



Delft University of Technology

Performance and interaction assessment of neural network architectures and bivariate smart predict-then-optimize

Wen, Junhan; Abeel, Thomas; de Weerd, Mathijs

DOI

[10.1007/s10994-025-06910-1](https://doi.org/10.1007/s10994-025-06910-1)

Publication date

2025

Document Version

Final published version

Published in

Machine Learning

Citation (APA)

Wen, J., Abeel, T., & de Weerd, M. (2025). Performance and interaction assessment of neural network architectures and bivariate smart predict-then-optimize. *Machine Learning*, 114(11), Article 252. <https://doi.org/10.1007/s10994-025-06910-1>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.



Performance and interaction assessment of neural network architectures and bivariate smart predict-then-optimize

Junhan Wen¹ · Thomas Abeel¹ · Mathijs de Weerd¹

Received: 16 June 2025 / Revised: 16 June 2025 / Accepted: 26 September 2025
© The Author(s) 2025

Abstract

Smart “predict, then optimize” (SPO) (Elmachtoub in *Manag Sci* 68(1): 9–26, 2022) is an end-to-end learning strategy for models that predict parameters in optimization problems. Unlike minimizing mean squared error (MSE) which cares about prediction accuracies, SPO aims to ensure that predictions lead to the best possible decisions. The associated loss function, termed *SPO loss*, measures the decision’s *regret* from optimal outcomes with parameter realizations. Existing literature has demonstrated the viability of SPO, however, these studies often focus on classical optimization problems and employ a limited set of models for benchmarking. In this study, we tackled a decision-making task inspired by real-world challenges across a wide range of neural network models. Unlike classical problems, our task requires a unique approach: collaboratively training two models to predict different variables. On top of that, one of the decision variables also affects the feasibility of the decisions, further increasing the complexity. While our implementation validates the benefits of SPO, we were surprised to find that models trained exclusively on SPO loss do not consistently attain the minimum regret. Our further investigation into hyperparameters illustrates that the well-tuned models learned very similar patterns from the feature set, irrespective of whether MSE or SPO loss was used. In other words, the change from MSE to SPO loss in training primarily affected the layer biases. Therefore, to improve the learning efficacy with SPO loss, we propose prioritizing learning feature patterns as the fundamental step. Possible strategies include using specialized neural network layers to capture deeper patterns more effectively or simply warming up by training with MSE. Specifically, a warming-up process is particularly advantageous for model(s) where the outputs are closely tied to constraints, as their prediction accuracy significantly impacts the decision feasibility. The insights are investigated empirically through two real-world trading scenarios. By leveraging datasets with diverse properties, we demonstrate the novelty and generalizability of our investigation.

Keywords Decision-focused learning · Machine learning and optimization · Deep neural networks · Constraint optimization · Real-world applications

Editor: Xiaolin Huang.

Extended author information available on the last page of the article

1 Introduction

When machine learning models are employed to assist with downstream tasks, the impact of prediction errors can be asymmetric. In cases where the downstream task involves solving an optimization problem, this scenario is commonly referred to as a predict-then-optimize problem (Elmachtoub and Grigas 2022). To enhance overall performance, it is essential to shift the focus of the learning process from prediction-oriented methods to decision-centric approaches. This paradigm shift is known as decision-focused learning (DFL) (Mandi et al. 2023).

Regret is the core metric for evaluating decision-making performance. It describes the disparity between the optimal outcome and the outcome from the actual decision made with decision parameter estimations. To optimize the DFL process, (Elmachtoub and Grigas, 2022) proposed training models directly on regret rather than on accuracy-based losses like mean squared error (MSE). They named this method *Smart “Predict, then Optimize”* (SPO). As such, the regret is also referred to as the *SPO loss*.

The SPO method has been widely used to tackle classical optimization problems (Donti et al. 2017; Wilder et al. 2019) and in a few abstractions of real-world cases (Wang et al. 2020; Verma et al. 2022). However, current research leaves two significant gaps when it comes to addressing a broader range of real-world challenges. First, most studies oversimplify decision-making by assuming only one type of unknown parameter, which can therefore be estimated with a single model or optimizer. Second, existing research in DFL is often investigated with using only one machine learning model – typically a small neural network (NN) with fixed hyperparameters, or even simple linear models (Kotary et al. 2021; Yan et al. 2020; Tian et al. 2023; Mandi et al. 2022). This facilitates the explainability of the models within the DFL task; nevertheless, with the vast variety and more complex architectures of neural network models available, it is still an open question about how different architectures could enhance DFL performance.

To advance DFL research in realistic scenarios, we evaluated the performance of SPO in training diverse neural networks for a type of financial optimization task. We focused on an integrated system for both futures and real-time market trading, where products can be sold immediately or traded as futures, with transactions agreed upon in advance. Many real-world trading activities follow a pattern similar to this system, such as day-ahead bidding and last-minute trading in the renewable energy market (Sen et al. 2006; Kirschen and Strbac 2018), the procurement of fresh agricultural products (Schieffer and Vassalos 2015; Dileep et al. 2002; Hueth et al. 1999), and in financial markets where stocks, options, and futures are traded concurrently (Brealey 2000; Clark et al. 2016; Brooks and Garrett 2002). In these scenarios where future production cannot be perfectly controlled, a pre-set contract may result in waste from overproduction or penalties for under-delivery. Further, improper market price predictions may reduce potential profits. As a consequence, the challenge requires proper forecasts of both the *price* and the *yield*, which also acts as a constraint to the decision-making.

In short, our study investigates how different architectures and training strategies affect the models’ abilities to predict and promote the downstream optimization task. Our study adopts and tailors the SPO method to train multiple prediction models that estimate decision variables (some of which act as soft constraints) respectively and collaboratively, aiming

at minimizing the overall *regret* incurred by the resulting contracting decisions. The main contributions of this work consequently are:

- We introduced a novel and realistic optimization task for product trading, of which the decision is incentivized by prices and the feasibility is bounded by yield.
- When tackling this task, we evaluated the impact of training with *smart predict-then-optimize* (SPO), and illustrated that exclusive use of SPO loss did not improve regret consistently.
- With two real-world scenarios, we analyzed how neural networks' learning capacity fits with SPO, discovering that simpler models show more significant discrepancies of training and testing on regret, while too complex models tend to get stuck in local optima.
- The results validate that layer weights of models trained with MSE and SPO loss are similar, suggesting that training with MSE as a warm-up can enhance both effectiveness and efficiency, especially when output impacts the feasible region.

2 Related work

2.1 Decision-focused learning

Apart from prediction-focused learning (PFL) that maximizes the accuracy of predictions, decision-focused learning (DFL) optimizes the downstream impact of decisions based on those predictions (Mandi et al. 2023). The *smart predict-then-optimize* (SPO), one of the latest DFL frameworks, introduced the decision regret as the SPO loss to train DFL models. In a linear problem that optimizes w to minimize $c^T w$, the SPO loss is written as (Elmachtoub and Grigas 2022):

$$L_{\text{SPO}}(c, \hat{c}) := \max_w (c^T w) - c^T \arg\max_w (\hat{c}^T w) \quad (1)$$

, where c represents the decision parameter that describes a linear optimization problem in a known feasible region and w is the decision variable optimized according to c (or the predicted value $\hat{c}$). Training models with SPO loss have proven to be highly effective across a wide range of optimization problems (Wang et al. 2020; Mandi et al. 2020; van Staden et al. 2022; Chang et al. 2023). However, researchers have identified a trade-off between prediction accuracy and SPO loss during training. Thus, some suggest that accuracy may serve as a regularizer in DFL (Mandi et al. 2023; Tang and Khalil 2022).

Among DFL-related studies, most work has focused on a single model or a lone optimizer to demonstrate the SPO method. However, this approach falls short of realistic scenario modeling, where multiple decision variables often need to be predicted – each potentially having varied impacts on the optimal solution. For instance, the standard setting of SPO uses a known feasible region (Elmachtoub and Grigas 2022; H. Liu and Grigas 2022), but the problem when the predictions may lead to decisions that violate constraints has not been well-defined yet (Mandi et al. 2023). While methods like relaxing constraints or imposing penalties improved the feasibility of solutions, they did not consistently ensure constraint satisfaction (Sang et al. 2022; Chang et al. 2023; Hu et al. 2023; Sang et al. 2022; Kotary

et al. 2021). In our study, we apply SPO to a bivariate optimization problem to address the increased variable complexities that exceed those of existing setups. By training models individually, we demonstrate the effectiveness of DFL and PFL in fine-tuning models to meet specific targets.

Meanwhile, most existing works fit in learning programming formulations and simple prediction models (Verma et al. 2022). Some applied small (two to four layers) neural networks and compared the benchmarks with classical regressors (HU et al. 2023; Liu and Grigas 2021). Nevertheless, the choice of model architecture was always predetermined without explicitly reasoned or finetuning discussion. In fact, the family of deep learning models is much broader, with techniques ranging from traditional grid search to advanced metaheuristics for systematically exploring those hyperparameters (Wu and Tsai 2024). For instance, recurrent and attention modules that are beneficial in-/extrinsic time-series analysis tasks (Connor et al. 1994; Vaswani et al. 2017; Ren et al. 2023) align well with the properties of many feature sets in DFL tasks. However, to the best of our knowledge, these neural network architectures have yet to be explored within the context of DFL.

Furthermore, many training strategies that were originally designed for PFL can be adapted for DFL tasks. For instance, (Mandi, Demirovi Stuckey, and Guns, 2020) “warmed up” the SPO learning with a transferring learning-based idea, aiming at accelerating the training process. This approach was initially applied to demonstrate its potential for speeding up training. However, the setting of a limited number of pre-training epochs restrict the methodology’s generalizability, and other benefits remain unexplored. Despite this, the warm-up process used in PFL can easily be connected to other PFL and DFL solutions, enabling deeper discussions on how these strategies fine-tune models.

2.2 Trading optimizations constrained by contracting decisions

At present, there exist many situations where the same product is traded in multiple markets, with varying time lags between the trading and transaction dates. For example, in the renewable energy sector, prices fluctuate due to shifts in supply and demand (Kirschen and Strbac 2018; Block et al. 2008). To manage this volatility, traders commonly use supply contracts to stabilize and secure transactions and turn to real-time markets for last-minute adjustments (Yang et al. 2023; Rahimiyan and Baringo 2016). A comparable strategy is found in the trade of perishable agricultural products, commonly referred to as *contract farming* in many countries (Schieffer and Vassalos 2015; Dileep et al. 2002; Hueth et al. 1999). Price stability encourages growers to sell most of the harvests through contracts, with surplus typically directed to local markets. However, imbalances in negotiation power—particularly between small suppliers and large buyers such as supermarkets—can lead to inefficiencies, which often manifest as either over-supply or under-supply (Fathallahi et al. 2020; Ncube 2020; Singh 2005).

Since contracts require advance commitments while the real-time market has higher uncertainty but lower capacity, producers face the challenge of determining how much future production to include in their contracts. Existing studies estimate the profitability under such situations by statistical models such as mean-variance and probability density distribution models (Sen et al. 2006; Krishnamurthy et al. 2018; Chen et al. 2021). To optimize contracting decisions, popular methods include stochastic optimization and worst-case analysis models (Rahimiyan and Baringo 2016; Liu et al. 2016; Sen et al. 2006; Safdar-

ian et al. 2015). However, these statistical models generally prioritize accuracy in predicting future production and real-time prices, often overlooking the specific consequences of under- or over-supply, such as contract-breaching fines (Block et al. 2008). Consequently, such predictions should be (also) evaluated based on their ultimate impact, which underscores the importance of the DFL concept.

3 Research questions

Previous works have demonstrated the advancement of using SPO loss in training models for decision-focused learning (DFL) purposes. However, there is no common understanding of what machine learning models we need to use and which model is suited for which problems. To enhance the integration of the SPO method with a broader range of deep learning tools, we explore a variety of neural network architectures with diverse learning capabilities and assess the PFL and DFL performance. In essence, this research is designated to answer the following questions:

- RQ1** How does the learning capacity of models influence PFL and DFL?
- RQ2** How do MSE and SPO loss functions influence the fine-tuning of models?
- RQ3** How to optimize the DFL performance under a given model architecture?
- RQ4** How can training strategies be tailored to align with the roles of the target (decision) parameters being predicted?

4 Methodology

In this study, we comprehensively investigated the influence of model design and training setup in the implementation of the end-to-end training strategy *smart predict-then-optimize* (SPO) (Elmachtoub and Grigas 2022). We addressed our research inquiries by benchmarking against realistic variants of a portfolio decision-making scenario. The objective of the use case is to decide how much future production should be committed to a given trading contract while both the (other) market's price and the future yield are unknown. Under this scenario, the yield directly influences the scale of the decision and the price mainly incentivizes the decision towards one of the trading channels. This section outlines our methodology from four key perspectives: model selections, metric choices, loss function decisions, and how we came up with the improvement strategy.

Models: To explore DFL with a border range of models, we exploited neural networks with different multi-layer perception (MLP) depths, with a recurrent neural network (RNN) layer, with pair-wise self-attention module after the input layer (ATT), and with just one linear layer to represent a linear regressor, etc. RNN and ATT were chosen because the features are extrinsic time-series data. In the use case specifically, two models were trained to predict the market price and production. They were trained collaboratively from scratch with specified optimizers. By employing diverse training settings and strategies, we empirically studied the effectiveness of PFL and DFL on models with different learning capacities.

Metrics: The primary metric for evaluating model performance is the average *regret* on the testing set, which reflects the expected “unearned” monetary units from real-world decisions. To assess real-world impact, we compared the *profit* distributions under variable

predictions with optimal decision outcomes under parameter realizations. We also analyzed the mean squared error (*MSE*) of predictions, as it indicates the generalizability of the model and the training strategy. We used the mean error (*ME*) of predictions to demonstrate the bias of models. Additionally, we discussed the performance related to *timing*. To compare how PFL and DFL fine-tune the models, we measured the *cosine similarity* of corresponding layers in the trained models, which reveals how each method influences the weight and bias of the layers.

Training losses: Our study began with models trained exclusively on MSE or SPO loss. We then refined the experiments by training the models using a weighted average of both MSE and SPO loss, i.e. the loss $L = w_1 \cdot \text{regret} + w_2 \cdot \text{MSE}$, s.t. $w_1 + w_2 = 1$. By varying the ratio $w_1 : w_2$ from 0 to infinite, we evaluated how the models are progressively tailored in terms of PFL and DFL goals. We applied two independent ratios for the price and yield prediction models respectively, so as to explore specific preferences for each model's training and optimize the learning efficacy.

Improvements: By comparing the final weights of models and investigating the metrics during the training process of PFL and DFL, we decided to “warm up” the DFL training from models converged on PFL process. We compared the performance with the other models that were trained by DFL from scratch, and with models trained under diverse numbers of PFL epochs, so as to generalize the methodology. On top of that, we conducted such initialization on only one of the models to further explore whether there is a core model to start with or if the collaboration essentially affects the training. The in-depth exploration suggests the optimal design of learning settings, referring to the outputs' specific functionalities in the downstream optimization task.

5 Problem and task formulation

5.1 Background

When the future production of a product cannot be perfectly controlled, such as with agricultural products or renewable energy, maintaining steady relationships with downstream partners can be challenging. To manage the risks associated with forecast errors, traders often diversify their strategies across markets that differ in how far in advance decisions must be made (Wang et al. 2022). For example, some contracts require early commitments, while real-time markets are typically smaller and more volatile (Kirschen and Strbac 2018; Hueth et al. 1999). Poor forecasts can lead to overproduction and waste, or shortfalls that result in monetary penalties and damage relationships with downstream buyers. In addition, inaccurate predictions about market conditions might also cause producers to miss more profitable sales opportunities.

In related studies, the profitability is estimated by statistical models, e.g. by a mean-variance model or probability density distributions (Sen et al. 2006; Krishnamurthy et al. 2018; Chen et al. 2021). However, these methods can lead to high penalties when the decisions are diverged from the predictions. To avoid such, growers, for example, may opt for conservative yield estimations (Ncube 2020; Singh 2005). Thus, it is more insightful to evaluate predictions based on their impact on decisions rather than on accuracy alone. This approach aligns well with the DFL strategy.

We chose this type of use cases because it represents a common scenario across many financial systems, featuring a straightforward and distinct goal that serves as a basis for experimenting with various models. Even so, the problem can be more complex and pose greater challenges than many existing DFL works, as it involves predicting multiple parameters (prices, production, etc.) and the available feature set may not always have a linear relationship with the target outputs.

5.2 Problem characteristics

In this paper, we investigated a use case where the goal is to maximize the overall profit for each batch of production by determining the best trading strategy within an integrated contract and market system. For the decision, we specify how much of a future production batch will be sold through contractual agreements at a predetermined price, and how much will be reserved for sale in the market, where prices fluctuate in real time. Generally, the closer the time to the transaction, the higher accuracy of prediction can be achieved. Nevertheless, real-time demand and price have significantly higher variations than trading through futures or options (Conejo et al. 2010; Vu et al. 2022; Schieffer and Vassalos 2015) – sometimes there can be barely any demand and thus significantly low prices, but at other times their profitability can surge in response to sudden demand or an unforeseen supply shortage.

In specific, this problem scenario has the following characteristics:

1. We act as a price-taker in all trading activities.
2. At the time of decision-making, the contract prices $\mathbf{P}_c$ ¹ and transaction time t are known, and both the final production volume (shortened as Y for yield) and the (real-time) market price $\mathbf{P}_r$ can be predicted based on current trends.
3. Once production is completed, we first fulfill our contract obligations by delivering the agreed-upon amount, or as much as we can. The remaining product is then sold in the market at the prevailing price.
4. The volume of contract-based trading is constrained by the capacity of the contract and the actual yield of our production. Moreover, contracted buyers generally do not accept purchases beyond the committed amount at decision-making time.

5.3 Objective function and specific SPO framework

The optimization objective is the overall profit of a batch of production, denoted as $\text{Pft}(\mathcal{D}_c, T_c, T_r, Y)$, wherein $\mathcal{D}_c$ is the contracting decision, $\mathbf{T}_c$ means the contract transaction, $\mathbf{T}_r$ means the market transaction, and Y represents the final production.

By definition, the profit is the difference between the total revenue, i.e. $P_c \cdot T_c + P_r \cdot T_r$, and the costs. In line with established practices in the field (Lee et al. 2015; Fathallahi et al. 2020), we considered linear competitive market models to describe market capacities, i.e. the maximum volumes for contract trading is $T_{c,\max} := a_{ct} \cdot P_c + b_{ct}$, and in the real-time market is $T_{r,\max} := a_{rt} \cdot P_r + b_{rt}$. Consequently, to focus solely on the parameters derived from the datasets, the objective function can be formulated as $\text{Pft}(\mathcal{D}_c, P_c, P_r, Y)$.

¹ We use bold fonts when introducing new parameters. After that, we use roman font for decision parameters such as P_c , P_r , Y , and calligraphic font for the decision variable, i.e. the contracting amount $\mathcal{D}_c$ and $\bar{\mathcal{D}}_c$.

According to the trading mechanism, at the transaction date, the actual transaction amount with the contract is the (original) decision bounded by the actual production and the contract capacity: $T_c = \min\{\mathcal{D}_c, Y, T_{c,\max}\}$. The market transaction is bounded by the remaining production and the market capacity: $T_r = \min\{Y - T_c, T_{r,\max}\}$. Given the real-time price and production are unknown at the time of contracting, the decision is actually made based on their predicted values $\hat{P}_r$ and $\hat{Y}$. Hence, the best decision that we can make is $\tilde{\mathcal{D}}_c = \operatorname{argmax}_{\mathcal{D}_c} \text{Pft}(\mathcal{D}_c, P_c, \hat{P}_r, \hat{Y})$. Consequently, later at the transaction execution date, the actual contract transaction is $T_c = \min\{\tilde{\mathcal{D}}_c, Y, T_{c,\max}\}$. In this way, the yield Y serves as a soft constraint of the optimization problem.

The costs consist of (i) the internal production cost that depends on the production: $a_{c1} \cdot Y$ and (ii-iii) the external cost from improper decisions: (ii) if the actual production falls short of the contracted amount, i.e. $\mathcal{D}_c > Y$, we pay a penalty for every under-delivery unit $a_{c2} \cdot \text{pos}[\mathcal{D}_c - Y]$ and we trade to the contract with $T_c = Y$ instead of $T_c = \tilde{\mathcal{D}}_c$ because $T_c = \min\{\tilde{\mathcal{D}}_c, Y, T_{c,\max}\}$; (iii) or if the overall production exceeds the contracted amount and the market's capacity, the waste is accounted for in financial terms: $a_{c3} \cdot \text{pos}[Y - T_c - T_r]$. Note that in this specific context, the penalty for infeasible decisions is smoothly incorporated by the under-delivery costs.

Hence, the objective function, i.e. the profit Pft, is written as:

$$\begin{aligned} & \mathfrak{t}(\mathcal{D}_c, P_c, P_r, Y) \\ &= P_c \cdot T_c + P_r \cdot T_r - a_{c1} \cdot Y - a_{c2} \cdot \text{pos}[\mathcal{D}_c - Y] - a_{c3} \cdot \text{pos}[Y - T_c - T_r] \end{aligned} \quad (2)$$

$$\begin{aligned} &= P_c \cdot \min\{\mathcal{D}_c, Y, (a_{ct} \cdot P_c + b_{ct})\} + P_r \cdot \min\{Y - \min\{\mathcal{D}_c, Y, (a_{ct} \cdot P_c + b_{ct})\}, (a_{rt} \cdot P_r + b_{rt})\} \\ &- a_{c1} \cdot Y \\ &- a_{c2} \cdot \text{pos}[\mathcal{D}_c - Y] \\ &- a_{c3} \cdot \text{pos}[Y - \min\{\mathcal{D}_c, Y, (a_{ct} \cdot P_c + b_{ct})\} - \min\{Y - \min\{\mathcal{D}_c, Y, (a_{ct} \cdot P_c + b_{ct})\}, (a_{rt} \cdot P_r + b_{rt})\}] \end{aligned} \quad (3)$$

By capturing the core concept of the original SPO loss as in Eq. 1 and given that the contract price P_c is known at the moment of decision-making, we derived the specific SPO loss for the market price and yield prediction $\hat{P}_r$ and $\hat{Y}$ as:

$$\begin{aligned} \mathfrak{s}PO(P_r, \hat{P}_r, Y, \hat{Y}) &:= \max_{\mathcal{D}_c} \text{Pft}(\mathcal{D}_c, P_c, P_r, Y) - \text{Pft}(\tilde{\mathcal{D}}_c, P_c, P_r, Y) \\ &= \max_{\mathcal{D}_c} \text{Pft}(\mathcal{D}_c, P_c, P_r, Y) - \text{Pft}(\operatorname{argmax}_{\mathcal{D}_c} \text{Pft}(\mathcal{D}_c, P_c, \hat{P}_r, \hat{Y}), P_c, P_r, Y) \end{aligned} \quad (4)$$

5.4 Data

To illustrate the decision problem at hand and to assess the effectiveness of the proposed methodology in realistic settings, the experiments of the paper are conducted as case studies. The experiment section mainly focuses on the decision-making performance of a wind farm that participates in the day-ahead and real-time market, using the energy price and consumption data sourced from (Jhana, 2019) (refer to as the “wind” case).

The decisions of trading in the day-ahead market, which we identify as the channel for “contracting”, must be finalized before the bid clearing deadline at 12 pm (Yang et al. 2023; Aguiar et al. 2020; Conejo et al. 2010). When there is a shortage of production, we may opt

to purchase additional electricity to fulfill our committed amount; nevertheless, in instances of unplanned overproduction, the cost of finding immediate consumers can be prohibitively high (Conejo et al. 2010). Consequently, the expense incurred from overproduction significantly outweighs that of underproduction, underscoring the importance of employing DFL in predicting the unknowns. The dataset covers 34,613 timestamps, collected hourly from 2015 to 2018. Due to the large computational load of DFL, we used the data from every tenth hour for exploitative studies. The feature set had 143 attributes in total, consisting of local weather records from the past week, trading information from the past four realizations, and the contract price for the current decision round. The price was scaled by 100 from the original monetary unit (euro), and the generation was scaled by 10,000 to facilitate neural network training. Detailed settings, exploration, and accessibility of the data utilized in our study can be referred to the appendix.

In fact, in 90% of the instances in our primary dataset, the real-time market price exceeds the daily market price, which serves as our contract price. To address this bias, we performed an ablation study that ensures an equal likelihood of either price being higher than the other, and we refer to this as the “*wind-50*” case.

In addition, we curated a dataset for another use case involving a similar trading optimization problem, but with different characteristics from the wind trading scenario. In this case, we consider the role of a tomato grower, where the task is to decide how much yield to commit through farming contracts (Dileep et al. 2002; Ncube 2020). The available price data, sourced from (Borsa, 2023), is reported daily, though not consistently for every single day. Given that prices are provided in upper and lower limits but without specifying the trading channel, we considered two scenarios: in the first, the upper limit is seen as the contracting price and the lower limit as the market price (referred to as “*tomato-c*” as the contract price is higher); in the second scenario, we reversed this arrangement, ensuring that the contract is not more favored than the market (referred to as “*tomato-m*”).

We used similar attributes to form the features for the two tomato cases. Nonetheless, since the decision-making time and available data are much sparser, we used the daily temperature record in the past 28 days of the decision date and the realized cases of the past four available records within a month. Together with the current contracting price, the feature set had 41 attributes. In total, we had 369 data points, collected from January 2014 to August 2023. For both cases, we scaled the features by constants to facilitate the training. These two cases, together with the “*wind-50*” case, compared the impact of DFL on yield (Y) and market price (P_r) predictions under more diverse scenarios. The decision performances on these three cases are discussed in Sect. 6.5.

6 Experimental results

In this section, we present experimental results on investigating and refining the decision-focused learning (DFL) process with SPO loss by primarily the *wind* case, so as to gain deeper insights and address the research questions outlined in Sect. 3.

Our exploration begins with the examination of multi-layer perception (MLP) models with varied numbers of layers. This includes a one-layer model without activation, which essentially functions as a linear regressor. Additionally, we analyzed the performance implications of incorporating a fully-connected recurrent (RNN) layer or a pair-wise self-atten-

tion (ATT) layer alongside the MLP layers. All models started with 256 neurons and ended with 512, except for the linear regressor, which had only one layer of 256 neurons. Dropout layers were added after every two MLP layers. For model trained by DFL, the outputs of the neural networks flowed into a CVXPY layer (Diamond and Boyd 2016) to compute and back-propagate the SPO loss. We trained both the price- and yield-prediction models using identical model architectures and input features. Unless specified otherwise, the models were trained together from scratch. Hyperparameter fine-tuning is further discussed in the appendix.

In this section, we present results from models trained by stochastic gradient descent (SGD) optimizers with a $2e-3$ learning rate, a $1e-5$ weight decay regularization, and a maximum of 1,000 epochs with early stopping. The dataset is divided into 7:1:2 for training, validation, and testing. All experiments were run with five fixed random seeds unless otherwise noted. The experiments were carried out on an NVIDIA A40 GPU.

6.1 Model architecture impacts performance and training–testing discrepancy on regret

This subsection addresses the first research question regarding how the learning capabilities of MLP models affect outcomes. Figure 1 depicted the final regrets, resulting from decisions optimized using parameters predicted by models of different depths. The plot on the left compares the total regret across three cases: when neither model, only one, or both models were trained using the SPO loss. The four smaller plots on the right provide a closer look at regret under various combinations of training losses. Regret results from models trained with both MSE and SPO loss are included to link the findings between PFL and DFL approaches.

As illustrated in the first subplot, models trained both by PFL exhibited the poorest decision-making performance, underscoring the necessity of integrating SPO loss into the training process. The subsequent four colored plots further reveal that decision performance generally improves when the price model training includes more SPO loss. However, this trend isn't consistently seen when increasing SPO loss in the yield prediction model's training. In other words, the lowest regret doesn't always occur when all models are trained exclusively with SPO loss. This inconsistency is especially clear in shallower models, while deeper models tend to perform worse overall, likely because they get stuck in local optima.

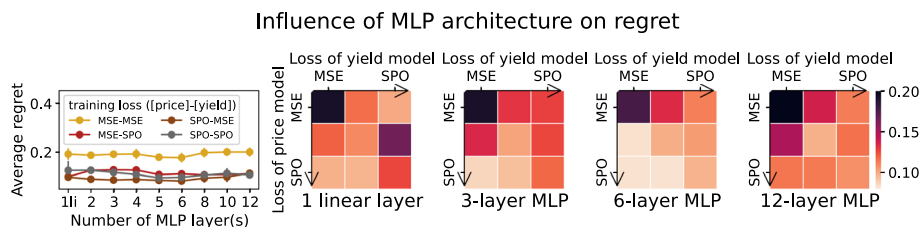


Fig. 1 Regret of models varies with MLP layer numbers and loss function combinations. The left plot compares regret over depths for models trained on different loss functions. The right four plots compare regrets under different loss functions and model architectures. For the right plots, the x-axis label means the loss for yield prediction training and the y-axis label is that for price. The arrows on the x- and y-axis indicate the increasing involvement of SPO in the loss function. All plots share the same color scale, shown on the right. The results in the middle boxes used loss = $0.5 \cdot MSE + 0.5 \cdot SPO$. Results are collected over five seeds

Next, we compare the performance among four types of neural networks: MLP, RNN-MLP, ATT-MLP, ATT-MLP and with MLP residuals. Figure 2 presents the average MSE and regret from the models with varying numbers of MLP layers and different loss functions. The prediction accuracies of the models are close to state-of-the-art –for example, comparable to the method by (Nate Gruver and Wilson, 2023) – and can improve slightly with more complex models. This suggests that all architectures are suitable for the learning tasks in this scenario. However, these gains in accuracy only modestly reduce final regret to a limited extent. In contrast, training models with DFL produces more substantial gains in decision-making performance. As shown in the third plot of Fig. 2, this advantage is consistent across all architectures, highlighting that optimizing directly for regret is more effective than simply increasing model complexity while using the same loss function.

We also analyzed the correlations of Kullback–Leibler (KL) divergence between regret and the MSEs of the price and yield predictions across five test sets. Our results show that regret is moderately positively correlated with errors in yield prediction (0.61), suggesting that yield prediction quality does influence the regret. In contrast, regret has almost no correlation with price prediction errors (-0.01), indicating that perfect accuracy in price prediction is not always necessary for DFL purposes. Interestingly, the PFL performances of the two prediction models show a slight negative correlation (-0.37), which further supports the idea that these models play different roles in the optimization task.

6.2 Models learned alike feature patterns from PFL and DFL

To better understand the mechanisms behind PFL and DFL training, we analyzed the cosine similarity between layer parameters of models trained with different loss functions. Figure 3 shows the results for models with six MLP layers, which was the best-performing architecture among models in Fig. 1 and 2. We observe notable similarities in the weights of MLP layers, while the differences appear mainly in the layer biases.² These results suggest that the divergence between PFL and DFL primarily affects the bias terms, with limited impact on feature representations. This observation motivates our strategy of pre-training on PFL

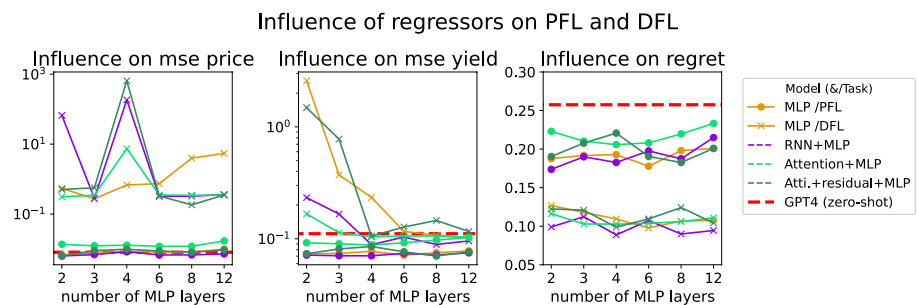


Fig. 2 MSE of the price (left), yield (middle) prediction models, and their resulting regret (right). The x-axis represents the number of MLP layers. Lines are color-coded by neural network architecture, as indicated in the legend. Dot markers show performance for models trained under PFL, while 'x' markers indicate DFL-trained models. Except for the zero-shot performance of GPT-4, results are averaged over five seeds

² Similar trends were found in the self-attention and RNN layers, detailed in the appendix.



Fig. 3 Cosine similarity of layer parameters between models with six MLP layers (“fc”) trained using SPO loss and MSE. The final plot compares a PFL model with a DFL model that was initialized with the PFL model’s weights (referred to as a “warm-up”), while the other plots compare PFL-only models to DFL models trained from scratch. Layer indices are shown on the y-axis. All plots share the color bar on the right, ranging from 0 to 1

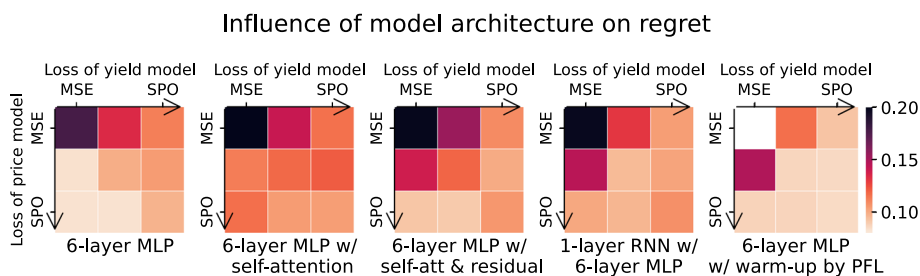


Fig. 4 Regret from decision-making, using parameters from models with different architectures and trained on different loss functions. The models are aligned with those used in the layer parameter comparisons (Fig. 3). The x- and y-axis follow the same configuration as in Fig. 1. All plots share the same color scale as on the right. The upper-left corner of the final plot is left empty, as it represents again a purely PFL-trained model

before fine-tuning with DFL, as shown in the final plot of Fig. 4. The impact is further discussed in Sect. 6.3.

Interestingly, as shown in Fig. 4, with increasing involvement of SPO loss during training, the regret performance of ATT models changes similarly to a much deeper MLP model, such as that with 12 layers as depicted in Fig. 1. Both architectures show weaker DFL performance compared to others, suggesting that models with very high learning capacity may be more prone to getting stuck in local optima. As illustrated in Fig. 3, the price prediction component in these models appears to be overly influenced by DFL, which may contribute to the performance decline. One potential solution is to use MLP outputs as residuals alongside the self-attention layers. As demonstrated in Fig. 3 and Fig. 4, this adjustment helps produce more balanced prediction biases and improves performance in both tasks. Furthermore, as indicated in Fig. 2, these models also achieve lower MSEs, indicating improved generalizability.

6.3 Pre-training mitigates the discrepancy by improving regret

Motivated by the strikingly similar feature patterns learned by PFL and DFL, we investigated using PFL as a form of pre-training – or “warm-up” – for the DFL process. As illustrated in

Fig. 4, incorporating SPO loss after such a warm-up significantly reduced the performance discrepancy between training and testing on regret, leading to improved overall performance. Notably, when comparing models trained with identical loss functions from scratch, the improvement is most pronounced when the model is trained exclusively with SPO loss. Further analysis in Fig. 3 reveals that this warm-up strategy results in stronger alignment of layer-wise biases with the original PFL model, compared to models where DFL is trained entirely from scratch. This suggests that warm-starting DFL encourages smoother and more consistent bias adaptation, which likely improves the models' generalizability.

It is important to note that in our problem, *yield* naturally serves as a soft constraint of the objective function, whereas *price* primarily incentivizes the decision direction, rather than its magnitude. This difference affects how DFL introduces bias into the two prediction models. Since decision feasibility is highly sensitive to changes in the yield model, preserving its pre-trained structure is critical. With warm-started DFL, the yield model changes very little, helping to avoid harmful bias and supporting stable decision-making. Conversely, the price model can tolerate – or can even benefit from – certain bias introduced during DFL, as this bias helps align predictions with task-specific goals. Still, compared to training from scratch, the bias introduced by performing DFL as a fine-tuning process remains more moderate, reflecting a better balance between task adaptation and generalization.

Considering that price and yield predictions function differently, we conducted comparative experiments to assess the effectiveness of selectively warming up the training of each model. As is shown in Fig. 5, initializing the yield prediction model with pre-trained weights proves essential for solving the optimization problem effectively, as it ensures that decisions are made under more realistic constraints, which contributes to the lower regret observed. These findings reinforce that, for the yield model, preserving accuracy is more important than inducing bias aligned with the task. In other words, an accurate estimation of feasible regions provides a more reliable starting point for the following DFL phase.

In the above experiments, the warm-up phase lasts until the model converges over 20 epochs based on MSE. As illustrated in Fig. 6, the impact of warm-up on final performance also depends on how well the model is pre-trained. To further explore this, we compared our convergence-based, dynamic approach against using a fixed warm-up of 6 epochs, as used by (Mandi, Demirovi Stuckey, and Guns, 2020). While a short fixed warm-up can still offer some benefit, our results suggest that a convergence-based schedule with sufficient patience contributes to more robust and consistently better performance.

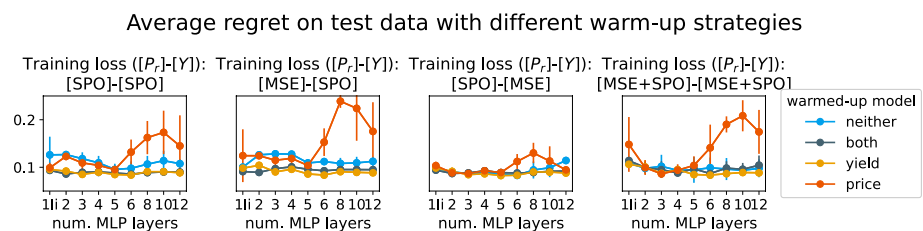


Fig. 5 Comparison of the effect of the pre-training strategy by the regret, of models with different training losses. The color codes show which model(s) was/were pre-training on the MSE. The x-axis shows the model depth, with “1li” referring to linear regression

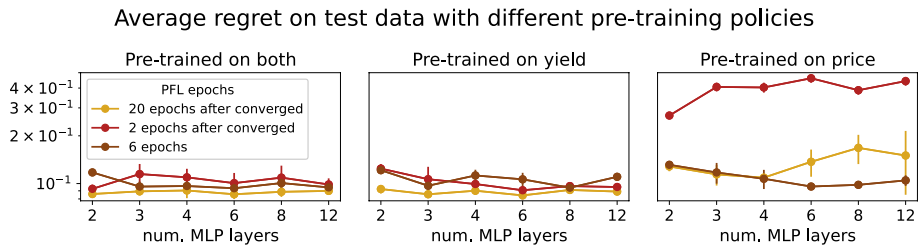


Fig. 6 Resulting regret of models warmed up differently. The titles indicate which model(s) was/were pre-trained. The color shows the periods of pre-training

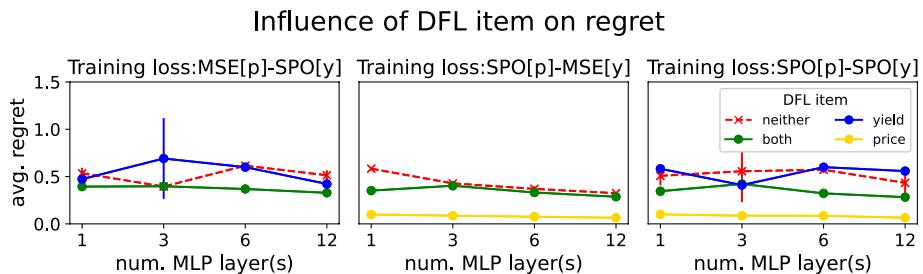


Fig. 7 Regret of MLP models when one parameter is considered as known in the DFL task, compared to previous PFL and (both-)DFL models' performances. The subtitles indicate the specific loss function of the yield ([y]) and price ([p]) prediction model. The x-axis shows the depth of the MLP model

Meanwhile, implementing a warm-up process for both models reduces the average training time from 1.23 h to 0.79 h³. This time-saving effect is particularly notable when only the yield prediction model undergoes the warm-up process. In contrast, warming up only the price prediction model does not result in a significant reduction in computational time.

6.4 The sub-problems unveiled as classical DFL tasks

By using the ground truth of one parameter and implementing DFL with SPO loss on the other, we simplify the real-world problem to a more ideal and classical DFL task. From the middle plot in Fig. 7, it is clear that SPO is a powerful solution when the yield is known. In this case, where all constraints are fixed, the problem setting closely resembles the one proposed by (Elmachtoub and Grigas, 2022), underscoring the strength of SPO under such conditions. Thus, applying DFL to the task of price prediction represents a meaningful and novel advancement over prior approaches.

However, the scenario changes when prices across all trading channels are known and the prediction target shifts to yield. In this setting, yield directly influences key operational outcomes such as overproduction costs and supply shortage penalties. As a result, achieving high prediction accuracy becomes even more crucial. This shift is illustrated by the third subplot in Fig. 7: the varying gaps between the three solid lines further emphasize how the benefits of DFL depend significantly on the decision parameters involved and their specific roles within the optimization.

³ Averaged across all models in Fig. 5 and all seeds. More comparisons can be referred to in the appendix.

Furthermore, when reviewing the training progress in detail, we observed that gradient vanished several times when training the price prediction model solely on the SPO loss. This issue, which is known as the zero-gradient problem in several DFL studies (Mandiet et al. 2023), occurred because price, in this context, functions purely as a decision-driven parameter. Consequently, without incorporating a regularization strategy or a warm-up phase using MSE during training, the model can become significantly biased, as demonstrated in Fig. 3.

6.5 Real-world impact

In previous sections, we focused on model performance across various training losses and strategies, using the “*wind-ori*” dataset as an illustrative example due to its realistic and sufficient data. Here, we evaluate the regret from 3-layer MLP models across the four scenarios introduced in Sect. 5.4: the original wind farm case (“*wind-ori*”), a modified version with more balanced price signals (“*wind-50*”), and two tomato-grower scenarios (“*tomato-c*” and “*tomato-m*”), where underproduction incurs heavier penalties than overproduction, contrasting with the wind farm cases. We also consider the real-world impact of DFL by evaluating mean error (ME) across the four cases and the profitability in the “*wind-ori*” scenario. A summary of the resulting regrets under different learning settings are depicted in Fig. 8. To better understand the real-world implications of applying DFL, we also evaluate the mean error (ME) across these four cases, as described in Table 1.

Table 1 shows that DFL intentionally introduces bias in both yield and price predictions to account for asymmetric penalties – yield predictions are skewed to avoid costly underproduction, while price predictions exaggerate contract-market price differences to favor more profitable decisions. These biases reduce overall regret compared to models trained with MSE alone, which suffer from symmetric error distributions that hurt downstream tasks.

Nonetheless, as Fig. 8 illustrates, regret is a challenging loss for models to optimize directly. This difficulty is especially evident when training jointly on regret for both tasks, resulting in a noticeable gap between training and testing performance. Interestingly, we find that lower bias in yield prediction often aligns with reduced regret in these cases. This suggests that carefully controlled bias in DFL can improve prediction models and lead to better downstream outcomes.

Fig. 8 Average regret of price and yield prediction models in different experiment cases. The loss function is indicated in the x and y labels respectively. The wind cases share the color bar in the first row and the tomato cases share that in the second row on the right

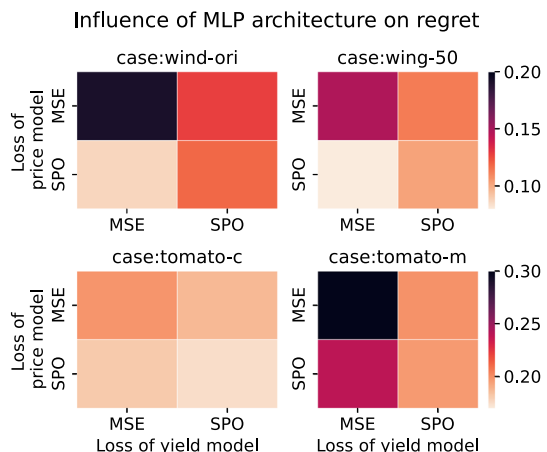


Table 1 ME of price and yield predictions in different experiment cases

Case	Price		Yield	
	Training Loss	Testing ME	Training Loss	Testing ME
Tomato-c	MSE	0.26	MSE	− 0.07
	MSE	0.23	SPO	0.34
	SPO	− 1.31	MSE	− 0.07
	SPO	− 1.32	SPO	0.26
Tomato-m	MSE	1.10	MSE	− 0.02
	MSE	1.01	SPO	0.02
	SPO	− 1.18	MSE	− 0.04
	SPO	− 1.32	SPO	− 0.04
Wind-ori	MSE	− 0.08	MSE	− 0.02
	MSE	− 0.01	SPO	0.50
	SPO	1.74	MSE	− 0.05
	SPO	0.76	SPO	0.05
Wind-50	MSE	− 0.08	MSE	− 0.03
	MSE	− 0.22	SPO	0.47
	SPO	2.11	MSE	− 0.05
	SPO	− 0.23	SPO	0.40

The loss function is indicated by the 2nd and 4th columns

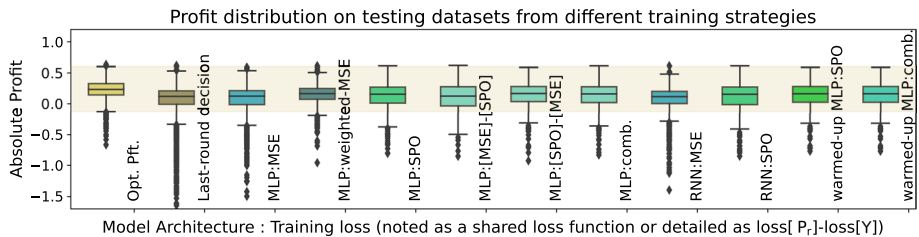


Fig. 9 Profit distribution in the “wind-ori” case, with a 6-layer MLP model. The [architecture:training loss] configuration is indicated next to each box plot. The distribution plots are color-coded to distinguish the loss functions and training strategies. Profits from optimal decisions based on parameter realizations are shown in the yellow box, with the lighter yellow band serving as a value reference for other distributions

Figure 9 shows the profiting performance of selected models and two more case-specific baselines: one used the optimal decision from the latest parameter realization (“last-round decision”), and another trained models using MSE weighted by under- and over-supply costs, i.e. a_{c2} and a_{c3} in Eq. 3 (“MLP: weighted-MSE”). These baselines, along with the two blue boxplots, represent traditional two-stage approaches: first the forecasting models are trained independently, and then decisions are optimized based on their predictions. Moreover, while weighting MSE helps reduce extreme losses, it falls short of matching the profit levels achieved by DFL-trained models. This suggests that simply improving the forecasting model has limited impact unless the prediction is aligned more directly with the downstream objective.

Across all boxplots, the models with DFL applied on top of PFL-pretrained models (cyan boxes) stand out by both raising the upper bound of achievable profits and reducing losses in the lower tail. This indicates that their improved performance comes not just from higher average returns, but from a more favorable overall distribution – achieving stronger gains

while limiting downside risk. The stability introduced by PFL helps maintain reliable yield predictions, which in turn supports the SPO algorithm in enforcing constraints more effectively. This enables DFL to better guide the prediction of decision parameters and ultimately leads to more robust decision-making outcomes.

7 Discussion

In this study, we investigated the comparative effectiveness of prediction-focused learning (PFL) and decision-focused learning (DFL) within the “*smart predict-then-optimize*” (SPO) framework. We empirically assess model performance through a real-world task: making selling decisions in an integrated contract and (real-time) market trading system. Our research extends existing DFL studies by (i) incorporating diverse neural network architectures and exploring the impact of their learning capabilities, and (ii) addressing a more intricate decision-making problem, where models collaboratively predict two types of variables: market *price* as a decision parameter and *yield* as both a decision parameter and a soft constraint.

Our benchmarks underlined a notable performance discrepancy: Models trained solely on the *regret*, which is also known as the SPO loss, do not consistently lead decisions on the test set to achieve the lowest regret. This inconsistency appears across various model architectures and is especially pronounced in simpler models. Although we have explored a variety of network architectures, our study covers only a small slice of the landscape of deep learning. Future work aiming to discover the optimal network depth and architecture in more problem settings may benefit from metaheuristic methods such as those proposed in (Kumar and Bhasker, 2023) and (Akay et al., 2022).

In terms of the efficacy of DFL and PFL, our investigation into model hyperparameters demonstrates that although models trained with Mean Squared Error (MSE) and SPO loss exhibit different biases, they tend to converge on similar layer weights and self-attention patterns. Considering that PFL is more computationally efficient, it gives a clear motivation for enhancing DFL by starting from models pre-trained with MSE. This warm-up approach offers three main benefits: (a) narrowing the training–testing performance gap by boosting DFL effectiveness; (b) reducing computational cost; and (c) improving generalization by regularizing the bias introduced during DFL training.

In our specific use cases involving decision making as a wind farm operator or a tomato grower, our findings highlight two key insights: DFL can change price predictions compared to PFL, while accurate yield prediction is especially important when the yield serves as a constraint in the decision-making process. This difference comes from the distinct roles that price and yield play in the objective functions. In this way, our findings offer clearer insights into the SPO method’s applicability and highlights the importance of modeling constraint-related variables with care. Similar advantages were also evidenced in other constrained decision-making scenarios, such as safe reinforcement learning (Yang et al. 2023).

In all case studies, the same input features were used for both prediction tasks, naturally motivating the use of multitask learning (MTL) to improve learning and data efficiency. Prior work (Tang and Khalil 2023) has shown that MTL with DFL can be especially effective for smaller datasets. While we do not evaluate MTL in the wind farm setting due to its larger dataset size, it presents a promising direction for sparser cases like the tomato price dataset discussed in Sect. 6.5.

Acknowledgements We gratefully acknowledge the generous support and funding of this project from Tops-ector Tuinbouw & Uitgangsmaterialen (The Netherlands), Innovatiefonds Hagelunie, and Interpolis. We also extend our sincere thanks to Delphy Improvement Centre B.V. and FruitMasters B.V. for their valuable insights and contributions regarding their participation in the supply chain.

Author contributions Junhan Wen conducted the research and led the development of the manuscript. Thomas Abeel and Mathijs de Weerd supervised the project, provided guidance throughout the research, and contributed to the writing and revision of the paper. All authors reviewed and approved the final version of the manuscript.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no Conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aguiar, N., Gupta, V., & Khargonekar, P. P. (2020). A real options market-based approach to increase penetration of renewables. *IEEE Transactions on Smart Grid*, 11(2), 1691–1701.
- Akay, B., Karaboga, D., & Akay, R. (2022). A comprehensive survey on optimizing deep learning models by metaheuristics. *Artificial Intelligence Review*, 55(2), 829–894.
- Block, C., Neumann, D., & Weinhardt, C. (2008). A market mechanism for energy allocation in micro-chp grids. Proc. of the 41st Annual Hawaii International Conference on System Sciences (hicc 2008) (pp. 172–172).
- Borsa, N. (2023). Tomato prices. <http://novataborsa.com/%D1%86%D0%B5%D0%BD%D0%B8-%D0%B4%D0%BE%D0%BC%D0%B0%D1%82%D0%B8/>. (Accessed on: August 2023)
- Brealey, R.A. (2000). Stock prices, stock indexes and index funds. Bank of England Quarterly Bulletin
- Brooks, C., & Garrett, I. (2002). Can we explain the dynamics of the UK FTSE 100 stock and stock index futures markets? *Applied Financial Economics*, 12(1), 25–31.
- Chang, X., Wu, J., Sun, H., & Yan, X. (2023). A smart predict-then-optimize method for dynamic green bike relocation in the free-floating system. *Transportation Research Part C: Emerging Technologies*, 153, Article 104220.
- Chen, W., Zhang, H., Mehlatat, M. K., & Jia, L. (2021). Mean-variance portfolio optimization using machine learning-based stock price prediction. *Applied Soft Computing*, 100, Article 106943.
- Clark, E., Qiao, Z., & Wong, W.-K. (2016). Theories of risk: testing investor behavior on the Taiwan stock and stock index futures markets. *Economic Inquiry*, 54(2), 907–924.
- Conejo, A.J., Carrión, M., & Morales, J.M. (2010). Decision making under uncertainty in electricity markets (Vol. 1).
- Connor, J. T., Martin, R. D., & Atlas, L. E. (1994). Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, 5(2), 240–254.
- Diamond, S., & Boyd, S. (2016). Cvxpy: A python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83), 1–5.
- Dileep, B., Grover, R., & Rai, K. (2002). Contract farming in tomato: An economic analysis. *Indian Journal of Agricultural Economics*, 57(2), 197–210.
- Donti, P., Amos, B., & Kolter, J. Z. (2017). Task-based end-to-end model learning in stochastic optimization. *Advances in Neural Information Processing Systems*, 30, 5484–5494.

- Elmachtoub, A. N., & Grigas, P. (2022). Smart predict, then optimize. *Management Science*, 68(1), 9–26.
- Fathallahi, F., Ponnambalam, K., Huang, Y., & Karray, F. (2020). Agent-based modeling to simulate real-world prices: A strawberry market study. 2020 IEEE International Conference on Systems, Man, and Cybernetics (smc) (p.3606–3611).
- Hu, X., Lee, J.C., & Lee, J.H. (2023). Predict+optimize for packing and covering LPs with unknown parameters in constraints. (Vol. 37, p.3987–3995).
- Hu, X., Lee, J.C., & Lee, J.H. (2023). Two-stage predict+optimize for MILPs with unknown parameters in constraints. Thirty-Seventh Conference on Neural Information Processing Systems.
- Hueth, B., Ligon, E., Wolf, S., & Wu, S. (1999). Incentive instruments in fruit and vegetable contracts: input control, monitoring, measuring, and price risk. *Applied Economic Perspectives and Policy*, 21(2), 374–389.
- Jhana, N. (2019). Hourly energy demand generation and weather. <https://www.kaggle.com/datasets/nicholasjghana/energy-consumption-generation-prices-and-weather>. (Accessed on: October, 2023)
- Kirschen, D.S., & Strbac, G. (2018). Fundamentals of power system economics.
- Kotary, J., Fioretto, F., Van Hentenryck, P., & Wilder, B. (2021). End-to-end constrained optimization learning: A survey. [arXiv:2103.16378](https://arxiv.org/abs/2103.16378)
- Krishnamurthy, D., Uckun, C., Zhou, Z., Thimmapuram, P. R., & Botterud, A. (2018). Energy storage arbitrage under day-ahead and real-time price uncertainty. *IEEE Transactions on Power Systems*, 33(1), 84–93.
- Kumar, S.V., & Bhasker, R. (2023). An artificial rabbit optimized neural network-based hybrid dc-ac microgrid energy management system. 2023 7th International Conference on Computing, Communication, Control and Automation (icccubc) (pp. 1–7).
- Lee, J., Guo, J., Choi, J. K., & Zukerman, M. (2015). Distributed energy trading in microgrids: A game-theoretic model and its equilibrium analysis. *IEEE Transactions on Industrial Electronics*, 62(6), 3524–3533.
- Liu, G., Xu, Y., & Tomsovic, K. (2016). Bidding strategy for microgrid in day-ahead market based on hybrid stochastic/robust optimization. *IEEE Transactions on Smart Grid*, 7(1), 227–237.
- Liu, H., & Grigas, P. (2021). Risk bounds and calibration for a smart predict-then-optimize method. *Advances in Neural Information Processing Systems*, 34, 22083–22094.
- Liu, H., & Grigas, P. (2022). Online contextual decision-making with a smart predict-then-optimize method. [arXiv preprint](https://arxiv.org/abs/2205.12345)
- Mandi, J., Bucarey, V., Tchomba, M.M.K., & Guns, T. (2022). Decision-focused learning: Through the lens of learning to rank. K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato (Eds.), *Proceedings of the 39th international conference on machine learning* (Vol. 162, pp. 14935–14947).
- Mandi, J., Demirović, E., Stuckey, P.J., & Guns, T. (2020). Smart predict-and-optimize for hard combinatorial optimization problems. *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, p.1603–1610).
- Mandi, J., Kotary, J., Berden, S., Mulamba, M., Bucarey, V., Guns, T., & Fioretto, F. (2023). Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. [arXiv preprint](https://arxiv.org/abs/2305.12345)
- Nate Gruver, S.Q., Marc Finzi, & Wilson, A.G. (2023). Large language models are zero shot time series forecasters. *Advances in Neural Information Processing Systems*.
- Ncube, D. (2020). The importance of contract farming to small-scale farmers in africa and the implications for policy: A review scenario. *The Open Agriculture Journal*. <https://doi.org/10.1126/sciadvg.2248>
- Rahimiyan, M., & Baringo, L. (2016). Strategic bidding for a virtual power plant in the day-ahead and real-time markets: A price-taker robust optimization approach. *IEEE Transactions on Power Systems*, 31(4), 2676–2687.
- Ren, L., Mo, T., Cheng, X., & Li, X. (2023). Temporal-frequency attention focusing for time series extrinsic regression via auxiliary task. *IEEE Transactions on Neural Networks and Learning Systems*, 1–13, 15494–15506.
- Safdarian, A., Fotuhi-Firuzabad, M., & Lehtonen, M. (2015). A medium-term decision model for discos: forward contracting and tou pricing. *IEEE Transactions on Power Systems*, 30(3), 1143–1154.
- Sang, L., Xu, Y., Long, H., Hu, Q., & Sun, H. (2022). Electricity price prediction for energy storage system arbitrage: A decision-focused approach. *IEEE Transactions on Smart Grid*, 13(4), 2822–2832.
- Schieffer, J., & Vassalos, M. (2015). Risk and the use of contracts by vegetable growers. *Choices*, 30(3), 1–4.
- Sen, S., Yu, L., & Genc, T. (2006). A stochastic programming approach to power portfolio optimization. *Operations Research*, 54(1), 55–72.
- Singh, S. (2005). Role of the state in contract farming in thailand: experience and lessons. *ASEAN Economic Bulletin*, 22(2), 217–228.
- Tang, B., & Khalil, E.B. (2022). PyEPO: A PyTorch-based end-to-end predict-then-optimize library for linear and integer programming. [arXiv:2206.14234](https://arxiv.org/abs/2206.14234)

- Tang, B., & Khalil, E.B. (2023). Multi-task predict-then-optimize. *International Conference on Learning and Intelligent Optimization* (pp. 506–522).
- Tian, X., Yan, R., Liu, Y., & Wang, S. (2023). A smart predict-then-optimize method for targeted and cost-effective maritime transportation. *Transportation Research Part B: Methodological*, 172, 32–52.
- van Staden, H. E., Deprez, L., & Boute, R. N. (2022). A dynamic predict, then optimize preventive maintenance approach using operational intervention data. *European Journal of Operational Research*, 302(3), 1079–1096.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*. <https://doi.org/10.48550/arXiv.1706.03762>
- Verma, S., Mate, A., Wang, K., Taneja, A., & Tambe, M. (2022). Case study: Applying decision focused learning in the real world. *Workshop on Trustworthy and Socially Responsible Machine Learning, neurips 2022*.
- Vu, Q.H., Cen, L., Ruta, D., & Liu, M. (2022). Key Factors to Consider when Predicting the Costs of Forwarding Contracts. *Proc. of the 17th Conference on Computer Science and Intelligence Systems, Fedcsis 2022* (pp. 447–450).
- Wang, K., Wilder, B., Perrault, A., & Tambe, M. (2020). Automatically learning compact quality-aware surrogates for optimization problems. *Advances in Neural Information Processing Systems*, 33, 9586–9596.
- Wang, L., Ahmad, F., Luo, G., Umar, M., & Kirikkaleli, D. (2022). Portfolio optimization of financial commodities with energy futures. *Annals of Operations Research*, 313(1), 401–439.
- Wilder, B., Dilkina, B., & Tambe, M. (2019). Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, pp. 1658–1665).
- Wu, M.-T., & Tsai, C.-W. (2024). Training-free neural architecture search: A review. *ICT Express*, 10(1), 213–231.
- Yan, R., Wang, S., & Fagerholt, K. (2020). A semi-smart predict then optimize (semi-spo) method for efficient ship inspection. *Transportation Research Part B: Methodological*, 142, 100–125.
- Yang, Q., Simão, T.D., Jansen, N., Tindemans, S.H., & Spaan, M.T. (2023). Reinforcement learning by guided safe exploration. [arXiv:2307.14316](https://arxiv.org/abs/2307.14316)
- Yang, X., Fan, L., Li, X., & Meng, L. (2023). Day-ahead and real-time market bidding and scheduling strategy for wind power participation based on shared energy storage. *Electric Power Systems Research*, 214, Article 108903.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Junhan Wen¹ · Thomas Abeel¹ · Mathijs de Weerd¹

✉ Mathijs de Weerd
m.m.deweerd@tudelft.nl

Junhan Wen
junhan.wen@tudelft.nl

Thomas Abeel
t.abeel@tudelft.nl

¹ Delft University of Technology, Van Mourik Broekmanweg 6, 2628XE Delft, Netherlands