

Towards Improving Unit Commitment Economics: An Add-On Tailor for Renewable Energy and Reserve Predictions

Xianbang Chen, *Student Member, IEEE*, Yikui Liu, *Member, IEEE*, Lei Wu, *Fellow, IEEE*

Abstract—Generally, day-ahead unit commitment (UC) is conducted in a predict-then-optimize process: it starts by predicting the renewable energy source (RES) availability and system reserve requirements; given the predictions, the UC model is then optimized to determine the economic operation plans. In fact, predictions within the process are *raw*. In other words, if the predictions are further tailored to assist UC in making the economic operation plans against realizations of the RES and reserve requirements, UC economics will benefit significantly. To this end, this paper presents a cost-oriented tailor of RES-and-reserve predictions for UC, deployed as an add-on to the predict-then-optimize process. The RES-and-reserve tailor is trained by solving a bi-level mixed-integer programming model: the upper level trains the tailor based on its induced operating cost; the lower level, given tailored predictions, mimics the system operation process and feeds the induced operating cost back to the upper level; finally, the upper level evaluates the training quality according to the fed-back cost. Through this training, the tailor learns to customize the raw predictions into cost-oriented predictions. Moreover, the tailor can be embedded into the existing predict-then-optimize process as an add-on, improving the UC economics. Lastly, the presented method is compared to traditional, binary-relaxing, neural network-based, stochastic, and robust methods.

Index Terms—Renewable energy, unit commitment, prescriptive analytics, bi-level mixed-integer programming.

NOMENCLATURE

Sets and Indexes

\mathcal{B}/b	Set/index of branches.
\mathcal{E}/e	Set/index of iterations. The iteration limitation is E .
\mathcal{I}/i	Set/index of non-RES units, i.e., $\mathcal{I} = \mathcal{I}^{\text{ns}} \cup \mathcal{I}^{\text{qs}}$, where \mathcal{I}^{ns} and \mathcal{I}^{qs} are sets of non-quick-start and quick-start non-RESs.
\mathcal{J}/j	Set/index of RES units.
\mathcal{K}/k	Set/index of piecewise linear segments of generation curve.

Manuscript received 4 October 2023; revised 4 February 2024 and 12 May 2024; accepted 28 June 2024. Date of publication DD MMMM YYYY; date of current version DD MMMM YYYY. This work was supported in part by the Army Combat Capabilities Development Command (CCDC) project on Resiliency of Energy Resources and Supply Chain for the Industrial Base and the PSEG Foundation gift. Paper no. TSTE-xxxxx-xxxx. (*Corresponding author: Yikui Liu.*)

X. Chen and L. Wu are with the ECE Department, Stevens Institute of Technology, Hoboken, NJ, 07030 USA. (e-mail: xchen130@stevens.edu and lei.wu@stevens.edu).

Y. Liu is with the Electrical Engineering College, Sichuan University, Chengdu, 610017 China. (e-mail: yikuiliu89@outlook.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier xx.xxxx/TSTE.xxxx.xxxxxxx

S/s	Set/index of training samples.
$\mathcal{T}/t, t'$	Set/indexes of hours. The total number of hours is T .
$\mathcal{T}_i^{\text{su/sd}}$	Set defined as $\{T_i^{\text{su}}, \dots, T\}/\{T_i^{\text{sd}}, \dots, T\}$.
$ \cdot $	The cardinality of a set.

Decision Variables

D_{it}	Shutdown status of unit i at hour t .
I_{it}	On-off status of unit i at hour t .
O_{it}	Non-spinning reserve (NR) commitment of unit i at hour t .
P_{it}	Generation schedule of unit i at hour t .
P_{ik}^{sg}	Generation schedule of unit i in segment k at hour t .
$R_{it}^{\text{sr/nr}}$	Spinning reserve (SR)/NR schedule of unit i at hour t .
S	Slack variable.
U_{it}	Startup status of unit i at hour t .
\mathcal{W}/\mathcal{R}	Tailor of RES power/reserve requirements.
W_{jt}	Generation schedule of RES j at hour t .
\mathbf{x}, \mathbf{y}	Vectors of decision variables in the UC model, including binaries $\mathbf{x} = \{U, I, O, D\}$ and continuous $\mathbf{y} = \{P, P^{\text{sg}}, R^{\text{sr}}, R^{\text{nr}}, W\}$.
$\bar{\mathbf{x}}^{\text{ev}}$	Enumerated solutions to binary variables.
\mathbf{z}	Vector of decision variables in the re-dispatch (RD) model, including binaries $\{U^{\text{RD}}, I^{\text{RD}}, I^{\text{RD,qs}}, D^{\text{RD}}\}$ and continuous $\{P^{\text{RD}}, P^{\text{RD,sg}}, W^{\text{RD}}, S\}$.
.dv/gv	Duplicated/generated variables.
.RD	Variables of RD problem.
.*	Indicating the optimal solution to a variable.

Parameters

B_b	Transmission capacity of branch b .
C_{ik}^{sg}	Generation price of unit i in segment k .
$C_i^{\text{su/nl}}$	Startup/no-load price of unit i .
$C_{gs/lr/sr}$	Penalty price of slack variables.
$\mathcal{P}^{\text{w/r}}$	Feasible region of predictor $\mathcal{W}(\cdot)/\mathcal{R}(\cdot)$.
$P_i^{\text{M/m}}$	Maximum/minimum generation limit of unit i .
\bar{P}_{ik}^{sg}	Power limit of unit i at segment k .
$\bar{R}_i^{\text{sr/nr}}$	SR/NR limit of unit i .
$\hat{R}_t^{\text{sr/nr}}$	Predicted SR/NR requirement at hour t , which together form the vector $\hat{\mathbf{r}}$.
$R_i^{\text{su/sd}}$	Startup/shutdown ramping capacity of unit i .
$R_i^{\uparrow/\downarrow}$	Upward/downward ramping capacity of unit i .
$T_i^{\text{su/sd}}$	Minimum on/off time requirement of unit i .
\tilde{W}/\tilde{W}_{jt}	Predicted/actual available power of RES j at hour t , forming the vector $\tilde{\mathbf{w}}/\tilde{\mathbf{w}}$.
$\hat{\mathbf{w}}/\hat{\mathbf{r}}$	Raw prediction of RES/reserve requirements.
$\hat{\mathbf{w}}^\diamond/\hat{\mathbf{r}}^\diamond$	Tailored counterpart of raw prediction $\hat{\mathbf{w}}/\hat{\mathbf{r}}$.

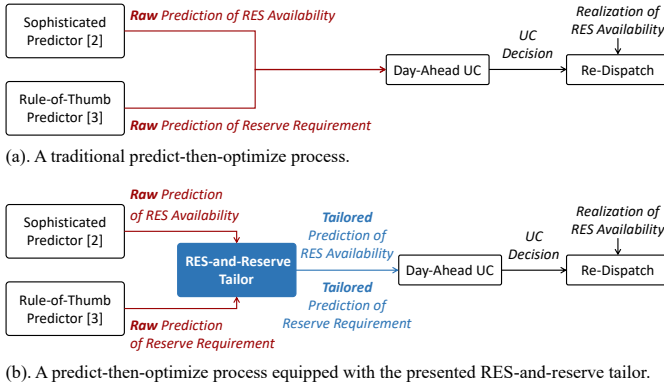


Fig. 1. Predict-then-optimize processes with and without the add-on tailor.

I. INTRODUCTION

A. Problem Statement

THIS paper focuses on the unit commitment (UC) economics of power systems integrated with renewable energy sources (RES), where system operators are the top administrators [1]. The daily UC is generally executed in a predict-then-optimize process, shown in Fig. 1(a) and summarized as the following three steps:

- *Step 1:* The RES availability [2], as well as the system spinning reserve (SR) and non-spinning reserve (NR) requirements [3], are predicted in the day-ahead stage. These predictions are generated for serving various downstream applications, including but not limited to UC. Thus, they are regarded as *raw predictions*¹;
- *Step 2:* Using the raw predictions as inputs, system operators solve a deterministic UC problem [4] to determine day-ahead operation plans involving startup/shutdown schedules and generation baselines;
- *Step 3:* To evaluate the economics of a UC plan, a re-dispatch (RD) problem [5], which is built upon the UC plan, is solved to calculate the eventual *balancing cost* against the actual RES realizations as well as SR and NR deployments.

With respect to the day-ahead predictions and the afterward RES realizations, the startup and no-load costs from day-ahead UC and the balancing costs from RD constitute the *actual operating cost*. This is referred to as *UC economics* in this paper. Indeed, a UC plan with superior economics can enable a low actual operating cost.

Although the current predict-then-optimize process seems plausible, its UC economics actually suffers from the raw predictions for at least two reasons: *i)* the RES predictors are generally trained by statistical accuracy criteria, such as mean absolute percentage error (MAPE). However, due to the inevitable prediction errors and the inherent nonlinearity of the UC-RD optimization process, *a statistically more accurate prediction may NOT necessarily induce a lower actual operating cost* (as revealed in Section II-C); and *ii)* the raw SR/NR requirements sized by traditional rule-of-thumb

¹Raw predictions are typically provided by sophisticated [2] or rule-of-thumb [3] predictors. Some systems make the raw predictions publicly accessible, enabling various entities to use these predictions for their specific applications.

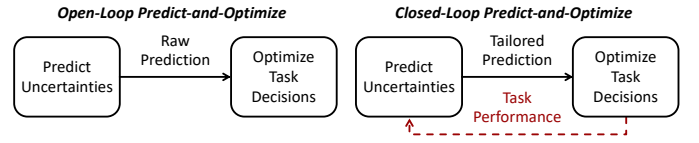


Fig. 2. The ideas of open-loop and closed-loop predict-and-optimize.

predictors could be overly redundant or insufficient, further worsening the UC economics.

Considering that the current raw predictors have been sophisticatedly developed and are utilized across various downstream applications, this paper targets to explore the following open question: *How to leverage the predictors currently used by the existing predict-then-optimize practice and tailor their output, i.e., raw predictions, to improve the UC economics?*

B. Literature Review

To improve the UC economics against the inevitable inaccuracy of raw predictions, one approach is to augment the deterministic UC to its two-stage stochastic programming (TSP) [6] or two-stage robust optimization (TRO) [7] counterpart. Differently, the *closed-loop predict-and-optimize* idea, proposed by [8] as an alternative approach, has gained growing attention. As illustrated in Fig. 2, this idea emphasizes that *predictors should support operators by prioritizing making good final decisions over pursuing immediate prediction accuracy, thus necessitating a closed-loop interaction between the prediction and optimization*. Indeed, this closed-loop idea has also been highlighted in the IEEE-CIS predict+optimize report [9] and the INFORMS special series entitled “Blending Predictive & Prescriptive Methods” [10]. Moreover, pioneering methodologies [11]–[13] and software packages [14]–[16] have also been developed.

The closed-loop idea has been increasingly applied to power systems. Table I lists 30 power system-related references, compiled to the best of the authors’ knowledge as of January 2024. These references are classified into “*integrated predict-and-optimize (IPO)*” and “*cost-oriented predict, then optimize (CPO)*.” Both IPO [17]–[21] and CPO [22]–[46] aim to improve the performance—generally in terms of the economics—of optimization tasks, while they adopt different philosophies:

- IPO streamlines the predict-then-optimize process as a single prescription step, by using machine learning (ML) models such as neural networks (NN) to prescribe decisions from features directly. IPO-based frameworks are well-suited for tasks whose decision feasibility is easily reachable. However, they generally cannot assure decision feasibility when facing complex constraints;
- CPO follows the two-step predict-then-optimize practice but enhances the prediction step to induce superior decisions in the optimization step. CPO-based frameworks are distinguished by their custom-trained predictors, which can yield *cost-oriented* predictions to enable superior decision-making in the optimization task. Although the tailored predictions may slightly compromise the statistical accuracy, they can boost the task performance greatly.

TABLE I
REVIEW OF RELEVANT APPLICATION REFERENCES (INVOLVING CLOSED-LOOP PREDICT-AND-OPTIMIZE IDEA) IN POWER SYSTEM FIELDS

Type	Ref.	Optimization Task		Feedback Path between Prediction and Optimization	Training Method	Mathematical Model of ERM Problem
		Task Content	Task Model			
IPO	[17]	RES trading	LP	Merge the prediction step and optimization step as a single prescription step	Heuristic	-
	[18]	RES trading	QP			
	[19]	Voltage regulation	SOCP		Gradient-based updating	-
	[20]	RES trading	LP		Optimally solve the ERM model to get optimally-trained predictors	Single-level LP Single-level MIP
	[21]	Reserve sizing	LP			
	[22]	UC	MIP	Use approximating error-and-cost function as loss function	Heuristic	-
	[23]	RES trading	LP		Gradient-based updating	-
	[24]	ED	LP			
	[25]	System planning	MIP	Identify scenarios that are helpful in reducing task cost	Depend on the applied ML algorithm	-
	[26]	ED	LP			
CPO	[27]	UC	MIP			
	[28]–[30]	ED	LP	Solve the task optimization model with given predictions and then feed its cost back to the predictor training phase	Gradient-based updating	-
	[31]–[33]	UC	MIP			
	[34]	Storage arbitrage	MIP			
	[35]	ED with storage	MIP		Contextual bandit-type	-
	[36]	ED	LP			
	[37]	Inertia management	NLP	Integrate predictor training and task optimization in the ERM model via explicit constraints	Optimally solve the ERM model to get optimally-trained predictors	Single-level NLP
	[38], [39]	UC	MIP			Single-level MIP
	[40]	RES trading	LP			Bi-level LP
	[41]–[46]	ED	LP			
This paper		UC & RD	MIP & MIP			Bi-level MIP

QP: Quadratic programming; SOCP: Second-order cone programming; NLP: Non-linear programming.

Early CPO frameworks [22]–[24] approximate the relationship between prediction error and task cost via analytical functions, which are then used as training loss functions to replace traditional accuracy-oriented loss functions. Furthermore, references [28]–[30] train a novel cost-aware load/RES predictor for economic dispatch (ED). Their training takes the ED cost induced by predictions as the loss, and their predictor parameters are updated by the derivative of ED cost w.r.t. predictor parameters. Such a cost-aware predictor can generate predictions tailored to reduce the ED cost. However, references [31]–[33] indicate that this approach is limited to linear programming (LP)-based operation tasks, primarily due to: *i*) the need for gradient information in training; and *ii*) the optimization task model is repeatedly solved per updating epoch, resulting in a time-consuming training. To adapt this approach for mixed-integer programming (MIP)-based tasks, great efforts have been made in [31]–[35]. Nevertheless, the adaptability is achieved by either compromisingly relaxing/fixing integer variables [31], [33], [35] or ignoring the non-differentiable issue [32], [34].

In response to this limitation, references [37]–[39] propose to use mathematical programming-based methods for training predictors. This training typically involves two steps: *i*) constructing the empirical risk minimization (ERM) problem as an explicit mathematical model (e.g., MIP), which integrates the predictor training and task optimization in a closed-loop manner; and *ii*) solving this ERM model to yield optimally trained predictors. As illustrated in [37]–[39], CPO frameworks based on this training inherit two key properties of mathematical programming-based methods: they can uncompromisingly handle the integer variables because the training

does not depend on the gradient, and they exhibit consistent performance as the predictor is trained optimally.

Nevertheless, the ERM model in [37]–[39] is being challenged recently. One major concern arises from their *single-level* structure—a structure that cannot capture the chronological sequence between the prediction and optimization. This inability may incur an over-fitting predictor, especially when the training samples are limited, as emphasized in [41]. To overcome this issue, references [40]–[46] use *bi-level* programming-based ERMs—training the predictor at the upper level while mimicking the optimization task at the lower level. Their numerical results demonstrate that CPO frameworks with the bi-level ERM model can improve task economics remarkably while mitigating the over-fitting issue.

Based on the above literature review and Table I, an important research gap can be revealed—the existing bi-level programming-based CPO frameworks [40]–[46] exclusively target LP tasks. This is largely because their training methods rely on the favorable properties of LP models. Typically, a lower-level LP subproblem can be equivalently replaced with its Karush–Kuhn–Tucker (KKT) conditions, thereby converting the original bi-level model into a tractable single-level form. However, it has been widely recognized that many practical operation tasks can only be reasonably formed as MIP problems, in which the KKT conditions cannot be applied directly. To this end, a broader adaptability of the bi-level programming-based CPO frameworks is necessary.

C. Main Works and Contributions

Motivated by the above-identified research gap, this work presents a bi-level MIP-based CPO framework, which trains

a cost-oriented tailor of RES-and-reserve predictions for the MIP-based UC. The term “*tailor*” refers to a custom-made predictor that can modify the raw predictions into cost-oriented predictions for improving the UC economics.

To train the tailor, the ERM problem is constructed in a bi-level MIP form. Its upper level trains the tailors for RES power as well as SR and NR requirements, in which the input/output is the raw/tailored prediction. The bi-level ERM employs *two* MIP-based lower levels to mimic UC and RD operation processes. These lower levels utilize the upper-level predictions to calculate the actual operating cost, which is then fed back to the upper level for improving the tailoring quality.

Regarding the solving method for the ERM model, it begins by proving that optimal solutions of the original bi-level MIP (with two lower levels) can be obtained by solving a relatively tractable bi-level MIP (with only one lower level). The more tractable bi-level MIP is then solved by a cutting plane method, yielding the optimally trained tailors of RES, SR, and NR.

Finally, as shown in Fig. 1(b), the tailors act as add-ons to the existing predict-then-optimize practice, positioned between the currently-used predictors and the UC model. The tailors can customize the raw RES-SR-NR predictions into their cost-oriented counterparts, thus improving the UC economics.

Our contributions are summarized as twofold:

1) To improve the UC economics, a CPO framework including a bi-level MIP-based ERM model and its solution method is presented. Compared to the existing works, the proposed CPO framework is distinguished by the following aspects:

- The presented bi-level MIP-based CPO framework can economically benefit UC and other MIP-based operation tasks in power systems. As compared, most gradient-based methods (e.g., [28]) and bi-level LP-based frameworks (e.g., [40]–[46]) are restricted to LP tasks or require compromises, e.g., relaxing binary variables;
- Recall that revealing the eventual UC economics necessitates solving the MIP-based RD model. To this end, the presented bi-level ERM model is designed to incorporate both UC and RD into the training as two lower-level sub-problems. In comparison, the existing UC-related works (i.e., [22], [27], [31]–[33], [38], [39]) neglect the RD task. Therefore, the presented framework is versatile for more general tasks;
- Theoretically, the presented mathematical programming-based training method can consistently approach a globally optimal predictor. Contrarily, the training in [22]–[36] typically provides a locally-optimal predictor at random. This means that the presented framework can perform consistently, especially when the number of training samples is limited.

2) To illustrate the effectiveness of the presented CPO framework, comprehensive comparisons are conducted using real-world data. Specifically, the presented CPO framework is compared to the predict-then-optimize practice, bi-level relaxed MIP, NN-based CPO [24], TSP [6], and TRO [7].

The rest of the paper is organized as follows: Section II details the necessary preliminaries; Section III expounds on the training of the tailor; Section IV analyzes the numerical results; and Section V concludes this paper.

II. PRELIMINARIES

A. Operation Models

Taking the raw predictions \hat{w} and \hat{r} as inputs, the operators solve the MIP-based UC model to determine the day-ahead operation plans. The compact form of UC is shown as (1), and its detailed formulation is presented in Appendix A.

$$\min_{\mathbf{x}, \mathbf{y}} \mathbf{b}^\top \mathbf{x} + \mathbf{c}^\top \mathbf{y} \quad (1.1)$$

$$s. t. \mathbf{x}, \mathbf{y} \in \mathcal{X}(\hat{w}, \hat{r}); \quad (1.2)$$

$$\mathcal{X}(\hat{w}, \hat{r}) = \left\{ \mathbf{x}, \mathbf{y} \mid \begin{array}{l} \mathbf{F}(\mathbf{x}, \mathbf{y}) = 0; \\ \mathbf{G}(\mathbf{x}, \mathbf{y}, \hat{w}, \hat{r}) \leq 0; \end{array} \right\}; \quad (1.3)$$

Here, $\mathbf{F}(\cdot)/\mathbf{G}(\cdot)$ represents the set of UC equality/inequality constraints, forming the feasible region \mathcal{X} . With raw predictions \hat{w} and \hat{r} as inputs, UC is solved to deliver optimal solutions of startup, commitment, baseline generation, SR, and NR schedules [47].

Next, with respect to the operation plans $\{\mathbf{x}^*, \mathbf{y}^*\}$ from UC, the best balancing operations against actual RES realization \tilde{w} can be revealed by solving the MIP-based RD model. The compact form of the RD is shown as (2), and its detailed formulation is presented in Appendix B.

$$\min_{\mathbf{z}} \mathbf{d}^\top \mathbf{z} \quad (2.1)$$

$$s. t. \mathbf{z} \in \mathcal{Z}(\mathbf{x}^*, \mathbf{y}^*, \tilde{w}); \quad (2.2)$$

$$\mathcal{Z}(\mathbf{x}^*, \mathbf{y}^*, \tilde{w}) = \left\{ \mathbf{z} \mid \begin{array}{l} \mathbf{M}(\mathbf{x}^*, \mathbf{z}) = 0; \\ \mathbf{N}(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}, \tilde{w}) \leq 0 \end{array} \right\}; \quad (2.3)$$

Here, $\mathbf{M}(\cdot)/\mathbf{N}(\cdot)$ denotes the set of RD equality/inequality constraints, and \mathcal{Z} is the feasible region. Solving RD provides optimal solutions to status switch of quick-start units, generation adjustments of all units, and slack penalties against the RES realization \tilde{w} [48].

B. The Evaluation of UC Economics

The UC economics can be assessed in three steps:

i) As the realization \tilde{w} remains unknown in the day-ahead stage, operators first conduct UC based on raw predictions \hat{w} and \hat{r} . This step provides the *actual UC cost* $\mathbf{b}^\top \mathbf{x}^*$ consisting of startup and no-load costs;

ii) After \tilde{w} is revealed, RD is executed based on the UC decisions \mathbf{x}^* . This step yields the *balancing cost* $\mathbf{d}^\top \mathbf{z}^*$, including startup and no-load costs of quick-start units additionally committed in RD, generation costs of all units, and slack penalties;

iii) Eventually, equation (3) provides the *best possible actual operating cost*, i.e., c^{act} , that is realistically achievable w.r.t. the UC solution and actual RES realization. This is referred to as the *UC economics* in this paper.

$$\begin{aligned} c^{\text{act}} &= \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} (C_i^{\text{su}} U_{it}^* + C_i^{\text{nl}} I_{it}^*) \} \mathbf{b}^\top \mathbf{x}^* \\ &+ \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} [(C_i^{\text{su}} U_{it}^{\text{RD}*} + C_i^{\text{nl}} I_{it}^{\text{RD}*}) + \sum_{k \in \mathcal{K}} C_{ik}^{\text{sg}} P_{itk}^{\text{RD}, \text{sg}*}] \} \mathbf{d}^\top \mathbf{z}^* \\ &+ \sum_{t \in \mathcal{T}} [C_t^{\text{gs}} S_t^{\text{gs}*} + C_t^{\text{ls}} S_t^{\text{ls}*} + \sum_{b \in \mathcal{B}} C_b^{\text{rb}} (S_{bt}^{\text{+}*} + S_{bt}^{\text{-}*})] \} \mathbf{d}^\top \mathbf{z}^* \\ &= \mathbf{b}^\top \mathbf{x}^* + \mathbf{d}^\top \mathbf{z}^* \end{aligned} \quad (3)$$

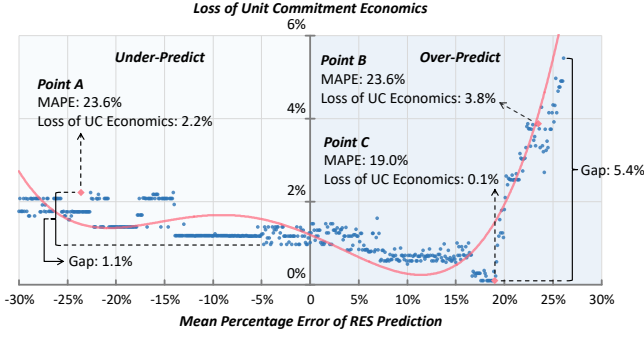


Fig. 3. Impact of RES prediction error on UC economics.

C. The Asymmetric Relationship between UC Economics and RES Prediction Accuracy

Most RES predictors use symmetrical metrics to evaluate statistical prediction accuracy. However, the relationship between the prediction accuracy and UC economics is inherently asymmetrical. To show this point, multiple raw RES predictions $\hat{\mathbf{w}}$ are generated based on actual Belgian system data [2] on 12/17/2020. These raw predictions are applied on an IEEE 14-bus system to simulate the existing predict-then-optimize process, as shown in Fig. 1(a). Finally, the loss of UC economics is evaluated by (4), where $c^{\text{act,raw}}$ and $c^{\text{act,perf}}$ are actual operating costs induced by raw $\hat{\mathbf{w}}$ and perfect error-free prediction $\tilde{\mathbf{w}}$. The reserve requirements are set as raw.

$$\text{Loss of UC Economics} = \frac{c^{\text{act,raw}} - c^{\text{act,perf}}}{c^{\text{act,perf}}} \times 100\% \quad (4)$$

Fig. 3 shows the induced results of all raw prediction points, and the red line fits the relationship between MAPE and UC economics loss. The line clearly illustrates the asymmetry from a macro perspective: the loss caused by the over-predictions is fairly small in small-MAPE (5% to 15%) cases but becomes significant in large-MAPE (20% to 30%) cases, resulting in a 5.4% width gap between the best and worst economic losses; in comparison, the loss induced by the under-predictions, the blue dots on the left half plane, is more stable with a narrower gap. Moreover, two cases are highlighted as follows:

Case 1: Raw predictions of the same MAPE may lead to different losses in UC economics. Note that points A and B in Fig. 3 have the same MAPE (23.6%) but different losses (2.2% vs 3.8%). This is because the under-prediction of point A merely causes downward generation adjustment and RES curtailments in the RD; in comparison, the over-prediction of point B necessitates deploying more expensive NR from quick-start units in the RD;

Case 2: Raw predictions of a larger MAPE may even lead to a smaller loss in UC economics. Although the MAPE of point C is noticeably high (19.0%), it achieves the lowest loss (0.1%) among all raw predictions. This is because each $\hat{\mathbf{w}}$ is associated with a just-enough reserve level $\hat{\mathbf{r}}^*$ w.r.t. $\hat{\mathbf{w}}$ —a proper $\hat{\mathbf{r}}^*$ depends on the difference between $\hat{\mathbf{w}}$ and $\tilde{\mathbf{w}}$ as well as other system specifications. Consequently, if $\{\hat{\mathbf{w}}, \hat{\mathbf{r}}^*\}$ is taken as input to (1), the induced UC decisions can deploy $\hat{\mathbf{r}}^*$ to enable the most effective utilization of available RES in RD operations, achieving the best UC economics.

Thus, predictors trained via symmetrical accuracy metrics, although suitable for general purposes, ignore the inherent asymmetry of the UC problem and the subsequent RD process. This fact inspires us to develop a RES-and-reserve tailor for the UC problem. The tailor can customize raw predictions into a cost-oriented counterpart by considering the subsequent RD result, thus boosting the UC economics.

D. Major Flaws in Single-Level ERM

Model (5) is a typical single-level ERM, where s/S is the index/set of training samples; c_s^{perf} is the objective value (1.1) induced by the actual RES realization $\tilde{\mathbf{w}}_s$; $\mathcal{W}(\cdot)$ is the RES predictor to be trained, with \mathcal{P}^w being its feasible region; \mathbf{f}_s is the input feature. The ERM aims to train a predictor $\mathcal{W}(\cdot)$ that can induce the UC cost closer to c_s^{perf} for individual samples.

$$\min_{\mathcal{W}, \mathbf{x}_s, \mathbf{y}_s} \frac{1}{|S|} \sum_{s \in S} \|\mathbf{b}^\top \mathbf{x}_s + \mathbf{c}^\top \mathbf{y}_s - c_s^{\text{perf}}\|_1 \quad (5.1)$$

$$s.t. \mathcal{W}(\cdot) \in \mathcal{P}^w; \quad (5.2)$$

$$\mathbf{x}_s, \mathbf{y}_s \in \mathcal{X}(\mathcal{W}(\mathbf{f}_s), \hat{\mathbf{r}}_s), \forall s \in S; \quad (5.3)$$

The major flaw of (5) is that its in-sample UC solutions $\{\mathbf{x}_s^*, \mathbf{y}_s^*\}$ may use more expensive units, even if cheaper units are feasible, to pursue the minimum loss (5.1). Such UC decisions violate the least-cost principle² of UC models, thus being compromised.

An example with two samples, s_1 and s_2 , can illustrate this flaw [41]. Both samples have the same features ($\mathbf{f}_{s_1} = \mathbf{f}_{s_2}$) but different RES realizations ($\tilde{\mathbf{w}}_{s_1} \neq \tilde{\mathbf{w}}_{s_2}$). With these two samples, the single-level model (5) *cannot* ensure their in-sample UC decisions, $\{\mathbf{x}_{s_1}^*, \mathbf{y}_{s_1}^*\}$ and $\{\mathbf{x}_{s_2}^*, \mathbf{y}_{s_2}^*\}$, are the same. This fact can be explained by the following two steps:

- i) $\mathbf{f}_{s_1} = \mathbf{f}_{s_2}$ means $\mathcal{W}(\mathbf{f}_{s_1}) = \mathcal{W}(\mathbf{f}_{s_2})$. That is, their in-sample predictions are the same;
- ii) $\tilde{\mathbf{w}}_{s_1} \neq \tilde{\mathbf{w}}_{s_2}$ will lead to $c_{s_1}^{\text{perf}} \neq c_{s_2}^{\text{perf}}$. Given this, (5) will derive different in-sample UC decisions $\{\mathbf{x}_{s_1}^*, \mathbf{y}_{s_1}^*\} \neq \{\mathbf{x}_{s_2}^*, \mathbf{y}_{s_2}^*\}$ to achieve the minimum training loss (5.1). However, the UC model (1), which follows the least-cost principle, necessitates that the same predictions must lead to the same optimal UC decisions. Thus, it can be concluded that at least one of the two in-sample decisions is compromised by violating the least-cost principle.

From the perspective of mathematical modeling, this flaw arises because the predictor training and UC decision-making are packed into a single level without following the chronological order—they are mutually anticipatable³ in (5). As a result, (5) will derive unequal UC solutions $\{\mathbf{x}_{s_1}^*, \mathbf{y}_{s_1}^*\} \neq \{\mathbf{x}_{s_2}^*, \mathbf{y}_{s_2}^*\}$ to chase a lower training loss (5.1). In other words, in-sample UC decisions of (5) are made to adapt the in-sample cost c^{perf} instead of minimizing operating cost, thus failing to reflect the least-cost principle-based decision-making of operators. As a result, predictors trained via the single-level

²As explained in [44], system operators are responsible for identifying the least-cost UC decision against the currently available predictions, known as the least-cost principle. Thus, operators generally use (1.1) as the objective.

³Anticipativity and non-anticipativity are concepts in stochastic programming. They indicate that in a multi-stage model, the current stage can (anticipativity) or cannot (non-anticipativity) access and manipulate the information and decisions of future stages.

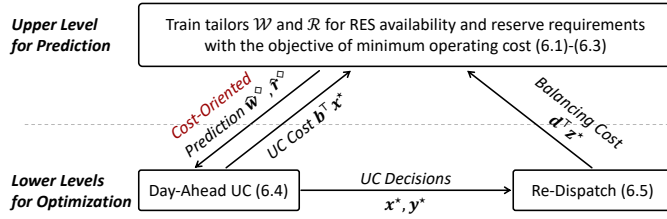


Fig. 4. The closed-loop predict-and-optimize mechanism in the ERM (6).

ERM (5), although presenting good in-sample losses, could fail to generalize well (i.e., over-fitting) because the in-sample UC decisions are compromised.

In comparison, constructing ERM as a bi-level MIP can layer the prediction and optimization problems in a proper chronological order, thus satisfying the non-anticipativity naturally. In this way, the in-sample UC decisions always minimize the operating cost (1.1), thus strictly obeying the least-cost principle and effectively mitigating the over-fitting issue.

III. RES-AND-RESERVE TAILORS BASED ON BI-LEVEL MIP

A. Constructing the ERM Problem

Using a pre-specified dataset \mathcal{S} (Section IV-A1 will detail the dataset construction) as well as raw predictions \hat{w}_s and \hat{r}_s , the bi-level ERM problem is constructed as in (6). The upper level (6.1)-(6.3) trains tailors \mathcal{W} and \mathcal{R} for RES, SR, and NR, taking actual operating costs (3) as the loss function (6.1). The loss function maximizes the weighted sum of UC economics for all samples in \mathcal{S} . Constraint (6.2) limits the tailor parameters within their feasible regions \mathcal{P}^w and \mathcal{P}^r ; (6.3) defines \mathcal{W} and \mathcal{R} , which respectively take raw predictions \hat{w}_s and \hat{r}_s as inputs. The two lower levels, (6.4) and (6.5), model UC (1) and RD (2), respectively. Solving (6) can provide cost-oriented tailors, \mathcal{W}^* and \mathcal{R}^* , that are trained optimally.

Upper Level (Training Tailors of Raw Predictions):

$$\min_{\mathcal{W}, \mathcal{R}} \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \mathbf{b}^\top \mathbf{x}_s + \mathbf{d}^\top \mathbf{z}_s \quad (6.1)$$

$$s.t. \mathcal{W}(\cdot) \in \mathcal{P}^w, \mathcal{R}(\cdot) \in \mathcal{P}^r; \quad (6.2)$$

$$\hat{w}_s^\diamond = \mathcal{W}(\hat{w}_s), \hat{r}_s^\diamond = \mathcal{R}(\hat{r}_s), \quad \forall s \in \mathcal{S}; \quad (6.3)$$

Lower Level (Day-Ahead Unit Commitment):

$$\mathbf{x}_s, \mathbf{y}_s \in \arg \min_{\mathbf{x}_s, \mathbf{y}_s \in \mathcal{X}(\hat{w}_s^\diamond, \hat{r}_s^\diamond)} \mathbf{b}^\top \mathbf{x}_s + \mathbf{c}^\top \mathbf{y}_s, \quad \forall s \in \mathcal{S}; \quad (6.4)$$

Lower Level (Re-Dispatch):

$$\mathbf{z}_s \in \arg \min_{\mathbf{z}_s \in \mathcal{Z}(\mathbf{x}_s, \mathbf{y}_s, \hat{w}_s)} \mathbf{d}^\top \mathbf{z}_s, \quad \forall s \in \mathcal{S}; \quad (6.5)$$

The bi-level ERM (6) features that: *i*) its bi-level structure and the $\arg \min$ operators of (6.4) and (6.5) enable the in-sample decisions to obey the least-cost principle; and *ii*) it can incorporate two MIP-based tasks into the training.

Fig. 4 expounds how the bi-level ERM (6) achieves the closed-loop predict-and-optimize idea. The upper level (6.1) generates the tailored predictions \hat{w}_s^\diamond and \hat{r}_s^\diamond , but cannot directly assess their induced UC economics. Thus, \hat{w}_s^\diamond and \hat{r}_s^\diamond are first passed to the lower-level UC (6.4), forming the UC

feasible region \mathcal{X} . Solving (6.4) provides the actual UC cost $\mathbf{b}^\top \mathbf{x}^*$ to the upper level as well as the UC decisions $\{\mathbf{x}^*, \mathbf{y}^*\}$ to the lower-level RD (6.5). Taking $\{\mathbf{x}^*, \mathbf{y}^*\}$ as inputs, (6.5) is solved to further provide the balancing cost $\mathbf{d}^\top \mathbf{z}^*$ to the upper level. Finally, the upper level can assess the prediction quality via the actual operating cost $\mathbf{b}^\top \mathbf{x}^* + \mathbf{d}^\top \mathbf{z}^*$, and tune the tailor parameters accordingly. This process repeats until the tailors \mathcal{W} and \mathcal{R} are trained optimally. Since tailors \mathcal{W}^* and \mathcal{R}^* are trained in favor of the UC economics, they are cost-oriented customization for the UC problem.

B. On Selecting Specific Models for Tailor

The tailors act as add-ons in the existing predict-then-optimize process, in which they output prediction vectors \hat{w}_s^\diamond and \hat{r}_s^\diamond for the day-ahead UC. Given that these tailors are tasked with predicting continuous values, their training could be better treated as a *regression problem*.

In theory, the ERM model (6) can accommodate various *regression forms* by formulating them as explicit constraints (6.3) to be handled by the solvers. That is, (6.3) can take various forms, such as linear, polynomial, logarithmic, and parametric ReLU. Among these forms, this paper selects the affine linear form (7).

$$\mathcal{W}(\hat{w}_s) = \mathbf{m} \circ \hat{w}_s, \mathbf{m} \in \mathbb{R}_+^{(|\mathcal{T}| \times |\mathcal{J}|) \times 1}; \quad (7.1)$$

$$\mathcal{R}(\hat{r}_s) = \mathbf{n} \circ \hat{r}_s, \mathbf{n} \in \mathbb{R}_+^{2|\mathcal{T}| \times 1}; \quad (7.2)$$

In (7), \circ is the element-wise product, and the tailor parameters are constrained to nonnegative real numbers. With this, the essence of the tailoring is to linearly scale raw predictions in favor of UC economics. The training is basically to solve the ERM (6) for determining the optimal vectors \mathbf{m} and \mathbf{n} .

Notably, the topic of selecting a proper regression form has attracted increasing attention within the IPO and CPO fields. This interest mainly stems from the fact that IPO and CPO methods are designed for a specific task (e.g., UC), necessitating their forms to align with the task properties.

Specifically, the prescription model of IPO methods directly maps features to the ultimate operation decisions. Consequently, complex forms, such as NN [19], are typically used to capture the highly complicated and non-linear relationship. On the other hand, the prediction model of CPO methods first transforms the input feature into cost-oriented prediction, which is subsequently used as input to operation models for optimizing ultimate decisions. In this case, the form of the prediction model mainly depends on the relationship between the input feature and the cost-oriented prediction.

In the presented CPO method, the high-quality raw predictions sourced from sophisticated predictors, rather than basic meteorological data like temperature, are used as features. For such a case, [20], [38], and [41]–[43] have demonstrated that the raw prediction generally requires only minor adjustments to render its cost-oriented counterpart. This is why the paper uses “tailor” instead of “predictor” to describe \mathcal{W} and \mathcal{R} . Thus, the straightforward linear form (7) would be suitable for the presented method due to its balanced effectiveness, efficiency, interpretability, and numerical stability.

C. Solving the ERM Problem

The primary computational challenge in solving the ERM problem (6) arises from the binary variables in the two lower-level problems, i.e., (6.4) and (6.5). If no binary variables exist, (6.4) and (6.5) can be equivalently converted to their KKT conditions, enabling (6) to be solved as an MIP problem.

The presented method takes two steps to handle the above challenge: *i*) proving that the optimal solutions of (6) can be obtained by solving a more tractable bi-level MIP with only one lower-level subproblem; and *ii*) solving this more tractable bi-level MIP with only one lower-level subproblem through a cutting plane-based method.

1) *Converting (6) with Two Lower-Level MIP Subproblems to (8) with One Lower-Level MIP Subproblem:* The bi-level ERM (6) includes two MIP subproblems in the lower level, introducing non-negligible computational complexity.

To solve (6), Proposition 1 is proposed to convert (6) into a more tractable bi-level MIP (8) with one MIP in the lower level. Solving (8) provides the optimal solution of the original ERM (6), as proven in Proposition 1.

Proposition 1: The optimal solution of the bi-level MIP (8) is also optimal for the original bi-level MIP (6).

$$\min_{\mathcal{W}, \mathcal{R}, \mathbf{z}_s} \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \mathbf{b}^\top \mathbf{x}_s + \mathbf{d}^\top \mathbf{z}_s \quad (8.1)$$

$$s. t. \mathcal{W}(\cdot) \in \mathcal{P}^w, \mathcal{R}(\cdot) \in \mathcal{P}^r; \quad (8.2)$$

$$\hat{\mathbf{w}}_s^\diamond = \mathcal{W}(\hat{\mathbf{w}}_s), \hat{\mathbf{r}}_s^\diamond = \mathcal{R}(\hat{\mathbf{r}}_s), \quad \forall s \in \mathcal{S}; \quad (8.3)$$

$$\mathbf{z}_s \in \mathcal{Z}(\mathbf{x}_s, \mathbf{y}_s, \hat{\mathbf{w}}_s), \quad \forall s \in \mathcal{S}; \quad (8.4)$$

$$\mathbf{x}_s, \mathbf{y}_s \in \arg \min_{\mathbf{x}_s, \mathbf{y}_s \in \mathcal{X}(\hat{\mathbf{w}}_s^\diamond, \hat{\mathbf{r}}_s^\diamond)} \mathbf{b}^\top \mathbf{x}_s + \mathbf{c}^\top \mathbf{y}_s, \quad \forall s \in \mathcal{S}; \quad (8.5)$$

Proof: To begin with, model (9) is introduced to bridge (6) and (8), which is constructed by adding (9.4) to (6), or equivalently, by adding (9.6) to (8).

$$\min_{\mathcal{W}, \mathcal{R}, \mathbf{z}_s} \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \mathbf{b}^\top \mathbf{x}_s + \mathbf{d}^\top \mathbf{z}_s \quad (9.1)$$

$$s. t. \mathcal{W}(\cdot) \in \mathcal{P}^w, \mathcal{R}(\cdot) \in \mathcal{P}^r; \quad (9.2)$$

$$\hat{\mathbf{w}}_s^\diamond = \mathcal{W}(\hat{\mathbf{w}}_s), \hat{\mathbf{r}}_s^\diamond = \mathcal{R}(\hat{\mathbf{r}}_s), \quad \forall s \in \mathcal{S}; \quad (9.3)$$

$$\mathbf{z}_s \in \mathcal{Z}(\mathbf{x}_s, \mathbf{y}_s, \hat{\mathbf{w}}_s), \quad \forall s \in \mathcal{S}; \quad (9.4)$$

$$\mathbf{x}_s, \mathbf{y}_s \in \arg \min_{\mathbf{x}_s, \mathbf{y}_s \in \mathcal{X}(\hat{\mathbf{w}}_s^\diamond, \hat{\mathbf{r}}_s^\diamond)} \mathbf{b}^\top \mathbf{x}_s + \mathbf{c}^\top \mathbf{y}_s, \quad \forall s \in \mathcal{S}; \quad (9.5)$$

$$\mathbf{z}_s \in \arg \min_{\mathbf{z}_s \in \mathcal{Z}(\mathbf{x}_s, \mathbf{y}_s, \hat{\mathbf{w}}_s)} \mathbf{d}^\top \mathbf{z}_s, \quad \forall s \in \mathcal{S}; \quad (9.6)$$

First, we claim that (6) and (9) have the same optimal solutions. This can be proved by showing that they have identical objective functions as well as feasible regions. It can be directly observed that their objective functions, (6.1) and (9.1), are the same. To show that (6) and (9) have the same feasible region, we point out the following two facts: *i*) a feasible solution of (6) satisfies all constraints of (9) and thus is feasible to (9); and *ii*) a feasible solution of (9) also satisfies all constraints of (6) and thus is feasible to (6). Thus, it is proven that (6) and (9) have the same optimal solutions.

Second, we claim that the optimal solution of (8), denoted as $\Omega^* = \{\mathbf{x}_s^*, \mathbf{y}_s^*, \mathbf{z}_s^*\}$, is also optimal for (9). Note that since (8) is equivalent to (9.1)-(9.5), Ω^* is optimal for (9.1)-(9.5). Thus, we only need to show that Ω^* also satisfies (9.6), or

Algorithm 1: A C&CG algorithm for solving (8)

Input: Desired optimality gap ϵ^* , initial tailors $\mathcal{W}_{e=0}$ and $\mathcal{R}_{e=0}$, initial lower/upper bound LB/UB .

Output: Optimally trained tailors \mathcal{W}^* and \mathcal{R}^* .

while $e \leq E$ **do**

Solve **SP1** and **SP2** for each sample $s \in \mathcal{S}$;

$UB \leftarrow \min\{UB, \sum_{s \in \mathcal{S}} \frac{c_s^{\text{act}*}}{|\mathcal{S}|} + \lambda^w \|\mathcal{W}\|_1 - \lambda^r \|\mathcal{R}\|_1\}$;

$\epsilon \leftarrow 100\% \times \frac{UB-LB}{UB}$;

if $\epsilon \leq \epsilon^*$ **then**

$\mathcal{W}^* \leftarrow \mathcal{W}_e, \mathcal{R}^* \leftarrow \mathcal{R}_e$;

break;

else

$\bar{\mathbf{x}}_{s,e}^{\text{ev}} \leftarrow$ optimal solution of decision \mathbf{x}_s in **SP2**;

Add new $\{\bar{\mathbf{x}}_{s,e}^{\text{ev}}, \bar{\mathbf{y}}_{s,e}^{\text{ev}}\}$ and cuts (13.6)-(13.10) to

MP;

Solve **MP**;

$e \leftarrow e + 1, LB \leftarrow$ optimal objective value of **MP**;

\mathcal{W}_e and $\mathcal{R}_e \leftarrow$ optimal \mathcal{W} and \mathcal{R} from **MP**;

end

end

equivalently, that \mathbf{z}_s^* satisfies (9.6). This can be achieved by three steps: *i*) construct model (10) by removing (9.2)-(9.5) from (9) and fixing the UC decisions at their optima \mathbf{x}_s^* and \mathbf{y}_s^* . Note that (10) is equivalent to (9.6); *ii*) recall that Ω^* is an optimal solution of (8), thus, its \mathbf{z}_s^* must be optimal for (10); *iii*) by combining steps *i*) and *ii*), it is direct to conclude that \mathbf{z}_s^* satisfies (9.6).

As we have proved that Ω^* is optimal for (9.1)-(9.5) and its \mathbf{z}_s^* also satisfies (9.6), it can be claimed that the optimal solution Ω^* of (8) must be optimal for (9).

$$\min_{\mathbf{z}_s} \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \mathbf{b}^\top \mathbf{x}_s^* + \mathbf{d}^\top \mathbf{z}_s \quad (10.1)$$

$$s. t. \mathbf{z}_s \in \arg \min_{\mathbf{z}_s \in \mathcal{Z}(\mathbf{x}_s^*, \mathbf{y}_s^*, \hat{\mathbf{w}}_s)} \mathbf{d}^\top \mathbf{z}_s, \quad \forall s \in \mathcal{S}; \quad (10.2)$$

Finally, by combining the first claim (i.e., (6) and (9) have the same optimal solutions) and the second claim (i.e., the optimal solution of (8) is also optimal for (9)), it is direct to conclude that the optimal solution of (8) is optimal for (6). ■

2) *Solving (8) by C&CG-based Algorithm:* The bi-level MIP (8) can be solved by either Benders decomposition [49] or column-and-constraint generation (C&CG). Since Benders decomposition may converge slowly, a C&CG-based algorithm is selected, which iterates between the master problem **MP** and two subproblems **SP1** and **SP2**, as summarized in Algorithm 1.

The first subproblem **SP1** is presented as in (11). It is a duplication of the UC problem (8.5), which takes the incumbent tailors \mathcal{W}_e and \mathcal{R}_e as inputs. Solving **SP1** can reveal the least-cost UC decision induced by the incumbent tailors, as well as its associated cost $c_s^{\text{SP1}*}$.

$$\mathbf{SP1}: c_s^{\text{SP1}*} = \min_{\mathbf{x}_s, \mathbf{y}_s} \mathbf{b}^\top \mathbf{x}_s + \mathbf{c}^\top \mathbf{y}_s \quad (11.1)$$

$$s. t. \hat{\mathbf{w}}_s^\diamond = \mathcal{W}_e(\hat{\mathbf{w}}_s), \hat{\mathbf{r}}_s^\diamond = \mathcal{R}_e(\hat{\mathbf{r}}_s); \quad (11.2)$$

$$\mathbf{F}(\mathbf{x}_s, \mathbf{y}_s) = 0, \mathbf{G}(\mathbf{x}_s, \mathbf{y}_s, \hat{\mathbf{w}}_s^\diamond, \hat{\mathbf{r}}_s^\diamond) \leq 0; \quad (11.3)$$

Note that multiple optimal binaries \mathbf{x}_s^* for **SP1** may exist, and they could lead to different objective values of (8.1). Thus, the second subproblem **SP2**, formed as in (12) with $c_s^{\text{SP1}*}$ as

input, is further solved to select a solution among multiple \mathbf{x}_s^* of **SP1** in favor of the objective (8.1). This is referred to as *optimistic bi-level programming* [50].

$$\mathbf{SP2}: c_s^{\text{act}*} = \min_{\mathbf{x}_s, \mathbf{y}_s, \mathbf{z}_s} \mathbf{b}^\top \mathbf{x}_s + \mathbf{d}^\top \mathbf{z}_s \quad (12.1)$$

$$s.t. \hat{\mathbf{w}}_s^\diamond = \mathcal{W}_e(\hat{\mathbf{w}}_s), \hat{\mathbf{r}}_s^\diamond = \mathcal{R}_e(\hat{\mathbf{r}}_s); \quad (12.2)$$

$$\mathbf{b}^\top \mathbf{x}_s + \mathbf{c}^\top \mathbf{y}_s \leq c_s^{\text{SP1}*}; \quad (12.3)$$

$$\mathbf{F}(\mathbf{x}_s, \mathbf{y}_s) = 0, \mathbf{G}(\mathbf{x}_s, \mathbf{y}_s, \hat{\mathbf{w}}_s^\diamond, \hat{\mathbf{r}}_s^\diamond) \leq 0; \quad (12.4)$$

$$\mathbf{M}(\mathbf{x}_s, \mathbf{z}_s) = 0, \mathbf{N}(\mathbf{x}_s, \mathbf{y}_s, \mathbf{z}_s, \hat{\mathbf{w}}_s) \leq 0; \quad (12.5)$$

Given the binary decisions \mathbf{x}_s^* identified by **SP2**, the master problem **MP** is formulated as (13), which is composed of the *high-point problem* [50] (13.1)-(13.5) of (8) and the cutting planes (13.6)-(13.10). Three variants of variables $\{\mathbf{x}_s, \mathbf{y}_s\}$ are used in **MP**, including duplicated variables $\{\mathbf{x}_s^{\text{dv}}, \mathbf{y}_s^{\text{dv}}\}$, enumerated binary variables $\hat{\mathbf{x}}_{s,e}^{\text{ev}}$, and generated continuous variables $\mathbf{y}_{s,e}^{\text{gv}}$. The duplicated variables serve as the proxy of the original variables, but with a larger feasible region; the enumerated and generated variables and their associated cutting planes (13.6)-(13.10) serve to cut the feasible region of duplicated variables gradually. **MP** is built in three parts:

- The first part (13.1)-(13.4) is based on the duplicated variables. The objective function (13.1) includes two regularization terms, $\lambda^w \|\mathcal{W}\|_1$ and $\lambda^r \|\mathcal{R}\|_1$, to mitigate over-predictions and insufficient reserves via hyper-parameters λ^w and λ^r . The regularization terms could be removed when the number of training samples is sufficient. Constraints (13.2)-(13.4) are the original upper-level constraints (8.2)-(8.4);
- The second part (13.5) is a duplication of UC constraints to ensure the feasibility of the duplicated variables;
- The third part (13.6)-(13.10) acts as the *optimality cuts*. This part is built by three steps: *i*) duplicating the MIP-based UC subproblem (8.5) for each sample s in each iteration e ; *ii*) for each duplicated UC subproblem, fixing its binary variables as $\hat{\mathbf{x}}_{s,e}^{\text{ev}}$ that are given by **SP2**, which makes these MIPs degenerate to LPs; and *iii*) substituting these LPs by their KKT conditions (13.6)-(13.9). Here, $\mathcal{L}(\cdot)$ denotes the Lagrangian function; $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are dual variables of equality and inequality constraints. The KKT conditions (13.6)-(13.9) ensure the optimality of the generated variables w.r.t. $\hat{\mathbf{x}}_{s,e}^{\text{ev}}$. Finally, (13.10) links the duplicated variables with their optimality cuts.

$$\mathbf{MP}: \min_{\Xi} \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} (\mathbf{b}^\top \mathbf{x}_s^{\text{dv}} + \mathbf{d}^\top \mathbf{z}_s) + \lambda^w \|\mathcal{W}\|_1 - \lambda^r \|\mathcal{R}\|_1 \quad (13.1)$$

$$\text{where } \Xi = \{\mathcal{W}(\cdot), \mathcal{R}(\cdot), \hat{\mathbf{w}}_s^\diamond, \hat{\mathbf{r}}_s^\diamond, \mathbf{z}_s, \mathbf{x}_s^{\text{dv}}, \mathbf{y}_s^{\text{dv}}, \mathbf{y}_{s,e}^{\text{gv}}\} \quad (13.1)$$

Subject to:

Original Constraints:

$$\mathcal{W}(\cdot) \in \mathcal{P}^w, \mathcal{R}(\cdot) \in \mathcal{P}^r; \quad (13.2)$$

$$\hat{\mathbf{w}}_s^\diamond = \mathcal{W}(\hat{\mathbf{w}}_s), \hat{\mathbf{r}}_s^\diamond = \mathcal{R}(\hat{\mathbf{r}}_s), \quad \forall s \in \mathcal{S}; \quad (13.3)$$

$$\mathbf{M}(\mathbf{x}_s^{\text{dv}}, \mathbf{z}_s) = 0, \mathbf{N}(\mathbf{x}_s^{\text{dv}}, \mathbf{y}_s^{\text{dv}}, \mathbf{z}_s, \hat{\mathbf{w}}_s) \leq 0, \quad \forall s \in \mathcal{S}; \quad (13.4)$$

Duplication of UC Constraints:

$$\mathbf{F}(\mathbf{x}_s^{\text{dv}}, \mathbf{y}_s^{\text{dv}}) = 0, \mathbf{G}(\mathbf{x}_s^{\text{dv}}, \mathbf{y}_s^{\text{dv}}, \hat{\mathbf{w}}_s^\diamond, \hat{\mathbf{r}}_s^\diamond) \leq 0, \quad \forall s \in \mathcal{S}; \quad (13.5)$$

Stationarity of KKT (Optimality Cuts):

$$\nabla \mathcal{L}(\mathbf{y}_{s,e}^{\text{gv}}, \boldsymbol{\mu}_{s,e}, \boldsymbol{\nu}_{s,e}) = 0, \quad \forall s \in \mathcal{S}, e \in \mathcal{E}; \quad (13.6)$$

Primal and Dual Feasibilities of KKT (Optimality Cuts):

$$\mathbf{F}(\hat{\mathbf{x}}_{s,e}^{\text{ev}}, \mathbf{y}_{s,e}^{\text{gv}}) = 0, \quad \forall s \in \mathcal{S}, e \in \mathcal{E}; \quad (13.7)$$

$$\mathbf{G}(\hat{\mathbf{x}}_{s,e}^{\text{ev}}, \mathbf{y}_{s,e}^{\text{gv}}, \hat{\mathbf{w}}_s^\diamond, \hat{\mathbf{r}}_s^\diamond) \leq 0, \boldsymbol{\nu}_{s,e} \geq 0, \quad \forall s \in \mathcal{S}, e \in \mathcal{E}; \quad (13.8)$$

Complementary Slackness of KKT (Optimality Cuts):

$$\boldsymbol{\nu}_{s,e} \perp \mathbf{G}(\hat{\mathbf{x}}_{s,e}^{\text{ev}}, \mathbf{y}_{s,e}^{\text{gv}}, \hat{\mathbf{w}}_s^\diamond, \hat{\mathbf{r}}_s^\diamond), \quad \forall s \in \mathcal{S}, e \in \mathcal{E}; \quad (13.9)$$

Objective Cuts (Optimality Cuts):

$$\mathbf{b}^\top \mathbf{x}_s^{\text{dv}} + \mathbf{c}^\top \mathbf{y}_s^{\text{dv}} \leq \mathbf{b}^\top \hat{\mathbf{x}}_{s,e}^{\text{ev}} + \mathbf{c}^\top \mathbf{y}_{s,e}^{\text{gv}}, \quad \forall s \in \mathcal{S}, e \in \mathcal{E}; \quad (13.10)$$

By linearizing the complementary slackness (13.9) via the Big-M method, **MP** can be converted into an equivalent MIP problem and directly solved by commercial MIP solvers.

D. Embedding RES-and-Reserve Tailor into UC as Add-On

Recall that the trained cost-oriented tailors, \mathcal{W}^* and \mathcal{R}^* , are linear functions (7). Hence, they can be embedded into the deterministic UC (1) as linear constraints (14.3), resulting in a prescriptive UC (14).

$$\min_{\mathbf{x}, \mathbf{y}, \hat{\mathbf{w}}^\diamond, \hat{\mathbf{r}}^\diamond} \mathbf{b}^\top \mathbf{x} + \mathbf{c}^\top \mathbf{y} \quad (14.1)$$

$$s.t. \mathbf{x}, \mathbf{y} \in \mathcal{X}(\hat{\mathbf{w}}^\diamond, \hat{\mathbf{r}}^\diamond); \quad (14.2)$$

$$\hat{\mathbf{w}}^\diamond = \mathcal{W}^*(\hat{\mathbf{w}}), \hat{\mathbf{r}}^\diamond = \mathcal{R}^*(\hat{\mathbf{r}}); \quad (14.3)$$

The prescriptive UC features that: *i*) without introducing extra computation burden, the new linear constraints (14.3) can tailor the raw predictions into their cost-oriented counterpart in favor of the UC economics; and *ii*) the objective (14.1) strictly obeys the least-cost principle, thus (14) is compatible with the predict-then-optimize practice, as illustrated in Fig. 1.

Finally, system operators can solve the prescriptive UC model (14), instead of the deterministic UC model (1), to boost the UC economics. This is equivalent to using \mathcal{W}^* and \mathcal{R}^* to tailor the raw predictions into cost-oriented predictions and solve the corresponding deterministic UC (1).

IV. CASE STUDIES

A. Experimental Setting

1) A Weekly-Rolling Training for Tailors: To leverage compatibility with the operators' practice and ensure solution quality of (14), the tailors are re-trained on a weekly basis to update (14) based on the past NT days. Specifically, to study the dispatch week of days $(D+1)$ to $(D+7)$, the past NT days (i.e., days $(D-NT)$ to $(D-1)$) are selected to build the training dataset \mathcal{S} . Then, the ERM (6) is solved to provide the cost-oriented tailors, \mathcal{W}^* and \mathcal{R}^* , for the targeted week. At the end of this period, the above process is repeated to update the tailors for the next week, i.e., days $(D+8)$ to $(D+14)$.

TABLE II
SELECTIVE WEEKS FOR OUT-OF-SAMPLE TESTING

Quarter 1, 2020	Quarter 2, 2020	Quarter 3, 2020	Quarter 4, 2020
Feb 04 - Feb 10	Jun 28 - Jul 04	Aug 20 - Aug 26	Dec 12 - Dec 18

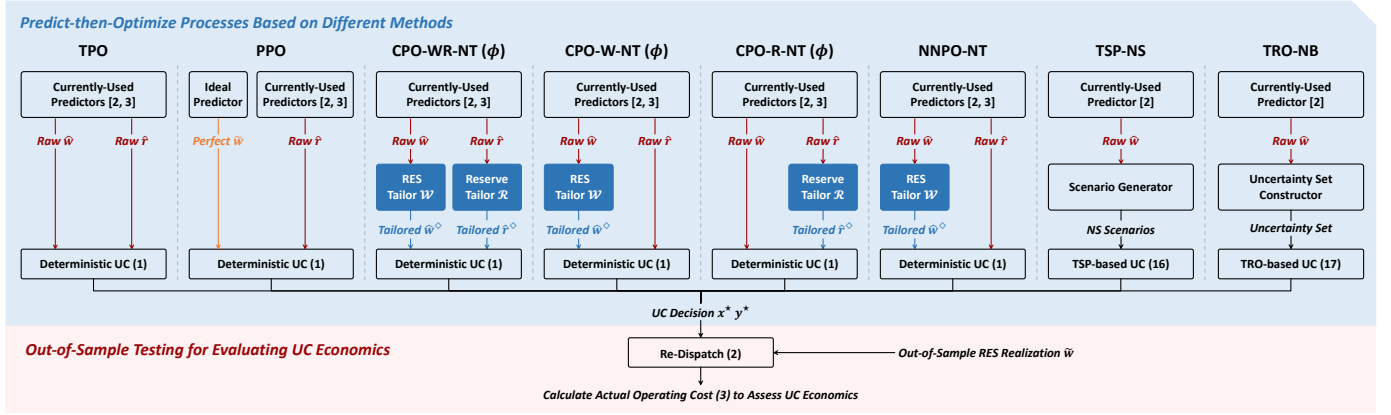


Fig. 5. The methods to be analyzed and the comparison scheme.

2) *Data Setup*: Data, including raw predictions and actual realizations, are collected from the data platform of a Belgian system [2]. Based on the data, four weeks in 2020, as listed in Table II, are selected, representing peak RES availabilities in the four quarters. The data are scaled to build the RES and load data for the testing systems. The slack penalty prices are all set as \$2,000/MWh. The rule-of-thumb reserve requirement is set according to the CAISO practice [3]: the system reserve requirement, represented as a proportion α of the total system load, is met by non-RES units, of which at least 50% is SR.

3) *Methods to be Compared*: Various methods are compared and analyzed to discuss the advantages and disadvantages of the presented framework. Fig. 5 shows the overall comparison strategy, which follows the evaluation process introduced in Section II-B. Note that *NT* denotes the number of training samples. The methods are introduced as follows:

- *TPO*: Traditional predict-then-optimize. It uses raw predictions as inputs to (1), describing the current practice of system operators;
- *PPO*: Perfect TPO. It uses error-free RES predictions and raw reserve requirements, representing the best UC economics an operator can achieve;
- *CPO-WR-NT*: TPO enhanced with the cost-oriented RES and reserve tailors. This is the presented method;
- *CPO-W-NT*: TPO enhanced with the cost-oriented RES tailor only—a variant of CPO-WR-NT. It tailors RES predictions but keeps the raw reserve requirements;
- *CPO-R-NT*: TPO enhanced with the cost-oriented reserve tailor only—a variant of CPO-WR-NT. It keeps raw RES predictions but tailors the raw reserve requirements;
- *CPO-NT- ϕ* : The same approach as CPO-NT, except that the binary requirement of lower-level operation models, (6.4) and (6.5), are relaxed as in (21) and (22). Parameter ϕ is pre-determined and falls within $[0, 1]$, representing the “relaxing degree”: $\phi = 1$ means a 100% relaxing degree—the lower-level models become LP; $\phi = 0$ means no (0%) relaxation. A larger ϕ indicates a higher relaxing degree of the binary requirement, i.e., the binary variables are allowed to take more fractional values between 0 and 1. Appendix C details this method;
- *NNPO-NT* [24]: TPO enhanced with an NN-based cost-

oriented RES tailor. NNPO-NT is trained using the same samples as those of CPO-W-NT. Appendix D details this method;

- *TSP-NS* [6]: Within the TPO framework, the deterministic UC model (1) is augmented to its TSP counterpart. The first stage models UC, while the second stage simulates RD (w.r.t. the first-stage UC variables) in *NS* net-load scenarios. Thus, the induced UC decisions can adapt to net-load realizations in the *NS* scenarios;
- *TRO-NB* [7]: Within the TPO framework, the deterministic UC model (1) is augmented to its TRO counterpart. The first stage models UC, while the second stage simulates RD (w.r.t. the first-stage UC variables) in the worst-case net-load scenario. *NB* denotes the budget to handle the worst case.

The threshold on the optimality gap of all methods is set as 1%. For CPO-WR-NT, the hyper-parameters, λ^w and λ^r , are set as 10^6 . This is because the magnitude of the ERM objective (8.1) is between 10^5 and 10^6 , so a hyper-parameter of 10^5 would pose a sufficient, but not overly dominating, effect on the ERM. All cases are solved by Gurobi 9.1, called by YALMIP on MATLAB, on a 3.5 GHz PC.

B. CPO vs TPO

Based on an IEEE 14-bus system (with $\alpha = 50\%$, 1,700MW non-RES capacity, 5 RES units, and 400MW RES capacity), this subsection compares CPO and TPO via economics improvement (EI) (15) and value of tailoring (VoT) (16).

$$EI = \frac{c^{\text{act}, \text{TPO}} - c^{\text{act}, \text{CPO}}}{c^{\text{act}, \text{TPO}}} \times 100\% \quad (15)$$

$$\text{VoT} = \frac{c^{\text{act}, \text{TPO}} - c^{\text{act}, \text{CPO}}}{c^{\text{act}, \text{TPO}} - c^{\text{act}, \text{PPO}}} \quad (16)$$

1) *Results*: Fig. 6 shows that six out of the seven CPO-WR-NTs, with *NT* ranging from 2 to 7, outperform TPO economically. Notably, CPO-WR-7 renders the best performance—an average EI of 2.54%. As the ideal PPO with the 1.00 VoT only achieves a 4.43% EI, the 2.54% EI of CPO-WR-7 can be regarded as non-insignificant. In fact, the 2.54% EI corresponds to a considerable monthly cost reduction of \$171,360 compared to TPO.

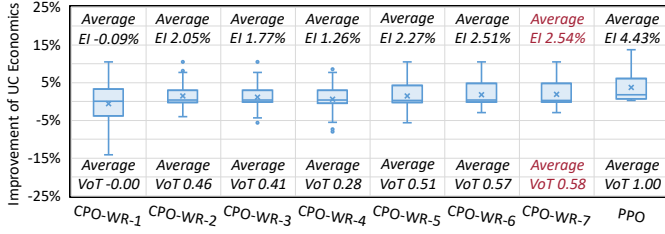


Fig. 6. Out-of-sample UC economics of seven CPO-WR-NTs.

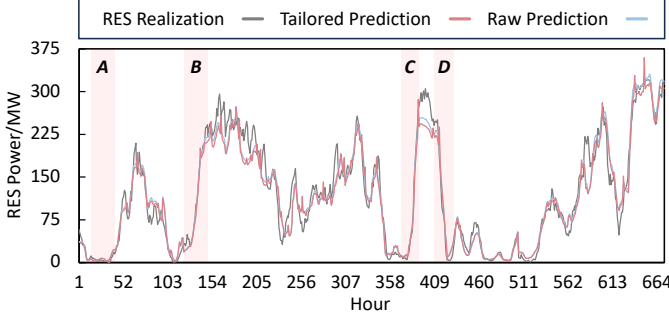


Fig. 7. Raw (from TPO) and tailored (from CPO-WR-7) RES predictions.

TABLE III
BREAKDOWN OF OUT-OF-SAMPLE COSTS OF TPO AND CPO-WR-7

	Actual UC Cost/10 ³ \$		Balancing Cost/10 ³ \$		$c^{\text{act}}/10^3\$$
	Startup	No-Load	Startup plus No-Load	Gen.	
TPO	1.1	2.1	0.2	237.9	241.3
CPO-WR-7	1.0	2.2	0.2	231.9	235.3

Table III breaks down the actual operating costs of TPO and CPO-WR-7. The main difference between TPO and CPO-WR-7 stems from the generation cost (the 5th column) in RD. It is also noteworthy that CPO-WR-7 has a lower startup cost and a higher no-load cost in UC.

Table IV compares six CPO-NTs. Note that both CPO-WR-NT and CPO-R-NT render lower SR and NR than the raw schedules of TPO, among which SR is always higher than NR. Regarding the training, both the computational time and number of iterations for solving the ERM problem increase with NT. Moreover, it is noteworthy that four CPO-NTs cause negative EI.

Table V evaluates RES prediction accuracy via five metrics, including mean absolute error (MAE), root mean square error (RMSE), MAPE, mean over-prediction percentage error (MOPE), and mean under-prediction percentage error (MUPE). Compared to TPO, CPO-WR-7 has a better MAPE but worse MAE and RMSE. Moreover, CPO-WR-7 has a better MOPE but a worse MUPE, indicating that CPO-WR-7 tends to conservatively predict RES power (i.e., vector \hat{w}^\diamond is statistically smaller than vector \hat{w}).

Finally, Fig. 7 compares raw and tailored RES predictions (including 672 hours) of the four weeks, where two observations are noteworthy. First, the predictions after tailoring are not significantly different from raw predictions. This is because raw predictions are already of high accuracy, especially when RES is scarce (e.g., area A) or the trend is stable

TABLE IV
COMPARISON OF DIFFERENT CPO-NTs

	Scheduled Reserve		Training		EI/%
	SR/MW	NR/MW	Time/s	Iterations	
TPO	113.2	113.2	-	-	0.00
CPO-WR-1	110.8	75.6	19	2	-0.09
CPO-WR-7	92.9	65.4	786	6	2.54
CPO-W-1	113.2	113.2	21	2	-0.22
CPO-W-7	113.2	113.2	750	3	-0.02
CPO-R-1	98.8	85.8	13	2	-0.48
CPO-R-7	88.8	63.3	1,857	4	1.77

TABLE V
OUT-OF-SAMPLE PREDICTION ACCURACY OF TPO AND CPO-WR-7

	MAE	RMSE	MAPE	MOPE	MUPE
Raw \hat{w}	15.0MW	21.0MW	36.4%	29.6%	6.8%
Tailored \hat{w}^\diamond	16.5MW	23.2MW	35.9%	27.7%	8.2%

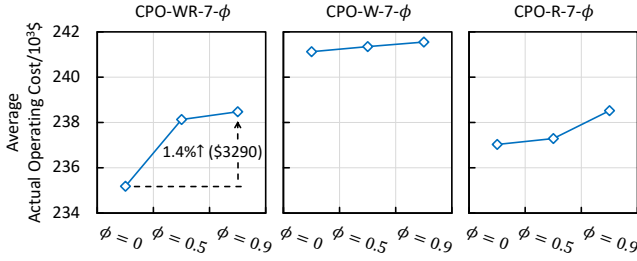
(e.g., areas B, C, and D), so there is no need to tailor them dramatically. Indeed, most elements of the vector \mathbf{m} in (7) are between 0.9 and 1.1. Second, 86.9% of tailored predictions are lower than their raw predictions, and the tailoring causes a larger deviation from the realizations.

2) *Discussions*: The above results have indicated that CPO-WR-NT can outperform TPO on the UC economics. Still, the reasons behind this conclusion deserve further discussion.

Table III shows that the cost difference mainly stems from the generation. This is because CPO-WR-7 tends to use units with cheaper generation costs (e.g., large-capacity units). Although these units have higher no-load costs, CPO-WR-7 keeps them online long enough by setting predictions \hat{w}^\diamond and \hat{r}^\diamond properly. As a result, CPO-WR-7 uses fewer units in RD and operates them most efficiently.

According to Table IV, the results of CPO-WR-NT and CPO-R-NT reveal that a proper reserve deployment should have a higher SR and a lower NR, instead of the rule-of-thumb 50%-50% as in TPO. This is because the bi-level ERM (6) enables CPO-NT to learn that triggering quick-start units in RD is expensive. Moreover, although reserve requirements of CPO-NT are lower than those of TPO in general, our experiments show that: *i*) reserve requirements of CPO-WR-7 are 5.28% higher than those of TPO in quarter 2, indicating that the rule-of-thumb \hat{r} could be still insufficient in certain cases; and *ii*) CPO-WR-NT leverages lower \hat{w}^\diamond to induce higher base-generation P_{it}^* in the day-ahead UC, allowing system operators to possess enough capacity and avoid load shedding when faced with \tilde{w} in the RD. Overall, this is how CPO-NT leverages the cost-oriented prediction to improve the UC economics. Last but not least, two more observations are implied by Table IV:

- The negative EI indicates that bi-level learning may worsen the UC economics. Indeed, the tailors possess a common issue in wide ML applications—they cannot guarantee that learning will always bring in extra benefits. Nevertheless, this issue could be relieved with sufficient training samples and well-tuned hyper-parameters;

Fig. 8. Out-of-sample actual operating costs of different CPO-NT- ϕ .

- Although Table IV indicates that the EI monotonically increases with NT , Fig. 6 shows that this relationship may not hold at a 1-sample grain. Two possible reasons are responsible for this fluctuation: *i*) a single tailor is used throughout one week, regardless of weekdays and weekends; and *ii*) the number of training samples is limited. To relieve the fluctuation, the first approach is to train different tailors for weekdays and weekends, and the second approach is to increase NT —the best performance is generally achieved when NT reaches about ten times the number of parameters of the tailor. However, the two approaches would significantly increase the training time. Thus, operators should weigh up the pros and cons according to the computation budget.

Table V indicates that tailored predictions have worse MAE and RMSE. This is because the tailors evaluate the prediction quality via the actual operating cost c^{act} instead of traditional accuracy metrics. Moreover, it is interesting to observe that \hat{w}^\diamond outperforms \hat{w} in MAPE. This is because \hat{w}^\diamond generally under-predicts RES, hence the MAPE of \hat{w}^\diamond is smaller than that of \hat{w} in the cases of scarce RES. However, these cases have a limited impact on both MAE and RMSE, because they are relatively insensitive to small values. To this end, Table V implies a fact—assessing prediction quality with a single accuracy metric may be myopic as each metric owns a specific property. *To this end, this paper suggests evaluating the prediction quality with the ultimate operation goal.*

In summary, by leveraging the bi-level MIP structure to learn the UC-RD operation information, the RES-and-reserve tailor can customize raw $\{\hat{w}, \hat{r}\}$ to cost-oriented $\{\hat{w}^\diamond, \hat{r}^\diamond\}$ for improving UC economics. Although the tailored predictions are slightly worse in certain accuracy metrics, they enable a more economic operation for systems. Thus, by assessing the prediction performance via the UC economics, it is direct to conclude that tailored $\{\hat{w}^\diamond, \hat{r}^\diamond\}$ significantly outperforms raw $\{\hat{w}, \hat{r}\}$. The reduction in prediction accuracy, as listed in Table V, shall not be considered a compromise, because the ultimate goal of the prediction is to improve the optimization performance instead of simply improving the prediction accuracy. More importantly, the tailors are deployed as add-ons to the operators' accuracy-oriented predictors and thus are compatible with their current practice.

C. CPO vs CPO- ϕ

The binary requirement constraints (i.e., (19.16) and (20.14)) are crucial in the operation models. To illustrate the

TABLE VI
RELATIONSHIP BETWEEN RELAXING DEGREE AND ERM OBJECTIVE VALUE (IN-SAMPLE TRAINING RESULTS OF QUARTER-4 WEEK)

	Objective Value of ERM (6)/10 ⁵ \$		
	CPO-WR-7- ϕ	CPO-W-7- ϕ	CPO-R-7- ϕ
$\phi = 0$ (0% Relaxation)	3.16	3.56	3.15
$\phi = 0.9$ (90% Relaxation)	2.15	2.16	2.15

TABLE VII
RELATIONSHIP BETWEEN RELAXING DEGREE AND TRAINING TIME (IN-SAMPLE TRAINING RESULTS OF QUARTER-4 WEEK)

	Training Time/s		
	CPO-WR-7- ϕ	CPO-W-7- ϕ	CPO-R-7- ϕ
$\phi = 0$ (0% Relaxation)	70.81	200.52	81.20
$\phi = 0.9$ (90% Relaxation)	35.11	37.63	25.63

impact of these constraints, this subsection compares CPO-NT- ϕ with different settings of the relaxing degree ϕ .

1) *Results*: Fig. 8 compares the average actual operating costs (over the four weeks in Table II) of $\phi = 0$, $\phi = 0.5$, and $\phi = 0.9$. It is observed that the actual operating cost increases with ϕ , especially when the reserve requirement tailor is involved (i.e., CPO-WR-7- ϕ and CPO-R-7- ϕ in Fig. 8). The largest cost increment reaches 1.4% between CPO-WR-7-0 and CPO-WR-7-0.9, representing a daily cost increment of \$3,290. These observations indicate that tightening the binary requirement in the in-sample operation models (i.e., (6.4) and (6.5)) can enhance the out-of-sample performance of tailors.

2) *Discussions*: The binary requirement is pivotal for ensuring the clear physical meaning of decision variables within the operation models. Considering a relaxed UC model with $\phi = 0.9$, to achieve a lower objective value (19.1), constraint (19.9) allows on-off status variable I_{it} and NR commitment variable O_{it} to be non-zero simultaneously. That is, it allows unit i to provide both SR and NR for hour t , which violates the physical operation rules.

Table VI further reports the ERM objective value (6.1) (in-sample actual operating cost) of the quarter-4 training, showing that the costs of $\phi = 0.9$ are significantly lower than those of $\phi = 0$. However, as explained above, when $\phi = 0.9$, many of the in-sample operation decisions could be infeasible in practice. As a result, such compromised in-sample operation decisions will adversely affect the out-of-sample performance of tailors. Moreover, our experimental results report that such adverse effects will be further amplified when the hyper-parameters, λ^w and λ^r , take smaller values. For example, when λ^w and λ^r decrease from 10⁵ to 10², the difference in actual operating costs between CPO-WR-7-0 and CPO-WR-7-0.9 increases from 1.4% to 8.1%.

Finally, Table VII indicates that relaxing the binary requirement may accelerate the training process, because solving **MP** (13) becomes faster. Therefore, from the perspective of operators, even though rigorously incorporating binary variables in the training is beneficial in reducing the actual operating cost, the associated long training time will necessitate a trade-off between the out-of-sample performance and training time, especially when the computation resources are limited.

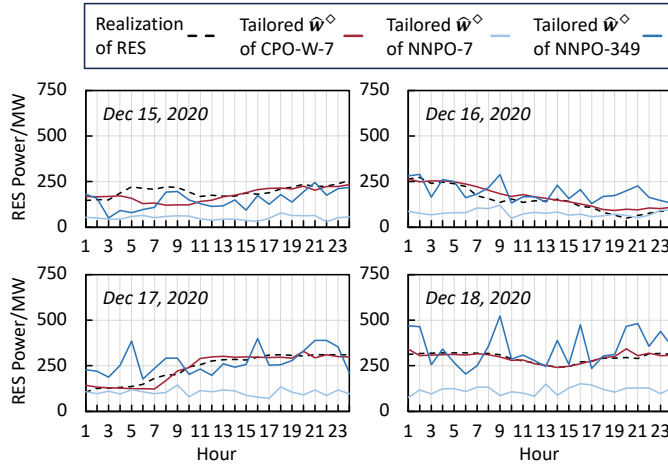


Fig. 9. Tailored RES predictions of CPO-W-7 and two NNPO-NTs.

D. CPO vs NNPO

The underlying idea of NNPO-NT [24] is to build a differentiable function to approximate the relationship between the statistical error of a *single* prediction factor (e.g., RMSE of RES predictions) and the actual operating cost, which is then used as the loss function to train NNs.

To facilitate the comparison, the settings below are adopted:

- The loss function of day D is approximated by NT scenarios in day $D - 1$;
- NNPO-NT uses tailored RES prediction and keeps raw reserve requirements. This is because NNPO-NT cannot be directly applied to tailor multiple prediction factors if these factors affect each other mutually. Therefore, for the sake of fairness, NNPO-NT is compared to CPO-W-NT;
- Dec 15-18, 2020, the last four days in Table II, are used for comparison. Hence, up to 349 training samples (from Jan-1 to Dec-14, 2020) are available for NNPO-NT;
- Two NNPO-NTs—NNPO-7 and NNPO-349—are trained for comparison, in which NNPO-7 uses the same training samples as CPO-W-7;
- Note that randomness exists in the NN training due to certain heuristic steps. Thus, NNPO-NT is evaluated statistically. Taking NNPO-7 as an example, the evaluation process involves three steps: *i*) training NNPO-7 ten times, which yields ten NNPO-7s; *ii*) conducting the out-of-sample test for each of the ten NNPO-7s; and *iii*) calculating the mean, standard deviation, and confidence interval (CI) of their out-of-sample results, thus providing a statistical evaluation of NNPO-7.

1) *Results*: Fig. 9 compares the tailored RES predictions, showing that: *i*) NNPO-7 tends to under-predict, and its curves are fairly flat; *ii*) compared to NNPO-7, the curves of NNPO-349 are higher and more volatile; and *iii*) the curves of CPO-W-7 are less volatile and closer to the actual realization.

Table VIII lists the actual UC cost and balancing cost, showing that CPO-W-7 falls between NNPO-7 and NNPO-349 regarding the mean actual UC cost. This phenomenon can be explained by Fig. 9—NNPO-7 tends to tailor the raw \hat{w} to be a flat and under-predicted \hat{w}^\diamond , while NNPO-349 generally produces a \hat{w}^\diamond involving over-predicted spikes. Consequently,

TABLE VIII
OUT-OF-SAMPLE COSTS OF CPO-W-7 AND TWO NNPO-NTs

		Actual UC Cost/ $10^3\$$		Balancing Cost/ $10^3\$$	
		Mean	Std. Dev.	Mean	Std. Dev.
Dec 15	CPO-W-7	3.1	0.0	223.1	0.0
	NNPO-7	4.2	0.4	264.9	5.8
	NNPO-349	3.8	0.2	231.6	2.4
Dec 16	CPO-W-7	3.6	0.0	250.9	0.0
	NNPO-7	4.1	0.3	284.3	8.0
	NNPO-349	3.5	0.3	256.1	3.2
Dec 17	CPO-W-7	2.9	0.0	190.8	0.0
	NNPO-7	3.5	0.2	228.3	4.5
	NNPO-349	2.8	0.2	198.0	3.4
Dec 18	CPO-W-7	2.9	0.0	146.6	0.0
	NNPO-7	3.3	0.0	210.5	8.2
	NNPO-349	2.7	0.2	161.1	5.2

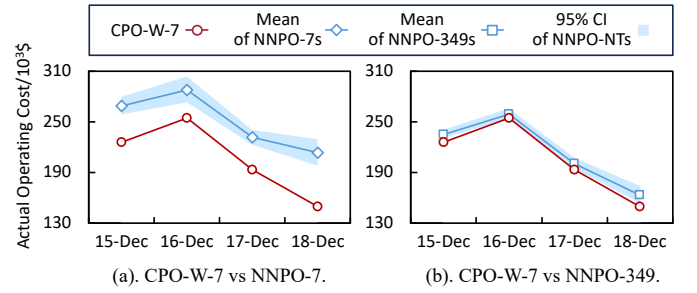


Fig. 10. Out-of-sample operating costs of CPO-W-7 and two NNPO-NTs.

NNPO-7/NNPO-349 turns on more/fewer units in the day-ahead stage, causing higher/lower actual UC costs. Although the differences in the actual UC cost seem minor, they affect the balancing cost considerably. For example, on Dec 18, 2020, the mean actual UC cost of NNPO-7 is only $\$0.4 \times 10^3$ higher than that of CPO-W-7, but the difference it causes in the mean balancing cost is as high as $\$63.9 \times 10^3$.

Table VIII also indicates that balancing costs of NNPO-NT have noticeable standard deviations, NNPO-7 in particular. This is because the gradient-based NN training involves certain heuristic steps and can only provide a locally optimal tailor, i.e., it cannot guarantee a consistent training result. In contrast, CPO-W-7 does not exhibit such noticeable standard deviations, as its training relies on mathematical programming that can consistently approach a globally optimal predictor.

Fig. 10(a) shows that using the same 7 training samples, CPO-W-7 significantly outperforms NNPO-7 on the actual operating cost. This finding indicates that CPO-W-7 is more effective than NNPO-7 when training samples are limited—a benefit attributed to the bi-level structure of the ERM problem (6). Furthermore, Fig. 10(b) shows that compared to NNPO-7, NNPO-349 has a lower mean and a narrower CI for the actual operating cost. However, CPO-W-7 still economically outperforms NNPO-349.

Overall, the above results have clearly demonstrated that, even though the number of training samples is limited, CPO can still learn to properly tailor the raw \hat{w} . The tailored \hat{w}^\diamond leads to a moderate actual UC cost and a low balancing cost, ultimately resulting in a low actual operating cost.

2) *Discussions*: To facilitate a more in-depth comparison between CPO and NNPO, their differences and the underlying reasons are further discussed.

Performances with Few Training Samples: Our experiments report that NNPO-7 performs poorly in both in-sample and out-of-sample losses—a clear sign of under-fitting. This is mainly due to the existence of “bad” training samples. With only 7 training samples, even one or two poor-quality samples could noticeably mislead the training and worsen the generalizability of NNPO-7. Our experiments also report that although NNPO-7 can be improved by removing the “bad” samples, it remains economically worse than CPO-W-7.

In comparison, CPO-W-7 exhibits robustness against the “bad” training samples. This advantage primarily arises from the bi-level structure of (6). Within this structure, the non-anticipativity between the prediction, UC, and RD stages can be ensured, and the UC and RD processes can be modeled uncompromisingly. As a result, such an ERM model can accurately express the predict-then-optimize practice—sequentially conduct predictions, UC optimization (1), and RD optimization (2) with the information at hand (i.e., non-anticipativity). Consequently, while tailors trained via the bi-level ERM might not show a considerably low in-sample loss, they can still achieve good out-of-sample performance.

Tuning Burdens of Hyper-Parameters: The high performance of NNPO can be achieved by tuning hyper-parameters finely. However, tuning a vast array of hyper-parameters on a daily basis is challenging. Indeed, most commercial NN-based predictors, once delicately tuned, rarely undergo frequent re-tuning processes due to significant computational burdens. As compared, CPO only requires tuning the two hyper-parameters, λ^w and λ^r . Thus, from the perspective of hyper-parameter tuning, CPO is more user-friendly.

Interpretability of Tailored Predictions: It is noteworthy that the NN-based tailor may yield *negative* RES predictions. This issue, together with NNPO-349’s spikes observed in Fig. 9, could raise operators’ concerns: *i*) dramatically tailoring the high-quality raw predictions lacks interpretability; and *ii*) the spikes would trigger the needs of extra flexibility resources. Indeed, these issues stem from the fact that the NN training ignores critical physical information (e.g., ramping and generation limits) within the UC and RD models. That is, the training exclusively aims to minimize the training loss, regardless of physical principles, e.g., RES predictions shall be non-negative.

Regarding this aspect, CPO possesses two valuable properties: *i*) the linear form (7) offers strong interpretability; and *ii*) the bi-level ERM (6) effectively integrates the UC-RD process into the training. Within this ERM, irrational predictions (e.g., spikes and negative values) will cause extreme costs and even infeasibility in the lower-level UC and/or RD, thus being excluded. As a result, tailors trained by the physically-informed ERM barely produce irrational predictions.

Form of Tailors’ Model: NNPO possesses the ability to capture complex non-linear relationships. As compared, the presented tailors (7) can only scale raw predictions linearly. Nevertheless, for daily UC practice, tailors with simpler forms may be more appropriate. This is because operators have al-

ready utilized sophisticated models like NNs to produce high-quality raw predictions [2]; thus, only moderate fine-tuning instead of dramatic modification would be needed. Indeed, our extensive experiments show that NNPO performs better with linear activation than with ReLU, sigmoid, softmax, and tanh activations. This observation implies that straightforward tailor models could be more effective when high-quality raw predictions are used as inputs.

E. CPO vs TSP and TRO Models with MIP Recourse

CPO-WR-NT, TSP-NS, and TRO-NB are further compared using a modified IEEE 118-bus system (with 8,600MW non-RES capacity and 1,000MW RES capacity). For the sake of comparison, the following settings on TSP-NS (17) and TRO-NB (18) models with MIP recourse are adopted:

- The reserve requirement constraints (19,19) are not explicitly included in the two-stage models (17)-(18), because the reserve schedules have been implicated in their second-stage RD. To enable the first-stage UC to deploy enough reserve, not only RES but also load are regarded as uncertainty factors. The day-ahead raw point predictions (denoted as $\hat{\mathbf{u}}$) and their corresponding confidence intervals are provided by the Belgian system [2];
- TSP-NS is modeled as a scenario-based model (17). First, 3,000 scenarios are generated via Latin hypercube sampling within the 90% confidence intervals. And then, these 3,000 scenarios are reduced to NS scenarios via a scenario-tree method [51]. For scenario h , $\tilde{\mathbf{u}}_h^{\text{scan}}$ denotes its realization, and p_h denotes its probability. Model (17) is directly solved by Gurobi;

$$\min_{\mathbf{x}, \mathbf{y}, \mathbf{z}} \mathbf{b}^\top \mathbf{x} + \mathbf{c}^\top \mathbf{y} + p_h \sum_{h=1}^{NS} \mathbf{d}^\top \mathbf{z}_h \quad (17.1)$$

$$s. t. \mathbf{x}, \mathbf{y} \in \mathcal{X}(\hat{\mathbf{u}}); \quad (17.2)$$

$$\mathbf{z}_h \in \mathcal{Z}(\mathbf{x}, \mathbf{y}, \tilde{\mathbf{u}}_h^{\text{scan}}), h = 1, \dots, NS; \quad (17.3)$$

- TRO-NB, as in (18), uses an adjustable budget parameter $\Gamma \in [0, 48]$ to control the solution robustness, in which $\Gamma = NB$. The box uncertainty set \mathcal{U} is built using the 90% confidence intervals, where the worst-case realization of the uncertainty is denoted as $\tilde{\mathbf{u}}^{\text{wcr}}$. A nested C&CG method [7] is used to solve (18).

$$\min_{\mathbf{x}, \mathbf{y}} \mathbf{b}^\top \mathbf{x} + \mathbf{c}^\top \mathbf{y} + \max_{\tilde{\mathbf{u}}^{\text{wcr}} \in \mathcal{U}} \min_{\mathbf{z}} \mathbf{d}^\top \mathbf{z} \quad (18.1)$$

$$s. t. \mathbf{x}, \mathbf{y} \in \mathcal{X}(\hat{\mathbf{u}}); \quad (18.2)$$

$$\mathbf{z} \in \mathcal{Z}(\mathbf{x}, \mathbf{y}, \tilde{\mathbf{u}}^{\text{wcr}}, \Gamma); \quad (18.3)$$

1) *Results of CPO vs TSP-NS*: Fig. 11(a) sketches the UC economics of seven CPO-WR-NTs and seven TSP-NSs, leading to the following three observations:

- TSP-NS requires many scenarios to perform well, leaving a wide cost gap between TSP-1 and TSP-99. Relatively, CPO-WR-NT needs fewer samples, with a narrower cost gap between CPO-WR-1 and CPO-WR-7. This point means that CPO can leverage training samples more effectively, ensuring its performance in the case of few training samples;

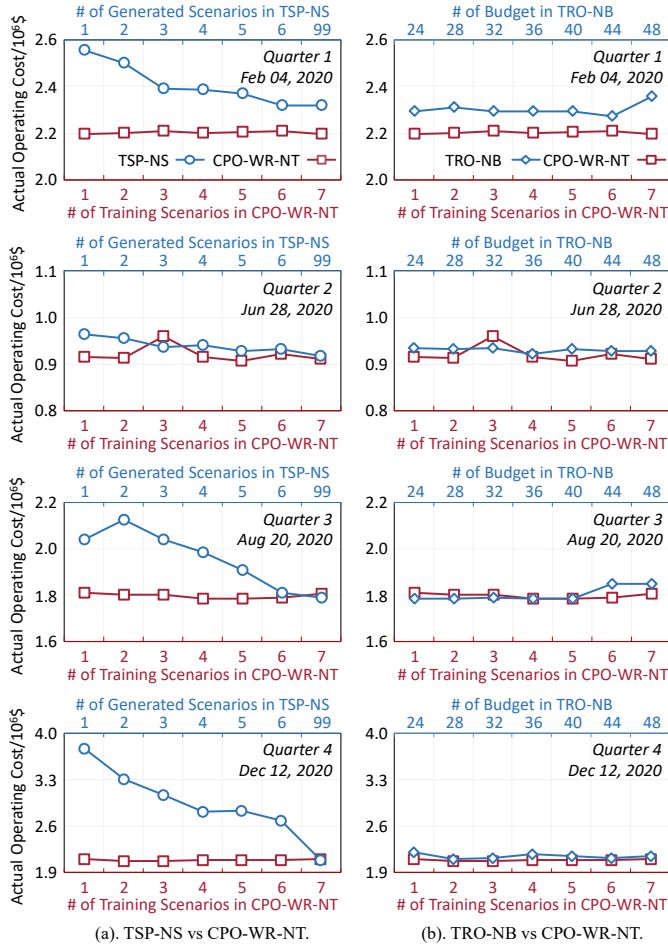


Fig. 11. Comparison of CPO-WR-NT (red) and TSP-NS/TRO-NB (blue).

- Except for quarter 1, the actual operating cost of TSP-NS and CPO-WR-NT eventually intersects, implying that the performances of TSP-NS and CPO-WR-NT are mutually reachable with suitable NS/NT ;
- In quarter 1, even with $NS = 99$, TSP-NS still performs poorly. This is because the confidence intervals cannot exactly cover the underlying distributions of \tilde{u} , and TSP-NS would encounter significant slack penalties.

Table IX breaks down the actual operating cost of the quarter-2 day. It shows that the three TSP-NSs bring lower startup and no-load costs. This is because the reserve requirements (19.19) are implicated in (17) rather than being explicit as in (14). Thus, the three TSP-NSs turn on fewer units and schedule less reserve in the UC stage, causing relatively more extensive usage of quick-start units in the RD stage. Indeed, our results show that the average hourly reserve schedules (SR plus NR) of TSP-1 and TSP-99 are, respectively, 89.9MW and 207.2MW. In comparison, these average schedules are 1,265.8MW and 1,286.8MW in CPO-WR-1 and CPO-WR-7. As quick-start units have limited ramping ranges, they are operated away from the most efficient points in RD, causing higher generation costs in TSP-NS.

Indeed, the difference in reserve schedules implies that TSP-NS and CPO-WR-NT achieve similar economics in different ways. Specifically, TSP-NS schedules fewer reserves and uses

TABLE IX
BREAKDOWN OF ACTUAL OPERATING COST ON 28 JUN 2020

	Actual UC Cost/ $10^3\$$		Balancing Cost/ $10^3\$$		
	Startup	No-Load	Startup plus No-Load	Gen.	Slack
CPO-WR-7	6.7	9.0	0.0	894.9	00.0
TSP-1	6.5	8.1	1.1	934.3	13.7
TSP-4	5.1	7.7	1.0	926.4	00.0
TSP-99	4.8	7.8	0.7	904.7	00.0

TABLE X
BREAKDOWN OF ACTUAL OPERATING COST ON 20 AUG 2020

	Actual UC Cost/ $10^3\$$		Balancing Cost/ $10^3\$$		
	Startup	No-Load	Startup plus No-Load	Gen.	Slack
CPO-WR-7	5.9	12.6	0.2	1789.7	0
TRO-24	6.5	11.3	1.2	1764.9	0
TRO-36	6.7	11.4	1.2	1764.3	0
TRO-48	6.5	11.3	1.2	1764.9	63.5

more RES curtailments in place of SR in the RD. This may work well in case of ample RES but becomes economically risky when the actual RES availability is low, as expensive slack penalties will be triggered (e.g., due to insufficient capacities, TSP-1 is penalized as in Table IX). This may also cause a lower RES utilization.

2) *Results of CPO vs TRO-NB*: Fig. 11(b) sketches the economics of CPO-WR-NT and TRO-NB, showing that both methods are relatively stable—the gap between the best and worst costs is small. Moreover, multiple intersections exist in the curves, implying that TRO-NB and CPO-WR-NT can reach similar performance by tuning parameters properly.

Table X further compares the costs on the quarter-3 day. It shows that TRO-NB has a higher startup cost but a lower no-load cost in the UC stage. The reason is that TRO-NB schedules SR for those hours of potential large volatility, so that multiple units in the UC stage are turned on before these hours and turned off afterward. On the other hand, CPO-WR-7 constantly keeps more units online to meet the reserve requirements of individual hours, with higher no-load costs. In addition, the penalty of TRO-48 means that even a large Γ could lead to an insufficient reserve schedule. Indeed, the hourly average reserve schedules of TRO-48 and CPO-WR-7 are 278.4MW and 1,214.7MW, implying that CPO-WR-7 is more conservative. With this, it can be pointed out that CPO-WR-NT achieves economics similar to the state-of-the-art methods in a more reliable way—scheduling more reserves to avoid potential slack penalties.

3) *Discussions*: The above results have shown that CPO is comparable to TSP and TRO regarding UC economics. Although a clear-cut economic advantage of CPO has yet to be observed, it does offer two promising practical advantages.

Computational Affordability to System Operators: Computation time of the proposed CPO framework includes the training time of tailors and the solving time of UC tasks. As the tailors are trained offline and rather infrequently, their computational time is not considered a bottleneck for practical

TABLE XI
SOLVING TIME OF UC MODELS

TPO	CPO-WR-7	TSP-99	TRO-48
4.21s	4.29s	1,209.32s	3,082.86s

applications. In addition, the training time is significantly impacted by the efficiency of the codes, such as Python's TensorFlow library used in case studies. Therefore, this subsection focuses on the solving time of UC models.

The day-ahead UC to determine optimal operation plans has been regarded as one of the most computationally challenging tasks, even with the deterministic UC formulation (1). Table XI compares the solving time of UC of TPO (1), CPO-WR-7 (14), TSP-99 (17), and TRO-48 (18). TSP-99 and TRO-48 are much more computationally taxing than TPO. Although advanced accelerating techniques, e.g., parallel computation and decomposition methods, could accelerate the solving time of TSP and TRO, they still lag behind the deterministic UC [52]. This is one of the main reasons that most operators still use the deterministic UC (1). As compared, the prescriptive UC (14) of CPO keeps a computation burden similar to (1) of TPO, thus being more affordable to operators.

Compatibility with System Operators' Practice: In practice, system operators aim to identify the economically optimal UC decisions given the predictions. However, UC decisions of TSP/TRO may not fully follow the least-cost principle in the short term due to the treatment of uncertainties. Although better UC economics may be reached in the long term, a short-term financial deficit would be inevitable [44].

While pursuing to maximize system economics, operators must have a stably interpretable operation plan (e.g., which units to dispatch), so that both operation schedules and post-operation analyses can be rationally understood. Regarding this, TSP and TRO could expose intrinsic obstacles in practical applications. Indeed, for TSP, a stable schedule result requires sufficient scenarios, which, however, will further magnify the computational challenge; while for TRO, to assure a tractable reformulation, the uncertainty set is usually designed to be heavily parameter-dependent. These together cause volatility and poor interpretability of schedule results in TSP and TRO.

Specifically, the process of constructing scenario sets for TSP inevitably introduces randomness. However, different scenario sets, even with tiny differences, may lead to noticeable changes in commitment results, i.e., certain units are ON/OFF from one set to another. These issues adversely affect the interpretability of the schedule plan. Analogously, the setting of parameter Γ suffers from the same issue. This is because the impacts of different Γ on the schedule plan are difficult to interpret and quantify. Thus, the operators have to face the same issue on a daily basis—what is the best scenario set or budget parameter for the next operation day?

Last but not least, the difference between the scheduled reserves of CPO and TSP/TRO implies that the second stage of TSP/TRO may not be able to implicitly secure enough reserves in the UC plan, indicated by the non-zero slacking penalties in Tables IX and X. In other words, TSP and TRO may

not be as robust as expected against uncertainties, implying that the explicit expression (19.19) of reserve requirements would remain necessary for the UC model. Regarding this, CPO leverages the raw predictions of reserve requirements to achieve good robustness.

In summary, the prescriptive UC (14) of CPO has the same structure as the model (1) of TPO, thus following the least-cost principle, remaining strong interpretability, and being computationally efficient. Moreover, Fig. 11 has shown that the UC economics of TSP/TRO and CPO are mutually reachable. Thus, by selecting training samples properly, it is promising to harness the compatibility of CPO in bridging the gap between TSP/TRO and current practice.

V. CONCLUSION

To improve the existing predict-then-optimize practice and reduce the actual operating cost, this paper presents a CPO framework based on the bi-level MIP formulation. The idea is to train an add-on tailor that can customize raw predictions of RES and reserve requirements into cost-oriented predictions, thereby reducing the actual operating cost. Case studies compare the presented framework to five other frameworks in the literature, leading to the following conclusions:

- Compared to the existing predict-then-optimize practice, results over four selective weeks from different seasons demonstrate that the presented framework can reduce the average actual operating cost by up to 2.54%;
- Compared to the bi-level relaxed MIP method, the presented framework with an uncompromising binary requirement can achieve up to an 8.1% reduction in average actual operating costs;
- Compared to the NN-based CPO framework, the presented bi-level MIP-based CPO framework exhibits two advantages: *i*) it can reduce the actual operating cost even when the training samples are few; and *ii*) it has a relatively consistent and stable out-of-sample performance;
- The actual operating cost induced by the presented framework, TSP, and TRO could be mutually reachable. However, since the RES-and-reserve tailor acts as an add-on to the deterministic UC, the presented framework offers higher implementation values to the predict-then-optimize practice.

Future work could further consider other UC-related issues, including: *i*) how can the impacts of system topology changes (e.g., transmission contingency and transmission line switching) on tailor training be managed? *ii*) what strategies can be employed to effectively handle data drifts (e.g., seasonal and weekly load pattern changes)? and *iii*) how can the computational efficiency of the proposed solution approach be further enhanced to accommodate larger-scale systems?

APPENDIX

A. The Deterministic UC Model with Predictions

The UC objective (19.1) is to minimize the total system cost, including startup, no-load, and generation costs. The generator constraints include generation limits (19.2)-(19.3), segment-based generation representation (19.4)-(19.5), SR

(19.6) and NR (19.7)-(19.9) capacity limits, startup-shutdown-commitment status logic (19.10), minimum on and off requirements (19.11)-(19.12), ramping limits (19.13)-(19.14), and RES power limits (19.15) [53]. Constraint (19.16) is the binary requirement. System constraints include power balance (19.17), transmission limits (19.18) based on DC power flow-based function $\mathcal{F}(\cdot)$ [1], and reserve requirements (19.19). Note that \hat{L}_q denotes the load demand prediction of bus q .

$$\min_{\mathbf{x}, \mathbf{y}} \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} (C_i^{\text{su}} U_{it} + C_i^{\text{nl}} I_{it}) + \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} C_{ik}^{\text{sg}} P_{itk}^{\text{sg}} \quad (19.1)$$

where $\mathbf{x} = \{\mathbf{D}, \mathbf{I}, \mathbf{O}, \mathbf{U}\}$, $\mathbf{y} = \{\mathbf{P}, \mathbf{P}^{\text{sg}}, \mathbf{R}^{\text{sr}}, \mathbf{R}^{\text{nr}}, \mathbf{W}\}$

Generator Constraints:

$$P_{it} - R_{it}^{\text{sr}} \geq P_i^{\text{m}} I_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (19.2)$$

$$P_{it} + R_{it}^{\text{sr}} \leq P_i^{\text{m}} I_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (19.3)$$

$$P_{it} = \sum_{k \in \mathcal{K}} P_{itk}^{\text{sg}}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (19.4)$$

$$0 \leq P_{itk}^{\text{sg}} \leq \bar{P}_{ik}^{\text{sg}} I_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, k \in \mathcal{K}; \quad (19.5)$$

$$0 \leq R_{it}^{\text{sr}} \leq \bar{R}_i^{\text{sr}} I_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (19.6)$$

$$P_i^{\text{m}} O_{it} \leq R_{it}^{\text{nr}} \leq \bar{R}_i^{\text{nr}} O_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (19.7)$$

$$O_{it} = 0, \quad \forall i \in \mathcal{I}^{\text{ns}}, t \in \mathcal{T}; \quad (19.8)$$

$$O_{it} + I_{it} \leq 1, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (19.9)$$

$$U_{it} - D_{it} = I_{it} - I_{i,t-1}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (19.10)$$

$$\sum_{t'=t-T_i^{\text{su}}+1}^t U_{it'} \leq I_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}_i^{\text{su}}; \quad (19.11)$$

$$\sum_{t'=t-T_i^{\text{sd}}+1}^t D_{it'} \leq 1 - I_{it}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}_i^{\text{sd}}; \quad (19.12)$$

$$P_{it} - P_{i,t-1} \leq P_i^{\text{M}}(1 - I_{it})$$

$$+ R_i^{\uparrow} I_{i,t-1} + R_i^{\text{su}}(I_{it} - I_{i,t-1}), \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (19.13)$$

$$P_{i,t-1} - P_{it} \leq P_i^{\text{M}}(1 - I_{i,t-1})$$

$$+ R_i^{\downarrow} I_{it} + R_i^{\text{sd}}(I_{i,t-1} - I_{it}), \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (19.14)$$

$$0 \leq W_{jt} \leq \hat{W}_{jt}, \quad \forall j \in \mathcal{J}, t \in \mathcal{T}; \quad (19.15)$$

$$D_{it}, I_{it}, O_{it}, U_{it} \in \{0, 1\}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (19.16)$$

System Constraints:

$$\sum_{i \in \mathcal{I}} P_{it} + \sum_{j \in \mathcal{J}} W_{jt} = \sum_{q \in \mathcal{Q}} \hat{L}_{qt}, \quad \forall t \in \mathcal{T}; \quad (19.17)$$

$$-B_b \leq \mathcal{F}_b(\mathbf{P}_t, \mathbf{W}_t, \hat{\mathbf{L}}_t) \leq B_b, \quad \forall b \in \mathcal{B}, t \in \mathcal{T}; \quad (19.18)$$

$$\sum_{i \in \mathcal{I}} R_{it}^{\text{sr}} \geq \hat{R}_t^{\text{sr}}, \quad \sum_{i \in \mathcal{I}} (R_{it}^{\text{sr}} + R_{it}^{\text{nr}}) \geq \hat{R}_t^{\text{sr}} + \hat{R}_t^{\text{nr}}, \quad \forall t \in \mathcal{T}; \quad (19.19)$$

B. The Re-Dispatch Model with Realizations

In the RD model (20), those variables with superscript \star are optimal variables provided by solving (19). The objective (20.1) is to minimize startup and no-load costs of quick-start units that are not committed in UC but scheduled for providing NR, generation costs of all units, and slack penalty costs. Constraints (20.2)-(20.4) mean that only the quick-start units not committed in UC may change statuses. All units are subjected to generation limits (20.5)-(20.8), dispatch adjustment limits (20.9)-(20.10), ramping limits (20.11)-(20.12), RES power limits (20.13), and binary requirements (20.14). The system shall satisfy power balance (20.15) and transmission limits (20.16)-(20.17) against actual load $\hat{\mathbf{L}}$ and RES realizations.

The non-negative slack variables (20.18)-(20.19) ensure the RD feasibility w.r.t. the given UC solutions.

$$\min_{\mathbf{z}} \left\{ \begin{aligned} & \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} (C_i^{\text{su}} U_{it}^{\text{RD,qs}} + C_i^{\text{nl}} I_{it}^{\text{RD,qs}}) \\ & + \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} C_{ik}^{\text{sg}} P_{itk}^{\text{RD,sg}} \\ & + \sum_{t \in \mathcal{T}} (C_t^{\text{gs}} S_t^{\text{gs}} + C_t^{\text{ls}} S_t^{\text{ls}}) \\ & + \sum_{t \in \mathcal{T}} \sum_{b \in \mathcal{B}} C_b^{\text{bs}} (S_{bt}^+ + S_{bt}^-) \end{aligned} \right\} \quad (20.1)$$

$$\text{where } \mathbf{z} = \left\{ \begin{aligned} & D^{\text{RD,qs}}, I^{\text{RD}}, I^{\text{RD,qs}}, U^{\text{RD,qs}}, \\ & P^{\text{RD}}, P^{\text{RD,sg}}, S^{\text{gs}}, S^{\text{ls}}, S^+, S^-, W^{\text{RD}} \end{aligned} \right\}$$

Generator Constraints:

$$I_{it}^{\text{RD}} = I_{it}^{\star} + I_{it}^{\text{RD,qs}}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (20.2)$$

$$I_{it}^{\text{RD,qs}} \leq O_{it}^{\star}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (20.3)$$

$$U_{it}^{\text{RD,qs}} - D_{it}^{\text{RD,qs}} = I_{it}^{\text{RD,qs}} - I_{i,t-1}^{\text{RD,qs}}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (20.4)$$

$$P_{it}^{\text{RD}} = \sum_{k \in \mathcal{K}} P_{itk}^{\text{RD,sg}}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (20.5)$$

$$0 \leq P_{itk}^{\text{RD,sg}} \leq \bar{P}_{ik}^{\text{sg}} I_{it}^{\text{RD}}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, k \in \mathcal{K}; \quad (20.6)$$

$$P_{it}^{\text{RD}} \geq P_i^{\text{m}} I_{it}^{\text{RD}}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (20.7)$$

$$P_{it}^{\text{RD}} \leq P_i^{\text{M}} I_{it}^{\star} + R_{it}^{\text{nr}\star} I_{it}^{\text{RD,qs}}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (20.8)$$

$$P_{it}^{\text{RD}} - P_{it}^{\star} \geq -R_{it}^{\text{sr}\star} I_{it}^{\star} - P_i^{\text{M}} I_{it}^{\text{RD,qs}}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (20.9)$$

$$P_{it}^{\text{RD}} - P_{it}^{\star} \leq R_{it}^{\text{sr}\star} I_{it}^{\star} + P_i^{\text{M}} I_{it}^{\text{RD,qs}}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (20.10)$$

$$P_{it}^{\text{RD}} - P_{i,t-1}^{\text{RD}} \leq P_i^{\text{M}}(1 - I_{it}^{\text{RD}})$$

$$+ R_i^{\uparrow} I_{i,t-1}^{\text{RD}} + R_i^{\text{su}}(I_{it}^{\text{RD}} - I_{i,t-1}^{\text{RD}}), \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (20.11)$$

$$P_{i,t-1}^{\text{RD}} - P_{it}^{\text{RD}} \leq P_i^{\text{M}}(1 - I_{i,t-1}^{\text{RD}})$$

$$+ R_i^{\downarrow} I_{it}^{\text{RD}} + R_i^{\text{sd}}(I_{i,t-1}^{\text{RD}} - I_{it}^{\text{RD}}), \quad \forall i \in \mathcal{I}, i \in \mathcal{I}; \quad (20.12)$$

$$0 \leq W_{jt}^{\text{RD}} \leq \tilde{W}_{jt}, \quad \forall j \in \mathcal{J}, t \in \mathcal{T}; \quad (20.13)$$

$$D_{it}^{\text{RD,qs}}, I_{it}^{\text{RD}}, I_{it}^{\text{RD,qs}}, U_{it}^{\text{RD,qs}} \in \{0, 1\}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (20.14)$$

System Constraints:

$$\sum_{i \in \mathcal{I}} P_{it}^{\text{RD}} + \sum_{j \in \mathcal{J}} W_{jt}^{\text{RD}} - S_t^{\text{gs}} = \sum_{q \in \mathcal{Q}} \tilde{L}_{qt} - S_t^{\text{ls}}, \quad \forall t \in \mathcal{T}; \quad (20.15)$$

$$\mathcal{F}_b(\mathbf{P}_t^{\text{RD}}, \mathbf{W}_t^{\text{RD}}, \tilde{\mathbf{L}}_t) - S_{bt}^- \leq B_b, \quad \forall b \in \mathcal{B}, t \in \mathcal{T}; \quad (20.16)$$

$$\mathcal{F}_b(\mathbf{P}_t^{\text{RD}}, \mathbf{W}_t^{\text{RD}}, \tilde{\mathbf{L}}_t) + S_{bt}^+ \geq -B_b, \quad \forall b \in \mathcal{B}, t \in \mathcal{T}; \quad (20.17)$$

$$S_t^{\text{gs}}, S_t^{\text{ls}} \geq 0, \quad \forall t \in \mathcal{T}; \quad (20.18)$$

$$S_{bt}^+, S_{bt}^- \geq 0, \quad \forall b \in \mathcal{B}, t \in \mathcal{T}; \quad (20.19)$$

Following the current industry practice, the DC power flow model is used in this paper to formulate network security constraints. Nevertheless, the presented bi-level MIP framework can incorporate other complex forms of network security constraints explored in the academic literature, such as linearized [54] and SOCP-based [55] AC power flow formulations.

C. Relaxed Operation Models of CPO-NT- ϕ With Tunable Relaxing Degree

CPO-NT- ϕ is trained by the bi-level methodology described in Section III, in which the binary requirement in the lower-level operation models is relaxed.

The relaxed UC and RD models are formulated as in (21) and (22), in which the binary variables are relaxed as in (21.3) and (22.3), while their relaxing degrees are further controlled by constraints (21.4)-(21.8) and (22.4)-(22.8) via

a pre-determined parameter $\phi \in [0, 1]$ together with four sets of new variables [56], i.e., those hatted by ι .

ϕ -Relaxed Unit Commitment Model:

$$\min \text{ The original UC objective function (19.1)} \quad (21.1)$$

$$\text{s. t. (19.2) -- (19.15), (19.17) -- (19.19);} \quad (21.2)$$

$$0 \leq D_{it}, I_{it}, O_{it}, U_{it} \leq 1, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (21.3)$$

$$D_{it} = (1 - \phi) \sum_{a=1}^{\lfloor \log_2 \frac{1}{1-\phi} \rfloor + 1} 2^{a-1} \hat{D}_{ita}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (21.4)$$

$$I_{it} = (1 - \phi) \sum_{a=1}^{\lfloor \log_2 \frac{1}{1-\phi} \rfloor + 1} 2^{a-1} \hat{I}_{ita}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (21.5)$$

$$O_{it} = (1 - \phi) \sum_{a=1}^{\lfloor \log_2 \frac{1}{1-\phi} \rfloor + 1} 2^{a-1} \hat{O}_{ita}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (21.6)$$

$$U_{it} = (1 - \phi) \sum_{a=1}^{\lfloor \log_2 \frac{1}{1-\phi} \rfloor + 1} 2^{a-1} \hat{U}_{ita}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (21.7)$$

$$\hat{D}_{ita}, \hat{I}_{ita}, \hat{O}_{ita}, \hat{U}_{ita} \in \{0, 1\}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, a \in \{1, \dots, \lfloor \log_2 \frac{1}{1-\phi} \rfloor + 1\}; \quad (21.8)$$

ϕ -Relaxed Re-Dispatch Model:

$$\min \text{ The original RD objective function (20.1)} \quad (22.1)$$

$$\text{s. t. (20.2) -- (20.13), (20.15) -- (20.19);} \quad (22.2)$$

$$0 \leq D_{it}^{\text{RD,qs}}, I_{it}^{\text{RD}}, I_{it}^{\text{RD,qs}}, U_{it}^{\text{RD,qs}} \leq 1, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (22.3)$$

$$D_{it}^{\text{RD,qs}} = (1 - \phi) \sum_{a=1}^{\lfloor \log_2 \frac{1}{1-\phi} \rfloor + 1} 2^{a-1} \hat{D}_{ita}^{\text{RD,qs}}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (22.4)$$

$$I_{it}^{\text{RD}} = (1 - \phi) \sum_{a=1}^{\lfloor \log_2 \frac{1}{1-\phi} \rfloor + 1} 2^{a-1} \hat{I}_{ita}^{\text{RD}}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (22.5)$$

$$I_{it}^{\text{RD,qs}} = (1 - \phi) \sum_{a=1}^{\lfloor \log_2 \frac{1}{1-\phi} \rfloor + 1} 2^{a-1} \hat{I}_{ita}^{\text{RD,qs}}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (22.6)$$

$$U_{it}^{\text{RD,qs}} = (1 - \phi) \sum_{a=1}^{\lfloor \log_2 \frac{1}{1-\phi} \rfloor + 1} 2^{a-1} \hat{U}_{ita}^{\text{RD,qs}}, \quad \forall i \in \mathcal{I}, t \in \mathcal{T}; \quad (22.7)$$

$$\hat{D}_{ita}^{\text{RD,qs}}, \hat{I}_{ita}^{\text{RD}}, \hat{I}_{ita}^{\text{RD,qs}}, \hat{U}_{ita}^{\text{RD,qs}} \in \{0, 1\}; \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, a \in \{1, \dots, \lfloor \log_2 \frac{1}{1-\phi} \rfloor + 1\}; \quad (22.8)$$

Here, $\lfloor \cdot \rfloor$ is the floor function. Models (21) and (22) highlight that their relaxing degree can be tuned via parameter ϕ . Specifically, taking the status variable I_{it} as an example, if $\phi = 0$ (0% relaxation), then $I_{it} \in \{0, 1\}$ —(21) and (22) are equivalent to the original MIP models (19) and (20); if $\phi = 0.5$ (50% relaxation), then $I_{it} \in \{0, 0.5, 1\}$; if $\phi = 0.9$ (90% relaxation), then $I_{it} \in \{0, 0.1, 0.2, \dots, 0.9, 1\}$. Note that when $\phi = 1$ (100% relaxation), (21.4)–(21.8) and (22.4)–(22.8) will be naturally removed, transforming (21) and (22) into LP.

Regarding the training of CPO-NT- ϕ , when $\phi \in [0, 1]$, (21) and (22) remain MIP models, and the training method described in Section III-C is still valid; when $\phi = 1$, (21) and (22) become LP models, thus, the training follows the method presented in [44]—replacing the lower-level problems with their KKT conditions, then solving the ERM as an MIP.

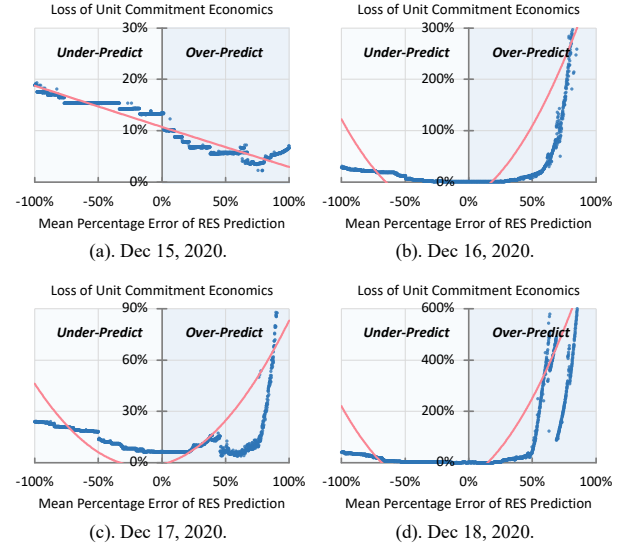


Fig. 12. Approximating results for NNPO-NT.

TABLE XII
COEFFICIENTS OF APPROXIMATING RESULTS

	a	b	c
Dec 15, 2020	0.0012	-0.0791	0.1074
Dec 16, 2020	2.9703	1.4027	-0.3433
Dec 17, 2020	0.6565	0.1846	-0.0097
Dec 18, 2020	6.0497	3.2362	-0.6100

TABLE XIII
HYPER-PARAMETERS OF NNPO-NT

	NNPO-7	NNPO-349
Learning Rate	0.001	0.0001
Hidden Layer Structure	$ \mathcal{T} \times \mathcal{J} $	$ \mathcal{T} \times \mathcal{J} $
Activation Function	linear	linear
Number of Epochs	50	400
Batch Size	1	2

D. Implementation Details of NNPO-NT

NNPO-NT is a variant of the cost-oriented prediction method [24], including the following major steps:

Step 1: For day D , 2,000 scenarios are generated to simulate a set of imperfect RES predictions. Then, the UC-RD process is executed for each scenario to calculate its UC economics loss (4). This procedure yields 2,000 data points describing the relationship between the mean percentage error (MPE) and UC economics loss, as depicted by the blue dots in Fig. 12;

Step 2: Approximate the blue dots using quadratic functions (23), as depicted by the red lines in Fig. 12. These quadratic functions approximate the relationship between MPE and UC economics loss (4). Coefficients of the approximate quadratic functions are listed in Table XII. Note that the loss function (23) can be linear, quadratic, cubic, or quartic. The quadratic function is selected for two reasons: *i)* the insights gained from [24] and *ii)* its outperformance over other options.

$$\text{Loss Caused by MPE} = a \times \text{MPE}^2 + b \times \text{MPE} + c \quad (23)$$

Step 3: Using the approximate functions (23) as loss functions, the NN-based tailors are trained via TensorFlow 2.14. The setting of the hyper-parameters is listed in Table XIII.

REFERENCES

- [1] W. Wei, F. Liu, and S. Mei, "Distributionally robust co-optimization of energy and reserve dispatch," *IEEE Trans. Sustain. Energy*, vol. 7, no. 1, pp. 289–300, 2016.
- [2] OpenDataElia. [Online]. Available: <https://opendata.elia.be/pages/home/>
- [3] Maintaining operating reserves fact sheet of CAISO. [Online]. Available: <https://www.caiso.com/Documents/maintaining-operating-reserves-fact-sheet.pdf>
- [4] M. Parvania and A. Scaglione, "Unit commitment with continuous-time generation and ramping trajectory models," *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 3169–3178, 2016.
- [5] T. Ding, X. Zhang, R. Lu, M. Qu, M. Shahidepour, Y. He, and T. Chen, "Multi-stage distributionally robust stochastic dual dynamic programming to multi-period economic dispatch with virtual energy storage," *IEEE Trans. Sustain. Energy*, vol. 13, no. 1, pp. 146–158, 2022.
- [6] T. Ding, Q. Yang, X. Liu, C. Huang, Y. Yang, M. Wang, and F. Blaabjerg, "Duality-free decomposition based data-driven stochastic security-constrained unit commitment," *IEEE Trans. Sustain. Energy*, vol. 10, no. 1, pp. 82–93, 2019.
- [7] B. Hu and L. Wu, "Robust SCUC considering continuous/discrete uncertainties and quick-start units: A two-stage robust optimization with mixed-integer recourse," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1407–1419, 2016.
- [8] Y. Bengio, "Using a financial training criterion rather than a prediction criterion," *Int. J. Neural Syst.*, vol. 8, no. 4, pp. 433–443, 1997.
- [9] C. Bergmeir and et al., "Comparison and evaluation of methods for a predict+optimize problem in renewable energy," *arXiv preprint arXiv:2212.10723*, 2022.
- [10] A. Gillian Robinson, P. Van Hentenryck, and R. Krishnan, "INFORMS editor's cut: Advances in integrating AI & O.R." 2021. [Online]. Available: <https://pubsonline.informs.org/editorscut/AI-ML/methods>
- [11] D. Bertsimas and N. Kallus, "From predictive to prescriptive analytics," *Manage. Sci.*, vol. 66, no. 3, pp. 1025–1044, 2020.
- [12] A. N. Elmachtoub and P. Grigas, "Smart "predict, then optimize"," *Manage. Sci.*, vol. 68, no. 1, pp. 9–26, 2022.
- [13] G.-Y. Ban and C. Rudin, "The big data newsvendor: Practical insights from machine learning," *Oper. Res.*, vol. 67, no. 1, pp. 90–108, 2019.
- [14] B. Tang and E. B. Khalil, "PyEPO: A PyTorch-based end-to-end predict-then-optimize library for linear and integer programming," *arXiv preprint arXiv:2206.14234*, 2022.
- [15] P. Vossler, S. Aghaei, N. Justin, N. Jo, A. Gómez, and P. Vayanos, "ODTlearn: A package for learning optimal decision trees for prediction and prescription," *arXiv preprint arXiv:2307.15691*, 2023.
- [16] D. Bergman, T. Huang, P. Brooks, A. Lodi, and A. Raghunathan, "JANOS: An integrated predictive and prescriptive modeling framework," *INFORMS J. Comput.*, vol. 34, no. 2, pp. 807–816, 2022.
- [17] T. Carriere and G. Kariniotakis, "An integrated approach for value-oriented energy forecasting and data-driven decision-making application to renewable energy trading," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6933–6944, 2019.
- [18] A. Stratigakos, S. Camal, A. Michiorri, and G. Kariniotakis, "Prescriptive trees for integrated forecasting and optimization applied in trading of renewable energy," *IEEE Trans. Power Syst.*, vol. 37, no. 6, pp. 4696–4708, 2022.
- [19] L. Sang, Y. Xu, H. Long, and W. Wu, "Safety-aware semi-end-to-end coordinated decision model for voltage regulation in active distribution network," *IEEE Trans. Smart Grid*, vol. 14, no. 3, pp. 1814–1826, 2023.
- [20] M. Á. Muñoz, J. M. Morales, and S. Pineda, "Feature-driven improvement of renewable energy forecasting and trading," *IEEE Trans. Power Syst.*, vol. 35, no. 5, pp. 3753–3763, 2020.
- [21] C. Zhao, C. Wan, and Y. Song, "Operating reserve quantification using prediction intervals of wind power: An integrated probabilistic forecasting and decision methodology," *IEEE Trans. Power Syst.*, vol. 36, no. 4, pp. 3701–3714, 2021.
- [22] Y. Wang and L. Wu, "Improving economic values of day-ahead load forecasts to real-time power system operations," *IET Gener. Transm. Distrib.*, vol. 11, no. 17, pp. 4238–4247, 2017.
- [23] G. Li and H.-D. Chiang, "Toward cost-oriented forecasting of wind power generation," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2508–2517, 2018.
- [24] J. Zhang, Y. Wang, and G. Hug, "Cost-oriented load forecasting," *Electr. Power Syst. Res.*, vol. 205, p. 107723, 2022.
- [25] M. Sun, F. Teng, X. Zhang, G. Strbac, and D. Pudjianto, "Data-driven representative day selection for investment decisions: A cost-oriented approach," *IEEE Trans. Power Syst.*, vol. 34, no. 4, pp. 2925–2936, 2019.
- [26] C. Zhang, S. Lasaulce, M. Hennebel, L. Saludjian, P. Panciatici, and H. V. Poor, "Decision-making oriented clustering: Application to pricing and power consumption scheduling," *Appl. Energy*, vol. 297, p. 117106, 2021.
- [27] O. Yurdakul, F. Qiu, and A. Meyer, "A predictive prescription framework for stochastic unit commitment using boosting ensemble learning algorithms," *IEEE Trans. Power Syst.*, vol. 1, no. 1, pp. 1–14, 2023.
- [28] P. L. Donti, B. Amos, and J. Z. Kolter, "Task-based end-to-end model learning in stochastic optimization," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, Conference Proceedings.
- [29] J. Han, L. Yan, and Z. Li, "A task-based day-ahead load forecasting model for stochastic economic dispatch," *IEEE Trans. Power Syst.*, vol. 36, no. 6, pp. 5294–5304, 2021.
- [30] Y. Zhang, M. Jia, H. Wen, and Y. Shi, "Value-oriented renewable energy forecasting for coordinated energy dispatch problems at two stages," *arXiv preprint arXiv:2309.00803*, 2023.
- [31] A. Ferber, B. Wilder, D. B., and T. M., "Mipaal: Mixed integer program as a layer," *Proc. Innov. Appl. Artif. Intell. Conf.*, pp. 1504–1511, 2020.
- [32] J. Mandi, E. Demirovic, P. J. Stuckey, and T. Guns, "Smart predict-and-optimize for hard combinatorial optimization problems," *Proc. Innov. Appl. Artif. Intell. Conf.*, vol. 34, pp. 1603–1610, 2020.
- [33] J. Mandi and T. Guns, "Interior point solving for lp-based prediction+optimisation," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 7272–7282, 2020.
- [34] L. Sang, Y. Xu, H. Long, Q. Hu, and H. Sun, "Electricity price prediction for energy storage system arbitrage: A decision-focused approach," *IEEE Trans. Smart Grid*, vol. 13, no. 4, pp. 2822–2832, 2022.
- [35] Y. Zhou, Q. Wen, J. Song, X. Cui, and Y. Wang, "Load data valuation in multi-energy systems: An end-to-end approach," *arXiv preprint arXiv:2311.09839*, 2023.
- [36] Y. Zhang, H. Wen, and Q. Wu, "A contextual bandit approach for value-oriented prediction interval forecasting," *IEEE Trans. Smart Grid*, pp. 1–11, 2023.
- [37] H. Zhang, R. Li, Y. Chen, Z. Chu, M. Sun, and F. Teng, "Risk-aware objective-based forecasting in inertia management," *IEEE Trans. Power Syst.*, vol. 39, no. 2, pp. 4612–4623, 2024.
- [38] X. Chen, Y. Yang, Y. Liu, and L. Wu, "Feature-driven economic improvement for network-constrained unit commitment: A closed-loop predict-and-optimize framework," *IEEE Trans. Power Syst.*, vol. 37, no. 4, pp. 3104–3118, 2022.
- [39] H. Wu, D. Ke, L. Song, S. Liao, J. Xu, Y. Sun, and K. Fang, "A novel stochastic unit commitment characterized by closed-loop forecast-and-decision for wind integrated power systems," *IEEE Trans. Power Syst.*, vol. 1, no. 1, pp. 1–15, 2023.
- [40] J. M. Morales, M. Zugno, S. Pineda, and P. Pinson, "Electricity market clearing with improved scheduling of stochastic production," *Eur. J. Oper. Res.*, vol. 235, no. 3, pp. 765–774, 2014.
- [41] M. Á. Muñoz, S. Pineda, and J. M. Morales, "A bilevel framework for decision-making under uncertainty with contextual information," *Omega*, vol. 108, p. 102575, 2022.
- [42] J. M. Morales, M. Á. Muñoz, and S. Pineda, "Prescribing net demand for two-stage electricity generation scheduling," *Oper. Res. Perspect.*, vol. 10, p. 100268, 2023.
- [43] J. D. Garcia, A. Street, T. Homem-de Mello, and F. D. Munoz, "Application-driven learning via joint prediction and optimization of demand and reserves requirement," *arXiv:2102.13273*, 2021.
- [44] V. Dvorkin, S. Delikaraoglou, and J. M. Morales, "Setting reserve requirements to approximate the efficiency of the stochastic dispatch," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1524–1536, 2019.
- [45] N. Viofora, S. Delikaraoglou, P. Pinson, G. Hug, and J. Holbøll, "Dynamic reserve and transmission capacity allocation in wind-dominated power systems," *IEEE Trans. Power Syst.*, vol. 36, no. 4, pp. 3017–3028, 2021.
- [46] C. Zhao, C. Wan, and Y. Song, "Cost-oriented prediction intervals: On bridging the gap between forecasting and decision," *IEEE Trans. Power Syst.*, vol. 37, no. 4, pp. 3048–3062, 2022.
- [47] Z. Shao, Q. Zhai, Z. Han, and X. Guan, "A linear ac unit commitment formulation: An application of data-driven linear power flow model," *Int. J. Electr. Power Energy Syst.*, vol. 145, p. 108673, 2023.
- [48] Z. Lu, J. Wang, M. Shahidepour, L. Bai, and Z. Li, "Convex-hull pricing of ancillary services for power system frequency regulation with renewables and carbon-capture-utilization-and-storage systems," *IEEE Trans. Power Syst.*, vol. 1, no. 1, pp. 1–18, 2024.
- [49] J. F. Bard, *Practical bilevel optimization: algorithms and applications*. Springer Science & Business Media, 2013, vol. 30.

- [50] B. Zeng and Y. An, "Solving bilevel mixed integer program by reformulations and decomposition," *Optim. Online*, pp. 1–34, 2014.
- [51] J. Dupačová, N. Gröwe-Kuska, and W. Römisch, "Scenario reduction in stochastic programming," *Math. Program.*, vol. 95, no. 3, pp. 493–511, 2003.
- [52] C. Wang and Y. Fu, "Fully parallel stochastic security-constrained unit commitment," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3561–3571, 2016.
- [53] Z. Lu, J. Wang, M. Shahidehpour, L. Bai, Y. Xiao, and H. Li, "Co-operative operation of distributed energy resources and thermal power plant with a carbon-capture-utilization-and-storage system," *IEEE Trans. Power Syst.*, vol. 39, no. 1, pp. 1850–1866, 2024.
- [54] T. Zhang, M. Sun, J. L. Cremer, N. Zhang, G. Strbac, and C. Kang, "A confidence-aware machine learning framework for dynamic security assessment," *IEEE Trans. Power Syst.*, vol. 36, no. 5, pp. 3907–3920, 2021.
- [55] D. Tuncer and B. Kocuk, "An misocp-based decomposition approach for the unit commitment problem with ac power flows," *IEEE Trans. Power Syst.*, vol. 38, no. 4, pp. 3388–3400, 2023.
- [56] J. Zou, S. Aluned, and X. A. Sun, "Multistage stochastic unit commitment using stochastic dual dynamic integer programming," *IEEE Trans. Power Syst.*, vol. 34, no. 3, pp. 1814–1823, 2019.

Xianbang Chen (Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Sichuan University, Chengdu, China, in 2017 and 2020, respectively. He is currently working toward the Ph.D. degree with the Stevens Institute of Technology, Hoboken, NJ, USA. His research interest focuses on applying prescription methods on unit commitment.

Yikui Liu (Member, IEEE) received the B.S. degree in electrical engineering and automation from the Nanjing Institute of Technology, China, in 2012, the M.S. degree in power system and automation from Sichuan University, China, in 2015, and the Ph.D. degree in electrical and computer engineering from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2020. He worked in Siemens, USA, as an Energy Market Engineer, during 2020–2021. He worked as a Postdoctoral Researcher with the Stevens Institute of Technology, Hoboken, NJ, USA, during 2021–2023. He is currently an associate researcher with Sichuan University, Chengdu, China. His research interests include the power market and OPF in distribution systems.

Lei Wu (Fellow, IEEE) received the B.S. degree in electrical engineering and the M.S. degree in systems engineering from Xi'an Jiaotong University, Xi'an, China, in 2001 and 2004, respectively, and the Ph.D. degree in electrical engineering from Illinois Institute of Technology (IIT), Chicago, IL, USA, in 2008. From 2008 to 2010, he was a Senior Research Associate with the Robert W. Galvin Center for Electricity Innovation, IIT. He was a summer Visiting Faculty at NYISO in 2012. He was a Professor with the Electrical and Computer Engineering Department, Clarkson University, Potsdam, NY, USA, till 2018. Currently, he is Anson Wood Burchard Chair Professor with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA. His research interests include power systems operation and planning, energy economics, and community resilience microgrid.