Lagrangian Duality for Constrained Deep Learning

Ferdinando Fioretto
Syracuse University
ffiorett@syr.edu

Pascal Van Hentenryck
Georgia Institute of Technology
pvh@isye.gatech.edu

Terrence W.K. Mak
Georgia Institute of Technology
wmak@gatech.edu

Cuong Tran
Syracuse University
cutran@syr.edu

Federico Baldo
University of Bologna
federico.baldo2@unibo.it

Michele Lombardi
University of Bologna
michele.lombardi2@unibo.it

Abstract

This paper explores the potential of Lagrangian duality for learning applications that feature complex constraints. Such constraints arise in many science and engineering domains, where the task amounts to learning optimization problems which must be solved repeatedly and include hard physical and operational constraints. The paper also considers applications where the learning task must enforce constraints on the predictor itself, either because they are natural properties of the function to learn or because it is desirable from a societal standpoint to impose them.

This paper demonstrates experimentally that Lagrangian duality brings significant benefits for these applications. In energy domains, the combination of Lagrangian duality and deep learning can be used to obtain state of the art results to predict optimal power flows, in energy systems, and optimal compressor settings, in gas networks. In transprecision computing, Lagrangian duality can complement deep learning to impose monotonicity constraints on the predictor without sacrificing accuracy. Finally, Lagrangian duality can be used to enforce fairness constraints on a predictor and obtain state-of-the-art results when minimizing disparate treatments.

1 Introduction

Deep Neural Networks, in conjunction with progress in GPU technology and the availability of large data sets, have proven enormously successful at a wide array of tasks, including image classification [28], speech recognition [3], and natural language processing [13], to name but a few examples. More generally, deep learning has achieved significant success on a variety of regression and classification tasks. On the other hand, the application of deep learning to aid computationally challenging constrained optimization problems has been more sparse, but is receiving increasing attention, such as the efforts in jointly training prediction and optimization models [38, 25, 27] and incorporating optimization algorithms into differentiable systems [17, 4, 39].

This research originated in an attempt to apply deep learning to fundamentally different application areas: The learning of constrained optimization problems and, in particular, optimization problems with hard physical and engineering constraints. These constrained optimization problems arise in numerous contexts including in energy systems, mobility, resilience, and disaster management. Indeed, these applications must capture physical laws such as Ohm's law and Kirchhoff's law in electrical power systems, the Weymouth equation in gas networks, flow constraints in transportation models, and the Navier-Stoke's equations for shallow water in flood mitigation. Moreover, they often feature constraints that represent good engineering and operational practice to protect various devices. For instance, they may include thermal limits, voltage and pressure bounds, as well as generator and pump limitations, when the domain is that of energy systems. Direct applications of deep learning to these applications may result in predictions with severe constraint violations, as shown in Section 5.

Preprint. Work in progress.

There is thus a need to provide deep learning architectures with capabilities that would allow them to capture constraints directly. Such models can have a transformative impact in many engineering applications by providing high-quality solutions in real-time and be a cornerstone for large planning studies that run multi-year simulations. To this end, this paper proposes a *Lagrangian Dual Framework* (LDF) for Deep Learning that addresses the challenge of enforcing constraints during learning: Its key idea is to exploit Lagrangian duality, which is widely used to obtain tight bounds in optimization, during the training cycle of a deep learning model.

Interestingly, the proposed LDF can be applied to two distinct context: (1) constrained optimization problems, which are characterized by constraints modeling relations among features of each data sample, and (2) problems that require specific properties to hold on the predictor itself, called constrained predictor problems. For instance, energy optimization problems are example problems of the first class. These problems impose constraints that are specific to each data sample, such as, flow conservation constraints or thermal limits bounds. An example problem of the second class is *transprecision computing*, a technique that achieves energy savings by adjusting the precision of power-hungry algorithms. An important challenge in this area is to predict the error resulting from a loss in accuracy and the error should be monotonically decreasing with increases in accuracy. As a result, the learning task may impose constraints over different samples with their predictions used during training. Other applications in dataset-dependent constraint learning may impose fairness constraints on the predictor, e.g., a constraint ensuring equal opportunity [21] or no disparate impact [41] in a classifier that enforces a relation among multiple samples of the dataset.

This paper shows that the proposed LDF provides a versatile tool to address these constrained learning problems, it presents the theoretical foundations of the proposed framework, and demonstrates its practical potential on both constrained optimization and constrained predictors problems. The LDF is evaluated extensively on a variety of real benchmarks in power system optimization and gas compression optimization, that present hard engineering and operational constraints. Additionally, the proposed method is tested on several datasets that enforce non-discriminatory decisions and on a realistic transprecision computing application, that requires constraints to be enforced on the predictors themselves. The results present a dramatic improvement in the number of constraint violations reduction, and often result in substantial improvements in the prediction accuracy in energy optimization problems.

2 Preliminaries: Lagrangian Duality

Consider the optimization problem

$$\mathcal{O} = \underset{y}{\operatorname{argmin}} f(y) \quad \text{subject to} \quad g_i(y) \leqslant 0 \;\; (\forall i \in [m]). \tag{1}$$

In *Lagrangian relaxation*, some or all the problem constraints are relaxed into the objective function using *Lagrangian multipliers* to capture the penalty induced by violating them. When all the constraints are relaxed, the *Lagrangian function* becomes

$$f_\lambda(y) = f(y) + \sum_{i=1}^{m} \lambda_i g_i(y) \tag{2}$$

where the terms $\lambda_i \geqslant 0$ describe the Lagrangian multipliers, and $\lambda = (\lambda_1, \ldots, \lambda_m)$ denotes the vector of all multipliers associated to the problem constraints. Note that, in this formulation, $g(y)$ can be positive or negative. An alternative formulation, used in augmented Lagrangian methods [23] and constraint programming [20], uses the following Lagrangian function

$$f_\lambda(y) = f(y) + \sum_{i=1}^{m} \lambda_i \max(0, g_i(y)) \tag{3}$$

where the expressions $\max(0, g_i(y))$ capture a quantification of the constraint violations. This paper abstracts the constraints formulations in (2) and (3) by using a function $\nu(\cdot)$ that returns either the constraint satisfiability or the violation degree of a constraint.

When using a Lagrangian function, the optimization problem becomes

$$LR_\lambda = \underset{y}{\operatorname{argmin}} f_\lambda(y) \tag{4}$$

and it satisfies $f(LR_\lambda) \leqslant f(\mathcal{O})$. That is, the Lagrangian function is a lower bound for the original function. Finally, to obtain the strongest Lagrangian relaxation of $\mathcal{O}$, the *Lagrangian dual* can be used to find the best Lagrangian multipliers, i.e.,

$$LD = \underset{\lambda \geqslant 0}{\operatorname{argmax}} f(LR_\lambda). \tag{5}$$

For various classes of problems, the Lagrangian dual is a strong approximation of $\mathcal{O}$. Moreover, its optimal solutions can often be translated into high-quality feasible solutions by a post-processing step, i.e., using a *proximal operator* that minimizes the changes to the Lagrangian dual solution while projecting it into the problem feasible region [35].

3 Learning Constrained Optimization Problems

This section describes how to use the Lagrangian dual framework for approximating constrained optimization problems in which constraints model relations among features of each data sample. Importantly, in the associated learning task, each data sample represents a different instantiation of a constrained optimization problem. The section first reviews two fundamental applications that serve as motivation.

3.1 Motivating Applications

Several energy systems require solving challenging (non-convex, non-linear) optimization problems in order to derive the best system operational controls to serve the energy demands of the customers. Power grid and gas pipeline systems are two examples of such applications. While these problems can be solved using effective optimization solvers, their resolution relies on *accurate* predictions of the energy demands. The increasing penetration of renewable energy sources, including those behind the meter (e.g., solar panels on roofs), has rendered accurate predictions more challenging. In turn, predictions need to be performed at minute time scales to ensure sufficient accuracy. Thus, finding optimal solutions for these underlying optimization problems in these reduced time scales becomes computationally challenging, opening opportunities for machine-learning approaches. The next paragraphs review two energy applications that motivate the proposed framework. An extended description of these models is provided in the supplemental material.

Optimal Power Flow The *Optimal Power Flow* (OPF) problem determines the best generator dispatch ($y = S^g$) of minimal cost ($\mathcal{O} = \min_{S^g} \text{cost}(S^g)$) that meets the demands ($d = S^d$) while satisfying the physical and engineering constraints ($g(y)$) of the power system [11], where S^g and S^d denote the vectors (in the complex domain) of generator dispatches and power demands. Typical constraints include the non-linear non-convex AC power flow equations, Kirchhoff's current laws, voltage bounds, and thermal limits. The OPF problem is a fundamental building bock of many applications, including security-constrained OPFs [31]), optimal transmission switching [19], capacitor placement [6], and expansion planning [33] which are of fundamental importance for ensuring a reliable and efficient behavior of the energy system.

Optimal Gas Compressor Optimization The Optimal Gas Compressor Optimization (OGC) problem aims at determining the best compression controls ($y = R$) with minimum compression costs ($\mathcal{O} = \min_R \text{cost}(R)$) to meet gas demands ($d = q^d$) while satisfying the physical and operational limits ($g(y)$) of the natural gas pipeline systems [22]. Therein, R and q^d are compressors control values and gas demands. Typical constraints include: the non-linear gas flow equations describing pressure losses, the flow balance equations, the non-linear non-convex compressor objective $\mathcal{O}$, and the pressure bounds. Similar to the OPF problem, the OGC is a non-linear non-convex optimization problem with physical and engineering constraints and a fundamental building block for many gas systems.

The next section describes how to approximate OPFs and OGCs, by viewing them as parametric optimization problems, using the proposed Lagrangian dual framework.

3.2 The Learning Task

The learning task estimates a parametric version of problem (1), defined as

$$\mathcal{O}(d) = \operatorname*{argmin}_{y} f(y, d) \quad \text{subject to} \quad g_i(y, d) \leqslant 0 \;\; (\forall i \in [m]) \tag{6}$$

with a set of samples $D = \{(d_l, y_l = \mathcal{O}(d_l))\}_{l=1}^{n}$. More precisely, given a parametric model $\mathcal{M}[w]$ with weights w and a loss function $\mathcal{L}$, the learning task must solve the following optimization problem

$$w^* = \operatorname*{argmin}_{w} \sum_{l=1}^{n} \mathcal{L}(\mathcal{M}[w](d_l), y_l) \tag{7a}$$

$$\text{subject to} \;\; g_i(\mathcal{M}[w](d_l), d_l) \leqslant 0 \;\; (\forall i \in [m], l \in [n]) \tag{7b}$$

to obtain the approximation $\widehat{\mathcal{O}} = \mathcal{M}[w^*]$ of $\mathcal{O}$.

The main difficulty lies in the constraints $g_i(y, d) \leqslant 0$, which can represent physical and operational limits, as mentioned in the motivating applications. Observe that the model weights must be chosen so that the constraints are satisfied for all samples, which makes the learning particularly challenging. A naive approach to the learning task is thus likely to result in predictors that significantly violate these constraints, as demonstrated in Section 5, producing a model that would not be useful in practice.

3.3 Lagrangian Dual Framework for Constrained Optimization Problems

To learn constrained optimization problems, the paper proposes a *Lagrangian dual framework* (LDF) to the learning task. The framework relies on the notion of *Augmented Lagrangian* [23] used for solving constrained optimization problems [20].

In more details, LDF exploits a Lagrangian dual approach in the learning task to approximate the minimizer $\mathcal{O}$. Given multipliers $\lambda = (\lambda_1, \ldots, \lambda_m)$, consider the Lagrangian loss function

$$\mathcal{L}_\lambda(\hat{y}_l, y_l, d_l) = \mathcal{L}(\hat{y}_l, y_l) + \sum_{i=1}^{m} \lambda_i \, \nu\left(g_i(\hat{y}_l, d_l)\right),$$

where $\hat{y}_l = \mathcal{M}[w](d_l)$ represents the model prediction. For multipliers λ, solving the optimization problem

$$w^*(\lambda) = \operatorname*{argmin}_{w} \sum_{l=1}^{n} \mathcal{L}_\lambda(\mathcal{M}[w](d_l), y_l, d_l) \tag{8}$$

produces an approximation $\widehat{\mathcal{O}}_\lambda = \mathcal{M}[w^*(\lambda)]$ of $\mathcal{O}$. The Lagrangian dual computes the optimal multipliers, i.e.,

$$\lambda^* = \operatorname*{argmax}_{\lambda} \min_{w} \sum_{l=1}^{n} \mathcal{L}_\lambda(\mathcal{M}[w](d_l), y_l, d_l) \tag{9}$$

to obtain $\widehat{\mathcal{O}}^* = \mathcal{M}[w^*(\lambda^*)]$, i.e., the strongest Lagrangian relaxation of $\mathcal{O}$.

Learning $\widehat{\mathcal{O}}^*$ relies on an iterative scheme that interleaves the learning of a number of Lagrangian relaxations (for various multipliers) with a subgradient method to learn the best multipliers. The LDF, described in Equations (8) and (9), is summarized in Algorithm 1. Given the input dataset D, the optimizer step size $\alpha > 0$, and a Lagrangian step size s_k, the Lagrangian multipliers are initialized in line 1. The training is performed for a fixed number of epochs, and each epoch k optimizes the model weights w of the optimizer $\mathcal{M}[w, \lambda^k]$ using the Lagrangian multipliers λ^k associated with current epoch (lines 3–5). Finally, after each epoch, the Lagrangian multipliers are updated according to a *dual ascent* rule [9] (line 6).

4 Learning Constrained Predictors

This section describes how to use the Lagrangian dual framework for problems in which constraints are not sample-independent, but enforcing global properties between different samples in the dataset and the predictor outputs. It starts with two motivating applications.

Algorithm 1: LDF for Constrained Optimization Problems

input : $D = (d_l, y_l)_{l=1}^n$: Training data;
$\alpha, s = (s_0, s_1, \ldots)$: Optimizer and Lagrangian step sizes.

$\lambda_i^0 \leftarrow 0 \;\; \forall i \in [m]$
for *epoch* $k = 0, 1, \ldots$ **do**
  foreach $(y_l, d_l) \in D$ **do**
    $\hat{y}_l \leftarrow \mathcal{M}[w, \lambda^k](d_l)$
    $w \leftarrow w - \alpha \nabla_w \mathcal{L}_{\lambda^k}(\hat{y}_l, y_l, d_l)$
  $\lambda_i^{k+1} \leftarrow \lambda_i^k + s_k \sum_{l=1}^n \nu_i \left(g_i(\hat{y}_l, d_l)\right) \;\; \forall i \in [m]$

4.1 Motivating Applications

Several applications require to enforce constraints on the learning process itself to attain desirable properties of the predictor. These constraints impose conditions on subsets of the samples that must be satisfied. For instance, assume that there is a partial order $\preceq$ on the optimization inputs and the following property holds:

$$d_1 \preceq d_2 \Rightarrow f(\mathcal{O}(d_1), d_1) \leqslant f(\mathcal{O}(d_2), d_2).$$

The predictor should ideally satisfy these constraints as well:

$$d_1 \preceq d_2 \Rightarrow f(\widehat{\mathcal{O}}(d_1), d_1) \leqslant f(\widehat{\mathcal{O}}(d_2), d_2).$$

Transprecision computing Transprecision computing is the idea of reducing energy consumption by reducing the precision (a.k.a. number of bits) of the variables involved in a computation [30]. It is especially important in low-power embedded platforms, which arise in many contexts such as smart wearable and autonomous vechicles. Increasing precision typically reduces the error of the target algorithm. However, it also increases the energy consumption, which is a function of the maximal number of used bits. The objective is to design a *configuration* d_l, i.e., a mapping from input computation to the precision for the variables involved in the computation. The sought configuration should balance *precision* and *energy consumption*, given a bound to the error produced by the loss in precision when the highest precision configuration is adopted.

However, given a configuration, computing the corresponding error can be very time-consuming and the task considered in this paper seeks to learn a mapping between configurations and error. This learning task is non-trivial, since the solution space precision-error is non-smooth and non-linear [30]. The samples (d_l, y_l) in the dataset represent, respectively, a configuration d_l and its associated error y_l obtained by running the configuration d_l for a given computation. The problem $\mathcal{O}(d_l)$ specifies the error obtained when using configuration d_l.

Importantly, transcomputing expects a *monotonic* behavior: Higher precision configurations should generate more accurate results (i.e., a smaller error). Therefore, the structure of the problem imposes the learning task to require a dominance relation $\preceq$ between instances of the dataset. More precisely, $d_2 \preceq d_2$ holds if

$$\forall i \in [N] : \; x_{1_i} \leqslant x_{2_i}$$

where N is the number of variables involved in the computation and x_{1_i}, x_{2_i} are the precision values for the variables in d_1 and d_2 respectively.

Fair Classifier The second motivating application considers the task of building a classifier that satisfies *disparate impact* [41] with respect to a protected attribute d^s and outcome y. A binary classifier does not suffer from disparate impact if

$$\Pr(\hat{y} = 1 | d^s = 0) = \Pr(\hat{y} = 1 | d^s = 1). \tag{10}$$

For outcome $y = 1$, the constraint above requires the predictor $\hat{y}$ to have equal *predicted positive rates* across the different sensitive classes: $d^s = 0$ and $d^s = 1$, in the binary task example above. For $y = 0$, the constraint enforces equal *predicted negative rates*. Disparate impact constraints the predicted positive (or negative) rates to be similar across all sensitive attributes. To construct an estimator that minimizes the disparate impact, the paper considers $|\mathcal{D}_s| = 2$ estimators $\mathcal{M}_0$ and $\mathcal{M}_1$,

each associated with a dataset partition $D_{|s_i} = \{(d_l, y_l)|d_l^s = s_i\}$ that marginalizes for a particular (combination of) value(s) of the protected feature(s), in addition to the classical estimator $\mathcal{M}$ that is trained over the entire dataset D. Thus, the learning process is defined by the following objective:

$$\min_{w,w_0,w_1} \mathcal{L}\left(\mathcal{M}[w](D)\right) + \sum_{i=0}^{1} \mathcal{L}\left(\mathcal{M}_i[w_i](D_{|s_i})\right) \tag{11a}$$

$$\text{such that } \frac{\left|\sum_{x_i \in D_{s_0}} I(\hat{y}_i = 1)\right|}{|D_{s_0}|} - \frac{\left|\sum_{x_i \in D_{s_1}} I(\hat{y}_i = 1)\right|}{|D_{s_1}|}, \tag{11b}$$

where I is the indicator function. It enforces a constraint on the output of the classifiers $\mathcal{M}_0$, trained on data D_{s_0} to be equivalent to that of the output of the classifier $\mathcal{M}_1$, trained on the dataset D_{s_1}, when their predicted outcome is positive.

The next section will specify how to encode such type of constraints as well as how to express and enforce dominance relations in the proposed constrained learning framework.

4.2 The Learning Task

Consider a set $\mathcal{S} = \{S_1, \ldots, S_q\}$ where S_i is a subset of the inputs that must satisfy the associated constraint

$$h_i(\{\mathcal{O}(d_l)\}_{l \in S_i}, \boldsymbol{d}_{S_i}),$$

where $\boldsymbol{d}_{S_i} = \{d_l\}_{l \in S_i}$, and denote $\boldsymbol{y}_{S_i} = \{y_l\}_{l \in S_i}$.

In this context, the learning task is defined by the following optimization problem

$$\operatorname*{argmin}_{w} \sum_{l=1}^{n} \mathcal{L}(\mathcal{M}[w](d_l), y_l) \tag{12a}$$

$$\text{subject to } g_i\left(\mathcal{M}[w](d_l), d_l\right) \leqslant 0 \quad (\forall i \in [m], l \in [n]) \tag{12b}$$

$$h_i\left(\{\mathcal{M}[w](d_l)\}_{l \in S_i}, \boldsymbol{d}_{S_i}\right) \quad (\forall i \in [q]). \tag{12c}$$

4.3 Lagrangian Dual Framework for Constrained Predictors

To approximate Problem (12), the learning task considers Lagrangian loss functions, for subset of the inputs $S_i \in \mathcal{S}$, of the form

$$\mathcal{L}_{\mu,\lambda}(\tilde{\boldsymbol{y}}_{S_i}, \boldsymbol{y}_{S_i}, \boldsymbol{d}_{S_i}) = \sum_{l \in S_i} \mathcal{L}_\lambda(\tilde{y}_l, y_l, d_l) + \sum_{i=1}^{q} \mu_i \, \nu\left(h_i(\tilde{\boldsymbol{y}}_{S_i}, \boldsymbol{d}_i)\right), \tag{13}$$

where $\tilde{y}_l = \mathcal{M}[w](d_l)$ and $\tilde{\boldsymbol{y}}_{S_i} = \{\mathcal{M}[w](d_l)_{l \in S_i}\}$. It learns approximations of the Lagrangian relaxations $\widehat{\mathcal{O}}_{\lambda,\mu}$ of the form

$$w^*(\mu, \lambda) = \operatorname*{argmin}_{w} \sum_{i=1}^{q} \mathcal{L}_{\mu,\lambda}(\{\mathcal{M}[w](d_l)\}_{l \in S_i}, \boldsymbol{y}_{S_i}, \boldsymbol{d}_{S_i}), \tag{14}$$

as well as the Lagrangian duals of Equation (14) of the form

$$\lambda^*(\mu) = \operatorname*{argmax}_{\lambda} \min_{w} \sum_{i=1}^{q} \mathcal{L}_{\mu,\lambda}(\{\mathcal{M}[w](d_l)\}_{l \in S_i}, \boldsymbol{y}_{S_i}, \boldsymbol{d}_{S_i}), \tag{15}$$

and, finally, the Lagrangian dual of the Lagrangian duals (Equation (15)) as

$$\mu^* = \operatorname*{argmax}_{\mu} \max_{\lambda} \min_{w} \sum_{i=1}^{q} \mathcal{L}_{\mu,\lambda}(\{\mathcal{M}[w](d_l)\}_{l \in S_i}, \boldsymbol{y}_{S_i}, \boldsymbol{d}_{S_i}) \tag{16}$$

to obtain the best estimator $\widehat{\mathcal{O}}^* = \mathcal{M}[w^*]$, where

$$w^* = \operatorname*{argmin}_{w} \sum_{i=1}^{q} \mathcal{L}_{\mu^*,\lambda^*(\mu^*)}(\{\mathcal{M}[w](d_l)\}_{l \in S_i}, \boldsymbol{y}_{S_i}, \boldsymbol{d}_{S_i}).$$

Algorithm 2: LDF for Constrained Predictor Problems

input : $D = (d_l, y_l)_{l=1}^n, \mathcal{S} = \{S_1, \ldots, S_n\}$ Training data and data partitions;
$\alpha, s = (s_0, s_1, \ldots), t = (t_0, t_1, \ldots)$: Optimizer and Lagrangian step sizes.

1 $\lambda_i^0 \leftarrow 0 \;\; \forall i \in [m]$
2 $\mu_i^0 \leftarrow 0 \;\; \forall i \in [q]$
3 **for** *epoch* $k = 0, 1, \ldots$ **do**
4 **foreach** $S_i \in \mathcal{S}$ **do**
5 $\hat{\boldsymbol{y}}_{S_i} \leftarrow \{\mathcal{M}[w, \lambda^k, \mu^k](d_l)\}_{l \in S_i}$
6 $w \leftarrow w - \alpha \nabla_w \mathcal{L}_{\lambda^k, \mu^k}(\hat{\boldsymbol{y}}_{S_i}, \boldsymbol{y}_{S_i}, \boldsymbol{d}_{S_i})$
7 $\lambda_i^{k+1} \leftarrow \lambda_i^k + s_k \sum_{l=1}^n \nu_i \left(g_i(\hat{y}_l, d_l)\right) \;\; \forall i \in [m]$
8 $\mu_i^{k+1} \leftarrow \mu_i^k + t_k \nu_i \left(h(\hat{\boldsymbol{y}}_{S_i}, \boldsymbol{d}_{S_i})\right) \;\; \forall i \in [q]$

The Lagrangian dual framework for constrained predictors, described in Equations (14)–(16), is summarized in Algorithm 2. The learning algorithm interleaves the learning of the Lagrangian duals with the subgradient optimization of the multipliers μ. Given the input dataset D, a set $\mathcal{S}$ of subsets of inputs, the optimizer step size $\alpha > 0$, and Lagrangian step sizes s_k, and t_k, the Lagrangian multipliers are initialized in lines 1 and 2. The training is performed for a fixed number of epochs, and each epoch k optimizes the model weights w of the optimizer $\mathcal{M}$ using the Lagrangian multipliers λ^k and μ^k associated with current epoch k, denoted $\mathcal{M}[w, \lambda^k, \mu^k]$ in the algorithm (lines 4–6). Similarly to Algorithm 1, the Lagrangian multipliers λ_i for the dual variables are updated after each epoch, on line 7). Finally, the algorithm updates the multipliers μ_i associated to the Lagrangian duals of the Lagrangian duals (line 8).

5 Experiments

This section evaluates the proposed LDF on constrained optimization problems for energy and gas networks and on constrained learning problems–that enforce constraints on the predictors–for applications in transprecision computing and fairness.

5.1 Constrained Optimization Problems

Data set Generation The experiments examine the proposed models on a variety of power networks from the NESTA library [12] and natural gas benchmarks from [29] and GasLib [36]. The ground truth data are constructed as follows: For each power and gas network, different benchmarks are generated by altering the amount of nominal demands $d = S^d$ (for power networks) and $d = q^d$ (for gas networks) within a $\pm 20\%$ range. The resulting 4000 demand vectors are used to generate solutions to the OPF and OGC problems. Increasing loads causes heavily congestions to the system, rendering the computation of optimal solutions challenging. A network value, that constitutes a dataset entry $(d_l, y_l = \mathcal{O}(d))$, is a feasible solution obtained by solving the AC-OPF problem [11], for electricity networks, or the OGC problem, for gas networks [22]. The experiments use a $80/20$ train-test split and results are reported on the test set.

Learning Models The experiments use a baseline ReLU network $\mathcal{M}$, with 5 layers which minimizes the Mean Squared Error (MSE) loss $\mathcal{L}$ to predict to active power $\hat{p}$, voltage magnitude $\hat{v}$, and voltage angle $\hat{\theta}$, for energy networks, and compression ratios $\hat{R}$, pressure $\hat{p}$, and gas flows $\hat{q}$, for gas networks.

This baseline model is compared with a model $\mathcal{M}_{\text{C}}$ that exploits the problem constraints and minimizes the loss: $\mathcal{L} + \lambda\nu(\cdot)$, with multiplier values λ fixed to $\mathbf{1}$. Finally, $\mathcal{M}_{\text{C}}^{\text{D}}$ extends model $\mathcal{M}_{\text{C}}$ by learning the Lagrangian multipliers using the LDF introduced in Section 3.3. The constrained learning model for power systems also exploits the hot-start techniques used in [18], with states differing by at most 1%. Experiments using larger percentages (up to 3%) showed similar trends. The training uses the Adam optimizer with learning rate ($\alpha = 10^{-3}$) and was performed for 80 epochs using batch sizes $b = 64$. Finally, the Lagrangian step size ρ is set to 10^{-4}. Extensive additional information about the network structure, the optimization model (OPF and OGC), the learning loss functions and constraints, as well as additional experimental analysis is provided in the appendix.

Test Case	Type	$\mathcal{M}$	$\mathcal{M}_{\text{C}}$		$\mathcal{M}_{\text{C}}^{\text{D}}$	
		err (%)	err (%)	gain	err (%)	gain
30_ieee	$\hat{p}$	3.3465	0.3052	(10.96)	**0.0055**	(608.4)
	$\hat{v}$	14.699	0.3130	(46.96)	**0.0070**	(2099)
	$\hat{\theta}$	4.3130	0.0580	(74.36)	**0.0041**	(1052)
	$\tilde{p}^f$	27.213	0.2030	(134.1)	**0.0620**	(438.9)
118_ieee	$\hat{p}$	0.2150	0.0380	(5.658)	**0.0340**	(6.323)
	$\hat{v}$	7.1520	0.1170	(61.12)	**0.0290**	(246.6)
	$\hat{\theta}$	4.2600	1.2750	(3.341)	**0.2070**	(20.58)
	$\tilde{p}^f$	38.863	0.6640	(58.53)	**0.4550**	(85.41)
300_ieee	$\hat{p}$	0.0838	0.0174	(4.816)	**0.0126**	(6.651)
	$\hat{v}$	28.025	3.1130	(9.002)	**0.0610**	(459.4)
	$\hat{\theta}$	12.137	7.2330	(1.678)	**2.5670**	(4.728)
	$\tilde{p}^f$	125.47	26.905	(4.663)	**1.1360**	(110.4)

Table 1: table

Mean Prediction Errors (%) and accuracy gain (%) on OPF Benchmarks.

Test Case	Type	$\mathcal{M}$	$\mathcal{M}_{\text{C}}$		$\mathcal{M}_{\text{C}}^{\text{D}}$	
		err (%)	err (%)	gain	err (%)	gain
24-pipe	$\hat{R}$	0.0052	0.0079	(0.658)	**0.0025**	(2.080)
	$\hat{p}$	**0.0057**	0.0068	(0.838)	**0.0057**	(1.000)
	$\hat{q}$	0.0029	0.0592	(0.049)	**0.0007**	(4.142)
40-pipe	$\hat{R}$	0.0009	0.0103	(0.087)	**0.0006**	(1.833)
	$\hat{p}$	0.0011	0.0025	(0.240)	**0.0006**	(1.500)
	$\hat{q}$	0.0006	0.0329	(0.033)	**0.0004**	(1.500)
135-pipe	$\hat{R}$	0.0206	0.0317	(0.650)	**0.0199**	(1.307)
	$\hat{p}$	0.0260	**0.0209**	(1.067)	0.0225	(0.916)
	$\hat{q}$	0.0223	0.0572	(0.455)	**0.0222**	(1.005)

Table 2: table

Mean Prediction Errors (%) and accuracy gain (%) on OGC Benchmarks.

Prediction Errors Table 1 and 2 report the average L_1-distance and the *prediction errors* between a subset of predicted variables y (marked with $\hat{y}$) on both the power and gas benchmarks and their original ground-truth quantities. The error for y is reported in percentage as $100\frac{\|\hat{y}-y\|_1}{\|y\|_1}$ and the gain (in parenthesis) reports the ratio between the error obtained by the baseline model accuracy and the constrained models.

For the power networks, the models focus on predicting the active generation dispatches $\hat{p}^g = Re(S^g)$, voltage magnitudes $\hat{v}$, voltage angles $\hat{\theta}$, and the active transmission line (including transformers) flows $\tilde{p}^f$. Power flows $\tilde{p}^f$ are not directly predicted but computed from the predicted quantities through the Ohm's laws (See section B.2). For the gas networks, the models focus on predicting compression ratios $\hat{R}$, pressure values $\hat{p}$, and gas flows $\hat{q}$. The best results are highlighted in bold.

A clear trend appears: The prediction errors decrease with the increase in model complexity. In particular, model $\mathcal{M}_{\text{C}}$, which exploits the problem constraints, predicts voltage quantities and power flows that are up to two order of magnitude more precise than those predicted by $\mathcal{M}$, for OPF problems. The prediction errors on OGC benchmarks, instead remain of the same order of magnitude as those obtained by the baseline model $\mathcal{M}$, albeit the accuracy of the prediction increases consistently when adopting the constrained models. This can be explained by the fact that the gas networks behave largely monotonically in compressor costs for varying loads. *Finally, the LDF that finds the best multipliers ($\mathcal{M}_C^D$) consistently improves the baseline model on OGC benchmarks, and further improves $\mathcal{M}_C$ predictions by an additional order of magnitude, for OPF problems.*[1]

[1]The accuracy gains appear more pronounced on OPF problems since the baseline model $\mathcal{M}$ produces already extremely accurate results for OGC benchmarks.

Measuring The Constraint Violations This section simulates the prediction results in an operational environment, by measuring the minimum required adjustments in order to satisfy the operational limits and the physical constraints in the energy domains studied. Given the predictions $\hat{y}$ returned by a model, the experiments compute a projection $\bar{y}$ of $\hat{y}$ into the feasible region and reports the minimal distance $\|\bar{y} - \hat{y}\|_2$ of the predictions from the satisfiable solution. This step is executed on all the predicted control variables: generator dispatch and voltage set points, for power systems, and compression ratios, for gas systems. Table 3 reports the minimum distance (normalized in percentage) required to satisfy the operational limits and physical constraints, and the best results are highlighted in bold. These results provide a proxy to evaluate the degree of constraint violations of a model. Notice that the adjustment required decrease with the increase in model complexity. *The results show that the LDF can drastically reduce the effort required by a post-processing step to satisfy the problem constraints.*

Test Case	Type	$\mathcal{M}$	$\mathcal{M}_{\text{C}}$		$\mathcal{M}_{\text{C}}^{\text{D}}$	
		violation (%)	violation (%)	gain	violation (%)	gain
30_ieee	p^g	2.0793	0.1815	(11.45)	**0.0007**	(2970.0)
	v	83.138	0.0944	(880.7)	**0.0037**	(22469)
118_ieee	p^g	0.1071	0.0043	(24.91)	**0.0038**	(28.184)
	v	3.4391	0.0956	(35.97)	**0.0866**	(39.712)
300_ieee	p^g	0.0447	0.0091	(4.912)	**0.0084**	(5.3214)
	v	31.698	0.2383	(133.0)	**0.1994**	(158.97)
24-pipe	R	0.1012	0.1033	(0.978)	**0.0897**	(1.1282)
40-pipe	R	0.0303	0.0277	(1.094)	**0.0207**	(1.4638)
135-pipe	R	0.0322	0.0264	(1.219)	**0.0005**	(64.4)

Table 3: Average distances (in percentage) for the active power p^g, voltage magnitude v, and compressor ratios R of the simulated solutions w.r.t. the corresponding predictions.

5.2 Constrained Predictor Problems

This section examines the LDF for constrained predictor problems discussed in Section 4 on transprecision computing and fairness application domains.

Transprecision computing The benchmark considers training a neural network to predict the error of transprecision configurations. The monotonicity property is expressed as a constraint exploiting the relation of dominance among configurations of the train set, i.e. $\nu_i = \max(0, \mathcal{M}(x_1) - \mathcal{M}(x_2))$ if $x_1 \preceq x_2$ for every pair (x_1, x_2) in the dataset. This approach is particularly suited for instances of training with scarce data points with a high rate of violated constraints, since it guides the learning process towards a more general approximation of the target function. In order to explore different scenarios the experiments use 5 different train sets of increasing size, i.e. 200, 400, 600, 800, and 1000. The test set size is fixed to 1000 samples. The data sets are constructed by generating random configuration (d_i) and computing errors (y) by measuring the performance loss obtained when running the configuration d_i on the target algorithm. Ten disjoint training sets are constructed so that the violation constraint ratio is 0.5, while the test set was fixed.

Table 4 illustrates the average results comparing a model ($\mathcal{M}$) that minimizes the Mean Absolute Error (MAE) prediction error, one ($\mathcal{M}_C$) that include the Lagrangian loss functions $\mathcal{L}_\lambda$ associated to each constraint and where all weights λ are fixed to value 1.0, and the proposed model ($\mathcal{M}_C^D$) that uses the LDF to find the optimal Lagrangian weights. All prediction model are implemented as classical feed-forward neural network with 3 hidden layers and 10 units and minimize the MAE as loss function. The training uses 150 epochs, Lagrangian step sizes $t_k = 10^{-3}$ and learning rate 10^{-3}. The table also show the average number of constraint violations (VC) and the sum of the magnitudes of the violated constraints (SMVC), i.e., $\sum_{x_i, x_j \in \mathcal{D}; x_i \preceq x_j \wedge \mathcal{M}(x_i) > \mathcal{M}(x_j)} |\mathcal{M}(x_i) - \mathcal{M}(x_j)|$.

The table clearly illustrates the positive effect of adding the constraints within the LDF on reducing the number of constraint violations. Notice that model $\mathcal{M}_C$, that weights all the constraints violations equally, produces a degradation of both the MAE score and the number and magnitude of the constraint violations, when compared to the baseline model ($\mathcal{M}$). The benefit of using the LDF is substantial in both reducing the number of constraint violations and in retaining a high model

n_{tr}	$\mathcal{M}$			$\mathcal{M}_C$			$\mathcal{M}_D^s$		
	MAE	VC	SMVC	MAE	VC	SMVC	MAE	VC	SMVC
200	0.1902	9.6	0.2229	0.1919	35.8	0.4748	**0.1883**	**7.4**	**0.1872**
400	0.1765	4.5	0.0804	0.1999	19.4	0.2149	**0.1763**	**2.6**	**0.0369**
600	**0.1687**	2.5	0.0397	0.2022	9.1	0.0683	0.1723	**1.7**	**0.0224**
800	**0.1672**	3.0	0.0600	0.2007	8.5	0.0746	0.1704	**0.6**	**0.0131**
1000	**0.1640**	0.4	0.0048	0.2012	5.7	0.0511	0.1642	**0.5**	**0.0043**

Table 4: Mean Absolute Error (MAE), number of constraints violations (VC), and sum of absolute magnitude of violated constraints (SMVC). Best results are highlighted in bold.

precision (i.e., a low MAE score). The most significant contribution was obtained on training sets with fewer data points, confirming that *exploiting the Lagrangian Duals of the Constraint Violations can be an important tool for constrained learning*.

Fairness Constraints The benchmark considers building a classifier that minimizes *disparate treatment* [41]. The paper considers the disparate DT index, introduced by Aghaei et al. [2], to quantify the disparate impact in a dataset. Given a dataset of samples $D = (x_i, y_i)_{i\in[n]}$, this index is defined as:

$$\text{DT}(D) = \left|\sum_{x_i\in D_{s_0}} I(\hat{y}_i = 1)\right| / |D_{s_0}| - \left|\sum_{x_i\in D_{s_1}} I(\hat{y}_i = 1)\right| / |D_{s_1}|\,.$$

where I is the characteristic function and $\hat{y}_i$ is the predicted outcome for sample x_i. The idea is to use a locally weighted average to estimate the conditional expectation. The higher is the DT score for a dataset, the more it suffer from disparate treatment, with DT $= 0$ meaning that the dataset does not suffer from disparate treatment.

Since the DT constraint introduced in Equation (11b) is not differentiable with the respect to the model parameters, the paper uses an expectation matching constraints between the predictors for the protected classes, defined as:

$$\left|E_{x\sim D_{s_0}}[\mathcal{M}_0(x)|z(x) = 0] - E_{x\sim D_{s_1}}[\mathcal{M}_1(x)|z(x) = 1]\right| = 0 \tag{17}$$

The effect of the Lagrangian Dual framework on reducing disparate treatment was evaluated on three datasets: The *Adult* dataset [26], containing 30,000 samples and 23 features, in which the prediction task is that of assessing whether an individual earns more than $50K$ per year and the protected attribute is *race*. The *Default* of Taiwanese credit card users [40], containing 45,000 samples and 13 features, in which the task is to predict whether an individual will default and the protected attribute is *gender*. Finally, the *Bank* dataset [41], containing 41,188 samples, each with 20 attributes, where the task is to predict whether an individual has subscribed or not and the protected attribute is *age*. The experiments use a 80/20 train/test split and executes a 5-fold cross-validation to evaluate the accuracy and the fairness score (DT) of the predictors.

Table 5 illustrates the results comparing model $\mathcal{M}$ that minimizes the Binary Cross Entropy (BCE) loss, model $\mathcal{M}_C$ that includes the Lagrangian loss functions $\mathcal{L}_\lambda$ associated with each constraint and where all λ are fixed to value 1.0, and the proposed model $\mathcal{M}_C^D$ that uses the LDF to find the optimal Lagrangian weights. All prediction models use a classical feed-forward neural network with 3 layers and 10 hidden units. The training uses 100 epochs, Lagrangian step size $s_k = 10^{-4}$ and learning rate 10^{-3}. The models are also compared against a state-of-the-art fair classifier which enforces fairness by limiting the covariance between the loss function and the sensitive variable [41]. While [41] focuses on logistic regression, the model is implemented as a neural network with the same hyper parameters of model $\mathcal{M}$. The table clearly shows the effect of the Lagrangian constraints on reducing the DT score. Not only such reduction attains state-of-the-art results on the DT score, but it also comes at a much more contained cost of accuracy degradation.

6 Related Work

The application of Deep Learning to constrained optimization problems is receiving increasing attention. Approaches which embed optimization components in neural networks include [38, 25, 27].

Dataset	$\mathcal{M}$		$\mathcal{M}_C$		$\mathcal{M}_C^D$		Zafar'19	
	Acc.	DT	Acc.	DT	Acc.	DT	Acc.	DT
Adult	**0.8423**	0.1853	0.8333	0.0627	0.8335	**0.0545**	0.7328	0.1037
Default	0.8160	0.0162	**0.8182**	0.0216	0.8166	**0.0101**	0.6304	0.0109
Bank	**0.8257**	0.4465	0.7744	0.4515	0.8135	0.1216	0.7860	**0.0363**

Table 5: Classification accuracy (Acc.) and fairness score (DT)

These approaches typically rely on problems exhibiting properties like convexity or submodularity. Another line of work leverages explicit optimization algorithms as a differentiable layer into neural networks [4, 17, 39]. A further collection of works interpret constrained optimization as a two-player game, in which one player optimizes the objective function and a second player attempt at satisfying the problem constraints [24, 32, 1]. For instance Agarwal et al. [1], proposes a best-response algorithm applied to fair classification for a class of linear fairness constraints. To study generalization performance of training algorithms that learn to satisfy the problem constraints, Cotter et al. [14] propose a two-players game approach in which one player optimizes the model parameters on the training data and the other player optimizes the constraints on a validation set. Arora et al. [5] proposed the use of a multiplicative rule to iteratively changing the weights of different distributions to maintaining some properties and discuss the applicability of the approach to a constraint satisfaction domain.

A different strategy for minimizing empirical risk subject to a set of constraints is that of using projected stochastic gradient descent (PSGD). Cotter et al. [15] proposed an extension of PSGD that stay close to the feasible region while applying constraint probabilistically at each iteration of the learning cycle.

Different from these proposal, this paper proposes a framework that exploits key ideas in Lagrangian duality to encourage the satisfaction of generic constraints within a neural network learning cycle and apply to both sample dependent constraints (as in the case of energy problems) and dataset dependent constraints (as in the case of transprecision computing and fairness problems). This paper builds on the recent results that were dedicated to learning and optimization in power systems [18].

7 Conclusions

This paper proposed a Lagrangian dual framework to encourage the satisfaction of constraints in deep learning. It was motivated by a desire to learn parametric constrained optimization problems that feature complex physical and engineering constraints. The paper showed how to exploit Lagrangian duality for deep learning to obtain predictors that minimize constraint violations. The proposed framework can be applied to constrained optimization problems, in which the constraints model relations among features of each data sample, and to constrain predictors in which the constraints enforce global properties over multiple dataset samples and the predictor outputs.

The Lagrangian dual framework for deep learning was evaluated on a collection of realistic energy networks, by enforcing non-discriminatory decisions on a variety of datasets, and on a transprecision computing application. The results demonstrated the effectiveness of the proposed method that dramatically decreases constraint violations committed by the predictors and, in some applications, as in those in energy optimization, increases the prediction accuracy by up to two orders of magnitude.

References

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. *arXiv preprint arXiv:1803.02453*, 2018.

[2] Sina Aghaei, Mohammad Javad Azizi, and Phebe Vayanos. Learning optimal and fair decision trees for non-discriminative decision-making. In *AAAI*, pages 1418–1426, 2019.

[3] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *ICML*, pages 173–182, 2016.

[4] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *ICML*, pages 136–145. JMLR. org, 2017.

[5] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[6] M. E. Baran and F. F. Wu. Optimal capacitor placement on radial distribution systems. *IEEE TPD*, 4(1):725–734, Jan 1989.

[7] R. Bent, S. Blumsack, P. Van Hentenryck, C. Borraz-Sánchez, and M. Shahriari. Joint electricity and natural gas transmission planning with endogenous market feedbacks. *IEEE Transactions on Power Systems*, 33(6):6397–6409, Nov 2018.

[8] Conrado Borraz-Sánchez, Russell Bent, Scott Backhaus, Hassan Hijazi, and Pascal Van Hentenryck. Convex relaxations for gas expansion planning. *INFORMS Journal on Computing*, 28(4):645–656, 2016.

[9] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[10] Mary B. Cain, Richard P. O'neill, and Anya Castillo. History of optimal power flow and formulations optimal power flow paper 1. `https://www.ferc.gov/industries/electric/indus-act/market-planning/opf-papers.asp`, 2012.

[11] B. H. Chowdhury and S. Rahman. A review of recent advances in economic dispatch. *IEEE Transactions on Power Systems*, 5(4):1248–1259, Nov 1990.

[12] Carleton Coffrin, Dan Gordon, and Paul Scott. NESTA, the NICTA energy system test case archive. *CoRR*, abs/1411.0359, 2014.

[13] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, pages 160–167, 2008.

[14] Andrew Cotter, Maya Gupta, Heinrich Jiang, Nathan Srebro, Karthik Sridharan, Serena Wang, Blake Woodworth, and Seungil You. Training well-generalizing classifiers for fairness metrics and other data-dependent constraints. *arXiv preprint arXiv:1807.00028*, 2018.

[15] Andrew Cotter, Maya Gupta, and Jan Pfeifer. A light touch for heavily constrained sgd. In *Conference on Learning Theory*, pages 729–771, 2016.

[16] Deutche-Energue-Agentur. The e-highway2050 project. `http://www.e-highway2050.eu`, 2019. Accessed: 2019-11-19.

[17] Priya Donti, Brandon Amos, and J Zico Kolter. Task-based end-to-end model learning in stochastic optimization. In *NIPS*, pages 5484–5494, 2017.

[18] Ferdinando Fioretto, Terrence W.K. Mak, and Pascal Van Hentenryck. Predicting ac optimal power flows: Combining deep learning and lagrangian dual methods. In *AAAI*, page to appear, 2020.

[19] E. B. Fisher, R. P. O'Neill, and M. C. Ferris. Optimal transmission switching. *IEEE Transactions on Power Systems*, 23(3):1346–1355, Aug 2008.

[20] Daniel Fontaine, Michel Laurent, and Pascal Van Hentenryck. Constraint-based lagrangian relaxation. In *CP*, pages 324–339, 2014.

[21] Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3315–3323. 2016.

[22] M. Herty, J. Mohring, and V. Sachers. A new model for gas flow in pipe networks. *Mathematical Methods in the Applied Sciences*, 33(7):845–855, 2010.

[23] Magnus R Hestenes. Multiplier and gradient methods. *Journal of optimization theory and applications*, 4(5):303–320, 1969.

[24] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. *arXiv preprint arXiv:1711.05144*, 2017.

[25] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. In *NIPS*, pages 6348–6358, 2017.

[26] Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *KDD*, volume 96, pages 202–207, 1996.

[27] Wouter Kool, Herke Van Hoof, and Max Welling. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*, 2018.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[29] Terrence W. K. Mak, Pascal Van Hentenryck, Anatoly Zlotnik, and Russell Bent. Dynamic compressor optimization in natural gas pipeline systems. *INFORMS Journal on Computing*, 31(1):40–65, 2019.

[30] A Cristiano I Malossi, Michael Schaffner, and et al. The transprecision computing paradigm: Concept, design, and applications. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2018*, pages 1105–1110. IEEE, 2018.

[31] A Monticelli, MVF Pereira, and S Granville. Security-constrained optimal power flow with post-contingency corrective rescheduling. *IEEE TPS*, 2(1):175–180, 1987.

[32] Harikrishna Narasimhan. Learning with complex loss functions and constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 1646–1654, 2018.

[33] Niharika, S. Verma, and V. Mukherjee. Transmission expansion planning: A review. In *International Conference on Energy Efficient Technologies for Sustainability*, pages 350–355, 2016.

[34] C. Pache, J. Maeght, B. Seguinot, A. Zani, S. Lumbreras, A. Ramos, S. Agapoff, L. Warland, L. Rouco, and P. Panciatici. Enhanced pan-european transmission planning methodology. In *IEEE Power Energy Society General Meeting*, July 2015.

[35] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[36] M. Pfetsch, A. Fügenschuh, B. Geißler, N. Geißler, R. Gollmer, B. Hiller, J. Humpola, T. Koch, T. Lehmann, A. Martin, A. Morsi, J. Rövekamp, L. Schewe, M Schmidt, R. Schultz, R. Schwarz, J. Schweiger, C. Stangl, M. Steinbach, S. Vigerske, and B. Willert. Validation of nominations in gas network optimization: Models, methods, and solutions. *Optimization Methods and Software*, 30(1):15–53, 2015.

[37] J. Tong and H. Ni. Look-ahead multi-time frame generator control and dispatch method in PJM real time operations. In *IEEE Power and Energy Society General Meeting*, July 2011.

[38] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *NIPS*, pages 2692–2700, 2015.

[39] Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *AAAI*, volume 33, pages 1658–1665, 2019.

[40] I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480, 2009.

[41] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P Gummadi. Fairness constraints: A flexible approach for fair classification. *JMLR*, 20(75):1–42, 2019.

[42] A. Zlotnik, M. Chertkov, and S. Backhaus. Optimal control of transient flow in natural gas networks. In *CDC*, pages 4563–4570, 2015.

A Energy System Case Studies

This section extends the description of the energy applications for evaluating the Lagrangian Dual Framework: Optimal power flow in electricity networks and the Optimal compressor controls in gas networks. Both problems are nonlinear and nonconvex.

Model 1 $\mathcal{O}_{\text{OPF}}$: AC Optimal Power Flow

$$\textbf{variables:} \quad S_i^g, V_i \;\; \forall i \in N, \;\; S_{ij}^f \;\; \forall (i,j) \in E \cup E^R$$

$$\textbf{minimize:} \quad \mathcal{O}(\boldsymbol{S^d}) = \sum_{i \in N} c_{2i}(\Re(S_i^g))^2 + c_{1i}\Re(S_i^g) + c_{0i} \tag{18}$$

$$\textbf{subject to:} \quad \angle V_i = 0, \;\; i \in N \tag{19}$$

$$v_i^l \leqslant |V_i| \leqslant v_i^u \;\; \forall i \in N \tag{20}$$

$$\theta_{ij}^l \leqslant \angle(V_i V_j^*) \leqslant \theta_{ij}^u \;\; \forall (i,j) \in E \tag{21}$$

$$S_i^{gl} \leqslant S_i^g \leqslant S_i^{gu} \;\; \forall i \in N \tag{22}$$

$$|S_{ij}^f| \leqslant s_{ij}^{fu} \;\; \forall (i,j) \in E \cup E^R \tag{23}$$

$$S_i^g - S_i^d = \textstyle\sum_{(i,j) \in E \cup E^R} S_{ij}^f \;\; \forall i \in N \tag{24}$$

$$S_{ij}^f = Y_{ij}^* |V_i|^2 - Y_{ij}^* V_i V_j^* \;\; \forall (i,j) \in E \cup E^R \tag{25}$$

A.1 Optimal Power Flow

Optimal Power Flow (OPF) is the problem of finding the best generator dispatch to meet the demands in a power network, while satisfying challenging transmission constraints such as the nonlinear nonconvex AC power flow equations and also operational limits such as voltage and generation bounds. Finding good OPF predictions are important, as a 5% reduction in generation costs could save billions of dollars (USD) per year [10]. In addition, the OPF problem is a fundamental building bock of many applications, including security-constrained OPFs [31]), optimal transmission switching [19], capacitor placement [6], and expansion planning [33].

Typically, generation schedules are updated in intervals of 5 minutes [37], possibly using a solution to the OPF solved in the previous step as a starting point. In recent years, the integration of renewable energy in sub-transmission and distribution systems has introduced significant stochasticity in front and behind the meter, making load profiles much harder to predict and introducing significant variations in load and generation. This uncertainty forces system operators to adjust the generators setpoints with increasing frequency in order to serve the power demand while ensuring stable network operations. However, the resolution frequency to solve OPFs is limited by their computational complexity. To address this issue, system operators typically solve OPF approximations such as the linear DC model (DC-OPF). While these approximations are more efficient computationally, their solution may be sub-optimal and induce substantial economical losses, or they may fail to satisfy the physical and engineering constraints.

Similar issues also arise in expansion planning and other configuration problems, where plans are evaluated by solving a massive number of multi-year Monte-Carlo simulations at 15-minute intervals [34, 16]. Additionally, the stochasticity introduced by renewable energy sources further increases the number of scenarios to consider. Therefore, modern approaches recur to the linear DC-OPF approximation and focus only on the scenarios considered most pertinent [34] at the expense of the fidelity of the simulations.

A power network $\boldsymbol{\mathcal{N}}$ can be represented as a graph (N, E), where the nodes in N represent buses and the edges in E represent lines. The edges in E are directed and E^R is used to denote those arcs in E but in reverse direction. The AC power flow equations are based on complex quantities for current I, voltage V, admittance Y, and power S, and these equations are a core building block in many power system applications. Model 1 shows the AC OPF formulation, with variables/quantities shown in the complex domain. Superscripts u and l are used to indicate upper and lower bounds for variables. The objective function $\mathcal{O}(\boldsymbol{S^g})$ captures the cost of the generator dispatch, with $\boldsymbol{S^g}$ denoting the vector of generator dispatch values $(S_i^g \mid i \in N)$. Constraint (19) sets the reference angle to zero for the slack bus $i \in N$ to eliminate numerical symmetries. Constraints (20) and (21) capture the voltage and phase angle difference bounds. Constraints (22) and (23) enforce the generator output and line flow limits. Finally, Constraints (24) capture Kirchhoff's Current Law and Constraints (25) capture Ohm's Law.

Table 6 describes the power network benchmarks used in main text, including the number of buses $|\mathcal{N}|$, transmission lines/transformers $|\mathcal{E}|$, loads l, and generators g.

Test Case	$\|\mathcal{N}\|$	$\|\mathcal{E}\|$	l	g
30_ieee	30	82	21	2
118_ieee	118	372	99	19
300_ieee	300	822	201	57

Table 6: The Power Networks DataSet.

Model 2 $\mathcal{O}_{\text{OGF}}$: Optimal Gas Flow

$$\textbf{variables:}\ p_i, q_i\ \forall i \in \mathcal{J},\ q_{ij}\ \forall (i,j) \in \mathcal{P},\ R_{ij}\ \forall (i,j) \in \mathcal{C}$$

$$\textbf{minimize:}\ \mathcal{O}(\boldsymbol{q}) = \sum_{(i,j)\in C} \mu^{-1} |q_{ij}| (\max\{R_{ij}, 1\}^{2(\gamma-1)/\gamma} - 1) \tag{26}$$

$$\textbf{subject to:}\ \textstyle\sum_{(i,j)\in\mathcal{P}} q_{ij} - \sum_{(j,i)\in\mathcal{P}} q_{ji} = q_i,\ \ \forall i \in \mathcal{J} \tag{27}$$

$$ {p_i}^l \leqslant p_i \leqslant {p_i}^u\ \ \forall i \in N,\ \ {q_{ij}}^l \leqslant q_{ij} \leqslant {q_{ij}}^u\ \ \forall (i,j) \in \mathcal{P} \tag{28}$$

$$ {R_{ij}}^l \leqslant R_{ij} \leqslant {R_{ij}}^u\ \ \forall (i,j) \in \mathcal{C} \tag{29}$$

$$p_i = p_i^T\ \ \forall i \in \mathcal{J}^B, q_i = 0\ \ \forall i \in \mathcal{J}^T, q_i = q_i^d\ \ i \in \mathcal{J}^D \tag{30}$$

$$R_{ij}^2 p_i^2 - p_j^2 = L_{ij} \frac{\lambda a^2}{D_{ij} A_{ij}^2} q_{ij} |q_{ij}|\ \ \forall (i,j) \in \mathcal{C} \tag{31}$$

$$p_i^2 - p_j^2 = L_{ij} \frac{\lambda a^2}{D_{ij} A_{ij}^2} q_{ij} |q_{ij}|\ \ \forall (i,j) \in \mathcal{P} - \mathcal{C} \tag{32}$$

A.2 Optimal Compressor Optimization

Optimal Gas Flow (OGF) is the problem of finding the best compression control to maintain pressure requirements in a natural gas pipeline system. Similar to the OPF problem, the gas flow problem is a non-convex non-linear optimization problem, with challenging nonconvex function to measure the costs of compressors. It is also a fundamental building block for many gas pipeline problems, including: gas pipeline expansion planning [8], dynamic compressor optimization [29], and joint gas-grid transmission planning problem [7].

Historically, natural gas demands came from utilities or large industrial customers whose demands are predictable with little variations. These demands are often traded using day-ahead contracts. Therefore, operators can often assume that injections and withdrawals would be similar to the past and re-use precomputed control set-points. In the past decade, the increasing penetration of renewable energies into power systems has driven an increase in installations of gas-powered electric generators. Gas-powered generators can start up and shut down several times a day, and also capable to rapidly adjust their production to balance the fluctuation of renewable energy sources. However, the growing use of gas-powered generators implies substantial intra-day high-volume gas fluctuations, and has prompted concerns within the industries. Similar to OPF, the resolution frequency to solve OGF problem is limited by the computational complexity of the system, and system operators typically solve OGF approximations instead such as reduced-order models [42] or convex relaxations [8]. These approximations are more efficient computationally, but can be sub-optimal and/or fail to capture the physical and operational limits.

A natural gas network can be represented as a directed graph $\boldsymbol{\mathcal{N}} = (\mathcal{J}, \mathcal{P})$, where a node $i \in \mathcal{J}$ represents a junction point and an arc represents a pipeline $(i,j) \in \mathcal{P}$ represent the edges. Compressors ($\mathcal{C} \subseteq \mathcal{P}$) are installed in a subset of the pipelines for boosting the gas pressure p in order to maintain pressure requirements for gas flow q. The set $\mathcal{J}^D$ of gas demands and the set $\mathcal{J}^T$ of transporting nodes are modeled as junction points, with net gas flow q_i set to the gas demand (q_i^d) and zero respectively. For simplicity, the paper assumes no pressure regulation and losses within junction nodes and gas flow/flux are conserved throughout the system. A subset $\mathcal{J}^B \in \mathcal{J}$ of the nodes are regulated with constant pressure p_i^T. The length of pipe (i,j) is denoted by L_{ij}, its diameter by D_{ij}, and its cross-sectional area by A_{ij}. Universal quantities include isentropic coefficient γ, compressor efficiency factor μ, sound speed a, and gas friction factor λ. Model 2 depicts the OGF formulation. The objective function $\mathcal{O}(\boldsymbol{q})$ captures the compressor costs using the compressor control values $(R_{ij} \mid (i,j) \in \mathcal{C})$. Constraints (27) capture the flow conversation equations. Constraints (28) and (29)

Test Case	$\lvert\mathcal{J}\rvert$	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{J}^D\rvert$
24-pipe	25	24	5	8
40-pipe	40	45	6	26
135-pipe	135	170	10	19

Table 7: The Gas Networks DataSet.

capture the pressure, flux flow, and compressor control bounds. Constraints (30) set the boundary conditions for the demands and the regulated pressures. Finally, Constraints (31) and (32) capture the steady-state isothermal gas flow equation.

Table 7 describes the gas network benchmarks used in the main text, including the number of junctions $|\mathcal{J}|$, pipelines $|\mathcal{P}|$, compressors $|\mathcal{C}|$, and active gas loads $|\mathcal{J}^D|$.

B Learning Model Details

This section describes the modeling details of the proposed LDF related to the energy applications described in the previous section.

B.1 Lagrangian relaxation

This paper uses a Lagrangian relaxation approach based on constraint violations [20] used in generalized augmented Lagrangian relaxation [23]. To fully describe the modeling details for the OPF and OGF problems, an optimization problem O definition will be slightly expanded as:

$$\begin{aligned} \textbf{minimize:} \quad & f(\boldsymbol{y}) \\ \textbf{subject to:} \quad & h(\boldsymbol{y}) = 0 \\ & g(\boldsymbol{y}) \leqslant 0 \end{aligned}$$

where equality constraints h and inequality constraints g are separated. The Lagrangian relaxation of O is given by

$$\textbf{minimize:} \quad f(\boldsymbol{y}) + \lambda_h h(\boldsymbol{y}) + \lambda_g g(\boldsymbol{y})$$

where λ_h and $\lambda_g \geqslant 0$ are the Lagrangian multipliers for the equality constraints and inequality constraints. In contrast, the violation-based Lagrangian relaxation is

$$\textbf{minimize:} \quad f(\boldsymbol{y}) + \lambda_h |h(\boldsymbol{y})| + \lambda_g \max(0, g(\boldsymbol{y}))$$

with $\lambda_h, \lambda_g \geqslant 0$. In other words, the traditional Lagrangian relaxation exploits the satisfiability degrees of constraints, while the violation-based Lagrangian relaxation is expressed in terms of violation degrees. The satisfiability degree of an inequality constraint measures how well the constraint is satisfied, with negative values representing the slack and positive values representing violations, while the violation degree is always non-negative and represents how much the constraint is violated. More formally, the satisfiability degree of a constraint $c : \mathbb{R}^n \rightarrow$ Bool is a function $\sigma_c : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $\sigma_c(\boldsymbol{y}) \leqslant 0 \equiv c(\boldsymbol{y})$. The violation degree of a constraint $c : \mathbb{R}^n \rightarrow$ Bool is a function $\nu_c : \mathbb{R}^n \rightarrow \mathbb{R}^+$ such that $\sigma_c(\boldsymbol{y}) \equiv 0 \equiv c(\boldsymbol{y})$. For instance, for a linear constraints $c(\boldsymbol{y})$ of type $A\boldsymbol{y} \geqslant b$, the *satisfiability degree* is defined as

$$\sigma_c(\boldsymbol{y}) \equiv \boldsymbol{b} - A\boldsymbol{y}$$

and the *violation degrees* for inequality and equality constraints are specified by

$$\nu_c^{\geqslant}(\boldsymbol{y}) = \max\left(0, \sigma_c(\boldsymbol{y})\right) \qquad \nu_c^{=}(\boldsymbol{y}) = \left|\sigma_c(\boldsymbol{y})\right|.$$

Although the resulting term is not differentiable (but admits subgradients), computational experiments indicated that violation degrees are more appropriate for prediction than satisfiability degrees. Even though in theory all the constraints can be naively incorporated into the learning model by constructing the violation degree metric functions, in practice not all the constraints are needed. For example, if a constraint is guaranteed to be satisfied by the input data (e.g. (13) in Model 2), the constraint can be omitted as the violation degree is always zero. On the other hand, if the satisfiability of a constraint depends on the prediction or the constraint is used to compute an indirect prediction (e.g. Ohm's Law (8) in Model 1), the violation degree of the constraint can be measured directly against the ground truth.

B.2 OPF satisfiability and violation degrees

Given the predicted values: $\hat{\boldsymbol{V}} = \hat{\boldsymbol{v}}\angle\hat{\boldsymbol{\theta}}$ for voltages, $\hat{\boldsymbol{S}}^g = \hat{\boldsymbol{p}}^g + i\hat{\boldsymbol{q}}^g$ for generation dispatches, and $\hat{\boldsymbol{S}}^f = \hat{\boldsymbol{p}}^f + i\hat{\boldsymbol{q}}^f$ for lines/transformers flows, this section extends the main paper by reporting the complete set of satisfiability $\sigma(\cdot)$ and violation degrees $\nu(\cdot)$ for the OPF problem.

$$\sigma_{3a}(\hat{v}_i) = v_i^l - \hat{v}_i \qquad \forall i \in N \qquad \sigma_{3b}(\hat{v}_i) = \hat{v}_i - v_i^u \qquad \forall i \in N$$

$$\sigma_{4a}(\hat{\theta}_{ij}) = (\hat{\theta}_j - \hat{\theta}_i) - \theta_{ij}^{\Delta} \qquad \forall (ij) \in E \qquad \sigma_{4b}(\hat{\theta}_{ij}) = (\hat{\theta}_i - \hat{\theta}_j) - \theta_{ij}^{\Delta} \forall (ij) \in E$$

$$\sigma_{5a}(\hat{p_i}^g) = p_i^{gl} - \hat{p}_i^g \qquad \forall i \in N \qquad \sigma_{5b}(\hat{p_i}^g) = \hat{p}_i^g - p_i^{gu} \qquad \forall i \in N$$

$$\sigma_{5c}(\hat{q_i}^g) = q_i^{gu} - \hat{q}_i^g \qquad \forall i \in N \qquad \sigma_{5d}(\hat{q_i}^g) = \hat{q}_i^g - q_i^{gu} \qquad \forall i \in N$$

$$\sigma_6(\hat{p}_{ij}^f, \hat{q}_{ij}^f) = (\hat{p}_{ij}^f)^2 + (\hat{q}_{ij}^f)^2 - s_{ij}^u \, \forall (ij) \in E \cup E^R$$

$$\sigma_{7a}(\hat{p}_i^g, p_i^d, \hat{\boldsymbol{p}}^f) = \sum_{(ij)\in E} \hat{p}_{ij}^f - (\hat{p}_i^g - p_i^d) \qquad \forall i \in N$$

$$\sigma_{7b}(\hat{q}_i^g, q_i^d, \hat{\boldsymbol{q}}^f) = \sum_{(ij)\in E} \hat{q}_{ij}^f - (\hat{q}_i^g - q_i^d) \qquad \forall i \in N$$

$$\sigma_{8a}(\hat{p}_{ij}^f, p_{ij}^f) = \hat{p}_{ij}^f - p_{ij}^f \qquad \forall (ij) \in E \quad \sigma_{8b}(\hat{q}_{ij}^f, q_{ij}^f) = \hat{q}_{ij}^f - q_{ij}^f \qquad \forall (ij) \in E$$

Functions σ_{3a} and σ_{3b} correspond to Constraints (3) and capture the distance of the voltage predictions $\hat{v}_i$ and its bounds. Similarly, σ_{4a} and σ_{4b} correspond to (4) and measure the difference between voltage phase angle differences and its bound. Functions σ_{5a} to σ_{5d} correspond to (5) and describe the distance of the generator active and reactive dispatch predictions from their upper and lower bounds. σ_6 corresponds to (6) and measures distance between the squared apparent power and its bound. Functions σ_{7a} and σ_{7b} relate to the Kirchhoff Current Law (7) and measure the power flow violations at every bus (i.e. bus injection violations). Finally, functions σ_{8a} and σ_{8b} measure the deviation of the indirect predicted flows $\hat{p}_{ij}^f$ and $\hat{q}_{ij}^f$ to their ground truth values p_{ij}^f and q_{ij}^f. Note that $\hat{p}_{ij}^f$ and $\hat{q}_{ij}^f$ are not direct predictions from the output of the DNN. These quantities are indirect predictions computed using the voltage predictions.

The violation degrees associated to the satisfiability degree above are defined below.

$$\nu_3(\hat{\boldsymbol{v}}) = \frac{1}{|N|} \sum_{i\in N} \left(\nu_c^{\geqslant}\left(\sigma_{3a}(\hat{v}_i)\right) + \nu_c^{\geqslant}\left(\sigma_{3b}(\hat{v}_i)\right)\right)$$

$$\nu_4(\hat{\boldsymbol{\theta}}) = \frac{1}{|E|} \sum_{(ij)\in E} \left(\nu_c^{\geqslant}\left(\sigma_{4a}(\hat{\theta}_{ij})\right) + \nu_c^{\geqslant}\left(\sigma_{4b}(\hat{\theta}_{ij})\right)\right)$$

$$\nu_{5a}(\hat{\boldsymbol{p}}^g) = \frac{1}{|N|} \sum_{i\in N} \left(\nu_c^{\geqslant}\left(\sigma_{5a}(\hat{p}_i^g)\right) + \nu_c^{\geqslant}\left(\sigma_{5b}(\hat{p}_i^g)\right)\right) \qquad \nu_{5b}(\hat{\boldsymbol{q}}^g) = \frac{1}{|N|} \sum_{i\in N} \left(\nu_c^{\geqslant}\left(\sigma_{5c}(\hat{q}_i^g)\right) + \nu_c^{\geqslant}\left(\sigma_{5d}(\hat{q}_i^g)\right)\right)$$

$$\nu_6(\hat{\boldsymbol{p}}^f, \hat{\boldsymbol{q}}^f) = \frac{1}{|E|} \sum_{(ij)\in E} \nu_c^{\geqslant}\left(\sigma_6(\hat{p}_{ij}^f, \hat{q}_{ij}^f)\right)$$

$$\nu_{7a}(\hat{\boldsymbol{p}}^g, \boldsymbol{p}^d, \boldsymbol{p}^f) = \frac{1}{|E|} \sum_{(ij)\in E} \nu_c^{=}\left(\sigma_{7a}(\hat{p}_i^g, p_i^d, \tilde{\boldsymbol{p}}^f)\right) \qquad \nu_{7b}(\hat{\boldsymbol{q}}^g, \boldsymbol{q}^d, \boldsymbol{q}^f) = \frac{1}{|E|} \sum_{(ij)\in E} \nu_c^{=}\left(\sigma_{7b}(\hat{q}_i^g, q_i^d, \tilde{\boldsymbol{q}}^f)\right)$$

$$\nu_{8a}(\hat{\boldsymbol{p}}^f, \boldsymbol{p}^f) = \frac{1}{|E|} \sum_{(ij)\in E} \nu_c^{=}\left(\sigma_{8a}(\hat{p}_{ij}^f, p_{ij}^f)\right) \qquad \nu_{8b}(\hat{\boldsymbol{q}}^f, \boldsymbol{q}^f) = \frac{1}{|E|} \sum_{(ij)\in E} \nu_c^{=}\left(\sigma_{8b}(\hat{q}_{ij}^f, q_{ij}^f)\right).$$

These functions capture the average deviation by which the prediction violates the associated constraint. The violations degrees define penalties that will be used to enrich the DNN loss function to encourage their satisfaction.

B.3 OGF satisfiability and violation degrees

Given the predicted values: $\hat{\boldsymbol{R}}$ for compression controls, $\hat{\boldsymbol{p}}$ for pressure values, $\hat{\boldsymbol{q}}$ for natural gas supply, and $\hat{\boldsymbol{q}}^f$ for pipelines flows, this section extends the main paper by reporting the complete set of satisfiability $\sigma(\cdot)$ and violation degrees $\nu(\cdot)$ for the OGF problem.

$$\sigma_{10}(\hat{\boldsymbol{q}}^f, \hat{q}_i) = \sum_{(i,j)\in\mathcal{P}} \hat{q}_{ij}^f - \sum_{(j,i)\in\mathcal{P}} \hat{q}_{ji}^f - \hat{q}_i \qquad \forall i \in \mathcal{J}$$

$$\sigma_{11a}(\hat{p}_i) = p_i^l - \hat{p}_i \qquad \forall i \in \mathcal{J} \qquad \sigma_{11b}(\hat{p}_i) = \hat{p}_i - p_i^u \qquad \forall i \in \mathcal{J}$$
$$\sigma_{11c}(\hat{q}_{ij}^f) = q_{ij}^l - \hat{q}_{ij} \qquad \forall (i,j) \in \mathcal{P} \qquad \sigma_{11d}(\hat{q}_{ij}^f) = \hat{q}_{ij} - q_{ij}^u \qquad \forall (i,j) \in \mathcal{P}$$
$$\sigma_{12a}(\hat{R}_{ij}) = R_{ij}^l - \hat{R}_{ij} \qquad \forall (i,j) \in \mathcal{C} \qquad \sigma_{12b}(\hat{R}_{ij}) = \hat{R}_{ij} - R_{ij}^u \qquad \forall (i,j) \in \mathcal{C}$$
$$\sigma_{14}(\hat{q}_{ij}^f, q_{ij}^f) = \hat{q}_{ij}^f - q_{ij}^f \qquad \forall (i,j) \in \mathcal{P}$$

Function σ_{10} corresponds to Constraints (10) and measure the gas flow violations at every junction. Functions $\sigma_{11a} - \sigma_{11d}$ relate to Constraints (11) and capture the distance of the pressure and gas flow predictions $\hat{p}_i$ and $\hat{q}_{ij}$ to their bounds. Similarly, σ_{12a} and σ_{12b} correspond to Constraints (12) and capture the distance of the compression control $\hat{R}_{ij}$ to its bounds. Finally, functions σ_{14} measure the deviation of the indirect predicted flows $\hat{q}_{ij}^f$ to its ground truth value q_{ij}^f. Similar to OPF, $\hat{q}_{ij}^f$ are not direct predictions from the output of the DNN and computed using the pressure and compressor control predictions. Note that Constraints (13) are skipped since

The violation degrees associated to the satisfiability degree above are defined below.

$$\nu_{10}(\hat{\boldsymbol{q}}^f, \hat{\boldsymbol{q}}) = \frac{1}{|\mathcal{J}|} \sum_{i \in \mathcal{J}} \nu_c^{=}\big(\sigma_{10}(\hat{\boldsymbol{q}}^f, \hat{q}_i)\big)$$
$$\nu_{11a}(\hat{\boldsymbol{p}}) = \frac{1}{|\mathcal{J}|} \sum_{i \in \mathcal{J}} \big(\nu_c^{\geqslant}\big(\sigma_{11a}(\hat{p}_i)\big) + \nu_c^{\geqslant}\big(\sigma_{11b}(\hat{p}_i)\big)\big)$$
$$\nu_{11b}(\hat{\boldsymbol{q}}^f) = \frac{1}{|\mathcal{P}|} \sum_{i \in P} \Big(\nu_c^{\geqslant}\big(\sigma_{11c}(\hat{q}_{ij}^f)\big) + \nu_c^{\geqslant}\big(\sigma_{11d}(\hat{q}_{ij}^f)\big)\Big)$$
$$\nu_{12}(\hat{\boldsymbol{R}}) = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} \Big(\nu_c^{\geqslant}\big(\sigma_{12a}(\hat{R}_{ij})\big) + \nu_c^{\geqslant}\big(\sigma_{12b}(\hat{R}_{ij})\big)\Big)$$
$$\nu_{14}(\hat{\boldsymbol{q}}^f, \boldsymbol{q}^f) = \frac{1}{|\mathcal{P}|} \sum_{(i,j) \in \mathcal{P}} \nu_c^{=}\big(\sigma_{14}(\hat{q}_{ij}^f, q_{ij}^f)\big)$$

C Additional Experiments: Classification under Fairness Constraints

This section reports additional results on LDF for constrained predictor problems on a fairness application domain.

Figure 1 illustrates the evolution of the accuracy and DT metrics for all the models evaluated at the increasing of the number of epochs. The results are reported on the validation sets and the figure illustrates the results for the Adult (top) Default (middle) and Bank (bottom) datasets. A clear trend appears: The Lagrangian Dual method $\mathcal{M}_C^D$, reports a lower DT scores, and this happens early in the training stages. Surprising, on the Bank dataset, the DT score could be reduced quite significantly without much loss on accuracy terms.

C.1 Relaxed fairness constraints

Next, this section reports an extended evaluation on a *relaxed* notion of the fairness constraints. To do so, the experiments allow the constraint a slack $0 \leqslant \Delta_f \leqslant \epsilon$ ($\epsilon \geqslant 0$). The idea is to exploit such relaxation to allow the model to achieve even better accuracy.

The experiments report the DT and accuracy scores attained by the $\mathcal{M}_C^D$ model on the Adult, Default, and Bank datasets using several levels of constraint relaxation, which act on the slack ϵ. For each dataset, the experiments compute the DI-score obtained in the original data, i.e $DI_{\text{org}} = |Pr(y(x) = 1|z(x) = 1) - Pr(y(x) = 1|z(x) = 0)$ and choose ϵ to be 5%, 20% and 50% of this quantity.

Table 8 reports the results. It can be observed that the DI-score increases with the increasing of the constraint relaxation ϵ. At the same time, the accuracy decreases with the increasing of value ϵ.

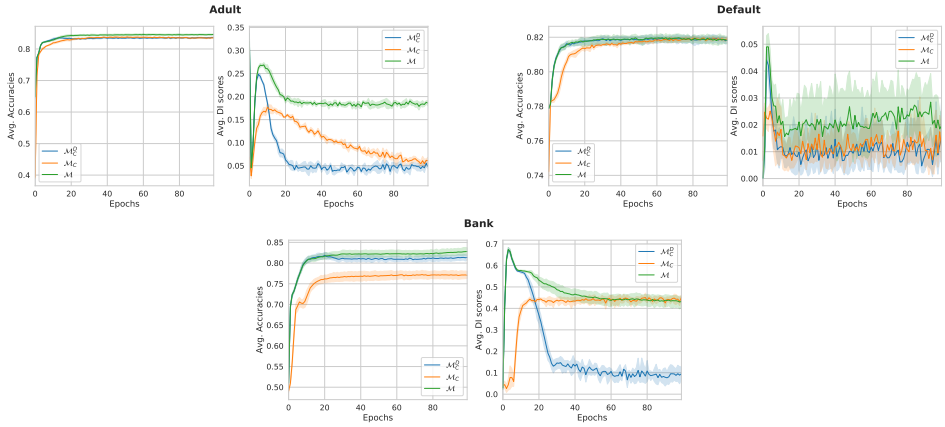

Figure 1: Average (Avg.) of Accuracies/DT-scores on the validation sets during optimization.

Dataset	$\epsilon = 5\%\text{DI}_{\text{org}}$		$\epsilon = 20\%\text{DI}_{\text{org}}$		$\epsilon = 50\%\text{DI}_{\text{org}}$	
	Acc.	DT	Acc.	DT	Acc.	DT
Adult	0.8336	0.0567	0.8339	0.0586	0.8358	0.0812
Default	0.8164	0.0073	0.8162	0.0087	0.8159	0.0092
Bank	0.8161	0.1556	0.8215	0.2437	0.8251	0.3372

Table 8: Accuracy/DT-score of $\mathcal{M}_C^D$ under different relaxed fairness parameters ϵ