Bandit Algorithms for Prophet Inequality and Pandora's Box

Khashayar Gatmiry* Thomas Kesselheim† Sahil Singla‡ Yifan Wang§

Abstract

The Prophet Inequality and Pandora's Box problems are fundamental stochastic problem with applications in Mechanism Design, Online Algorithms, Stochastic Optimization, Optimal Stopping, and Operations Research. A usual assumption in these works is that the probability distributions of the n underlying random variables are given as input to the algorithm. Since in practice these distributions need to be learned under limited feedback, we initiate the study of such stochastic problems in the Multi-Armed Bandits model.

In the Multi-Armed Bandits model we interact with n unknown distributions over T rounds: in round t we play a policy $x^{(t)}$ and only receive the value of $x^{(t)}$ as feedback. The goal is to minimize the regret, which is the difference over T rounds in the total value of the optimal algorithm that knows the distributions vs. the total value of our algorithm that learns the distributions from the limited feedback. Our main results give near-optimal $\widetilde{O}\big(\mathsf{poly}(n)\sqrt{T}\big)$ total regret algorithms for both Prophet Inequality and Pandora's Box.

Our proofs proceed by maintaining confidence intervals on the unknown indices of the optimal policy. The exploration-exploitation tradeoff prevents us from directly refining these confidence intervals, so the main technique is to design a regret upper bound function that is learnable while playing low-regret Bandit policies.

1 Introduction

The field of Stochastic Optimization deals with optimization problems under uncertain inputs, and has had tremendous success since [Bel57]. A standard model is that the inputs are random variables that are drawn from known probability distributions. The goal is to design a policy (an adaptive algorithm) to optimize the expected objective function. Examples of such problems include Prophet Inequality [HKS07, CHMS10, KW12, Rub16], Pandora's Box [KWW16, Sin18b, GKS19], and Auction Design [Har22, Rou16]. Most prior works assume that the underlying distributions are known to the algorithm and the challenge is in computing an (approximately) optimal policy. However, in practical applications, the distributions are typically *unknown* and must be learned concurrently with decision-making.

A foundational framework that examines stochastic problems with unknown distributions is the stochastic online learning model; see books [CBL06, BC12, Haz16]. Here, the learner interacts with the environment for T days. On each day $t \in [T]$, the learner plays a certain policy $a^{(t)} \in A$, where A represents the set of all policies (actions/algorithms). The environment draws a sample $X^{(t)} \sim \mathcal{D}$, where $\mathcal{D}$ indicates the environment's *unknown* underlying distribution, and then the learner receives a reward $a^{(t)}(X^{(t)})$ along with some "feedback". For a maximization problem, the goal of the online learning model is to approach the optimal policy with reward $\mathsf{Opt} := \max_{a \in A} \mathbb{E}_{X \sim \mathcal{D}}[a(X)]$ while minimizing in expectation the total regret:

$$T \cdot \mathsf{Opt} - \textstyle\sum_{t \in [T]} a^{(t)}(X^{(t)}).$$

The best regret bound that can be achieved for an online learning problem highly depends on the feedback given to the algorithm. In the full-feedback model, the learner observes the complete sample $X^{(t)}$ as daily feedback. Since accessing the entire sample $X^{(t)}$ is often not feasible in many real-world applications, several partial feedback models have been considered. The most limiting of them is the bandit feedback model where the only feedback available is the reward $a^{(t)}(X^{(t)})$; see books [Sli19, LS20].

*(gatmiry@mit.edu) Electrical Engineering and Computer Science, Massachusetts Institute of Technology

†(thomas.kesselheim@uni-bonn.de) Institute of Computer Science and Lamarr Institute for Machine Learning and Artificial Intelligence, University of Bonn

‡(ssingla@gatech.edu) School of Computer Science, Georgia Tech. Supported in part by NSF award CCF-2327010.

§(ywang3782@gatech.edu) School of Computer Science, Georgia Tech. Supported in part by NSF award CCF-2327010.

Interestingly, in many online learning scenarios, limiting feedback does not excessively impair the regret bound. For instance, consider the classic Learning from Experts problem where the goal is to identify the optimal action. In this case, for a small action set, both full feedback and bandit feedback result in an optimal regret bound of $\Theta(\sqrt{T})$. This motivates us to address the following question for general online stochastic optimization problems:

> *What is the minimum amount of feedback necessary to learn a stochastic optimization problem while maintaining a near-optimal regret bound in T as the full feedback model?*

In addition to being an intellectually intriguing question, there are several other motivations for designing low regret algorithms that operate with limited feedback.

- In numerous real-world scenarios, accessing the complete sample $X^{(t)}$ as feedback is infeasible. Furthermore, in order to safeguard data privacy to the greatest extent possible, it is advantageous to utilize minimal information in real-world online learning tasks.
- An online learning algorithm that operates with less feedback is concurrently applicable to all partial feedback models that incorporate the required feedback. We can therefore obtain near-optimal online learning algorithms that function uniformly across different feedback models.

Specifically, in this paper, we address the above question in the context of the fundamental Prophet Inequality and Pandora's Box problems, which have wide-ranging applications in areas such as Mechanism Design, Online Algorithms, Microeconomics, Operations Research, and Optimal Stopping. Our main results imply near-optimal $\widetilde{O}\big(\mathsf{poly}(n)\sqrt{T}\big)$ regret algorithms for both these problems under most limited bandit feedback, where $\widetilde{O}(\cdot)$ hides logarithmic factors.

1.1 Prophet Inequality under Bandit Feedback

In the classical Optimal Stopping problem of Prophet Inequality [KS77, KS78, SC84], we are given distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$ of n independent random variables. The outcomes $X_i \sim \mathcal{D}_i$ for $i \in [n]$ are revealed one-by-one and we have to immediately select/discard X_i with the goal of maximizing the selected random variable in expectation. They have become popular in Algorithmic Game Theory in the last 15 years since they imply posted pricing mechanisms that are "simple" (and hence more practical) and approximately optimal; see related work in Section 1.4.

The optimal policy for Prophet Inequality is given by a simple (reverse) dynamic program: always select X_n on reaching it and select X_i for $i < n$ if its value is more than the expected value of this optimal policy on $X_{i+1}, \ldots, X_n$. Thus, the optimal policy with expected value Opt can be thought of as a *fixed-threshold* policy where we select X_i iff $X_i > \tau_i$ for τ_i being the expected value of this policy after i. How to design this optimal policy for unknown distributions? (See Remark 1.2 on the "hindsight optimum" benchmark.)

As a motivating example, consider a scenario where you want to sell a perishable item (e.g., cheese) in the market each day for the entire year. For simplicity, assume that there are 8 buyers, one arriving in each hour between 9 am to 5 pm. Your goal is to set price thresholds for each hour to maximize the total value. If the buyer value distributions are known, this can be modeled as a Prophet Inequality problem with $n = 8$ distributions. However, for unknown value distributions this becomes a repeated game with a fixed arrival order where on each day you play some price thresholds and obtain a value along with feedback. Next, we formally describe this repeated game.

Online Learning Prophet Inequality. In this problem the distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$ of Prophet Inequality are unknown to the algorithm in the beginning. We make the standard normalization assumption that each $\mathcal{D}_i$ is supported on $[0, 1]$. Without this normalization, a non-trivial additive regret is not achievable. Now we play a T rounds repeated game[1]: in round $t \in [T]$ we play a policy, which is a set of n thresholds $(\tau_1^{(t)}, \ldots, \tau_n^{(t)})$, and receives as reward its value on freshly drawn independent random variables $X_1^{(t)} \sim \mathcal{D}_1, \ldots, X_n^{(t)} \sim \mathcal{D}_n$, i.e., the reward is $X^{(t)}_{\mathsf{Alg}(t)}$ where $\mathsf{Alg}(t) \in [n]$ is the smallest index i with $X_i^{(t)} > \tau_i^{(t)}$. The goal is to minimize the *total regret*:

$$T \cdot \mathsf{Opt} - \mathbf{E}\left[\sum_{t=1}^{T} X^{(t)}_{\mathsf{Alg}(t)}\right].$$

[1] We will always assume $T \geq n$ since otherwise getting an $O(\mathsf{poly}(n))$ regret algorithm is trivial.

Since per-round reward is bounded by 1, the goal is to get $o(T)$ regret. Moreover, standard examples show that every algorithm incurs $\Omega(\sqrt{T})$ regret; see Section 5.

An important question is what amount of feedback the algorithm receives after a round. One might consider a *full-feedback* setting, where after each round t the algorithm gets to know the entire sample $X_1^{(t)}, \dots, X_n^{(t)}$ as feedback, which could be used to update beliefs regarding the distributions $\mathcal{D}_1, \dots, \mathcal{D}_n$. Here it is easy to design an $\widetilde{O}\big(\mathsf{poly}(n)\sqrt{T}\big)$ regret algorithm. This is because after discretization, we may assume that the there are only T candidate thresholds for each X_i, so there are only T^n candidate policies. Now the classical multiplicative weights algorithm [AHK12] implies that the regret is $O\big(\sqrt{T \log(\#\text{policies})}\big) = \widetilde{O}\big(\mathsf{poly}(n)\sqrt{T}\big)$. Although this naïve algorithm is not polytime, a recent work of [GHTZ21] on $O(n/\epsilon^2)$ sample complexity for prophet inequality can be interpreted as giving a polytime $\widetilde{O}(\mathsf{poly}(n)\sqrt{T})$ regret algorithm under full-feedback[2]. These results, however, do not extend to *bandit feedback*, where the algorithm does not see the entire sample.

Bandit Feedback. In many applications, it is unreasonable to assume that the algorithm gets the entire sample $X_1^{(t)}, \dots, X_n^{(t)}$. For instance, in the above scenario of selling a perishable item, we may only see the winning bid (e.g., if you don't run the shop and delegate someone else to sell the item at the given price thresholds). There are several reasonable partial feedback models, namely:

(a) We see $X_1^{(t)}, \dots, X_{\mathsf{Alg}(t)}^{(t)}$ but not $X_{\mathsf{Alg}(t)+1}^{(t)}, \dots, X_n^{(t)}$, meaning that we do not observe the sequence after it has been stopped.
(b) We see the index $\mathsf{Alg}(t)$ and the value $X_{\mathsf{Alg}(t)}$ that we select but no other X_i.
(c) We only see the value of $X_{\mathsf{Alg}(t)}$ that we select and not even the index $\mathsf{Alg}(t)$.

What is the least amount of feedback needed to obtain $\widetilde{O}(\mathsf{poly}(n)\sqrt{T})$ regret?

Our first main result is that even with the most restrictive feedback (c), it is possible to obtain $\widetilde{O}(\mathsf{poly}(n)\sqrt{T})$ regret. Thus, the same bounds also hold under (a) and (b). Note that these bounds are almost optimal because standard examples show that even with full feedback every algorithm incurs $\Omega(\sqrt{T})$ regret (see Section 5).

Theorem 1.1. *There is a polytime algorithm with $O(n^3\sqrt{T}\log T)$ regret for the Bandit Prophet Inequality problem where we only receive the selected value as the feedback.*

(We remark that it is possible to improve the n^3 factor in this result but we do not optimize it to keep the presentation cleaner.)

Theorem 1.1 may come as a surprise since there are several stochastic problems that admit $O(\mathsf{poly}(n)/\epsilon^2)$ sample complexity but do not admit $\widetilde{O}\big(\mathsf{poly}(n)\sqrt{T}\big)$ regret bandit algorithms. Indeed, a close variant of prophet inequality is sequential posted pricing. Here, the reward is defined as the revenue, i.e., it is the threshold itself if a random variable crosses it rather than the value of the random variable (welfare) as in prophet inequality. It is easy to show that sequential posted pricing has $O(1/\epsilon^2)$ sample complexity [GHTZ21], but even for $n = 1$ every bandit algorithm incurs $\Omega(T^{2/3})$ regret [LSTW23].

One might wonder whether $\widetilde{O}\big(\mathsf{poly}(n)\sqrt{T}\big)$ regret in Theorem 1.1 holds even for adversarial online learning, i.e., where $X_1^{(t)}, \dots, X_n^{(t)}$ are chosen by an adversary in each round t and we compete against the optimal fixed-threshold policy in hindsight. In Section 5 we prove that this is impossible since every online learning algorithm incurs $\Omega(T)$ regret for adversarial inputs, even under full-feedback.

Remark 1.2 (Hindsight Optimum)**.** *There is a lot of work on Prophet Inequality (with Samples) where the benchmark is the expected hindsight optimum $\mathbf{E}[\max X_i]$; see Section 1.4. However, we will be interested in the more realistic benchmark of the optimal policy, or in other words the optimal solution to the underlying MDP, which is standard in stochastic optimization. Firstly, comparing to the hindsight optimum does not make sense for most stochastic problems, including Pandora's Box, since it cannot be achieved even approximately. Secondly, optimal policy gives us a much more fine-grained picture than comparing to the offline optimum. For instance, it is known that a single sample suffices to get the optimal 2-competitive guarantee compared to the offline optimum [RWW20]. This might give the impression that there is nothing to be learned about the distributions for Prophet Inequality*

[2]Their results are in the PAC model for "strongly monotone" stochastic problems. They immediately imply $\widetilde{O}(\sqrt{nT})$ regret under full-feedback using the standard doubling-trick.

and sublinear regrets are impossible. However, this is incorrect as Theorem 1.1 obtains sublinear regret bounds w.r.t. the optimal policy.

1.2 Pandora's Box under Bandit Feedback The Pandora's Box problem was introduced by Weitzman, motivated by Economic search applications [Wei79]. For example, how should a large organization decide between competing research technologies to produce some commodity. In the classical setting, we are given distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$ of n independent random variables. The outcome $X_i \sim \mathcal{D}_i$ for $i \in [n]$ can be obtained by the algorithm by paying a known *inspection cost* c_i. The goal is to find a policy to adaptively inspect a subset $S \subseteq [n]$ of the random variables to maximize *utility*: $\mathbf{E}\left[\max_{i \in S} X_i - \sum_{i \in S} c_i\right]$. Note that unlike the Prophet Inequality, we may now inspect the random variables in any order by paying a cost and we don't have to immediately accept/reject X_i.

Even though Pandora's Box has an exponential state space, [Wei79] showed a simple optimal policy where we inspect in a fixed order (using "indices") along with a stopping rule. We study this problem in the Online Learning model where the distributions $\mathcal{D}_i$ supported on $[0, 1]$ are unknown-but-fixed. Without loss of generality, we will assume that the deterministic costs $c_i \in [0, 1]$ are known to the algorithm[3].

Formally, in Online Learning for Pandora's Box we play a T rounds repeated game where in round $t \in [T]$ we play a policy $a^{(t)}$, which is an order of inspection along with a stopping rule. As reward, we receive our utility (value minus total inspection cost) on freshly drawn independent random variables $X_1^{(t)} \sim \mathcal{D}_1, \ldots, X_n^{(t)} \sim \mathcal{D}_n$. The goal is to minimize the total *regret*, which is the difference over T rounds in the expected utility of the optimal algorithm that knows the underlying distributions and the total utility of our algorithm.

In the full-feedback setting the algorithm receives the entire sample $X_1^{(t)}, \ldots, X_n^{(t)}$ as feedback in each round. Here, it is again easy to design an $\widetilde{O}\big(\mathsf{poly}(n)\sqrt{T}\big)$ regret polytime algorithm relying on the results in [GHTZ21, FL20]. But these results do not extend to partial feedback.

There are again multiple ways of defining partial feedback. E.g., we could see the values of all X_i for $i \in S$, meaning that we get to see the values of the inspected random variables. Indeed, our results again apply to the most restrictive form of partial feedback: We only see the total utility of a policy and not even the indices of inspected random variables or any of their values.

Theorem 1.3. *There is a polytime algorithm with $O(n^{5.5}\sqrt{T}\log T)$ regret for the Bandit Pandora's Box problem where we only receive utility (selected value minus total cost) as feedback.*

Again, standard examples show that every algorithm incurs $\Omega(\sqrt{nT})$ regret even will full feedback; see Section 5. Furthermore, we will prove in Section 5 that Theorem 1.3 cannot hold for adversarial online learning where $X_1^{(t)}, \ldots, X_n^{(t)}$ are chosen by an adversary: every online learning algorithm incurs $\Omega(T)$ regret for adversarial inputs, even under full-feedback.

1.3 High-Level Techniques Let's consider the general Prophet Inequality problem or the subproblem of Pandora's Box where the optimal order is given. In both cases, a policy is described by n thresholds $\tau_1, \ldots, \tau_n \in [0, 1]$, defining when to stop inspecting. It would be tempting to apply standard multi-armed bandit algorithms to maximize the expected reward over $[0, 1]^n$. However, such approaches are bound to fail because the expected reward is not even continuous[4], let alone convex or Lipschitz. Discretizing the action space and applying a bandit algorithm only leads to $\Omega(T^{2/3})$ regret. Another reasonable approach is to try to learn the distributions $\mathcal{D}_i$. However, recall that we only get feedback regarding the overall reward of a policy and do not see which X_i is selected. It is possible to obtain samples from each X_i by considering policies that ignore all other boxes; however, such algorithms that use *separate* exploration and exploitation also have $\Omega(T^{2/3})$ regret.

[3]If the costs c_i are unknown but fixed then the problem trivially reduces to the case of known costs. This is because we could simply open each box once without keeping the prize inside and receive as feedback the cost c_i.

[4]For example, consider the Prophet Inequality instance in which X_1 is a distribution that returns $\frac{1}{4}$ w.p. $\frac{1}{2}$ and $\frac{3}{4}$ otherwise, while X_2 is a distribution that always returns $\frac{1}{2}$. The reward of this example is a piece-wise constant function: When $\tau < \frac{1}{4}$ or $\tau \geq \frac{3}{4}$, the expected reward is $\frac{1}{2}$. When $\frac{1}{4} \leq \tau < \frac{3}{4}$, the expected reward is $\frac{5}{8}$.

Our algorithms combine exploration and exploitation. We maintain confidence intervals $[\ell_i, u_i]$ for $i \in [n]$ satisfying w.h.p. that the optimal thresholds $\tau_i^* \in [\ell_i, u_i]$. The crucial difference from UCB-style algorithms [ACBF02] is that we don't get unbiased samples with low regret, so we cannot maintain or play upper confidences. Instead, we need a "refinement" procedure to shrink the intervals while ensuring that the regret during the refinement is bounded.

More precisely, our algorithm works in $O(\log T)$ phases. In each phase, we start with confidence intervals $[\ell_i, u_i]$ that satisfy: (i) $\tau_i^* \in [\ell_i, u_i]$ and (ii) playing any thresholds within the confidence intervals incur at most some ϵ regret. During the phase, we refine the confidence interval to $[\ell'_i, u'_i]$ while only playing thresholds within our original confidence intervals, so that we don't incur much regret. We will show that the new confidence intervals satisfy that $\tau_i^* \in [\ell'_i, u'_i]$ and that playing any thresholds within $[\ell'_1, u'_1], \ldots, [\ell'_n, u'_n]$ incur at most $\frac{\epsilon}{2}$ regret. Thus, the regret bound goes down by a constant factor in each phase.

Bounding Function to Refine for $n = 2$. To illustrate the idea behind a refinement phase, let's discuss the case of $n = 2$; see Section 2 for more technical details. In this case, there is only one confidence interval $[\ell, u]$ that we have to refine. Our idea is to define a "bounding function" $\delta(\cdot)$ such that the expected regret in a single round when using threshold $\tau \in [\ell, u]$ is bounded by $|\delta(\tau)|$. Ideally, we would like to choose the optimal threshold τ^* for which $\delta(\tau^*) = 0$. However, this requires the knowledge of δ, which we don't have since the distributions are unknown. Instead, we compute an estimate $\hat{\delta}$ of δ and construct the new confidence interval $[\ell', u']$ to include all τ for which $|\hat{\delta}(\tau)|$ is small. The main technical difficulty is to obtain $\hat{\delta}$ while only playing low-regret policies. We achieve this by choosing δ such that $\hat{\delta}$ can be obtained by using only the estimates $\hat{F}_i$ of the CDF and the empirical average rewards when choosing the boundaries of the confidence interval as thresholds. Note that we do not make any statements about the width of the confidence interval; we only ensure that the regret is bounded when choosing any threshold inside the confidence interval.

Prophet Inequality for General n. In the case of general n, each refinement phase updates the confidence intervals from the last random variable X_n to the first one X_1. To refine confidence interval $[\ell_i, u_i]$, we use our algorithm for the $n = 2$ case as a subroutine, i.e., we play ℓ_i and u_i sufficiently many times keeping the other thresholds fixed. However, there are several challenges in this approach. The first important one is that the probability of reaching X_i will change depending on which thresholds are applied before it. We deal with this issue by always using thresholds from our confidence intervals that maximize the probability of reaching X_i. Another important challenge while refining $[\ell_i, u_i]$ is that the current choice of thresholds for $X_{i+1}, \ldots, X_n$ is not optimal, so we maybe learning a threshold different from τ_i^*. We handle this issue by choosing the other thresholds in a way that they only improve from phase to phase. We then leave some space in the confidence intervals to accommodate for the improvements in later phases.

Pandora's Box for General n. We still maintain confidence intervals and refine them using ideas similar to Prophet Inequality for general n. The main additional challenge arising in Pandora's box is that the inspection ordering is not fixed. The optimal order is given by ordering the random variables by decreasing thresholds. However, there might be multiple orders consistent with our confidence intervals. Therefore, we keep a set S of constraints corresponding to a directed acyclic graph on the variables, where an edge from X_i to X_j means that X_i comes before X_j in the optimal order. We update this set by consider pairwise swaps. Then, during refinement of confidence interval $[\ell_i, u_i]$, we choose an inspection order satisfying these constraints while (approximately) maximizing a difference of products objective.

1.4 Further Related Work There is a long line of work on both Prophet Inequality (PI) and Pandora's Box (PB), so we only discuss the most relevant papers. For more references, see [Luc17, Sin18a]. Both PI and PB are classical single-item selection problems, but were popularized in TCS in [HKS07] and [KWW16], respectively, due to their applications in mechanism design. Extensions of these problems to combinatorial settings have been studied in [CHMS10, KW12, FGL15, FSZ16, Rub16, RS17, EFGT20] and in [KWW16, Sin18b, GKS19, GJSS19, FTW+21], respectively. Although the optimal policy for PI with known distributions is a simple dynamic program, designing optimal policies for free-order or in combinatorial PI settings is challenging. Some recent works designing approximately-optimal policies are [ANSS19, PPSW21, SS21, LLP+21, BDL22].

Starting with Azar, Kleinberg, and Weinberg [AKW14], there is a lot of work on PI-with-Samples where the distributions are unknown but the algorithm has sample access to it [CDFS19, RWW20, GHTZ21, CDF+22]. These works, however, compete against the benchmark of expected hindsight optimum, so lose at least a multiplicative factor of 1/2 due the classical single-item PI and do not admit sublinear regret algorithms.

The field of Online Learning under both full- and bandit-feedback is well-established; see books [CBL06, BC12, Haz16, Sli19, LS20]. Most of the initial works focused on obtaining sublinear regret for single-stage problems (e.g., choosing the reward maximizing arm). The last decade has seen progress on learning multi-stage policies for tabular MDPs under bandit feedback; see [LS20, Chapter 38]. However, these algorithms have a regret that is polynomial in the state space, so they do not apply to PI and PB that have large MDPs.

Finally, there is some recent work at the intersection of Online Learning and Prophet Inequality/Pandora's Box [EHLM19, ACG+22, GT22]. These models are significantly different from ours, so do not apply to our problems. The closest one is [GT22], where the authors consider Pandora's Box under partial feedback (akin to model (a)), but for *adversarial* inputs (i.e., no underlying distributions). They obtain $O(1)$-competitive algorithms and leave open whether sublinear regrets are possible [Ger22]. Our lower bounds in Section 5.2 resolve this question by showing that sublinear regrets are impossible for adversarial inputs (even under full feedback), and one has to lose a multiplicative factor in the approximation.

2 Prophet Inequality and Pandora's Box for $n = 2$

In this section, we give $O(\sqrt{T}\log T)$ regret algorithms for both Bandit Prophet Inequality and Bandit Pandora's Box problems with $n = 2$ distributions. We discuss this special case of Theorem 1.1 before since it's already non-trivial and showcases one of our main ideas of designing a regret bounding function that is learnable while playing low-regret Bandit policies.

Our algorithms run in $O(\log T)$ phases, where the number of rounds doubles each phase. Starting with an initial confidence interval containing the optimal threshold τ^*, the goal of each phase is to refine this interval such that the one-round regret drops by a constant factor for the next phase. In Section 2.1 we discuss each phase's algorithm for Prophet Inequality with $n = 2$. In Section 2.2 we give a generic doubling framework that combines all phases to prove total regret bounds. Finally, in Section 2.3 we extend these ideas to Pandora's Box with $n = 2$.

2.1 Prophet Inequality via an Interval-Shrinking Algorithm We first introduce the setting of the Bandit Prophet Inequality Problem with two distributions. Let $\mathcal{D}_1, \mathcal{D}_2$ denote the two unknown distributions over $[0, 1]$ with cdfs F_1, F_2 and densities f_1, f_2. Consider a T rounds game where in each round t we play a threshold $\tau^{(t)} \in [0, 1]$ and receive as feedback the following reward:

- Independently draw $X_1^{(t)}$ from $\mathcal{D}_1$. If $X_1^{(t)} \geq \tau^{(t)}$, return $X_1^{(t)}$ as the reward.
- Otherwise, independently draw $X_2^{(t)}$ from $\mathcal{D}_2$ and return it as the reward.

The only feedback we receive is the reward, and not even which random variable gets selected.

If the distributions are known then the optimal policy is to play $\tau^* := \mathbf{E}[X_2]$ in each round. For $\tau \in [0, 1]$, let $R(\tau)$ be the expected reward of playing one round with threshold τ, i.e.,

$$R(\tau) \;:=\; F_1(\tau) \cdot \mathbf{E}[X_2] + \int_\tau^1 x \cdot f_1(x)dx \;=\; 1 + F_1(\tau)(\mathbf{E}[X_2] - \tau) - \int_\tau^1 F_1(x)dx, \tag{2.1}$$

where the second equality uses integration by parts. The *total regret* is $T \cdot R(\tau^*) - \sum_{t=1}^T R(\tau^{(t)})$.

Initialization. For the initialization, we get $\Theta(\sqrt{T}\log T)$ samples from both $\mathcal{D}_1$ and $\mathcal{D}_2$ by playing $\tau = 0$ and $\tau = 1$, respectively. This incurs $\Theta(\sqrt{T}\log T)$ regret since each round incurs at most 1 regret. The following simple lemma uses the samples to obtain initial distribution estimates.

Lemma 2.1. *After getting $C \cdot sqrtT \log T$ samples from $\mathcal{D}_1$ and $\mathcal{D}_2$, with probability $1 - T^{-10}$ we can:*

- *Calculate $\hat{F}_1(x)$ such that $|\hat{F}_1(x) - F_1(x)| \leq T^{-\frac{1}{4}}$ for all $x \in [0, 1]$ simultaneously.*
- *Calculate ℓ and u such that $u - \ell \leq T^{-\frac{1}{4}}$ and $\mathbf{E}[X_2] \in [\ell, u]$.*

Proof. The first statement follows the DKW inequality (Theorem A.3). After taking $N = C \cdot \sqrt{T}\log T$ samples, the probability that $\exists x$ s.t. $|\hat{F}_1(x) - F_1(x)| > \varepsilon = T^{-\frac{1}{4}}$ is at most $2\exp(-2N\varepsilon^2) = 2T^{-2C} < T^{-2C+1}$. So, the first statement holds with probability at least $1 - T^{-11}$ when $C > 10$.

The second statement follows the Hoeffding's Inequality (Theorem A.1). After taking $N = C \cdot \sqrt{T}\log T$ samples, let μ be the average reward. Let $\ell = \mu - \varepsilon$ and $u = \mu + \varepsilon$ for $\varepsilon = \frac{1}{2}T^{-\frac{1}{4}}$. Then, $u - \ell \leq T^{-\frac{1}{4}}$ by definition. Since the reward of each sample in inside $[0,1]$, by Hoeffding's Inequality the probability that $|\mu - \mathbf{E}[X_2]| > \varepsilon$ is bounded by $2\exp(-2N\varepsilon^2) = 2T^{-2C} < T^{-2C+1}$. So, the second statement holds with probability at least $1 - T^{-11}$ when $C > 10$. Taking a union bound for two statements gives the desired lemma. □

Next we discuss our core algorithm.

Interval-Shrinking Algorithm. Starting with an initial confidence interval containing $\tau^* = \mathbf{E}[X_2]$, our Interval-Shrinking algorithm (Algorithm 1) runs for $\Theta(\frac{\log T}{\epsilon^2})$ rounds and outputs a refined confidence interval. In the following lemma, we will show that this refined interval still contains τ^* and that the regret of playing any τ inside this refined interval is bounded by $O(\epsilon)$.

Algorithm 1: Interval-Shrinking Algorithm for Prophet Inequality

Input: Interval $[\ell, u]$, approximate cdf $\hat{F}_1(x)$, and accuracy ϵ.
1. Run $C \cdot \frac{\log T}{\epsilon^2}$ rounds with $\tau = \ell$. Let $\hat{R}_\ell$ be the average reward.
2. Run $C \cdot \frac{\log T}{\epsilon^2}$ rounds with $\tau = u$. Let $\hat{R}_u$ be the average reward.
3. For $\tau \in [\ell, u]$, define $\hat{\Delta}(\tau) := \hat{F}_1(u)(\tau - u) - \hat{F}_1(\ell)(\tau - \ell) + \int_\ell^u \hat{F}_1(x)dx$.
4. For $\tau \in [\ell, u]$, define $\hat{\delta}(\tau) := \hat{\Delta}(\tau) - (\hat{R}_u - \hat{R}_\ell)$.
5. Let $\ell' := \min\{\tau \in [\ell, u] \text{ s.t. } \hat{\delta}(\tau) \geq -5\epsilon\}$ and let $u' := \max\{\tau \in [\ell, u] \text{ s.t. } \hat{\delta}(\tau) \leq 5\epsilon\}$.

Output: $[\ell', u']$

Lemma 2.2. *Suppose we are given:*
- Initial interval $[\ell, u]$ *of length* $u - \ell \leq T^{-\frac{1}{4}}$ *and satisfying* $\tau^* \in [\ell, u]$.
- Distribution estimate $\hat{F}_1(x)$ *satisfying* $|F_1(x) - \hat{F}_1(x)| \leq T^{-\frac{1}{4}}$ *for all* $x \in [0,1]$ *simultaneously.*

Then, for $\epsilon > T^{-\frac{1}{2}}$ *Algorithm 1 runs thresholds inside* $[\ell, u]$ *for at most* $1000 \cdot \frac{\log T}{\epsilon^2}$ *rounds, and outputs a sub-interval* $[\ell', u'] \subseteq [\ell, u]$ *satisfying with probability* $1 - T^{-10}$ *the following statements:*
1. $\tau^* \in [\ell', u']$.
2. *For every* $\tau \in [\ell', u']$ *the expected one-round regret of playing* τ *is at most* 10ϵ.

Proof Overview of Lemma 2.2. The main idea is to define a *bounding function*

$$\delta(\tau) \;:=\; (F_1(u) - F_1(\ell)) \cdot (\tau - \tau^*).$$

As we show in Claim 2.3 below, this function satisfies $R(\tau^*) - R(\tau) \leq |\delta(\tau)|$ for all $\tau \in [\ell, u]$, i.e., $|\delta(\tau)|$ is an upper bound on the one-round regret when choosing τ instead of τ^*. So, ideally, we would like to choose τ that minimizes $|\delta(\tau)|$. However, we do not know $\delta(\tau)$. Therefore, we derive an estimate $\hat{\delta}(\tau)$ for all $\tau \in [\ell, u]$ and discard τ for which $|\hat{\delta}(\tau)|$ is too large because these cannot be the minimizers.

In order to estimate $\delta(\tau)$, we rewrite it in a different way as sum of terms that can be estimated well. First, consider the difference in expected rewards when choosing thresholds u and ℓ, i.e.,

$$R(u) - R(\ell) \;=\; (F_1(u) - F_1(\ell))\tau^* - \int_\ell^u x f_1(x)dx \;=\; F_1(u)\cdot(\tau^* - u) - F_1(\ell)\cdot(\tau^* - \ell) + \int_\ell^u F_1(x)dx,$$

where we used integration by parts. Adding this with $\delta(\tau)$ gives $\delta(\tau) + (R(u) - R(\ell))$ equals

$$F_1(u)(\tau - u) - F_1(\ell)(\tau - \ell) + \int_\ell^u F_1(x)dx \;=:\; \Delta(\tau), \tag{2.2}$$

which gives an alternate way of expressing $\delta(\tau) = \Delta(\tau) - (R(u) - R(\ell))$. (Another way of understanding the definition of $\Delta(\tau)$ is that it represents the difference of playing thresholds u and ℓ, assuming that $\mathbf{E}[X_2] = \tau$.) So, we define the estimate

$$\hat{\delta}(\tau) := \hat{\Delta}(\tau) - (\hat{R}_u - \hat{R}_\ell),$$

where $\hat{\Delta}$ uses the estimate $\hat{F}_1$ instead of F_1 in (2.2) and to estimate $\hat{R}_u$ and $\hat{R}_\ell$ we use empirical averages obtained in the current phase. The advantage is that besides the coarse knowledge of $\hat{F}_1$ we assumed to be given, we only need to choose thresholds from within our current confidence interval to obtain $\hat{\delta}$. Claim 2.4 will show that $\hat{\delta}(\tau)$ estimates $\delta(\tau)$ within an additive error of $O(\epsilon)$.

Completing the Proof of Lemma 2.2. Now we complete the missing details. We first prove that $|\delta(\tau)|$ gives an upper bound on one-round regret with threshold τ.

Claim 2.3. *If $\tau, \tau^* \in [\ell, u]$, then $R(\tau^*) - R(\tau) \le |\delta(\tau)|$.*

Proof. Consider $R(\tau^*) - R(\tau)$. The two settings are different only when X_1 is between τ^* and τ, and the difference of the reward is bounded by $|\tau^* - \tau|$. Therefore, $R(\tau^*) - R(\tau) \le |\tau^* - \tau| \cdot |F_1(\tau^*) - F_1(\tau)| \le |\tau^* - \tau| \cdot |F_1(u) - F_1(\ell)| = |\delta(\tau)|$, where the second inequality uses $\tau^*, \tau \in [\ell, u]$ implies $|F_1(\tau) - F_1(\tau^*)| \le |F_1(u) - F_1(\ell)|$. ☐

Next, we prove that $\hat{\delta}(\tau)$ is a good estimate of $\delta(\tau)$.

Claim 2.4. *In Algorithm 1, if the conditions in Lemma 2.2 hold then with probability $1 - T^{-10}$ we have $|\hat{\delta}(\tau) - \delta(\tau)| \le 5 \cdot \epsilon$ for all $\tau \in [\ell, u]$ simultaneously.*

Proof. Recall that $\delta(\tau) = \Delta(\tau) - \big(R(u) - R(\ell)\big)$. We first bound the error $|\hat{\Delta}(\tau) - \Delta(\tau)|$. Notice,

$$|\hat{\Delta}(\tau) - \Delta(\tau)| \le |F_1(u) - \hat{F}_1(u)| \cdot |\tau - u| + |F_1(\ell) - \hat{F}_1(\ell)| \cdot |\tau - \ell| + \int_\ell^u |F_1(x) - \hat{F}_1(x)| dx.$$

The main observation is that all three terms on the right-hand-side can be bounded by $T^{-\frac{1}{2}}$ since $|F_1(x) - \hat{F}_1(x)| \le T^{-\frac{1}{4}}$ and $u - \ell \le T^{-\frac{1}{4}}$. Hence, $|\hat{\Delta}(\tau) - \Delta(\tau)| \le 3T^{-\frac{1}{2}} \le 3\epsilon$.

Next, we bound the errors for $|\hat{R}_\ell - R(\ell)|$ and for $|\hat{R}_u - R(u)|$. For $|\hat{R}_\ell - R(\ell)|$, notice that $\hat{R}_\ell$ is an estimate of $R(\ell)$ with $N = C \cdot \frac{\log T}{\epsilon^2}$ samples. Since the reward of each sample is in $[0, 1]$, by Hoeffding's Inequality (Theorem A.1) the probability that $|\hat{R}_\ell - R(\ell)| > \epsilon$ is bounded by $2\exp(-2N\epsilon^2) = 2T^{-2C}$. Then, $|\hat{R}_\ell - R(\ell)| \le \epsilon$ holds with probability at least $1 - T^{-11}$ when $C > 10$. The error bound for $|\hat{R}_u - R(u)|$ is identical. Taking a union bound for two error for $|\hat{R}_\ell - R(\ell)|$ and for $|\hat{R}_u - R(u)|$, and then summing them with the error for $|\hat{\Delta}(\tau) - \Delta(\tau)|$ completes the proof. ☐

Now, we are ready to prove Lemma 2.2.

Proof of Lemma 2.2. We will assume that $|\hat{\delta}(\tau) - \delta(\tau)| \le 5\epsilon$, which is true with probability $1 - T^{-10}$ by Claim 2.4.

Observe that $\hat{\delta}(\tau)$ is a monotone increasing function because $\hat{\delta}'(\tau) = \hat{\Delta}'(\tau) = \hat{F}_1(u) - \hat{F}_1(\ell) \ge 0$. Therefore, according to the definition of ℓ' and u', we have $[\ell', u'] = \{\tau \in [\ell, u] : |\hat{\delta}(\tau)| \le 5\epsilon\}$. Now, we can use this property to prove the two statements of this lemma separately.

For Statement 1, notice that $\delta(\tau^*) = 0$. Claim 2.4 gives $|\hat{\delta}(\tau^*)| \le 5\epsilon$. Then, since $\tau^* \in [\ell, u]$ and $|\hat{\delta}(\tau^*)| \le 5\epsilon$, we must have $\tau^* \in [\ell', u']$ as $[\ell', u'] = \{\tau \in [\ell, u] : |\hat{\delta}(\tau)| \le 5\epsilon\}$.

Next, we prove Statement 2. By Claim 2.3, it suffices to bound $|\delta(\tau)|$ for all $\tau \in [\ell', u']$. By Claim 2.4, we have w.h.p. for all $\tau \in [\ell', u']$ that $|\delta(\tau)| \le |\hat{\delta}(\tau)| + 5\epsilon \le 10\epsilon$, where the last inequality uses the definition of ℓ' and u'. ☐

2.2 Doubling Framework for Low-Regret Algorithms In this section we show how to run Algorithm 1 for multiple phases with a doubling trick to get $O(\sqrt{T}\log T)$ regret. Instead of directly proving the regret bound for Prophet Inequality with $n = 2$, we first give a general doubling framework that will later be useful for Prophet Inequality and Pandora's Box problems with n random variables:

Lemma 2.5. *Consider an online learning problem with size n. Assume the one-round regret for every possible action is bounded by 1. Suppose there exists an action set-updating algorithm* Alg *satisfying: Given accuracy ϵ and action set A, algorithm* Alg *runs $\Theta(\frac{n^\alpha \log T}{\epsilon^2})$ rounds in A and outputs $A' \subseteq A$ satisfying the following with probability $1 - T^{-10}$:*

- *The optimal action in A belongs to A'.*
- *For $a \in A'$, the one-round regret of playing a is bounded by ϵ.*

Then, with probability $1 - T^{-9}$ the regret of Algorithm 2 is $O(n^{\alpha/2}\sqrt{T}\log T)$.

Algorithm 2: General Doubling Algorithm

Input: Time horizon T, problem size n, action space A, algorithm Alg, and parameter α.

```
Let i = 1, ε_1 = 1, A_1 = A
while ε_i > n^{α/2} log T / √T do
    Call Alg with input ε_i and A_i, and get output A_{i+1}
    ε_{i+1} ← ε_i / 2
    i ← i + 1
Run a ∈ A_i for the remaining rounds.
```

The proof of the lemma uses simple counting; see Section B.

Based on Lemma 2.5, we can immediately give the Bandit Prophet Inequality regret bound.

Theorem 2.6. *There exists an algorithm that achieves $O(\sqrt{T} \cdot \log T)$ regret with probability $1 - T^{-9}$ for Bandit Prophet Inequality problem with two distributions.*

Proof. The initialization runs $O(\sqrt{T}\log T)$ rounds, so the regret is $O(\sqrt{T}\log T)$. For the following interval shrinking procedure, Algorithm 1 matches the algorithm Alg described in Lemma 2.5 with $\alpha = 0$. Therefore, applying Lemma 2.5 completes the proof. □

2.3 Extending to Pandora's Box with a Fixed Order In order to extend the approach to Pandora's Box, in this section we consider a simplified problem with a *fixed box order*. There are two boxes taking values in $[0,1]$ from unknown distributions $\mathcal{D}_1, \mathcal{D}_2$ with cdfs F_1, F_2 and densities f_1, f_2. The boxes have known costs $c_1, c_2 \in [0,1]$. We assume that we always pay c_1 to observe X_1 (i.e., $\mathbf{E}[X_1] > c_1$), and then decide whether to observe X_2 by paying c_2. Indeed, it might be better to open the second box before the first box or not to open any box. We make these simplifying assumptions in this section to make the presentation cleaner. Generally, determining an approximately optimal order will be one of the main technical challenges that we will need to handle for general n in Section 4.

Formally, consider a T rounds game where in each round t we play a threshold $\tau^{(t)} \in [0,1]$ and receive as feedback the following utility:

- Independently draw $X_1^{(t)}$ from $\mathcal{D}_1$. If $X_1^{(t)} \geq \tau^{(t)}$, we stop and receive $X_1^{(t)} - c_1$ as the utility.
- Otherwise, we pay c_2 to see $X_2^{(t)}$ drawn independently from $\mathcal{D}_2$, and receive $\max\{X_1, X_2\} - (c_1 + c_2)$ as utility.

The only feedback we receive is the utility, and not even which random variable gets selected.

To see the optimal policy, define a *gain function* $g(v) := \mathbf{E}[\max\{0, X_2 - v\} - c_2]$ to represent the expected additional utility from opening X_2 assuming we already have $X_1 = v$, i.e.,

$$g(v) \;=\; -c_2 + \int_v^1 (x - v) f_2(x)dx \;=\; -c_2 + (1 - v) - \int_v^1 F_2(x)dx. \tag{2.3}$$

The optimal threshold (Weitzman's reservation value) τ^* is now the solution to $g(\tau^*) = 0$, i.e., $\mathbf{E}[\max\{X_2 - \tau^*, 0\}] = c_2$. Since our algorithm does not know $F_2(x)$ but only an approximate distribution $\hat{F}_2(x)$, we get an estimate $\hat{g}(v)$ of $g(v)$ by replacing $F_2(x)$ with $\hat{F}_2(x)$ in (2.3).

For $\tau \in [0, 1]$, let *reward function* $R(\tau)$ denote the expected reward of playing τ. With the definition of gain function $g(v)$ and linearity of expectation, we can write

$$R(\tau) \ := \ - c_1 + \mathbf{E}\left[X_1\right] + \int_0^{\tau} f_1(x) g(x) dx.$$

The *total regret* of our algorithm is now defined as $T \cdot R(\tau^*) - \sum_{t=1}^{T} R(\tau^{(t)})$.

Interval-Shrinking Algorithm. Starting with an initial confidence interval $[\ell, u]$ containing τ^*, we again design an Interval-Shrinking algorithm (Algorithm 3) that runs for $\Theta(\frac{\log T}{\epsilon^2})$ rounds and outputs a refined confidence interval $[\ell', u']$. We will show that this refined interval still contains τ^* and that the regret of playing any τ inside this refined interval is bounded by $O(\epsilon)$. Now we give the algorithm and the theorem.

Algorithm 3: Interval-Shrinking Algorithm for Pandora's Box

Input: Interval $[\ell, u]$, length m, and CDF estimates $\hat{F}_1(x), \hat{F}_2(x)$.

Run $C \cdot \frac{\log T}{\epsilon^2}$ rounds with $\tau = \ell$. Let $\hat{R}_\ell$ be the average reward.

Run $C \cdot \frac{\log T}{\epsilon^2}$ rounds with $\tau = u$. Let $\hat{R}_u$ be the average reward.

For $\tau \in [\ell, u]$, define $\hat{\Delta}(\tau) := (\hat{g}(u) - \hat{g}(\tau))\hat{F}_1(u) - (\hat{g}(\ell) - \hat{g}(\tau))\hat{F}_1(\ell) - \int_\ell^u \hat{g}'(x)\hat{F}(x)dx$.

For $\tau \in [\ell, u]$, define $\hat{\delta}(\tau) := \hat{\Delta}(\tau) - (\hat{R}_u - \hat{R}_\ell)$.

Let $\ell' = \min\{\tau \in [\ell, u] \text{ s.t. } \hat{\delta}(\tau) \geq -4\epsilon\}$ and let $u' = \max\{\tau \in [\ell, u] \text{ s.t. } \hat{\delta}(\tau) \leq 4\epsilon\}$.

Output: $[\ell', u']$

Lemma 2.7. *Suppose we are given:*

- Initial interval $[\ell, u]$ *satisfying* $\tau^* \in [\ell, u]$, *gain function* $|g(\tau)| \leq T^{-\frac{1}{4}}$, *and bounding function* $|\delta(\tau)| \leq 16\epsilon$ *where* δ *is defined in* (2.4).
- *CDF estimate* $\hat{F}_1(x)$ *which is constructed via* $1000 \cdot \frac{\log T}{\epsilon}$ *new i.i.d. samples of* X_1.
- *CDF estimate* $\hat{F}_2(x)$ *which is constructed via* $1000 \cdot \frac{\log T}{\epsilon}$ *new i.i.d. samples of* X_2.

Then, for $\epsilon > T^{-\frac{1}{2}}$, *Algorithm 3 runs thresholds inside* $[\ell, u]$ *for no more than* $10000 \cdot \epsilon^{-2} \log T$ *rounds and outputs with probability* $1 - T^{-10}$ *a sub-interval* $[\ell', u'] \subseteq [\ell, u]$ *satisfying:*

1. $\tau^* \in [\ell', u']$.
2. *Simultaneously for every* $\tau \in [\ell', u']$, *we have* $|\delta(\tau)| \leq 8\epsilon$.
3. *Simultaneously for every* $\tau \in [\ell', u']$, *the expected one-round regret of playing* τ *is at most* 8ϵ.

To understand the main idea of the proof, let's compare the expected reward of choosing the optimal threshold τ^* and an arbitrary threshold $\tau \in [\ell, u]$. The difference is given by

$$R(\tau^*) - R(\tau) \ = \ \int_0^{\tau^*} f_1(x) g(x) dx - \int_0^{\tau} f_1(x) g(x) dx \ = \ \int_{\tau}^{\tau^*} f_1(x) g(x) dx.$$

Note that g is non-increasing since $g'(x) = F_2(x) - 1 \leq 0$. So, using $\tau^*, \tau \in [\ell, u]$ imply $|F_1(\tau^*) - F_1(\tau)| \leq |F_1(\ell) - F_1(u)|$, we get $R(\tau^*) - R(\tau) \leq |(F_1(\ell) - F_1(u)) \cdot g(\tau)|$. This motivates defining *bounding function*

$$\delta(\tau) \ := \ (F_1(u) - F_1(\ell)) \cdot \big(g(\tau^*) - g(\tau)\big) \ = \ -(F_1(u) - F_1(\ell)) \cdot g(\tau), \tag{2.4}$$

and we get the following upper bound on the one-round regret when choosing τ instead of τ^*.

Claim 2.8. *If* $\tau, \tau^* \in [\ell, u]$ *then* $R(\tau^*) - R(\tau) \leq |\delta(\tau)|$.

In order to define an estimate $\hat{\delta}(\tau)$ that can be computed using the available information, again consider the rewards when playing thresholds u and ℓ. The difference is given by

$$R(u) - R(\ell) \;=\; \int_\ell^u f_1(x)g(x)dx \;=\; F_1(u)g(u) - F_1(\ell)g(\ell) - \int_\ell^u F_1(x)g'(x)dx$$
$$= \; F_1(u)g(u) - F_1(\ell)g(\ell) - \int_\ell^u F_1(x)\cdot(F_2(x)-1)dx.$$

Adding this equation with the definition of $\delta(\tau)$ gives $\delta(\tau) + R(u) - R(\ell)$ equals

$$F_1(u)\cdot\big(g(u)-g(\tau)\big) - F_1(\ell)\cdot\big(g(\ell)-g(\tau)\big) - \int_\ell^u F_1(x)\cdot(F_2(x)-1)dx \;=:\; \Delta(\tau), \tag{2.5}$$

which gives us an alternate way to express $\delta(\tau) = \Delta(\tau) - (R(u) - R(\ell))$. So, we define the estimate

$$\hat{\delta}(\tau) := \hat{\Delta}(\tau) - (\hat{R}_u - \hat{R}_\ell),$$

where $\hat{\Delta}$ uses the estimates $\hat{F}_1$ and $\hat{g}$ instead of F_1 and g in (2.5), and to estimate $(\hat{R}_u - \hat{R}_\ell)$ we use empirical averages obtained in the current phase. We have the following claim on the accuracy of $\hat{\delta}$ in Section B.2, which is similar to Claim 2.4.

Claim 2.9. *In Algorithm 3, if the conditions in Lemma 2.7 hold, then with probability* $1-T^{-10}$ $|\hat{\delta}(\tau)-\delta(\tau)| \le 4\epsilon$ *simultaneously for all* $\tau \in [\ell, u]$.

The proof of Claim 2.9 is different from Claim 2.4: After the initialization, it's not possible to give an initial confidence interval of length at most $T^{-\frac{1}{4}}$. So, we cannot prove an $O(T^{-\frac{1}{2}})$ accuracy for $\Delta(\tau)$. Instead, we use the fact that $\mathsf{Var}\Delta(\tau) \le O(\epsilon)$ to give an $O(\epsilon)$ accuracy bound using Bernstein inequality (Theorem A.2) for a single τ. To extend the bound to the whole interval, we discretize and apply a union bound. To avoid the dependency from the previous phases when discretizing, in each phase we use new samples to construct $\hat{F}_1$ and $\hat{F}_2$. This is the reason that we introduce sample sets in Algorithm 3.

Now the proof of Lemma 2.7 is similar to the proof of Lemma 2.2 via Claims 2.8 and 2.9.

Finally, we state the main theorem for Pandora's Box problem with two boxes in a fixed order.

Theorem 2.10. *For Bandit Pandora's Box learning problem with two boxes in a fixed order, there exists an algorithm that achieves* $O(\sqrt{T}\log T)$ *total regret.*

The proof of Theorem 2.10 is similar to Theorem 2.6: We first show that $\Theta(\sqrt{T}\log T)$ initial samples are sufficient to meet the conditions in Lemma 2.7. Combining this with Lemma 2.5 proves the theorem. See Section B.2 for details.

3 Prophet Inequality for General n

In the Bandit Prophet Inequality problem, there are n unknown independent distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$ taking values in $[0,1]$ with cdfs $F_1, \ldots, F_n$ and densities $f_1, \ldots, f_n$. Consider a T rounds game where round t we play thresholds $\boldsymbol{\tau}^{(t)} = (\tau_1^{(t)}, \tau_2^{(t)}, \ldots, \tau_{n-1}^{(t)}, \tau_n^{(t)} = 0)$ and receive the following reward: For $i \in [n]$, independently draw $X_i^{(t)}$ from $\mathcal{D}_i$. Let $j = \min\{i \in [n] : X_i^{(t)} \ge \tau_i^{(t)}\}$. $X_j^{(t)}$ is returned as the reward. The only feedback is the reward, and we do not see the index j of the selected random variable. Since we have $\tau_n^{(t)} = 0$, the algorithm will always select a value. In the following, we omit $\tau_n^{(t)}$ and only use $\boldsymbol{\tau}^{(t)} := (\tau_1^{(t)}, \tau_2^{(t)}, \ldots, \tau_{n-1}^{(t)})$ to represent a threshold setting.

Let Opt_i represent the optimal expected reward if only running on distributions $\mathcal{D}_i, \mathcal{D}_{i+1}, \ldots, \mathcal{D}_n$. Then, the optimal i-th threshold setting is exactly Opt_{i+1}. We can calculate $\{\mathsf{Opt}_{i+1}\}$ as follows:

- Let $\mathsf{Opt}_n = \mathbf{E}[X_n]$
- For $i = n-1 \to 1$: Let $\mathsf{Opt}_i = R(1, 1, \ldots, 1, \mathsf{Opt}_{i+1}, \mathsf{Opt}_{i+2}, \ldots, \mathsf{Opt}_n)$, where the function $R(\boldsymbol{\tau})$ represents the expected one-round reward under thresholds $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{n-1})$.

The *total regret* is defined

$$T \cdot \mathsf{Opt}_1 - \sum_{t=1}^{T} R(\boldsymbol{\tau}^{(t)}).$$

High-Level Approach. Following the doubling framework from Algorithm 2, we only need to design an initialization algorithm and a constraint-updating algorithm. For the initialization, we get $O(\mathsf{poly}(n)\sqrt{T}\log T)$ i.i.d. samples for each X_i by playing thresholds $(1, 1, \ldots, \tau_{i-1} = 1, \tau_i = 0, 0, \ldots, 0)$. Besides, we run $O(\mathsf{poly}(n)\sqrt{T}\log T)$ samples to get the initial confidence intervals with small length. For the constraint-updating algorithm, we reuse the idea from the $n = 2$ case where we shrink confidence intervals by testing X_i with thresholds ℓ_i or u_i. However, there are two major new challenges while testing X_i.

The first challenge while testing X_i is that we may stop early, and not get sufficiently many samples for X_i. Although the probability of reaching X_i could be very small, this also means that we will not reach X_i frequently. To avoid this problem, for $j < i$, we use the upper confidence bounds as thresholds since they maximize the probability of reaching X_i. In particular, it is at least as high as in the optimal policy. Therefore, we will be able to show that the probability term cancels in calculation, so the total loss from X_i can still be bounded.

The second challenge is that when we are testing X_i, we need to also set thresholds τ_j for $j > i$. The problem is that the optimal choice for τ_i depends on τ_j for $j > i$. To cope this this problem, in our algorithm we use the **lower confidence bounds** as thresholds for $j > i$. Formally, let Alg_i denote the expected reward if only running on distributions $\mathcal{D}_i, \ldots, \mathcal{D}_n$ with lower confidence bounds as the thresholds, i.e.,

$$\mathsf{Alg}_i \ := \ R(1, \ldots, 1, \tau_i = \ell_i, \tau_{i+1} = \ell_{i+1}, \ldots, \tau_{n-1} = \ell_{n-1}).$$

Now, under our threshold setting, we can only hope to learn Alg_{i+1}, while the optimal threshold is Opt_{i+1}. So, our key idea is to first get a new confidence interval for Alg_{i+1}. Then, since we have $\mathsf{Alg}_{i+1} \le \mathsf{Opt}_{i+1}$, the lower bound for Alg_{i+1} is also a lower bound for Opt_{i+1}. For the upper bound, we first bound the difference between Opt_{i+1} and Alg_{i+1}, and adding this difference to the upper bound for Alg_{i+1} gives the upper bound for Opt_{i+1}.

3.1 Interval-Shrinking Algorithm for General n In this section, we give the interval shrinking algorithm, and provide the regret analysis to show that we can get a new group of confidence intervals that achieves $O(\epsilon)$ regret after $\widetilde{O}(\frac{\mathsf{poly}(n)}{\epsilon^2})$ rounds. We first give the algorithm and the corresponding lemma.

Algorithm 4: Interval shrinking Algorithm for general n

Input: Intervals $[\ell_1, u_1], \ldots, [\ell_{n-1}, u_{n-1}]$, CDF estimates $\hat{F}_1(x), \ldots, \hat{F}_n(x)$, and ϵ.

For $i \in [n-1]$, define $\hat{P}_i := \prod_{j \in [i-1]} \hat{F}_j(u_j)$

for $i = n-1 \to 1$ **do**

- Run $C \cdot \frac{\log T}{\epsilon^2}$ rounds with thresholds $(u_1, \ldots, u_{i-1}, \ell_i, \ell'_{i+1}, \ldots \ell'_{n-1})$ and $C \cdot \frac{\log T}{\epsilon^2}$ rounds with $(u_1, \ldots, u_{i-1}, u_i, \ell'_{i+1}, \ldots \ell'_{n-1})$. Let $\hat{D}_i$ be the difference of the average rewards.
- For $\tau \in [\ell_i, u_i]$, define $\hat{\Delta}_i(\tau) := \hat{P}_i(\hat{F}_i(u_i)(\tau - u_i) - \hat{F}_i(\ell_i)(\tau - \ell_i) + \int_{\ell_i}^{u_i} \hat{F}_i(x)dx)$.
- For $\tau \in [\ell_i, u_i]$, define $\hat{\delta}_i(\tau) := \hat{\Delta}_i(\tau) - \hat{D}_i$.
- Let $\ell'_i = \min\{\tau \in [\ell_i, u_i] \text{ s.t. } \hat{\delta}_i(\tau) \ge -\epsilon\}$.
- Let $u'_i = \max\{\tau \in [\ell_i, u_i] \text{ s.t. } \hat{\delta}_i(\tau) \le (2n - 2i - 1)\epsilon\}$.

Output: $[\ell'_2, u'_2], \ldots, [\ell'_n, u'_n]$

Lemma 3.1. *Suppose we are given:*

- Distribution estimates $\hat{F}_i(x)$ *for* $i \in [n-1]$ *satisfying* $|\prod_{i \in S} \hat{F}_i(x) - \prod_{i \in S} F_i(x)| \le T^{-1/4}$ *for all* $x \in [0, 1]$ *and* $S \subseteq [n]$.
- Initial intervals $[\ell_i, u_i]$ *for* $i \in [n-1]$ *of length* $u_i - \ell_i \le T^{-1/4}$ *that satisfy* $\mathsf{Opt}_{i+1} \in [\ell_i, u_i]$ *and* $\mathsf{Alg}_{i+1} \in [\ell_i, u_i]$.

Then, for $\epsilon > 12T^{-\frac{1}{2}}$ *Algorithm 4 runs no more than* $1000 \cdot \frac{n \log T}{\epsilon^2}$ *rounds such that in each round the threshold* $\boldsymbol{\tau}$ *satisfies* $\tau_i \in [\ell_i, u_i]$ *for all* $i \in [n-1]$. *Moreover, with probability* $1 - T^{-10}$ *the following statements hold:*

(i) $\mathsf{Opt}_{i+1} \in [\ell'_i, u'_i]$ *for all* $i \in [n-1]$.
(ii) *Let* $\mathsf{Alg}'_i := R(1, \ldots 1, \ell'_i, \ldots, \ell'_{n-1})$ *for* $i \in [n-1]$. *Then* $\mathsf{Alg}'_{i+1} \in [\ell'_i, u'_i]$.
(iii) *For every threshold setting* $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{n-1})$ *where* $\tau_i \in [\ell'_i, u'_i]$, *the expected one-round regret of playing* $\boldsymbol{\tau}$ *is at most* $2n^2\epsilon$.

We first introduce some notation to prove Lemma 3.1. First, we define a single-dimensional function $R_i(\tau)$ to generalize reward function $R(\tau)$ from the $n = 2$ case in Section 2.1. Ideally, $R_i(\tau)$ should represent the reward of playing $\tau_i = \tau$, but thresholds τ_j for $j > i$ also affect its expected reward. So, to match the setting in Algorithm 4, we set thresholds $\tau_{i+1}, \ldots, \tau_{n-1}$ to be the updated lower bounds, i.e., define

$$R_i(\tau) \;:=\; R(1, \ldots, 1, \tau_i = \tau, \ell'_{i+1}, \ldots, \ell'_{n-1}).$$

Next, we introduce P_i, representing the maximum probability of observing X_i when we have confidence intervals $\{[\ell_i, u_i]\}$, i.e.,

$$P_i \;:=\; \textstyle\prod_{j=1}^{i-1} F_j(u_j).$$

Replacing F_j with $\hat{F}_j$ in this equation defines estimate $\hat{P}_i$.

Notice that P_i also equals the probability of reaching X_i when we play thresholds $\tau_j = u_j$ for all $j < i$ in Algorithm 4. So, the loss of playing a sub-optimal threshold τ_i will be $P_i \cdot (R_i(\mathsf{Alg}'_{i+1}) - R_i(\tau))$ because P_i is the probability of reaching X_i and Alg'_{i+1} is the optimal threshold when $\tau_j = \ell'_j$ for all $j > i$. We define the generalized *bounding function*:

$$\delta_i(\tau) \;:=\; P_i \cdot \big(F_i(u_i) - F_i(\ell_i)\big) \cdot (\tau - \mathsf{Alg}'_{i+1}).$$

We will show in Claim 3.2 below that $|\delta_i(\tau)|$ upper bounds $P_i \cdot (R_i(\mathsf{Alg}'_{i+1}) - R_i(\tau))$ for all $\tau \in [\ell_i, u_i]$. Since we don't know $\delta_i(\tau)$, we will estimate it by writing in a different way.

Consider the difference in expected rewards between $\tau_i = u_i$ and $\tau_i = \ell_i$ when the other thresholds are set to $\tau_j = u_j$ for $j < i$ and $\tau_j = \ell'_j$ for $j > i$. The difference between these two settings only comes from τ_i, so the expected difference is

$$\begin{aligned} P_i \cdot (R_i(u_i) - R_i(\ell_i)) \;&=\; P_i \cdot \left((F_i(u_i) - F_i(\ell_i))\mathsf{Alg}'_{i+1} - \int_{\ell_i}^{u_i} x f_i(x) dx \right) \\ &=\; P_i \cdot \left(F_i(u_i)(\mathsf{Alg}'_{i+1} - u_i) - F_i(\ell_i)(\mathsf{Alg}'_{i+1} - \ell_i) + \int_{\ell_i}^{u_i} F_i(x) dx \right). \end{aligned}$$

Adding this with $\delta_i(\tau)$ implies $\delta_i(\tau) + P_i \cdot (R_i(u_i) - R_i(\ell_i))$ equals

$$P_i \cdot \left(F_i(u_i)(\tau - u_i) - F_i(\ell_i)(\tau - \ell_i) + \int_{\ell_i}^{u_i} F_i(x) dx \right) \;=:\; \Delta_i(\tau), \tag{3.6}$$

which gives another way of writing $\delta_i(\tau) = \Delta_i(\tau) - P_i \cdot (R_i(u_i) - R_i(\ell_i))$. Since $\hat{D}_i$ from Algorithm 4 is the difference between average rewards of taking samples with $\tau_i = u_i$ and $\tau_i = \ell_i$, it is an unbiased estimator of $P_i \cdot (R_i(u_i) - R_i(\ell_i))$. So, we define estimate

$$\hat{\delta}_i(\tau) \;:=\; \hat{\Delta}_i(\tau) - \hat{D}_i,$$

where $\hat{\Delta}_i(\tau)$ is obtained by replacing F_i with $\hat{F}_i$ and P_i with $\hat{P}_i$ in (3.6).

Similar to Claim 2.3 and Claim 2.4, we introduce the following claims for Algorithm 4.

Claim 3.2. *For* $i \in [n-1]$, *if* $\mathsf{Alg}'_{i+1} \in [\ell_i, u_i]$ *and* $\tau \in [\ell_i, u_i]$, *then* $P_i \cdot \big(R_i(\mathsf{Alg}'_{i+1}) - R_i(\tau)\big) \leq |\delta_i(\tau)|$.

Proof. We only need to prove that $R_i(\mathsf{Alg}'_{i+1}) - R_i(\tau) \leq \frac{|\delta_i(\tau)|}{P_i} = \big(F_i(u_i) - F_i(\ell_i)\big) \cdot (\tau - \mathsf{Alg}'_{i+1})$. Now the proof is identical to Claim 2.3 by replacing function $R(\cdot)$ with $R_i(\cdot)$. □

Claim 3.3. *In Algorithm 4, if the conditions in Lemma 3.1 hold, then with probability* $1 - T^{-10}$*, we have* $|\hat{\delta}_i(\tau) - \delta_i(\tau)| \le \epsilon$ *simultaneously for all* $\tau \in [\ell_i, u_i]$.

Proof. There are two terms in $\delta_i(\tau) = \Delta_i(\tau) - P_i \cdot (R_i(u) - R_i(\ell))$. We prove that the error of each term is bounded by $\frac{\epsilon}{2}$ with high probability, which will complete the proof by a union bound.

We first bound $|\hat{\Delta}_i(\tau) - \Delta_i(\tau)|$. There are three terms in $\frac{\Delta_i(\tau)}{P_i} = F_i(u_i)(\tau - u_i) - F_i(\ell_i)(\tau - \ell_i) + \int_{\ell_i}^{u_i} F_i(x)dx$. Since the conditions in Lemma 3.1 guarantee that $|\hat{F}_i(x) - F_i(x)| \le T^{-1/4}$ and $u_i - \ell_i \le T^{-1/4}$, the error in each term is at most $\frac{1}{\sqrt{T}}$ and the total error $\left|\frac{\Delta_i(\tau)}{P_i} - \frac{\hat{\Delta}_i(\tau)}{\hat{P}_i}\right| \le \frac{3}{\sqrt{T}}$.

For P_i, the preconditions in Lemma 3.1 guarantee that $|\hat{P}_i - \hat{P}_i| \le T^{-1/4}$. Moreover, observe that $u_i - \ell_i \le T^{-1/4}$ implies that $\frac{|\Delta_i(\tau)|}{P_i} \le 3T^{-1/4}$. So,

$$\left|\hat{\Delta}_i(\tau) - \Delta_i(\tau)\right| \;\le\; \left|\hat{P}_i\left(\frac{\hat{\Delta}_i(\tau)}{\hat{P}_i} - \frac{\Delta_i(\tau)}{P_i}\right)\right| + \left|(\hat{P}_i - P_i)\frac{\Delta_i(\tau)}{P_i}\right| \;\le\; \frac{6}{\sqrt{T}} \;\le\; \frac{\epsilon}{2},$$

where the last inequality uses $\epsilon > 12T^{-\frac{1}{2}}$.

For $P_i \cdot (R_i(u_i) - R_i(\ell_i))$, note that $\hat{D}_i$ is an unbiased estimator of $P_i \cdot (R_i(u_i) - R_i(\ell_i))$ with $N = C \cdot \frac{\log T}{\epsilon^2}$ samples. So, by Hoefdding's Inequality,

$$\mathbf{Pr}\left[\left|\hat{D}_i - P_i \cdot (R_i(u_i) - R_i(\ell_i))\right| > \frac{\epsilon}{2}\right] \;\le\; 2\exp(-8N\epsilon^2) \;=\; 2T^{-8C}.$$

Thus, $|\hat{D}_i - P_i \cdot (R_i(u_i) - R_i(\ell_i))| \le \frac{\epsilon}{2}$ holds with probability $1 - T^{-10}$ when $C > 10$. ☐

Besides Claim 3.2 and Claim 3.3, we also need some other properties of Algorithm 4 to prove Lemma 3.1. The next claim shows that the expected reward of playing lower confidence bounds increases phase to phase.

Claim 3.4. *Assume the conditions in Lemma 3.1 and the bound in Claim 3.3 hold. Then, for* $i \in [n]$*, we have* $\mathsf{Alg}'_i \ge \mathsf{Alg}_i$.

Proof. We prove by induction for i going from n to 1. The base case $i = n$ holds because $\mathsf{Alg}'_n = \mathsf{Alg}_n = \mathsf{Opt}_n = \mathbf{E}[X_n]$ by definition.

For the induction step, assume that $\mathsf{Alg}'_{i+1} \ge \mathsf{Alg}_{i+1}$ by induction hypothesis. Observe that

$$\begin{aligned} R(1,\ldots,1,\ell_i,\ell_{i+1},\ldots,\ell_{n-1}) &= \mathbf{E}[X_i \cdot \mathbf{1}_{X_i>\ell_i}] + \mathbf{Pr}[X_i \le \ell_i]\, R(1,\ldots,1,1,\ell_{i+1},\ldots,\ell_{n-1}) \\ &\le \mathbf{E}[X_i \cdot \mathbf{1}_{X_i>\ell_i}] + \mathbf{Pr}[X_i \le \ell_i]\, R(1,\ldots,1,1,\ell'_{i+1},\ldots,\ell'_{n-1}) \\ &= R(1,\ldots,\ell_i,\ell'_{i+1},\ldots,\ell'_{n-1}), \end{aligned} \tag{3.7}$$

where the inequality uses induction hypothesis as $R(1,\ldots,1,1,\ell_{i+1},\ldots,\ell_{n-1}) = \mathsf{Alg}_{i+1} \le \mathsf{Alg}'_{i+1} = R(1,\ldots,1,1,\ell'_{i+1},\ldots,\ell'_{n-1})$.

Next, we have $R_i(\ell_i) \le R_i(\ell'_i)$, i.e.,

$$R(1,\ldots,1,\ell_i,\ell'_{i+1},\ldots,\ell'_{n-1}) \;\le\; R(1,\ldots,1,\ell'_i,\ell'_{i+1},\ldots,\ell'_{n-1}). \tag{3.8}$$

To prove this, we first observe that if $\ell_i = \ell'_i$, then the inequality is an equality. Otherwise, there must be $\hat{\delta}_i(\ell'_i) = -\epsilon$. Next, combining the definition of $\delta_i(\tau)$ and Claim 3.3, we have $\hat{\delta}_i(\mathsf{Alg}'_{i+1}) \ge \delta_i(\mathsf{Alg}'_{i+1}) - |\delta_i(\mathsf{Alg}'_{i+1}) - \hat{\delta}_i(\mathsf{Alg}'_{i+1})| \ge -\epsilon$. Since $\hat{\delta}'_i(\tau) = P_i \cdot (\hat{F}_i(u_i) - \hat{F}_i(\ell_i)) \ge 0$ means $\hat{\delta}_i(\tau)$ is increasing, there must be $\mathsf{Alg}'_{i+1} \ge \ell'_i$.

Now consider function $R_i(\tau)$. Recall that $R_i(\tau) = R(1,...1,\tau_i = \tau, \ell'_{i+1},...,\ell'_{n-1})$. Therefore,

$$R_i(\tau) \;=\; \mathbf{Pr}[X_i \le \tau] \cdot \mathsf{Alg}'_{i+1} + \mathbf{E}[X_i \cdot \mathbf{1}_{X_i>\tau}] \;=\; F_i(\tau) \cdot \mathsf{Alg}'_{i+1} + \int_\tau^1 f_i(x)\, x\, dx,$$

which means $R_i'(\tau) = f_i(\tau)(\mathsf{Alg}'_{i+1} - \tau)$, showing that $R_i(\tau)$ is a unimodular function and reaches its maximum when $\tau = \mathsf{Alg}'_{i+1}$. Hence, (3.8) holds because $\ell_{i+1} \le \ell'_{i+1} \le \mathsf{Alg}'_{i+1}$.

Combining (3.7) and (3.8) proves the claim. ☐

Next, we prove that $\mathsf{Alg}'_{i+1} \in [\ell_i, u_i]$, which is crucial for us to use Claim 3.2.

Claim 3.5. *Assume that the preconditions in Lemma 3.1 and the bound in Claim 3.3 hold, then* $\mathsf{Alg}'_{i+1} \in [\ell_i, u_i]$ *for all* $i \in [n-1]$.

Proof. Claim 3.4 shows that $\mathsf{Alg}_{i+1} \le \mathsf{Alg}'_{i+1}$. On the other hand, $\mathsf{Alg}'_{i+1} \le \mathsf{Opt}_{i+1}$ holds because Opt_{i+1} is the maximum achievable reward. Then, Claim 3.5 holds because $\mathsf{Opt}_{i+1}, \mathsf{Alg}_{i+1} \in [\ell_i, u_i]$ by the preconditions in Lemma 3.1. ☐

Finally, we show that Alg'_i cannot be much smaller than Opt_i.

Claim 3.6. *Assume that the preconditions in Lemma 3.1 and the bound in Claim 3.3 hold, then* $\mathsf{Opt}_i - \mathsf{Alg}'_i \le \frac{2(n-i)\epsilon}{P_i}$ *for all* $i \in [n-1]$.

Proof. We prove by induction for i going from n to 1. The base case $i = n$ holds because $\mathsf{Opt}_n = \mathsf{Alg}'_n = \mathbf{E}[X_n]$.

For the induction step, we assume that $\mathsf{Opt}_{i+1} - \mathsf{Alg}'_{i+1} \le \frac{2(n-i-1)\epsilon}{P_{i+1}}$ and would like to show that $\mathsf{Opt}_i - \mathsf{Alg}'_i \le \frac{2(n-i)\epsilon}{P_i}$. We first have

$$\begin{aligned} R(1,\ldots,1,\mathsf{Opt}_{i+1},\mathsf{Opt}_{i+2},\ldots,\mathsf{Opt}_n) &= \mathbf{E}\left[X_i \cdot \mathbf{1}_{X_i > \mathsf{Opt}_{i+1}}\right] + \mathbf{Pr}\left[X_i \le \mathsf{Opt}_{i+1}\right]\mathsf{Opt}_{i+1} \\ &\le \mathbf{E}\left[X_i \cdot \mathbf{1}_{X_i > \mathsf{Opt}_{i+1}}\right] + \mathbf{Pr}\left[X_i \le \mathsf{Opt}_{i+1}\right](\mathsf{Alg}'_{i+1} + \tfrac{2(n-i-1)\epsilon}{P_{i+1}}) \\ &\le \mathbf{E}\left[X_i \cdot \mathbf{1}_{X_i > \mathsf{Opt}_{i+1}}\right] + \mathbf{Pr}\left[X_i \le \mathsf{Opt}_{i+1}\right]\mathsf{Alg}'_{i+1} + \tfrac{2(n-i-1)\epsilon}{P_i} \\ &= R(1,\ldots,1,\mathsf{Opt}_{i+1},\ell'_{i+1},\ldots,\ell'_{n-1}) + \tfrac{2(n-i-1)\epsilon}{P_i}, \end{aligned} \tag{3.9}$$

where we use the induction hypothesis in the second line, and the fact that $\mathbf{Pr}\left[X_i \le \mathsf{Opt}_{i+1}\right] \le \mathbf{Pr}[X_i \le u_i] = \frac{P_{i+1}}{P_i}$ in the third line.

Next, since Alg'_{i+1} is the optimal threshold, we have

$$R(1,\ldots,1,\mathsf{Opt}_{i+1},\ell'_{i+2},\ldots,\ell'_{n-1}) \;\le\; R(1,\ldots,1,\mathsf{Alg}'_{i+1},\ell'_{i+2},\ldots,\ell'_{n-1}). \tag{3.10}$$

Finally,

$$|\delta(\ell'_i)| \;\le\; |\hat{\delta}(\ell'_i)| + |\hat{\delta}(\ell'_i) - \delta(\ell'_i)| \;\le\; \epsilon + \epsilon \;=\; 2\epsilon,$$

where the bound of $|\hat{\delta}(\ell'_i) - \delta(\ell'_i)|$ is from Claim 3.3, and the bound of $|\hat{\delta}(\ell'_i)|$ is from Algorithm 4. Combining this with Claim 3.2, we have $R_i(\mathsf{Alg}'_{i+1}) - R_i(\ell'_i) \;\le\; \frac{|\delta_i(\ell'_i)|}{P_i} \;\le\; \frac{2\epsilon}{P_i}$, which is exactly

$$R(1,\ldots,1,\mathsf{Alg}'_{i+1},\ell'_{i+1},\ldots,\ell'_{n-1}) \;\le\; R(1,\ldots,1,\ell'_i,\ell'_{i+1},\ldots,\ell'_{n-1}) + \frac{2\epsilon}{P_i}.$$

Summing this with (3.9) and (3.10) completes the induction step. ☐

Finally, we can prove Lemma 3.1.

Proof of Lemma 3.1. In this proof, we assume Claim 3.3 always holds. Then the whole proof should success with probability $1 - T^{-10}$.

We prove the three statements separately:

Statement (i). For the upper bound, Claim 3.6 shows that $\mathsf{Opt}_i - \mathsf{Alg}'_i \le \frac{2(n-i)\epsilon}{P_i}$. Therefore, $\delta_i(\mathsf{Opt}_{i+1}) \le P_i \cdot F_i(u_i) \cdot \frac{2(n-i-1)\epsilon}{P_{i+1}} = 2(n-i-1)\epsilon$. Combining this with Claim 3.3, we have $\hat{\delta}_i(\mathsf{Opt}_{i+1}) \le 2(n-i-1)\epsilon + \epsilon = (2n-2i-1)\epsilon$. Then $\mathsf{Opt}_{i+1} \le u'_i$, because $\mathsf{Opt}_{i+1} \in [\ell_i, u_i]$, $u'_i = \max\{\tau : \tau \in [\ell_i, u_i] \wedge \hat{\delta}_i(\tau) \le (2n-2i-1)\epsilon\}$ and the monotonicity of $\hat{\delta}_i$ function.

For the lower bound, at least we have $\mathsf{Opt}_{i+1} \ge \mathsf{Alg}'_{i+1}$. Therefore, $\delta_i(\mathsf{Opt}_{i+1}) \ge 0$, so $\hat{\delta}_i(\mathsf{Opt}_{i+1}) \ge -\epsilon$. Then $\mathsf{Opt}_{i+1} \ge \ell'_i$, because $\mathsf{Opt}_{i+1} \in [\ell_i, u_i]$, $\ell'_i = \max\{\tau : \tau \in [\ell_i, u_i] \wedge \hat{\delta}_i(\tau) \ge -\epsilon\}$ and the monotonicity of $\hat{\delta}_i$ function. Combining the two bounds proves Statement (i).

Statement (ii). The proof idea is the same as Statement (i). Notice that $\delta_i(\mathsf{Alg}'_{i+1}) = 0$. Then, according to Claim 3.3, $|\hat{\delta}_i(\mathsf{Alg}'_{i+1})| \le \epsilon$. So Statement (ii) hold because $\mathsf{Alg}'_{i+1} \in [\ell_i, u_i]$, which is from Claim 3.5, and $[\ell'_i, u'_i] \supseteq \{\tau \in [\ell_i, u_i] : |\hat{\delta}_i(\tau)| \le \epsilon\}$.

Statement (iii). We prove the following stronger statement by induction on i: If $\tau_j \in [\ell'_j, u'_j]$ for all $j \in \{i, \dots, n\}$, then

$$\mathsf{Alg}'_i \; - \; R(1, \dots 1, \tau_i, \dots, \tau_{n-1}) \; \le \; \frac{(n-i+1)^2\epsilon}{P_i}.$$

When the statement above holds, taking $i = 1$ gives $R(\tau_1, \dots, \tau_{n-1}) \ge \mathsf{Alg}'_1 - n^2\epsilon$. Furthermore, Claim 3.6 shows that $\mathsf{Alg}'_1 \ge \mathsf{Opt}_1 - 2(n-1)\epsilon$. Combining these two inequalities proves Statement (iii).

It remains to prove the induction statement. The base case $i = n$ holds trivially.

For the induction step, we will assume that the statement holds for $i + 1$ and we have to show it also holds for i. By induction hypothesis,

$$\begin{aligned} R(1, \dots, 1, \tau_i, \dots, \tau_{n-1}) \; &= \; \mathbf{E}\left[X_i \cdot \mathbf{1}_{X_i \ge \tau_i}\right] + \mathbf{Pr}\left[X_i < \tau_i\right] R(1, \dots, 1, \tau_{i+1}, \dots, \tau_{n-1}) \\ &\ge \mathbf{E}\left[X_i \cdot \mathbf{1}_{X_i \ge \tau_i}\right] + \mathbf{Pr}\left[X_i < \tau_i\right] \left(R(1, \dots, 1, \ell'_{i+1}, \dots, \ell'_{n-1}) - \tfrac{(n-i)^2\epsilon}{P_{i+1}}\right) \\ &\ge \mathbf{E}\left[X_i \cdot \mathbf{1}_{X_i \ge \tau_i}\right] + \mathbf{Pr}\left[X_i < \tau_i\right] R(1, \dots, 1, \ell'_{i+1}, \dots, \ell'_{n-1}) - \tfrac{(n-i)^2\epsilon}{P_i} \\ &= R(1, \dots, 1, \tau_i, \ell'_{i+1}, \dots, \ell'_{n-1}) - \tfrac{(n-i)^2\epsilon}{P_i}. \end{aligned}$$

Furthermore, $|\delta_i(\tau_i)| \le |\hat{\delta}_i(\tau_i)| + \epsilon$ by Claim 3.3 and $|\hat{\delta}_i(\tau)| \le (2n-2i-1)\epsilon$ by the definitions of ℓ'_i and u'_i, which means $|\delta_i(\tau)| \le 2(n-i)\epsilon$. So, Claim 3.2 implies

$$R(1, \dots, 1, \mathsf{Alg}'_{i+1}, \ell'_{i+1}, \dots, \ell'_{n-1}) - R(1, \dots, 1, \tau_i, \ell'_{i+1}, \dots, \ell'_{n-1}) \; \le \; \frac{2(n-i)\epsilon}{P_i}.$$

Finally, using $R(1, \dots, 1, \mathsf{Alg}'_{i+1}, \ell'_{i+1}, \dots, \ell'_{n-1}) \ge \mathsf{Alg}'_i$, we get

$$R(1, \cdots, 1, \tau_i, \dots, \tau_{n-1}) \; \ge \; \mathsf{Alg}'_i - \frac{2(n-i)\epsilon}{P_i} - \frac{(n-i)^2\epsilon}{P_i} \; \ge \; \mathsf{Alg}'_i - \frac{(n-i+1)^2\epsilon}{P_i}.$$ □

3.2 Initialization and Putting Everything Together Now, we can give the initialization algorithm. The main goal of the initialization is to satisfy the conditions listed in Lemma 3.1. Starting from the second call of Algorithm 4, the confidence interval length constraint and the distribution estimates constraints hold from the initialization, and the constraints $\mathsf{Opt}_{i+1}, \mathsf{Alg}_{i+1} \in [\ell_i, u_i]$ are guaranteed by Statements (i) and (ii) in Lemma 3.1. Then, we can apply Lemma 2.5 to bound the total regret.

We first give the initialization algorithm:

Lemma 3.7. *Algorithm 5 runs $O(n^3\sqrt{T}\log T)$ rounds. The output satisfies with probability $1 - T^{-10}$ all constraints listed in Lemma 3.1.*

Proof. For the accuracy bound of $\hat{F}_i(x)$, we first show that $|\hat{F}_i(x) - F_i(x)| \le \frac{T^{-1/4}}{2n}$ with probability $1 - T^{-11}$

Algorithm 5: Initialization

Input: Time horizon T, problem size n.

for $i = 1 \to n$ **do**

  Run $1000n^2\sqrt{T}\log T$ free samples for X_i to estimate $\hat{F}_i(x)$.

for $i = n-1 \to 1$ **do**

  Run $1000n^2\sqrt{T}\log T$ samples under the threshold setting $(1, \ldots, 1, \tau_{i+1} = \ell'_{i+1}, \ldots, \tau_{n-1} = \ell'_{n-1})$. Let μ_i be the average reward.

  Let $\ell_i = \mu_i - \frac{T^{-1/4}}{10n}$, $u_i = \mu_i + (2n-2i-1) \cdot \frac{T^{-1/4}}{10n}$

Output: $[\ell_1, u_1], \ldots, [\ell_{n-1}, u_{n-1}]$.

after running $N = C \cdot n^2\sqrt{T}\log T$ samples with $C = 1000$. With DKW inequality (Theorem A.3), we have

$$\mathbf{Pr}\left[|\hat{F}_i(x) - F_i(x)| > \varepsilon \;=\; \frac{T^{-\frac{1}{4}}}{2n}\right] \;\le\; 2\exp(-2N\varepsilon^2) \;=\; 2T^{-C/4}.$$

So the bound holds with probability $1 - T^{-12}$ when $C = 1000$. By the union bound, with probability $1 - T^{-11}$, we have $|\hat{F}_i(x) - F_i(x)| \le \frac{T^{-\frac{1}{4}}}{2n}$ holds for every $i \in [n]$. Then, for the accuracy of $\prod_{i\in S} F_i(x)$, we have $\big((1 - \frac{T^{-\frac{1}{4}}}{2n})^n - 1\big) \le \prod_{i\in S}\hat{F}_i(x) - \prod_{i\in S} F_i(x) \le \big((1 + \frac{T^{-\frac{1}{4}}}{2n})^n - 1\big)$. For the lower bound, we have $(1 - \frac{T^{-\frac{1}{4}}}{2n})^n - 1 \ge 1 - \frac{T^{-\frac{1}{4}}}{2} - 1 > -T^{-\frac{1}{4}}$. For the upper bound, we have $(1 + \frac{T^{-\frac{1}{4}}}{2n})^n - 1 \le \exp(\frac{T^{-\frac{1}{4}}}{2n} \cdot n) - 1 \le 1 + 2 \cdot \frac{T^{-\frac{1}{4}}}{2} - 1 = T^{-\frac{1}{4}}$. Combining two bounds finishes the proof.

For the confidence interval, the constraints $u_i - \ell_i \le T^{-\frac{1}{4}}$ hold by definition. Then, it only remains to show $\mathsf{Opt}_{i+1} \in [\ell_i, u_i]$ and $\mathsf{Alg}_{i+1} \in [\ell_i, u_i]$.

We start from proving $\mathsf{Alg}_{i+1} \in [\ell_i, u_i]$. Notice that μ_i is an estimate of Alg_{i+1} with $N = C \cdot n^2\sqrt{T}\log T$ samples with $C = 1000$. With Hoeffding's Inequality (Theorem A.1), we have

$$\mathbf{Pr}\left[|\mu_i - \mathsf{Alg}_{i+1}| > \varepsilon = \frac{T^{-1/4}}{10n}\right] \;<\; 2\exp(-2N\varepsilon^2) \;=\; 2T^{-C/50}.$$

Notice that $\ell_i = \mu_i - \frac{T^{-1/4}}{10n}$ and $u_i = \mu_i + \frac{T^{-1/4}}{10n}$. Then, by the union bound for all $i \in [n]$, we have $\mathsf{Alg}_{i+1} \in [\ell_i, u_i]$ holds for all i with probability $1 - T^{-11}$ when $C \ge 1000$.

For Opt_{i+1}, we prove the statement by doing induction with the assumption that $|\mathsf{Alg}_{i+1} - \mu_i| \le \frac{T^{-1/4}}{10n}$ for all i. The base case is $i = n$, the statement simply holds because $\mathsf{Alg}_n = \mathsf{Opt}_n$. Next, we consider i, with the condition that $\mathsf{Opt}_{j+1} \in [\ell_j, u_j]$ for all $j > i$. For the lower bound, since we know that $\mathsf{Alg}_{i+1} \ge \ell_i$, there must be $\mathsf{Opt}_{i+1} \ge \ell_i$, because $\mathsf{Opt}_{i+1} \ge \mathsf{Alg}_{i+1}$. For the upper bound, we first bound the difference between Alg_{i+1} and Opt_{i+1}. Consider the setting $(1, \ldots 1, \tau_{i+1} = \ell_{i+1}, \ldots, \tau_{n-1} = \ell_{n-1})$ and $(1, \ldots, 1, \tau_{i+1} = \mathsf{Opt}_{i+2}, \ldots, \tau_{n-1} = \mathsf{Opt}_n)$. The first setting incurs an extra loss only when its behavior is different from the second setting. Assume the two settings behave differently when meeting a threshold τ_j. Notice that this extra loss is bounded by $|\ell_j - \mathsf{Opt}_{j+1}|$. Since $\mathsf{Opt}_{j+1} \in [\ell_j, u_j]$ for all $j > i$, this difference is upper bounded by $\max_{j>i} u_j - \ell_j = u_{i+1} - \ell_{i+1} = (2n-2i-2) \cdot \frac{T^{-1/4}}{10n}$. Therefore,

$$\mathsf{Opt}_{i+1} \;\le\; \mathsf{Alg}_{i+1} + (2n-2i-2) \cdot \frac{T^{-1/4}}{10n} \;\le\; \mu_i + (2n-2i-1) \cdot \frac{T^{-1/4}}{10n} \;=\; u_i.$$

Combining the lower bound and the upper bound proves $\mathsf{Opt}_{i+1} \in [\ell_i, u_i]$. Finally, taking union bounds for all events that hold with probability $1 - T^{-11}$ finishes the proof. □

Now we are ready to prove the main theorem.

Theorem 1.1. *There is a polytime algorithm with $O(n^3\sqrt{T}\log T)$ regret for the Bandit Prophet Inequality problem* *where we only receive the selected value as the feedback.*

Proof. For the initialization, Algorithm 5 runs $O(n^3\sqrt{T}\log T)$ rounds, so the total regret from the initialization is $O(n^3\sqrt{T}\log T)$.

For the main algorithm, we run Algorithm 2 with Algorithm 4 being the required sub-routine Alg. This is feasible because the requirements in Lemma 3.1 are guaranteed by the initialization and Lemma 3.1 itself. Besides, Lemma 3.1 implies that Algorithm 4 upper-bound the one-round regret by ϵ after $O(\frac{n^5 \log T}{\epsilon^2})$ samples. Applying Lemma 2.5 with $\alpha = 5$, we have the $O(n^{2.5}\sqrt{T}\log T)$ regret bound. Combining two parts finishes the proof. □

4 Pandora's Box for General n

In the Bandit Pandora's Box problem, there are n unknown independent distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$ representing the values of the n boxes. The distributions have cdfs $F_1, \ldots, F_n$ and densities $f_1, \ldots, f_n$. Moreover, each box/distribution $\mathcal{D}_i$ has a known inspection cost c_i. Although in the original problem in introduction we assumed that the values and costs have support $[0, 1]$, in this section we will scale down the costs and values by a factor of $2n$, so that they have support $[0, \frac{1}{2n}]$. This scaling helps to bound the utility in each round between $[-0.5, 0.5]$. To obtain bounds for the original unscaled problem, we will multiply our bounds with this factor $2n$ in the final analysis.

Consider a T rounds game where in each round we play some permutation π representing the order of inspection and n thresholds $(\tau_{\pi(1)}, \ldots, \tau_{\pi(n)})$. Our algorithm receives the following utility as feedback: For $i \in [n]$, draw $X_\pi(i) \sim \mathcal{D}_\pi(i)$. Let j be the minimum index that satisfies $\max\{X_\pi(1), \ldots, X_\pi(j-1)\} \geq \tau_{\pi(j)}$. If such j does not exist, j is set to be $n+1$ (all boxes opened). The utility we receive in this round is $\max\{X_\pi(1), \ldots, X_\pi(j-1)\} - \sum_{k<j} c_{\pi(k)}$.

Note that the only feedback is the utility, and we do not see any value or even the index j where we stop.

In the case of known distributions, the optimal one-round policy for this problem was designed by Weitzman [Wei79]: For every distribution $\mathcal{D}_i$, solve the equation $\mathbf{E}[\max\{X_i - \sigma_i, 0\}] = c_i$; now play permutation π by sorting in decreasing order of σ_i and set threshold $\tau_i^* = \sigma_i$. Let OPT be the optimal expected reward according to this optimal policy. Let ALG_t be the expected reward of our policy in the t-th round. Then, we want to design an algorithm with *total regret* $T \cdot OPT - \sum_{t\in T} ALG_t$ at most $\widetilde{O}(\mathsf{poly}(n)\sqrt{T})$.

Before introducing the algorithm, we define the *gain function* for this general case:

$$g_i(v) \;:=\; -c_i + \int_v^1 (x - v) f_i(x)dx \;=\; -c_i + (1 - v) - \int_v^1 F_i(x)dx. \tag{4.11}$$

Similar to the $n = 2$ case, this *gain function* is the expected additional utility we get on opening X_i when we already have value v in hand. Note that the optimal threshold τ_i^* satisfies $g_i(\tau_i^*) = 0$.

4.1 High-Level Approach via Valid Policies. We first briefly introduce the initialization algorithm. The following lemma shows what we achieve in the initialization (proved in Section C.1).

Lemma 4.1. *The initialization algorithm runs $1000 \cdot \sqrt{T}\log T$ samples for each distribution to output interval $[\ell_i, u_i]$, such that with probability $1 - T^{-10}$ the following hold simultaneously for all $i \in [n]$:*

- $\ell_i \leq \sigma_i \leq u_i$.
- $|g_i(x)| \leq T^{-\frac{1}{4}}$ *simultaneously for all $x \in [\ell_i, u_i]$.*

After initialization, the main part is the action set-updating algorithm. Similar to the algorithm for $n = 2$, we hope to use estimates $\hat{F}_i(x)$ to gradually shrink the intervals $[\ell_i, u_i]$. However, one major challenge is that we don't have a fixed order. If n is a constant, we can just simply try all possible permutations and use a multi-armed bandit style algorithm to find the optimal permutation. But the number of permutations is exponential in n, so this approach is impossible when n is a general parameter. To get a polynomial regret algorithm, we can only test $\mathsf{poly}(n)$ number of different orders.

Another challenge is that the idea for $n = 2$ can bound the regret when we play a sub-optimal threshold, but it tells nothing about playing a sub-optimal order. We don't have a direct way to bound the regret when playing an incorrect order.

Both difficulties imply that only keeping the confidence intervals as the constraint for the actions is not enough. Therefore, we also introduce a set of order constraints:

Definition 4.2 (Valid Constraint Group). *Given a set of confidence intervals $I = \{[\ell_1, u_1], [\ell_2, u_2], ..., [\ell_n, u_n]\}$ and a set S of order constraints, satisfying:*

- $u_i - \ell_i \le T^{-\frac{1}{4}}$.
- $\sigma_i \in [\ell_i, u_i]$
- *Every constraint in S can be defined as (i, j) that means $\sigma_i > \sigma_j$.*
- *The constraints in S are closed, i.e., if $(i, j), (j, k) \in S$, there must be $(i, k) \in S$.*
- *If $(i, j) \in S$, we must have $u_i \ge u_j$ and $\ell_i \ge \ell_j$.*

For (I, S) satisfying the conditions above, we call it a valid constraint group.

The intuition of the extra order constraints is: When we are shrinking the intervals, if it is evident that $\sigma_i > \sigma_j$, we will require $\mathcal{D}_i$ to be in front of $\mathcal{D}_j$ in the following rounds. Correspondingly, we give the following definition for a "valid" policy. During the algorithm, we will only run valid policies, according to the current constraint group we have.

Definition 4.3 (Valid Policy). *Let $(\tau_{\pi(1)}, \tau_{\pi(2)}, ..., \tau_{\pi(n)})$ be a policy to play in one round, where π is the distribution permutation for this policy, and the threshold in front of box $\pi(i)$ is $\tau_{\pi(i)}$. For simplicity, we use π to represent a policy.*

For a policy π, we say it is valid *for a constraint group (I, S) if the following conditions hold:*

- *For $i \in [n]$, $\tau_{\pi(i)} \in [\ell_{\pi(i)}, u_{\pi(i)}]$.*
- *If $(i, j) \in S$, then $\mathcal{D}_i$ must be in front of $\mathcal{D}_j$, i.e., $\pi^{-1}(i) < \pi^{-1}(j)$.*
- *For $i < j$, $\tau_{\pi(i)} \ge \tau_{\pi(j)}$.*

Notice that for a valid constraint group, we have $\sigma_i \in [\ell_i, u_i]$ for all $i \in [n]$, and $\sigma_i > \sigma_j$ for all $(i, j) \in S$. Then, the optimal policy is valid. Therefore, we can always find a valid policy from the constraint group.

Now, we are ready to give the main idea of the constraint-updating algorithm. In each phase, we first update the confidence intervals and then update the order constraints as follows:

- Step 1: For each $i \in [n]$, we run $\widetilde{O}(\frac{\mathsf{poly}(n)}{\epsilon^2})$ samples to update the confidence interval to $[\ell'_i, u'_i]$, such that for every threshold pair $\tau_i, \tau'_i \in [\ell'_i, u'_i]$, the *moving difference* is small, i.e., if we move τ_i to τ'_i and keep the validity, the difference of the expected reward is bounded by $O(\mathsf{poly}(n) \cdot \epsilon)$.
- Step 2: For each distribution pair (i, j) without a constraint, we run $\widetilde{O}(\frac{\mathsf{poly}(n)}{\epsilon^2})$ samples to test the order between them, such that we can either clarify which one is bigger between σ_i and σ_j, or we can claim that the *swapping difference* (the difference before and after swapping $\mathcal{D}_i$ and $\mathcal{D}_j$) is bounded by $O(\mathsf{poly}(n) \cdot \epsilon)$.

Finally, we argue that for every valid policy, we can convert it into the optimal policy by using $\mathsf{poly}(n)$ number of moves and swaps. This is sufficient for us to give $O(\mathsf{poly}(n) \cdot \epsilon)$ regret bound.

In the following analysis, we use separate sub-sections to introduce each part. Section 4.2 provides the Interval-Shrinking algorithm to bound the moving difference. Section 4.3 introduces the way to add a new order constraint to bound the swapping difference. Section 4.4 shows how to convert a valid policy to the optimal policy using a $\mathsf{poly}(n)$ number of moves and swaps. Finally, Section 4.5 combines the results of three sub-sections to complete the analysis.

4.2 Step 1: Interval-Shrinking to Bound Moving Difference The goal of this sub-section is: Given $i \in [n]$ and an original constraint group (I, S), we want to update the confidence interval $[\ell_i, u_i]$, to make sure that moving τ_i inside the new confidence interval incurs a small difference. The key idea of the Interval-Shrinking algorithm is similar to the case when $n = 2$: For each $i \in [n]$, we want to play two different values for τ_i, and see the difference of the expected reward. However, playing $\tau_i = \ell_i$ and $\tau_i = u_i$ might be impossible. The reason is: We hope to keep a decreasing threshold setting. There may not be a policy that allow τ_i to be set to u_i and ℓ_i without changing other thresholds. If we need different permutations to test $\tau_i = u_i$ and $\tau_i = \ell_i$, this makes the analysis involved. Therefore, we should find a policy that fixes the order and other thresholds, then test τ_i under this fixed policy while keeping a decreasing thresholds.

When we set τ_i to be different values, the two policies will be different only when the maximum reward before τ_i falls between the two thresholds. Therefore, to see the largest difference, we hope the probability of this event is maximized. This intuition allows us to give the following definition:

Definition 4.4 (MoveBound Policy)**.** *Given (I, S) and $i \in [n]$, a MoveBound policy is a valid partial policy π parameterized by ℓ and u[5], such that $F_{\pi,i}(u) - F_{\pi,i}(\ell)$ is maximized.*

In the definition, $F_{\pi,i}(x)$ is the probability that the algorithm reaches distribution X_i with maximum value $v < x$ in hand, i.e.,

$$F_{\pi,i}(x) \;:=\; \prod_{j<\pi^{-1}(i)} F_{\pi(j)}(x).$$

Furthermore, u and ℓ represents two possible value of τ_i to keep a valid π, i.e., π is valid when both $\tau_i = u$ and $\tau_i = \ell$.

A key fact of MoveBound policy is that for every different distribution, we might find a different MoveBound policy. This is different from the Prophet Inequality problem: In the Pandora's Problem, we don't keep a fixed order. Every order that satisfies the constraints (I, S) is possible to be tested.

Now, the key idea of the Interval-Shrinking algorithm is clear: For each i, find the MoveBound policy and run samples with $\tau_i = u$ and $\tau_i = \ell$. Then, use a method similar to Algorithm 3 to calculate the new interval. The following algorithm describes the details of this idea:

Algorithm 6: Interval-Shrinking Algorithm

Input: (I, S), ϵ, i, $\hat{F}_1(x), ..., \hat{F}_n(x)$
1. Get an *approximate* MoveBound policy $\hat{\pi}$ and ℓ, u using Lemma 4.8.
2. Calculates $\hat{F}_{\hat{\pi},i}(x)$.
3. For $\tau \in [\ell_i, u_i]$, let $\hat{\Delta}_i(\tau) := \hat{F}_{\hat{\pi},i}(u) \int_\tau^u (\hat{F}_i(x) - 1)dx + \hat{F}_{\hat{\pi},i}(\ell) \int_\ell^\tau (\hat{F}_i(x) - 1)dx - \int_\ell^u \hat{F}_{\hat{\pi},i}(x)(\hat{F}_i(x) - 1)dx$.
4. Run $C \cdot \epsilon^{-2} \log T$ samples with $\tau_i = u$. Let the average reward be $\hat{R}_u$.
5. Run $C \cdot \epsilon^{-2} \log T$ samples with $\tau_i = \ell$. Let the average reward be $\hat{R}_\ell$.
6. Define $\hat{\delta}_i(\tau) := \hat{\Delta}_i(\tau) - (\hat{R}_u - \hat{R}_\ell)$.
7. Let $u'_i = \max_{\tau \in [\ell_i, u_i]} |\hat{\delta}_i(\tau)| < \epsilon$ and let $\ell'_i = \min_{\tau \in [\ell_i, u_i]} |\hat{\delta}_i(\tau)| < \epsilon$.

Output: $[\ell'_i, u'_i]$

Then, the following lemma shows the bound when modifying a threshold:

Lemma 4.5 (Moving Difference Bound)**.** *Suppose we are given (I, S), $\epsilon > 16T^{-\frac{1}{2}}$, CDF estimates $\hat{F}_1(x), \cdots, \hat{F}_n(x)$, and $i \in [n]$, satisfying the following conditions for all $j \in [n]$:*

- *$|g_j(\tau)| \le T^{-\frac{1}{4}}$ for all $\tau \in [\ell_j, u_j]$.*
- *(I, S) is valid.*
- *For any valid partial policy π' of (I, S), we fix the order and the other thresholds except τ_j. Assume π' is valid when both $\tau_j = \ell'$ and $\tau_j = \ell'$. Define $\delta_{\pi',u',\ell',j}(\tau) = (F_{\pi',j}(\ell') - F_{\pi',j}(u'))g_i(\tau)$. Then $|\delta_{\pi',u',\ell',j}(\tau)| \le 6\epsilon$.*
- *CDF estimate $\hat{F}_j(x)$ is constructed via $10^5 \cdot \frac{n^2 \log T}{\epsilon}$ fresh i.i.d. samples of X_j.*

Then, Algorithm 6 runs $O(\frac{\log T}{\epsilon^2})$ samples and calculates a new interval $[\ell'_i, u'_i]$, such that the following properties hold with probability $1 - T^{-11}$:

(i) $\sigma_i \in [\ell'_i, u'_i]$

[5] Here, we say π is a partial policy because it's not completely fixed. We fix the permutation of the distributions and the value of all other thresholds, but the value of τ_i is flexible.

(ii) *Let $I'_i = (I \setminus \{[\ell_i, u_i]\}) \cup \{[\ell'_i, u'_i]\}$. For any valid partial policy π' of (I'_i, S), we fix the order and the other thresholds. Assume π' is valid when both $\tau_i = u'$ and $\tau_i = \ell'$. Define $\delta_{\pi',u',\ell',i}(\tau) = (F_{\pi',i}(\ell') - F_{\pi',i}(u'))g_i(\tau)$. Then $|\delta_{\pi',u',\ell',i}(\tau)| \le 3\epsilon$.*

(iii) *For any valid policy of (I'_i, S), if we fix the order and the other thresholds, but modify τ_i to τ'_i, satisfying that the new policy is still valid, the difference of the expected reward between these two policies is less than 3ϵ.*

Before starting the proof, we first give an accuracy bound of the distribution estimates, which is proved in Section C.2.

Claim 4.6. *Assume the preconditions in Lemma 4.5 hold. Then with probability $1 - T^{-12}$, we have $|\prod_{i\in S}\hat{F}_i(x) - \prod_{i\in S}F_i(x)| \le \sqrt{\epsilon}$ simultaneously hold for all $x \in [0,1]$ and $S \subseteq [n]$.*

Proof of Lemma 4.5. Fix the MoveBound policy π. Assume we want to move τ_i from $\tau_i = u$ to $\tau_i = \ell$, such that the policies are both valid when $\tau_i = \ell$ and $\tau_i = u$. Since we only care about the absolute value of the difference between two expected rewards, we may assume $u > \ell$.

If moving τ_i from u to ℓ, the performance of the two policies will only be different if the previous maximum reward falls between ℓ and u: It will reject the previous maximum if $\tau_i = u$, but accept it when $\tau_i = \ell$. Besides, since ℓ is greater than the next threshold in π, when the previous maximum is inside $[\ell, u]$, the algorithm must stop before the next threshold, which means the difference only comes from τ_i and X_i.

Recall that $F_{\hat{\pi},i}(x) = \prod_{j<\hat{\pi}^{-1}(i)} F_j(x)$, i.e., $F_{\hat{\pi}(i)}(x)$ is the probability that Algorithm 4 reaches τ_i with $v \le x$ in hand. Let $f_{\hat{\pi},i}(x) = F'_{\hat{\pi},i}(x)$. Then, the difference of the expected reward is $\int_\ell^u f_{\hat{\pi},i}(x)g_i(x)dx = F_{\hat{\pi},i}(u)g_i(u) - F_{\hat{\pi},i}(\ell)g_i(\ell) - \int_\ell^u F_{\hat{\pi},i}(x)g'_i(x)dx$. To upper-bound this difference, define *generalized bounding function*

$$\delta_i(\tau) \;:=\; -\left(F_{\hat{\pi},i}(u) - F_{\hat{\pi},i}(\ell)\right)\cdot g_i(\tau). \tag{4.12}$$

Then, to learn $\delta_i(\tau)$, we define

$$\begin{aligned}\Delta_i(\tau) \;&:=\; F_{\hat{\pi},i}(u)(g_i(u) - g_i(\tau)) - F_{\hat{\pi},i}(\ell)(g_i(\ell) - g_i(\tau)) - \int_\ell^u F_{\hat{\pi},i}(x)g'_i(x)dx \\ &=\; F_{\hat{\pi},i}(u)\int_\tau^u (F_i(x) - 1)dx + F_{\hat{\pi},i}(\ell)\int_\ell^\tau (F_i(x) - 1)dx - \int_\ell^u F_{\hat{\pi},i}(x)(F_i(x) - 1)dx.\end{aligned}$$

Observe that $\delta_i(\tau) = \Delta_i(\tau) - (R_u - R_\ell)$, where R_u and R_ℓ correspond to the expected reward in $\hat{\pi}$ with $\tau_i = u$ and $\tau_i = \ell$ respectively. Then, by replacing $F_i(x)$ with $\hat{F}_i(x)$, we can get $\hat{\Delta}_i(\tau)$, which is an estimate of $\Delta_i(\tau)$. For R_u and R_ℓ, we can learn the estimates $\hat{R}_u$ and $\hat{R}_\ell$ via running samples. Combining these estimates results in $\hat{\delta}_i(\tau)$. Then, the following claim shows that $\hat{\delta}_i(\tau)$ estimates $\delta_i(\tau)$ accurately (proved in Section C.3).

Claim 4.7. *In Algorithm 6, if the conditions in Lemma 4.5 holds, then with probability $1 - T^{-12}$ we have $|\hat{\delta}_i(\tau) - \delta_i(\tau)| \le \epsilon$ simultaneously for all $\tau \in [\ell_i, u_i]$.*

Now we prove the statements in Lemma 4.5. In the following proofs, we assume $|\delta_i(\tau) - \hat{\delta}_i(\tau)| \le \epsilon$ holds simultaneously for all $\tau \in [\ell_i, u_i]$.

Statement (i). Look at Algorithm 6: It finds $\hat{\pi}, \ell, u$, gets $\hat{\delta}_i(\tau)$, then calculates $[\ell'_i, u'_i] = \{\tau \in [\ell_i, u_i] : |\hat{\delta}_i(\tau) < \epsilon|\}$. Since $\delta_i(\tau_i^*) = 0$, there must be $|\hat{\delta}_i(\tau_i^*)| \le \epsilon$. Therefore, $\tau_i^* \in [\ell'_i, u'_i]$.

Statement (ii). Notice that $[\ell'_i, u'_i] = \{\tau \in [\ell_i, u_i] : |\hat{\delta}_i(\tau) \le \epsilon|\}$. Therefore, for all $\tau \in [\ell'_i, u'_i]$, $|\delta_i(\tau)| \le |\hat{\delta}_i(\tau)| + |\delta_i(\tau) - \hat{\delta}_i(\tau)| \le 2\epsilon$. We first assume that $\hat{\pi}$ is an *accurate* MoveBound policy. Then, from the definition, we have $F_{\hat{\pi},i}(u) - F_{\hat{\pi},i}(\ell) \ge F_{\pi',i}(u') - F_{\pi',i}(\ell')$ for all valid partial policy π' parameterized by u', ℓ'. Therefore, $|\delta_{\pi',u',\ell',i}(\tau)| \le |\delta_i(\tau)| \le 2\epsilon$.

Statement (iii). We again assume that $\hat{\pi}$ is an accurate MoveBound policy. Recall that we just proved $|\delta_i(\tau)| \le 2\epsilon$. Combining this with (4.12), we have $|g_i(\tau)| \le \frac{2\epsilon}{(F_{\hat{\pi},i}(u) - F_{\hat{\pi},i}(\ell))}$.

Now, consider the policy π'. Assume we first have $\tau_i = u'$ and we want to move it to $\tau_i = \ell'$, satisfying $\ell', u' \in [\ell'_i, u'_i]$ and π' is valid when both $\tau_i = \ell'$ and $\tau_i = u'$. Then, the difference of the expected reward is $\left|\int_{\ell'}^{u'} f_{\pi',i}(x)g_i(x)dx\right|$, and we have the following bound:

$$\left|\int_{\ell'}^{u'} f_{\pi',i}(x)g_i(x)dx\right| \;\le\; |F_{\pi',i}(u') - F_{\pi',i}(\ell')| \max_{v\in[\ell',u']} |g_i(v)| \;\le\; 2\epsilon, \tag{4.13}$$

where in the last inequality we use the fact that $F_{\hat{\pi},i}(u) - F_{\hat{\pi},i}(\ell) \ge |F_{\pi',i}(u') - F_{\pi',i}(\ell')|$ when π is a MoveBound policy, and $|g_i(v)| \le \frac{2\epsilon}{(F_{\pi,i}(u) - F_{\pi,i}(\ell))}$ for all $v \in [\ell'_i, u'_i]$. This gives an upper bound on the difference of the expected reward when we want to move τ_i.

The remaining part is to show how to get a MoveBound policy. However, since we only have CDF estimates $\hat{F}_i(x)$ instead of an accurate $F_i(x)$, there is no hope to get an accurate MoveBound policy. The following Lemma then shows that we can calculate an approximate MoveBound policy:

Lemma 4.8. *There exists an algorithm with time complexity $O(n \cdot 2^n)$ that calculates a* MoveBound *policy with an extra $4\sqrt{\epsilon}$ additive error.*

We leave the details of the algorithm and the proof to Section C.4.

Finally, we show that this $4\sqrt{\epsilon}$ error doesn't hurt too much for both Statement (ii) and (iii). Define

$$q_i \;:=\; \max_{\pi} F_{\pi,i}(u) - F_{\pi,i}(\ell), \quad \text{and} \quad \hat{q}_i \;:=\; F_{\hat{\pi},i}(u) - F_{\hat{\pi},i}(\ell),$$

where $\hat{\pi}$ is the approximate MoveBound policy we get via Lemma 4.8. Then, we have $q_i \le \hat{q}_i + 4\sqrt{\epsilon}$. For Statement (ii), we have

$$\begin{aligned} |\delta_{\pi',u',\ell',i}(\tau)| \;&\le\; q_i \max_{v\in[\ell'_i,u'_i]} |g_i(v)| \;\le\; (\hat{q}_i + 4\sqrt{\epsilon}) \cdot \max_{v\in[\ell'_i,u'_i]} |g_i(v)| \\ &\le\; \hat{q}_i \cdot \frac{\max_{v\in[\ell'_i,u'_i]} |\delta_i(v)|}{\hat{q}_i} + 4\sqrt{\epsilon} \cdot T^{-1/4} \\ &\le\; 2\epsilon + 4\sqrt{\epsilon} \cdot T^{-\frac{1}{4}} \;<\; 3\epsilon. \end{aligned}$$

Here, the second line follows the definition of $\delta_i(v)$ and the precondition in Lemma 4.5. The third line holds because the condition $|\delta_i(v)| \le 2\epsilon$ does not require $\hat{\pi}$ to be accurate, and the last inequality holds when $\epsilon > 16T^{-\frac{1}{2}}$.

For Statement (iii), following (4.13), we can bound the moving difference to

$$\left|\int_{\ell'}^{u'} f_{\pi',i}(x)g_i(x)\right| \;\le\; q_i \max_{v\in[\ell'_i,u'_i]} |g_i(v)|.$$

Therefore, the same 3ϵ bound holds. □

4.3 Step 2: Updating Order Constraints to Bound Swapping Difference In this section, our goal is to verify σ_i and σ_j which one is larger, or claiming that reversing the order of X_i and X_j doesn't hurt too much. We first provide the following lemma, which shows the difference of the expected reward when we swap two distributions with a same threshold:

Lemma 4.9. *For a policy π, such that X_i and X_j are consecutive with $\tau_i = \tau_j = \tau$, let $\Delta_{\pi,i,j}(\tau)$ be the change of the expected reward after swapping X_i and X_j, then*

$$\Delta_{\pi,i,j}(\tau) \;=\; F_{\pi,i}(\tau)(g_i(\tau)(1 - F_j(\tau)) - g_j(\tau)(1 - F_i(\tau))).$$

Proof. Assume we have value v in hand before arriving X_i and X_j. To pass the threshold, there must be $v \le \tau$. If X_i is in the front, the expected gain of opening X_i is $g_i(v)$. After that, if $X_i < \tau$, we can play X_j as well. The expected gain is $F_i(v)g_j(v) + \int_v^\tau f_i(x)g_j(x)dx = F_i(\tau)g_j(\tau) - \int_u^\tau F_i(x)g_j'(x)dx$. Therefore, the total expected gain from X_j and X_i is $g_i(v) + F_i(\tau)g_j(\tau) - \int_u^\tau F_i(x)g_j'(x)dx$. Similarly, if X_j is in the front, the total expected gain from X_i and X_j is $g_j(v) + F_j(\tau)g_i(\tau) - \int_u^\tau F_j(x)g_i'(x)dx$.

Notice that the order of X_i and X_j doesn't affect the expected gain from the distributions behind X_i and X_j. Therefore, the difference of the gain from X_i and X_j is exactly the difference of the expected reward:

$$\begin{aligned}
&\left(g_i(v) + F_i(\tau)g_j(\tau) - \int_u^\tau F_i(x)g_j'(x)dx\right) - \left(g_j(v) + F_j(\tau)g_i(\tau) - \int_u^\tau F_j(x)g_i'(x)dx\right) \\
&= g_i(v) + F_i(\tau)g_j(\tau) - g_j(v) - F_j(\tau)g_i(\tau) + \int_u^\tau \left(F_j(x)(F_i(x)-1) - F_i(x)(F_j(x)-1)\right)dx \\
&= \left(g_i(v) + u - \tau + \int_u^\tau F_i(x)dx\right) - \left(g_j(v) + u - \tau + \int_u^\tau F_j(x)dx\right) + F_i(\tau)g_j(\tau) - F_j(\tau)g_i(\tau) \\
&= g_i(\tau)(1 - F_j(\tau)) - g_j(\tau)(1 - F_i(\tau)).
\end{aligned}$$

Since the probability that v arrives with $v < \tau$ is exactly $F_{\pi,i}(\tau)$, the expected difference is $\Delta_{\pi,i,j}(\tau) = F_{\pi,i}(\tau)(g_i(\tau)(1 - F_j(\tau)) - g_j(\tau)(1 - F_i(\tau)))$. □

Lemma 4.9 shows the following properties:

1. Assume $\sigma_i > \sigma_j$. When $\tau \in [\sigma_j, \sigma_i]$, $\Delta_{\pi,i,j}(\tau) < 0$, i.e., letting X_i be in the front is better. This implies: If we know the sign of $\Delta_{\pi,i,j}(\tau)$, and we are sure that τ is between σ_i and σ_j, then we can determine that σ_i and σ_j which one is greater.
2. Fix i, j, τ, $|\Delta_{\pi,i,j}(\tau)|$ is maximized when $F_{\pi,i}(\tau)$ is maximized.

According to Property 2, we hope to test X_i and X_j with a policy π that maximizes $F_{\pi,i}(\tau)$. If the difference is bounded when $F_{\pi,i}(\tau)$ is maximized, the swapping difference is bounded in all policies. Inspired by this, we give the definition of the SwapTest policy:

Definition 4.10 (SwapTest Policy)**.** *Given (I, S) and $i, j \in [n]$ with $i \neq j$ and $(i,j),(j,i) \notin S$. Assume we have $[\ell_i', u_i'], [\ell_j', u_j'] \in I$. A* SwapTest *policy is a pair of valid policies (π, π'), such that*

- $\tau_i = \tau_j = \max\{\ell_i', \ell_j'\}$.
- *X_i and X_j are adjacent in both π and π', but under different orders, and this is the only difference between π and π'. W.l.o.g, assume X_i is in the front in π, while X_j is in the front in π', i.e., $\pi^{-1}(i) = \pi^{-1}(j) - 1$, and $\pi'^{-1}(j) = \pi'^{-1}(i) - 1$.*
- *The* SwapTest *policy maximizes $F_{\pi,i}(\tau)$ when the first two conditions are satisfied.*

Then, the algorithm for testing X_i and X_j is clear: We find the SwapTest policy for X_i and X_j, run some samples for two policies and see the difference. If the difference is too large, we can verify σ_i and σ_j which one is larger. Otherwise, we can bound the swapping difference. Algorithm 7 gives the details of this idea.

Algorithm 7: SwapTest Algorithm

Input: Distribution indices i and j

Run Algorithm 8 to get SwapTest policy (π, π')
Run $C \cdot \frac{\log T}{n^2\epsilon^2}$ samples with policy π. Let $\hat{R}_{i,j}$ be the average reward.
Run $C \cdot \frac{\log T}{n^2\epsilon^2}$ samples with policy π'. Let $\hat{R}_{j,i}$ be the average reward.
if $|\hat{R}_{i,j} - \hat{R}_{j,i}| > 40n\epsilon$ **then**
    Add constraint (i,j) into S' if $\hat{R}_{i,j} > \hat{R}_{j,i}$, otherwise add constraint (j,i) into S'.
    Update S' according to the transitivity. Update I' according to the new order constraints, i.e., when adding a constraint (a,b), let $u_b' \leftarrow \min\{u_a', u_b'\}$ and $\ell_a' \leftarrow \max\{\ell_a', \ell_b'\}$.

Output: Updated constraint group (I', S')

Before analysing the algorithm, we point out two facts of Algorithm 7:

Algorithm 8: Finding SwapTest Policy

Input: Input: (I', S'), m, i, j

1 Let $\tau = \tau_i = \tau_j = \max\{\ell'_i, \ell'_j\}$

2 Let $T = \{k | (k, i) \in S' \vee (k, j) \in S' \vee \ell'_k > \tau\}$.

3 For $k \in T$, let $\tau_k = u'_k$

4 For $k \in [n] \setminus (\{i, j\} \cup T)$, let $\tau_k = \ell'_k$

5 Let π and π' be two policies that sort the distributions in a decreasing threshold order, and break ties according to S'. The only difference is: X_i is in front of X_j in π, but X_j is in front of X_i in π'.

Output: π and π'

- Algorithm 7 relies on Algorithm 6, i.e., we need to first run Algorithm 6 to get n new confidence intervals, then run Algorithm 7 to update order constraints. This is critical to the regret analysis.
- In the SwapTest algorithm, we only test the swapping difference with $\tau_i = \tau_j = \max\{\ell'_i, \ell'_j\}$, and give the difference bound only with this threshold. This is sufficient for our regret analysis.

Lemma 4.11 (Swapping Difference Bound)**.** *Given (I', S), ϵ, and $i, j \in [n]$ with $i \neq j$ and $(i, j), (j, i) \notin S$, where I' is generated by Algorithm 6. Assume the preconditions in Lemma 4.1 hold. Algorithm 7 runs $O(\frac{\log T}{n^2\epsilon^2})$ samples and achieves one of the following:*

- *Clarify σ_i and σ_j which one is bigger with probability $1 - T^{-12}$, and give a new constraint (i, j) or (j, i).*
- *Make the following claim with probability $1 - T^{-12}$: For every two valid policies of (I', S), satisfying:*
  - *$\tau_i = \tau_j = \max\{\ell'_i, \ell'_j\}$.*
  - *X_i and X_j are consecutive in both policies but in a different order. This is the only difference between two policies.*

  The difference of the expected reward between these two policies is no more than $60n\epsilon$.

Proof. We first prove the theorem assuming Algorithm 8 returns an accurate SwapTest policy (π, π'). According to the definition of SwapTest policy, π and π' maximizes the probability of reaching X_i and X_j when $\tau_i = \tau_j = \max\{\ell'_i, \ell'_j\}$. According to Property 2, for any valid policy, such that X_i and X_j are consecutive with $\tau_i = \tau_j = \max\{\ell'_i, \ell'_j\}$, the swapping difference is no more than the difference between π and π'. Therefore, if we are evident that the difference between π and π' is no more than $60n\epsilon$, we can claim that this upper bounds the swapping difference between X_i and X_j for any other policy. The proof idea is the following: We run multiple samples to estimate $R_{i,j}$ and $R_{j,i}$, where $R_{i,j}$ is the expected reward of π and $R_{j,i}$ is the expected reward of π'. Next, we show that $|R_{i,j} - \hat{R}_{i,j}| \leq 10n\epsilon$ and $|R_{j,i} - \hat{R}_{j,i}| \leq 10n\epsilon$ with probability $1 - T^{-12}$. Then, $|R_{i,j} - R_{j,i}| \leq 60n\epsilon$ when $|\hat{R}_{i,j} - \hat{R}_{j,i}| \leq 40n\epsilon$.

Now, we bound $|R_{i,j} - \hat{R}_{i,j}|$ with Hoeffding's Inequality (Theorem A.1). $\hat{R}_{i,j}$ is an estimate of $R_{i,j}$ by running $N = C \cdot \frac{\log T}{n^2\epsilon^2}$ samples, and the per-round reward is bounded by $[-0.5, 0.5]$. Then, $\mathbf{Pr}\left[|R_{i,j} - \hat{R}_{i,j}| > 10n\epsilon\right] < 2\exp(-2N \cdot 100n^2\epsilon^2/4) = 2T^{-50C}$. Hence, $|R_{i,j} - \hat{R}_{i,j}| \leq 10n\epsilon$ with probability $1 - T^{-13}$ when $C > 10$. Bounding $|R_{j,i} - \hat{R}_{j,i}|$ is identical, and by the union bound, $|R_{i,j} - \hat{R}_{i,j}| \leq 10n\epsilon$ and $|R_{j,i} - \hat{R}_{j,i}| \leq 10n\epsilon$ simultaneously hold with probability $1 - T^{-12}$.

The concentration proof above also shows that when $|\hat{R}_{i,j} - \hat{R}_{j,i}| > 40n\epsilon$, we can claim that w.h.p. $|R_{i,j} - R_{j,i}| > 20n\epsilon$. Next, we show that this is evident to clarify which of σ_i and σ_j is greater. We first introduce a special case to give the intuition: Consider the case that all other confidence intervals are disjoint with $[\ell'_i, u'_i]$ or $[\ell'_j, u'_j]$. W.l.o.g., assume π (X_i in the front) is better than π' (X_j in the front). If $\tau = \max\{\ell'_i, \ell'_j\}$ is between σ_i and σ_j, we can immediately claim that $\sigma_i > \sigma_j$ according to Property 1. If τ doesn't fall between σ_i and σ_j, there must be $\tau < \min\{\sigma_i, \sigma_j\}$. Then, we adjust π and π' by increasing τ_i and τ_j to $\min\{\sigma_i, \sigma_j\}$. According to Lemma 4.5, these operations do not change the expected reward too much: Since we move two thresholds in each policy, the expected reward of π can decrease by at most 6ϵ, and the expected reward of π' can increase by at most 6ϵ. Therefore, if the original π is at least 20ϵ better than π', we can still claim that $\sigma_i > \sigma_j$.

However, this moving process can be invalid in the general case: $\min\{\sigma_i, \sigma_j\}$ might be greater than some thresholds in front of X_i and X_j. To fix this issue, consider the following process:

- Step 1: Increase τ_i and τ_j until reaching τ_k, where X_k is the distribution just in front of X_i and X_j.
- Step 2: Swap X_i and X_j with X_k.
- Repeat Step 1 and 2 until $\tau_i = \tau_j = \min\{\sigma_i, \sigma_j\}$.

Let $\Delta_{\pi,\pi'}$ be the difference between expected values of π and π'. We monitor the change of $\Delta_{\pi,\pi'}$ during these operations. Step 1 can decrease $\Delta_{\pi,\pi'}$ by at most $12\epsilon < 20\epsilon$. Step 2 can increase the absolute value of $\Delta_{\pi,\pi'}$. Since there can be at most n Step 1 and 2, if initially $\Delta_{\pi,\pi'} > 20n\epsilon$, this is sufficient to guarantee that $\Delta_{\pi,\pi'} > 0$ at the end of the process. Then, we are evident to claim $\sigma_i > \sigma_j$.

It remains to show that Algorithm 8 returns a SwapTest policy. Besides, this policy should also guarantee that when we are swapping X_i and X_j with X_k, the policy after doing a swap is still valid. Therefore, we introduce the following lemma:

Lemma 4.12. *Algorithm 8 calculates a* SwapTest *policy. Besides, it has the following property: Let* $\tau = \max\{\ell'_i, \ell'_j\}$ *and* $\tau' = \min\{\sigma_i, \sigma_j\}$*. If* $\tau' > \tau$*, then for all* $k \in [n] \setminus \{i, j\}$*, if* $\tau_k \in [\tau, \tau']$*, there must be* $(k, i) \notin S$ *and* $(k, j) \notin S$*.*

Proof. The first two conditions in Definition 4.10 directly follows Algorithm 8. For the objective condition, observe that no distribution in the set T can be moved behind X_i and X_j. Therefore, the policy calculated by Algorithm 8 minimizes $F_{\pi,i}(\tau)$, which means the third condition holds.

For the additional property, assume there exists k satisfying $\tau_k = u'_k$, $\tau_k < \min\{\sigma_i, \sigma_j\}$. Notice that if $(k, i) \in S'$, there must be $u'_k \geq u'_i \geq \min\{\sigma_i, \sigma_j\}$, which is in contrast to the condition $\tau_k = u'_k < \min\{\sigma_i, \sigma_j\}$. Therefore, $(k, i) \notin S'$. Similarly, $(k, j) \notin S'$. Therefore, the additional property in Lemma 4.12 holds. □

Finally, applying Lemma 4.12 immediately proves Lemma 4.11. □

4.4 Converting our Policy to the Optimal Policy in Polynomial Steps In this section, we show that using $\mathsf{poly}(n)$ number of moves and swaps can convert any valid policy into the optimal policy. Since Lemma 4.5 and Lemma 4.11 already show that the difference of each move and swap is bounded by $O(\mathsf{poly}(n)\epsilon)$, combining these results, we can argue that the per-round loss of a valid policy is bounded by $O(\mathsf{poly}(n)\epsilon)$. Formally, we give the following lemma:

Lemma 4.13. *Given a valid constraint group* (I, S)*. For a valid policy of* (I, S)*, we use a "move" to represent the action that modifies a single threshold, and guarantees that the policy after modifying the threshold is still valid. Besides, we use a "swap" to represent the action that swaps two consecutive distributions with the same threshold. This threshold should be equal to the maximum of the two lower confidence bounds, and the policy after swapping the distributions should still be valid.*

For any valid policy of (I, S)*, it can be converted into the optimal policy using* $2n^2$ *moves and* $2n^2$ *swaps.*

Proof. Let π be the policy that $\tau_i = \ell_i$ for all $i \in [n]$, and the distributions are sorted in a decreasing order of τ. Since for every constraint $(i, j) \in S$, we have $\ell_i \geq \ell_j$, π must be a valid policy.

We can prove Lemma 4.13 by showing the following statement: Starting from the policy π, we can move it to any valid policy π' using n^2 moves and n^2 swaps:

- Step 1: Let $i = \arg\max_i \tau'_i$, where τ'_i is the threshold of X_i in policy π'.
- Step 2: If X_i is not the first distribution in π, move τ_i to $\tau_{\pi_{\pi^{-1}(i)-1}}$, then swap X_i and $X_{\pi_{\pi^{-1}(i)-1}}$.
- Step 3: Do Step 2 until X_i is moved to the first place. Then move τ_i to τ'_i.
- Step 4: Ignore X_i in both π and π', repeat Step 1, 2 and 3 until every distribution is settled.

Each distribution only involves in n swaps and n moves, so the total number of moves and swaps are both bounded by n^2. Then, we need to show the validity of every operation. For each move, we increase τ_i to let it be closer to τ'_i. Since $\tau'_i \in [\ell_i, u_i]$, every move is valid. For each swap, the threshold in the front must reach its lower confidence bound. Besides, every swap happens only when there is no constraint between two distributions, so every swap is valid.

Finally, notice that every operation is bidirected. It means that starting from any valid policy π', we can convert it to the policy π, and then convert it to the optimal policy using $2n^2$ moves and swaps, which finishes the proof. $\square$

4.5 Putting Everything Together In this section, we show how to combine Algorithm 6 and Algorithm 7 to generate a new valid constraint group (I', S'), then proves that this leads to an $\widetilde{O}(\mathsf{poly}(n)\sqrt{T})$ regret algorithm. We first give the one-phase algorithm:

Algorithm 9: Constraint Updating Algorithm for Pandora's Box

Input: $I = \{[\ell_1, u_1], ..., [\ell_n, u_n]\}, S = \{(i, j)\}, \hat{F}_1(x), \dots, \hat{F}_n(x), m$

1 //STEP 1: Calculate new confidence interval for each distribution
2 **for** $i \in [n]$ **do**
3   For $j \in [n]$, construct $\hat{F}_j(x)$ using $10^5 \cdot \frac{n^2 \log T}{\epsilon}$ new i.i.d. samples of X_j
4   Run Algorithm 6 with new CDF estimates to get ℓ'_i and u'_i.
5 //Adjust the confidence intervals to meet constraints in S.
6 **for** $(i, j) \in S$ **do**
7   Let $\ell'_i = \max\{\ell'_i, \ell'_j\}$ and $u'_j = \min\{u'_j, u'_i\}$.
8 Let $I' = \{[\ell'_i, u'_i]\}$ and $S' = S$
9 //Add new constraints for disjoint confidence intervals
10 **for** $(i, j) \notin S'$ **do**
11   **if** $\ell'_i > u'_j$ **then** Add (i, j) into S' ;
12
13 //STEP 2: Calculate new constraints for each distribution pair
14 Let $Q = \{(i, j) | (i, j) \notin S' \wedge (j, i) \notin S'\}$
15 **while** $Q \neq \emptyset$ **do**
16   Choose $(i, j) \in Q$ and remove (i, j) from Q
17   Run Algorithm 7 with input (i, j) and update I' and S'
18   //New constraints may fail some previous tests. Should add them back
19   For every k such that ℓ'_k changes in Algorithm 7, if $\exists k'$ such that $(k, k'), (k', k) \notin S'$, add (k, k') into Q.

Output: (I', S')

We can directly give the following lemma according to the three lemmas above:

Lemma 4.14 (Main Lemma)**.** *Given (I, S) and $\epsilon > 16T^{-\frac{1}{2}}$. Assume the pre-conditions in Lemma 4.5 hold, i.e.,*

- *$|g_j(\tau)| \leq T^{-\frac{1}{4}}$ for all $\tau \in [\ell_j, u_j]$.*
- *(I, S) is valid.*
- *For any valid partial policy π' of (I, S), we fix the order and the other thresholds except τ_j. Assume π' is valid when both $\tau_j = \ell'$ and $\tau_j = \ell'$. Define $\delta_{\pi',u',\ell',j}(\tau) = (F_{\pi',j}(\ell') - F_{\pi',j}(u'))g_i(\tau)$. Then $|\delta_{\pi',u',\ell',j}(\tau)| \leq 6\epsilon$.*
- *CDF estimate $\hat{F}_j(x)$ is constructed via $10^5 \cdot \frac{n^2 \log T}{\epsilon}$ fresh i.i.d. samples of X_j.*

Then, Algorithm 9 runs $O(\frac{n \log T}{\epsilon^2})$ rounds, such that the policy in each round is valid for (I, S) (except Line 3), and output a new constraint group (I', S'), satisfying the following statements with probability $1 - T^{-10}$:

- *(I', S') is valid.*
- *For all $j \in [n]$, for any valid partial policy π' of (I', S'), we fix the order and the other thresholds except τ_j. Assume π' is valid when both $\tau_j = \ell'$ and $\tau_j = \ell'$. Define $\delta_{\pi',u',\ell',i}(\tau) = (F_{\pi',j}(\ell') - F_{\pi',j}(u'))g_i(\tau)$. Then $|\delta_{\pi',u',\ell',i}(\tau)| \leq 3\epsilon$.*
- *For a valid policy of (I', S'), the per-round regret is no more than $126n^3\epsilon$.*

Proof. In this proof, we assume Lemma 4.5 and Lemma 4.11 holds. We use Lemma 4.5 for no more than n

times and Lemma 4.11 for no more than n^2 times. By the union bound[6], our proof fails with probability at most $n \cdot T^{-11} + n^2 \cdot T^{-12} \leq T^{-10}$.

For the validity of (I', S'), the statement $\sigma_i \in [\ell'_i, u'_i]$ follows Lemma 4.5, and the statement $\sigma_i > \sigma_j$ for all $(i, j) \in S'$ follows Lemma 4.11. All other statements hold by definition. Therefore, (I', S') is valid.

For the bound of $|\delta_{\pi', u', \ell', i}(\tau)|$, it's guaranteed directly by Lemma 4.5. Notice that Lemma 4.5 even provides a stronger bound for the constraint group (I'_i, S). Since all possible choices of π', ℓ', u' must be valid for (I'_i, S) when it's valid for (I', S'), this doesn't hurt the statement.

For the per-round regret bound, Lemma 4.5 says that the difference of a move is bounded by 3ϵ. Lemma 4.11 says that the difference of a swap is bounded by $60n\epsilon$. Then, according to Lemma 4.13, we can convert any valid policy to the optimal policy using $2n^2$ moves and swaps. Therefore, the per-round regret is bounded by $126n^3\epsilon$.

Next, we argue that Algorithm 9 runs no more than $O(\frac{n \log T}{\epsilon^2})$ rounds. Note that Algorithm 6 is called n times, and Algorithm 6 uses $O(\frac{\log T}{\epsilon^2})$ rounds in one call. So the number of rounds is $O(\frac{n \log T}{\epsilon^2})$. For Algorithm 7, we might test a distribution pair (X_i, X_j) for multiple times. The reason is the following: When using Lemma 4.13, we need to make sure that the value of the final $\max\{\ell'_i, \ell'_j\}$ is the one that we test. Therefore, if the value of ℓ'_i changes, we need to re-test some distribution pairs (i, j). We can argue that the total number of tests is bounded: When doing an extra test for (i, j), at least one of ℓ'_i or ℓ'_j must change. This can happen only when a new constraint related to i or j is added into S'. There are only $2n$ constraints related to i and j, so we can test (i, j) for at most $4n$ times. Therefore, the total number of calls of Algorithm 7 is no more than $4n^3$, and Algorithm 7 uses $O(\frac{\log T}{n^2 \epsilon^2})$ samples in one call, so the number of samples is bounded by $O(\frac{n \log T}{\epsilon^2})$. Combining the two results finishes the proof. ☐

Now, we are ready to show the total regret bound.

Theorem 4.15. *There exists an $O(n^{4.5}\sqrt{T} \log T)$ regret algorithm for Pandora's Box problem.*

Proof. We run Algorithm 2 and then use Lemma 2.5 to bound the main part of the total regret. To run Algorithm 2, we require the pre-conditions listed in Lemma 4.14 hold. We discuss them separately:

- $|g_j(\tau)| \leq T^{-1/4}$: This is guaranteed by Lemma 4.1.
- (I, S) is valid: For the first phase, the condition $\tau_i^* \in [\ell_i, u_i]$ is guaranteed by Lemma 4.1, and we don't have any initial order constraints between distributions (except those distributions with disjoint confidence intervals). Therefore, (I, S) is valid for the first phase. Starting from the second phase, this is guaranteed by Lemma 4.14.
- $|\delta_{\pi', u', \ell', i}(\tau)| \leq 6\epsilon$: For the first phase, this is true because $|\delta_{\pi', u', \ell', i}(\tau)| \leq |g(\tau)| \leq T^{-\frac{1}{4}}$, and initially we have $\epsilon = O(1)$. Starting from the second phase, this is from Lemma 4.14 regarding the previous phase. Notice that parameter ϵ in the new phase is exactly $\frac{\epsilon}{2}$ in the previous phase. Therefore, there is an extra 2 factor in the condition.
- New CDF estimates: This is guaranteed by Algorithm 9.

Lemma 4.14 implies that after $O(\frac{n^7 \log T}{\epsilon^2})$ rounds, the one-round regret in the new constraint group is bounded by ϵ. Applying Lemma 2.5 with $\alpha = 7$, we have the $O(n^{3.5}\sqrt{T} \log T)$ regret bound.

Besides, there are some extra rounds not covered by Lemma 2.5, including the initialization and the CDF estimates construction (Line 3 in Algorithm 9). For the initialization, Lemma 4.1 runs $O(n\sqrt{T} \log T)$ samples, so the regret is $O(n\sqrt{T} \log T)$. For the CDF estimates construction, let k be the number of phases in the doubling algorithm. Then, the total number of samples is

$$\sum_{i=1}^{k} n \cdot O(\frac{n^2 \log T}{\epsilon_i}) = O(\sqrt{T})$$

Combining three parts of regret, the total regret is $O(n^{3.5}\sqrt{T} \log T)$.

[6]We assume $T > 10n$, otherwise an $O(n)$ regret algorithm is trivial.

Finally, recall that until now we are working on a scaled Pandora's Box problem: We scale down the values and the costs by a factor of $2n$. Therefore, for the original problem, the final regret bound is $O(n^{4.5}\sqrt{T}\log T)$. $\square$

4.6 Making the Algorithm Efficient Currently, the running time of the whole algorithm is exponential in n as just Lemma 4.8 introduces an algorithm with $O(n2^n)$ running time. If we want a polynomial time algorithm, we may need an approximation. The following lemma shows a new regret bound with approximation:

Lemma 4.16. *Assume for every i, we can γ-approximate $\max_{\pi,u,\ell} \hat{F}_{\pi,i}(u) - \hat{F}_{\pi,i}(\ell)$, then there exists an $O(\max\{\gamma n^{4.5}, \gamma^2 n\}\sqrt{T}\log T)$ regret algorithm.*

Proof. In this proof, we first discuss the problem for the scaled Pandora's Box problem, and add the scaled $2n$ factor back at last.

We first see how the γ approximation changes Lemma 4.5. Recall that $q_i = \max_\pi F_{\pi,i}(u) - F_{\pi,i}(\ell)$. We further define $\tilde{q}_i = \max_\pi \hat{F}_{\pi,i}(u) - \hat{F}_{\pi,i}(\ell)$ and $\bar{q}_i = F_{\hat{\pi},i}(u) - F_{\hat{\pi},i}(\ell)$, where $\hat{\pi}$ is the chosen policy that γ-approximates $\max_{\pi,u,\ell} \hat{F}_{\pi,i}(u) - \hat{F}_{\pi,i}(\ell)$. According to Claim 4.6, we have $\tilde{q}_i \ge q_i - 2\sqrt{\epsilon}$ and $\bar{q}_i \ge \frac{\tilde{q}_i}{\gamma} - 2\sqrt{\epsilon}$. So $q_i \le \gamma\bar{q}_i + (2\gamma+2)\sqrt{\epsilon}$. According to (4.13), Statement (ii) and (iii) are both bounded by

$$\begin{aligned} q_i \max_{v\in[\ell_i',u_i']} |g_i(v)| &\le (\gamma\bar{q}_i + (2\gamma+2)\sqrt{\epsilon}) \max_{v\in[\ell_i',u_i']} |g_i(v)| \\ &\le \gamma\bar{q}_i \cdot \frac{2\epsilon}{\bar{q}_i} + (2\gamma+2)\sqrt{\epsilon}\cdot T^{-\frac{1}{4}} \le 3\gamma\epsilon. \end{aligned}$$

For Statement (ii), this changes the bound of $|\delta_{\hat{\pi},u',\ell',i}(\tau)|$ to $O(\gamma\epsilon)$. In our proof, we use this bound when proving Claim 4.7: The bound of $|\delta_{\hat{\pi},u',\ell',i}(\tau)|$ provides a bound for the variance of the $\Delta_i(\tau)$ function, and then we use Bernstein Inequality to show $|\hat{\Delta}_i(\tau) - \Delta_i(\tau)| \le O(\epsilon)$. When the bound changes to $O(\gamma\epsilon)$, to get an $O(\epsilon)$ approximation of $\Delta_i(\tau)$, the number of samples for constructing CDF estimates should be multiplied by γ^2, leading to an $O(\gamma^2\sqrt{T}\log T)$ regret bound.

For Statement (iii), notice that we need to use this moving difference to bound the swapping difference. The main idea of the original proof is: Assume we want to test X_i and X_j. After $O(n)$ moves, we can adjust τ_i and τ_j to $\min\{\sigma_i, \sigma_j\}$, then bound the swapping difference by $O(n)\cdot O(\epsilon)$. Since there is an extra γ factor in the new moving difference bound, the new swapping difference should be $O(\gamma n\epsilon)$.

Next, Lemma 4.13 shows that we need $2n^2$ move operations and swap operations to convert a policy to the optimal one, so the new regret bound after $O(\frac{n\log T}{\epsilon^2})$ samples is $O(\gamma n^3\epsilon)$. Then, the parameter α in Lemma 2.5 changes to $\gamma^2 n^7$, so the total regret from the doubling algorithm is $O(\gamma n^{3.5}\sqrt{T}\log T)$.

Finally, after combining these two new regret bounds and adding the scaled $2n$ factor back to the regret bound, we get the $O(\max\{\gamma n^{4.5}, \gamma^2 n^2\}\sqrt{T}\log T)$ final regret bound. $\square$

Lemma 4.16 shows that: If we can get a $\mathsf{poly}(n)$ approximation for the MoveBound policy in polynomial time, we can still get an $O(\mathsf{poly}(n)\sqrt{T})$ regret algorithm. To achieve this goal, we introduce the following sub-routine:

Definition 4.17 (sub-routine)**.** *Let Problem A be the following: Given n and real numbers $a_1, ..., a_n$, $b_1, ..., b_n$, satisfying $0 \le a_i \le b_i \le 1$ for all $i \in [n]$. The objective of Problem A is to calculate*

$$\max_{B\in[n]} \prod_{i\in B} b_i - \prod_{i\in B} a_i.$$

under a set of constraints $\{(i,j)\}$, where a constraint (i,j) means that if we have $i \in B$, there must be $j \in B$.

Lemma 4.18. *If there exists an algorithm that calculates an γ-approximation for Problem A, then there exists an algorithm that γ-approximates $\max_\pi \hat{F}_{\pi,i}(u) - \hat{F}_{\pi,i}(\ell)$. If the running time of the algorithm for approximating Problem A is polynomial, then the algorithm for approximating $\hat{F}_{\pi,i}(u) - \hat{F}_{\pi,i}(\ell)$ is also polynomial.*

Proof. Consider calculating a MoveBound policy for X_i. Assume that we know the value of ℓ and u. Then, we only need to pick a subset $B \subseteq [n]\setminus\{i\}$ to maximize $\prod_{j\in B} b_j - \prod_{j\in N} a_j$, where $b_j = \hat{F}_j(u)$, and $a_j = \hat{F}_j(\ell)$.

However, not all subsets B are valid. Firstly, for $j \in B$, there must be $\tau_j \geq u$, which means $u_j \geq u$ is required. Similarly, we should also guarantee that $\ell_j \leq \ell$ for all $j \in [n] \setminus B$. Besides, if there is an order constraint (j,k), then $k \in B$ implies $j \in B$, which can be represented as a constraint in Problem A. If all constraints are satisfied, policy $\tau_j = u_j$ for $j \in B$ and $\tau_j = \ell_j$ for $j \notin B \cup \{i\}$ is a feasible policy. Therefore, finding the optimal policy with fixed ℓ and u is captured by Problem A. So, an γ-approximation algorithm for Problem A also γ-approximates $\hat{F}_{\pi,i}(u) - \hat{F}_{\pi,i}(\ell)$.

Notice that when maximizing $\hat{F}_{\pi,i}(u) - \hat{F}_{\pi,i}(\ell)$, we want to push the thresholds to the boundaries to give τ_i enough space. Therefore, the value of ℓ and u must be equal to some ℓ_j or u_j, which means that there are only $O(n^2)$ candidates. Therefore, if the algorithm that γ- approximates Problem A runs in polynomial time, the running time of the algorithm for approximating $\hat{F}_{\pi,i}(u) - \hat{F}_{\pi,i}(\ell)$ is also polynomial. ☐

It remains to give a $\mathsf{poly}(n)$-approximation algorithm for Problem A, with $O(\mathsf{poly}(n))$ running time. The following theorem shows that this is possible:

Lemma 4.19. *Given an instance of Problem A. Let B_j be the subset with the smallest size that contains j. Let $q_j = \prod_{i\in B_j} b_i - \prod_{i \in B_j} a_i$ Then, $\max_j q_j$ is an n-approximation of problem A.*

Proof. Construct a graph $G = (V, E)$, such that $V = [n]$, and E is the set of all constraints, i.e., a constraint (u, v) is represented as a directed edge $(u, v) \in E$. Then, B_j is the set of all vertices which is reachable from j.

Notice that when G contains a connected component, we can shrink the component into one single vertex, because picking any single vertex in the connected component means picking the whole component. Therefore, we only need to prove the theorem when G is a directed acyclic graph (DAG).

Re-index the vertices in G, to make sure that for every edge $(u, v) \in E$, there must be $u > v$. Besides, make sure that $B^* = \{1, ..., k\}$ is exactly the optimal set of Problem A. Then,

$$
\begin{aligned}
\prod_{i\in[k]} b_i - \prod_{i\in[k]} a_i &= \sum_{j\in[k]} \left(\prod_{i=1}^{j-1} b_i \cdot (b_j - a_j) \cdot \prod_{i=j+1}^{k} a_i \right) \\
&\leq \sum_{j\in[k]} \left(\prod_{i\in B_j \setminus \{j\}} b_i \cdot (b_j - a_j) \cdot 1 \right) \\
&\leq \sum_{j\in[k]} \left(\prod_{i\in B_j} b_i - \prod_{i\in T_j} a_i \right) \quad \leq \quad \sum_{j\in[n]} q_j,
\end{aligned}
$$

where the second-last inequality uses $a_i \leq b_i$. Therefore, $\max_j q_j$ is an n-approximation of $\prod_{i\in B^*} b_i - \prod_{i\in B^*} a_i$. ☐

Finally, combining Lemma 4.16, Lemma 4.18, and Lemma 4.19 gives the following main theorem:

Theorem 1.3. *There is a polytime algorithm with $O(n^{5.5}\sqrt{T}\log T)$ regret for the Bandit Pandora's Box problem where we only receive utility (selected value minus total cost) as feedback.*

5 Lower Bounds

In this section we prove lower bounds for Online Learning Prophet Inequality and Online Learning Pandora's Box. Our lower bounds will hold even against full-feedback.

5.1 $\Omega(\sqrt{T})$ Lower Bound for Stochastic Input We show an $\Omega(\sqrt{T})$ regret lower bound for Bandit Prophet Inequality and an $\Omega(\sqrt{nT})$ lower bound for Pandora's Box problem, which implies that the $\sqrt{T}$ factor in our regret bounds is tight. We first give the lower bound for Prophet Inequality.

Theorem 5.1. *For Bandit Prophet Inequality there exists an instance with $n = 2$ such that all online algorithms incur $\Omega(\sqrt{T})$ regret.*

Proof. Let $\mathcal{D}_1$ be a distribution that always gives $\frac{1}{2}$. Let $\mathcal{D}_2$ be a Bernoulli distribution. The probability of $X_2 = 1$ might be $\frac{1}{2} + \frac{1}{\sqrt{T}}$ or $\frac{1}{2} - \frac{1}{\sqrt{T}}$. Both settings appear w.p. $\frac{1}{2}$. The online algorithm doesn't know which is the real setting. If it chooses not to open X_2, it will lose $\sqrt{T}$ w.p. $\frac{1}{2}$. Otherwise, because of the variance, the algorithm needs $\Omega(T)$ samples from X_2 to learn the real setting, and loses $\frac{1}{2} \cdot \sqrt{T}$ for each round it runs. In both cases, the online algorithm should lose $\Omega(\sqrt{T})$, which finishes the proof. □

For the Pandora's Box problem, [GHTZ21] already shows a lower bound for the sample complexity of Pandora's Box problem, which directly implies a lower bound for the online learning setting.

Theorem 5.2 ([GHTZ21])**.** *For any instance of Pandora's problem in which the rewards are bounded in* $[0, 1]$*, running* $\Omega(\frac{n}{\epsilon^2})$ *samples is necessary to get an* ϵ*-additive algorithm.*

Corollary 5.3. *For Pandora's Box problem, all online algorithms incur* $\Omega(\sqrt{nT})$ *regret.*

Proof. Assume there exists an online algorithm that achieves $o(\sqrt{nT})$ regret. This implies that after T rounds, we can achieve $o(\frac{n}{\sqrt{T}})$ per-round regret, which is in contradiction with Theorem 5.2. □

We remark that [GHTZ21] claims that $\Omega(\frac{n}{\epsilon^2})$ samples are necessary to get an ϵ-additive algorithm for Prophet Inequality but without giving a proof. However, this claim seems incorrect since in an ongoing work we show an $\widetilde{O}(\sqrt{T})$ regret algorithm for Prophet Inequality with full-feedback.

5.2 $\Omega(T)$ Lower Bound for Adversarial Input In this paper, we study Bandit Prophet Inequality and Bandit Pandora's Box problems under the stochastic assumption that input is drawn from unknown-but-fixed distributions. A natural extension would be: can we obtain $o(T)$ regret for adversarial inputs where the the input distribution may change in each time step? The following theorems shows that sub-linear regret is impossible even for oblivious adversarial inputs with $n = 2$ under full-feedback.

Theorem 5.4. *For Bandit Prophet Inequality with oblivious adversarial inputs, there exists an instance with* $n = 2$ *such that the optimal fixed-threshold strategy has total value* $\frac{3}{4}T$ *but no online algorithm (even under full-feedback) can obtain total value more than* $\frac{1}{2}T$*.*

Proof. We first introduce a notation used in this proof. Let s be a 01-string. Define $Bin(s)$ to be the binary decimal corresponding to s. For example, $Bin(1) = (0.1)_2 = \frac{1}{2}$, $Bin(0011) = (0.0011)_2 = \frac{3}{16}$.

Now, we introduce the main idea of the counter example: At the beginning, the adversary will choose a T-bits code $s = s_1 s_2 ... s_T$ uniformly at random (i.e., s_i is set to be 0 or 1 w.p. $\frac{1}{2}$ independently). The value of X_1 is $\frac{1}{2}$ plus a small bias that contains the information of the code. The value of X_2 is either 1 or 0, which is decided by the code. Formally, in the i-th round:

- $X_1 = \frac{1}{2} + \epsilon \cdot v_i$, where ϵ is an arbitrarily small constant that doesn't effect the reward, and v_i is a value between $Bin(s_1 s_2 ... s_{i-1} + 0 + 1^{T-i})$ and $Bin(s_1 s_2 ... s_{i-1} + 1 + 0^{T-i})$. The notation 0^k represents a length-k string with all 0s, and 1^k represents a length-k string with all 1s.
- $X_2 = 1$ if $s_i = 0$, otherwise $X_2 = 1$.

For an online algorithm, it only knows that the next s_i can be 0 or 1 w.p. $\frac{1}{2}$. Therefore, no matter it switches to the next box or not, it can only get $\frac{1}{2}$ in expectation. So the maximum total reward it can achieve is $\frac{1}{2}T$.

However, if we know the code, playing $\tau = \frac{1}{2} + \epsilon \cdot Bin(s)$ gets $\frac{3}{4}T$: $X_2 = 1$ when $X_1 < \tau$, while $X_2 = 0$ when $X_1 \geq \tau$. Therefore, playing τ allows us to pick every 1, but stays in $X_1 = \frac{1}{2}$ when $X_2 = 0$. Since we generate the code uniformly at random, X_2 is 1 w.p. $\frac{1}{2}$. Therefore, the expected reward is $T \cdot (\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \frac{1}{2}) = \frac{3}{4}T$. □

Next, we use a similar proof idea to prove lower bound for Pandora's Box. This resolves an open question of [Ger22, GT22] on whether sublinear regrets are possible for Online Learning of Pandora's Box with adversarial inputs.

Theorem 5.5. *For Bandit Pandora's Box with oblivious adversarial inputs, there exists an instance with* $n = 2$ *such that the optimal fixed-threshold strategy has total utility* $\frac{1}{4}T$ *but no online algorithm (even under full-feedback) can obtain total utility more than* 0*.*

Proof. At the beginning, the adversary will choose a T-bits code $s = s_1 s_2 ... s_T$ uniformly at random (s_i is set to be 0 or 1 w.p. $\frac{1}{2}$ independently). The cost c_1 is 0, and the value of X_1 is 0 plus a small bias that contains the information of the code. The cost c_2 is $\frac{1}{2}$, and the value of X_2 is either 1 or 0, which is decided by the code. Formally, in the i-th round:

- $X_1 = 0 + \epsilon \cdot v_i$, where ϵ is an arbitrarily small constant that doesn't effect the reward, and v_i is a value between $Bin(s_1 s_2 ... s_{i-1} + 0 + 1^{T-i})$ and $Bin(s_1 s_2 ... s_{i-1} + 1 + 0^{T-i})$. The notation 0^k represents a length-k string with all 0s, and 1^k represents a length-k string with all 1s.
- $X_2 = 1$ if $s_i = 0$, otherwise $X_2 = 1$.

The cost of X_1 is 0, so we can always first open X_1. Then, for an online algorithm, it doesn't know whether X_2 is 1 or 0. No matter it opens X_2 or not, the expected reward will only be 0.

However, when we know the code, playing $\tau = \epsilon \cdot Bin(s)$ gets $\frac{1}{4}T$, because it will open X_2 whenever $X_2 = 1$, and skip it when X_2 is 0. Since we generate the code uniformly at random, X_2 is 1 w.p. $\frac{1}{2}$. Therefore, the expected reward is $T \cdot (\frac{1}{2} \cdot (1 - \frac{1}{2})) = \frac{1}{4}T$. □

References

[ACBF02] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

[ACG+22] Alexia Atsidakou, Constantine Caramanis, Evangelia Gergatsouli, Orestis Papadigenopoulos, and Christos Tzamos. Contextual pandora's box. *arXiv preprint arXiv:2205.13114*, 2022.

[AHK12] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of computing*, 8(1):121–164, 2012.

[AKW14] Pablo Daniel Azar, Robert Kleinberg, and S. Matthew Weinberg. Prophet inequalities with limited information. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*, pages 1358–1377, 2014.

[ANSS19] Nima Anari, Rad Niazadeh, Amin Saberi, and Ali Shameli. Nearly optimal pricing algorithms for production constrained and laminar bayesian selection. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC*, pages 91–92, 2019.

[BC12] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[BDL22] Mark Braverman, Mahsa Derakhshan, and Antonio Molina Lovett. Max-weight online stochastic matching: Improved approximations against the online benchmark. In *ACM Conference on Economics and Computation, EC*, 2022.

[Bel57] Richard Bellman. *Dynamic programming*. Princeton University Press, 1957.

[CBL06] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[CDF+22] Constantine Caramanis, Paul Dütting, Matthew Faw, Federico Fusco, Philip Lazos, Stefano Leonardi, Orestis Papadigenopoulos, Emmanouil Pountourakis, and Rebecca Reiffenhäuser. Single-sample prophet inequalities via greedy-ordered selection. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1298–1325, 2022.

[CDFS19] José R. Correa, Paul Dütting, Felix A. Fischer, and Kevin Schewior. Prophet inequalities for I.I.D. random variables from an unknown distribution. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC*, pages 3–17, 2019.

[CHMS10] Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC*, pages 311–320, 2010.

[EFGT20] Tomer Ezra, Michal Feldman, Nick Gravin, and Zhihao Gavin Tang. Online stochastic max-weight matching: Prophet inequality for vertex and edge arrival models. In *The 21st ACM Conference on Economics and Computation, EC*, pages 769–787, 2020.

[EHLM19] Hossein Esfandiari, Mohammad Taghi Hajiaghayi, Brendan Lucier, and Michael Mitzenmacher. Online pandora's boxes and bandits. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI*, pages 1885–1892, 2019.

[FGL15] Michal Feldman, Nick Gravin, and Brendan Lucier. Combinatorial auctions via posted prices. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 123–135, 2015.

[FL20] Hu Fu and Tao Lin. Learning utilities and equilibria in non-truthful auctions. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*, 2020.

[FSZ16] Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1014–1033, 2016.

[FTW+21] Hu Fu, Zhihao Gavin Tang, Hongxun Wu, Jinzhao Wu, and Qianfan Zhang. Random order vertex arrival contention resolution schemes for matching, with applications. In *48th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 68:1–68:20, 2021.

[Ger22] Evangelia Gergatsouli. Personal communication. 2022.

[GHTZ21] Chenghao Guo, Zhiyi Huang, Zhihao Gavin Tang, and Xinzhi Zhang. Generalizing complex hypotheses on product distributions: Auctions, prophet inequalities, and pandora's problem. In *Conference on Learning Theory, COLT*, pages 2248–2288, 2021.

[GJSS19] Anupam Gupta, Haotian Jiang, Ziv Scully, and Sahil Singla. The markovian price of information. In *Proceedings of Integer Programming and Combinatorial Optimization, IPCO*, volume 11480, pages 233–246, 2019.

[GKS19] Buddhima Gamlath, Sagar Kale, and Ola Svensson. Beating greedy for stochastic bipartite matching. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 2841–2854, 2019.

[GT22] Evangelia Gergatsouli and Christos Tzamos. Online learning for min sum set cover and pandora's box. In *International Conference on Machine Learning, ICML*, pages 7382–7403, 2022.

[Har22] Jason D Hartline. *Mechanism design and approximation*. Book draft., 2022.

[Haz16] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[HKS07] Mohammad Taghi Hajiaghayi, Robert D. Kleinberg, and Tuomas Sandholm. Automated online mechanism design and prophet inequalities. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 58–65, 2007.

[KS77] Ulrich Krengel and Louis Sucheston. Semiamarts and finite values. *Bull. Am. Math. Soc*, 1977.

[KS78] Ulrich Krengel and Louis Sucheston. On semiamarts, amarts, and processes with finite value. *Advances in Prob*, 4:197–266, 1978.

[KW12] Robert Kleinberg and S. Matthew Weinberg. Matroid prophet inequalities. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC*, pages 123–136, 2012.

[KWW16] Robert D. Kleinberg, Bo Waggoner, and E. Glen Weyl. Descending price optimally coordinates search. In *Proceedings of the ACM Conference on Economics and Computation, EC*, pages 23–24, 2016.

[LLP+21] Allen Liu, Renato Paes Leme, Martin Pál, Jon Schneider, and Balasubramanian Sivan. Variable decomposition for prophet inequalities and optimal ordering. In *The 22nd ACM Conference on Economics and Computation, EC*, page 692, 2021.

[LS20] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[LSTW23] Renato Paes Leme, Balasubramanian Sivan, Yifeng Teng, and Pratik Worah. Pricing query complexity of revenue maximization. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 399–415, 2023.

[Luc17] Brendan Lucier. An economic view of prophet inequalities. *SIGecom Exch.*, 16(1):24–47, 2017.

[PPSW21] Christos H. Papadimitriou, Tristan Pollner, Amin Saberi, and David Wajc. Online stochastic max-weight bipartite matching: Beyond prophet inequalities. In *The 22nd ACM Conference on Economics and Computation, EC*, pages 763–764, 2021.

[Rou16] Tim Roughgarden. *Twenty lectures on algorithmic game theory*. Cambridge University Press, 2016.

[RS17] Aviad Rubinstein and Sahil Singla. Combinatorial prophet inequalities. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1671–1687, 2017.

[Rub16] Aviad Rubinstein. Beyond matroids: secretary problem and prophet inequality with general constraints. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 324–332, 2016.

[RWW20] Aviad Rubinstein, Jack Z. Wang, and S. Matthew Weinberg. Optimal single-choice prophet inequalities from samples. In *11th Innovations in Theoretical Computer Science Conference, ITCS*, pages 60:1–60:10, 2020.

[SC84] Ester Samuel-Cahn. Comparison of threshold stop rules and maximum for independent nonnegative random variables. *the Annals of Probability*, pages 1213–1216, 1984.

[Sin18a] Sahil Singla. Combinatorial optimization under uncertainty: Probing and stopping-time algorithms. *Unpublished doctoral dissertation, Carnegie Mellon University*, 2018.

[Sin18b] Sahil Singla. The price of information in combinatorial optimization. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2018.

[Sli19] Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning*, 12(1-2):1–286, 2019.

[SS21] Danny Segev and Sahil Singla. Efficient approximation schemes for stochastic probing and prophet problems. In

The 22nd ACM Conference on Economics and Computation, EC, pages 793–794, 2021.
[Wei79] Martin L. Weitzman. Optimal search for the best alternative. *Econometrica: Journal of the Econometric Society*, pages 641–654, 1979.

A Basic Probabilistic Inequalities

Theorem A.1 (Hoeffding's Inequality)**.** *Let $X_1, \ldots, X_N$ be independent random variables such that $a_i \le X_i \le b_i$. Let $S_N = \sum_{i\in[N]} X_i$. Then for all $t > 0$, we have $\mathbf{Pr}\left[|S_N - \mathbf{E}[S_n]| \ge t\right] \le 2\exp\big(-\frac{2t^2}{\sum_{i\in[n]}(b_i-a_i)^2}\big)$. This implies, that if X_i are i.i.d. samples of random variable X, and $a = a_i, b = b_i$ for all $i \in [N]$, let $\hat{X} := \frac{1}{N}\sum_{i\in[N]} X_i$, then for every $\varepsilon > 0$,*

$$\mathbf{Pr}\left[|\hat{X} - \mathbf{E}[X]| \ge \varepsilon\right] \le 2\exp\left(-\frac{2N\varepsilon^2}{(b-a)^2}\right).$$

Theorem A.2 (Bernstein Inequality)**.** *Given mean zero random variables $\{X_i\}_{i=1}^N$ with $\mathbb{P}(|X_i| \le c) = 1$ and $\mathsf{Var}X_i \le \sigma_i^2$. If $\bar{X}_N$ denotes their average and $\sigma^2 = \frac{1}{N}\sum_{i=1}^n \sigma_i^2$, then*

$$\mathbb{P}(|\bar{X}_N| \ge \varepsilon) \;\le\; 2\exp\big(-\frac{N\varepsilon^2}{2\sigma^2 + 2c\varepsilon/3}\big).$$

Theorem A.3 (DKW Inequality)**.** *Given a natural number N, let $X_1, \ldots, X_n$ be i.i.d. samples with cumulative distribution function $F(\cdot)$. Let $\hat{F}(\cdot)$ be the associated empirical distribution function $\hat{F}(x) := \frac{1}{N}\sum_{i\in[N]} \mathbf{1}_{X_i \le x}$. Then, for every $\varepsilon > 0$, we have*

$$\mathbf{Pr}\left[\sup_x |\hat{F}(x) - F(x)| > \varepsilon\right] \le 2\exp(-2N\varepsilon^2).$$

B Missing Proofs from Section 2

B.1 Proof of Lemma 2.5

Proof. There are two different sources of regret. We bound them separately.

Loss 1 from the while loop: The main idea of the proof is to use the regret bound from the previous phase to bound the total regret in the next phase. Specifically, assume $\epsilon_0 = O(1)$ be the maximum possible one-round regret, and assume there are k phases in the while loop. Then the total regret can be bounded by

$$\text{(B.1)} \qquad \sum_{i=1}^{k} O\left(\frac{n^\alpha \log T}{\epsilon_i^2}\right) \cdot \epsilon_{i-1} \;=\; \sum_{i=1}^{k} O\left(\frac{n^\alpha \log T}{\epsilon_i}\right) \;=\; O(n^{\alpha/2}\sqrt{T}).$$

Therefore, the total regret from the while loop is bounded by $O(n^{\alpha/2}\sqrt{T})$.

Loss 2 after the while loop: After the while loop, the one-round regret is bounded by $\epsilon_k = \frac{n^{\alpha/2}\log T}{\sqrt{T}}$, so the total regret can be bounded by $O(\frac{n^{\alpha/2}\log T}{\sqrt{T}}) \cdot T = O(n^{\alpha/2}\sqrt{T}\log T)$.

Finally, combining the two sources of regret proves the theorem.

Besides, we should also verify that Algorithm 2 succeeds with probability $1 - T^{-9}$, and it runs no more than $O(T)$ rounds. For the success probability, Algorithm 2 runs $k = O(\log T) < T$ rounds, and the subroutine Alg succeeds with probability $1 - T^{-10}$. By the union bound, Algorithm 2 succeeds with probability $1 - T^{-9}$. As for the number of rounds, in the while loop, Algorithm 2 runs

$$\sum_{i=1}^{k} \frac{n^\alpha \log T}{\epsilon_i^2} \;\le\; \frac{4n^\alpha \log T}{\epsilon_k^2} \;=\; O(T)$$

number of rounds. Therefore, Algorithm 2 is a valid algorithm with respect to time horizon T. □

B.2 Missing Details of Pandora's Box Algorithm for $n = 2$

B.2.1 Proof of Lemma 2.7 To prove Lemma 2.7, we need two claims. The first claim says that when we have a good guess τ with a small $\delta(\tau)$, the loss of playing τ is bounded:

Claim 2.8. *If $\tau, \tau^* \in [\ell, u]$ then $R(\tau^*) - R(\tau) \leq |\delta(\tau)|$.*

Proof. We first upper-bound $R(\tau^*) - R(\tau)$. The two settings are different only when X_1 is between τ and τ^*: Playing τ will loss an extra $|g(X_1)|$. Since $g(x)$ is monotone, we can bound $|g(X_1)|$ by $|g(\tau)|$. Therefore, the extra loss of playing τ is no more than $|F_1(\tau^*) - F_1(\tau)||g(\tau)|$.

On the other hand,

$$|\delta(\tau)| = (F_1(u) - F_1(\ell)) \left| \int_{\tau}^{\tau^*} g'(x)dx \right| = (F_1(u) - F_1(\ell)) \cdot |g(\tau)|.$$

When $\tau, \tau^* \in [\ell, u]$, $F_1(u) - F_1(\ell) \geq |F_1(\tau^*) - F_1(\tau)|$. Therefore, $|\delta(\tau)| \geq R(\tau^*) - R(\tau)$. ☐

The second claim shows that we can get a good estimate for function $\delta(\tau)$:

Claim 2.9. *In Algorithm 3, if the conditions in Lemma 2.7 hold, then with probability $1 - T^{-10}$ $|\hat{\delta}(\tau) - \delta(\tau)| \leq 4\epsilon$ simultaneously for all $\tau \in [\ell, u]$.*

Proof. Recall that $\delta(\tau) = \Delta(\tau) - (R(u) - R(\ell))$. We will give the bound for $|\Delta(\tau^*) - \Delta(\tau)|$, $|R(u) - \hat{R}(u)|$ and $|R(\ell) - \hat{R}(\ell)|$ separately.

For $|\Delta(\tau^*) - \Delta(\tau)|$, we first bound the magnitude of $\Delta(\tau)$:

$$\Delta(\tau) = \int_{\tau}^{u} (F_1(u) - F_1(x))(F_2(x) - 1)dx - \int_{\ell}^{\tau} (F_1(x) - F_1(\ell))(F_2(x) - 1)dx,$$

which implies

$$|\Delta(\tau)| \leq (F_1(u) - F_1(\tau)) \int_{\tau}^{u} (1 - F_2(x))dx + (F_1(\tau) - F_1(\ell)) \int_{\ell}^{\tau} (1 - F_2(x))dx \tag{B.2}$$

$$= (F_1(u) - F_1(\tau))(g(\tau) - g(u)) + (F_1(\tau) - F_1(\ell))(g(\ell) - g(\tau)) \tag{B.3}$$

$$\leq (F_1(u) - F_1(\ell))(g(\ell) - g(u)) \tag{B.4}$$

$$\leq |\delta(u)| + |\delta(\ell)| \ \leq \ 32\epsilon, \tag{B.5}$$

where the last equality follows from the bound $|\delta(\tau)| \leq 16\epsilon$ for all $\tau \in [\ell, u]$ in Lemma 2.7.

Now notice that the estimate $\hat{\Delta}(\tau)$ we have based on our initial estimates $\hat{F}_1$ and $\hat{F}_2$ is unbiased i.e. $\mathbf{E}\left[\hat{\Delta}(\tau)\right] = \Delta(\tau) \leq 32\epsilon$. This simply follows from exchanging interval integration and expectation combined with the independence of X_1 and X_2:

$$\begin{aligned}\mathbf{E}\left[\int_{\tau}^{u} (\hat{F}_1(u) - \hat{F}_1(x))(\hat{F}_2(x) - 1)dx\right] &= \int_{\tau}^{u} (\mathbf{E}\left[\hat{F}_1(u)\right] - \mathbf{E}\left[\hat{F}_1(x)\right])(\mathbf{E}\left[\hat{F}_2(x)\right] - 1)dx \\ &= \int_{\tau}^{u} (F_1(u) - F_1(x))(F_2(x) - 1)dx. \end{aligned} \tag{B.6}$$

Now let us define $\hat{\Delta}(\tau)$ per sample i for each initial sample. We run $N = C \cdot \frac{\log T}{\epsilon}$ samples for $C = 1000$. Then for $i \in [N]$, we define

$$\hat{\Delta}^{(k)}(\tau) = \int_{\tau}^{u} (\hat{F}_1^{(k)}(u) - \hat{F}_1^{(k)}(x))(\hat{F}_2^{(k)}(x) - 1)dx - \int_{\ell}^{\tau} (\hat{F}_1^{(k)}(x) - \hat{F}_1^{(k)}(\ell))(\hat{F}_2^{(k)}(x) - 1)dx,$$

where $\hat{F}_1^{(k)}(.)$ and $\hat{F}_2^{(k)}(.)$ are simple threshold functions at the ith initial sample, which are estimates for the densities F_1 and F_2 respectively. Note that

$$\hat{\Delta}(\tau) = \frac{1}{N} \sum_{k \in [N]} \hat{\Delta}^{(k)}(\tau).$$

Now again similar to (B.6) we have

$$\mathbf{E}\left[\Delta^{(k)}(\tau)\right] \;=\; \mathbf{E}\left[\Delta(\tau)\right] \;\leq\; 32\epsilon. \tag{B.7}$$

Moreover, note that the random variable $\hat{\Delta}^{(i)}(\tau)$ is bounded by one since

$$\begin{aligned} |\hat{\Delta}^{(k)}(\tau)| &\leq \int_\tau^u \left|(\hat{F}_1^{(k)}(u) - \hat{F}_1^{(k)}(x))(\hat{F}_2^{(k)}(x) - 1)\right| dx - \int_\ell^\tau \left|(\hat{F}_1^{(k)}(x) + \hat{F}_1^{(k)}(\ell))(\hat{F}_2^{(k)}(x) - 1)\right| dx \\ &\leq \int_\tau^u 1 dx + \int_\ell^\tau 1 dx \;=\; u - \ell \;\leq\; 1. \end{aligned} \tag{B.8}$$

Combining Equations (B.7) and (B.8), we have the variance bound:

$$\mathsf{Var}[\hat{\Delta}(\tau)] \;\leq\; \mathbf{E}\left[\hat{\Delta}(\tau)^2\right] \;\leq\; \mathbf{E}\left[\hat{\Delta}(\tau)\right] \;\leq\; 32\epsilon. \tag{B.9}$$

Now, combining (B.8) and (B.9), we can apply Bernstein inequality for the random variables $\hat{\Delta}^{(i)}(\tau)$. We have:

$$\mathbf{Pr}\left[|\hat{\Delta}(\tau) - \Delta(\tau)| \geq \epsilon\right] \leq 2\exp\left(-\frac{N\epsilon^2}{2\mathsf{Var}[\hat{\Delta}(\tau)] + \frac{2}{3}\epsilon}\right) = 2T^{-\frac{3C}{194}}. \tag{B.10}$$

Therefore, $|\hat{\Delta}(\tau) - \Delta(\tau)| < \epsilon$ holds with probability $1 - T^{-12}$ when $C = 1000$.

Notice that we only prove the bound for a single τ. To strengthen this concentration bound to hold simultaneously for all τ and $[\ell, u]$, we take a union over appropriate cover sets. In particular, consider $\mathcal{C}$ as a discretization of the interval $[\ell, u]$ with accuracy $1/T$. To be able to exploit the high probability argument for the elements inside the cover for the ones outside, we need to show that Δ is Lipschitz with respect to τ, u and ℓ.

For Δ function, we have $|\Delta(\tau) - \Delta(\tau')| \leq 2|\tau - \tau'|$ since

$$\begin{aligned} \left|\Delta(\tau) - \Delta(\tau')\right| &= \left|\int_\tau^{\tau'} (F_1(u) - F_1(x))(F_2(x) - 1)dx\right| + \left|\int_\tau^{\tau'} (F_1(u) - F_1(x))(F_2(x) - 1)dx\right| \\ &\leq 2|\tau - \tau'|. \end{aligned}$$

It is easy to see that the same Lipschitz bound also holds for $\hat{\Delta}$.

Now for an arbitrary $\tau' \in [\ell, u]$, if we consider the closest τ to it in $\mathcal{C}$, we have $|\tau' - \tau| \leq \frac{1}{T}$. Then, using the Lipschitz constant of Δ and $\hat{\Delta}$:

$$\left|\Delta(\tau) - \Delta(\tau')\right| \;\leq\; \frac{2}{T} \qquad \text{and} \qquad \left|\hat{\Delta}(\tau) - \hat{\Delta}(\tau')\right| \leq \frac{2}{T}. \tag{B.11}$$

Now we apply a union bound over the events $|\hat{\Delta}(\tau) - \Delta(\tau)| < \epsilon$ for all $\tau \in \mathcal{C}$. Since running over all possibilities of $|C| \leq T$, after taking a union bound we know that all of these events happen simultaneously with probability at least $1 - T^{-11}$. We then have for τ' and its closest element τ in $\mathcal{C}$:

$$|\Delta(\tau) - \Delta(\tau')| + |\hat{\Delta}(\tau) - \hat{\Delta}(\tau')| \;\leq\; 4|\tau - \tau'| \;\leq\; \frac{4}{T}. \tag{B.12}$$

We simply upper-bound $\frac{4}{T}$ by ϵ. This must be true because $\epsilon \geq T^{-\frac{1}{2}} = \omega(\frac{1}{T})$. Then, combining the bound in (B.12) with (B.10) implies $|\hat{\Delta}(\tau') - \Delta(\tau')| \leq 2\epsilon$ holds with probability $1 - T^{-11}$ for all $\tau \in [\ell, u]$.

Next, we bound $|\hat{R}_\ell - R(\ell)|$ and $|\hat{R}_u - R(u)|$. For $|\hat{R}_\ell - R(\ell)|$, Notice that $\hat{R}_\ell$ is an estimate of $R(\ell)$ with $N = C \cdot \frac{\log T}{\epsilon^2}$ samples, and the reward of each sample falls in $[-1, 1]$. By Hoeffding's Inequality (Theorem A.1), the probability that $|\hat{R}_\ell - R(\ell)| > \epsilon$ is bounded by $2\exp(-2N\epsilon^2/4) = 2T^{-C/2}$. So, with probability $1 - T^{-11}$ $|\hat{R}_\ell - R(\ell)| \leq \epsilon$ when $C > 100$. The bound for $|\hat{R}_u - R(u)|$ is identical. Finally, combining three parts with union bound finishes the proof. □

Finally, we have the tools to prove Lemma 2.7:

Proof of Lemma 2.7. We will assume that $|\hat{\delta}(\tau) - \delta(\tau)| \leq 4\epsilon$, which is true with probability $1 - T^{-10}$ by Claim 2.9.

Observe that $\hat{\delta}(\tau)$ is a monotone increasing function, because $\hat{\delta}'(\tau) = \hat{\Delta}'(\tau) = (\hat{F}_1(u) - \hat{F}_1(\ell))(1 - \hat{F}_2(\tau) \geq 0$. Therefore, according to the definition of ℓ' and u', we have $[\ell', u'] = \{\tau \in [\ell, u] : |\hat{\delta}(\tau)| \leq 4\epsilon\}$. Now, we can use this property to prove two statements separately:

For the statement that $\tau^* \in [\ell', u']$, notice that $\delta(\tau^*) = 0$. According to Claim 2.9, $|\hat{\delta}(\tau^*)| \leq 4\epsilon$. Then, since $\tau^* \in [\ell, u]$ and $|\hat{\delta}(\tau^*)| \leq 4\epsilon$, there must be $\tau^* \in [\ell', u']$, because $[\ell', u'] = \{\tau \in [\ell, u] : |\hat{\delta}(\tau)| \leq 4\epsilon\}$.

Next, we prove that $|\delta(\tau)| \leq 8\epsilon$ for all $\tau \in [\ell, u]$. This is true because $[\ell', u'] = \{\tau \in [\ell, u] : |\hat{\delta}(\tau)| \leq 4\epsilon\}$, and we have $|\hat{\delta}(\tau) - \delta(\tau)| \leq 4\epsilon$ from Claim 2.9. Therefore, $|\delta(\tau)| \leq |\hat{\delta}(\tau)| + |\hat{\delta}(\tau) - \delta(\tau)| \leq 8\epsilon$ for all $\tau \in [\ell', u']$.

Finally, the bound $R(\tau^*) - R(\tau) \leq 8\epsilon$ directly follows Claim 2.8 and that $|\delta(\tau)| \leq 8\epsilon$. □

B.2.2 Proof of Theorem 2.10 To prove Theorem 2.10, we need to first give an initialization algorithm such that its output should satisfy the conditions listed in Lemma 2.7. Formally, we have the following lemma:

Lemma B.1. *After running no more than $1000\sqrt{T}\log T$ samples from $\mathcal{D}_1$ and $\mathcal{D}_2$, with probability $1 - T^{-10}$ we can output an* initial interval *$[\ell, u]$ that satisfies $|g(\tau)| \leq T^{-1/4}$ and $\tau^* \in [\ell, u]$.*

Proof. We first run $1000\sqrt{T}\log T$ extra samples for X_2 and calculate an estimate $\hat{F}_2(x)$. We can show that $|\hat{F}_2(x) - F_2(x)| \leq \frac{1}{2}T^{-\frac{1}{4}}$ with probability $1 - T^{-10}$: After running $N = C \cdot \sqrt{T}\log T$ samples, the DKW inequality (Theorem A.3) shows that $\mathbf{Pr}\left[|\hat{F}_2(x) - F_2(x)| > \varepsilon = \frac{1}{2}T^{-\frac{1}{4}}\right] \leq 2\exp(-2N\varepsilon^2) = 2T^{-C/2}$. Then, with probability $1 - T^{-10}$, we have $|\hat{F}_2(x) - F_2(x)| \leq \frac{1}{2}T^{-\frac{1}{4}}$ simultaneously holds for all $x \in [0, 1]$ when $C > 100$. In the following proof, we assume this accuracy bound always holds.

Next, we calculate $\hat{g}(\tau)$ by replacing $F_2(x)$ with $\hat{F}_2(x)$ in (2.3). When $|\hat{F}_2(x) - F_2(x)| \leq \frac{1}{2}T^{-\frac{1}{4}}$ holds simultaneously for all $x \in [0, 1]$, we have $|\hat{g}(\tau) - g(\tau)| \leq \int_\tau^1 |\hat{F}_2(x) - F_2(x)| \leq \frac{1}{2}T^{-\frac{1}{4}}$. Then, we let $[\ell, u] := \{\tau : |\hat{g}(\tau)| \leq \frac{1}{2}T^{-\frac{1}{4}}\}$. Since $\hat{g}'(\tau) = \hat{F}_2(\tau) - 1 \leq 0$, function $\hat{g}(\tau)$ is a non-increasing. So, the set $\{\tau : |\hat{g}(\tau)| \leq \frac{1}{2}T^{-\frac{1}{4}}\}$ must form an interval. Besides, notice that $g(\tau^*) = 0$, which means $|\hat{g}(\tau^*)| \leq \frac{1}{2}T^{-\frac{1}{4}}$, so we must have $\tau^* \in [\ell, u]$. Furthermore, for every $\tau \in [\ell, u]$, $|g(\tau)| \leq |\hat{g}(\tau)| + |\hat{g}(\tau) - g(\tau)| \leq T^{-\frac{1}{4}}$, which finishes the proof. □

Now, we are ready to prove Theorem 2.10:

Proof of Theorem 2.10. For the core part of the algorithm, we run Algorithm 2 and then use Lemma 2.5 to bound the regret. To run Algorithm 2, we let the constraints to mean that the threshold played in each round is inside the interval $[\ell, u]$ given by Algorithm 3. Besides, we require the conditions listed in Lemma 2.7 hold (with high probability). We discuss them separately:

- $|g(\tau)| \leq T^{-\frac{1}{4}}$ for all $\tau \in [\ell, u]$: This is guaranteed by Lemma B.1.
- $\tau^* \in [\ell, u]$: For the first phase, this is guaranteed by Lemma B.1. Starting from the second phase, this is from Lemma 2.7 of the previous phase.
- $|\delta(\tau)| \leq 16\epsilon$: For the first phase, this is true because $\epsilon_1 = 1$. Starting from the second phase, this is from Lemma 2.7 of the previous phase. Notice that the statement in Lemma 2.7 is a little bit different: It guarantees that $|\delta(\tau)| \leq 8\epsilon$ with respect to the $[\ell, u]$ and ϵ from the previous phase. When switching to the

new phase, notice that $F_2(u') - F_2(\ell') \le F_2(u) - F_2(\ell)$, which means $|\delta(\tau)|$ drops when switching to the new phases. Besides, the parameter ϵ_{new} in the new phase is exactly $\frac{1}{2}\epsilon_{old}$. Combining these two differences shows that $|\delta(\tau)| \le 16\epsilon$ holds in the new phase.

Therefore, Algorithm 3 satisfies algorithm Alg in Lemma 2.5. Applying Lemma 2.5 gives the $O(\sqrt{T}\log T)$ regret bound.

Besides, we also run samples for initialization and constructing CDF estimates for Algorithm 3. These are not coverd by Lemma 2.5. For the initialization, Lemma B.1 states that $\Theta(\sqrt{T}\log T)$ rounds are sufficient. So the regret from the initialization is $O(\sqrt{T}\log T)$. For constructing $\hat{F}_1(x)$ and $\hat{F}_2(x)$, assume we run k phases, then the total number of samples is

$$\sum_{i=1}^{k}\Theta(\frac{\log T}{\epsilon_i}) = O(\sqrt{T}\log T).$$

Combining three parts finishes the proof. □

C Missing Proofs from Section 4

C.1 Proof of Lemma 4.1

Proof. We first prove the lemma for a single i. For $[\ell_i, u_i]$, we run $C\cdot\sqrt{T}\log T$ extra samples for X_i with $C = 1000$, and calculate an estimate $\hat{F}_i(x)$. We can show that $|\hat{F}_i(x) - F_i(x)| \le \frac{1}{2}T^{-\frac{1}{4}}$ with probability $1 - T^{-11}$: After running $N = C\cdot\sqrt{T}\log T$ samples, the DKW inequality (Theorem A.3) shows that $\mathbf{Pr}\left[|\hat{F}_i(x) - F_i(x)| > \varepsilon = \frac{1}{2}T^{-\frac{1}{4}}\right] \le 2\exp(-2N\varepsilon^2) = 2T^{-C/2}$. Then with probability $1 - T^{-11}$, we have $|\hat{F}_i(x) - F_i(x)| \le \frac{1}{2}T^{-\frac{1}{4}}$ holds for every $x \in [0,1]$ when $C > 100$. In the following proof, we assume this accuracy bound always holds. By the union bound over all $i \in [n]$, the whole proof succeeds with probability $1 - T^{-10}$.

Next, we calculate $\hat{g}_i(\tau)$ by replacing $F_i(x)$ with $\hat{F}_i(x)$ in (4.11). When $|\hat{F}_i(x) - F_i(x)| \le T^{-\frac{1}{4}}$ holds for all $x \in [0,1]$, we have $|\hat{g}_i(\tau) - g_i(\tau)| \le \int_\tau^1 |\hat{F}_i(x) - F_i(x)| \le \frac{1}{2}T^{-\frac{1}{4}}$. Then, we let $[\ell_i, u_i] := \{\tau : |\hat{g}_i(\tau)| \le \frac{1}{2}T^{-\frac{1}{4}}\}$. Since $\hat{g}'(\tau) = \hat{F}_i(\tau) - 1 \le 0$, which means $\hat{g}_i(\tau)$ is a decreasing function, then the set $\{\tau : |\hat{g}_i(\tau)| \le \frac{1}{2}T^{-\frac{1}{4}}\}$ must form an interval. Besides, notice that $g_i(\tau^*) = 0$, which means $|\hat{g}_i(\tau^*)| \le \frac{1}{2}T^{-\frac{1}{4}}$, so there must be $\tau_i^* = \sigma_i \in [\ell_i, u_i]$. Furthermore, for every $\tau \in [\ell_i, u_i]$, $|g_i(\tau)| \le |\hat{g}_i(\tau)| + |\hat{g}_i(\tau) - g_i(\tau)| \le T^{-\frac{1}{4}}$.

Finally, combining the statements for all n intervals finishes the proof. □

C.2 Proof of Claim 4.6

Proof. We first show that $|\hat{F}_i(x) - F_i(x)| \le \frac{\sqrt{\epsilon}}{2n}$ with probability $1 - T^{-13}$ with $N = C\cdot\frac{n^2\log T}{\epsilon}$ samples, where C is set to be 1000. Using DKW inequality (Theorem A.3), we have $\mathbf{Pr}\left[|\hat{F}_i(x) - F_i(x)| > \frac{\sqrt{\epsilon}}{2n}\right] \le 2\exp(-2N\frac{\epsilon}{4n^2}) = 2T^{-C/4}$. So the bound holds with probability $1 - T^{-13}$ when $C = 1000$. By the union bound, with probability $1 - T^{-12}$ we have $|\hat{F}_i(x) - F_i(x)| \le \frac{\sqrt{\epsilon}}{2n}$ holds for every $i \in [n]$. Then, for the accuracy of $\prod_{i\in S} F_i(x)$, we have $\left((1 - \frac{\sqrt{\epsilon}}{2n})^n - 1\right) \le \prod_{i\in S}\hat{F}_i(x) - \prod_{i\in S}F_i(x) \le \left((1 + \frac{\sqrt{\epsilon}}{2n})^n - 1\right)$. For the lower bound, we have $(1 - \frac{\sqrt{\epsilon}}{2n})^n - 1 \ge 1 - \frac{\sqrt{\epsilon}}{2} - 1 > -\sqrt{\epsilon}$. For the upper bound, we have $(1 + \frac{\sqrt{\epsilon}}{2n})^n - 1 \le \exp(\frac{\sqrt{\epsilon}}{2n}\cdot n) - 1 \le 1 + 2\cdot\frac{\sqrt{\epsilon}}{2} - 1 = \sqrt{\epsilon}$. Combining two bounds finishes the proof. □

C.3 Proof of Claim 4.7

Proof. Since $\delta_i(\tau) = \Delta_i(\tau) - (R_u - R_\ell)$, there are three parts in $\delta_i(\tau)$. We show that the accuracy of each part is bounded by $\frac{\epsilon}{3}$ with probability $1 - T^{13}$, then taking a union bound over three accuracy bounds gives Claim 4.7.

First, similar to the derivation in Equation (B.5) we bound the magnitude of the Δ_i function:

$$\begin{aligned}|\Delta_i(\tau)| &\leq (F_{\pi,i}(u)-F_{\pi,i}(\tau))\int_\tau^u (1-F_i(x))dx + (F_{\pi,i}(\tau)-F_{\pi,i}(\ell))\int_\ell^\tau (1-F_i(x))dx\\ &= (F_{\pi,i}(u)-F_{\pi,i}(\tau))(g_i(\tau)-g_i(u)) + (F_{\pi,i}(\tau)-F_{\pi,i}(\ell))(g_i(\ell)-g_i(\tau))\\ &\leq (F_{\pi,i}(u)-F_{\pi,i}(\ell))(g_i(\ell)-g_i(u))\\ &\leq |\delta_{\pi,u,\ell,i}(\ell)| + |\delta_{\pi,u,\ell,i}(u)| \leq 12\epsilon, \end{aligned} \tag{C.13}$$

where we use the bound $|\delta_{\pi',u',\ell',i}(\tau)| \leq 6\epsilon$ in Lemma 4.5.

Next, we hope to propose an estimator $\hat{\Delta}_i^{(k)}(\tau)$ for the Δ_i function which uses $N = C \cdot \frac{\log T}{\epsilon}$ samples for $C = 10^5$. For $k \in [N]$, define

$$\hat{\Delta}_i^{(k)}(\tau) = \int_\tau^u (\hat{F}_{\pi,i}^{(k)}(u) - \hat{F}_{\pi,i}^{(k)}(x))(\hat{F}_i^{(k)}(x)-1)dx - \int_\ell^\tau (\hat{F}_{\pi,i}^{(k)}(x) - \hat{F}_{\pi,i}^{(k)}(\ell))(\hat{F}_i^{(k)}(x)-1)dx,$$

where $\hat{F}_{\pi,i}^{(k)}(.)$ and $\hat{F}_i^{(k)}(.)$ are simple threshold functions at the ith initial sample, which are estimates for the densities $F_{\pi,i}$ and F_i, respectively. This definition implies $\hat{\Delta}_i(\tau) = \frac{1}{N}\sum_{k\in[N]} \hat{\Delta}_i^{(k)}(\tau)$, and Equation (C.13) implies

$$\mathbf{E}\left[\hat{\Delta}_i^{(k)}(\tau)\right] = \Delta_i(\tau) \leq 12\epsilon. \tag{C.14}$$

Now it is easy to see that $\hat{F}_{\pi,i}^{(k)}(x) - \hat{F}_{\pi,i}^{(k)}(y)$ is a Bernoulli random variable which are one if and only if the maximum value obtained from $X_{\pi(1)}, \ldots, X_{\pi(\pi^{-1}(i)-1)}$ is in $[\ell_i, u_i]$. In particular, this implies that $\hat{\Delta}_i^{(k)}(\tau)$ is bounded by 1 since

$$\begin{aligned}|\hat{\Delta}_i^{(k)}(\tau)| &\leq \int_\tau^u \left|(\hat{F}_{\pi,i}^{(k)}(u) - \hat{F}_{\pi,i}^{(k)}(x))(\hat{F}_i^{(k)}(x)-1)\right|dx - \int_\ell^\tau \left|(\hat{F}_{\pi,i}^{(k)}(x) - \hat{F}_{\pi,i}^{(k)}(\ell))(\hat{F}_i^{(k)}(x)-1)\right|dx\\ &\leq \int_\tau^u 1dx + \int_\ell^\tau 1dx = u-\ell \leq 1.\end{aligned} \tag{C.15}$$

Combining Equations (C.14) and (C.15), we have the variance bound:

$$\mathsf{Var}[\hat{\Delta}_i(\tau)] \leq \mathbf{E}\left[\hat{\Delta}_i(\tau)^2\right] \leq \mathbf{E}\left[\hat{\Delta}_i(\tau)\right] \leq 12\epsilon. \tag{C.16}$$

Hence, using Bernstein inequality, we have

$$\mathbf{Pr}\left[|\hat{\Delta}_i(\tau) - \Delta_i(\tau)| \geq \frac{\epsilon}{12}\right] \leq 2\exp\left(-\frac{N\epsilon^2/144}{2\mathsf{Var}[\hat{\Delta}_i(\tau)] + \frac{2}{3}\frac{\epsilon}{12}}\right) = 2T^{-\frac{C}{3464}}. \tag{C.17}$$

Therefore, $|\hat{\Delta}_i(\tau) - \Delta_i(\tau)| \leq \frac{\epsilon}{12}$ holds with probability $1 - T^{-14}$ when $C = 10^5$.

The bound above is only for a single τ. To give the bound for a whole interval, we discretize $[\ell_i, u_i]$ uniformly into a discrete set $\mathcal{C}$ and make sure that each pair of adjacent $\tau, \tau' \in \mathcal{C}$ follows $|\tau - \tau'| \leq \frac{1}{T}$. Then, there must be $|\mathcal{C}| \leq T$ and the union bound implies $|\hat{\Delta}_i(\tau) - \Delta_i(\tau)| \leq \frac{\epsilon}{12}$ holds with probability $1 - T^{-13}$ for all $\tau \in \mathcal{C}$.

Next, we bound the Lipschitz constant of Δ_i (and similarly $\hat{\Delta}_i$):

$$\begin{aligned}\left|\Delta_i(\tau) - \Delta_i(\tau')\right| &= \left|\int_\tau^{\tau'} (F_{\pi,i}(u) - F_{\pi,i}(x))(F_i(x)-1)dx\right| + \left|\int_\tau^{\tau'} (F_{\pi,i}(u) - F_{\pi,i}(x))(F_i(x)-1)dx\right|\\ &\leq 2|\tau - \tau'|.\end{aligned}$$

Finally, for every $\tau \in [\ell_i, u_i]$, let τ' be the closest value in $\mathcal{C}$. Then, we have:

$$\left|\hat{\Delta}_i(\tau) - \Delta_i(\tau)\right| \le \left|\Delta_i(\tau) - \Delta_i(\tau')\right| + \left|\hat{\Delta}_i(\tau') - \Delta_i(\tau')\right| + \left|\hat{\Delta}_i(\tau') - \hat{\Delta}_i(\tau)\right| \le \frac{\epsilon}{12} + \frac{4}{T} \le \frac{\epsilon}{6},$$

where the last inequality is true because $\epsilon > T^{-\frac{1}{2}} = \omega(\frac{1}{T})$. Therefore, with probability $1 - T^{-13}$, we have $\left|\hat{\Delta}_i(\tau) - \Delta_i(\tau)\right| \le \frac{\epsilon}{6}$ for all $\tau \in [\ell_i, u_i]$ simultaneously.

Next, we use Hoeffding's Inequality (Theorem A.1) to bound the accuracy of $|R_\ell - \hat{R}_\ell|$. In each round, the reward falls in [-0.5,0.5]. Then, after running $N = C \cdot \epsilon^{-2} \log T$ samples, we have $\mathbf{Pr}\left[|R_\ell - \hat{R}_\ell| > \frac{\epsilon}{6}\right] \le 2\exp(-2N\epsilon^2/36) = 2T^{-C/18}$. Therefore, $|R_\ell - \hat{R}_\ell| \le \frac{\epsilon}{6}$ with probability $1 - T^{-13}$ when $C > 1000$. Besides, the proof for $|R_u - \hat{R}_u|$ is identical. Combining three parts with union bound finishes the proof. □

C.4 Proof of Lemma 4.8 In this section, we show that Algorithm 10 finds an approximately clever threshold setting. We first introduce the following lemma:

Algorithm 10: Finding Approximately Clever Threshold

Input: (I, S), m, i, $\hat{F}_1(x), ..., \hat{F}_n(x)$

for $P \subseteq [n]$ **do**

  if $\exists k : (k, i) \in S \wedge k \notin P$ *or* $\exists k : (i, k) \in S \wedge k \in P$ *or* $\exists k, j : (k, j) \in S \wedge k \notin P \wedge j \in P$ **then** Skip this P ;

  For $k \in P$, let $\tau_k = u_k$

  For $k \in [n] \setminus (T \cup \{i\})$, let $\tau_k = \ell_k$

  Let $u_T = \min\{u_i, \tau_{k:k\in T}\}$, $\ell_T = \max\{\ell_i, \tau_{k:k\notin(T\cup\{i\})}\}$

  Set partial setting π_T be: Let $\tau_i \in [\ell_T, u_T]$. π_T sorts the thresholds in a decreasing order. Break the ties according to the constraints in S.

  Let $\hat{F}_{\pi_T,i}(x) = \prod_{k\in T} \hat{F}_k(x)$.

  Calculate $q_T := \hat{F}_{\pi_T,i}(u_T) - \hat{F}_{\pi_T,i}(\ell_T)$

Let $T^* = \arg\max q_T$.

Output: $\pi_{T^*}, \ell_{T^*}, u_{T^*}, \hat{F}_{\pi_{T^*},i}$.

Lemma C.1. *Algorithm 10 calculates a clever threshold setting, up to an $4\sqrt{\epsilon}$ additive error. The running time of Algorithm 10 is $O(n \cdot 2^n)$.*

Proof. The goal of a *clever* threshold setting is to maximize $F_{\pi,i}(u) - F_{\pi,i}(\ell)$. Fix i. When the set P, which represents the distributions in front of X_i is determined, the function $F_{\pi,i}(x)$ is fixed. Therefore, to maximize $F_{\pi,i}(u) - F_{\pi,i}(\ell)$, we should maximize u and minimize ℓ. This can be achieved by maximizing the thresholds in P and minimizing the thresholds in $[n] \setminus (P \cup \{i\})$, which is exactly lines 8 and 9 in Algorithm 6. Then, after enumerating all valid subsets P, we can find a setting that maximizes $F_{\pi,i}(u) - F_{\pi,i}(\ell)$.

There is one missing detail: we only know the value of $\hat{F}_i(x)$. From Claim 4.6, we know $\hat{F}_i(x)$ is an estimate of $F_i(x)$ with accuracy $\sqrt{\epsilon}$. Therefore, $\max_\pi \hat{F}_{\pi,i}(u) - \hat{F}_{\pi,i}(\ell)$ is at most $2\sqrt{\epsilon}$ different from $\max_\pi F_{\pi,i}(u) - F_{\pi,i}(\ell)$. After getting $\pi' = \arg\max_\pi \hat{F}_{\pi,i}(u) - \hat{F}_{\pi,i}(\ell)$, the real value of $F_{\pi',i}(u) - F_{\pi',i}(\ell)$ is at most $2\sqrt{\epsilon}$ different from $\hat{F}_{\pi',i}(u) - \hat{F}_{\pi',i}(\ell)$. Combining two errors proves the $4\sqrt{\epsilon}$ error bound.

For the running time of Algorithm 10, we need to enumerate a subset S, then calculate the corresponding $F_{\pi,i}(x)$ function. So the running time is $O(n \cdot 2^n)$. □