Online Algorithms with Uncertainty-Quantified Predictions

Bo Sun [1] **Jerry Huang** [2] **Nicolas Christianson** [2] **Mohammad Hajiesmaili** [3] **Adam Wierman** [2] **Raouf Boutaba** [1]

Abstract

The burgeoning field of algorithms with predictions studies the problem of using possibly imperfect machine learning predictions to improve online algorithm performance. While nearly all existing algorithms in this framework make no assumptions on prediction quality, a number of methods providing *uncertainty quantification (UQ)* on machine learning models have been developed in recent years, which could enable additional information about prediction quality at decision time. In this work, we investigate the problem of optimally utilizing uncertainty-quantified predictions in the design of online algorithms. In particular, we study two classic online problems, ski rental and online search, where the decision-maker is provided predictions augmented with UQ describing the likelihood of the ground truth falling within a particular range of values. We demonstrate that non-trivial modifications to algorithm design are needed to fully leverage the UQ predictions. Moreover, we consider how to utilize more general forms of UQ, proposing an online learning framework that learns to exploit UQ to make decisions in multi-instance settings.

1. Introduction

Classic online algorithms are designed to ensure worst-case performance guarantees. However, such algorithms are often overly pessimistic and perform poorly in real-world applications since worst-case instances rarely occur. To address the pessimism of these algorithms, a recent surge of work has investigated the design of algorithms utilizing machine-learned predictions (Mitzenmacher & Vassilvitskii, 2020; Lykouris & Vassilvitskii, 2021; Purohit et al., 2018). In this line of research, an algorithm is given additional information on the problem instance in the form of predictions or "advice", possibly from a machine learning model. Notably, it is typically the case that no assumptions are made on predictions' quality. Thus, algorithms must treat them as "untrusted", seeking to exploit them when they are accurate while ensuring worst-case guarantees when they are not.

Driven by safety-critical applications, uncertainty quantification (UQ) has recently become a prominent field of research in machine learning. UQ aims to provide quantitative measurements on machine learning models' uncertainty about their predictions. One of the state-of-the-art methods for UQ is *conformal inference* (Vovk et al., 1999; 2005; Papadopoulos et al., 2002), which can transform the predictions of any black-box algorithm into a prediction interval (or prediction set) that contains the true value with high probability. Although UQ has been widely used for general decision-making under uncertainty, as in (Vovk & Bendtsen, 2018; Marusich et al., 2023; Sun et al., 2023), there has been limited study on its use for online problems. Thus, the key question we aim to answer in this paper is:

How can we incorporate uncertainty-quantified predictions into the design of competitive online algorithms?

To address the above question, we require a new design objective that interpolates between worst-case analysis and average-case analysis for online algorithms. Algorithms augmented with UQ predictions have access to both predictions of future inputs and the associated prediction quality, which can be leveraged to improve upon worst-case performance; however, UQ predictions often cannot exactly reconstruct distributional information to enable typical average-case guarantees. As such, we design algorithms to minimize a new performance metric, which we term *distributionally-robust competitive ratio* (DRCR): we seek algorithms that perform well on the worst-case distribution drawn from the ambiguity set determined by a given UQ.

In particular, this paper makes contributions in threefold for designing online algorithms that leverage UQ predictions.

Optimal online algorithms under distributionally-robust analysis We frame the online algorithms with UQ predic-

[1]David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada. [2]Computing + Mathematical Sciences Department, California Institute of Technology, Pasadena, USA. [3]Manning College of Information and Computer Sciences, University of Massachusetts Amherst, Amherst, USA. Correspondence to: Bo Sun <bo.sun@uwaterloo.ca>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

tions as a distributionally-robust online algorithm design problem. Then we design online algorithms using probabilistic interval predictions for the ski rental and online search problems, in both cases showing that they attain the optimal DRCR (see Theorems 2 and 3). These two problems have played a key role in the development of learning-augmented algorithms and thus are natural problems with which to begin the study of UQ for online algorithms.

Optimization-based algorithmic approach Technically, we propose an optimization-based approach for incorporating UQ predictions into online algorithms. The approach consists of building an ancillary optimization problem based on predictions with the objective of minimizing DRCR for hard instances of the problem. The solution of the optimization problem then yields the optimal algorithm design, considering the provided UQ. This approach is general in the sense that the optimization can be tuned based on the specific forms of the predictions and design goals, and, thus, this approach can potentially be applied to incorporate other forms of UQ predictions and to devise algorithms for other online problems beyond ski rental and online search.

Online learning for exploiting UQs across multiple instances Finally, we propose an online learning approach that learns to exploit general forms of UQ across multiple problem instances. Here, the probabilistic interval predictions may be imperfect (e.g., due to non-exchangeability of the data) or alternative notions of UQ are employed. We show that, under mild Lipschitzness conditions, one can obtain sublinear regret guarantees with respect to solving the full optimization formulation of the DRCR problem. We demonstrate the regret guarantees obtained by this framework in the ski rental and online search problems. Moreover, when problem instances are not fully adversarial (i.e., the distribution generating problem instances is not the worst case for the given UQ), our online learning approach outperforms the optimization-based approach, as we demonstrate in experiments in Section 5.

1.1. Related Literature

Algorithms with untrusted predictions A significant body of work has emerged considering the design of algorithms that incorporate untrusted predictions of either problem parameters or optimal decisions (Lykouris & Vassilvitskii, 2021; Purohit et al., 2018; Mahdian et al., 2012; Mitzenmacher & Vassilvitskii, 2022; Wei & Zhang, 2020; Antoniadis et al., 2020; Christianson et al., 2023; Sun et al., 2021). However, in nearly all of these works, predictions are assumed to be point predictions of problem parameters or decisions, i.e., individual untrusted decisions with no further assumptions on quality, uncertainty, probability of correctness, etc. Several recent studies have considered alternative prediction paradigms, including the setting of learning predictions from samples or distributional advice (Anand et al., 2020; Diakonikolas et al., 2021; Besbes et al., 2022; Khodak et al., 2022), and predictions which are assumed to be correct with a certain, known or unknown probability (Gupta et al., 2022). Our work is distinguished from these prior results in that we consider a more general class of *uncertainty-quantified* predictions. In particular, our model of probabilistic interval predictions allows for predictions that fall into a certain interval with a given probability, thus generalizing the prediction paradigm of (Gupta et al., 2022) to one more closely matched to uncertainty quantification methods in the machine learning literature.

Online learning Our online learning-based approach to utilizing UQ builds on techniques from the online learning literature and, specifically, online learning with side information and data-driven algorithm design. The problem of exploiting additional side information to improve performance in online learning has been widely studied in both bandit (Agrawal & Goyal, 2013; Slivkins, 2011; Bastani & Bayati, 2020) and partial/full-feedback (Hazan & Megiddo, 2007; Dekel et al., 2017; Kuzborskij & Cesa-Bianchi, 2020) settings. Our work is most closely related to the results in (Hazan & Megiddo, 2007); however, our results go beyond the Lipschitz assumptions on the policy class employed in (Hazan & Megiddo, 2007), and we show that we can exploit Lipschitzness of *any* problem instance cost upper bound to enable competing against general policies when exploiting UQ in online problems. Our online learning formulation is also aligned with the data-driven algorithm design framework (Balcan, 2020; Balcan et al., 2018) that adaptively selects the parameterized algorithms across multiple instances without using side information. Our work extends the problem setting by exploring how to utilize the additional UQ predictions in the algorithm selection, and showing regret guarantees under mild Lipschitzness assumptions.

2. Online Algorithms with UQ Predictions

For an online cost minimization problem, let $\mathcal{I}$ denote the set of all instances. For each instance $I \in \mathcal{I}$, let $\texttt{ALG}(A, I)$ and $\texttt{OPT}(I)$ denote, respectively, the (expected) cost attained by an online algorithm A and the cost of the offline solution. Under the classic competitive analysis framework (Borodin & El-Yaniv, 2005), online algorithms have *no prior knowledge* of the instance I. Algorithmic design is framed as a single-instance min-max problem, with the objective of finding an online algorithm A to minimize the worst-case competitive ratio , i.e., $\max_{I\in\mathcal{I}} \frac{\texttt{ALG}(A,I)}{\texttt{OPT}(I)}$.

To improve the performance of online algorithms and go beyond worst-case analysis, there has recently been research emerging on algorithms with (untrusted) predic-

tions (Mitzenmacher & Vassilvitskii, 2022; Lykouris & Vassilvitskii, 2021; Purohit et al., 2018). In an abstract setup, we consider the input instance of an online problem that can be characterized by a critical value V (e.g., the number of skiing days for the ski-rental problem). Machine learning tools can be leveraged to make a prediction P about the critical value V. In most scenarios, the *quality of the prediction is unknown* to the online decision-maker; hence, the goal of algorithm design with predictions is to guarantee good performance when the prediction is accurate (i.e., consistency) while still maintaining worst-case guarantees regardless of the prediction accuracy (i.e., robustness). Let $\mathcal{I}_P \subseteq \mathcal{I}$ denote a *consistent* set that contains all instances confirming with the prediction P. Then the consistency η and robustness γ of an online algorithm A are defined as

$$\eta = \max_{I \in \mathcal{I}_P} \frac{\texttt{ALG}(A, I)}{\texttt{OPT}(I)} \text{ and } \gamma = \max_{I \in \mathcal{I}} \frac{\texttt{ALG}(A, I)}{\texttt{OPT}(I)}, \quad (1)$$

which are the worst-case ratios over $\mathcal{I}_P$ and $\mathcal{I}$, respectively. Prior work has shown that there exist strong trade-offs between consistency and robustness bounds (Purohit et al., 2018; Wei & Zhang, 2020; Balseiro et al., 2023; Sun et al., 2021; Étienne Bamas et al., 2020). Therefore, algorithms with predictions usually provide a parameterized class of online algorithms (using a hyper-parameter λ) to achieve different trade-offs. Due to the lack of prediction quality, the selection of the hyper-parameter is left to end users.

In practice, we often have access to some forms of uncertainty quantification about the prediction of the input instance. We model an uncertainty-quantified (UQ) prediction by a vector $\theta := \{P; Q\}$, where P is the prediction and Q specifies the quality of the prediction. For a given θ, we assume that instance I belongs to a fixed unknown distribution ξ_θ. If ξ_θ can be completely specified by θ, the average-case analysis aims to design the online algorithm that can minimize the expected competitive ratio, i.e., $\mathbb{E}_{\xi_\theta}[\frac{\texttt{ALG}(A,I)}{\texttt{OPT}(I)}]$. However, in most cases, UQ can only partially specify the instance distribution, and thus an interpolation between the worst-case analysis and average-case analysis is desired.

2.1. Distributionally-Robust Competitive Analysis

When UQ can coarsely characterize the instance distribution, for a given θ, we can construct an ambiguity set $\mathcal{D}_\theta$ that includes all instance distributions that conform with UQ. An important example of such UQ is probabilistic quantification of prediction correctness, and the ambiguity set contains all distributions that conform with such predictions. In this case, an online algorithm A can be designed to minimize the distributionally-robust competitive ratio (DRCR)

$$\texttt{DRCR}_\theta(A) = \max_{\xi_\theta \in \mathcal{D}_\theta} \mathbb{E}_{\xi_\theta}\left[\frac{\texttt{ALG}(A, I)}{\texttt{OPT}(I)}\right], \quad (2)$$

which is the worst expected competitive ratio over instance distributions from $\mathcal{D}_\theta$. Such an algorithmic design can be considered as an interpolation between worst-case analysis and average-case analysis.

For average-case analysis of competitive algorithms, the performance can be evaluated by expectation of ratios or ratio of expectations. We choose to define the DRCR as the expectation of ratios for two reasons. First, this metric is more commonly considered as the average-case performance measure for the ski rental problem and online search problems (e.g., (Fujiwara & Iwama, 2005) and (Fujiwara et al., 2011)), which are the focus of this paper. Second, the DRCR defined based on the expectation of ratios can be shown to be a convex combination of the consistency and robustness of the online algorithms with untrusted algorithms. This connection makes the DRCR more appealing as an extension of the consistency-robustness metric given additional information on the quality of prediction.

Probabilistic interval predictions One important class of UQs that can be leveraged for distributionally-robust analysis is *probabilistic interval predictions* (PIP).

Definition 1. *For $\ell \leq u$ and $\delta \in [0, 1]$, $\texttt{PIP}(\theta)$ with $\theta = \{\ell, u; \delta\}$ is called a probabilistic interval prediction for a critical value V of an input instance, and it predicts that with at least probability $1 - \delta$, the true value V is within $[\ell, u]$, i.e., $\mathbb{P}(V \in [\ell, u]) \geq 1 - \delta$.*

In the literature of algorithms with untrusted predictions, the untrusted prediction P is a special case of $\texttt{PIP}(\ell, u; \delta)$ when the prediction is a point prediction $\ell = u = P$ and there is no guarantee on this prediction $\delta = 1$.

PIP can be obtained through *conformal predictions* (Vovk et al., 1999; 2005; Papadopoulos et al., 2002). Given exchangeable data, conformal prediction can transform the outputs of any black-box predictors into a prediction set/interval that can cover the true value with high probabilities. In particular, conformal inference certifies that the prediction P over the critical value V is accurate within an error ε with at least probability $1 - \delta$, i.e., $\mathbb{P}(|V - P| \leq \varepsilon) \geq 1 - \delta$. In this case, the prediction quality is characterized by the prediction error ε and prediction confidence δ. Equivalently, we can frame this UQ as a $\texttt{PIP}(\theta)$ over the instance, i.e., $\mathbb{P}(V \in [\ell, u]) \geq 1 - \delta$, where $\ell := P - \varepsilon$ and $u := P + \varepsilon$.

For a given PIP, let $\mathcal{I}_{\ell,u} \subseteq \mathcal{I}$ denote a consistent set that contains all instances that confirm with the interval prediction. Then the ambiguity set $\mathcal{D}_\theta$ can include all instance distributions such that $\mathbb{P}_{\xi_\theta}(I \in \mathcal{I}_{\ell,u}) \geq 1 - \delta, \forall \xi_\theta \in \mathcal{D}_\theta$, i.e., under distribution ξ_θ, the probability that an instance I belongs to a set $\mathcal{I}_{\ell,u}$ is at least $1 - \delta$. Further, we can observe that the worst instance distribution that maximizes the DRCR in Equation (2) is a two-point distribution, with probability $1 - \delta$ for instance I^η and probability δ

for instance I^γ, where $I^\eta = \arg\max_{I\in\mathcal{I}_{\ell,u}} \frac{\texttt{ALG}(A,I)}{\texttt{OPT}(I)}$ and $I^\gamma = \arg\max_{I\in\mathcal{I}} \frac{\texttt{ALG}(A,I)}{\texttt{OPT}(I)}$. Thus, the DRCR of online algorithm with PIP θ can be transformed into

$$(1-\delta)\cdot\max_{I\in\mathcal{I}_{\ell,u}}\frac{\texttt{ALG}(A,I)}{\texttt{OPT}(I)}+\delta\cdot\max_{I\in\mathcal{I}}\frac{\texttt{ALG}(A,I)}{\texttt{OPT}(I)} := (1-\delta)\cdot\eta+\delta\cdot\gamma, \tag{3}$$

where η and γ are the consistency and robustness of algorithms with untrusted interval predictions.

2.2. An Optimization-Based Algorithmic Approach

We introduce an optimization-based algorithmic approach for the *single-instance* distributionally-robust analysis that can be leveraged to systematically design online algorithms with UQ predictions. We focus on a class of parameterized online algorithms. Let $A(\boldsymbol{w})$ denote the online algorithm with parameter $\boldsymbol{w}\in\Omega$, where Ω is the parameter set. The design of an online algorithm augmented by a UQ prediction $\texttt{PIP}(\theta)$ is to find a policy $\pi\in\Pi:\Theta\to\Omega$ that maps from θ to an online algorithm $A(\boldsymbol{w})$. We propose a general optimization-based approach to design the policy by solving an ancillary optimization problem.

We start by constructing a family of representative hard instances $\mathcal{H}\subseteq\mathcal{I}$ and parameterized algorithms $\{A(\boldsymbol{w})\}_{\boldsymbol{w}\in\Omega}$ for the online problem based on the problem-specific knowledge. Let $\texttt{ALG}(\boldsymbol{w},I)$ and $\texttt{OPT}(I)$ denote the costs of online algorithm $A(\boldsymbol{w})$ and offline algorithm under the instance $I\in\mathcal{H}$. Given the prediction θ, we can further determine a subset $\mathcal{H}_{\ell,u}$ of $\mathcal{H}$, containing instances that conform with the interval prediction, i.e., $\mathcal{H}_{\ell,u}=\mathcal{I}_{\ell,u}\cap\mathcal{H}$. Then, we formulate an optimization problem to minimize DRCR over all parameterized algorithms under such hard instances.

$$\min_{\eta,\gamma\geq 1;\boldsymbol{w}\in\Omega} \quad (1-\delta)\eta+\delta\gamma \tag{4a}$$

$$\text{s.t.} \quad \texttt{ALG}(\boldsymbol{w},I)\leq\eta\cdot\texttt{OPT}(I),\forall I\in\mathcal{H}_{\ell,u}, \tag{4b}$$

$$\texttt{ALG}(\boldsymbol{w},I)\leq\gamma\cdot\texttt{OPT}(I),\forall I\in\mathcal{H}. \tag{4c}$$

Each constraint from either constraint (4b) or constraint (4c) ensures that the ratio between the expected cost of the online algorithm and the cost of the offline optimum is upper bounded by η or γ, respectively. If restricted only to hard instances $\mathcal{H}$, the variables η and γ represent the consistency and robustness of the algorithm $A(\boldsymbol{w})$, and the objective directly optimizes DRCR over all parameterized algorithms. Let $\{\eta^*,\gamma^*,\boldsymbol{w}^*\}$ denote the optimal solution of the above problem. Then we propose to choose $A(\boldsymbol{w}^*)$ as the online algorithm with UQ prediction θ.

Since the optimization problem (4) is based on hard instances, its optimal objective provides a lower bound for DRCR over the parameterized algorithms.

Proposition 1. *No parameterized algorithms $A(\boldsymbol{w}),\boldsymbol{w}\in\Omega$ can achieve a DRCR smaller than $(1-\delta)\eta^*+\delta\gamma^*$.*

This lower bound can be extended for all online algorithms if the parameterized algorithms can characterize all online algorithms under the hard instances (e.g., see examples in Sections 3 and 4). Note that the ancillary problem often involves an infinite number of variables and constraints, which correspond to the high dimension of parameter $\boldsymbol{w}$ and the cardinality of hard instance set $\mathcal{H}$. This necessitates efficient methods for obtaining (approximately) optimal solutions to the problem (4). Furthermore, although the optimization can give a lower bound for the target performance, it is essential to additionally establish an upper bound on DRCR of the algorithm $A(\boldsymbol{w}^*)$ that is devised based on the solution of the optimization. Developing an online algorithm with matching upper and lower bounds requires carefully constructing the hard instances, crafting the parameterized algorithms, and (approximately) solving the ancillary optimization problem simultaneously. In Sections 3 and 4, we showcase how to use this approach to design online algorithms that can make the best use of a given UQ prediction to minimize DRCR in two classic online algorithms problems, the ski rental problem and the online search problem.

3. Ski Rental Problem with UQ Prediction

Problem statement A player aims to ski for an unknown time horizon $N\in\mathbb{Z}^+$. Each day she needs to decide whether to rent skis, which cost \$1 for this day or buy the skis at the cost of $\$B\in\mathbb{Z}^+$ and ski for free from then on. The goal is to minimize the cost of buying and renting skis.

The difficulty of the problem lies in the uncertain time horizon N. If N is known in advance, then the optimal decision is to buy in the beginning if $N\geq B$ and keep renting otherwise. When N is completely unknown, a deterministic online algorithm can achieve a competitive ratio of 2 (Karlin et al., 1988), and this result can be improved to $e/e-1$ by randomization (Karlin et al., 1990). Both results have been proven to be optimal in the worst case. In previous work on the learning-augmented setting of ski rental, the algorithm is assumed to additionally have access to a deterministic point prediction P over the time horizon N but has no information on the quality of this prediction. There exist both deterministic and randomized algorithms that can attain the Pareto-optimal trade-off between consistency and robustness (Purohit et al., 2018; Wei & Zhang, 2020; Étienne Bamas et al., 2020).

We study online algorithms for ski rental with UQ predictions. In particular, UQ about N is given in the form of a probabilistic interval prediction $\theta=\{\ell,u;\delta\}$, i.e., the time horizon N is predicted to be within $\mathbb{Z}^+_{\ell,u}:=\{\ell,\ell+1,\ldots,u\}$ with at least probability $1-\delta$. Instead of making a rent-or-buy decision each day, online decision-making for ski rental can be described as an online (randomized) algorithm with a (random) variable $Y\in\mathbb{Z}^+$ that keeps renting skis until day

$Y-1$ (if the time horizon has not ended) and buys on day Y. We aim to leverage UQ prediction to design the determination of Y so that DRCR can be minimized. In Section 3.1, we first introduce a deterministic algorithm as a warm-up problem to provide insights on algorithms with probabilistic predictions , and then in Section 3.2, we further propose an optimal randomized algorithm augmented with probabilistic interval predictions using the optimization-based approach.

3.1. Warm-up: A Deterministic Algorithm

We first focus on a deterministic algorithm for ski rental with a probabilistic point prediction $\texttt{PPP}(P;\delta)$, which forecasts the skiing horizon is P with probability at least $1-\delta$. To simplify the presentation, we show the results based on a continuous version of the ski rental, where the number of skiing days increases continuously, and $N, B, Y \in \mathbb{R}^+$.

A simple meta-algorithm Based on the definition in Equation (3), the DRCR of online algorithms is a linear combination of consistency and robustness from an algorithm with untrusted predictions. Therefore, we can devise a simple meta-algorithm by leveraging existing consistent and robust algorithms. Let $\texttt{LA}_P(\lambda)$ denote the algorithms with untrusted prediction P designed in (Purohit et al., 2018) for a hyper-parameter $\lambda \in (0,1]$. In particular, $\texttt{LA}_P(\lambda)$ determines the day of purchase $Y = B/\lambda$ if $P < B$ and $Y = B\lambda$ otherwise. $\texttt{LA}_P(\lambda)$ has been proved $(1+\lambda)$-consistent and $(1+1/\lambda)$-robust. A simple meta-algorithm then can take $\texttt{LA}_P(\lambda)$ as input and select the online algorithm with parameter λ to optimize DRCR. Specifically, it determines $\lambda_\delta = \arg\min_{\lambda\in(0,1]}(1-\delta)(1+\lambda)+\delta(1+1/\lambda) = \min\{\sqrt{\delta/(1-\delta)}, 1\}$, and the meta-algorithm is given as $\texttt{LA}_P(\lambda_\delta)$. Further, the DRCR of $\texttt{LA}_P(\lambda_\delta)$ is derived as

$$\chi(\delta) = \begin{cases} 1+2\sqrt{\delta(1-\delta)} & \delta \in [0, \frac{1}{2}] \\ 2 & \delta \in (\frac{1}{2}, 1] \end{cases}. \tag{5}$$

The meta-algorithm can improve DRCR beyond the worst-case competitive ratio of 2 when the prediction quality is high ($\delta \in [0, 1/2]$), with $\chi(\delta)$ rapidly converging to 1 as δ approaches 0. Nonetheless, the prediction becomes ineffective as its quality deteriorates beyond $\delta > 1/2$, reducing the meta-algorithm to the worst-case performance. However, a fundamental question remains: *Can we extract the benefit from low-quality predictions?* Furthermore, the DRCR of the meta-algorithm is independent of the prediction P as the algorithm $\texttt{LA}_P(\lambda)$ treats the prediction P as untrusted and does not leverage its quality δ in its design. Instead, the prediction quality is only used for the hyper-parameter selection. These limitations of the meta-algorithm motivate us to design a new algorithm capable of harnessing the probabilistic predictions more effectively.

Algorithm 1 DSR: Deterministic algorithm for ski rental

input: prediction $\texttt{PPP}(P;\delta)$, buying cost B;
if $P < B$ **then** determine $Y = B$;
else if $P \in (\frac{\sqrt{5}+1}{2}B, +\infty)$ **then** determine $Y = B \cdot \min\{\sqrt{\delta/(1-\delta)}, 1\}$;
else if $P \in [B, \frac{\sqrt{5}+1}{2}B]$ **then**
  if $\chi(\delta) \le \delta + \frac{P}{B}$ **then** determine $Y = B \cdot \min\{\sqrt{\delta/(1-\delta)}, 1\}$;
  else determine $Y = P$;
buy skis on day Y.

An optimal deterministic algorithm In Algorithm 1, we introduce a new deterministic algorithm, referred to as DSR. This algorithm operates within distinct prediction regions: (i) in the *pro-rent* region, defined as $P \in (0, B)$, the algorithm purchases on day B regardless of the specific prediction and prediction quality; (ii) in the *pro-buy* region, defined as $P \in (\frac{\sqrt{5}+1}{2}B, +\infty)$, the algorithm makes an early purchase within the initial B days, with the specific day determined by the design of DSR; (iii) in the *rent-or-buy* region, denoted by $P \in [B, \frac{\sqrt{5}+1}{2}B]$, this algorithm can opt to buy on the predicted day P or make a purchase within the first B days. The decision, in this case, is influenced by both the prediction and its quality, creating a nuanced trade-off between buy and rent. We show that DSR can achieve the optimal DRCR among all deterministic algorithms.

Theorem 1. *Given a* $\texttt{PPP}(P;\delta)$*,* DSR *is the optimal deterministic algorithm for ski rental and achieves the* DRCR

$$\begin{cases} 1+\delta & P \in (0, B) \\ \min\left\{\chi(\delta), \delta + \frac{P}{B}\right\} & P \in [B, \frac{\sqrt{5}+1}{2}B] \\ \chi(\delta) & P \in (\frac{\sqrt{5}+1}{2}B, +\infty) \end{cases}.$$

The crux of DSR's design lies in identifying the dominant decision when the prediction falls within distinct prediction regions. Compared to the meta-algorithm, DSR achieves an improved DRCR over the meta-algorithm for any given prediction. The performance gain can be attributed to explicit utilization of both the prediction and its associated quality in the decision-making process. In particular, even in scenarios where the prediction quality is low $\delta > 1/2$, DSR still manages to enhance the DRCR, especially when the prediction $P < \frac{\sqrt{5}+1}{2}B$. In such cases, any probabilistic information from the prediction can mitigate the worst-case scenarios. Although we can extend the ideas of DSR to incorporate a probabilistic interval prediction (see Appendix B.2 for more details), it becomes increasingly complicated to identify the dominant decisions. In the following section, we show that we can design algorithms using a more systematic optimization-based approach proposed in Section 2.2.

Algorithm 2 $\texttt{RSR}(\boldsymbol{y})$: Randomized algorithm for ski rental

input: purchase distribution $\boldsymbol{y} \in \mathcal{Y}$;
draw a buying day Y from the distribution $\boldsymbol{y}$;
rent skis up to day $Y-1$ and buy on day Y.

3.2. An Optimal Randomized Algorithm

We now introduce a more general randomized algorithm with probabilistic interval prediction $\texttt{PIP}(\ell, u; \delta)$. We can consider a parameterized algorithm $\texttt{RSR}(\boldsymbol{y})$ (described in Algorithm 2) with the purchasing probability $\boldsymbol{y} := \{y(t)\}_{t\in\mathbb{Z}^+}$ as the parameter. Specifically, $y(t)$ denotes the probability of purchasing on day t. Then any online randomized algorithm for ski rental problems can be captured by $\boldsymbol{y} := \{y(t)\}_{t\in\mathbb{Z}^+}$, where $\boldsymbol{y}$ is a distribution of the buying day with support $\mathbb{Z}^+$ and the feasible set of $\boldsymbol{y}$ is given by $\mathcal{Y} := \{\boldsymbol{y} : \sum_{t\in\mathbb{Z}^+} y(t) = 1, y(t) \geq 0, \forall t \in \mathbb{Z}^+\}$.

Let I_N denote an instance of the ski rental problem with time horizon N. We consider the hard instance set $\mathcal{H} := \mathcal{I} = \{I_N\}_{N\in\mathbb{Z}^+}$, which in fact contains all instances of the ski rental problem. The instances conforming with the interval prediction can then be denoted by $\mathcal{H}_{\ell,u} = \{I_N\}_{N\in\mathbb{Z}^+_{\ell,u}}$. Given each instance I_N, the expected cost of a randomized algorithm $\texttt{RSR}(\boldsymbol{y})$ is $\texttt{ALG}(\boldsymbol{y}, I_N) = \sum_{t=1}^{N}(B+t-1)y(t) + N\sum_{t=N+1}^{+\infty} y(t)$, and the cost of the offline algorithm is $\texttt{OPT}(I_N) = \min\{N, B\}$. Given a $\texttt{PIP}(\ell, u; \delta)$, we can formulate an optimization problem (4) to minimize $\texttt{DRCR}$. Let $\{\eta^*, \gamma^*, \boldsymbol{y}^*\}$ and $\texttt{CR}^*_{\texttt{sr}}$ denote the optimal solution and the optimal objective value of the problem, respectively. The optimal randomized algorithm is then given by $\texttt{RSR}(\boldsymbol{y}^*)$.

Theorem 2. *Given a* $\texttt{PIP}(\ell, u; \delta)$*, the* $\texttt{DRCR}$ *of* $\textit{RSR}(\boldsymbol{y}^*)$ *is* $\textit{CR}^*_{sr}$*. Further,* $\textit{CR}^*_{sr}$ *is the optimal* $\textit{DRCR}$ *for ski rental.*

The optimization-based approach for designing $\texttt{RSR}(\boldsymbol{y}^*)$ is general in the sense that it can be tuned to design others algorithms for related problems. For example, one can derive a deterministic algorithm with $\texttt{PIP}(\ell, u; \delta)$ by replacing the feasible set with $\hat{\mathcal{Y}} := \{\boldsymbol{y} : \sum_{t\in\mathbb{Z}^+} y(t) = 1, y(t) \in \{0, 1\}, \forall t \in \mathbb{Z}^+\}$ that restricts the decisions to be deterministic. This systemic design stands in contrast to the ad-hoc development of the deterministic algorithm discussed in the previous section. Given that $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$, the ancillary problem for the randomized algorithm is a relaxation of that of the deterministic algorithm. Thus, $\texttt{RSR}(\boldsymbol{y}^*)$ outperforms the optimal deterministic algorithms.

Noting that the optimization problem for ski rental is a linear program with an infinite number of variables and constraints, to solve $\boldsymbol{y}^*$, we show that the problem can be reduced to an equivalent problem with a finite number of variables and constraints. Therefore, $\boldsymbol{y}^*$ can be solved optimally and efficiently by standard linear programs.

Lemma 1. *The problem* (4) *for ski rental can be reduced to an optimization with* $O(B)$ *variables and* $O(B)$ *constraints.*

4. Online Search Problem with UQ Prediction

Problem statement A player seeks to sell one unit of a resource over a sequence of prices $\{v_n\}_{n\in[N]}$ that arrive online. In response to each price v_n, the player must immediately decide an amount x_n of its remaining resource to sell (resulting in the player earning $v_n x_n$), without the knowledge of future prices or the sequence length N. If any resource remains unsold at the last step N, it is compulsorily sold at the final price v_N. The player's goal is to maximize its total profit $\sum_{n\in[N]} v_n x_n$. Following the standard assumption (El-Yaniv et al., 2001; Lorenz et al., 2009), prices are chosen (possibly adversarially) from a bounded interval, i.e., $v_n \in [m, M]$ for all $n \in [N]$, where $m > 0$.

In prior work, there exist several optimal deterministic algorithms (e.g., threat-based algorithm (El-Yaniv et al., 2001), threshold-based algorithm (Sun et al., 2020)) that can achieve the optimal worst-case competitive ratio $\alpha^* = O(\ln(M/m))$. Since it is known that randomization does not improve the performance of algorithms for one-way trading problems (El-Yaniv et al., 2001; Im et al., 2021). We focus on deterministic algorithms in this section.

In online search, if the actual maximum price is known in advance, the offline optimal algorithm simply waits until the maximum price to sell the whole resource. Previous work on online search with machine-learned advice has considered point predictions of the maximum price (Sun et al., 2021). Following this prediction paradigm, in our setting, we consider a probabilistic interval prediction $\texttt{PIP}(\ell, u; \delta)$ of the maximal price, which represents a prediction that the maximum price V lies within the interval $[\ell, u]$ with probability at least $1-\delta$. In the following, we design algorithms to minimize $\texttt{DRCR}$ given predictions of the form $\texttt{PIP}(\ell, u; \delta)$.

4.1. An Optimal-Protection-Function Based Algorithm

We first introduce a class of "protection function"-based algorithms ($\texttt{PFA}$) in Algorithm 3. The $\texttt{PFA}$ is parameterized by a protection function $G(v) : [m, M] \to [0, 1]$ that defines the maximum selling amount upon receiving a price $v \in [m, M]$. Then a $\texttt{PFA}$ only sells the resource if the current price v_n is the maximum one among all previous prices, and the selling amount is $G(v_n) - G(\hat{v})$, where $\hat{v}$ is the previous maximum price. $\texttt{PFA}$ can optimally solve the one-way trading problem when the protection function is given by $G(v) = 0, v \in [m, \alpha^* m)$ and $G(v) = \frac{1}{\alpha^*}\ln\frac{v-m}{\alpha^* m - m}, v \in [\alpha^* m, M]$, where $\alpha^* = O(\ln(M/m))$ is the optimal worst-case competitive ratio. Let $\texttt{PFA}(G)$ denote the algorithm with protection function G. In the following, we aim to redesign the protection function G^* for $\texttt{PFA}$ based on the

Algorithm 3 $\texttt{PFA}(G)$: Protection-function-based algorithm

input: protection function G;
initiate running maximum price $\hat{v} = m$;
for $n = 1, \dots, N-1$ **do**
  sell $x_n = [G^*(v_n) - G^*(\hat{v})]^+$;
  update $\hat{v} = \max\{v_n, \hat{v}\}$;
end for
$x_N = 1 - G^*(\hat{v})$.

solution of an optimization problem for a given $\texttt{PIP}$, and show $\texttt{PFA}(G^*)$ can attain the optimal $\texttt{DRCR}$.

Optimization problem based on hard instances. We consider a family of hard instances $\mathcal{H} := \{I_V\}_{V\in[m,M]}$, where I_V includes a sequence of prices that continuously increase from m to V and then drop to the lowest price m in the end. Under any instance from $\{I_V\}_{V\in(v,M]}$, $\texttt{PFA}(G)$ sells $G(v+dv) - G(v)$ amount of resource at price v when the running maximum price increases from v to $v+dv$ for some small dv. For notational convenience, we define a new parameter $q(v) := [G(v+dv) - G(v)]/dv, \forall v \in [m, M]$. The protection function G can be uniquely determined by $\boldsymbol{q} := \{q(v)\}_{v\in[m,M]}$ and the feasible set of $\boldsymbol{q}$ is $\mathcal{Q} = \{\boldsymbol{q} : q(v) \geq 0, \forall v \in [m, M], \int_m^M q(v)dv \leq 1\}$. Since the online decision is irrevocable and all instances in $\{I_V\}_{V\in(v,M]}$ have the same prefix (i.e., the price sequence continuously increasing from m to v), $q(v)$ is the same for all $\{I_V\}_{V\in(v,M]}$. Moreover, note that any online algorithm corresponds to a solution $\boldsymbol{q} := \{q(v)\}_{v\in[m,M]}$ under the hard instances, and thus we can use $\boldsymbol{q}$ to model all online algorithms. Under an instance I_V, the profit of an online algorithm modeled by $\boldsymbol{q}$ is $\texttt{ALG}(\boldsymbol{q}, I_V) = \int_m^V v \cdot q(v)dv + (1 - \int_m^V q(v)dv)m$, where the first term is the profit of selling the item over prices from m to V and the second term is the profit from compulsory selling at the last price. The offline algorithm sells the entire item at the maximum price and thus $\texttt{OPT}(I_V) = V$. Given a $\texttt{PIP}(\ell, u; \delta)$, we can formulate an optimization problem (4) to minimize the $\texttt{DRCR}$ under hard instances. Let $\{\eta^*, \gamma^*, \boldsymbol{q}^*\}$ and $\texttt{CR}^*_{\texttt{OS}}$ denote the optimal solution and the optimal objective value. Based on $\boldsymbol{q}^*$, we can build a protection function $G^*(v) = \int_m^v q^*(s)ds, \forall v \in [m, M]$ and propose $\texttt{PFA}(G^*)$ as the algorithm for online search.

Theorem 3. *Given a $\texttt{PIP}(\ell, u; \delta)$, the $\texttt{DRCR}$ of $\texttt{PFA}(G^*)$ is $\texttt{CR}^*_{\texttt{OS}}$, and $\texttt{CR}^*_{\texttt{OS}}$ is optimal for online search.*

Proof of Theorem 3. Note that $\texttt{PFA}(G^*)$ only sells the resource when the current price is the running maximum one or when it is the last price. Thus, for any instance $I = \{v_n\}_{n\in[N]}$, we can instead focus on a new instance $I' = \{v'_n\}_{n\in[N'+1]}$, where $\{v'_n\}_{n\in[N']}$ is the N' strictly increasing prices of I and $v'_{N'+1} = v_N$. Thus, we can lower bound the profit of $\texttt{PFA}(G^*)$ by

$$\texttt{ALG}(\boldsymbol{q}^*, I) = \texttt{ALG}(\boldsymbol{q}^*, I') \tag{6a}$$

$$= \sum_{n\in[N']} v'_n \int_{v'_{n-1}}^{v'_n} q^*(v)dv + [1 - G^*(v'_{N'})]v'_{N'+1} \tag{6b}$$

$$\geq \sum_{n\in[N']} \int_{v'_{n-1}}^{v'_n} vq^*(v)dv + [1 - G^*(v'_{N'})]m \tag{6c}$$

$$\geq \int_0^{v'_{N'}} vq^*(v)dv + [1 - G^*(v'_{N'})]m. \tag{6d}$$

In addition, we have $\texttt{OPT}(I) = \texttt{OPT}(I') = v'_{N'}$. Since $\boldsymbol{q}^*$ is the optimal solution of the optimization problem (4), we have $\texttt{ALG}(\boldsymbol{q}^*, I) \geq \frac{\texttt{OPT}(I)}{\eta^*}, \forall v'_{N'} \in [\ell, u]$ and $\texttt{ALG}(\boldsymbol{q}^*, I) \geq \frac{\texttt{OPT}(I)}{\gamma^*}, \forall v'_{N'} \in [m, \ell) \cup (u, M]$. And thus the $\texttt{DRCR}$ of $\texttt{PFA}(G^*)$ is $(1-\delta)\eta^* + \delta\gamma^* = \texttt{CR}^*_{\texttt{OS}}$.

Since the parameterized $\texttt{PFA}(G)$ can capture the performance of all online algorithms under hard instances $\mathcal{H}$, based on Proposition 1, it is straightforward to show no online algorithms can achieve a $\texttt{DRCR}$ smaller than $\texttt{CR}^*_{\texttt{OS}}$. □

The optimization (4) is a problem with infinite number of variables and constraints. To obtain the solution, we propose a discrete approximation to solve it. Further, if we let $\hat{G}^*$ denote the protection function built based on the solution of the approximation problem, the following lemma shows that $\texttt{PFA}(\hat{G}^*)$ can achieve a $\texttt{DRCR}$ close to $\texttt{PFA}(G^*)$.

Lemma 2. *For a given parameter $\epsilon > 0$, there exists a discrete approximation problem with $O(\frac{\ln(M/m)}{\ln(1+\epsilon)})$ variables and constraints for the problem (4) of online search. Further, $\texttt{PFA}(\hat{G}^*)$ can achieve $\texttt{DRCR} \leq \texttt{CR}^*_{\texttt{OS}} + \epsilon M/m$.*

5. Learning Algorithms with UQ Prediction

In previous sections, we focused on a *single instance* of an online problem with UQ prediction θ, and designed algorithms to optimize the $\texttt{DRCR}$, i.e., the expected cost ratio under the worst-case distribution in the ambiguity set built by the UQ prediction. However, in practice, the conditional distribution ξ_θ is often not the worst-case one. Furthermore, the $\texttt{PIP}$ may be imprecise due to non-exchangeability of the data or distribution shift, and alternative notions of UQ may be employed, e.g., see (Abdar et al., 2021). For instance, one may approximate the predictive distributions, e.g., through Monte-Carlo methods, but these may be imprecise. In these cases, it may not be tractable to formulate proper ambiguity sets. This motivates us to consider the *multi-instance* setting, using online learning to learn the intrinsic correlation between UQ predictions and instance costs as well as to go beyond the $\texttt{DRCR}$ guarantees.

Online learning formulation The idea is to learn the mapping, or policy, from any given UQ prediction to an online algorithm over T rounds. At the beginning of round $t \in [T]$, we receive a UQ prediction $\theta_t \in \Theta$ about the input instance I_t. Then we select an algorithm parameter $\boldsymbol{w}_t = \pi_t(\theta_t) \in \Omega$ using a chosen policy $\pi_t \in \Pi : \Theta \mapsto \Omega$, and run the online algorithm $A(\boldsymbol{w}_t)$ to execute the instance I_t on the fly. In the end, we observe the entire instance I_t drawn from the unknown conditional distribution ξ_{θ_t}, and the cost function $f_t := f_t(\boldsymbol{w}_t; \theta_t) = \frac{\texttt{ALG}(A(\boldsymbol{w}_t), I_t)}{\texttt{OPT}(I_t)} : \Omega \to \mathbb{R}^+$, which is the cost ratio of the online algorithm $A(\boldsymbol{w}_t)$ and the offline optimal solution under the instance I_t. In our formulation, the goal is to compete against a function $U_t := U_t(\boldsymbol{w}_t; \theta_t)$ that upper bounds the expected cost function $\mathbb{E}_{\xi_{\theta_t}}[f_t(\boldsymbol{w}_t; \theta_t)]$, i.e., $U_t(\boldsymbol{w}_t; \theta_t) \geq \mathbb{E}_{\xi_{\theta_t}}[f_t(\boldsymbol{w}_t; \theta_t)], \forall \boldsymbol{w}_t \in \Omega$. This upper bound exhibits certain properties (e.g. Lipschitzness) that will allow one to conduct online learning on it. We aim to select policies $\{\pi_t\}_{t\in[T]}$, which determine the parameter selection of $\{\boldsymbol{w}_t\}_{t\in[T]}$ based on the UQ predictions $\{\theta_t\}_{t\in[T]}$, to minimize the policy regret over T instances $\texttt{PREG}_T$, i.e.,

$$\sum\nolimits_{t\in[T]} [\mathbb{E}_{\xi_{\theta_t}} f_t(\pi_t(\theta_t); \theta_t) - U_t(\pi^*(\theta_t); \theta_t)], \quad (7)$$

where $\pi^* = \arg\min_\pi \sum_{t\in[T]} U_t(\pi(\theta_t); \theta_t)$. In general, it is impossible to obtain sublinear policy regret if we do not impose any restrictions on the cost functions with respect to the UQ. This is because the instance of each round can only depend on the newly received UQ but may be unrelated to the past observations. Thus, we consider that there is some *local regularity* that encodes the notion that similar UQ predictions should yield similar instance costs. In this paper, we consider cost functions that are L-Lipschitz in θ, i.e., for any $\theta_i, \theta_j \in \Theta$, $\sup_{w\in\Omega} |U_i(w; \theta_i) - U_j(w; \theta_j)| \leq L \cdot \|\theta_i - \theta_j\|$. The goal of the online learning algorithm is to achieve a sublinear regret with respect to *any* cost upper bound function U_t that satisfies the local regularity condition. This allows our approach to be *adaptive* to the inherent difficulty of the problem instance: the closer the expected cost $\mathbb{E}_{\xi_{\theta_t}} f_t$ is to being L-Lipschitz, the tighter the cost upper bound U_t will be to the true expected cost, and the more optimally the algorithm will perform. Furthermore, the DRCR studied in the previous sections is by definition an upper bound of the expected cost. For certain forms of UQ including PIP predictions, the DRCR is Lipschitz with respect to the UQ. This means that our approach can at least compete against the optimal DRCR, and outperform them when distributions are not adversarially given.

Algorithms and results Using the algorithmic framework in (Hazan & Megiddo, 2007), one can obtain sublinear policy regret as we defined. The main idea for the algorithm is to cover the space of UQ prediction Θ with an ϵ-net, where an instance of a sublinear regret algorithm (e.g., randomized exponentiated gradients algorithm) is assigned to each point in the net. Whenever a UQ prediction falls into one of the ϵ-balls, only the algorithm instance assigned to that ball will be run and updated. Thus, every algorithm instance is only run on similar problem instances. In this way, the algorithm exploits local regularities in the UQ space to achieve improved guarantees. Specifically, given that the cost upper bound is L-Lipschitz, we show that an ϵ-net based algorithm can guarantee a sublinear policy regret $\tilde{O}(T^{1-\frac{1}{d+2}})$ for general UQ, where d is the covering dimension of the provided UQs. To attain this result, we extend the algorithm and regret analysis from (Hazan & Megiddo, 2007) by (i) indicating how the algorithm can exploit local regularities other than just Lipschitz policies for convex functions and (ii) refining the regret analysis for competing against cost upper bounds exhibiting Lipschitzness. As concrete examples, we apply the ϵ-net algorithm to derive policy regret guarantees on DRCR for the ski rental and online search problems with PIP $\theta = \{\ell, u; \delta\}$. Indeed, under mild conditions the DRCR exhibits Lipschitzness with respect to θ, which we prove using the optimization problems from previous sections. In particular, for ski-rental we can exploit that $\{\ell, u\}$ are discrete parameters to achieve an improved regret $\tilde{O}(T^{2/3})$ compared to the general guarantees $\tilde{O}(T^{4/5})$. For online search, we introduce a discretization on the UQ space to obtain Lipschitzness and achieve regret $\tilde{O}(T^{4/5})$. The detailed algorithms and results are in Appendix D.

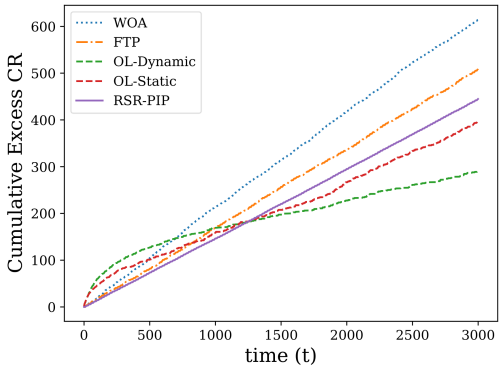

Figure 1. Comparisons of cumulative empirical ratios (minus 1) of the following algorithms: WOA: worst-case optimal randomized algorithm that is $e/e-1$-competitive. FTP: follow-the-prediction algorithm that fully trusts the prediction; OL-Dynamic: online learning with respect to policy regret by leveraging UQ predictions. OL-Static: online learning with respect to static regret without considering UQ predictions. RSR-PIP: randomized algorithm with PIP (Algorithm 3) that achieves the optimal DRCR.

Empirical results Figure 1 compares the empirical competitive ratios (CRs) of our proposed online algorithms in the setting of a multiple-instance ski rental problem. The setup details can be found in Appendix D.6. All our proposed algorithms use UQ to improve the performance compared

to those that are worst-case optimized (i.e., WOA) or just use the predictions blindly (i.e., FTP). Initially, RSR outperforms all other algorithms since RSR is designed to achieve the optimal DRCR, allowing it to perform well before the online learning approaches have had time to learn. As the number of instances increases, the cumulative CR of our proposed online learning algorithm OL-Dynamic increases sublinearly, gradually approaching and then outperforming RSR. This is because the distribution used to generate the problem instances is not the worst-case one for the given UQ. Thus, OL-Dynamic can better learn to use UQ for non-worst-case distributions, while RSR, designed toward this worst case, performs more conservatively over the long run. This emphasizes the importance of the online learning approaches in multiple-instance settings in real-world applications, where adversarial distributions rarely occur. In addition, the online learning algorithm OL-Static, which is designed for static regret, can also gradually learn to achieve a performance comparable to the optimal DRCR solution but fails to improve much beyond it. This further validates the importance of our policy regret guarantees compared to the classic static regret, which can be obtained without UQ.

6. Concluding Remarks

This paper has developed two paradigms for incorporating uncertainty-quantified predictions into the design and analysis of online algorithms. For UQ predictions that are descriptive and enable a tractable ambiguity set about the future input to be constructed, we have proposed an optimization-based approach that utilizes the predictions and minimizes a form of distributionally-robust competitive ratio on a per-instance basis. We applied this approach to design optimal online algorithms for two classic online problems with UQ predictions: ski rental and online search problems. Additionally, we devised an online learning approach that can learn to utilize the predictions across multiple instances and attain sublinear regret under mild Lipschitz conditions. We posit that both these paradigms for incorporating uncertainty-quantified advice in online decision-making hold promise for designing algorithms using UQ for other online problems, and can enable better and more reliable use of machine learning in general online decision-making.

Acknowledgements

Bo Sun and Raouf Boutaba acknowledge the NSERC Discovery Grant RGPIN-2019-06587. Jerry Huang is supported by a Caltech Summer Undergraduate Fellowship and the Kiyo and Eiko Tomiyasu SURF Fund. Nicolas Christianson and Adam Wierman acknowledge the support from an NSF Graduate Research Fellowship (DGE-2139433) and NSF Grants CNS-2146814, CPS-2136197, CNS-2106403, and NGSDI-2105648. The work of Mohammad Hajiesmaili is supported by NSF Grants CPS-2136199, CNS-2106299, CNS-2102963, CCF-2325956, and CAREER-2045641. We also thank Yiheng Lin and Yisong Yue for insightful discussions.

Impact Statement

This paper presents research concerned with utilizing uncertainty quantification to improve machine learning-augmented online decision-making. While there are a number of potential societal consequences of our work, we do not feel these must be specifically highlighted here.

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., and Nahavandi, S. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, December 2021.

Agrawal, S. and Goyal, N. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pp. 127–135. PMLR, 2013.

Anand, K., Ge, R., and Panigrahi, D. Customizing ML Predictions for Online Algorithms. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 303–313. PMLR, November 2020.

Antoniadis, A., Coester, C., Elias, M., Polak, A., and Simon, B. Online metric algorithms with untrusted predictions. In *International Conference on Machine Learning*, pp. 345–355. PMLR, 2020.

Balcan, M.-F. Data-driven algorithm design. *arXiv preprint arXiv:2011.07177*, 2020.

Balcan, M.-F., Dick, T., and Vitercik, E. Dispersion for data-driven algorithm design, online learning, and private optimization. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 603–614. IEEE, 2018.

Balseiro, S., Kroer, C., and Kumar, R. Single-leg revenue management with advice. In *Proceedings of the 24th ACM Conference on Economics and Computation*, pp. 207–207, 2023.

Bastani, H. and Bayati, M. Online decision making with high-dimensional covariates. *Operations Research*, 68 (1):276–294, 2020.

Besbes, O., Ma, W., and Mouchtaki, O. Beyond IID: Data-driven decision-making in heterogeneous environments. In *Advances in Neural Information Processing Systems*, May 2022.

Borodin, A. and El-Yaniv, R. *Online computation and competitive analysis*. Cambridge University Press, 2005.

Christianson, N., Shen, J., and Wierman, A. Optimal robustness-consistency tradeoffs for learning-augmented metrical task systems. In *International Conference on Artificial Intelligence and Statistics*, pp. 9377–9399. PMLR, 2023.

Dekel, O., flajolet, a., Haghtalab, N., and Jaillet, P. Online Learning with a Hint. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Diakonikolas, I., Kontonis, V., Tzamos, C., Vakilian, A., and Zarifis, N. Learning online algorithms with distributional advice. In *International Conference on Machine Learning*, pp. 2687–2696. PMLR, 2021.

El-Yaniv, R., Fiat, A., Karp, R. M., and Turpin, G. Optimal search and one-way trading online algorithms. *Algorithmica*, 30:101–139, 2001.

Fujiwara, H. and Iwama, K. Average-case competitive analyses for ski-rental problems. *Algorithmica*, 42:95–107, 2005.

Fujiwara, H., Iwama, K., and Sekiguchi, Y. Average-case competitive analyses for one-way trading. *Journal of combinatorial optimization*, 21(1):83–107, 2011.

Gupta, A., Panigrahi, D., Subercaseaux, B., and Sun, K. Augmenting online algorithms with ε-accurate predictions. *Advances in Neural Information Processing Systems*, 35:2115–2127, 2022.

Hazan, E. and Megiddo, N. Online learning with prior knowledge. In Bshouty, N. H. and Gentile, C. (eds.), *Learning Theory*, pp. 499–513, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

Im, S., Kumar, R., Montazer Qaem, M., and Purohit, M. Online knapsack with frequency predictions. *Advances in Neural Information Processing Systems*, 34:2733–2743, 2021.

Karlin, A. R., Manasse, M. S., Rudolph, L., and Sleator, D. D. Competitive snoopy caching. *Algorithmica*, 3: 79–119, 1988.

Karlin, A. R., Manasse, M. S., McGeoch, L. A., and Owicki, S. Competitive randomized algorithms for non-uniform problems. In *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pp. 301–309, 1990.

Khodak, M., Balcan, M.-F., Talwalkar, A., and Vassilvitskii, S. Learning predictions for algorithms with predictions, 2022.

Kuzborskij, I. and Cesa-Bianchi, N. Locally-adaptive nonparametric online learning, 2020.

Lorenz, J., Panagiotou, K., and Steger, A. Optimal algorithms for k-search with application in option pricing. *Algorithmica*, 55(2):311–328, 2009.

Lykouris, T. and Vassilvitskii, S. Competitive caching with machine learned advice. *Journal of the ACM (JACM)*, 68 (4):1–25, 2021.

Mahdian, M., Nazerzadeh, H., and Saberi, A. Online Optimization with Uncertain Information. *ACM Transactions on Algorithms*, 8(1):1–29, January 2012. ISSN 1549-6325, 1549-6333. doi: 10.1145/2071379.2071381.

Marusich, L. R., Bakdash, J. Z., Zhou, Y., and Kantarcioglu, M. Using ai uncertainty quantification to improve human decision-making. *arXiv preprint arXiv:2309.10852*, 2023.

McMahan, H. B. A survey of algorithms and analysis for adaptive online learning, 2015.

Mitzenmacher, M. and Vassilvitskii, S. Algorithms with predictions, 2020.

Mitzenmacher, M. and Vassilvitskii, S. Algorithms with predictions. *Communications of the ACM*, 65(7):33–35, 2022.

Papadopoulos, H., Proedrou, K., Vovk, V., and Gammerman, A. Inductive confidence machines for regression. In *European Conference on Machine Learning*, 2002.

Purohit, M., Svitkina, Z., and Kumar, R. Improving online algorithms via ml predictions. *Advances in Neural Information Processing Systems*, 31, 2018.

Shalev-Shwartz, S. Online learning and online convex optimization. *Found. Trends Mach. Learn.*, 4(2):107–194, feb 2012.

Slivkins, A. Contextual bandits with similarity information. In *Proceedings of the 24th annual Conference On Learning Theory*, pp. 679–702. JMLR Workshop and Conference Proceedings, 2011.

Sun, B., Zeynali, A., Li, T., Hajiesmaili, M., Wierman, A., and Tsang, D. H. Competitive algorithms for the online multiple knapsack problem with application to electric vehicle charging. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(3): 1–32, 2020.

Sun, B., Lee, R., Hajiesmaili, M., Wierman, A., and Tsang, D. Pareto-optimal learning-augmented algorithms for online conversion problems. *Advances in Neural Information Processing Systems*, 34:10339–10350, 2021.

Sun, C., Liu, S., and Li, X. Maximum optimality margin: A unified approach for contextual linear programming and inverse linear programming. In *International Conference on Machine Learning*, pp. 32886–32912. PMLR, 2023.

Vovk, V. and Bendtsen, C. Conformal predictive decision making. In *Conformal and Probabilistic Prediction and Applications*, pp. 52–62. PMLR, 2018.

Vovk, V., Gammerman, A., and Saunders, C. Machine-learning applications of algorithmic randomness. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 444–453, 1999.

Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

Wei, A. and Zhang, F. Optimal robustness-consistency trade-offs for learning-augmented online algorithms. *Advances in Neural Information Processing Systems*, 33: 8042–8053, 2020.

Étienne Bamas, Maggiori, A., and Svensson, O. The primal-dual method for learning augmented algorithms, 2020.

A. Proof of Proposition 1

Given a probabilistic interval prediction $\texttt{PIP}(\ell, u; \delta)$, for any parameterized algorithm $A(\bar{\boldsymbol{w}})$, $(\bar{\boldsymbol{w}} \in \Omega)$, let $\bar{\eta}$ and $\bar{\gamma}$ denote its consistency and robustness. By definition (1), we have

$$\bar{\eta} = \max_{I \in \mathcal{I}_{\ell,u}} \frac{\texttt{ALG}(\bar{\boldsymbol{w}}, I)}{\texttt{OPT}(I)} \text{ and } \bar{\gamma} = \max_{I \in \mathcal{I}} \frac{\texttt{ALG}(\bar{\boldsymbol{w}}, I)}{\texttt{OPT}(I)},$$

where $\mathcal{I}$ and $\mathcal{I}_{\ell,u}$ are the entire instance set and the instance subset that contains all instances confirming with the interval prediction, respectively. Then we have $\mathcal{H} \subseteq \mathcal{I}$ and $\mathcal{H}_{\ell,u} \subseteq \mathcal{I}_{\ell,u}$ since $\mathcal{H}$ and $\mathcal{H}_{\ell,u}$ only contain hard instances. Thus, $\{\bar{\eta}, \bar{\gamma}, \bar{\boldsymbol{w}}\}$ is a feasible solution of the optimization problem (4). Consequently, the $\texttt{DRCR}$ of $A(\bar{\boldsymbol{w}})$ is $(1-\delta)\bar{\eta} + \delta\bar{\gamma} \geq (1-\delta)\eta^* + \delta\gamma^*$, where η^* and γ^* are the optimal solution of the problem (4). Therefore, the $\texttt{DRCR}$ of parameterized algorithms is lower bounded by $(1-\delta)\eta^* + \delta\gamma^*$.

B. Technical Proofs and Supplementary Results for Ski Rental with UQ Predictions

B.1. Proof of Theorem 1

To design deterministic algorithms for the ski rental problem, we can first derive the distributionally-robust competitive ratio ($\texttt{DRCR}$) when the buying strategy Y operates in different prediction regions, and then choose the strategy that can minimize the $\texttt{DRCR}$. For notational convenience, we let $\texttt{ALG}$ and $\texttt{OPT}$ denote the cost of the online algorithm and the cost of offline algorithm, and let $\texttt{CR}$ be the cost ratio of $\texttt{ALG}$ and $\texttt{OPT}$.

Case I: $P < B$. We derive the cost ratios when Y falls in different regions.

Case I(a): when $0 < Y \leq P$**,**

- if $0 < N < Y$, we have $\texttt{ALG} = \texttt{OPT} = N$; and $\texttt{CR} = 1$;
- if $Y \leq N < P$, we have $\texttt{ALG} = Y + B$ and $\texttt{OPT} = N$; and thus $\texttt{CR} = \frac{Y+B}{N} \leq \frac{Y+B}{Y}$;
- if $N = P$, we have $\texttt{ALG} = Y + B$ and $\texttt{OPT} = P$; and thus $\texttt{CR} = \frac{Y+B}{P}$;
- if $N > P$, we have $\texttt{ALG} = Y + B$ and $\texttt{OPT} = \min\{N, B\}$; and thus $\texttt{CR} = \frac{Y+B}{\min\{N,B\}} \leq \frac{Y+B}{P}$.

Thus, we have $\texttt{DRCR} = (1-\delta)\frac{Y+B}{P} + \delta\frac{Y+B}{Y}$, which is minimized by $Y = \begin{cases} \sqrt{\frac{\delta}{1-\delta}BP} & \delta \in [0, \frac{P}{P+B}] \\ P & \delta \in (\frac{P}{P+B}, 1] \end{cases}$, and the corresponding $\texttt{DRCR}$ is $\begin{cases} 2\sqrt{\delta(1-\delta)\frac{B}{P}} + (1-\delta)\frac{B}{P} + \delta, & \delta \in [0, \frac{P}{P+B}] \\ 1 + \frac{B}{P} & \delta \in (\frac{P}{P+B}, 1] \end{cases}$.

Case I(b): when $P < Y \leq B$**,**

- if $0 < N < Y$, we have $\texttt{ALG} = \texttt{OPT} = N$; and $\texttt{CR} = 1$;
- if $Y \leq N$, we have $\texttt{ALG} = Y + B$ and $\texttt{OPT} = \min\{N, B\}$; and thus $\texttt{CR} = \frac{Y+B}{\min\{N,B\}} \leq \frac{Y+B}{Y}$.

Thus, we have $\texttt{DRCR} = 1 - \delta + \delta\frac{Y+B}{Y}$, which is minimized when $Y = B$ and $\texttt{DRCR} = 1 + \delta$.

Case I(c): when $Y > B$**,**

- if $0 < N \leq B$, we have $\texttt{ALG} = \texttt{OPT} = N$; and $\texttt{CR} = 1$;
- if $B < N < Y$, we have $\texttt{ALG} = N$ and $\texttt{OPT} = B$; and thus $\texttt{CR} = \frac{N}{B} \leq \frac{Y}{B}$;
- if $N \geq Y$, we have $\texttt{ALG} = Y + B$ and $\texttt{OPT} = B$; and thus $\texttt{CR} = \frac{Y+B}{B}$.

Thus, we have $\texttt{DRCR} = 1 - \delta + \delta\frac{Y+B}{Y}$, which is minimized when $Y = B$ and $\texttt{DRCR} = 1 + \delta$.

By comparing the DRCR of three sub-cases, $Y = B$ minimizes the DRCR when $P < B$, and the minimum DRCR is $1 + \delta$.

Case II: $P > B$. Consider the following sub-cases.

Case II(a): when $0 < Y \leq B$**,**

- if $0 < N < Y$, we have $\texttt{ALG} = \texttt{OPT} = N$; and $\texttt{CR} = 1$;
- if $Y \leq N \leq B$, we have $\texttt{ALG} = Y + B$ and $\texttt{OPT} = N$; and thus $\texttt{CR} = \frac{Y+B}{N} \leq \frac{Y+B}{Y}$;
- if $B < N$, we have $\texttt{ALG} = Y + B$ and $\texttt{OPT} = B$; and thus $\texttt{CR} = \frac{Y+B}{B}$.

Thus, we have $\texttt{DRCR} = (1-\delta)\frac{Y+B}{B} + \delta\frac{Y+B}{Y}$, which is minimized when $Y = B \cdot \min\{\sqrt{\delta/(1-\delta)}, 1\}$, and $\texttt{DRCR} := \chi(\delta) = \begin{cases} 2\sqrt{\delta(1-\delta)} + 1 & \delta \in [0, \frac{1}{2}] \\ 2 & \delta \in (\frac{1}{2}, 1] \end{cases}$.

Case II(b): when $B < Y \leq P$**,**

- if $0 \leq N \leq B$, we have $\texttt{ALG} = \texttt{OPT} = N$; and $\texttt{CR} = 1$;
- if $B < N < Y$, we have $\texttt{ALG} = N$ and $\texttt{OPT} = B$; and thus $\texttt{CR} = \frac{N}{B} < \frac{Y}{B}$;
- if $Y \leq N \leq P$, we have $\texttt{ALG} = Y + B$ and $\texttt{OPT} = B$; and thus $\texttt{CR} = \frac{Y+B}{B}$;
- if $P < N$, we have $\texttt{ALG} = Y + B$ and $\texttt{OPT} = B$; and thus $\texttt{CR} = \frac{Y+B}{B}$.

Thus, we have $\texttt{DRCR} = \frac{Y+B}{B}$, which is minimized when $Y \to B$ and $\texttt{DRCR} \to 2$.

Case II(c): when $Y > P$**,**

- if $0 < N \leq B$, we have $\texttt{ALG} = \texttt{OPT} = N$; and $\texttt{CR} = 1$;
- if $B < N \leq P$, we have $\texttt{ALG} = N$ and $\texttt{OPT} = B$; and thus $\texttt{CR} = \frac{N}{B} \leq \frac{P}{B}$;
- if $P < N < Y$, we have $\texttt{ALG} = N$ and $\texttt{OPT} = B$; and thus $\texttt{CR} = \frac{N}{B} < \frac{Y}{B}$;
- if $N \geq Y$, we have $\texttt{ALG} = Y + B$ and $\texttt{OPT} = B$; and thus $\texttt{CR} = \frac{Y+B}{B}$.

Thus, we have $\texttt{DRCR} = (1-\delta)\frac{P}{B} + \delta\frac{Y+B}{B}$, which is minimized when $Y \to P$, and $\texttt{DRCR} \to \delta + \frac{P}{B}$.

By comparing the DRCR of the above three sub-cases, we have when $P > \frac{\sqrt{5}+1}{2}B$, $\chi(\delta) \leq \delta + \frac{P}{B}, \forall\delta \in [0,1]$ and thus, the optimal buying day is $Y = B\min\{\sqrt{\delta/(1-\delta)}, 1\}$. When $B \leq P \leq \frac{\sqrt{5}+1}{2}B$, the optimal buy strategy is $Y = B\min\{\sqrt{\delta/(1-\delta)}, 1\}$ if $\chi(\delta) \leq \delta + \frac{P}{B}$ and $Y = P$ otherwise.

B.2. Deterministic Algorithms with Probabilistic Interval Predictions

We can extend the ideas of DSR to incorporate a probabilistic interval prediction $\texttt{PIP}(\ell, u; \delta)$. The extension adheres to a decision structure similar to that of DSR when dealing with predicted intervals that clearly favor buying decisions (i.e., $B < \ell \leq u$) or renting decisions (i.e., $\ell < u \leq B$). The additional complexity arises when $\ell \leq B \leq u$. In such cases, the optimal decision greatly depends on the prediction interval and its quality, and this leads to three possible scenarios: (i) making a pro-rent decision by purchasing at the predicted interval upper bound u, (ii) opting for a pro-buy decision by purchasing within the first ℓ days , or (iii) buying on day B when the prediction is proved to be unhelpful. The full algorithm is presented in Algorithm 4 and its DRCR result is presented in Lemma 4. The proof of Lemma 4 follows the same proof idea as Theorem 1.

Algorithm 4 Online deterministic algorithm with PIP for ski rental

input: prediction $\texttt{PIP}(\ell, u; \delta)$, buying cost B;
if $\ell \le u < B$ **then**
  set $Y = B$;
else if $B < \ell \le u$ **then**
  if $\chi(\delta) \le \delta + \frac{u}{B}$ **then**
    set $Y = B \cdot \min\{\sqrt{\delta/(1-\delta)}, 1\}$;
  else
    set $Y = u$;
  end if
else if $\ell \le B \le u$ **then**
  if $\zeta(\delta, \ell) \ge 2$ and $\delta + \frac{u}{B} \ge 2$ **then**
    set $Y = B$;
  else if $\zeta(\delta, \ell) \le \delta + \frac{u}{B}$ **then**
    set $Y = \ell \cdot \min\{\sqrt{B\delta/(\ell(1-\delta))}, 1\}$;
  else
    set $Y = u$;
  end if
end if
buy skis on day Y.

Theorem 4. *Given a* $\texttt{PIP}(\ell, u; \delta)$*, Algorithm 4 for ski rental achieves the* DRCR

$$\begin{cases} 1+\delta & \ell \le u < B \\ \min\left\{\chi(\delta), \delta + \frac{u}{B}\right\} & B < \ell \le u \\ \min\left\{\zeta(\delta, \ell), \delta + \frac{u}{B}, 2\right\} & \ell \le B \le u \end{cases}, \tag{8}$$

where

$$\zeta(\delta, \ell) := \begin{cases} \delta + (1-\delta)B/\ell + 2\sqrt{\delta(1-\delta)B/\ell} & \delta \in [0, \frac{\ell}{\ell+B}) \\ 1 + B/\ell & \delta \in [\frac{\ell}{\ell+B}, 1] \end{cases}.$$

Further, the attained DRCR *is optimal for all deterministic algorithms for ski rental with probabilistic interval predictions.*

B.3. Proof of Theorem 2

First, the optimization formulation for the ski rental problem with a probabilistic interval prediction $\texttt{PIP}(\ell, u; \delta)$ can be formally stated as follows:

$$\min_{\eta, \gamma, \boldsymbol{y}} \quad (1-\delta)\eta + \delta\gamma \tag{9a}$$

$$\text{s.t.} \quad \sum_{t=1}^{N}(B+t-1)y(t) + N\sum_{t=N+1}^{\infty} y(t) \le \eta \min\{N, B\}, \forall N \in \mathbb{Z}^+_{\ell,u}, \tag{9b}$$

$$\sum_{t=1}^{N}(B+t-1)y(t) + N\sum_{t=N+1}^{\infty} y(t) \le \gamma \min\{N, B\}, \forall N \in \mathbb{Z}^+ \setminus \mathbb{Z}^+_{\ell,u}, \tag{9c}$$

$$1 \le \eta \le \gamma, \tag{9d}$$

$$\boldsymbol{y} \in \mathcal{Y}. \tag{9e}$$

Each constraint N from either constraint (9b) or constraint (9c) ensures that the ratio between the expected cost of the online algorithm and the cost of the offline optimum is upper bounded by η or γ, respectively. The constraint $1 \le \eta \le \gamma$ guarantees that $\sum_{t=1}^{N}(B+t-1)y(t) + N\sum_{t=N+1}^{\infty} y(t) \le \gamma \min\{N, B\}, \forall N \in \mathbb{Z}^+_{\ell,u}$, that corresponds to the constraint (4c) in the optimization problem (4).

Given any instance I_N, the expected cost of $\texttt{RSR}(\boldsymbol{y}^*)$ is $\texttt{ALG}(\boldsymbol{y}^*, I_N) = \sum_{t=1}^{N}(B+t-1)y^*(t) + N\sum_{t=N+1}^{\infty} y^*(t)$, and the offline optimal cost is $\texttt{OPT}(I_N) = \min\{N, B\}$. Thus, based on the definition in Equation (3), η^* and γ^* are the

consistency and robustness of the algorithm with untrusted interval prediction $[\ell, u]$, respectively. And the $\texttt{DRCR}$ of $\texttt{RSR}(\boldsymbol{y}^*)$ is $(1-\delta)\eta^* + \delta\gamma^* = \texttt{CR}^*_{\texttt{sr}}$.

To show the optimality of the result, we note that the hard instance set $\mathcal{H}$ used for formulating the problem is in fact the entire instance set $\mathcal{I} = \{I_N\}_{N\in\mathbb{Z}^+}$ for the ski rental problem, and the parameterized algorithms $\texttt{RSR}(\boldsymbol{y})$ can capture all online algorithms under $\mathcal{H}$. Thus, based on Proposition 1, no online algorithms can achieve a $\texttt{DRCR}$ smaller than $\texttt{CR}^*_{\texttt{sr}}$.

B.4. Proof of Lemma 1

The proof is based on the optimization formulation (9). Let $\mathcal{C}_N$ denote the constraint indexed by N from the constraints (9b) and (9c). Then the problem (9) can be reduced to an optimization with a finite number of constraints and variables. Particularly, variables $\{y(t)\}_{t\in\mathbb{Z}^+}$ and constraints $\{\mathcal{C}_N\}_{N\in\mathbb{Z}^+}$ (i.e., constraints (9b) and (9c)) can be reduced as follows:

- when $\ell \le u < B$, only B variables $\{y(t)\}_{t\in\mathbb{Z}^+_{1,B}}$ and B constraints $\{\mathcal{C}_N\}_{N\in\mathbb{Z}^+_{1,B}}$ are non-redundant;
- when $B < \ell \le u$ or $\ell \le B \le u$, only $B+1$ variables $\{y(t)\}_{t\in\mathbb{Z}^+_{1,B}\cup\{u+1\}}$ and $B+1$ constraints $\{\mathcal{C}_N\}_{N\in\mathbb{Z}^+_{1,B-1}\cup\{u,u+1\}}$ are non-redundant.

We start by showing the structural property of constraints (9b) and (9c). Let $\texttt{RHS}(N)$ and $\texttt{LHS}(\boldsymbol{y}, N)$ denote the right-hand-side and the left-hand-side of the constraint $\mathcal{C}_N$, respectively. Given a feasible solution $\{\eta, \gamma, \boldsymbol{y}\}$, if there exists a $k \in \mathbb{Z}^+$ such that $\texttt{RHS}(k) \ge \texttt{RHS}(k+1)$, we can move the probability mass from $y(k+1)$ to $y(k)$ and obtain a new solution $\hat{\boldsymbol{y}}$, i.e.,

$$\hat{y}(t) = \begin{cases} y(t) + y(t+1) & t = k \\ 0 & t = k+1 \\ y(t) & \text{otherwise} \end{cases},$$

and the solution $\{\eta, \gamma, \hat{\boldsymbol{y}}\}$ is also feasible to the problem (9). To show this, we make the following claims.

Claim 1. $\{\eta, \gamma, \hat{\boldsymbol{y}}\}$ satisfies the constraints $\{\mathcal{C}_N\}_{N\in\mathbb{Z}^+_{1,k-1}}$.

Note that $\texttt{LHS}(\boldsymbol{y}, N) = \sum_{t=1}^{N}(B+t-1)y(t) + N[1-\sum_{t=1}^{N} y(t)]$. Then we have $\texttt{LHS}(\hat{\boldsymbol{y}}, N) = \texttt{LHS}(\boldsymbol{y}, N), \forall N \le k-1$, and thus the first $k-1$ constraints are feasible for $\hat{\boldsymbol{y}}$.

Claim 2. $\{\eta, \gamma, \hat{\boldsymbol{y}}\}$ satisfies the constraints $\{\mathcal{C}_N\}_{N\in\mathbb{Z}^+_{k+1,\infty}}$.

When $N \ge k+1$, we have

$$\texttt{LHS}(\hat{\boldsymbol{y}}, N) - \texttt{LHS}(\boldsymbol{y}, N) = (B+k-1)[\hat{y}(k) - y(k)] - (B+k)y(k+1) = -y(k+1) \le 0,$$

and thus all constraints after $k+1$ are feasible for $\hat{\boldsymbol{y}}$.

Claim 3. $\{\eta, \gamma, \hat{\boldsymbol{y}}\}$ satisfies the constraint $\mathcal{C}_k$.

Note that $\texttt{LHS}(\hat{\boldsymbol{y}}, k)$ is dominated by $\texttt{LHS}(\hat{\boldsymbol{y}}, k+1)$ since

$$\texttt{LHS}(\hat{\boldsymbol{y}}, k+1) - \texttt{LHS}(\hat{\boldsymbol{y}}, k) = 1 - \sum_{t=1}^{k} \hat{y}(t) + (B-1)\hat{y}(k+1) > 0.$$

Therefore, if $\texttt{RHS}(k) \ge \texttt{RHS}(k+1)$ and $\mathcal{C}_{k+1}$ is satisfied, $\mathcal{C}_k$ is also feasible for $\hat{\boldsymbol{y}}$.

Combining above three claims gives the structural property. Next, we show the reduction of variables $\{y(t)\}_{t\in\mathbb{Z}^+}$ and constraints $\{\mathcal{C}_N\}_{N\in\mathbb{Z}^+}$ in the problem (9) as follows.

Case I: $\ell \le u < B$. In this case, $\texttt{RHS}(N)$ remains the same when $N \ge B$. Therefore, we can iteratively move all the probability mass from $\{y(t)\}_{t\in\mathbb{Z}^+_{B+1,\infty}}$ to $y(B)$ and obtain the new feasible solution

$$\hat{y}(t) = \begin{cases} y(t) & t \in \mathbb{Z}^+_{1,B-1} \\ \sum_{t=B+1}^{\infty} y(t) & t = B \\ 0 & t \in \mathbb{Z}^+_{B+1,\infty} \end{cases}.$$

Since $\hat{y}(t) = 0, \forall t \in \mathbb{Z}^+_{B+1,\infty}$, all variables $\{y(t)\}_{t\in\mathbb{Z}^+_{B+1,\infty}}$ and constraints $\{\mathcal{C}_N\}_{N\in\mathbb{Z}^+_{B+1,\infty}}$ are redundant. Thus, we only need to focus on B variables $\{y(t)\}_{t\in\mathbb{Z}^+_{1,B}}$ and B constraints $\{\mathcal{C}_N\}_{N\in\mathbb{Z}^+_{1,B}}$.

Case II: $B < \ell \leq u$**.** Note that (i) $\texttt{RHS}(N) = \gamma B$ remains the same when $N \in \mathbb{Z}^+_{u+1,\infty}$; and (ii) $\texttt{RHS}(N)$ is non-increasing in N when $N \in \mathbb{Z}^+_{B,u}$ since we have $\texttt{RHS}(N) = \gamma B, N \in \mathbb{Z}^+_{B,\ell-1}$ and $\texttt{RHS}(N) = \eta B, N \in \mathbb{Z}^+_{\ell,u}$. Based on the structural property, we can iteratively move the probability mass from $\{y(t)\}_{t\in\mathbb{Z}^+_{u+2,\infty}}$ to $y(u+1)$, and from $\{y(t)\}_{t\in\mathbb{Z}^+_{B+1,u}}$ to $y(B)$. This gives a new feasible solution

$$\hat{y}(t) = \begin{cases} y(t) & t \in \mathbb{Z}^+_{1,B-1} \\ \sum_{t=B}^{u} y(t) & t = B \\ 0 & t = B+1, \ldots, u \\ \sum_{t=u+1}^{\infty} y(t) & t = u+1 \\ 0 & t \in \mathbb{Z}^+_{u+2,\infty} \end{cases}. \tag{10}$$

Thus, we can just focus on the $B+1$ variables $\{y(t)\}_{t\in\mathbb{Z}^+_{1,B}\cup\{u+1\}}$.

For $N \in \mathbb{Z}^+_{B,u}$, note that $\texttt{LHS}(\hat{\boldsymbol{y}}, N)$ is non-decreasing and $\texttt{RHS}(N)$ is non-increasing in N. Thus, the last constraint $\mathcal{C}_u$ is the most difficulty one and we can just focus on $\mathcal{C}_u$. For $N \in \mathbb{Z}^+_{u+1,\infty}$, we can just focus on the constraint $\mathcal{C}_{u+1}$ since $\hat{y}(t) = 0, \forall t \in \mathbb{Z}^+_{u+2,\infty}$. Thus, we only need to consider the constraints $\{\mathcal{C}_N\}_{N\in\mathbb{Z}^+_{1,B-1}\cup\{u,u+1\}}$.

Case III: $\ell \leq B \leq u$**.** In this case, we have that (i) $\texttt{RHS}(N) = \gamma B$ remains the same when $N \in \mathbb{Z}^+_{u+1,\infty}$; and (ii) $\texttt{RHS}(N) = \eta B$ remains the same when $N \in \mathbb{Z}^+_{B,u}$. Then, the probability mass $\{y(t)\}_{t\in\mathbb{Z}^+_{u+2,\infty}}$ can be moved to $y(u+1)$, and the probability mass $\{y(t)\}_{t\in\mathbb{Z}^+_{B+1,u}}$ can be moved to $y(B)$. Thus, a new feasible solution $\hat{\boldsymbol{y}}$ is also given in the same form as Equation (10) and we can focus on the $B+1$ variables $\{y(t)\}_{t\in\mathbb{Z}^+_{1,B}\cup\{u+1\}}$.

For $N \in \mathbb{Z}^+_{B,u}$, $\texttt{LHS}(\hat{\boldsymbol{y}}, N)$ is non-decreasing and $\texttt{RHS}(N)$ is constant in N. We can just focus on the last constraint $\mathcal{C}_u$. For $N \in \mathbb{Z}^+_{u+1,\infty}$, we can just focus on the constraint $\mathcal{C}_{u+1}$ since $\hat{y}(t) = 0, \forall t \in \mathbb{Z}^+_{u+2,\infty}$. Thus, we only need to consider the constraints $\{\mathcal{C}_N\}_{N\in\mathbb{Z}^+_{1,B-1}\cup\{u,u+1\}}$.

Combining *Case I - Case III*, there are no more than $B+1$ non-redundant variables in $\{y(t)\}_{t\in\mathbb{Z}^+}$, and $B+1$ constraints in $\{\mathcal{C}_N\}_{N\in\mathbb{Z}^+}$. This completes the proof.

C. Technical Proofs and Supplementary Results for Online Search with UQ Predictions

C.1. Discrete Approximation for the Optimization Problem

We start by providing a formal formulation for the online search with a probabilistic interval prediction $\texttt{PIP}(\ell, u; \delta)$ under the hard instances $\mathcal{H} := \{I_V\}_{V\in[m,M]}$.

$$\min_{\eta,\gamma,\boldsymbol{q}} \quad (1-\delta)\eta + \delta\gamma \tag{11a}$$

$$\text{s.t.} \quad V \leq \eta \left[\int_m^V v \cdot q(v)dv + (1 - \int_m^V q(v)dv)m\right], V \in [\ell, u], \tag{11b}$$

$$V \leq \gamma \left[\int_m^V v \cdot q(v)dv + (1 - \int_m^V q(v)dv)m\right], V \in [m, \ell) \cup (u, M], \tag{11c}$$

$$1 \leq \eta \leq \gamma, \tag{11d}$$

$$\boldsymbol{q} \in \mathcal{Q}. \tag{11e}$$

Each constraint indexed by V in Equation (11b) (or in Equation (11c)) ensures the ratio between the profit of the offline optimum and the profit of the online algorithm is upper bounded by η (or γ) under the instance I_V. Thus, η and γ represent the consistency and robustness (under the hard instances), and the optimization objective is to minimize the $\texttt{DRCR}$ under the hard instances. Let $\{\boldsymbol{q}^*, \eta^*, \gamma^*\}$ and $\texttt{CR}^*_{\texttt{OS}}$ denote the optimal solution and the optimal objective value of the above problem.

We propose a discrete approximation for the problem (11) as follows. Fix a parameter $\epsilon > 0$. Let K' be the largest integer such that $m(1+\epsilon)^{K'} \leq M$, i.e., $K' = \lfloor \frac{\ln(M/m)}{\ln(1+\epsilon)} \rfloor$. Consider the following $K = K' + 4$ discrete values that include $K'+1$ values $\{m(1+\epsilon)^k\}_{k=0,\ldots,K'}$, and three additional values ℓ, u, M. We arrange these K values in a non-decreasing order and let V_k denote the k-th value. Particularly, we have $V_1 = m, V_K = M$, and we define index k_ℓ and k_u such that $V_{k_\ell} = \ell$ and $V_{k_u} = u$. The key property of the defined discrete values is that $V_k/V_{k-1} \leq 1+\epsilon, \forall k = 2, \ldots, K$. We define K variables $\hat{\boldsymbol{q}} := \{\hat{q}_k\}_{k\in[K]}$ and its feasible set $\hat{\mathcal{Q}} := \{\hat{\boldsymbol{q}} : \hat{q}_k \geq 0, \forall k \in [K], \sum_{k\in[K]} \hat{q}_k \leq 1\}$. Then we consider the following discrete version of the problem (11).

$$\min_{\hat{\eta},\hat{\gamma},\hat{\boldsymbol{q}}} \quad (1-\delta)\hat{\eta} + \delta\hat{\gamma} \tag{12a}$$

$$\text{s.t.} \quad V_k \leq \hat{\eta}\left[\sum\nolimits_{i=1}^{k} V_i\hat{q}_i + (1-\sum\nolimits_{i=1}^{k}\hat{q}_i)m\right], k = k_\ell, \ldots, k_u, \tag{12b}$$

$$V_k \leq \hat{\gamma}\left[\sum\nolimits_{i=1}^{k} V_i\hat{q}_i + (1-\sum\nolimits_{i=1}^{k}\hat{q}_i)m\right], k = 1, \ldots, k_\ell - 1, k_u + 1, \ldots, K, \tag{12c}$$

$$1 \leq \hat{\eta} \leq \hat{\gamma}, \tag{12d}$$

$$\hat{\boldsymbol{q}} \in \hat{\mathcal{Q}}, \tag{12e}$$

which only contains K variables in $\hat{\boldsymbol{q}}$ and K constraints in Equations (12b) and (12c).

C.2. Proof of Lemma 2

Based on the discrete approximation (12) proposed in Appendix C.1, we can have an approximate problem with $O(\frac{\ln(M/m)}{\ln(1+\epsilon)})$ variables and constraints. Below we prove the approximation error of $\texttt{PFA}(\hat{G}^*)$.

First, we show the discrete problem (12) is a relaxation of the original problem (11). The relaxation is by (i) removing all constraints except when V takes the K discrete values $\{V_k\}_{k\in[K]}$; and (ii) further relaxing the remaining K constraints as follows. The K remaining constraints of the original problem can be shown as

$$V_k \leq \hat{\eta}\left[\int_m^{V_k} v\cdot\hat{q}(v)dv + (1-\int_m^{V_k}\hat{q}(v)dv)m\right], k = k_\ell, \ldots, k_u, \tag{13a}$$

$$V_k \leq \hat{\gamma}\left[\int_m^{V_k} v\cdot\hat{q}(v)dv + (1-\int_m^{V_k}\hat{q}(v)dv)m\right], k = 1, \ldots, k_\ell - 1, k_u + 1, \ldots, K. \tag{13b}$$

Constraints (12b) and constraints (12c) are further relaxed constraints of the constraints (13a) and constraints (13b), respectively. In particular, we relax $\int_m^{V_k} v\cdot\hat{q}(v)dv$ to $\sum_{i=1}^{k} V_i\hat{q}_i, \forall k \in [K]$.

Let $\{\hat{\eta}^*, \hat{\gamma}^*, \hat{\boldsymbol{q}}^*\}$ denote the optimal solution of the discrete problem (12). Since the discrete problem is a relaxation of the original problem, we have $(1-\delta)\hat{\eta}^* + \delta\hat{\gamma}^* \leq (1-\delta)\eta^* + \delta\gamma^*$.

Based on $\hat{\boldsymbol{q}}^*$, we can build a piece-wise constant protection function $\hat{G}^*(v) = \sum_{i=1}^{k}\hat{q}_i$ if $v \in [V_k, V_{k+1}], k \in [K-1]$. We then aim to analyze the $\texttt{DRCR}$ of $\texttt{PFA}(\hat{G}^*)$. Following similar approach to the proof of Theorem 3, $\texttt{PFA}(\hat{G}^*)$ can guarantee

$$\texttt{ALG}(\hat{\boldsymbol{q}}^*, I_{v'_{N'}}) \geq \frac{\texttt{OPT}(I_{v'_{N'}})}{\frac{V_k}{V_{k-1}}\hat{\eta}^*} \geq \frac{\texttt{OPT}(I_{v'_{N'}})}{(1+\epsilon)\hat{\eta}^*}, \forall v'_{N'} \in [\ell, u]$$

$$\texttt{ALG}(\hat{\boldsymbol{q}}^*, I_{v'_{N'}}) \geq \frac{\texttt{OPT}(I_{v'_{N'}})}{\frac{V_k}{V_{k-1}}\hat{\gamma}^*} \geq \frac{\texttt{OPT}(I_{v'_{N'}})}{(1+\epsilon)\hat{\gamma}^*}, \forall v'_{N'} \in [m, \ell) \cup (u, M].$$

Thus, the $\texttt{DRCR}$ of $\texttt{PFA}(\hat{G}^*)$ is $(1+\epsilon)[(1-\delta)\hat{\eta}^* + \delta\hat{\gamma}^*] \leq (1+\epsilon)[(1-\delta)\eta^* + \delta\gamma^*] \leq (1-\delta)\eta^* + \delta\gamma^* + \epsilon M/m$. This completes the proof.

Algorithm 5 Online learning algorithm with uncertainty-quantified predictions

input: master algorithm $\mathcal{A}$ (e.g., exponentiated gradients algorithm); UQ space Θ; parameterized algorithms $A(\boldsymbol{w}), \boldsymbol{w} \in \Omega$; parameter ϵ;
initialize ϵ-net $\mathcal{N} = \emptyset$;
for each round $t = 1, ..., T$ **do**
  receive UQ $\theta_t \in \Theta$ and let $\tilde{\theta}_t = \arg\min_{\theta \in \mathcal{N}} \|\theta_t - \theta\|$ be the closest vector in $\mathcal{N}$;
  if $\|\theta_t - \tilde{\theta}_t\| > \epsilon$ **then**
    add θ_t to $\mathcal{N}$;
    start a new instance of algorithm $\mathcal{A}_{\theta_t}$ corresponding to θ_t;
    update $\tilde{\theta}_t \leftarrow \theta_t$;
  end if
  choose $\boldsymbol{w}_t$ as the output of $\mathcal{A}_{\tilde{\theta}_t}$;
  run algorithm $A(\boldsymbol{w}_t)$ to execute the instance I_t, and observe the cost function $f_t(\boldsymbol{w}_t; \theta_t)$;
  update $\mathcal{A}_{\tilde{\theta}_t}$ based on $f_t(\boldsymbol{w}_t; \theta_t)$.
end for

D. Technical Proofs and Supplementary Results for Learning Algorithms with UQ Predictions

D.1. Algorithm and Main Results

The key algorithmic framework we use is based on (Hazan & Megiddo, 2007), as illustrated in Algorithm 5. Given that there exists a master algorithm that can achieve $\tilde{O}(\sqrt{T})$ static regret on the cost sequence $\{f_t\}_{t\in[T]}$ and the cost upper bounds $\{U_t\}_{t\in[T]}$ are Lipschitz in UQ predictions, we show that one can use Algorithm 5 to to achieve a sublinear policy regret $\texttt{PREG}_T$. We prove this fact in the following theorem.

Theorem 5. *For each UQ prediction $\theta_t \in \Theta \subseteq [0,1]^D$, suppose cost function $f_t \sim \xi_{\theta_t}$, and that there exists an algorithm $\mathcal{A}$ that achieves $\tilde{O}(\sqrt{T})$ static regret in expectation for $f_1, ..., f_T$. Moreover, assume that the covering dimension of the $\{\theta_t\}_{t\in T}$ is $d \leq D$, i.e., the UQ vectors originate from a d-dimensional subspace of $[0,1]^D$. For any cost upper bound U_t of $\mathbb{E}_{\xi_{\theta_t}} f_t$, i.e. $U_t(\boldsymbol{w}_t; \theta_t) \geq \mathbb{E}_{\xi_{\theta_t}} f_t(\boldsymbol{w}_t; \theta_t)$ for all $\boldsymbol{w}_t \in \Omega$, such that U_t is L-Lipschitz in $\theta_t \in \Theta$, Algorithm 5 with $\mathcal{A}$ as its master algorithm achieves the expected policy regret with respect to the optimal policy π^* for $U_1, ..., U_T$*

$$\sum\nolimits_{t\in[T]} [\mathbb{E} f_t(\pi_t(\theta_t); \theta_t) - U_t(\pi^*(\theta_t); \theta_t)] = \tilde{O}(L^{1-\frac{2}{d+2}} T^{1-\frac{1}{d+2}}),$$

where $\pi^* = \arg\min_\pi \sum_{t\in[T]} U_t(\pi(\theta_t); \theta_t)$. Here, the expectation is taken over $f_t \sim \xi_{\theta_t}$ and the randomness of the master algorithm.

By definition, $\texttt{DRCR}$ is a natural cost upper bound for cost functions. However, we remark that in many instances there will likely be a tighter upper bound than the $\texttt{DRCR}$ that is also Lipschitz in θ, and we showed that Algorithm 5 can automatically compete against any such upper bound. Thus, we expect that in practice, Algorithm 5 can potentially do better than the bounds stated in this section, and in particular it likely can exploit $\texttt{PIP}$s *more optimally* than optimization-based algorithms. We confirm this in our numerical experiments.

D.2. Applications to Ski Rental and Online Search Problems

We consider how one can use a variant of Algorithm 5 with master algorithm being the randomized exponentiated (sub)gradient (EG) algorithm (Shalev-Shwartz, 2012; McMahan, 2015) to derive policy regret guarantees on the $\texttt{DRCR}$ for the ski rental and online search problems when UQs are probabilistic interval predictions $\texttt{PIP}(\theta) = \texttt{PIP}(\ell, u; \delta)$. This is done by observing that the $\texttt{DRCR}$ upper bounds the expected competitive ratio, and, under mild conditions, $\texttt{DRCR}$ exhibits Lipschitzness with respect to θ. Then, using the ideas in Theorem 5, we show that it is sufficient to learn the low-regret policy with respect to $\texttt{DRCR}$ by just using the realized cost ratio function f_t for each instance t. Also note that the learning algorithm is unaware of θ's status as UQ, instead treating the vectors θ_t as generic side information. However, by leveraging the structure of the information space for $\texttt{PIP}$, we can improve the regret guarantees for ski-rental (see Corollary 1) compared to the general guarantees in Theorem 5, and obtain provable Lipschitzness guarantees and, hence, regret guarantees for online search (see Corollary 2).

Algorithm 6 Online learning algorithm for multiple-instance ski rental

input: master algorithm $\mathcal{A}$; space of probabilistic interval prediction $\Theta := [\bar{N}] \times [\bar{N}] \times [0,1]$; parameterized algorithms $\texttt{RSR}(\boldsymbol{y}), \boldsymbol{y} \in \mathcal{Y}$; parameter ϵ;
Initialize $\mathcal{N} = \{\mathcal{N}_{(i,j)} = \emptyset\}_{(i,j)\in[\bar{N}]\times[\bar{N}]}$;
for each $t = 1, ..., T$ **do**
  receive $\theta_t = (\ell_t, u_t; \delta_t) \in [\bar{N}] \times [\bar{N}] \times [0,1]$ and let $\tilde{\theta}_t = \arg\min_{\theta \in \mathcal{N}_{(\ell_t,u_t)}} \|\theta_t - \theta\|$ be the closest vector in $\mathcal{N}_{(\ell_t,u_t)}$;
  if $\|\theta_t - \tilde{\theta}_t\| > \epsilon$ **then**
    add θ_t to $\mathcal{N}_{(\ell_t,u_t)}$;
    start a new instance of algorithm $\mathcal{A}_{\theta_t}$ corresponding to θ_t;
    update $\tilde{\theta}_t \leftarrow \theta_t$;
  end if
  choose $\boldsymbol{y}_t$ as the output of $\mathcal{A}_{\tilde{\theta}_t}$;
  run algorithm $\texttt{RSR}(\boldsymbol{y}_t)$ to execute the instance I_t;
  update $\mathcal{A}_{\tilde{\theta}_t}$.
end for

D.2.1. SKI RENTAL PROBLEM

Consider an online sequence of T instances of the ski rental problem, where we assume that $\bar{N} \geq 2$ bounds the number of skiing days and $B > 0$ is the buying cost. We will also assume that at the start of instance t, we receive a $\texttt{PIP}(\ell_t, u_t; \delta_t) \in [\bar{N}] \times [\bar{N}] \times [0,1]$. The Algorithm 6 we use is a slight modification of Algorithm 5, where we consider the fact that ℓ_t, u_t are discrete. Specifically, we separately consider the spaces $(\ell, u, \cdot) = [0,1]$ indexed by all $\bar{N}^2$ combinations of (ℓ, u) (the number of combinations can be further reduced in practice by noting $u \geq \ell$), and we separately cover each space with an ϵ-net $\mathcal{N}_{(\ell,u)}$. Whenever we receive a UQ vector $\theta_t = (\ell_t, u_t; \delta_t)$, we will run the online learning algorithm corresponding to the closest point in $\mathcal{N}_{(\ell_t,u_t)}$.

Let $F_t := F_t(Y_t; \theta_t) = \frac{\texttt{ALG}(Y_t, I_t)}{\texttt{OPT}(I_t)} : [\bar{N}] \to \mathbb{R}^+$ denote the cost ratio function for ski rental that can be constructed after observing the instance I_t, where $F_t(Y_t; \theta_t)$ is the cost ratio of buying on day $Y_t \in [\bar{N}]$. Let $f_t := f_t(\boldsymbol{y}_t; \theta_t)$ denote the expected cost ratio function over the randomized decision Y_t, i.e., $f_t(\boldsymbol{y}_t; \theta_t) = \mathbb{E}_{Y_t \sim \boldsymbol{y}_t} F_t(Y_t; \theta_t)$. We will also denote the DRCR function we derived in Section 3 as $U_t := U_t(\boldsymbol{y}_t; \theta_t)$, which upper bounds $\mathbb{E}_{\xi_\theta} f_t$, i.e., $U_t(\boldsymbol{y}_t; \theta_t) \geq \mathbb{E}_{\xi_\theta} f_t(\boldsymbol{y}_t; \theta_t), \forall \boldsymbol{y}_t \in \mathcal{Y}$, where $\mathcal{Y}$ is the simplex over support $[\bar{N}]$. By using similar proof ideas to Theorem 5, we can show that learning using the cost ratio f_t, which we can construct in hindsight after each instance I_t, allows us to compete against the DRCR U_t.

Corollary 1. *For the multi-instance ski rental problem, there is a policy $\{\pi_t\}_{t\in[T]}$ that can compete against the optimal policy π^* that maps Θ to the simplex $\mathcal{Y}$ over $[\bar{N}]$ with respect to DRCR $\{U_t\}_{t\in[T]}$, obtaining expected policy regret*

$$\sum\nolimits_{t\in[T]} [\mathbb{E} f_t(\pi_t(\theta_t); \theta_t) - U_t(\pi^*(\theta_t); \theta_t)] = \tilde{O}(\bar{N}(\max\{(\bar{N} + B)/B, B\})T^{2/3}),$$

where $\pi^* = \arg\min_\pi \sum_{t\in[T]} U_t(\pi(\theta_t); \theta_t)$. The expectation is taken over the randomness of the instance distribution and the algorithm. The $\tilde{O}(\cdot)$ hides factors of $\log \bar{N}$.

D.2.2. ONLINE SEARCH PROBLEM

Consider an online sequence of T instances of the online search problem with prices bounded within $[m, M]$. At the start of instance I_t, we receive a $\texttt{PIP}(\ell_t, u_t; \delta_t) \in [m, M] \times [m, M] \times [0,1]$. We use a slight modification of Algorithm 6 to solve this problem, but we further discretize the space of possible $(\ell_t, u_t) \in [m, M]^2$ before applying the algorithm.

Denote $f_t := f_t(\boldsymbol{q}_t; \theta_t) = \frac{\texttt{OPT}(I_t)}{\texttt{ALG}(\boldsymbol{q}_t, I_t)} : \mathcal{Q} \to \mathbb{R}^+$ as the profit ratio function, where $f_t(\boldsymbol{q}; \theta_t)$ is the profit ratio for choosing $\boldsymbol{q} \in \mathcal{Q} = \{\boldsymbol{q} : q(v) \geq 0, v \in [m, M], \int_m^M q(v)dv = 1\}$. The DRCR is again denoted as $U_t := U_t(\boldsymbol{q}_t; \theta_t)$, which upper bounds $\mathbb{E}_{\xi_{\theta_t}} f_t(\boldsymbol{q}; \theta_t)$. We can design an online learning algorithm that can compete against the optimal DRCR.

Corollary 2. *For the multi-instance online search problem, there is a policy $\{\pi_t\}_{t\in[T]}$ that can compete against the optimal*

policy π^* mapping Θ to $\mathcal{Q}$ with respect to DRCR $\{U_t\}_{t\in[T]}$*, obtaining expected policy regret*

$$\sum\nolimits_{t\in[T]}[\mathbb{E}f_t(\pi_t(\theta_t);\theta_t) - U_t(\pi^*(\theta_t);\theta_t)] = \tilde{O}((M-m+1)((M/m)^{8/5}+1)T^{4/5}),$$

where $\pi^* = \arg\min_\pi \sum_{t\in[T]} U_t(\pi(\theta_t);\theta_t)$. The expectation is taken over the instance distribution. The $\tilde{O}(\cdot)$ hides factors of $\log(M/m)$, $\log(M-m)$, and $\log T$.

D.3. Proof of Theorem 5

For each round t, let $\boldsymbol{w}_t = \pi_t(\theta_t)$ be the decision of Algorithm 5 and $\boldsymbol{u}_t = \pi^*(\theta_t)$ be the offline optimal decision for U_t. The policy $\pi_t(\theta_t) = \pi_t(\theta_t|\theta_1, f_1, ..., \theta_{t-1}, f_{t-1})$ determined by Algorithm 5 depends on all past observed cost functions $f_1, \ldots, f_{t-1}$ and UQ predictions $\theta_1, \ldots, \theta_{t-1}$. We can reformulate the policy regret defined in (7) as follows:

$$\sum\nolimits_{t\in[T]}[\mathbb{E}f_t(\pi_t(\theta_t);\theta_t) - U_t(\pi^*(\theta_t);\theta_t)] = \mathbb{E}\sum\nolimits_{\theta\in\mathcal{N}}\sum\nolimits_{t\in\mathcal{T}_\theta}[f_t(\boldsymbol{w}_t;\theta_t) - U_t(\boldsymbol{u}_t;\theta_t)], \tag{14}$$

where $\mathcal{T}_\theta = \{t\in[T] : \tilde{\theta}_t = \theta\}$ denotes set of rounds with UQ corresponding to $\theta\in\mathcal{N}$. Let $T_\theta = |\mathcal{T}_\theta|$ be the number of such rounds. We can further bound this quantity as follows:

$$\begin{aligned}
\mathbb{E}\sum_{t\in\mathcal{T}_\theta}[f_t(\boldsymbol{w}_t;\theta_t) - U_t(\boldsymbol{u}_t;\theta_t)] &= \mathbb{E}\sum_{t\in\mathcal{T}_\theta}[f_t(\boldsymbol{w}_t;\theta_t) - f_t(\boldsymbol{u}_1;\theta_t)] + \sum_{t\in\mathcal{T}_\theta}[\mathbb{E}f_t(\boldsymbol{u}_1;\theta_t) - U_t(\boldsymbol{u}_t;\theta_t)] && \text{(15a)}\\
&\le \tilde{O}(\sqrt{T_\theta}) + \sum_{t\in\mathcal{T}_\theta}[U_t(\boldsymbol{u}_1;\theta_t) - U_t(\boldsymbol{u}_t;\theta_t)] && \text{(15b)}\\
&\le \tilde{O}(\sqrt{T_\theta}) + \sum_{t\in\mathcal{T}_\theta}[U_t(\boldsymbol{u}_1;\theta_t) - U_1(\boldsymbol{u}_1;\theta_1) + U_1(\boldsymbol{u}_t;\theta_1) - U_t(\boldsymbol{u}_t;\theta_t)] && \text{(15c)}\\
&\le \tilde{O}(\sqrt{T_\theta}) + 2T_\theta\cdot\epsilon L, && \text{(15d)}
\end{aligned}$$

where the first inequality follows since the master algorithm $\mathcal{A}$ guarantees the static regret and $U_t(\boldsymbol{u}_1;\theta_t)$ upper bounds the expected cost $\mathbb{E}f_t(\boldsymbol{u}_1;\theta_t)$, the second inequality follows since $\boldsymbol{u}_1$ is the minimizer of $U_1(\boldsymbol{u};\theta_1)$, and the last one follows from our Lipschitz assumption of the cost upper bounds.

Then, using the Cauchy-Schwarz inequality,

$$\sum_{\theta\in\mathcal{N}}\sum_{t\in\mathcal{T}_\theta}[\mathbb{E}f_t(\boldsymbol{w}_t;\theta_t) - U_t(\boldsymbol{u}_t;\theta_t)] \le \sum_{\theta\in\mathcal{N}}\tilde{O}(\sqrt{T_\theta}) + 2T_\theta\epsilon L \le \tilde{O}(\sqrt{|\mathcal{N}|T}) + 2T\epsilon L. \tag{16}$$

Since the covering dimension is d and the distance between any two UQ points in the net $\mathcal{N}$ constructed by Algorithm 5 is ϵ, one can deduce by volume arguments that the ϵ-net has size $|\mathcal{N}| = O(1/\epsilon^d)$. Finally, choosing $\epsilon = (L^2T)^{-1/(d+2)}$ minimizes this expression, giving the final result.

D.4. Proof of Corollary 1

We use Algorithm 6 with the master algorithm being the randomized exponentiated (sub)gradient (EG) algorithm (Shalev-Shwartz, 2012; McMahan, 2015), which is an algorithm for learning from experts. EG runs on the simplex $\mathcal{Y}$ over $[\bar{N}]$ and the function f_t, in order to learn a randomized decision $\boldsymbol{y}\in\mathcal{Y}$, from which the decision Y is sampled from. Note that f_t is bounded above by $\max\{(\bar{N}+B)/B, B\}$. Thus, by setting the step size to be $\frac{\sqrt{\log\bar{N}}}{\max\{(\bar{N}+B)/B,B\}\sqrt{2T}}$ and using Corollary 2.14 in (Shalev-Shwartz, 2012), EG achieves expected static regret guarantee

$$\sum\nolimits_{t\in[T]}\mathbb{E}f_t(\boldsymbol{y}_t;\theta_t) - \mathbb{E}f_t(\boldsymbol{y}^*;\theta_t) = O\left(\max\{(\bar{N}+B)/B, B\}\sqrt{T\log\bar{N}}\right), \tag{17}$$

where $\boldsymbol{y}^* = \arg\min_{\boldsymbol{y}\in\mathcal{Y}}\sum_{t\in[T]}\mathbb{E}f_t(\boldsymbol{y};\theta_t)$ is the optimal decision.

Next, we show that we can compete against the offline optimal sequence of decisions with respect to the DRCR $U_t(\boldsymbol{y}_t;\theta_t)$, which upper bounds $\mathbb{E}f_t(\boldsymbol{y}_t;\theta_t)$. Consider the formulation of the DRCR in the problem (9): $U_t(\boldsymbol{y}_t;\theta_t) = (1-\delta)\eta_t + \delta\gamma_t$, where η_t and γ_t are chosen to be as small as possible while satisfying the constraints given $\boldsymbol{y}_t$. Here, we consider

Lipschitzness of U_t with respect to only δ, which is sufficient due to the specific manner in which the ϵ-net is constructed in Algorithm 6 (i.e., the (ℓ, u) coordinates of the points in the net are selected from a discrete set). Since η_t and γ_t are bounded by $\max\{(\bar{N}+B)/B, B\}$, U_t is $\max\{(\bar{N}+B)/B, B\}$-Lipschitz in δ. For brevity, we let $L := \max\{(\bar{N}+B)/B, B\}$. We also note that in Algorithm 6, any θ_t belonging to the same point of any $\mathcal{N}_{(i,j)}$ is within ϵ distance of each other. Thus, we can use the same steps of the proof of Theorem 5, which gives us

$$\sum\nolimits_{t\in\mathcal{T}_\theta}[\mathbb{E}f_t(\boldsymbol{y}_t;\theta_t) - U_t(\boldsymbol{y}_t^*;\theta_t)] \le \tilde{O}(L\sqrt{T_\theta}) + 2T_\theta\cdot\epsilon L,$$

where $\boldsymbol{y}_t = \pi_t(\theta_t)$ is the online decision by Algorithm 6 and $\boldsymbol{y}_t^* = \pi^*(\theta_t)$ is the optimal decision for U_t. Then, using Cauchy-Schwarz inequality,

$$\begin{aligned}\sum\nolimits_{\theta\in\mathcal{N}}\tilde{O}(L\sqrt{T_\theta}) + 2T_\theta\epsilon L &\le \tilde{O}(L\sqrt{|\mathcal{N}|T}) + 2T\epsilon L\\ &= \tilde{O}(L\sqrt{\textstyle\sum_{(\ell,u)}|\mathcal{N}_{(\ell,u)}|T}) + 2T\epsilon L\\ &\le \tilde{O}(\bar{N}L\sqrt{T/\epsilon}) + 2T\epsilon L,\end{aligned}$$

where the last inequality follows because $|\mathcal{N}_{(\ell,u)}| = O(1/\epsilon)$. Setting $\epsilon = T^{-1/3} < 1$ yields the final result.

D.5. Proof of Corollary 2

Define two sets of discretized points $\Lambda_1 = \{m(1+\lambda_1)^k\}_{k=0,\dots,K}$, where $K = \lfloor\frac{\ln(M/m)}{\ln(1+\lambda_1)}\rfloor$, and $\Lambda_2 = \{m, m+\lambda_2, m+2\lambda_2, \dots, M\}$. Then we consider the discretization $\Lambda = \Lambda_1 \cup \Lambda_2$. Note that Λ takes points from the discrete approximation of problem (11) and the evenly spaced points in $[m, M]$. In the algorithm, for each $t \in [T]$, we round (ℓ_t, u_t) into points $(\tilde{\ell}_t, \tilde{u}_t)$ in $\Lambda_2 \times \Lambda_2$, where ℓ_t is rounded down and u_t is rounded up. Then, Algorithm 6 with EG as the master algorithm is run on Λ and $(\tilde{\ell}_t, \tilde{u}_t, \delta_t)$. Here, EG runs over the simplex $\hat{\mathcal{Q}}$ supported on Λ and optimizes the profit ratio function $f_t(\hat{\boldsymbol{q}};\theta_t)$. Since f_t is bounded above by M/m, EG with step size $\frac{\sqrt{\log(\frac{M-m}{\lambda_2}+K+1)}}{(M/m)\sqrt{2T}}$ yields the following static regret guarantee with respect to any $\hat{\boldsymbol{q}}^* \in \hat{\mathcal{Q}}$:

$$\begin{aligned}\sum\nolimits_{t\in[T]}[f_t(\hat{\boldsymbol{q}}_t;\theta_t) - f_t(\hat{\boldsymbol{q}}^*;\theta_t)] &= O\left((M/m)\sqrt{T\log(\frac{M-m}{\lambda_2}+K+1)}\right)\\ &\le O\left((M/m)\sqrt{T\log(\frac{M-m}{\lambda_2}+\ln(M/m)(1+\frac{1}{\lambda_1})+1)}\right),\end{aligned}$$

where the last inequality follows because $\ln(1+\lambda_1) \ge \lambda_1/(1+\lambda_1)$ for $\lambda_1 > 0$. Ultimately, we will optimize the DRCR over points in $\hat{\mathcal{Q}}$ and a rounded $(\tilde{\ell}, \tilde{u}, \delta)$. We claim that this still allows us to compete against the optimal point in $\mathcal{Q}$. To show this, we will need to (i) verify that the optimal DRCR with respect to $\hat{\mathcal{Q}}$ is not far from the optimal DRCR with respect to original feasible set $\mathcal{Q}$, and (ii) verify that the optimal DRCR with $(\tilde{\ell}, \tilde{u})$ is not far from the DRCR with (ℓ, u), both being with respect to $\mathcal{Q}$.

For the first part, set $U_t(\boldsymbol{q};\theta_t) = (1-\delta_t)\eta_t + \delta_t\gamma_t$ as in the DRCR formulation in problem (11). Since η_t, γ_t are bounded by M/m, U_t is M/m-Lipschitz in δ. We will also consider another DRCR upper bound U_t' optimizing over $\hat{\mathcal{Q}}$ corresponding to problem (12), which is a relaxation of problem (11). For $\boldsymbol{q}^* = \arg\min_{\boldsymbol{q}\in\mathcal{Q}} U_t(\boldsymbol{q};\theta_t)$ and $\hat{\boldsymbol{q}}^* = \arg\min_{\hat{\boldsymbol{q}}\in\hat{\mathcal{Q}}} U_t'(\hat{\boldsymbol{q}};\theta_t)$, Lemma 2 yields $U_t'(\hat{\boldsymbol{q}}^*;\theta_t) \le U_t(\boldsymbol{q}^*;\theta_t) + \frac{M}{m}\lambda_1$. This holds even though we added $\{m+\lambda_2, m+2\lambda_2, ..., M\}$ to the original discretization in problem (12) because adding more points to the discretization will only decrease the approximation error of the discrete DRCR. Thus, competing against U_t' allows us to compete against U_t.

For the second part, note that the error incurred by rounding (ℓ, u) to $(\tilde{\ell}, \tilde{u})$ only affects η: in problem (11), constraints corresponding to $V \in [\tilde{\ell}, \ell) \cup (u, \tilde{u}]$ will be added to constraint set (11b) and removed from constraint set (11c). Thus, η will only increase, while γ is bounded below by η and can only increase by the amount that η increases since (11c) has no added constraints. Focusing on constraints (11b), we consider how much η can increase by adding constraints corresponding to $V = \tilde{\ell}$ and $V = \tilde{u}$. For any fixed $\boldsymbol{q}$, denote

$$G(c) = \int_m^c v\cdot q(v)dv + (1 - \int_m^c q(v)dv)m.$$

For the former, by adding the constraint

$$\tilde{\ell} \le \eta G(\tilde{\ell}),$$

η will change by at most

$$\begin{aligned}
|\ell/G(\ell) - \tilde{\ell}/G(\tilde{\ell})| &\le |\ell/G(\ell) - \ell/G(\tilde{\ell})| + \lambda_2/G(\tilde{\ell}) \\
&\le \ell \frac{G(\ell) - G(\tilde{\ell})}{G(\ell)G(\tilde{\ell})} + \lambda_2/G(\tilde{\ell}) \\
&\le \ell \frac{G(\ell) - G(\ell) + \int_{\tilde{\ell}}^{\ell} v \cdot q(v)dv - m\int_{\tilde{\ell}}^{\ell} q(v)dv}{m^2} + \lambda_2/m \\
&\le \ell(\ell^2 - \tilde{\ell}^2)/2m^2 + \lambda_2/m \le \lambda_2(M^2/m^2 + 1/m),
\end{aligned}$$

where the first inequality follows by triangle inequality and the third inequality follows by taking $G(c) \ge m$ and $x(v) \le 1$. Similar steps for the latter yields the same bound:

$$\begin{aligned}
|u/G(u) - \tilde{u}/G(\tilde{u})| &\le |u/G(u) - u/G(\tilde{u})| + \lambda_2/G(\tilde{u}) \\
&\le u \frac{G(\tilde{u}) - G(u)}{G(u)G(\tilde{u})} + \lambda_2/G(\tilde{u}) \\
&\le u \frac{G(\tilde{u}) - G(\tilde{u}) + \int_{u}^{\tilde{u}} v \cdot q(v)dv - m\int_{u}^{\tilde{u}} q(v)dv}{m^2} + \lambda_2/m \\
&\le u(\tilde{u}^2 - u^2)/2m^2 + \lambda_2/m \le \lambda_2(M^2/m^2 + 1/m),
\end{aligned}$$

Denote $\tilde{\theta}_t = (\tilde{\ell}_t, \tilde{u}_t, \delta_t)$, and let $\tilde{\boldsymbol{q}}^*, \boldsymbol{q}^*$ be the optimal decisions for $U_t(\cdot; \tilde{\theta}_t), U_t(\cdot; \theta_t)$, respectively. Thus, $U_t(\tilde{\boldsymbol{q}}^*; \tilde{\theta}_t) \le U_t(\boldsymbol{q}^*; \theta_t) + (\frac{M^2}{m^2} + \frac{1}{m})\lambda_2$. Finally, note that the number of points in the ϵ-net constructed by Algorithm 6 is $|\mathcal{N}| = \sum_{(\ell,u)} |\mathcal{N}_{(\ell,u)}| = \mathcal{O}((\frac{M-m}{\lambda_2} + 1)^2/\epsilon)$, and that U'_t is M/m-Lipschitz in δ. Putting everything together and again following the steps of the proof for Theorem 5,

$$\begin{aligned}
\mathbb{E}\sum_{t\in[T]} f_t(\hat{\boldsymbol{q}}_t; \theta_t) &\le \tilde{O}((\frac{M-m}{\lambda_2} + 1)(M/m)\sqrt{T/\epsilon}) + 2T\epsilon(M/m) + \sum_{t\in[T]} U'_t(\hat{\boldsymbol{q}}^*_t; \tilde{\theta}_t) \\
&\le \tilde{O}((\frac{M-m}{\lambda_2} + 1)(M/m)\sqrt{T/\epsilon}) + 2T\epsilon(M/m) + T\lambda_1(M/m) + \sum_{t\in[T]} U_t(\tilde{\boldsymbol{q}}^*_t; \tilde{\theta}_t) \\
&\le \tilde{O}((\frac{M-m}{\lambda_2} + 1)(M/m)\sqrt{T/\epsilon} + (M^2/m^2 + M/m)(\epsilon + \lambda_1 + \lambda_2)T) + \sum_{t\in[T]} U_t(\boldsymbol{q}^*_t; \theta_t).
\end{aligned}$$

Here $\tilde{O}$ hides factors of $\ln(M/m)$. Setting $\epsilon = \min\{(M/m)^{-2/5}, 1\}T^{-1/5} < 1$, $\lambda_1 = \min\{(M/m)^{-2/5}, 1\} \cdot \min\{M/m - 1, 1\}T^{-1/5} < M/m - 1$, and $\lambda_2 = \min\{(M/m)^{-2/5}, 1\}(M-m)T^{-1/5} < M - m$ yields the result.

D.6. Experiments

Setup. We set buying cost to $B = 2$. We generate $T = 3000$ instances, each with true skiing days n_t sampled uniformly at random from $\{1, \ldots, 8\}$. Synthetic `PIP` predictions are generated by sampling a point p_t from a normal distribution $\mathcal{N}(n_t, \sigma_t^2)$ and then taking the 90% confidence interval $(\ell_t, u_t) = (p_t - z_{0.95}\sigma_t, p_t + z_{0.95}\sigma_t)$. Here, σ_t is set to simulate oscillating good and bad predictors: the first 10 instances have $\sigma_t = 0$, followed by the next 10 with $\sigma_t = 6$, and repeating.

Comparison algorithms. We run the following online algorithms over the same sequence of UQs and instances 10 times and evaluate the average excess CR (i.e., the empirical ratio minus 1) of these algorithms over these runs. **`WOA`**: worst-case optimal randomized algorithm (Karlin et al., 1990) that is $e/e-1$-competitive. **`FTP`**: follow-the-prediction algorithm that fully trusts the prediction; it buys immediately on day 1 if the prediction is less than B, and otherwise rents forever. **`OL-Dynamic`**: online learning with respect to policy regret (Algorithm 5 adapted to ski rental, i.e., Algorithm 6), i.e., competing against $\pi^* = \arg\min_\pi \sum_{t\in[T]} U_t(\pi(\theta_t); \theta_t)$. **`OL-Static`**: online learning with respect to static regret, i.e., competing against the optimal fixed decision without considering UQ predictions. **`RSR-PIP`**: randomized algorithm with `PIP` (Algorithm 3) that achieves the optimal `DRCR`.