Decision-Focused Probabilistic Forecast Combination

Akylas Stratigakos, Salvador Pineda, Juan Miguel Morales

Decision-Focused Probabilistic Forecast Combination

Akylas Stratigakos[a,*], Salvador Pineda[b], Juan Miguel Morales[b]

[a] *Department of Electrical and Electronic Engineering, Imperial College London, London, SW7 2AZ, U.K.*
[b] *OASYS Research Group, University of Málaga, Málaga, 29010, Spain*

Abstract

In real-world settings, decision-makers often have access to multiple forecasts for the same unknown quantity. Combining different forecasts has long been known to improve forecast quality, as measured by scoring rules in the case of probabilistic forecasting. However, improved forecast quality does not always translate into better decisions in a downstream problem that utilizes the resultant combined forecast as input. To this end, this work proposes a novel probabilistic forecast combination approach that accounts for the downstream stochastic optimization problem by which the decisions will be made. Specifically, we propose a linear pool of probabilistic forecasts where the respective weights are learned by minimizing the expected decision cost of the induced combination, which we formulate as a nested optimization problem. Two methods are proposed for its solution: a gradient-based method that utilizes differential optimization layers and a performance-based weighting method. For experimental validation, we examine two integral problems associated with renewable energy integration in modern power systems and compare them against well-established combination methods based on linear pooling. Namely, we examine an electricity market trading problem under stochastic solar production and a grid scheduling problem under stochastic wind production. The results illustrate that the proposed decision-focused combination approach leads to lower expected downstream costs while optimizing for forecast quality when estimating linear pool weights does not always translate into better decisions. Notably, optimizing for a combination of downstream cost and a standard scoring rule consistently leads to better decisions while maintaining high forecast quality.

*Corresponding author.
Email address: a.stratigakos@imperial.ac.uk (Akylas Stratigakos)

Keywords: Probabilistic forecasting, Forecast combination, Decision-focused learning, Prescriptive analytics, Linear pool, Differentiable optimization

1. Introduction

1.1. Context and Motivation

Real-world optimization problems typically involve uncertain parameters, corresponding to quantities such as product demand or market prices which are unknown at the solution time. Classical stochastic optimization (Birge and Louveaux, 2011) assumes that these uncertain parameters follow a known distribution. In reality, as only observational data are available, uncertainty distributions are implicitly estimated through short-term probabilistic forecasting, which becomes a critical input in decision-making pipelines.

In many real-world settings, decision-makers often have multiple forecasts for the same uncertain parameter, provided by different vendors or experts. This setting is common, for instance, in power and energy systems with a high penetration of variable renewable energy sources. To ensure a reliable and cost-effective operation, grid operators further employ domain experts, such as meteorologists and power system engineers, who, given additional available information such as current weather conditions, select or combine individual forecasts (Motley, 2023). The resultant combined forecasts are subsequently inputted within operational workflows, such as market clearing algorithms and production cost models.

Ultimately, the decision-maker wants to combine forecasts to minimize her expected decision cost in a downstream problem. Recently, several works have illustrated that increased forecast quality (either in a point or probabilistic sense) does not always translate into better decisions (Mandi et al., 2023). This effect becomes even more pronounced in risk-critical infrastructures, such as power systems, where forecast errors can induce highly asymmetrical costs (Morales et al., 2023). Hence, it is critical to go beyond forecast quality and explicitly consider forecast *value* within a decision-making pipeline when implementing forecast combinations.

1.2. Literature Review

Probabilistic Forecasting. Gneiting et al. (2007) posit that probabilistic forecasting aims to maximize sharpness subject to calibration. Forecast quality is evaluated based on so-called *proper scoring rules* (Gneiting and Raftery,

2007), which compare probabilistic forecasts against realizations of a random target variable, jointly assessing sharpness and calibration and eliciting an honest uncertainty estimation by the expert forecaster. As the decision-maker might not be uniformly interested in forecast quality across the whole distribution, weighted scoring rules can be utilized to evaluate probabilistic forecasts with emphasis on specific regions of interest (Gneiting and Ranjan, 2011).

Contextual Stochastic Optimization. Closely related to probabilistic forecasting is the area of contextual stochastic optimization, i.e., stochastic optimization where the uncertain parameters are associated with some contextual information (or *features*). For instance, renewable energy production is associated with the weather, market-clearing prices depend on demand, etc. The goal is to minimize an expected decision cost given a realization of feature data (Bertsimas and Kallus, 2020). The standard two-step approach involves forecasting uncertain parameters (or their distributions) conditioned on available features and then solving an optimization problem. Recently, there has been a growing interest in decision-focused probabilistic forecasting (Donti et al., 2017; Stratigakos et al., 2022; Grigas et al., 2021; Kallus and Mao, 2022) which embeds the downstream problem within the training process. Note that a perfect probabilistic forecast, i.e., one that coincides with the true probability distribution, would always be preferred by decision-makers, regardless of their downstream objective (Diebold et al., 1998). However, a degree of model misspecification and, subsequently, forecast error is unavoidable, which motivates a decision-focused approach that aligns the estimation of the probabilistic model with the downstream objective, implicitly accounting for different regions of interest (Gneiting and Ranjan, 2011). Such an approach regularly outperforms the standard "forecast, then optimize" approach in several problems across multiple industries, such as power and energy systems (Chen et al., 2022).

Combining Probabilistic Forecasts. Forecast combination has long been known to improve forecast quality over component forecasts (Wang et al., 2023), both in point and probabilistic cases. The prevalent method for combining forecast distributions is the linear pool, which dates back at least to Stone (1961); Winkler (1968), where each expert is assigned a weight that signifies her forecast skill. An equally weighted linear pool, termed *ordinary linear pool*, has been proven fairly robust and is generally hard to beat in empirical evaluations. Linear pooling is typically used to combine, or average, probabilities in the form of cumulative density functions. Raftery et al. (2005)

linearly pool weather ensembles using Bayesian model averaging to generate probabilistic forecasts. Lichtendahl Jr et al. (2013) examine conditions under which it is favorable to average quantiles rather than probabilities, i.e., linear pooling of quantile functions. Papayiannis and Yannacopoulos (2018) show that quantile averaging is a special case of the Wasserstein barycenter and further propose learning optimal combination weights by minimizing the Wasserstein distance of the combined forecast and observed data. Gneiting and Ranjan (2013) highlight some shortcomings of linear pooling, namely, a linear pool of calibrated forecasts may be uncalibrated and underdispersed, and further propose nonlinear pooling based on the Beta transformation.

Learning weights of linear pools. Despite the shortcomings highlighted by Gneiting and Ranjan (2013), linear pooling performs very well in practice and is the most popular combination method. This can be attributed to the fact that individual probabilistic forecasts are usually underdispersed hence the linear pooling combination improves overall forecast quality. Given the robustness of simple approaches, Wang et al. (2023) posit that forecast combinations should aim to be "sophisticatedly simple." The question thus arises: *How to properly tune the weights of a linear pool?* Recent works focus on minimizing a proper scoring rule, such as the continuous ranked probability score (CRPS). Thorey et al. (2017) minimize a bias-corrected CRPS for ensemble forecasting and Thorey et al. (2018) further extend this work to probabilistic solar production forecasting. van der Meer et al. (2024) develop an online algorithm to adaptively minimize the CRPS of a beta-transformed linear pool of renewable production forecasts. Berrisch and Ziel (2023) observe differences in forecast quality across various distribution regions and propose a pointwise quantile combination minimizing the aggregate CRPS. Krannichfeldt et al. (2022) average quantiles by minimizing the quantile score, which approximates CRPS, and updating the combination weights only when the loss exceeds a predetermined threshold.

1.3. Aim and Contribution

In this work, we develop a decision-focused approach to estimate linear pool weights for probabilistic forecasting. Given a data-driven, contextual stochastic optimization problem, we learn combination weights by minimizing the expected incurred decision cost. We propose two solution methods for the resultant problem, namely, a differentiable optimization-based method (Agrawal et al., 2019a) and weighting component forecasts based on their in-sample decision quality, which adapts the classic approach of Bates and Granger (1969) in a contextual optimization framework. Further, we

extend our approach to a conditional combination setting, wherein the combination weights depend on additional features that become available before making decisions, which is of great practical interest. Importantly, we make two key contributions: *(i)* we connect the weight estimation process with the downstream decision-making process while retaining a simple combination scheme, and *(ii)* account for additional information that might be available to the decision-maker. We validate the proposed combination approach in critical applications related to variable renewable energy sources integration in power systems (Morales et al., 2013). We examine an electricity market trading problem under stochastic solar production and a grid scheduling problem under stochastic wind production. The numerical results show that embedding the downstream problem by co-optimizing decision costs and forecasting quality consistently improves decision outcomes. Notably, optimizing for a combination of downstream cost and CRPS consistently leads to better decisions while maintaining high forecast quality.

The rest of the paper is organized as follows. Section 2 presents preliminaries, formulates the proposed combination approach, and develops the solution methods. Section 3 presents the results of the numerical experiments. Section 4 summarizes and discusses future work.

2. Methodology

This section introduces our notation and develops the proposed methodology. Section 2.1 introduces contextual stochastic optimization, probabilistic forecast evaluation, and forecast combination. Section 2.2 formulates the proposed decision-focused forecast combination approach, which is our main contribution. Finally, Section 2.3 develops two solution methodologies for the decision-focused forecast combination.

Notation. Uppercase letters denote random variables, lowercase letters denote realizations, and $\hat{\cdot}$ denotes forecasts. Bold font denotes vectors and normal font denotes scalar quantities. Sets are denoted with calligraphic font, e.g., $\mathcal{S}$, and $|\mathcal{S}|$ denotes the cardinality (number of elements) of a set $\mathcal{S}$. The notation $[n]$ is used as a shorthand for $1, ..., n$. Further, let

$$\Sigma_n = \{\mathbf{a} \in \mathbb{R}^n_+ \,|\, \mathbf{a}^\top \mathbf{1}_n = 1\}$$

be the standard $(n-1)$-dimensional probability simplex, where $\mathbf{1}_n$ is an n-sized vector of ones. Finally, let $\mathbb{I}(\cdot)$ be the indicator function.

2.1. Methodology Preliminaries

2.1.1. Contextual Stochastic Optimization

We consider a contextual stochastic optimization problem given by

$$\min_{\mathbf{z}\in\mathcal{Z}} \mathbb{E}[c(\mathbf{z};Y)|\mathbf{X}=\mathbf{x}_0], \tag{1}$$

where $Y \in \mathcal{Y}$ denotes the uncertain problem parameters (e.g., uncertain demand), $\mathbf{X} \in \mathcal{X}$ denotes some associated contextual features (e.g., weather or market conditions), $\mathbf{x}_0$ denotes a realization of $\mathbf{X}$, $\mathbf{z}$ denotes the decision variables, $\mathcal{Z}$ denotes the convex set of feasible solutions, c denotes a convex cost function, and the expectation is taken with respect to (w.r.t.) the conditional distribution of Y given $\mathbf{X} = \mathbf{x}_0$. The set of feasible decisions $\mathcal{Z}$ may also depend on the uncertainty Y, e.g., a demand balancing constraint in a network flow problem, but the dependency is suppressed to keep the notation simple.

We further assume that the uncertain parameter Y is a discrete random variable with finite support $\mathcal{Y} \stackrel{\text{def}}{=} \{\xi_1, \ldots, \xi_K\}$, where K is the number of support locations. For any instance $\mathbf{x}_0 \in \mathcal{X}$, the true conditional distribution of Y is given by a discrete probability vector (or histogram) $\mathbf{p}(Y \,|\, \mathbf{x}_0) \in \Sigma_K$, where Σ_K is the $(K-1)$-dimensional probability simplex. The k-th component of $\mathbf{p}(Y \,|\, \mathbf{x}_0)$ is defined as $p_k = \mathbb{P}(Y = \xi_k|\mathbf{x}_0)$, i.e., the conditional probability of $Y = \xi_k$. Problem (1) can be equivalently written as

$$\min_{\mathbf{z}\in\mathcal{Z}} \mathbb{E}[c(\mathbf{z};Y)|\mathbf{x}_0] = \min_{\mathbf{z}\in\mathcal{Z}} \sum_{k\in[K]} p_k c(\mathbf{z};\xi_k). \tag{2}$$

In real-world applications, instead of the true probability vector $\mathbf{p}(Y \,|\, \mathbf{x}_0)$, we have access to some training data, which are used to approximate (2) by estimating the conditional distribution of Y through probabilistic forecasting. Let us further assume that S experts provide the decision-maker with a set of probabilistic forecasts $\{\hat{\mathbf{p}}^s\}_{s\in[S]}$ that model the conditional distribution of Y and that the decision-maker has collected a data set of N historical observations of $\mathcal{D} = \{(y_i, \hat{\mathbf{p}}_i^1, \ldots, \hat{\mathbf{p}}_i^S)\}_{i\in[N]}$ of the uncertain parameter and the respective probabilistic forecasts of each expert. Typically, all probability vectors are conditioned on some contextual information, with each expert potentially using a different set of features to model uncertainty; this dependency is suppressed here to simplify the notation.

2.1.2. Probabilistic Forecast Evaluation

Forecast Quality. The merit of each expert is typically assessed based on the quality of her forecasts. Gneiting et al. (2007) posit that probabilistic

forecasting aims to maximize the sharpness of the predictive distributions subject to calibration. Calibration refers to the statistical consistency between the induced probabilistic forecasts and the realizations of uncertainty, while sharpness refers to the concentration of the predictive distribution. Both calibration and sharpness can be assessed jointly through so-called proper scoring rules.

Any probability vector, e.g., $\mathbf{q}$, induces a estimation for the binary event $\{Y \leq u\}$ via its associated cumulative distribution function (CDF)

$$F(u) = \int_{-\infty}^{u} \mathbf{q}(y)dy = \sum_{y^k \leq u} q_k \tag{3}$$

and an associated quantile estimation via its associated inverse CDF $F^{-1}(\tau)$ at the level $\tau \in [0, 1]$.

The continuous ranked probability score (CRPS) (Gneiting and Raftery, 2007) is a strictly proper scoring rule that evaluates a predictive CDF with a realization of a random variable, i.e., a scoring rule that incentivizes the forecaster to provide truthful predictions about the probability distribution and is considered the state-of-the-art evaluation metric in probabilistic forecasting. Given a probability vector $\mathbf{q}$ and a realization y of Y, the CRPS is given

$$\text{CRPS}(\mathbf{q}, y) = \int_{-\infty}^{+\infty} (F(u) - \mathbb{I}(y \leq u))^2 du \tag{4}$$

$$= 2 \int_{0}^{1} (\alpha - \mathbb{I}\{y \leq F^{-1}(\alpha)\})(y - F^{-1}(\alpha))d\alpha. \tag{5}$$

The so-called quantile decomposition of CRPS (5) (Gneiting and Ranjan, 2013), shows that the CRPS has the same unit as the observation y. Even though closed-form solutions to (4) and (5) may not be available, the estimation of a discrete approximation is always feasible.

Decision Quality. In decision-focused learning, we are primarily interested in the quality of decisions induced by a (probabilistic) forecast, as measured by the expected value of (2). A salient notion is the so-called *regret*, i.e., the excess cost incurred compared to the perfect foresight solution. Given any $\mathbf{q} \in \Sigma_K$, let $\mathbf{z}(\mathbf{q}) = \arg\min_{\mathbf{z} \in \mathcal{Z}} \sum_{k \in [K]} q_k c(\mathbf{z}; \xi_k)$ be the corresponding induced decision. For a realization y_0 of uncertainty, the decision regret, w.r.t. the cost function c and the feasible set $\mathcal{Z}$, is estimated by

$$\text{Regret} = c(\mathbf{z}(\mathbf{q}); y_0) - c(\mathbf{z}^*; y_0), \tag{6}$$

where $\mathbf{z}^* = \arg\min_{\mathbf{z} \in \mathcal{Z}} c(\mathbf{z}; y_0)$. Evidently, regret is always non-negative.

2.1.3. Forecast Combination

Ultimately, the decision-maker wants to utilize the S available forecasts to minimize her downstream costs, defined by (2), or equivalently the decision regret (6). Formally, this translates into learning a function $h : \Sigma_K \times \cdots \times \Sigma_K \to \Sigma_K$ that takes as input S probability vectors $\{\hat{\mathbf{p}}^s\}_{s=1}^S$ and combines them in a decision-focused manner. Note that the decision-maker may also have access to additional contextual information, such as updated weather conditions, which can be used to improve the forecast combination. This gives rise to an augmented data set $\mathcal{D}' = \{(y_i, \hat{\mathbf{p}}_i^1, \ldots, \hat{\mathbf{p}}_i^S, \mathbf{x}_i)\}_{i\in[N]}$, where $\mathbf{x}_i$ are realizations of additional contextual information available to the decision-maker. The goal here would be to learn a forecast combination model $h : \mathcal{X} \times \Sigma_K \times \cdots \times \Sigma_K \to \Sigma_K$ that also accounts for these additional features. We refer to this case as an *conditional* combination, since it allows adapting to new contextual information and the case without additional feature data as a *static* combination.

2.2. Decision-focused Linear Pooling

In this section, we develop the proposed decision-focused forecast combination approach. We focus on linear pools of probabilistic forecasts, that is, we restrict the combination model h to be a linear function with non-negative coefficients that sum to one. While this might seem overly restrictive, plenty of empirical evidence suggests that the linear pool is robust and performs very well; extensions to non-linear pooling methods are straightforward to envision. Further, we allow the linear pool weights to be a function of additional contextual information, such as current weather conditions.

The proposed decision-focused forecast combination is given by

$$\min_{f\in\mathcal{F}} \quad \sum_{i\in[N]} \underbrace{c(\mathbf{z}_i^{\texttt{comb}}; y_i) - c(\mathbf{z}_i^*; y_i)}_{\text{Regret}} + \gamma \cdot \text{CRPS}(\mathbf{p}_i^{\texttt{comb}}, y_i), \tag{7a}$$

$$\text{s.t.} \quad \mathbf{z}_i^{\texttt{comb}} = \arg\min_{\mathbf{z}\in\mathcal{Z}} \sum_{k\in[K]} \hat{p}_{i,k}^{\texttt{comb}} c(\mathbf{z}; \xi_k), \qquad i \in [N], \tag{7b}$$

$$\hat{\mathbf{p}}_i^{\texttt{comb}} = \sum_{s\in[S]} \lambda_{i,s} \hat{\mathbf{p}}_i^s, \qquad i \in [N]. \tag{7c}$$

$$\boldsymbol{\lambda}_i \in \Sigma_S, \qquad i \in [N], \tag{7d}$$

$$\boldsymbol{\lambda}_i = f(\mathbf{x}_i), \qquad i \in [N]. \tag{7e}$$

Constraint (7b) finds the best decision given the combined forecast estimated from the linear pool (7c), for the i-th observation. Constraint (7d) ensures a convex combination of probabilistic forecasts and constraint (7e)

is the mapping from contextual information to combination weights. The objective function (7a) minimizes a combination of regret, i.e., incurred decision costs against the perfect foresight solution, and CRPS, controlled by hyperparameter γ. As $\gamma \to \infty$, we learn combination weights that minimize a discrete approximation of the CRPS for the induced forecast— see, e.g., (van der Meer et al., 2024).

The function f belongs in model class $\mathcal{F} : \mathcal{X} \to \Sigma_S$. The function classes considered in this work are linear models and feedforward neural networks. Note that the output of f needs to satisfy a set of constraints; a straightforward way to ensure this is using a softmax operator $\text{soft} : \mathbb{R}^S \to \mathbb{R}^S$ defined by $\text{soft}_s(v) = \frac{\exp(v_s)}{\sum_{j\in[S]}\exp(v_j)}$. For instance, the linear model would be $f(x) = \text{soft}(\mathbf{W}\mathbf{x} + \mathbf{b})$, where $\mathbf{W}$ are the linear coefficients and $\mathbf{b}$ is the bias. Evidently, if the decision-maker does not have access to additional features, then $\mathbf{x} = \mathbf{1}$ and problem (7) simplifies to one that estimates static combination weights $\boldsymbol{\lambda}$ ($\boldsymbol{\lambda}_i$ remains constant for all i in $[N]$).

Problem (7) is general enough to cover several applications of practical interest.

Deterministic Forecast Combination. Further, consider the case when the downstream optimization problem (2) is modeled as an expected-value problem. Then, the decision-focused combination problem (7) simplifies considerably to learning a decision-focused linear pool of point forecasts. For linear programming problems with unknown coefficients, the regret minimization term recovers the SPO loss (Elmachtoub and Grigas, 2022) and the CRPS term simplifies to the mean absolute error (Gneiting and Raftery, 2007) when considering point forecasts. Alternatively, one may want to use the mean-squared error as a regularization term; in that case, as γ grows, we converge to the method of Granger and Ramanathan (1984).

Multivariate Uncertainty. So far, we assumed that Y is a scalar variable. For the case of multivariate uncertainty, the Energy Score (Gneiting and Raftery, 2007), which is a proper scoring rule for multivariate probabilistic forecasts, can be used instead of the CRPS in (7a) (CRPS is a special case of the Energy Score).

2.3. Solution Methodology

Problem (7) is a non-convex, nested optimization problem. This section develops two solution methods, based on differential optimization layers and in-sample performance weighting.

2.3.1. Differential Optimization

We leverage the fact that the solution map of convex optimization problems (Agrawal et al., 2019a) is differentiable and propose a first-order, gradient-based method to approximate (7), akin to the method proposed by Agrawal et al. (2021).

Specifically, we use stochastic gradient descent to either tune problem parameters $\boldsymbol{\lambda}$ or, in the case of a conditional combination, learn the parameters of f. The learning process is described with the following pseudo-algorithm:

1. **Initialization:** Initialize model parameters f (for a conditional combination) or combination weights $\boldsymbol{\lambda}$ (for a static combination).
2. **Forward pass:** Sample $\mathcal{D}$, estimate linear pool of forecasts, and solve the inner maximization problems (7b).
3. **Regret and CRPS estimation:** Estimate the incurred decision cost (7a) and corresponding CRPS.
4. **Gradient estimation:** Estimate the gradient of the objective of (7a) w.r.t. model parameters, averaged over the batch size. Namely, we estimate:
   - The gradient of cost w.r.t. the decisions $\mathbf{z}$ induced by the forecast combination.
   - The gradient of decisions $\mathbf{z}$ w.r.t. the combination weights $\boldsymbol{\lambda}$.
   - The gradient of $\boldsymbol{\lambda}$ w.r.t. the parameters of the learning model f (this step applies only to the case of conditional combinations).
5. **Update parameters:** Update combination weights $\boldsymbol{\lambda}$ (for static combinations) or the parameters of f (for conditional combinations).
6. **Iterate:** Repeat steps 2-5 until convergence.

The main difficulty is evaluating the gradient of the decisions w.r.t. combination weights $\boldsymbol{\lambda}$, which requires differentiating through the $\arg\min$ operator of (7b). This is implemented through implicit differentiation of the Karush Kuhn Tucker (K.K.T.) equations (Amos and Kolter, 2017), which comes at a low computational cost. Note that the use of the softmax operator in the last layer of f ensures that $\boldsymbol{\lambda} \in \Sigma_S$, hence we do not require a projection step.

2.3.2. Performance-based Inverse Weighting

The main computational bottleneck of the gradient-based approach is that it requires solving multiple stochastic optimization problems at each forward pass. For the case of static combinations, we further consider weighing individual forecasts based on historical performance, which comes at a

lower overall cost, adapting the well-known weighting scheme of Bates and Granger (1969) in a decision-focused framework.

The process is described as follows. First, for each expert s and each observation i, we solve the contextual stochastic optimization problem (2). Next, for each expert s, we estimate the average in-sample decision regret, denoted by $\hat{v}_s$. Finally, the static combination weights are given by

$$\lambda_s = \frac{\frac{1}{\widehat{v}_s}}{\sum_{s \in [S]} \frac{1}{\widehat{v}_s}}, \tag{8}$$

i.e., by inversely weighing the in-sample performance of each expert.

3. Numerical Experiments

This section presents the numerical experiments. Section 3.1 describes the experimental setup. Section 3.2 provides an illustrative example with synthetic data. Section 3.3 presents the results of the trading case study. Section 3.4 presents the results of the grid scheduling case study. The code to recreate the experiments is made available on GitHub.[1]

3.1. Experimental Setup

3.1.1. Component Forecasts

For individual component forecasts, we utilize non-parametric machine learning models, which learn a function that assigns weights $\boldsymbol{\omega}(\mathbf{x}) \in \Sigma_N$ to training observations y_i based on a realization of contextual information $\mathbf{x}_0$. Then, the original contextual stochastic optimization problem (2) can be approximated by solving a weighted sample average approximation problem, given by

$$\min_{\mathbf{z} \in \mathcal{Z}} \sum_{i \in [N]} \omega_i(\mathbf{x}_0) c(\mathbf{z}; y_i). \tag{9}$$

As Y has finite support, we can count the number of times ξ_k appears in $\mathcal{D}$ and aggregate the respective weights $\omega_i(\mathbf{x}_0)$ to equivalently write (9) with a probability vector that weighs each support location. That is, the estimated probability of $Y = \xi_k$ conditioned on $\mathbf{x}_0$ is given by $\sum_{i \in [N]} \mathbb{I}(y_i = \xi_k)\, \omega_i(\mathbf{x}_0)$. The following models are used as experts:

[1] `https://github.com/akylasstrat/df-forecast-comb`

1. **Nearest Neighbors**: We consider weights learned with a k Nearest Neighbors (kNN) model (Hastie et al., 2001, Chap. 13), given by

$$\omega_i(\mathbf{x}_0) = \frac{1}{k}\mathbb{I}\left(\mathbf{x}_i \text{ is a neighbor of } \mathbf{x}_0\right),$$

   where hyperparameter k indicates the number of data points closest to $\mathbf{x}_0$.
2. **Decision Trees**: We consider weights learned with decision tree models, using the CART algorithm Breiman et al. (1984). Let $\tau : \mathcal{X} \to \{1, \ldots, L\}$ be a map that corresponds to a disjoint partition of $\mathcal{X}$ into L tree leaves and $\tau(\mathbf{x}_0)$ is the identity of the leaf that $\mathbf{x}_0$ falls into. The respective weights are given by

$$\omega_i(\mathbf{x}_0) = \frac{\mathbb{I}\left(\tau(\mathbf{x}_i) = \tau(\mathbf{x}_0)\right)}{\sum_{i=1}^{N}\mathbb{I}\left(\tau(\mathbf{x}_i) = \tau(\mathbf{x}_0)\right)}.$$

3. **Tree-based Ensembles**: We consider weights learned with an ensemble of decision trees, grown with the Random Forest (RF) algorithm (Breiman, 2001). Consider an ensemble of T decision trees $\{\tau_1, \ldots, \tau_T\}$, where $\tau_j : \mathcal{X} \to \{1, \ldots, L_j\}$ is a map that corresponds to a disjoint partition of $\mathcal{X}$ into L_j tree leaves and $\tau_j(\mathbf{x}_0)$ is the leaf identity. The respective weights are given by

$$\omega_i(\mathbf{x}_0) = \frac{1}{T}\sum_{j=1}^{T}\frac{\mathbb{I}\left(\tau_j(\mathbf{x}_i) = \tau_j(\mathbf{x}_0)\right)}{\sum_{i'=1}^{N}\mathbb{I}\left(\tau_j(\mathbf{x}_{i'}) = \tau_j(\mathbf{x}_0)\right)}.$$

3.1.2. Combination Methods

The following methods of estimating the weights of a linear pool are compared:

1. **Ordinary Linear Pool** (`OLP`): We assign uniform weights to all the experts (Stone, 1961), a robust approach that performs well in practice.
2. **CRPS learning** (`CRPSL`): We learn combination weights by minimizing the CRPS of the combined forecast, similarly to van der Meer et al. (2024); Berrisch and Ziel (2023).
3. **Decision-focused learning** (`DFL`$-\gamma$): We learn combination weights by solving the decision-focused combination problem (7) using the gradient-based approach outlined in Section 2.3.1 for different values of design parameter γ.

4. **Performance-based inverse weighting** (`invW`): Each expert is weighted inversely to their historical performance in terms of decision regret, as proposed in (8).

In the case of the conditional forecast combination using additional features, we consider two classes of models $\mathcal{F}$ for the function f that adapts the combination weights, namely linear regression (`LR`) and neural network-based (`NN`) models (`LR` is a special case of `NN` with a single layer). Conditional forecast combination is only tested for the case of `CRPSL` and `DFL`$-\gamma$. For clarity, `DFL-LR`$-\gamma$ is learned by solving the decision-focused combination problem (7) where f is a linear model, and so forth.

All variants of `DFL`$-\gamma$ and `CRPSL` are optimized with a stochastic gradient-based approach, using the Adam algorithm (Kingma and Ba, 2014) with a batch size of 500, a learning rate of 0.001. Modeling is implemented in `PyTorch` (Paszke et al., 2019) and the differentiable optimization layers are implemented with the `cvxpylayers` (Agrawal et al., 2019b) package. For the static combination methods (`CRPSL`, `DFL`$-\gamma$) we iterate till convergence. For the conditional combination methods (`CRPSL-LR`, `CRPSL-NN`, `DFL-LR`$-\gamma$, `DFL-NN`$-\gamma$) we evaluate performance using a subset of the training set as a validation set. We set the total number of epochs high and train using early stopping, i.e., we stop training if the validation performance fails to improve for 25 epochs. For the `NN`-based conditional combinations, we consider a feedforward network with 3 hidden layers and 20 nodes per layer.

3.1.3. Performance Evaluation

We evaluate the efficacy of the different combination methods w.r.t. decision and forecast quality, using a test set of N^{test} observations. Decision quality is assessed by estimating the average out-of-sample regret

$$\frac{1}{N^{\text{test}}} \sum_{i \in [N^{\text{test}}]} c(\mathbf{z}_i^{\texttt{comb}}; y_i) - c(\mathbf{z}_i^*; y_i),$$

where $\mathbf{z}_i^{\texttt{comb}}$ is the decision obtained by each forecast combination method and $\mathbf{z}_i^*$ is the perfect foresight decision. Forecast quality is measured by approximating the average CRPS over the test set using its quantile decomposition (5) for a grid of $\{0.01, 0.02, \ldots, 0.99\}$ of quantiles.

3.2. Synthetic Data Example

Problem Description. We first showcase the efficacy of the proposed combination methods in an illustrative example with synthetic data. We consider

an uncertain variable Y associated with features X_0, X_1, X_2, X_3, which all follow a standard normal distribution, through

$$Y = X_0 + \alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X_3 \cdot \mathbb{I}(X_3 \leq \beta) + \epsilon, \tag{10}$$

where ϵ denotes an error term and $\alpha_1 = \alpha_2 = 1.2$, $\alpha_3 = 4.5$, and $\beta = -1.3$ are known parameters. Two experts use the available features to generate probabilistic forecasts. Features X_1 and X_2 denote information measured only by expert 1, X_3 denotes information measured only by expert 2, while X_0 denotes publicly available information. The two experts issue the following conditional normal distributions:

$$p_1 = \mathcal{N}(X_0 + \alpha_1 X_1 + \alpha_2 X_2, 0.5),$$
$$p_2 = \mathcal{N}(X_0 + \alpha_3 X_3 \cdot \mathbb{I}(X_3 \leq \beta), 0.5).$$

The decision-maker, using these two forecasts as inputs, aims to minimize the newsvendor loss (Ban and Rudin, 2019)

$$c(z; Y) = \max\big(\frac{\tau}{1-\tau}(Y - z), (z - Y)\big),$$

where τ is the optimal quantile.

Assume that the optimal quantile is set at $\tau = 0.20$, which targets the left tail of the distribution. By design, expert 1 does a better job, on average, modeling the conditional distribution of Y, while expert 2 does a relatively better job at modeling the left tail of the distribution (i.e., lower quantiles). Hence, we expect that assigning a higher weight to expert 2 in the linear pool will lead to lower expected decision costs. We validate this assumption by sampling 10 000 observations with 50/50 training/test split and evaluating the different combination methods.

Results. Table 1 presents the experiment results, alongside the combination weights $\boldsymbol{\lambda}$ learned by each method. Concerning the component forecasts, expert 1 leads to higher forecast quality (lower CRPS), and expert 2 leads to higher decision quality (lower cost), as expected. All combination methods significantly improve upon the component forecasts in both evaluation metrics, with `CRPSL` ranking first in terms of forecast quality and `DFL`-0 ranking first in terms of decision quality, with `OLP` being second best in both evaluation metrics.

Interestingly, `CRPSL` is the worst-performing combination method in terms of decision quality, while `DFL`-0 is the worst-performing one in terms of forecast quality, which is attributed to the learned combination weights. As

Table 1: Results for synthetic data example. Bold font indicates the best-performing combination method.

	λ_1	λ_2	Regret	CRPS
`Expert 1`	1	0	66.468	7.905
`Expert 2`	0	1	60.300	11.401
`OLP`	0.500	0.500	22.839	5.954
`CRPSL`	0.636	0.364	23.504	**5.752**
`DFL−0`	0.445	0.555	**22.549**	6.191
`invW`	0.492	0.508	22.841	5.983

shown from the quantile decomposition in (5), CRPS can be approximated by the quantile score, hence, `CRPSL` weighs experts' performance equally across the whole distribution. As expert 1 does a good job modeling a larger part of the distribution, `CRPSL` assigns her a larger weight, with $\lambda_1 = 0.636$. Conversely, `DFL−0` only considers the quantile loss at the 20-th quantile, i.e., the newsvendor loss for $\tau = 0.20$; as expert 2 does a better job modeling the left tail of the uncertainty distribution, `DFL−0` assigns her higher weight, with $\lambda_2 = 0.555$.

3.3. Solar Production Forecasting and Trading in Electricity Markets

Problem Description. We consider a renewable producer participating in a competitive electricity market as a price-taker, a problem that has gathered a lot of attention in recent years (Morales et al., 2013). Before market closure, the producer submits an energy offer for each clearing period of the day-ahead market. During real-time operation, the system operator activates balancing reserves to maintain the demand-supply equilibrium and stabilize the system frequency. Based on real-time production, the producer buys back (sells) the amount of energy shortage (surplus) to balance her position.

We consider the producer's cost function is given by

$$c(z;Y) = (1-\rho)\max\big(\frac{\tau}{1-\tau}(Y-z),(z-Y)\big) + \rho(Y-z)^2, \qquad (12)$$

i.e., a combination of the newsvendor and mean squared error loss, where ρ is a design parameter that controls for the degree of risk aversion (Stratigakos et al., 2022). Further, Y is normalized by the nominal plant capacity and the feasible decision set is $\mathcal{Z} = \{0 \leq z \leq 1\}$.

Data and Component Forecasts. We use approximately 2.5 years of hourly production data for 3 solar plants located in Australia, provided by the

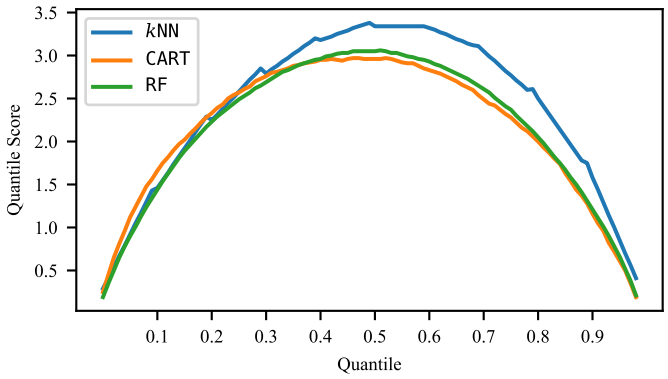

Figure 1: **Solar production forecasting:** Quantile score evaluated at an evenly spaced grid of quantiles $\{0.01, 0.02, \ldots, 0.99\}$.

Global Energy Forecasting Competition 2014 (GEFCom2014) (Hong et al., 2016). The contextual information comprises weather forecasts from a Numerical Weather Prediction (NWP) model obtained in a day-ahead horizon, including precipitation, solar radiation, and temperature — see Hong et al. (2016) for details. We use one year of data to train individual forecast experts, one year of data to estimate the combination weights, and seven months as a test set. The NWP forecasts are issued at 6:00 of the day $d-1$ and cover every hour of the day d, i.e., a forecast horizon of 18-42 hours ahead, which is a standard setting when participating in day-ahead electricity markets. We generate component forecasts using the kNN, CART, and RF models described in Section 3.1.1, using a different subset of features for each expert. We perform a grid search with 5-fold cross-validation for hyperparameter tuning and re-train using the whole training data set once the hyperparameters are selected. We further consider an additional set of 4 features that model diurnal patterns, using Fourier terms that model daily and hourly seasonality patterns. We assume these are available only to the decision-maker, and use it to evaluate conditional forecast combinations.

Results. Fig. 1 shows the probabilistic forecast performance of each expert. Although `RF` is the best-performing component forecast overall, as confirmed by its CRPS in Table 2, Fig. 1 shows that performance varies across the distribution regions. Namely, `CART` performs better for the middle part of the distribution, while k`NN` performs well for its lower tail and considerably worse for its right tail.

We evaluate decision performance for $\rho = 0.2$, $\tau = \{0.1, 0.2, \ldots, 0.9\}$,

Table 2: Learned combination weights $\boldsymbol{\lambda}$, average hourly decision regret, and CRPS for a single solar plant ($\tau = 0.2$, $\rho = 0.2$). Bold font indicates the best-performing method for static and conditional combinations, respectively. Underlined bold font indicates the best-performing method overall.

		λ_1	λ_2	λ_3	Regret	CRPS
Static combinations	kNN	1	0	0	2.663	5.412
	CART	0	1	0	2.653	5.075
	RF	0	0	1	2.578	5.049
	OLP	0.333	0.333	0.333	2.282	4.596
	invW	0.322	0.338	0.341	2.279	4.590
	CRPSL	0.033	0.534	0.433	2.263	**4.524**
	DFL-0	0.170	0.409	0.421	2.247	4.534
	DFL-0.1	0.169	0.414	0.417	2.246	4.532
	DFL-1	0.157	0.419	0.424	**2.244**	4.530
Conditional combinations	CRPSL-LR	-	-	-	2.256	**4.426**
	CRPSL-NN	-	-	-	2.249	4.436
	DFL-LR-0	-	-	-	2.240	4.530
	DFL-NN-0	-	-	-	2.253	4.568
	DFL-LR-0.1	-	-	-	2.239	4.522
	DFL-NN-0.1	-	-	-	2.236	4.502
	DFL-LR-1	-	-	-	2.235	4.448
	DFL-NN-1	-	-	-	**2.234**	4.439

and each solar plant in the data set. Table 2 shows the average decision cost and CRPS for all combination methods for a single solar farm and $\rho = 0.2, \tau = 0.2$, alongside learned combination weights $\boldsymbol{\lambda}$ (for static combinations). Results for the different values of τ and the rest of the solar farms are similar and provided as supplementary material.

We first examine static forecast combinations (indicated with white background color in Table 2). Overall, all combination methods significantly improve upon component forecasts both in terms of decision and forecast quality. RF is the best-performing component forecast in both metrics. CRPSL leads to the highest forecast quality (lowest CRPS), which is approximately 10.3% better than RF, while DFL-1 leads to the highest decision quality (lowest cost), which is approximately 5.2% better than RF. Indeed, the results of Table 2 highlight that higher forecast quality does not always translate into better decisions, as all DFL$-\gamma$ variants lead to lower cost and higher CRPS compared to CRPSL. Moreover, invW is worse than both DFL$-\gamma$ and CRPSL in both metrics, followed by OLP which is the worst-performing combination method. These results are attributed to the variability of learned combination weights $\boldsymbol{\lambda}$ across the static combination methods. We observe that CRPSL assigns a very small weight to kNN ($\lambda_1 = 0.033$), while the respective weight for CART is much larger ($\lambda_2 = 0.534$). Conversely, the weights of DFL$-\gamma$ appear to be more evenly distributed, with the respective weight of

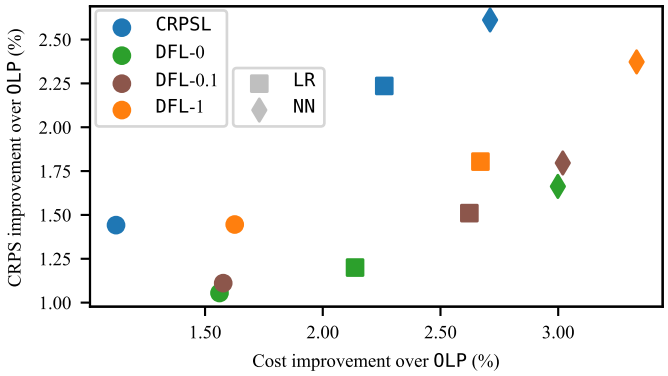

Figure 2: Cost improvement versus CRPS improvement against OLP, average value over the 3 solar plants and $\tau = \{0.1, 0.2, \ldots, .9\}$. The circle marker indicates static combinations. The square and diamond markers indicate conditional combinations using linear and neural network models, respectively. Values toward up and right are better.

kNN being closer to 0.20 and CART, RF having a weights closer to 0.40. Further, as γ increases, the respective weights of the DFL method move closer to the weights learned by CRPSL, as expected.

Next, we examine performance for conditional forecast combinations, highlighted with gray color in Table 2. Overall, conditional forecast combinations improve upon static combinations, both in terms of decision and forecast quality, which is somewhat expected as the former utilizes additional feature data. The average improvement, however, is rather small (less than 1% in all cases). This is primarily attributed to the experiment design and the nature of the feature data, as the NWP variables already capture the diurnal effect through solar radiation forecasts. The differences across learning models are also small, with NN leading to slightly better performance in most cases. CRSPL-LR ranks first in terms of forecast quality, while DFL-NN-1 ranks first in terms of decision quality. Again, the key differentiating factor is the objective of the forecast combination method, i.e., whether the forecast combination minimizes CRPS or expected decision cost.

Finally, Fig. 2 presents the average improvement of CRPL, DFL $-\gamma$, and their conditional variations, over OLP across all the experiments, namely the three solar plants and the different values of optimal quantile τ. A key takeaway is that optimizing the weights of the linear pool consistently improves over OLP. Each variation of CRPSL always leads to the largest improvement in terms of CRPS, with a maximum of 2.61% while variations of DFL$-\gamma$

always lead to the largest improvement in decision regret reduction, with a maximum of 3.34%. Notably, optimizing for a combination between decision regret and CRPS in the decision-focused combination (7) consistently leads to great performance in both metrics, as `DFL`-1 is always on the efficient frontier for each variation. This result corroborates other works (Bertsimas and Skali Lami, 2023) which posit that optimizing a combination of decision cost and forecast accuracy leads to the best trade-off.

3.4. Wind Production Forecasting and Grid Scheduling

Problem Description. We further consider a realistic grid scheduling problem under net demand uncertainty. A grid operator wants to schedule the production level of a set of G generators to meet an uncertain net demand Y (base electricity demand minus renewable production), which depends on the stochastic renewable production, in a look-ahead horizon, typically 18-42 hours ahead, while anticipating potential costs due to a real-time mismatch between demand and supply. This setting is common in power systems dominated by stochastic renewable energy sources (Morales et al., 2013). The grid scheduling problem is formulated as a two-stage stochastic optimization problem with fixed recourse given by

$$\min_{\mathbf{z},\mathbf{z}_k^u,\mathbf{z}_k^d} \quad \mathbf{c}^\top \mathbf{z} + \sum_{k\in[K]} p_k(\mathbf{c}^{u\top}\mathbf{z}_k^u - \mathbf{c}^{d\top}\mathbf{z}_k^d), \tag{13a}$$

$$\text{s.t.} \quad \mathbf{1}^\top(\mathbf{z} + \mathbf{z}_k^u - \mathbf{z}_k^d) = \xi_k, \qquad k \in [K], \tag{13b}$$

$$\mathbf{0} \le \mathbf{z} \le \overline{\mathbf{z}}, \tag{13c}$$

$$\mathbf{0} \le \mathbf{z}_k^u \le \min(\overline{\mathbf{z}}^u, \overline{\mathbf{z}} - \mathbf{z}), \qquad k \in [K], \tag{13d}$$

$$\mathbf{0} \le \mathbf{r}_k^d \le \min(\overline{\mathbf{z}}^d, \mathbf{z}), \qquad k \in [K]. \tag{13e}$$

where $\mathbf{z}$ is the vector of look-ahead (first-stage) dispatch decisions, $\{\mathbf{z}_k^u, \mathbf{z}_k^d\}$ is the set of recourse (second-stage) actions defined per each scenario k, $\overline{(\cdot)}$ indicates upper limits on decision variables, and $\mathbf{c}^d \le \mathbf{c} \le \mathbf{c}^u$ (inequality is applied elementwise) are non-negative cost vectors. The objective function (13a) minimizes the expected dispatch cost under net demand uncertainty considering real-time recourse actions, (13b) ensures the demand-supply balance under all possible realizations of uncertainty (modeled as discrete scenarios), and constraints (13c)-(13e) are the technical constraints of the generators, where the min operator is applied elementwise.

Data and Component Forecasts. We consider a scheduling problem where a constant base demand is coupled with a wind farm and two thermal generators are used to meet the resultant stochastic net demand. Thermal

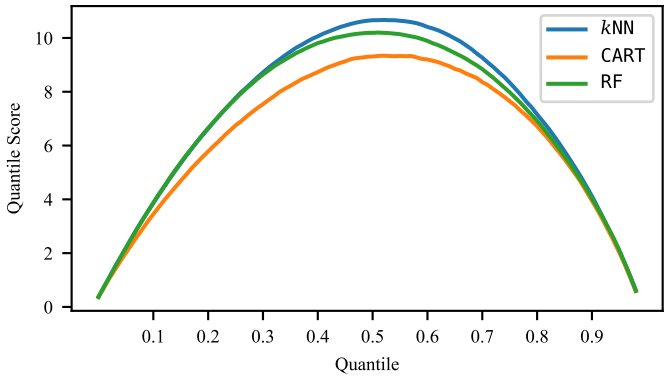

Figure 3: **Wind production forecasting:** Quantile score evaluated at an evenly spaced grid of quantiles $\{0.01, 0.02, \ldots, 0.99\}$.

generator data are obtained from the IEEE 14-bus system (Babaeinejadsarookolaee et al., 2019), with upward regulation costs $\mathbf{c}^u$ sampled from a range $[1.1 \cdot \mathbf{c}, 6 \cdot \mathbf{c}]$ and downward regulation costs $\mathbf{c}^d$ sampled from a range $[0.05 \cdot \mathbf{c}, 0.2 \cdot \mathbf{c}]$. We assume a constant demand of 100 MW, a wind farm of equal nominal capacity, and appropriately scale generator capacity data to the same aggregate capacity value. We use two years of hourly wind production data provided by GEFCom2014 (Hong et al., 2016) and use one year of data to train expert models, six months to estimate combination weights, and six months for out-of-sample testing. As in the previous case study, the contextual information comprises weather forecasts from an NWP model obtained in a day-ahead horizon, namely, wind speed and wind direction forecasts (both at 10m and 100m hub height). We generate component forecasts using the kNN, CART, and RF models as experts— see Section 3.1.1, and the wind production data from zone 1 as the target. All experts utilize the same weather variables as input features. To create variability in their outputs, each expert utilizes NWP features from different locations of the GEFCom2014 data set. As before, we perform a grid search with 5-fold cross-validation for hyperparameter tuning and re-train using the whole training data set once the hyperparameters are selected.

Results. Fig. 3 plots the forecast quality of component forecasts, with `CART` consistently outperforming k`NN`, `RF` across the distribution. This is attributed to the NWP features utilized by `CART` being more relevant to the target wind farm location. This presents a different setting compared to the previous case study in Section 3.3 where experts' performance varied across

Table 3: Learned combination weights $\boldsymbol{\lambda}$, average hourly dispatch regret, and CRPS for a grid scheduling problem with 100MW wind plant and 2 thermal generators. Bold font indicates the best-performing combination method.

	λ_1	λ_2	λ_3	Regret	CRPS
kNN	1	0	0	570.498	7.878
CART	0	1	0	507.336	7.004
RF	0	0	1	575.322	7.646
OLP	0.333	0.333	0.333	468.453	6.768
invW	0.331	0.340	0.329	467.890	6.759
CRPSL	0.276	0.463	0.261	458.047	6.638
DFL-0.001	0.258	0.498	0.243	**455.490**	**6.617**
DFL-0.01	0.272	0.507	0.221	455.543	6.620

the distribution.

Table 3 presents the average out-of-sample decision regret and CRPS over the test set, alongside the learned combination weights $\boldsymbol{\lambda}$ for static combination methods. To estimate the decision regret for the i-th test observation, which marks an hour of operation, we first solve the stochastic scheduling problem (13) to find the look-ahead dispatch actions. Once uncertainty y_i is realized, we fix the look-ahead decision variables and solve the scheduling problem (13) again to find the least-cost recourse actions that maintain the demand-supply balance during real-time operation. The final cost of the i-th observation is then estimated as the sum of the look-ahead and regulation costs, minus the perfect foresight cost (clearly, perfect foresight requires no real-time regulation).

Table 3 shows that all combination methods improve upon the component forecasts w.r.t. both forecast and decision quality. Note that to improve the convergence of the gradient-based algorithm, we included a small regularization parameter γ in all DFL variants. OLP and invW perform similarly and significantly improve upon the component forecasts, while CRPL and DFL$-\gamma$ further improve upon OLP. Specifically, DFL-0.001 is the best-performing combination method in both metrics, leading to approximately 2.8% lower regret and 2.2% lower CRPS compared to OLP. Interestingly, both DFL-0.001 and DFL-0.01 lead to lower CRPS compared to CRPL which means that, for this experiment, higher forecast quality coincides with better decisions. From Table 3, we observe that CRPL and DFL$-\gamma$ lead to similar combination weights $\boldsymbol{\lambda}$, with DFL$-\gamma$ assigning higher weight to CART and lower weight to RF. This can be contrasted to the trading case study— see Table 2, where variations in the learned weights $\boldsymbol{\lambda}$ are more pronounced.

4. Conclusions

This work proposed a novel approach for combining probabilistic forecasts that accounts for downstream decision costs. We developed decision-focused linear pooling, where combination weights are optimized to minimize an expected cost function of a stochastic optimization problem. We also extended the proposed method to the case where the combination weights adapt to the realization of additional features available to the decision-maker. A comprehensive evaluation was conducted considering synthetic and real-world test cases, examining integral problems associated with variable renewable energy sources integration in power and energy systems. The results highlighted the efficacy of the proposed decision-focused combination approach which improved upon ordinary linear pooling, with uniform weights, in decision and forecast quality. In particular, the average reduction in decision regret compared to ordinary linear pooling was approximately 2% in a trading problem with stochastic solar production and 2.8% in a grid scheduling problem with stochastic wind production. Importantly, minimizing a combination of decision regret and CRPS consistently led to a better trade-off between decision and forecast quality. Overall, this work highlighted the benefits of embedding the downstream objective when combining forecasts, which allowed us to maintain a simple setting (linear pooling), while significantly improving decision performance. Future work can focus on decision-focused quantile averaging and extending the proposed approach to nonlinear pooling.

Acknowledgements

The work of A. Stratigakos was supported in part by the Leverhulme Trust International Professorship, held by Professor M. O'Malley.

The work of J. M. Morales and S. Pineda was supported in part by the European Research Council (ERC) funded under the Horizon 2020 Framework Program (Grant No 755705) and in part by the Spanish Ministry of Science and Innovation (AEI/10.13039/501100011033) through project PID2020-115460GB-I00.

References

Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., Kolter, J.Z., 2019a. Differentiable convex optimization layers. Advances in neural information processing systems 32.

Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., Kolter, Z., 2019b. Differentiable convex optimization layers, in: Advances in Neural Information Processing Systems.

Agrawal, A., Barratt, S., Boyd, S., 2021. Learning convex optimization models. IEEE/CAA Journal of Automatica Sinica 8, 1355–1364.

Amos, B., Kolter, J.Z., 2017. Optnet: Differentiable optimization as a layer in neural networks, in: International Conference on Machine Learning, PMLR. pp. 136–145.

Babaeinejadsarookolaee, S., Birchfield, A., Christie, R.D., Coffrin, C., DeMarco, C., Diao, R., Ferris, M., Fliscounakis, S., Greene, S., Huang, R., et al., 2019. The power grid library for benchmarking AC optimal power flow algorithms. arXiv:1908.02788 .

Ban, G.Y., Rudin, C., 2019. The big data newsvendor: Practical insights from machine learning. Operations Research 67, 90–108.

Bates, J.M., Granger, C.W., 1969. The combination of forecasts. Journal of the operational research society 20, 451–468.

Berrisch, J., Ziel, F., 2023. Crps learning. Journal of Econometrics 237, 105221.

Bertsimas, D., Kallus, N., 2020. From predictive to prescriptive analytics. Management Science 66, 1025–1044.

Bertsimas, D., Skali Lami, O., 2023. Holistic prescriptive analytics for continuous and constrained optimization problems. INFORMS Journal on Optimization 5, 155–171.

Birge, J.R., Louveaux, F., 2011. Introduction to stochastic programming. Springer Science & Business Media.

Breiman, L., 2001. Random forests. Machine learning 45, 5–32.

Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A., 1984. Classification and regression trees. CRC press.

Chen, X., Yang, Y., Liu, Y., Wu, L., 2022. Feature-driven economic improvement for network-constrained unit commitment: A closed-loop predict-and-optimize framework. IEEE Transactions on Power Systems 37, 3104–3118. doi:10.1109/TPWRS.2021.3128485.

Diebold, F.X., Gunther, T.A., Tay, A.S., 1998. Evaluating density forecasts with applications to financial risk management. International Economic Review 39, 863–883. URL: http://www.jstor.org/stable/2527342.

Donti, P.L., Amos, B., Kolter, J.Z., 2017. Task-based end-to-end model learning in stochastic optimization, in: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 5490–5500.

Elmachtoub, A.N., Grigas, P., 2022. Smart "predict, then optimize". Management Science 68, 9–26.

Gneiting, T., Balabdaoui, F., Raftery, A.E., 2007. Probabilistic forecasts, calibration and sharpness. Journal of the Royal Statistical Society Series B: Statistical Methodology 69, 243–268.

Gneiting, T., Raftery, A.E., 2007. Strictly proper scoring rules, prediction, and estimation. Journal of the American statistical Association 102, 359–378.

Gneiting, T., Ranjan, R., 2011. Comparing density forecasts using threshold- and quantile-weighted scoring rules. Journal of Business & Economic Statistics 29, 411–422.

Gneiting, T., Ranjan, R., 2013. Combining predictive distributions .

Granger, C.W., Ramanathan, R., 1984. Improved methods of combining forecasts. Journal of forecasting 3, 197–204.

Grigas, P., Qi, M., Shen, M., 2021. Integrated conditional estimation-optimization. URL: https://arxiv.org/abs/2110.12351, doi:10.48550/ARXIV.2110.12351.

Hastie, T., Tibshirani, R., Friedman, J., 2001. The elements of statistical learning. springer series in statistics. New York, NY, USA .

Hong, T., Pinson, P., Fan, S., Zareipour, H., Troccoli, A., Hyndman, R.J., 2016. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. International Journal of Forecasting 32, 896–913. doi:10.1016/j.ijforecast.2016.02.001.

Kallus, N., Mao, X., 2022. Stochastic optimization forests. Management Science .

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

Krannichfeldt, L.V., Wang, Y., Zufferey, T., Hug, G., 2022. Online ensemble approach for probabilistic wind power forecasting. IEEE Transactions on Sustainable Energy 13, 1221–1233. doi:10.1109/TSTE.2021.3124228.

Lichtendahl Jr, K.C., Grushka-Cockayne, Y., Winkler, R.L., 2013. Is it better to average probabilities or quantiles? Management Science 59, 1594–1611.

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 arXiv:2307.13565 .

van der Meer, D., Pinson, P., Camal, S., Kariniotakis, G., 2024. CRPS-based online learning for nonlinear probabilistic forecast combination. International Journal of Forecasting URL: https://www.sciencedirect.com/science/article/pii/S0169207023001371, doi:10.1016/j.ijforecast.2023.12.005.

Morales, J.M., Conejo, A.J., Madsen, H., Pinson, P., Zugno, M., 2013. Integrating renewables in electricity markets: operational problems. volume 205. Springer Science & Business Media.

Morales, J.M., Muñoz, M., Pineda, S., 2023. Prescribing net demand for two-stage electricity generation scheduling. Operations Research Perspectives 10, 100268.

Motley, A., 2023. CAISO: Advances in the use of wind and solar forecasting. URL: https://www.esig.energy/event/g-pst-esig-webinar-series-advances-in-the-use-of-wind-and-solar-forecasting/.

Papayiannis, G.I., Yannacopoulos, A.N., 2018. A learning algorithm for source aggregation. Mathematical Methods in the Applied Sciences 41, 1033–1039.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. Pytorch: An imperative style, high-performance deep learning library,

in: Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Raftery, A.E., Gneiting, T., Balabdaoui, F., Polakowski, M., 2005. Using bayesian model averaging to calibrate forecast ensembles. Monthly weather review 133, 1155–1174.

Stone, M., 1961. The linear opinion pool. Ann. Math. Statist 32, 1339–1342.

Stratigakos, A., Camal, S., Michiorri, A., Kariniotakis, G., 2022. Prescriptive trees for integrated forecasting and optimization applied in trading of renewable energy. IEEE Transactions on Power Systems 37, 4696–4708. doi:10.1109/TPWRS.2022.3152667.

Thorey, J., Chaussin, C., Mallet, V., 2018. Ensemble forecast of photovoltaic power with online crps learning. International Journal of Forecasting 34, 762–773.

Thorey, J., Mallet, V., Baudin, P., 2017. Online learning with the continuous ranked probability score for ensemble forecasting. Quarterly Journal of the Royal Meteorological Society 143, 521–529.

Wang, X., Hyndman, R.J., Li, F., Kang, Y., 2023. Forecast combinations: An over 50-year review. International Journal of Forecasting 39, 1518–1547. URL: https://www.sciencedirect.com/science/article/pii/S0169207022001480, doi:https://doi.org/10.1016/j.ijforecast.2022.11.005.

Winkler, R.L., 1968. The consensus of subjective probability distributions. Management science 15, B–61.