with a larger $p_t^*(u_t)$ value at time t, then the system 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 $\mathsf{S}((u_{<t}, u_t))$ going forward. □

As noted in Remark 1, despite characterizations that involve the whole action trajectory u, such as $u \in \mathsf{S}$, these are *online* properties. This guarantees the feasibility of the online closed-loop control system depicted in Figure 1, and confirms the suitability of p_t^* for online applications.

D. Axiomatic justification of (6)

As explained in Remark 4, the maximum entropy feedback p_t^* quantifies succinctly for the system operator the future flexibility of the aggregator that will be enabled by the operator's choice of next action u_t. Intuitively, the system has "more flexibility" at time t if the distribution $p_t(\cdot|u_{<t})$ is "more uniform". This view suggests using an entropic measure to quantify flexibility, such as the cost function of the optimization problem (6) that underlies our proposed flexibility feedback. In this subsection we justify this intuition using an axiomatic argument.

Consider a flexibility metric as a function of any flexibility feedback $p \in \{p_1, \ldots, p_T\}$. Recall that p is a conditional distribution. For any p, let $F(p)$ represent a candidate metric for quantifying aggregate flexibility. Consider any time slots $\tau \in [T]$, the metric should also be able to provide a value, given the marginal distributions $\overline{p}_t := \sum_{u_{<t}} p_t(\cdot|u_{<t}) \prod_{\tau<t} p_\tau(u_\tau|u_{<\tau})$.

We require the metric F to satisfy several conditions (axioms):

1) *Continuity*: $F(\overline{p}_t)$ is a continuous function of $\overline{p}_t$, $t \in [T]$.
2) *(Strong) additivity*: $F(q) = \sum_{t=1}^T F(p_t)$ if $q := \prod_{t=1}^T p_t$.
3) *Subadditivity*: $F(q_{t,t'}) \le F(\overline{p}_t) + F(\overline{p}_{t'})$ where $\overline{p}_t, \overline{p}_{t'}$ are marginal distributions corresponding to time slots t and t' and $q_{t,t'}$ is their joint distribution.
4) *Symmetry*: $F(q_{t,t'}) = F(q_{t',t})$ where $q_{t,t'}$ and $q_{t',t}$ are joint distributions of time slots t and t'.
5) *Expansibility*: $F(\overline{p}_t') = F(\overline{p}_t')$ for all $\overline{p}_t$, $t \in [T]$ where $\overline{p}_t' = (\overline{p}_t, 0)$, i.e., concatenate a zero entry to $\overline{p}_t$.

Additivity is useful because the tracking of a random action trajectory $u := (u_1, \ldots, u_T)$ can then be decomposed using the chain rule into sub-problems of tracking each action u_t at time t, conditioned on previous action trajectory $u_{<t}$. Subadditivity is motivated by the property that fixing an action u_t may restrict the choice of feasible actions $u_{t'}$ since the actions $u_1, \ldots, u_T$ may be correlated. This means that measuring the joint distribution of $(u_t, u_{t'})$ gives lower flexibility than measuring the coordinates u_t and $u_{t'}$ independently. For symmetry, the permutation of components in the distribution $\overline{p}_t$ does not change $F(\overline{p}_t)$ since the switch of positions does not affect the underlying distribution. Expansibility is natural since adding a new component that equals to zero means u_t can never choose a certain power level. So the aggregate flexibility will not change.

These five conditions imply that the flexibility metric $F(\overline{p}_t)$ (for all $t \in [T]$) must be an entropy function:

$$\mathbb{H}(\overline{p}_t) := \sum_{u \in \mathsf{U}} \sum_{u_{<t}} p_t(u|u_{<t}) \prod_{\tau<t} p_\tau(u_\tau|u_{<\tau}) \cdot \log\left(\frac{1}{\sum_{u_{<t}} p_t(u|u_{<t}) \prod_{\tau<t} p_\tau(u_\tau|u_{<\tau})}\right)$$

up to multiplicative factors and $F(p_t)$ is the conditional entropy of p_t. This is a classical result about entropy; see [29], [30]. The results in this section justify the design of using the unique optimal solution of (6) as our flexibility feedback p_t^*. The design attains the system capacity F. Moreover it characterizes the aggregate flexibility in real-time and allows a decomposition of aggregate flexibility over t via

$$\sum_{t=1}^T \mathbb{H}(p_t^*) = F = \log|\mathsf{S}|. \tag{9}$$

We use this decomposition in Section VI-B for online cost minimization where p_t^* is used as a penalty in a RHC-based online algorithm. In the remainder of this section, we show that there is a connection between MEF and the system capacity $F = \log|\mathsf{S}|$.

E. System Capacity Estimation

In addition to minimizing cost, another important goal of the operator is to quantify the amount of flexibility available at each time. This is crucial for purposes of ensuring the ability to respond to failures and planning for capacity investment. However, given that the private constraints of loads are not visible to the operator, such estimation is challenging. Further, measuring the exact size of S is intractable even if such constraints were visible, since S can be non-convex and even computing the volume of a convex body can be a challenging problem [31].

The task of the operator is to, given the maximum entropy feedback received as a sequence $p_1, \ldots, p_T$ by the operator, estimate the system capacity F while also generating signals $x_1, \ldots, x_T$ that are always feasible with respect to both the operational constraints (2d) and the private aggregator constraints (2b). The approach we propose is an empirical estimation of the system capacity using Monte Carlo estimation. In particular, we consider

$$\mu_N := \frac{1}{N} \sum_{\ell=1}^N \sum_{t=1}^T \mathbb{H}\left(p_t(\cdot|x_{<t}(\ell))\right), \tag{10}$$

where the summation is over T discrete time slots and N trajectories. For each, the corresponding entropy function computes the entropy of the flexibility feedback vector p_t conditioned on the generated trajectory of actions $u_{<t}(\ell))$ at each time $t \in [T]$:

$$\mathbb{H}\left(p_t(\cdot|u_{<t}(\ell))\right) := -\sum_{u \in \mathsf{U}} p_t\left(u|u_{<t}(\ell)\right) \log p_t\left(u|u_{<t}(\ell)\right).$$

The goal of this approach is that, with suitable choices of operator functions ϕ_t, when the number N of sampled trajectories becomes large, the approximation converges to the system capacity F. To see why, suppose at each time $t \in [T]$, the operator function ϕ_t^{CE} for *capacity estimation* is a stochastic function that samples a random action U_t according to the MEF p_t^*, i.e., for all $t \in [T]$ and $v_t \in \mathsf{U}$,

$$\mathbb{P}\left(\phi_t\left(p_t^*(\cdot|u_{<t}\right) = v_t\right) = p_t^*(v_t|u_{<t}).$$

In this context, the theorem below shows that we obtain an estimate of the system capacity F using Monte Carlo estimation.

***Theorem* IV.2.** *If the N trajectories* $\{(x_1(\ell),\dots,x_T(\ell))\}_{\ell=1}^N$ *are generated i.i.d. by* $\{\phi_1^{\mathrm{CE}},\dots,\phi_T^{\mathrm{CE}}\}$*, then the empirical estimate in* (10) *converges to the system capacity almost surely, i.e.,*

$$\mu_N \xrightarrow{a.s.} \digamma \text{ as } N\to\infty.$$

Note that, in addition to providing a method for estimating the system capacity, the theorem also validates that the entropy of the flexibility feedback sent each time reflects the system's current flexibility. This indicates that, for instance, if the feedback vector is a uniform distribution on U, then the system has maximal flexibility. The proof of Theorem IV.2 can be found in Appendix B. Next, we consider practical computation of the MEF.

V. Approximating Maximum Entropy Feedback via Reinforcement Learning

For real-world applications, computing the maximum entropy feedback (MEF) could be computationally intensive. Thus, instead of computing it precisely, it is desirable to approximate it. In this section, we use model-free reinforcement learning (RL) to generate *aggregator functions* $\psi_1,\dots,\psi_T$. To be more precise, the learned aggregator function $\psi_t:\mathsf{X}\to\mathsf{P}$ 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. For estimating MEF, we use actor-critic architectures with separate policy and value function networks which enable the learning of policies on continuous action and state spaces. Among the actor-critic algorithms [32], soft actor-critic (SAC) [33] is an off-policy maximum entropy deep RL algorithm that maximizes both the expected return and the expected entropy of the policy:

$$J(\phi)=\sum_{t=1}^{T}\mathbb{E}_{(\bar{x}_t,p_t)\sim\rho_\phi}\left[r(\bar{x}_t,p_t)+\alpha\mathbb{H}(\phi(\cdot|x_t))\right] \tag{11}$$

where $\bar{x}_t:=(x_t,u_t)$ and ρ_ϕ denotes the state-action marginals of the trajectory distribution induced by a policy ϕ. To estimate MEF, we need to determine a reward function $r(\bar{x}_t,p_t)$ in (11). We adopt the following reward function that incorporates the constraints and the definition of MEF:

$$r(\bar{x}_t,p_t)=\mathbb{H}(p_t)+\sigma g(\bar{x}_t;\mathsf{X}_t,\mathsf{U}_t) \tag{12}$$

where the first term maximizes the entropy of the probability distribution p_t, based the definition of the MEF in Definition IV.1 and $g(\bar{x}_t)=g(x_t,u_t)$ is a function that rewards the state and action if they satisfy the constraints $x_t\in\mathsf{X}_t$ and $u_t\in\mathsf{U}_t$. The reward function is independent of the price functions. A concrete example of $g(\bar{x}_t)$ is given in Section VII. In the sequel, with the learned MEF, we introduce a closed-loop framework that combines model predictive control (MPC) and RL to coordinate a system operator and an aggregator in real-time.

The training data used for learning the aggregator function is the collection of episodes defined by sequences $(\mathsf{U}_t,\mathsf{X}_t,f_t)_{t=1}^T$. For example, for the EV charging application in Section III, the training data of each episode (day) consists of a collection of historical private vectors $(a(j),d(j),e(j),r(j))$ specified by the users visited the charging station on the corresponding day.

VI. Penalized Predictive Control

Consider the system model in Section II. In this setting, the operator seeks to minimize the cost in an online manner, i.e., at time $t\in[T]$ the operator only knows the objective functions $c_1,\dots,c_t$ and the flexibility feedback $p_1,\dots,p_t$. The task of the operator is to, given the maximum entropy feedback, design a sequence of *operator functions* $\phi_1,\dots,\phi_T$ to generate actions $u_1,\dots,u_T$ that are always feasible with respect to the constraints *and* that minimize the cumulative cost.

A. Key Idea: Maximum entropy feedback as a penalty term

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 centralized controller that balances the MEF and the cost functions.

To further illustrate this point, begin by noting that Corollary IV.1 guarantees that the online agent can always find a feasible action $u\in\mathsf{S}$. 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)-(2d) in the offline optimization can be rewritten as a penalty in the objective of (2a). The next theorem follows.

***Theorem* VI.1.** *The offline optimization* (2a)-(2d) *is equivalent to the following unconstrained minimization for any* $\beta>0$*:*

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

Proof of Theorem VI.1. Using (9), the offline optimization (2a)-(2d) is equivalent to

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

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

$$q(u):=\begin{cases}1/|\mathsf{S}| & \text{if } u\in\mathsf{S}\\ 0 & \text{otherwise}\end{cases}. \tag{15}$$

Further, decomposing the joint distribution $q(u)=\prod_{t=1}^T p_t^*(u_t|u_{<t})$ into the conditional distributions given by (6a)-(6c), the objective function (14) becomes

$$\begin{aligned}&\sum_{t=1}^{T}c_t(u_t)-\beta\log\left(\prod_{t=1}^{T}p_t^*(u_t|u_{<t})\right)\\ =&\sum_{t=1}^{T}\left(c_t(u_t)-\beta\log p_t^*(u_t|u_{<t})\right).\end{aligned}$$

□

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

Data: Sequentially arrived cost functions and MEF
Result: Actions $u = (u_1, \dots, u_T)$
for $t = 1, \dots, T$ **do**
  Choose an action u_t by minimizing:

$$u_t = \phi_t(p_t) := \arg\inf_{u_t \in \mathsf{U}} (c_t(u_t) - \beta \log p_t(u_t | u_{<t})) \quad (16)$$

end
Return u;

Algorithm 2: Penalized Predictive Control (PPC).

B. Algorithm: Penalized Predictive Control via MEF

Our proposed design, termed penalized predictive control (PPC), is a combination of model predictive control (MPC) (*c.f.* [34]), 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 IV). 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*.

We present PPC in Algorithm 2, where we use the following notation. Let $\beta > 0$ be a *tuning parameter* in predictive control to trade-off the flexibility in the future and minimization of the current system cost. The next corollary follows.

***Corollary* VI.1.** *When $p_t = p_t^*$ for all $t \in [T]$, the MEF defined in Definition IV.1, the sequence of actions $u = (u_1, \dots, u_T)$ generated by the PPC in* (16) *always satisfies $u \in \mathsf{S}$ for any tuning parameter $\beta > 0$.*

Proof of Corollary VI.1. The explicit expression in Theorem IV.1 ensures that whenever $p_t^*(u_t | u_{<t}) > 0$, then there is always a feasible sequence of actions in $\mathsf{S}(u_{<t})$. Now, if the tuning parameter $\beta > 0$, then the optimization (16) guarantees that $p_t^*(u_t | u_{<t}) > 0$ for all $t \in [T]$; otherwise, the objective value in (16) is unbounded. Corollary IV.1 guarantees that for any sequence of actions $u = (u_1, \dots, u_T)$, if $p_t^*(u_t | u_{<t}) > 0$ for all $t \in [T]$, then $u \in \mathsf{S}$. Therefore, the sequence of actions u given by the PPC is always feasible. □

C. Framework: Closed-loop control between local and centralized controllers

Given the PPC scheme described above, we can now formally present our online control framework for the distant centralized controller and local controller (defined in Section II). An overview is given in Algorithm 1, where ϕ denotes an operator function and ψ is an aggregator function. 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.

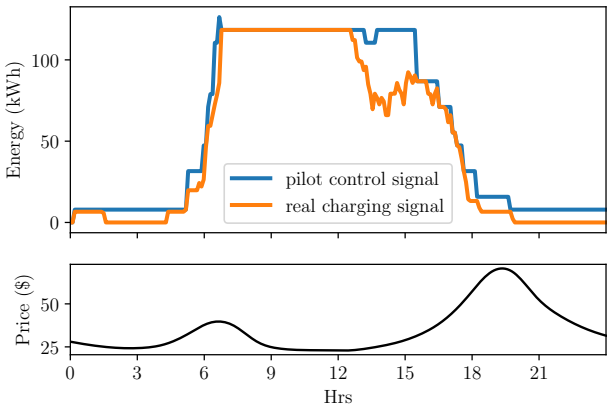

Fig. 3. Pilot control signals and real energy allocated to EVs generated by Algorithm 1.

At each time $t \in [T]$, the local controller first efficiently generates estimated MEF $p_t \in \mathsf{P}$ using an aggregator function ψ_t trained by a reinforcement learning algorithm. After receiving the current MEF p_t and cost function c_t (future w MEF and costs if predictions are available), the centralized controller uses the PPC scheme in Algorithm 2 to generate an action $u_t \in \mathsf{U}$ and sends it back to the local controller. The local controller then updates its state $x_t \in \mathsf{X}$ to a new state x_{t+1} based on the system dynamic in (1) and repeats this procedure again. In the next Section, we use an EV charging example to verify the efficacy of the proposed method.

VII. Numerical Results

In this section, we show our experimental results for online EV charging, introduced in Section III as an example of our system mode (see Section II). We use real EV charging data ACN-Data [23], which is a dataset collected from adaptive EV charging networks (ACNs) at Caltech and JPL. Next, we present the results, with details of the experiments provided in Section VII-B.

A. Experiments

Charging curves. In Figure 3, pilot control and real energy signals are shown. A pilot control signal is an action (corresponding to the trajectory $(u_1, \dots, u_T)$) sent by a system operator in our model and a real energy signal (corresponding to the trajectory $(\sum_j s_1(j), \dots, \sum_j s_T(j))$) is the total energy charged to the EVs. The agent is trained on data collected at Caltech from Nov. 1, 2018 to Dec. 1, 2019 with linear price functions $c_t = 1 - t/24$, where $t \in [0, 24]$ (unit: Hrs) is the time index and tested on Dec. 18, 2019 for JPL with average LMPs on the CAISO (California independent system operator) day-ahead market in 2016, shown on the bottom. The scheduling policy is fixed to be least-laxity-first (LLF). The set of power levels U is a discrete set that contains 60 distinct power levels from 0 kWh to 360 kWh. We use a tuning parameter $\beta = 4000$. The pilot control signals are optimal solutions of (16), which are always bounded from below by the real charging signals,

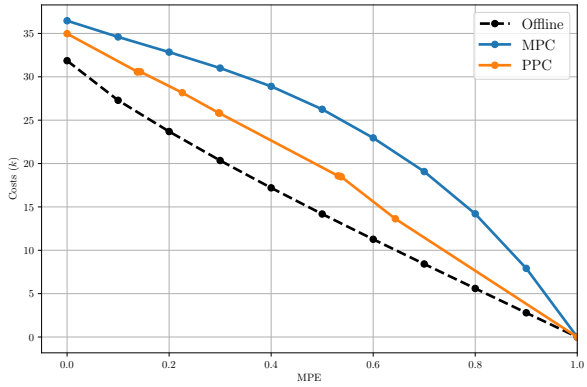

Fig. 4. Cost-energy curves for the offline optimization in (2a)-(2d) (for the example in Section III), MPC (defined in (18)) and PPC (introduced in Section VI-B).

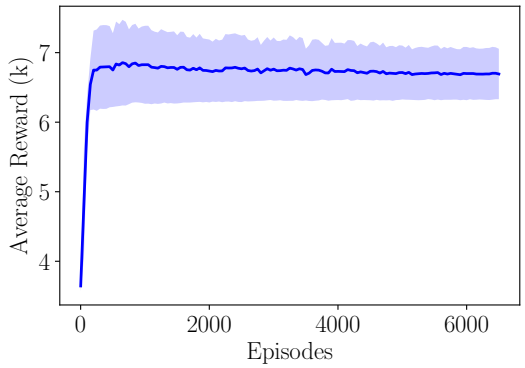

Fig. 5. Average rewards (defined in (12)) in the training stage with a tuning parameter $\beta = 6000$. Shadow region measures the variance.

representing the aggregate charging rates. The figure highlights that, with a suitable choice of tuning parameter, the operator is able to schedule charging at time slots where prices are lower and avoid charging at the peak of prices, as desired.

Cost-energy curves. In Figure 4 we show the changes of the cumulative costs by varying the mean percentage error (MPE) with respect to the undelivered energy defined as

$$\mathsf{MPE} := \sum_{k=1}^{N}\sum_{t=1}^{T}\sum_{j=1}^{n} s_t^{(k)}(j) \Big/ \Big((N\times T)\cdot\sum_{j=1}^{n} e_j\Big), \qquad (17)$$

where e_j is the energy request for each charging session $j \in [n]$; $s_t^{(k)}(j)$ is the energy scheduled to the j-th charging session at time t for the k-th test. There are in total $K = 14$ 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 $0 \leq \mathsf{MPE} \leq 1$. 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 $s_t = (s_t(1),\ldots,s_t(n'))$:

$$\begin{aligned} s_t = \arg\min_{s_t} \sum_{\tau=t}^{t'} c_\tau\Big(\sum_{i=1}^{n'} s_\tau(i)\Big) \text{ subject to: } \qquad (18)\\ s_\tau(i) = 0\ ,\ \tau < a(i),\ i = 1,\ldots,n',\\ s_\tau(i) = 0\ ,\ \tau > d(i),\ i = 1,\ldots,n',\\ \sum_{i=1}^{n'} s_\tau(i) = u_t,\ \tau = t,\ldots,t',\\ \sum_{\tau=1}^{T} s_\tau(i) = \gamma\cdot e(i),\ i = 1,\ldots,n',\\ 0 \leq s_\tau(i) \leq r(i),\ \tau = t,\ldots,t' \end{aligned}$$

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; $\gamma > 0$ 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 (4c) 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 is widely studied [35] and used in EV charging applications [36]. It requires the precise knowledge of a 108-dimensional state vector of 54 chargers at each time step. We observe that with *only* feasibility information, PPC outperforms MPC for all $0 \leq \mathsf{MPE} \leq 1$.

B. Hyper-parameters in experiments

TABLE I
HYPER-PARAMETERS IN THE EXPERIMENTS.

Parameter	Value
System Operator	
Number of power levels $\lvert\mathsf{U}\rvert$	10
Cost functions $c_1,\ldots,c_T$	Average LMPs
Operator function ϕ	Penalized Predictive Control
Tuning parameter β	1×10^3 - 1×10^6
EV Charging Aggregator	
Number of Chargers W	54
State space X	$\mathbb{R}_+^{108}$
Action space	$[0,1]^{10}$
Time interval Δ	12 minutes
Private vector $(a(j),d(j),e(j),r(j))$	ACN-Data [23]
Power rating	150 kW
Scheduling algorithm π	Least Laxity First (LLF)
Laxity	$d_t(j) - e_t(j)/r(j)$
RL algorithm ψ	Soft Actor-Critic (SAC) [33]
Optimizer	Adam [37]
Learning rate	$3\cdot10^{-4}$
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

The detailed parameters used in our experiments are shown in Table I. In the experiments presented in Section VII-A, the state space is $\mathsf{X} = \mathbb{R}_+^{2\times W}$ where W is the total number of charging stations and a state vector for each charging station is $(e_t, [d(j)-t]^+)$, i.e., the remaining energy to be charged and

the remaining charging time if it is being used (see Section III); otherwise the vector is an all-zero vector. The action space is $\mathsf{U} = \{0, 15, 30, \ldots, 150\}$ (unit: kW) with $|\mathsf{U}| = 10$. The action space of the Markov decision process used in the RL algorithm is $[0,1]^{10}$. The outputs of the neural networks are normalized into the probability simplex (space of MEF) P afterwards. We use the following specific reward function for our EV charging scenario, as a concrete example of (12):

$$\begin{aligned} r_{\mathrm{EV}}(\bar{x}_t, p_t) &= \mathbb{H}(p_t) \\ &+\sigma_1 \sum_{i=1}^{n'} ||u_t(i)||_2 \\ &-\sigma_2 \sum_{i=1}^{n'} \left(\mathbf{I}(a(j_i) \le t \le a(j_i) + \Delta) \Big[e(i) - \sum_{t=1}^{T} u_t(i) \Big]_+ \right) \\ &-\sigma_3 \left| \phi_t(p_t) - \sum_{i=1}^{n'} u_t(j) \right| \end{aligned} \tag{19}$$

where σ_1, σ_2 and σ_3 are positive constants; n' is the number of EVs being charged; ϕ_t is the operator function, which is specified by (16); $\mathbf{I}(\cdot)$ denotes an indicator function and $a(j_i)$ is the arrival time of the $i-th$ EV in the charging station with j_i being the index of this EV in the total accepted charging sessions $[n]$. The entropy function $\mathbb{H}(p_t)$ in the first term is a greedy approximation of the definition of MEF (see Definition IV.1). The second term is to further enhance charging performance and the last two terms are realizations of the last term in (12) for constraints (4b) and (4c). Note that The other constraints in the example shown in Section III can automatically be satisfied by enforcing the constraints in the fixed scheduling algorithm π.

With the settings described above, in Figure 5 we show a typical training curve of the reward function in (19). The constants in (19) are chosen as $\sigma_1 = 0.1$, $\sigma_2 = 0.2$ and $\sigma_3 = 2$.

VIII. Concluding remarks

This paper formalizes and studies the closed-loop control framework created by the interaction between a system operator and an aggregator. Our focus is on the feedback signal provided by the aggregator to the operator that summarizes the real-time availability of flexibility among the loads controlled by the aggregator. We present the design of an maximum entropy feedback (MEF) signal based on entropic maximization. We prove a close connection between the MEF signal and the system capacity, and show that when the signal is used the system operator can perform online cost minimization and system capacity estimation while provably respecting the private constraints of the loads controlled by the aggregator. Further, we illustrate the effectiveness of these designs using simulation experiments of an EV charging facility.

There is much left to explore about this MEF signal presented in this work. In particular, computing it is computationally intensive and we use reinforcement learning for approximating the MEF. Improving the learning design and developing other approximations are of particular interest. Further, exploring the use of flexibility feedback for operational objectives beyond cost minimization and capacity estimation is an important goal. Finally, exploring the application of the defined real-time aggregate flexibility in other settings, such as multi-aggregator systems, frequency regulation and real-time pricing, is exciting.

References

[1] D. S. Callaway and I. A. Hiskens, "Achieving controllability of electric loads," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 184–199, 2010.

[2] S. Burger, J. P. Chaves-Ávila, C. Batlle, and I. J. Pérez-Arriaga, "A review of the value of aggregators in electricity systems," *Renewable and Sustainable Energy Reviews*, vol. 77, pp. 395–405, 2017.

[3] H. Hao and W. Chen, "Characterizing flexibility of an aggregation of deferrable loads," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 4059–4064.

[4] H. Hao, B. M. Sanandaji, K. Poolla, and T. L. Vincent, "Aggregate flexibility of thermostatically controlled loads," *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 189–198, 2014.

[5] I. A. Sajjad, G. Chicco, and R. Napoli, "Definitions of demand flexibility for aggregate residential loads," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2633–2643, 2016.

[6] L. Zhao, W. Zhang, H. Hao, and K. Kalsi, "A geometric approach to aggregate flexibility modeling of thermostatically controlled loads," *IEEE Transactions on Power Systems*, vol. 32, no. 6, pp. 4721–4731, 2017.

[7] D. Madjidian, M. Roozbehani, and M. A. Dahleh, "Energy storage from aggregate deferrable demand: Fundamental trade-offs and scheduling policies," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 3573–3586, 2018.

[8] T. Chen, N. Li, and G. B. Giannakis, "Aggregating flexibility of heterogeneous energy resources in distribution networks," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 4604–4609.

[9] N. Sadeghianpourhamami, N. Refa, M. Strobbe, and C. Develder, "Quantitive analysis of electric vehicle flexibility: A data-driven approach," *International Journal of Electrical Power & Energy Systems*, vol. 95, pp. 451–462, 2018.

[10] M. P. Evans, S. H. Tindemans, and D. Angeli, "A graphical measure of aggregate flexibility for energy-constrained distributed resources," *IEEE Transactions on Smart Grid*, 2019.

[11] A. Bernstein, J.-Y. Le Boudec, M. Paolone, L. Reyes-Chamorro, and W. Saab, "Aggregation of power capabilities of heterogeneous resources for real-time control of power grids," in *2016 Power Systems Computation Conference (PSCC)*. IEEE, 2016, pp. 1–7.

[12] G. Wenzel, M. Negrete-Pincetic, D. E. Olivares, J. MacDonald, and D. S. Callaway, "Real-time charging strategies for an electric vehicle aggregator to provide ancillary services," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5141–5151, 2017.

[13] H. Hao, Y. Lin, A. S. Kowli, P. Barooah, and S. Meyn, "Ancillary service to the grid through control of fans in commercial building hvac systems," *IEEE Transactions on smart grid*, vol. 5, no. 4, pp. 2066–2074, 2014.

[14] S. P. Meyn, P. Barooah, A. Bušić, Y. Chen, and J. Ehren, "Ancillary service to the grid using intelligent deferrable loads," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 2847–2862, 2015.

[15] A. Subramanian, M. Garcia, A. Dominguez-Garcia, D. Callaway, K. Poolla, and P. Varaiya, "Real-time scheduling of deferrable electric loads," in *2012 American Control Conference (ACC)*. IEEE, 2012, pp. 3643–3650.

[16] A. Subramanian, M. J. Garcia, D. S. Callaway, K. Poolla, and P. Varaiya, "Real-time scheduling of distributed resources," *IEEE Transactions on Smart Grid*, vol. 4, no. 4, pp. 2122–2130, 2013.

[17] D. Papadaskalopoulos, G. Strbac, P. Mancarella, M. Aunedi, and V. Stanojevic, "Decentralized participation of flexible demand in electricity markets—part ii: Application with electric vehicles and heat pump systems," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 3667–3674, 2013.

[18] T. Li, S. H. Low, and A. 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*, 2020, pp. 279–292.

[19] S. Wang, S. Bi, and Y. J. A. Zhang, "Reinforcement learning for real-time pricing and scheduling control in ev charging stations," *IEEE Transactions on Industrial Informatics*, 2019.

[20] Y. Wang, X. Lin, and M. Pedram, "A near-optimal model-based control algorithm for households equipped with residential photovoltaic power generation and energy storage systems," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 1, pp. 77–86, 2015.

[21] B. J. Claessens, D. Vanhoudt, J. Desmedt, and F. Ruelens, "Model-free control of thermostatically controlled loads connected to a district heating network," *Energy and Buildings*, vol. 159, pp. 1–10, 2018.

[22] B. Chen, W. Yao, J. Francis, and M. Berges, "Learning a distributed control scheme for demand flexibility in thermostatically controlled loads," *arXiv preprint arXiv:2007.00791*, 2020.

[23] Z. J. Lee, T. Li, and S. 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*. ACM, 2019, pp. 139–149.

[24] L. Gan, U. Topcu, and S. H. Low, "Optimal decentralized protocol for electric vehicle charging," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 940–951, 2012.

[25] E. C. Kara, J. S. Macdonald, D. Black, M. Bérges, G. Hug, and S. Kiliccote, "Estimating the benefits of electric vehicle smart charging at non-residential locations: A data-driven approach," *Applied Energy*, vol. 155, pp. 515–525, 2015.

[26] X. Chen, E. Dall'Anese, C. Zhao, and N. Li, "Aggregate power flexibility in unbalanced distribution systems," *arXiv preprint arXiv:1812.05990*, 2018.

[27] M. Marzband, A. Sumper, J. L. Domínguez-García, and R. Gumara-Ferret, "Experimental validation of a real time energy management system for microgrids in islanded mode using a local day-ahead electricity market and minlp," *Energy Conversion and Management*, vol. 76, pp. 314–322, 2013.

[28] P. Siano and D. Sarno, "Assessing the benefits of residential demand response in a real time distribution energy market," *Applied Energy*, vol. 161, pp. 533–551, 2016.

[29] I. Csiszár, "Axiomatic characterizations of information measures," *Entropy*, vol. 10, no. 3, pp. 261–273, 2008.

[30] J. Aczél, B. Forte, and C. T. Ng, "Why the shannon and hartley entropies are 'natural'," *Advances in applied probability*, vol. 6, no. 1, pp. 131–146, 1974.

[31] M. Simonovits, "How to compute the volume in high dimension?" *Mathematical programming*, vol. 97, no. 1-2, pp. 337–374, 2003.

[32] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE transactions on systems, man, and cybernetics*, no. 5, pp. 834–846, 1983.

[33] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *arXiv preprint arXiv:1801.01290*, 2018.

[34] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.

[35] U. Rosolia, X. Zhang, and F. Borrelli, "Data-driven predictive control for autonomous systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 259–286, 2018.

[36] Z. J. Lee, D. Chang, C. Jin, G. S. Lee, R. Lee, T. Lee, and S. H. Low, "Large-scale adaptive electric vehicle charging," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2018, pp. 1–7.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

APPENDIX A
PROOF OF THEOREM IV.1 AND COROLLARY IV.1

Proof of Theorem IV.1. We prove the statement by induction on T. It is straightforward to verify the results when $T=1$. We suppose the theorem is true when $T=m$. Suppose $T=m+1$. Let

$$F(u_1) := \max_{p_2,\ldots,p_T} \sum_{t=2}^{T} \mathbb{H}(U_t|U_{<t})$$

denote the optimal value corresponding to the time horizon $t \in [T]\backslash\{1\}$, conditioning on u_1. We have

$$F = \max_{p_1} \sum_{u_1 \in \mathsf{U}} p_1(u_1) F(u_1) + \mathbb{H}(p_1).$$

By the induction hypothesis, $F(u_1) = \log|\mathsf{S}(u_1)|$. Therefore,

$$\begin{aligned} F &= \max_{p_1} \sum_{u_1 \in \mathsf{U}} p_1(u_1) \log|\mathsf{S}(u_1)| + \mathbb{H}(p_1) \\ &= \max_{p_1} \sum_{u_1 \in \mathsf{U}} p_1(u_1) \log\left(\frac{|\mathsf{S}(u_1)|}{p_1(u_1)}\right) \end{aligned}$$

whose optimizer p_1^* satisfies (7) and we get $F = \log|\mathsf{S}|$. The theorem follows by finding the optimal conditional distributions $p_2^*, \ldots, p_T^*$ inductively. □

APPENDIX B
PROOF OF THEOREM IV.2

Proof of Theorem IV.2. Suppose that N action trajectories $\{u(1),\ldots,u(N)\}$ are sampled i.i.d. according to the maximum entropy feedback. Equivalently, for all $\ell = 1,\ldots,N$, the entropy of the maximum entropy feedback $p_t^*(u_{<t}(\ell))$ can be written as the following conditional entropy $\mathbb{H}(p_t(\cdot|u_{<t}(\ell))) = \mathbb{H}(U_t|U_{<t} = u_{<t})$, where each $U_t \in \mathsf{U}$ is a random signal drawn according to $p_t^*(u_{<t}(\ell))$. We claim that, if the random power signal U_t is sampled according to $p_t^*(\cdot|u_{<t})$ conditioned on previous power signals $u_{<t} = (u_1,\ldots,u_{t-1})$ for all $t \in [T]$, then the accumulated flexibility over $t \in [T]$ is equal to the system capacity F in expectation,

$$\mathbb{E}_V\left[\sum_{t=1}^{T} \mathbb{H}(U_t|U_{<t} = V_{<t})\right] = F \tag{20}$$

where the expectation is taken over the randomness of the action trajectory V that has the same distribution as U. The equality in (20) follows by noticing that the left hand side equals to the objective function in (6a), with the flexibility feedback there at each time $t \in [T]$ being optimal. Noting that the expectation in (20) equals to F, the law of large numbers (LLN) implies the theorem. □