Proceedings of Machine Learning Research vol vvv:1–15, 2026

Improving EV Aggregate Flexibility with End-to-End Learning

Apoorva Thanvantri ATHANVAN@CALTECH.EDU
California Institute of Technology

Christopher Yeh CYEH@CALTECH.EDU
California Institute of Technology

Nicolas Christianson CHRISTIANSON@STANFORD.EDU
Stanford University

Adam Wierman ADAMW@CALTECH.EDU
California Institute of Technology

Abstract

As the adoption of electric vehicles (EVs) rises, meeting their charging demand efficiently while continuing to ensure reliable power grid operation has become increasingly challenging. One promising avenue for more efficient integration of EV charging demands is leveraging their flexibility. To facilitate this, aggregators—entities that pool energy resources into a single market participant—must combine the constraints encoding each EV's charging flexibility into an aggregate flexibility set. Computing this set exactly is computationally intractable, motivating the development of methods to approximate this set. However, current methods for approximating this aggregate flexibility set are either unreliable—in that they may contain infeasible power schedules which could lead to grid instability—or they are overly conservative, and may neglect regions of the true aggregate set which are important for optimizing grid-relevant costs. Motivated by these limitations, we develop a novel approach for learning inner approximations of aggregate flexibility sets using Input-Convex Neural Networks (ICNNs). In particular, we propose to train approximate flexibility sets parametrized by ICNNs to minimize a *decision cost*, while incorporating a feasibility projection at each step of training to ensure the reliability of the learned set. We experimentally validate our methodology on the problem of learning aggregate flexibility sets for a peak power minimization task with real-world load data, showing that our approach enables better performance than decision-agnostic methods while guaranteeing reliability.

Keywords: electric vehicle flexibility, Minkowski sum, input convex neural networks, differentiable optimization

1. Introduction

With the growing adoption of electric vehicles (EVs), the demand for energy is increasing, presenting new challenges for power grid operators and participants. EVs have unique energy constraints and flexible charging schedules, as they are often plugged in longer than necessary to reach full charge. To efficiently distribute energy, aggregators—entities that pool multiple energy resources into a single market participant—must combine these individual flexibility sets into one aggregate flexibility set while still accounting for the physical constraints and operational requirements of

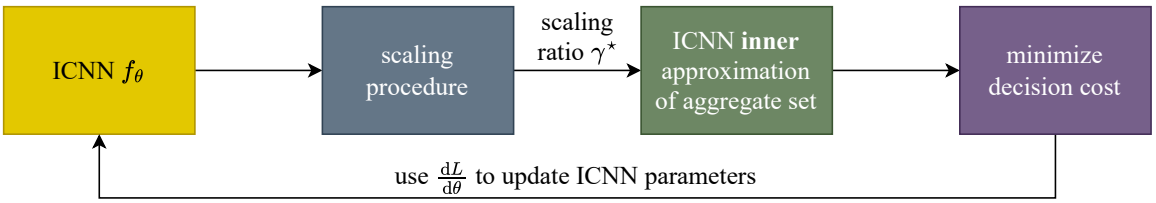

Figure 1: A high-level schematic of our methodology.

each EV. In addition to EVs, aggregators can also pool energy from various other distributed energy resources (DERs), which can lead to increased flexibility. Increased flexibility benefits the aggregator by enabling cost savings through lower energy prices while also facilitating the use of cleaner, lower-carbon energy sources.

Since aggregators participate in the market as a single entity, they must submit a description of their aggregate flexibility set to the power system operator. However, computing this set exactly is computationally intractable (Tiwary, 2008; Taha et al., 2024). To address this, previous approaches have focused on obtaining tractable inner approximations, but these methods often face significant challenges. For example, techniques like projecting convex polytopes or taking the union of hyperrectangles can lead to overly conservative approximations which under-utilize available flexibility or become computationally intractable, thereby limiting scalability. Machine learning models offer the potential to compute more expressive approximations while also focusing their learning capacity on regions of the flexibility set that are crucial for optimal performance, enabling better scalability. However, it is vital to ensure that they do not classify infeasible schedules as feasible ones, as this could lead to grid instability or inefficient energy allocation.

To overcome these limitations, we develop a new approach for obtaining data-driven, machine-learned inner approximations of these complex flexibility sets that are both efficient to compute and decision-focused (e.g., trained to reduce downstream cost) (Donti et al., 2017). Our main contributions can be summarized as follows:

1. We introduce a decision-focused inner approximation of the aggregate EV flexibility set based on the 0-sublevel set of an input-convex neural network (ICNN). The ICNN-based parameterization is more general than previous approaches, and our decision-focused learning objective improves performance over decision-agnostic approximations.
2. To overcome the computational scalability challenges with computing the exact decision-focused gradient, we introduce a heuristic inspired by projected gradient descent that leverages the structure of the ICNN to significantly speed up model training.
3. We test our method in a case study on peak power minimization and validate that our method improves upon existing methods.

1.1. Related Work

Several methods exist to approximate the aggregate flexibility set. Methods as in Hao and Chen (2014); Hao et al. (2015); Nayyar et al. (2013) use the parameters of each individual set to create an approximation of the aggregate flexibility set, yet this does not work well with heterogeneous

sets. Other works such as Müller et al. (2019); Zhao et al. (2017); Nazir et al. (2018) use certain classes of convex polytopes to create inner approximations of each individual set which can be much more accurate. However, this approximation may not follow guidelines needed for independent system operators (ISOs) and may not scale to larger dimensions. Mukhi et al. (2025) finds an efficient exact representation of the aggregate set, but optimizing over this set involves the usage of special algorithms, which may be difficult for the ISOs to do. Panda and Tindemans (2024) creates an aggregation for which the number of parameters is linear in the number of timesteps, but imposes many additional constraints on the setting, yielding additional conservativeness. Taha et al. (2024) creates an approximation by solving for an affine-transformation of each individual set, optimizing for a volume-maximizing approximation. To ensure that their polytope approximation is an inner approximation, Taha et al. (2024) relies on the approach of Sadraddini and Tedrake (2019), which we also adopt in our work. However, this approximation is not decision-focused, which potentially leads to suboptimal downstream performance. Taheri et al. (2022) creates a data-driven approximation for both convex and non-convex device models, but this work is not an inner approximation, meaning that it will contain potentially infeasible power schedules.

Unlike aggregate flexibility set approximations described above, which are agnostic to the downstream task, we are interested in approximating the aggregate flexibility set in a way that accounts for how it is used for downstream decision-making. We draw inspiration from the literature on decision-focused learning (Mandi et al., 2024; Sadana et al., 2025), also known as end-to-end learning (Donti et al., 2017) or predict-then-optimize (Elmachtoub and Grigas, 2022). Whereas these works generally focus on learning a machine learning regression model, we are interested in learning a set. In this vein, perhaps the most similar work to ours is Christianson et al. (2024) which uses decision-focused learning to find a tractable inner approximation for the power grid contingency screening problem, using a differentiable convex optimization-based scaling procedure to enforce the inner approximation property. In contrast, we are interested in EV aggregate flexibility sets, and we propose a training method inspired by projected gradient descent to enable more scalable training than their approach.

2. Model and Preliminaries

In this section, we begin by describing the model we use for EV flexibility sets and the problem of their aggregation. Then, we introduce the notion of input-convex neural networks (ICNNs) that we use to approximate these sets.

Notation. Define $[N] := \{1, \dots, N\}$. We write (A, B) to denote vertical concatenation of matrices A and B—i.e., $(A, B) = [A^\top, B^\top]^\top$. For a scalar α and a set $S \subseteq \mathbb{R}^n$, $\alpha S = \{\alpha x \mid x \in S\}$.

Definitions. An H-polytope $\mathbb{P}_H$ is a polytope represented by the intersection of its halfspaces, e.g., $\mathbb{P}_H = \{x \mid Ax \leq b\}$. An AH-polytope refers to an affine transformation of an H-polytope, e.g., $\mathbb{P}_{AH} = C\mathbb{P} + d = \{Cx + d \mid Ax \leq b\}$.

2.1. EV Flexibility Sets

Formally, we consider the problem of approximating the aggregate flexibility set for N EVs over T time periods, following the setting of Taha et al. (2024). Assume that each time period has equal length δ. Let $u_i(t)$ denote the charging rate of EV $i \in \mathcal{N} := [N]$ at time $t \in \mathcal{T} := \{0, \dots, T-1\}$,

and define $u_i := (u_i(0), \ldots, u_i(T-1)) \in \mathbb{R}^T$ as the charging profile. The net energy delivered to EV i is

$$x_i(t+1) = x_i(t) + u_i(t)\delta, \quad t \in \mathcal{T}, \tag{1}$$

where $x_i(0) = 0$. The net energy profile can then be defined as $x_i := (x_i(1), \ldots, x_i(T)) \in \mathbb{R}^T$. This can be rewritten as $x_i = Lu_i$, where $L \in \mathbb{R}^{T \times T}$ is a lower triangular matrix with $L_{ij} = \delta$ for all $j \leq i$.

The set of feasible charging profiles for an individual EV $i \in \mathcal{N}$ is called its individual flexibility set, defined as

$$\mathbb{U}_i := \{u \in \mathbb{R}^T \mid u \in [\underline{u}_i, \overline{u}_i],\ Lu \in [\underline{x}_i, \overline{x}_i]\}, \tag{2}$$

where $\underline{u}_i, \overline{u}_i \in \mathbb{R}^T$ are the minimum and maximum power limits on the charging profile, and $\underline{x}_i, \overline{x}_i \in \mathbb{R}^T$ are the minimum and maximum energy limits on the net energy profile. $\mathbb{U}_i$ can be represented as a H-polytope:

$$\mathbb{U}_i = \{u \in \mathbb{R}^T \mid Hu \leq h_i\} \tag{3}$$

with $H := (L, -L, I_T, -I_T) \in \mathbb{R}^{4T \times T}$ and $h_i := (\overline{x}_i, -\underline{x}_i, \overline{u}_i, -\underline{u}_i) \in \mathbb{R}^{4T}$.

Suppose that an aggregator is tasked with operating the N EVs' charging profiles. To the power grid operator, the aggregator is a single entity, and so it must represent its feasible power schedules as a single set. Formally, this *aggregate flexibility set* is the Minkowski sum of the individual EVs' flexibility sets, defined as

$$\mathbb{U} = \sum_{i \in \mathcal{N}} \mathbb{U}_i := \left\{ u \in \mathbb{R}^T \;\middle|\; u = \sum_{i \in \mathcal{N}} u_i,\ u_i \in \mathbb{U}_i \right\}. \tag{4}$$

$\mathbb{U}$ is a convex set which represents all possible net charge and discharge decisions, obtained from summing the individual decisions given by each EV's charging profile. However, computing the Minkowski sum $\mathbb{U}$ is NP-hard in general (Tiwary, 2008). As such, much recent work has proposed methods to approximate this set, often using specific parametrizations such as AH-polytopes (Taha et al., 2024) and ellipsoids (Taheri et al., 2022). In the next section, we will describe our approach to approximating flexibility sets with input-convex neural networks.

2.2. Representing Flexibility Sets with Input-Convex Neural Networks

Because computing $\mathbb{U}$ exactly is generally intractable, we seek a tractable approximation. Specifically, we seek an *inner* approximation so that every element of the approximation is a feasible aggregate EV charging profile. Whereas Taha et al. (2024) uses a specific volume-maximizing inner approximation based on a transformation of a fixed base polytope, we aim to *learn* a more general convex set as our inner approximation. Like Taha et al. (2024), for the remainder of the paper we will assume that $\mathbb{U}$ is full dimensional (i.e., has nonempty interior).

We specifically propose to represent inner approximations of an aggregate flexibility set via a sublevel set of a learned convex function. Let $f_\theta : \mathbb{R}^T \to \mathbb{R}$ be an *input-convex neural network* (ICNN,

Amos et al. (2017)) with $k \in \mathbb{N}$ layers:

$$f_\theta(x) = W_k^{(z)} z_k + W_k^{(x)} x + b_k \tag{5a}$$
$$z_{i+1} = \text{ReLU}(W_i^{(z)} z_i + W_i^{(x)} x + b_i), \qquad i = 0, \dots, k-1 \tag{5b}$$
$$z_0 = 0 \tag{5c}$$

where $W_0^{(z)} = 0$, and we impose the constraint $W_{1:k}^{(z)} \geq 0$ to enforce convexity in x. For simplicity of notation, we assume that all hidden layers $z_1, \dots, z_k$ have the same dimension D.

We call the 0-sublevel set of f_θ,

$$\mathcal{F}_\theta := \{x \in \mathbb{R}^T \mid f_\theta(x) \leq 0,\ p^{\text{lower}} \leq x \leq p^{\text{upper}}\},$$

the *feasibility set* associated with the ICNN. We introduce the lower and upper bounds $p^{\text{lower}}, p^{\text{upper}}$ to enforce that $\mathcal{F}_\theta$ is bounded. The learnable parameters of the ICNN feasibility set are thus

$$\theta = (W_{0:k}^{(x)}, W_{1:k}^{(z)}, b_{0:k}, p^{\text{lower}}, p^{\text{upper}}).$$

$\mathcal{F}_\theta$ will be the inner approximation of the aggregate flexibility set which we seek to learn. Because ICNNs are universal convex function approximators (Chen et al., 2018, Theorem 1), $\mathcal{F}_\theta$ can, in principle, approximate any aggregate flexibility set arbitrarily well, given an ICNN of sufficient depth and width.

Notably, $\mathcal{F}_\theta$ can be represented as an AH-polytope: let $\mathbf{z} := (z_1, \dots, z_k) \in \mathbb{R}^{kD}$, and define

$$\mathbb{P}_x = \left\{ \begin{bmatrix} x \\ \mathbf{z} \end{bmatrix} \;\middle|\; C \begin{bmatrix} x \\ \mathbf{z} \end{bmatrix} \leq d \right\}, \tag{6}$$

where $C \in \mathbb{R}^{(2kD+2T+1)\times(T+kD)}, d \in \mathbb{R}^{2kD+2T+1}$ encode the following constraints (see Appendix A for their full specification):

$$z_i \geq W_{i-1}^{(z)} z_{i-1} + W_{i-1}^{(x)} x + b_{i-1} \qquad \forall i = 1, \dots, k \tag{7a}$$
$$z_i \geq 0 \qquad \forall i = 1, \dots k \tag{7b}$$
$$0 \geq W_k^{(z)} z_k + W_k^{(x)} x + b_k \tag{7c}$$
$$x \leq p^{\text{upper}} \tag{7d}$$
$$x \geq p^{\text{lower}} \tag{7e}$$

Then $\mathcal{F}_\theta = A\mathbb{P}_x$, where $A \in \mathbb{R}^{T\times(T+kD)}$ is the matrix that projects onto the first T elements of $p \in \mathbb{P}_x$. Since there is a bijection between θ and (A, C, d), we may overload the notation θ to also refer to (A, C, d).

Thus far, we have described our parametrization for the approximate flexibility set $\mathcal{F}_\theta$. However, to ensure that this set enables reliable operation, we must ensure that it constitutes an inner approximation to the true aggregate flexibility set $\mathbb{U}$. In the next section, we will describe how to scale the ICNN-parametrized set to ensure it satisfies this property. Then, in Section 4, we will discuss the objective and training procedure used to learn the parameters of $\mathcal{F}_\theta$.

3. Scaling Procedure for Feasibility

To ensure that the predicted flexibility set $\mathcal{F}_\theta$ is reliable—i.e., to make sure that it only contains (net) power schedules which are feasible, in the sense that they correspond to the sum of feasible EV power schedules—we require that $\mathcal{F}_\theta \subseteq \mathbb{U}$. Equivalently, it must be the case that $f_\theta(x) \leq 0$ and $p^{\text{lower}} \leq x \leq p^{\text{upper}}$ implies $x \in \mathbb{U}$.

To ensure this containment while preserving downstream decision-making performance, we propose a method for scaling $\mathcal{F}_\theta$ that is based on the scaling procedure of Christianson et al. (2024) in the setting of power grid contingency screening. Specifically, we aim to find the largest scaling factor $\gamma \in \mathbb{R}_+$ such that the ICNN feasibility set $\mathcal{F}_\theta$, when scaled by γ, is fully contained in the Minkowski sum of the individual EV flexibility sets:

$$\gamma^{\text{opt}} := \underset{\gamma \in \mathbb{R}_+}{\arg\max}\ \gamma \quad \text{s.t.} \quad \gamma \mathcal{F}_\theta \subseteq \mathbb{U}'. \tag{8}$$

Here, we define $\mathbb{U}' := \mathbb{U} - \Delta$, where Δ is some translation applied to $\mathbb{U}$ pointwise such that $0 \in \operatorname{int} \mathbb{U}'$. It is clear that γ^{opt} will always exist and be strictly positive since $\mathbb{U}$ is full dimensional and bounded and $\mathcal{F}_\theta$ is bounded. In particular, since $\mathbb{U}$ is full dimensional (i.e., has nonempty interior), there is some point $\Delta \in \operatorname{int} \mathbb{U}$, meaning $0 \in \operatorname{int} \mathbb{U}'$. As a result, $\mathbb{U}'$ contains an ϵ-neighborhood of the origin for some $\epsilon > 0$, and so by choosing $\gamma > 0$ sufficiently small, $\gamma \mathcal{F}_\theta \in \mathbb{U}'$.

Note, however, that computing the Minkowski sum $\mathbb{U}$ is intractable. Therefore, in contrast to the contingency screening setting considered by Christianson et al. (2024), it is not possible in general to solve the containment problem in (8) exactly. To scale the ICNN flexibility set in a tractable manner, we will make use of the following result of Sadraddini and Tedrake (2019), which allows us to replace the constraint $\gamma \mathcal{F}_\theta \in \mathbb{U}'$ with a conservative sufficient condition.

Theorem 1 (from Sadraddini and Tedrake (2019)) *Suppose polytopes $\mathbb{A}$ and $\mathbb{B}_i$ for $i \in [Q]$ have the form $\mathbb{A} = \bar{a} + A\mathbb{P}_a$ and $\mathbb{B}_i = \bar{b} + B\mathbb{P}_i$ where $\mathbb{P}_a = \{x \mid H_a x \leq h_a\}$ and $\mathbb{P}_i = \{y \mid H_i y \leq h_i\}$. Define $\mathbb{B} = \sum_{i=1}^{Q} \mathbb{B}_i$. We can verify that*

$$\mathbb{A} \subseteq \mathbb{B}$$

if there exist Λ_i, β_i, and Γ_i, for $i \in [Q]$, such that the following conditions hold:

$$\Lambda_i H_a = H_i \Gamma_i \qquad \forall i \in [Q] \tag{9a}$$

$$\Lambda_i h_a \leq h_i + H_i \beta_i \qquad \forall i \in [Q] \tag{9b}$$

$$\sum_{i=1}^{Q} \bar{b} - B\beta_i = \bar{a} \tag{9c}$$

$$\sum_{i=1}^{Q} B\Gamma_i = A. \tag{9d}$$

Theorem 1 gives a way of certifying whether a given AH-polytope is contained in the Minkowski sum of other AH-polytopes. In our setting, we can apply this theorem by posing a "scaling form" of the result, replacing the constraint in (8) with the conditions in (9) to obtain the following scaling

problem:

$$\gamma^\star := \underset{\gamma,\Lambda_i,\beta_i,\Gamma_i,b}{\arg\max} \quad \gamma \tag{10a}$$

$$\text{s.t.} \quad \Lambda_i C = H\Gamma_i \qquad \forall i \in [N] \tag{10b}$$

$$\Lambda_i d \leq h_i + H\beta_i \qquad \forall i \in [N] \tag{10c}$$

$$\sum_{i=1}^{N} -\beta_i = \mathbf{0} \tag{10d}$$

$$\sum_{i=1}^{N} \Gamma_i = \gamma A, \tag{10e}$$

where $\Lambda_i \in \mathbb{R}_+^{4T\times(2kD+2T+1)}$, $\beta_i \in \mathbb{R}^T$, and $\Gamma_i \in \mathbb{R}^{T\times(kD+T)}$ $\forall i \in 1,\dots,N$. Because the conditions in (9) give a sufficient (but in general not necessary) condition for the containment property to hold, the solution $\gamma^\star$ to (10) is guaranteed to be a lower bound on the original scaling factor γ^{opt}.

We now show how to use $\gamma^\star$ to scale the parameters of the ICNN in order to produce a reliable flexibility set. Since we can (without loss of generality) apply a translation to $\mathcal{F}_\theta$ so it contains the origin, we can assert that $\gamma_1\mathcal{F}_\theta \subseteq \gamma_2\mathcal{F}_\theta$ for any scaling factors $0 \leq \gamma_1 \leq \gamma_2$. Thus, since $\gamma^\star \leq \gamma^{\text{opt}}$, we have

$$\{\gamma^\star Ax \mid Cx \leq d\} \;=\; \gamma^\star\mathcal{F}_\theta \;\subseteq\; \gamma^{\text{opt}}\mathcal{F}_\theta \;\subseteq\; \mathbb{U}'. \tag{11}$$

Therefore, by replacing A with $\gamma^\star A$, we can construct an ICNN $f_\theta^\star$ whose corresponding feasibility set $\gamma^\star\mathcal{F}_\theta$ forms an inner approximation of $\mathbb{U}'$.

4. Enforcing Reliability During Training

Thus far, we have described how the 0-sublevel set $\mathcal{F}_\theta$ of an ICNN can parameterize a convex set, and how to scale the parameters of the ICNN such that $\mathcal{F}_\theta$ is an inner approximation of the true aggregate flexibility set $\mathbb{U}$. In this section, we finally introduce how we train the ICNN parameters to optimize for a downstream objective while maintaining the inner approximation guarantee.

In practice, the (translated) aggregate flexibility set $\mathbb{U}'$ typically forms the feasible set of an optimization problem

$$u^{\text{opt}}(l) := \Delta + \underset{u}{\arg\min}\, c(u,l) \;\text{ s.t. }\; u \in \mathbb{U}', \qquad c^{\text{opt}}(l) := c(u^{\text{opt}}(l), l)$$

with a decision cost function c parameterized by context l. For example, if $l \in \mathbb{R}^T$ is a load profile of a set of households, then $c(u,l) = \|u+l\|_\infty$ describes the peak power minimization problem (Taha et al., 2024).

Because $\mathbb{U}'$ is generally intractable to compute, we replace the constraint $u \in \mathbb{U}'$ with the inner approximation $u \in \gamma^\star\mathcal{F}_\theta$:

$$u_\theta(l) := \Delta + \underset{u}{\arg\min}\, c(u,l) \;\text{ s.t. }\; u \in \gamma^\star\mathcal{F}_\theta, \qquad c_\theta(l) := c(u_\theta(l), l).$$

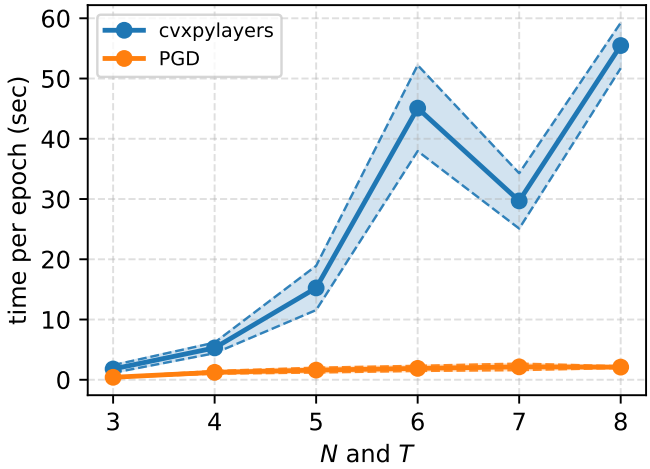

Figure 2: The mean time over 50 epochs for N vehicles and T timesteps is plotted with ± 1 standard deviation shaded in. As N and T increase, `cvxpylayers` takes substantially longer compared to PGD.

Given a dataset of context vectors $\{l^{(i)}\}_{i=1}^{M}$, we then define the average optimality gap over the context distribution as

$$L(\theta) := \mathbb{E}_y[c_\theta(l) - c^{\text{opt}}(l)] = \frac{1}{M}\sum_{i=1}^{M} c_\theta(l^{(i)}) - c^{\text{opt}}(l^{(i)})$$

To minimize $L(\theta)$ with gradient descent, we apply the chain rule to calculate the decision-focused gradient

$$\frac{\mathrm{d}L}{\mathrm{d}\theta} = \frac{1}{M}\sum_{i=1}^{M} \frac{\partial c(u_\theta(l), l)}{\partial u} \cdot \left(\frac{\partial u_\theta(l^{(i)})}{\partial \gamma} \cdot \frac{\partial \gamma^\star}{\partial \theta} + \frac{\partial u_\theta(l^{(i)})}{\partial \theta} \right)$$

where $\frac{\partial u_\theta(l^{(i)})}{\partial \theta}$ refers to the derivative of $u_\theta(l)$ with respect to θ while keeping $\gamma^\star$ fixed.

The three terms $\frac{\partial u_\theta(l^{(i)})}{\partial \gamma}$, $\frac{\partial \gamma^\star}{\partial \theta}$, and $\frac{\partial u_\theta(l^{(i)})}{\partial \theta}$ are all derivatives of the gradient of the solution of a convex optimization problem with respect to some parameter(s) of the problem. In order to compute these terms, we leverage differentiable convex optimization layers (`cvxpylayers`) from Agrawal et al. (2019).

However, as shown in Figure 2 (blue) this approach of using differentiable convex optimization layers to compute $\frac{\partial \gamma^\star}{\partial \theta}$ does not scale to a large number of EVs N and timesteps T. This is because solving for the scaling factor γ^* requires solving the linear program (10), whose decision variables grow quadratically in the time horizon T and linearly in the number of EVs N. As such, existing differentiable convex optimization layers like `cvxpylayers`, which are implemented only on CPU, cannot effectively scale to large problem instances, and take up to one minute per epoch of training even on relatively small-scale instances with 8 EVs and 8 timesteps.

As a workaround, we propose a heuristic approach inspired by projected gradient descent where we alternate between enforcing the inner approximation property via scaling and performing a decision-focused gradient update. For each minibatch during training, we first solve for $\gamma^\star$ using cuOpt, a

GPU-accelerated solver for linear programming from NVIDIA (2025). We use this $\gamma^\star$ to directly update θ to enforce that $\mathcal{F}_\theta$ is an inner approximation of $\mathbb{U}'$. Specifically, for $\gamma^\star > 0$, we have

$$\begin{aligned}\gamma^\star \mathcal{F}_\theta := \{\gamma^\star x \mid f_\theta(x) \le 0\} \subseteq \mathbb{U}' &\iff \{x \mid f_\theta(x/\gamma^\star) \le 0\} \subseteq \mathbb{U}' \\ &\iff \{x \mid \gamma^\star f_\theta(x/\gamma^\star) \le 0\} \subseteq \mathbb{U}'.\end{aligned}$$

(For simplicity of notation, the $p^{\text{lower}} \le x \le p^{\text{upper}}$ bound is omitted from the expressions above.)

Then, we observe

$$\gamma^\star f_\theta(x/\gamma^\star) \;=\; \gamma^\star(W_k^{(z)} z_k + W_k^{(x)} x_k/\gamma^\star + b_k) \;=\; W_k^{(z)}(\gamma^\star z_k) + W_k^{(x)} x_k + \gamma^\star b_k,$$

where we can recursively apply $\gamma^\star z_i = W_{i-1}^{(z)}(\gamma^\star z_{i-1}) + W_{i-1}^{(x)} x_{i-1} + \gamma^\star b_{i-1}$ for $i = k, k-1, \ldots, 1$. This observation suggests the following update rule to our ICNN bias parameters suffices to ensure that $\mathcal{F}_\theta \subseteq \mathbb{U}'$:

$$b_i \leftarrow \gamma^\star b_i \qquad\qquad \forall i = 0, 1, \ldots, k.$$

After applying this update, we then perform the gradient step, but now we treat $\gamma^\star$ as fixed:

$$\theta \leftarrow \theta - \eta \cdot \frac{1}{M} \sum_{i=1}^{M} \frac{\partial c(u_\theta(l), l)}{\partial u} \cdot \frac{\partial u_\theta(l^{(i)})}{\partial \theta}$$

where η is the learning rate. This approach, denoted "PGD" in Figure 2, is substantially faster compared to computing $\frac{\partial \gamma^\star}{\partial \theta}$ using `cvxpylayers`.

5. Experimental Evaluation

We test the performance of our method on simulated EV flexibility sets following the same parameters and procedure as Taha et al. (2024) (see Appendix C for the parameters used for EV flexibility set generation). Note that we do not evaluate the performance of the `cvxpylayers` approach described in the previous section, as the differentiable convex optimization layers do not converge to solutions on the larger-scale instances we examine in this section.

We studied performance on the peak power minimization problem:

$$\min \|u + l\|_\infty \text{ s.t. } u \in \mathbb{U}$$

where l refers to the aggregate load profile for a set of households. As in Taha et al. (2024), our load profiles are taken from Pecan Street Dataport (Street (2018)) and consist of the electricity consumption for $N = 25$ households over a $T = 18$ hour time horizon for a 6-month span in 2019. We sum the load profiles across the individual households to obtain $M = 184$ aggregate daily load profiles. We split the load profiles into training, validation, and testing sets.

We initialized our ICNN parameters to match the general affine model from Taha et al. (2024) (see Appendix B for details). Our ICNN uses $k = 1$ layers and hidden dimension $D = 72$. We train the ICNNs for 500 epochs with a learning rate of 10^{-3}. We find empirically that only updating $b_0, p^{\text{upper}}, p^{\text{lower}}$ (while keeping all other parameters frozen) achieved the best performance; we leave it to future work to explore how to achieve better performance when tuning all model parameters.

	ICNN (kW)	Struc.-Pres. (kW)	Gen. Affine (kW)
Mean $\pm$ Standard Error	$\mathbf{0.00 \pm 0.00}$	0.59 ± 0.14	0.20 ± 0.14

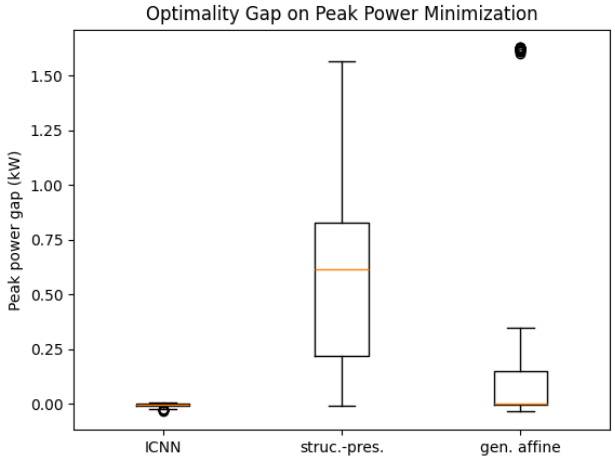

Figure 3: Optimality gap on peak power minimization problem. Our model performs close to optimal and shows an improvement compared to both approximations approaches from Taha et al. (2024).

Our results (Figure 3) show that our method achieves near-zero optimality gap on the peak power minimization problem. We replicate our experiments across 10 different random seeds, corresponding to 10 different collections of individual EV flexibility sets. We compare our method against the two decision-agnostic approaches of Taha et al. (2024)—structure-preserving approximation ("Struc.-Pres.") and general affine approximation ("Gen. Affine")—which show larger and higher-variance optimality gaps.

6. Conclusion

In this work, we develop an end-to-end decision-focused approach to learning inner approximations of EV aggregate flexibility sets and show improved performance compared to decision-agnostic methods. For future work, we plan on evaluating our method on other cost objectives such as cost minimization (Taha et al., 2024). Further theoretical analysis on the convergence of the projected gradient descent inspired algorithm would also be useful to understand model training behavior. Lastly, we believe that a promising direction would be to extend this framework to a contextual set approximation setting (similar to Yeh et al. (2024)) so that a single model can be used to represent aggregate flexibility sets over multiple different sets of EVs without requiring retraining.

References

NVIDIA/cuopt, November 2025. URL https://github.com/NVIDIA/cuopt. original-date: 2025-04-08T21:06:03Z.

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter. Differentiable Convex Optimization Layers. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/9ce3c52fc54362e22053399d3181c638-Abstract.html.

Brandon Amos, Lei Xu, and J. Zico Kolter. Input Convex Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 146–155. PMLR, August 2017. URL https://proceedings.mlr.press/v70/amos17b.html.

Yize Chen, Yuanyuan Shi, and Baosen Zhang. Optimal Control Via Neural Networks: A Convex Approach. September 2018. URL https://openreview.net/forum?id=H1MW72AcK7.

Nicolas Christianson, Wenqi Cui, Steven Low, Weiwei Yang, and Baosen Zhang. Fast and Reliable $N-k$ Contingency Screening with Input-Convex Neural Networks, October 2024. URL http://arxiv.org/abs/2410.00796. arXiv:2410.00796 [eess].

Priya L. Donti, Brandon Amos, and J. Zico Kolter. Task-based End-to-end Model Learning in Stochastic Optimization. In *Advances in Neural Information Processing Systems*, volume 30, Long Beach, CA, USA, December 2017. Curran Associates, Inc. URL http://papers.nips.cc/paper/7132-task-based-end-to-end-model-learning-in-stochastic-optimization.

Adam N. Elmachtoub and Paul Grigas. Smart "Predict, then Optimize". *Management Science*, 68(1):9–26, January 2022. ISSN 0025-1909. doi: 10.1287/mnsc.2020.3922. URL https://pubsonline.informs.org/doi/10.1287/mnsc.2020.3922. Publisher: INFORMS.

He Hao and Wei Chen. Characterizing flexibility of an aggregation of deferrable loads. In *53rd IEEE Conference on Decision and Control*, pages 4059–4064, December 2014. doi: 10.1109/CDC.2014.7040020. URL https://ieeexplore.ieee.org/document/7040020. ISSN: 0191-2216.

He Hao, Borhan M. Sanandaji, Kameshwar Poolla, and Tyrone L. Vincent. Aggregate Flexibility of Thermostatically Controlled Loads. *IEEE Transactions on Power Systems*, 30(1):189–198, January 2015. ISSN 1558-0679. doi: 10.1109/TPWRS.2014.2328865. URL https://ieeexplore.ieee.org/document/6832599.

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-Focused Learning: Foundations, State of the Art, Benchmark and Future Opportunities. *Journal of Artificial Intelligence Research*, 80:1623–1701, August 2024. ISSN 1076-9757. doi: 10.1613/jair.1.15320. URL https://www.jair.org/index.php/jair/article/view/15320.

Karan Mukhi, Georg Loho, and Alessandro Abate. Exact Characterization of Aggregate Flexibility via Generalized Polymatroids, March 2025. URL http://arxiv.org/abs/2503.23458. arXiv:2503.23458 [eess].

Fabian L. Müller, Jácint Szabó, Olle Sundström, and John Lygeros. Aggregation and Disaggregation of Energetic Flexibility From Distributed Energy Resources. *IEEE Transactions on Smart Grid*, 10(2):1205–1214, March 2019. ISSN 1949-3061. doi: 10.1109/TSG.2017.2761439. URL https://ieeexplore.ieee.org/document/8063901.

Ashutosh Nayyar, Josh Taylor, Anand Subramanian, Kameshwar Poolla, and Pravin Varaiya. Aggregate flexibility of a collection of loads. In *52nd IEEE Conference on Decision and Control*, pages 5600–5607, December 2013. doi: 10.1109/CDC.2013.6760772. URL https://ieeexplore.ieee.org/document/6760772. ISSN: 0191-2216.

Md Salman Nazir, Ian A. Hiskens, Andrey Bernstein, and Emiliano Dall'Anese. Inner Approximation of Minkowski Sums: A Union-Based Approach and Applications to Aggregated Energy Resources. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 5708–5715, December 2018. doi: 10.1109/CDC.2018.8618731. URL https://ieeexplore.ieee.org/document/8618731. ISSN: 2576-2370.

Nanda Kishor Panda and Simon H. Tindemans. Efficient quantification and representation of aggregate flexibility in Electric Vehicles. *Electric Power Systems Research*, 235:110811, October 2024. ISSN 0378-7796. doi: 10.1016/j.epsr.2024.110811. URL https://www.sciencedirect.com/science/article/pii/S0378779624006977.

Utsav Sadana, Abhilash Chenreddy, Erick Delage, Alexandre Forel, Emma Frejinger, and Thibaut Vidal. A survey of contextual optimization methods for decision-making under uncertainty. *European Journal of Operational Research*, 320(2):271–289, January 2025. ISSN 0377-2217. doi: 10.1016/j.ejor.2024.03.020. URL https://www.sciencedirect.com/science/article/pii/S0377221724002200.

Sadra Sadraddini and Russ Tedrake. Linear Encodings for Polytope Containment Problems, March 2019. URL http://arxiv.org/abs/1903.05214. arXiv:1903.05214 [math].

Pecan Street. Dataport | HOME, 2018. URL https://dataport.pecanstreet.org/.

Feras Al Taha, Tyrone Vincent, and Eilyan Bitar. An Efficient Method for Quantifying the Aggregate Flexibility of Plug-In Electric Vehicle Populations. *IEEE Transactions on Smart Grid*, pages 1–1, 2024. ISSN 1949-3061. doi: 10.1109/TSG.2024.3384871. URL https://ieeexplore.ieee.org/document/10490133. Conference Name: IEEE Transactions on Smart Grid.

Sina Taheri, Vassilis Kekatos, Sriharsha Veeramachaneni, and Baosen Zhang. Data-Driven Modeling of Aggregate Flexibility Under Uncertain and Non-Convex Device Models. *IEEE Transactions on Smart Grid*, 13(6):4572–4582, November 2022. ISSN 1949-3061. doi: 10.1109/TSG.2022.3185532. URL https://ieeexplore.ieee.org/document/9803223.

Hans Raj Tiwary. On the Hardness of Computing Intersection, Union and Minkowski Sum of Polytopes. *Discrete & Computational Geometry*, 40(3):469–479, October 2008. ISSN 1432-0444. doi: 10.1007/s00454-008-9097-3. URL https://doi.org/10.1007/s00454-008-9097-3.

Christopher Yeh, Nicolas Christianson, Alan Wu, Adam Wierman, and Yisong Yue. End-to-End Conformal Calibration for Optimization Under Uncertainty, September 2024. URL http://arxiv.org/abs/2409.20534. arXiv:2409.20534 [cs].

Lin Zhao, Wei Zhang, He Hao, and Karanjit Kalsi. A Geometric Approach to Aggregate Flexibility Modeling of Thermostatically Controlled Loads. *IEEE Transactions on Power Systems*, 32(6):4721–4731, November 2017. ISSN 0885-8950, 1558-0679. doi: 10.1109/TPWRS.2017.2674699. URL http://ieeexplore.ieee.org/document/7864461/.

Appendix A. ICNN sublevel set, matrix form

This section shows the exact forms of (A, C, d) as described in Section 2.

The ICNN sublevel set $\mathcal{F}$ can be written as

$$\mathcal{F} = A\mathbb{P}_g \qquad \text{where } \mathbb{P}_g = \left\{ \begin{bmatrix} x \\ \mathbf{z} \end{bmatrix} \;\middle|\; C \begin{bmatrix} x \\ \mathbf{z} \end{bmatrix} \leq d \right\}$$

where

$$A = \begin{bmatrix} I_T & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{T \times (T+kD)}$$

$$C = \begin{bmatrix} W_0^{(x)} & -I & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ W_1^{(x)} & W_1^{(z)} & -I & \mathbf{0} & \cdots & \mathbf{0} \\ W_2^{(x)} & \mathbf{0} & W_2^{(z)} & -I & \mathbf{0} & \cdots \\ \vdots & & & \ddots & \ddots & \\ W_{k-1}^{(x)} & \mathbf{0} & \cdots & \mathbf{0} & W_{k-1}^{(z)} & -I \\ \mathbf{0} & -I & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -I & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \\ \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} & -I \\ W_k^{(x)} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & W_k^{(z)} \\ I & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ -I & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{(2Dk+2T+1) \times (T+kD)}$$

$$d = \begin{bmatrix} -b_0 \\ -b_1 \\ \vdots \\ -b_{k-1} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -b_k \\ p^{\text{upper}} \\ -p^{\text{lower}} \end{bmatrix} \in \mathbb{R}^{2Dk+2T+1}$$

Appendix B. Converting Polytope to ICNN

This section shows how to initialize the ICNN such that it matches the general affine model from Taha et al. (2024) as referenced in Section 4.

Note that the general affine model is represented as an AH-polytope of the form $\{Px + \bar{p} \mid Hx \leq h_0\}$ where $h_0 = \frac{1}{N}\sum_{i=0}^{N} h_i$. Since P is empirically always invertible due to the volume maximization objective, we can represent the general affine model as a H-polytope $\{y \mid HP^{-1}y \leq h_0 + HP^{-1}\bar{p}\}$.

Any H-polytope

$$\{x \mid Ax \leq b\}$$

can be equivalently represented by a 1-layer ICNN's sublevel set $\{x \mid f_{\text{ICNN}}(x) \leq 0\}$ by setting

$$\begin{aligned} f_{\text{ICNN}}(x) &= 1^\top z_1 + \mathbf{0}^\top x + 0 \\ z_1 &= \text{ReLU}(\mathbf{0} z_0 + Ax - b), \\ z_0 &= 0. \end{aligned}$$

When $Ax \leq b$, then $z_1 = \mathbf{0}$ and $f_{\text{ICNN}}(x) = 0$. On the other hand, when $Ax > b$, then $z_1 > 0$ and $f_{\text{ICNN}}(x) > 0$.

For an ICNN with k layers, it suffices to adopt the weights above for the penultimate and final layers, while setting all other parameters $(W^{(x)}_{0:k-2}, W^{(z)}_{1:k-2}, b_{0:k-2})$ to zero.

Appendix C. Parameters for EV Data Generation

This table shows the parameters used for EV flexibility set generation as referenced in section 5 and is reproduced from Taha et al. (2024).

Param.	**Description**	**Value/Range**
δ	Time period length	1 hr
T	Time horizon	18
N	Number of EVs	25
a_i	Plug-in time period	0 (3 PM arrival)
d_i	Deadline time period	17 (9 AM departure)
$x_i^{\max}$	Battery capacity	$[25, 50]$ kWh
$u_i^{\max}$	Max charging rate	$[3, 10]$ kW
$u_i^{\min}$	Min charging rate	$[-10, -3]$ kW
x_i^{init}	Initial state-of-charge	$[0,\ 0.4\, x_i^{\max}]$ kWh
x_i^{fin}	Final state-of-charge	$[0.6\, x_i^{\max},\ x_i^{\max}]$ kWh

Table 1: Summary of EV charging parameters used in experiments. The parameters are either fixed at the specified value or uniformly distributed random variables over the specified interval. We associate the initial time period $t = 0$ with the 3:00–4:00 PM time interval.