Forecasting Electricity Prices with Decision-Focused Learning for Storage Optimization

Ruben Smets, Erik Delarue
KU Leuven, mechanical engineering

Jean-François Toubeau, Mihaly Dolanyi
UMons, electrical engineering

Kenneth Bruninix
TU Delft, TPM - E&I

***Abstract*—Energy Storage Systems (ESS) play a crucial role in managing renewable energy variability. Forecast-informed optimization is typically used to maximize ESS profit in electricity markets. Whereas traditional forecaster training methods use accuracy-based loss functions, Decision-Focused Learning uses a task-aware loss function with the aim of improving ESS profits. This can be achieved by integrating the downstream optimization in the forecaster training procedure. A common task-aware loss function is the Smart Predict-then-Optimize (SPO+) loss. However, its current implementation is prone to overfitting and is limited to linear forecasting models. Here, we extend the SPO+ framework to neural network forecasters with non-linear activation functions while introducing an interior-point training method to mitigate overfitting risks. When applied to an ESS participating in the day-ahead market, our approach outperforms both traditional and other decision-focused benchmarks in terms of obtained ESS profits.**

***Index Terms*—Electricity markets, Energy Storage Systems, Price forecasting, Interior point method, Neural networks.**

I. Introduction

To achieve a low-carbon electricity system with a high penetration of renewable energy sources, Energy Storage Systems (ESS) are a key technology in mitigating the intermittency of renewable generation. To maximize profits of ESS in electricity markets, researchers have adopted forecast-informed optimization. Here, forecasts of electricity prices serve as parameters in optimization programs to find the optimal (dis)charge schedule. This has been implemented in the Real-Time (RT) market [1], the Day-Ahead (DA) market [2], and in a multi-market setting [3], [4].

A. Related work

Because of their high predictive power, the use of Machine Learning (ML) models, and specifically Neural Networks (NN) has become common practice for forecasting electricity prices [5]. Traditional NN training procedures minimize a statistical loss function such as the mean absolute error [6] or the pinball loss [1]. In contrast, Decision-Focused Learning (DFL) acknowledges that the forecast is used in a subsequent, *downstream*, optimization problem, i.e., the ESS profit maximization problem. To that end, a task-aware loss function is introduced. A common task-aware loss function is regret [7], [8], which is the difference between Perfect Foresight (PF) profit and the profit obtained under uncertainty. The main difficulty with such a loss function is that the gradient descent training algorithm requires a well-defined gradient of the loss function with respect to the forecaster parameters. With a task-aware loss function like regret, this involves gradients of optimized decisions with respect to the forecast, which are highly non-trivial to calculate.

One approach to enable backpropagation through the downstream optimization problem is to apply implicit differentiation to its Karush-Kuhn-Tucker (KKT) conditions. This technique can be applied to convex non-linear optimization problems [9]–[11]. The discontinuous nature of the output of (mixed-integer) linear programs is, however, well-known to pose difficulties for this method. One work-around is to add a non-linear smoothing term to the objective function, usually with a quadratic [12], [13] or log-barrier [8] term. A remaining problem of this method is that they easily get stuck in local optima during the training procedure.

Another promising approach was proposed in [7], involving a convex surrogate of the regret loss function, coined ”SPO+” loss. Using that task-aware loss, DFL training can be achieved in two ways. Firstly, the convexity of the SPO+ loss is leveraged to use a subgradient training method. While this method can be used for price forecasting for ESS participating in the day-ahead market [2], it also suffers from getting stuck in local optima. Secondly, using duality theory, the training problem can be reformulated as a single-level optimization program. However, this approach has only been implemented for linear forecasters, and is prone to overfitting as the solution of the optimization yields the forecaster with the highest profit on the train data, disregarding its performance on unseen data.

B. Contributions

In this paper, we build upon the SPO+ reformulation method by solving the single-level reformulation of the forecaster training problem with an Interior Point (IP) method. This approach allows for tracking the validation performance along the steps in the iterative training procedure. Since IP methods can be used for non-convex optimization problems, this method also allows for extending the forecaster to a broader class of non-linear ML models. The scientific contribution of this work is threefold:

- We develop an interior point-based neural network training algorithm that iteratively tracks the task-specific validation performance, thus reducing the risk of overfitting.
- We extend the original SPO+ reformulation of the forecaster training problem to accommodate NNs.

- In an optimal scheduling problem of a storage system under price uncertainty using real-life data, we demonstrate the effectiveness of the method by showcasing higher out-of-sample profits compared to both traditional forecasting methods and state-of-the-art DFL models.

II. Decision-Focused Learning: Background

In its most generic form, an ESS profit maximization problem can be written as:

$$\begin{aligned} \underset{c,d,s}{\text{maximize}} \quad & \sum_{t=1}^{T} \hat{\lambda}_t \left(d_t \eta^+ - \frac{c_t}{\eta-} \right) \\ \text{subject to} \quad & 0 \le d_t + c_t \le \overline{P} && \forall t \\ & s_{t+1} = s_t + c_t - d_t && \forall t \\ & 0 \le s_t \le \overline{SoC} && \forall t \\ & s_0 = s_T = \overline{SoC}/2, \end{aligned} \tag{1}$$

with c, d and s the charge, discharge, the state of charge vectors. The maximum power and state of charge are represented by $\overline{P}$ and $\overline{SoC}$. The look-ahead horizon over which the ESS optimizes is T. In the objective function, $\hat{\lambda}$ indicates the vector of price forecasts, and $\eta^{+/-}$ the (dis)charge efficiency. By considering a cost which is the opposite of the price forecast, by replacing the equality constraints with two inequalities, and by aggregating the decision variables in a single vector x, this linear problem is recast to the standard form:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \hat{c}^T x \\ \text{subject to} \quad & Ax \ge b, \end{aligned} \tag{2}$$

where $\hat{c}$ is the forecasted cost. A forecaster $f : \mathbb{R}^n \to \mathbb{R}^m$, maps a vector of input features (relevant contextual information) $\alpha \in \mathbb{R}^n$ to the predictions of interest $\hat{c} \in \mathbb{R}^m$. The problem of training the forecaster, also referred to as Empirical Risk Minimization (ERM) problem, reads:

$$\min_{\theta \in \Theta} \sum_{i \in \mathcal{I}_{tr}} \mathcal{L}(c_i, f(\alpha_i; \theta)) + \rho \Omega(f), \tag{3}$$

with θ the trainable parameters of the forecaster, $\mathcal{I}_{tr}$ the train data set, (α_i, c_i) feature-label instances of the labeled training set, $\mathcal{L}$ the loss function and Ω a regularization function that helps avoiding to overfit on the train data. In DFL, a task-aware loss function is adopted. This is typically regret:

$$r(c, \hat{c}) = c^T x^*(\hat{c}) - c^T x^*(c), \tag{4}$$

which is the difference between the ex-post objective value obtained with the optimized decisions $x^*(\hat{c})$ from the forecasts, found by solving (2), and the objective value that would have been achieved if the true values c were known. The regret quantifies the performance loss due to imperfect forecasts and should be minimized. Embedding regret within the loss function in (3), the ERM problem becomes a bi-level optimization problem:

$$\min_{\theta \in \Theta} \left[\sum_{i \in \mathcal{I}_{tr}} c_i^T \cdot \underset{x \in \{y | Ay \ge b\}}{\arg\min} \left(f(\alpha_i; \theta)^T x \right) \right], \tag{5}$$

which is well-known to be intractable for large-scale problems.

A. Gradient-based methods

When using non-linear activation functions in the NN architecture, the traditional ERM problem (3) becomes highly non-convex (regardless of the loss function), and is typically solved with gradient descent. In this setting, the NN parameters are iteratively updated by calculating the gradient of the loss function with respect to that parameter, and taking a step in the direction of steepest descent:

$$\theta \leftarrow \theta - \psi \frac{\partial \mathcal{L}}{\partial \theta}, \tag{6}$$

where ψ represents the learning rate.

When regret (4) is used as the loss function, the gradients can be calculated using the chain rule:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial r}{\partial x^*} \frac{\partial x^*}{\partial \hat{c}} \frac{\partial \hat{c}}{\partial \theta}. \tag{7}$$

The first and third factors are straightforwardly calculated with traditional techniques. The second factor, $\frac{\partial x^*}{\partial \hat{c}}$, being the gradient "through" the optimization program, is more complex but can be calculated with implicit differentiation of the KKT optimality conditions [9]. Considering a downstream optimization problem of the form $\min_{x \in S} g(x, \hat{c})$, this leads to the following set of equations:

$$\begin{bmatrix} \frac{\partial^2 g}{\partial x^* \partial \hat{c}}(x^*, \hat{c}) \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{\partial^2 g}{\partial x^{*2}}(x^*, \hat{c}) & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial x^*}{\partial \hat{c}} \\ \frac{\partial \hat{\lambda}}{\partial \hat{c}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{8}$$

To solve for $\frac{\partial x^*}{\partial \hat{c}}$, the Hessian of the objective function should be non-zero, which is not the case for a (mixed-integer) linear optimization problem, like (1). To overcome this, smoothing terms in the objective of the downstream decision problem have been proposed in the form of a quadratic [12] or a log-barrier term [8].

B. SPO+ surrogate loss

As an alternative to smoothing terms, a convex surrogate of the regret loss function is proposed in [7]. This surrogate, named the "SPO+" loss, is defined as: $l^{SPO+} = \max_{x \in S}\{c^T x - 2\hat{c}^T x\} + 2\hat{c}^T x^*(c) - z^*(c)$. It is shown that minimizing this SPO+ loss corresponds to minimizing the regret function. The SPO+ loss function is convex in $\hat{c}$, which allows the authors to provide an expression for a subgradient:

$$2\left(x^*(c) - x^*(2\hat{c} - c)\right) \in \partial_{\hat{c}} l^{SPO+}(c, \hat{c}). \tag{9}$$

This enables the training of NNs to minimize the surrogate regret loss using the subgradient method [14]. A second, reformulation-based, approach is provided to train a forecaster with the SPO+ loss function in [7]. By substituting l^{SPO+} in Eq. (3), and leveraging duality theory, the ERM problem is reformulated as a single-level linear optimization program when the forecaster is assumed to be a linear function of the input features, i.e., $\hat{c} = B\alpha$. The resulting ERM problem reads:

$$\begin{aligned} \underset{B,P}{\text{minimize}} \quad & \sum_{i \in \mathcal{I}_{tr}} [-b^T p_i + 2(x^*(c_i)\alpha_i^T) \bullet B - z^*(c_i)] + \rho |B| \\ \text{subject to} \quad & A^T p_i = 2B\alpha_i - c_i && \forall i \\ & p_i \ge 0 && \forall i, \end{aligned} \tag{10}$$

where P represents the set of vectors p_i, containing the dual variables associated with the constraints of optimization problem (2) for every training sample i. In the objective, $\bullet$ refers to the trace inner product. More details of this reformulation can be found in Appendix A. Thus, the bi-level ERM (5) is reformulated as a linear optimization program that can be solved with off-the-shelve solvers. However, this approach is only applicable for linear forecasters which restricts the predictive power of the forecaster. Secondly, there is a significant risk of overfitting as the optimization yields a single minimum objective value over the training set.

III. Interior point-based training method

We improve the SPO+ framework by addressing its limitations. Section III-A introduces Sp-R-IP, an iterative IP-based training method to reduce overfitting. Section III-B extends the SPO+ reformulation framework to non-linear NN forecasters.

In NN training, the two main methods to avoid overfitting on the train data are (i) regularization and (ii) early stopping. When deploying a regularization term in the ERM problem, high values in the parameters of the forecaster f are penalized, which may lead to models which generalize better. Early stopping is a technique that monitors the forecaster's performance on validation data during each iteration of the gradient descent update. Training stops when the validation performance no longer improves, preventing overfitting.

Building on early stopping in gradient descent, we propose a tracking procedure of the validation performance for the SPO+ reformulation. This requires an iterative approach for solving the ERM problem (10). Unlike the simplex method, which explores only extreme points of the feasible region, IP methods explore interior solutions, potentially yielding better-generalizing forecasters. Moreover, IP methods also accommodate non-linear ERM problems, allowing the extension to neural network forecasters, see Section III-B. Our approach evaluates the validation performance across intermediate solutions, enabling to select the best forecaster. This SPO+ Reformulation with Interior Point (Sp-R-IP) method is the core of our contribution.

A. Sp-R-IP

A generic non-linear optimization problem, like an ERM problem, can be written as:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & g(x) \\ \text{subject to} \quad & c(x) = 0, \\ & x \geq 0. \end{aligned} \tag{11}$$

IP solution procedures involve the use of a barrier method. Practically, the inequality constraint is replaced with a log-barrier term in the objective function:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & g(x) - \mu ln(x) \\ \text{subject to} \quad & c(x) = 0, \end{aligned} \tag{12}$$

where μ is a positive number. The barrier term resulting from this non-zero μ penalizes solutions close to the boundary of the feasible region. As $\mu \to 0$, the barrier problem approaches the original formulation. The essence of IP solvers is to iteratively reduce μ, generating a sequence of solutions along the central path, $CP = \{x^*(\mu) | \mu \in \mathbb{R}_0^+\}$, that progressively approximate the true optimal solution. Available IP solvers generally prioritize speed of obtaining the final (locally) optimal solution. To that end, they design specialized algorithms for obtaining steps in the variable space, and obtaining updated values of μ.

However, when the optimization program is an ERM problem used to train a forecaster, the points along the central path should be considered as meaningful intermediate solutions that can be evaluated on the validation set. These points represent different parameter realizations of the forecaster, progressively reducing regret on the training set as μ decreases. Therefore, we propose to prioritize obtaining optimal solutions of Problem (12) for relevant values of μ rather than the speed of getting to the optimal solution on the training set. To that end, we propose a dynamic update strategy for decreasing μ, adapting the rate of decrease based on the validation performance:

$$\mu_{n+1} = \frac{\mu_n}{d \cdot Z_{1,n} \cdot Z_{2,n}}, \tag{13}$$

with

$$Z_{1,n} = \begin{cases} 1 - \epsilon_1, & \text{if } v_n < v_{n-1} \\ 1 & \text{otherwise} \end{cases} \tag{14}$$

$$Z_{2,n} = \begin{cases} 1 - \epsilon_2, & \text{if } v_n < v_i, \forall i < n \\ 1 & \text{otherwise.} \end{cases} \tag{15}$$

In Eq. (13), d represents a constant rate at which μ is decreased. In Eqs. (14)-(15), v_n denotes the validation regret to be minimized, while ϵ_1 and ϵ_2 are small pre-determined constants that modulate the rate at which μ decreases. Specifically, μ decreases more slowly when the latest validation performance improves compared to the previous iteration (Eq. (14)) or sets a new best score (Eq. (15)). Adopting this dynamic update strategy ensures a more granular search in areas of high validation performance. Algorithm 1 depicts a high-level overview of the training procedure.

Algorithm 1 Sp-R-IP algorithm

```
Input: D_tr = {α_i, c_i | i ∈ I_tr}, D_val = {α_j, c_j | j ∈ I_val}
Initialize μ                                   // barrier weight
Initialize π_0, δ_0                    // primal and dual variables
for n = 1, ..., epochs do
    π_n, δ_n ← Barrier(π_{n-1}, δ_{n-1}, μ, D_tr)          // (12)
    Retrieve θ_n ⊂ π_n
    v_n ← regret(f(·; θ_n), D_val)               // Using Eq. (4)
    if v_n < min({v_i | i = 1, ..., n-1}) then
        bestNet ← f(·; θ_n)
    end if
    μ ← updateMu(μ, {v_i | i = 1, ...n})          // Via Eq. (13)
end for
Output: bestNet
```

B. Neural Network in SPO+ Reformulation

We extend the SPO+ framework to accommodate feed-forward NN forecasters by using similar arguments based on duality theory as in [7]. Details can be found in Appendix A. The resulting ERM problem reads:

$$\begin{aligned}
\underset{W^{(l)},b^{(l)},P}{\text{minimize}} \quad & \sum_{i\in\mathcal{I}_{tr}} -b^T p_i + 2\,\mathrm{Tr}(x^*(c_i)\hat{c}_i^T) + \rho\sum_{(l)}|W^{(l)}| \\
\text{s.t.} \quad & \forall i: A^T p_i = 2\hat{c}_i - c_i \\
& \forall i: p_i \geq 0 \\
& \forall i,(l)<L: \alpha_i^{(l)} = a^{(l)}\left(W^{(l)}\alpha_i^{(l-1)} + b^{(l)}\right) \\
& \forall i: \hat{c}_i = W^{(L)}\alpha_i^{(L-1)} + b^{(L)} \\
& \forall i: \tilde{c}_i - \xi|\tilde{c}_i| \leq \hat{c}_i \leq \tilde{c}_i + \xi|\tilde{c}_i|,
\end{aligned} \tag{16}$$

where $\mathrm{Tr}()$ is the trace operator, L the total amount of NN layers, $W^{(l)}$, $b^{(l)}$ and $a^{(l)}$ the weights, biases and (non-linear) activation function of layer (l), and $\alpha^{(0)}$ the input features. This extension of the SPO+ ERM problem enhances predictive power but renders the optimization problem non-linear, eliminating the guarantee of a global optimum. However, as discussed earlier, a globally optimal solution is likely overfitted, and IP methods are preferable for finding well-generalizing intermediate solutions.

The final set of inequalities in (16) forces the NN outputs to remain close to the initial forecast $\tilde{c}$, which is produced by a pre-trained forecaster using a traditional accuracy-oriented loss function. This initial forecast $\tilde{c}$ is also used as part of the input features $\alpha^{(0)}$ for our proposed forecaster DFL-based training technique. Overall, the forecasting model is thus a two-stage process, with an accuracy-oriented forecaster (yielding $\tilde{c}$) that feeds the proposed task-aware re-forecaster (yielding $\hat{c}$). This solution enables to reduce the computational complexity of the training problem.

IV. Case Study

We apply our DFL training approach to the scheduling problem of an ESS in the Belgian DA market, aiming to maximize profits based on price forecasts. Section IV-A provides an overview of the main design choices of the case study. In Section IV-B, we give an in-depth view of the performance of the proposed interior point training procedure, showcasing its improvements compared to the existing SPO+ reformulation method. Section IV-C discusses the overall profit results, comparing our proposed method to relevant benchmarks from the literature.

A. Case study design

We assume the ESS makes decisions according to optimization problem (1). The lookahead time horizon T is 24 hours. The discharge and charge efficiency are set to $\eta^+ = \eta^- = 0.95$. The maximum (dis)charge power is set to $\overline{P} = 1$ MW, and the maximum state of charge $\overline{SoC} = 4$ MWh.

The feature set $\alpha^{(0)}$ for this model comprises electricity generation forecasts both for Belgium and its neighbouring countries, load forecasts, and calendar information. This data is publicly available at [15]. For the initial (first-stage) accuracy-oriented forecaster, which yields $\tilde{c}$, we adopt the *NBEATSx* NN architecture developed in [16], which has demonstrated strong performance in electricity price forecasting. The model is trained using data of 2022.

TABLE I: Summary of DFL-based training methods.

Method	Description
ID-Q	Implicit differentiation of KKT with Quadratic smoothing [12].
ID-LB	Implicit differentiation of KKT with Log-Barrier smoothing [8].
Sp-SG	SPO+ minimization via SubGradient method [7].
Sp-R	SPO+ minimization via Reformulation method [7].
Sp-R-IP	Interior Point-based method with IPOPT's default barrier update [17].
Sp-R-IPs	Interior Point with static barrier update with $\epsilon_1 = \epsilon_2 = 0$ and $d = 1.5$ in Eq. (13).
Sp-R-IPd	Interior Point with dynamic barrier update with $\epsilon_1 = \epsilon_2 = 0.1$ and $d = 1.5$ in Eq (13).

The (second-stage) DFL re-forecaster is trained with train data from January and February 2023. For the months of March and April 2023, the days are assigned to the validation and test set in alternating order. We use the regret loss (4), i.e., the difference in profit obtained by the ESS under perfect price foresight and that obtained by optimizing the schedule based on the price forecast, for training and evaluating the performance.

We evaluate the performance of the proposed methodology through a comparative analysis involving eight distinct training procedures. The baseline model relies solely on the initial forecaster's projections $\tilde{c}$. The other methods deploy different DFL-oriented re-forecaster implementations on top of the initial forecast, as laid out in Table I. The ID-based methods are based on the smoothing procedures put forth in [8], [12]. For the Sp-SG and Sp-R method, we refer the reader to [7]. The Sp-R-IP methods, comprising the methodological contribution of this paper, are henceforth collectively referred to as Sp-R-IPx. For all these DFL re-forecasting models, both a linear regression (-l) and a single hidden layer feed-forward NN variant with softplus activation function (-s) are implemented. These models feature different choices of hyperparameters, which are detailed in Appendix B.

B. Evaluation of interior point method

Figure 1 displays out-of-sample validation regret of solutions to ERM Problem (16) over the training iterations, and hence for decreasing values of the barrier weight μ in (12), comparing different Sp-R-IPx models.

We remind the reader that the original reformulation approach, Sp-R, only considers the final optimal solution on the train set, represented by the (rightmost) point with the smallest value of μ in Figure 1. Although this point corresponds to the optimal solution on the training set, Figure 1 clearly shows that

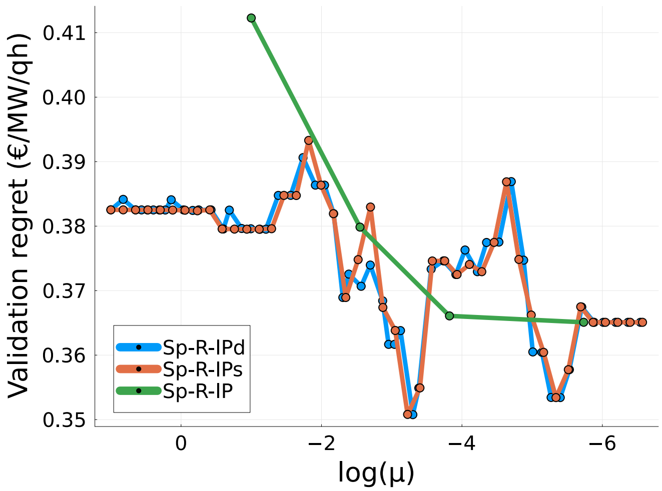

Fig. 1: Evolution of regret loss on the validation data set throughout the training procedure for different IP-based training methods using a NN forecaster with a single hidden layer and softplus activation function.

this is not necessarily the case for the validation set, suggesting that the original Sp-R method is prone to overfitting. In contrast, our proposed Sp-R-IPs/d models capture forecasters with improved validation performance for intermediate values of μ compared to the last point (most optimized on train data). This underscores the importance of our proposed validation tracking procedure. Figure 1 also highlights the importance of controlling the μ update as proposed in Eqs. (13)-(15). The Sp-R-IP model, using IPOPT's default procedure, misses intermediate solutions in two key regions (around $\log(\mu) \approx -3$ and $\log(\mu) \approx -5$), where lower validation regrets are located. While dynamic μ updates (Sp-R-IPd) yield slightly better validation regret than static updates (Sp-R-IPs), the main improvement comes from controlling μ rather than relying on IPOPT's default updates.

C. Overall profit performance

Table II presents a comprehensive evaluation of the test performance. Absolute regret (r_a) is calculated as the sum of the obtained regret over the days in test set, expressed in €/qh. Relative regret (r_r) measures this against the baseline of an ESS making decisions based on the output of the initial forecaster. As such, the relative regret is negative when the forecaster trained using DFL methods improves upon the performance of the initial forecast. Table II also reports the absolute profits (p_a) obtained over the test set.

The first key observation from Table II is that the majority of DFL-based re-forecasting methods outperform the initial forecast, which was trained using a state-of-the-art accuracy-focused technique. This underscores the effectiveness of DFL and the importance of incorporating the downstream problem into the forecasting model's training process.

Secondly, Table II shows that the models from the proposed methodology (Sp-R-IPx) systematically outperform the models using a (sub)gradient descent method, and the original SPO+ reformulation approach, thus affirming the effectiveness of the proposed methodology. The (sub)gradient-based methods likely underperform because they tend to get stuck in local optima, as visualized in Figure 2 in Appendix C.

Finally, the Sp-R-IPs/d methods have the drawback of longer train times compared to the (sub)gradient-based methods. Most notably, whereas the (sub)gradient-based methods show similar train times for the linear regression and NN re-forecaster architectures, the train time significantly increases for the IP-based methods. This increase is the result of the transition from the linear ERM formulation (10) to its non-linear counterpart (16).

TABLE II: Out-of-sample results on test set for absolute regret (r_a), absolute profit (p_a) and relative regret improvement compared to that obtained with the initial forecaster (r_r). T represents the total time in seconds for the training procedure.

Model	r_a **(€/qh)**	r_r **(%)**	p_a **(€/qh)**	T **(s)**
Initial FC	0.46	-	2.03	-
ID-Q-l	0.45	-1.3	2.04	87
ID-Q-s	0.43	-6.0	2.06	157
ID-LB-l	0.45	-1.3	2.04	1,960
ID-LB-s	0.41	-11.6	2.08	1,970
Sp-SG-l	0.49	+6.4	2.00	2,320
Sp-SG-s	0.46	+0.9	2.03	2,309
Sp-R-l	0.48	+5.1	2.01	738
Sp-R-s	0.44	-5.4	2.06	1,314
Sp-R-IP-l	0.40	-13.6	2.09	738
Sp-R-IP-s	0.45	-0.8	2.04	1,076
Sp-R-IPs-l	0.39	-15.7	2.10	1,295
Sp-R-IPs-s	**0.38**	**-17.2**	**2.11**	6,075
Sp-R-IPd-l	0.40	-13.9	2.09	2,649
Sp-R-IPd-s	0.40	-13.0	2.09	13,771
PF	0	-100	2.49	-

V. Conclusion and Future Work

In this paper, we have shown the practical value of Decision-Focused Learning (DFL) for price forecasting applied to storage optimization in electricity markets. Within the realm of DFL, the SPO+ reformulation, which is well-suited when predictions are embedded in a linear decision problem, was shown to be prone to overfitting. To address this problem, we introduced an interior point-based training method that tracks validation performance along the training iterations. Moreover, this paper extended the SPO+ method to accommodate non-linear forecasters such as those based on neural networks. The case study of an ESS participating in the day-ahead market shows that this approach outperforms available decision-focused benchmarks. However, this comes at an increased computational cost. Future research could investigate how this can be implemented more efficiently, for example by exploring mini-batch approaches to reduce the size of the reformulation problem.

References

[1] J. Bottieau, L. Hubert, Z. De Greve, F. Vallee, and J.-F. Toubeau, "Very-short-term probabilistic forecasting for a risk-aware participation in the single price imbalance settlement," eng, *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 1218–1230, 2020.

[2] L. Sang, Y. Xu, H. Long, Q. Hu, and H. Sun, "Electricity price prediction for energy storage system arbitrage: A decision-focused approach," *IEEE Transactions on Smart Grid*, vol. 13, no. 4, pp. 2822–2832, 2022.

[3] E. Wessel, R. Smets, and E. Delarue, "Risk-aware participation in day-ahead and real-time balancing markets for energy storage systems," *Electric Power Systems Research*, vol. 235, p. 110 741, 2024.

[4] J.-F. Toubeau, J. Bottieau, Z. De Grève, F. Vallée, and K. Bruninx, "Data-driven scheduling of energy storage in day-ahead energy and reserve markets with probabilistic guarantees on real-time delivery," *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 2815–2828, 2020.

[5] J. Lago, F. De Ridder, and B. De Schutter, "Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms," eng, *Applied energy*, vol. 221, pp. 386–405, 2018.

[6] A. Jedrzejewski, J. Lago, G. Marcjasz, and R. Weron, "Electricity price forecasting: The dawn of machine learning," *IEEE Power and Energy Magazine*, vol. 20, no. 3, pp. 24–31, 2022.

[7] A. N. Elmachtoub and P. Grigas, "Smart "predict, then optimize"," eng, *Management science*, vol. 68, no. 1, pp. 9–26, 2022, ISSN: 0025-1909.

[8] J. Mandi and T. Guns, "Interior point solving for lp-based prediction+optimisation," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 7272–7282.

[9] B. Amos and J. Z. Kolter, "OptNet: Differentiable optimization as a layer in neural networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70, Jun. 2017, pp. 136–145.

[10] P. Donti, B. Amos, and J. Z. Kolter, "Task-based end-to-end model learning in stochastic optimization," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017.

[11] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, "Differentiable convex optimization layers," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[12] B. Wilder, "Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 2019.

[13] A. Ferber, B. Wilder, B. Dilkina, and M. Tambe, "Mipaal: Mixed integer program as a layer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 1504–1511.

[14] W. Bian and X. Xue, "Subgradient-based neural networks for non-smooth nonconvex optimization problems," *IEEE Transactions on Neural Networks*, vol. 20, no. 6, pp. 1024–1038, 2009.

[15] ENTSO-E, *Transparency platform*, https://transparency.entsoe.eu/, Accessed: May 30 2024, 2022.

[16] K. G. Olivares, C. Challu, G. Marcjasz, R. Weron, and A. Dubrawski, "Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with nbeatsx," *International Journal of Forecasting*, vol. 39, no. 2, pp. 884–900, 2023.

[17] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.

Appendix

A. SPO+ Optimization Framework

Here, we elaborate on the optimization programs of the SPO+ reformulation framework. For completeness, we give the derivation of the original SPO+ reformulation as proposed by Elmachtoub and Grigas [7]. There, the authors start from a general ERM problem given by:

$$\min_{\theta \in \Theta} \sum_{i \in \mathcal{I}_{tr}} \mathcal{L}(c_i, f(\alpha_i; \theta)) + \rho \Omega(f), \tag{17}$$

where a loss function $\mathcal{L}$ is minimized over a the training set $\mathcal{I}_{tr}$ by tuning the parameters θ of the forecaster f, potentially considering a regularizer Ω. In the paper, a convex surrogate of the regret loss function (in the context of a linear downstream optimization program) is proposed, coined the SPO+ loss: $l^{SPO+}(c, \hat{c}) = \max_{x \in S}\{c^T x - 2\hat{c}^T x\} + 2\hat{c}^T x^*(c) - z^*(c)$. When considering a linear forecaster, i.e. $\hat{c} = B\alpha$, we can re-write (17) as:

$$\begin{aligned} \underset{B}{\text{minimize}} \quad & \sum_{i \in \mathcal{I}_{tr}} [o_i + 2(x^*(c_i)\alpha_i^T) \bullet B - z^*(c_i)] + \rho\Omega(B) \\ \text{subject to} \quad & o_i = \max_x \{(c_i - 2B\alpha_i)^T x \text{ s.t. } Ax \geq b\}. \ \forall i \end{aligned} \tag{18}$$

This nested optimization program requires evaluating the objective value of the inner optimization, which is a linear program. For linear programs in the form that appears in the lower level, strong duality dictates:

$$\begin{aligned} \max_x \quad & c^T x & = \quad \min_p \quad & -b^T p \\ \text{s.t.} \quad & Ax \geq b & \text{s.t.} \quad & -A^T p = c \\ & & & p \geq 0, \end{aligned} \tag{19}$$

where p is the vector of dual variables associated with the constraints of the primal problem. This is used to re-write (18) to:

$$\begin{aligned} \min_B. \quad & \sum_{i \in \mathcal{I}_{tr}} [o_i + 2(x^*(c_i)\alpha_i^T) \bullet B - z^*(c_i)] + \rho\Omega(B) \\ \text{s. t.} \quad & o_i = \min_p \{-b^T p \text{ s.t. } -A^T p = c_i - 2B\alpha_i, p \geq 0\}, \ \forall i \end{aligned} \tag{20}$$

which is equivalent to:

$$\begin{aligned} \min_{B,P}. \quad & \sum_{i \in \mathcal{I}_{tr}} [-b^T p_i + 2(x^*(c_i)\alpha_i^T) \bullet B - z^*(c_i)] + \rho\Omega(B) \\ \text{s. t.} \quad & A^T p_i = 2B\alpha_i - c_i \qquad \forall i \\ & p_i \geq 0, \qquad \forall i \end{aligned} \tag{21}$$

which is the same as (10) when using $L1$ regularization.

Here, we show that we can apply the reasoning more broadly. When assuming no specific form of the forecaster, we can generalize ERM problem (18) to:

$$\begin{aligned} \underset{f \in \mathcal{H}}{\text{minimize}} \quad & \sum_{i \in \mathcal{I}_{tr}} [o_i + 2\hat{c}_i^T x^*(c_i)] + \rho\Omega(f) \\ \text{subject to} \quad & o_i = \max_x \{(c_i - 2\hat{c}_i)x \text{ s.t. } Ax \leq b\} \qquad \forall i \\ & \hat{c}_i = f(\alpha_i), \qquad \forall i \end{aligned} \tag{22}$$

where f can be any (non-convex) function from hypothesis class $\mathcal{H}$. Note that we have omitted $z^*(c_i)$ from the objective as this constant term does not affect the outcome of the optimization. The cost forecast $\hat{c}$ is a parameter in the inner optimization program. As such, the inner problem is still a linear optimization program to which we can apply strong duality. Hence, the ERM becomes:

TABLE III: List of hyperparameters per model and explored values.

HP $\rightarrow$	ρ	ξ	ζ	lr	bs	γ
ID-Q	$[0, 0.01]$	-	-	$[5 \cdot 10^{-6}, 5 \cdot 10^{-5}]$	$[8, 64]$	$[0.1, 0.3, 1, 3, 10]$
ID-LB	$[0, 0.01]$	-	-	$[5 \cdot 10^{-6}, 5 \cdot 10^{-5}]$	$[8, 64]$	$[0.001, 0.003, 0.01, 0.03, 0.1]$
Sp-SG	$[0, 0.0001, 0.01, 1]$	-	-	$[0.001, 0.01, 0.1, 1]$	$[16, 64]$	-
Sp-R **Sp-R-IPx**	$[0, 0.001, 0.1, 10]$	$[0.02, 0.1, 0.5, \infty]$	$[0, 1]$	-	$[16, 64]$	-

$$\begin{aligned}
\underset{f \in \mathcal{H}}{\text{minimize}} \quad & \sum_{i \in \mathcal{I}_{tr}} [o_i + 2\hat{c}_i^T x^*(c_i)] + \rho\Omega(f) \\
\text{subject to} \quad & o_i = \min_p \{-b^T p \text{ s.t. } -A^T p = c_i - 2\hat{c}_i, p \geq 0\} \; \forall i \\
& \hat{c}_i = f(\alpha_i), \qquad \forall i
\end{aligned} \tag{23}$$

Again, we can reformulate this to an equivalent single-level optimization problem:

$$\begin{aligned}
\underset{f \in \mathcal{H}, P}{\text{minimize}} \quad & \sum_{i \in \mathcal{I}_{tr}} -b^T p_i + 2\,\mathrm{Tr}(x^*(c_i)\hat{c}_i^T) + \rho\Omega(f) \\
\text{subject to} \quad & A^T p_i = 2\hat{c}_i - c_i \qquad \forall i \\
& p_i \geq 0 \qquad \forall i \\
& \hat{c}_i = f(\alpha_i). \qquad \forall i
\end{aligned} \tag{24}$$

By replacing f with constraints that correspond to the forward pass of a NN, we end up with ERM Problem (16).

B. Hyperparameters

Table III provides an overview of the hyperparameter space explored during model training, detailing the range of values assessed through grid search for hyperparameter tuning. For the ID models, the hyperparameter γ refers to the weight of the smoothing term in the modified objective function. The hyperparameter ζ refers to an additional constraint in (10) that sets to 0 all NN weights that capture time dependencies. That is, input features belonging to timestep i cannot have an influence on $\hat{c}_j$ for $i \neq j$. Surprisingly, the results show that models with this restricted forecaster tend to outperform their non-restricted counterpart. The hyperparameters lr and bs refer to the learning rate and batch size.

C. Results of gradient-based methods

A widely-used strategy to smoothen downstream optimization problems exhibiting output decisions that are non-differentiable or exhibit zero gradients w.r.t. the input forecast is adding a term to the objective function of the original optimization problem. In the ID-Q method, we used the quadratic smoothing from [12]. The modified optimization problem becomes:

$$\begin{aligned}
\underset{x}{\text{minimize}} \quad & \hat{c}^T x + \gamma x^T x \\
\text{subject to} \quad & Ax \geq b,
\end{aligned} \tag{25}$$

with γ the smoothing strength. In the ID-LB method, the problem was smoothed with a log-barrier term, as proposed in [8]. Both models were implemented using the *Cvxpylayers* Python library [11].

Figure 2 shows the regret performance on the train set of an ESS when deploying the ID-Q forecaster in the ESS optimization program for different values of the weight γ of the smoothing term throughout the iterations of the training procedure. An SP-R-IPd benchmark is provided. We see that for small values of γ, the NN is unable to improve the in-sample train profit. This indicates that for those values, the original optimization problem is insufficiently smoothed to obtain useful gradients for training. Secondly, none of the ID forecasters can match the in-sample performance of that obtained by the SP-R-IPd model. This can be attributed to two possible causes: (i) the gradients are always imperfectly calculated, as a balance must be struck between a smooth mapping from price forecast to optimal decision leading to well-defined gradients (large γ), and a correspondence between the original and modified problem (small γ); (ii) a gradient-descent algorithm is more prone to getting stuck in a locally optimal solutions.

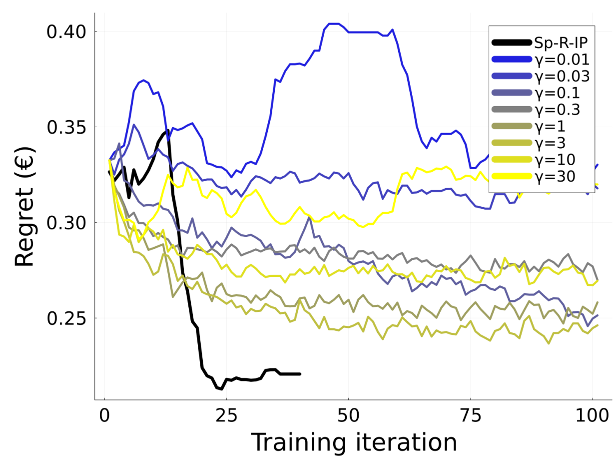

Fig. 2: Evolution of train regret throughout the training procedure of ID-Q models with softplus activation function for different values of the smoothing parameter γ in (25).