LINEAR CONTRACT DESIGN WITH PREDICTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We study the problem of contract design with predictions, where a principal delegates decision-making to an agent and aims to induce the agent to take actions that lead to outcomes in favor of the principal. The principal designs a contract that ties compensation to these outcomes. Faced with the contract, the agent chooses an action to maximize his utility. While it is common that only the agent knows his action's cost and the outcome distribution induced by his action, we consider a special case where the principal has access to prediction of this information. The principal's challenge in contract design is to pursue both consistency, i.e., achieving near-optimal performance if the prediction is highly accurate, and robustness, i.e., maintaining performance under inaccurate prediction. Our main contribution is the development of two algorithms that satisfy both consistency and robustness, along with the technical methodologies employed.

1 INTRODUCTION

The principal-agent problem studies how a principal designs incentives for a self-interested agent whose actions are hidden. Since the principal only observes the final outcomes, a contract specifies outcome-dependent payments to induce desirable agent behaviors. Given a contract, the agent chooses an action that maximizes the difference between expected payment and cost, while the principal aims to maximize the resulting profit.

Previous contract design studies often assume that the principal has complete knowledge of the agent's costs and outcome distributions (e.g., Dütting et al. (2023); Cacciamani et al. (2024); Castiglioni et al. (2023)). However, such information is typically private to the agent in practice. Recent works therefore study contract design under incomplete information, including Bayesian approaches Castiglioni et al. (2021) and robust contracts when only partial information is available Carroll (2015).

A common approach to address incomplete information is to leverage additional data to estimate the agent's private characteristics. In practice, the principal may obtain predictions of the action costs and expected rewards from historical data or machine learning models. Incorporating such predictive information into algorithm design has recently gained increasing attention, with the goal of improving performance while maintaining robustness against prediction errors.

In this work, we study linear contract design with predictions. A linear contract is parameterized by $\alpha \in [0, 1]$, representing the fraction of the outcome reward transferred to the agent. We assume that the principal has access to predictions of the agent's action costs and expected rewards. Our goal is to design a contract that achieves both consistency, i.e., near-optimal performance when predictions are accurate, and robustness, i.e., bounded performance degradation under inaccurate predictions.

Unlike previous prediction-augmented algorithmic problems, contract design with predictions faces an additional challenge arising from the agent's strategic behavior. Even small prediction errors can significantly alter the agent's best-response action, leading to substantial errors in estimating the principal's profit under a given contract. Therefore, identifying an optimal linear contract based on predictions is highly nontrivial (see Section 3).

Our contributions are summarized as follows:

- We formulate contract design with predictions and identify the fundamental challenge caused by the sensitivity of agents' best responses to prediction errors.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

- We propose a discretization-based algorithm that constructs candidate best-response sets and uses pessimistic profit estimation to achieve both consistency and robustness guarantees.
- We develop a polynomial-time breakpoint-based algorithm that handles the instability of predicted breakpoints by identifying reliable action pairs and performing pessimistic search over tolerant breakpoints, while preserving consistency and robustness guarantees.

1.1 Related Works

Algorithm with Predictions. The study of algorithms with predictions has garnered significant attention in recent years. For a comprehensive overview, see the survey Mitzenmacher & Vassilvitskii (2022). In the domain of online algorithms with predictions, a growing body of work (e.g., Purohit et al. (2018); Gollapudi & Panigrahi (2019); Bai & Coester (2023); Rohatgi (2020); Azar et al. (2024); Berg & Kamali (2024)) demonstrates how machine-learned predictions can enhance algorithmic performance. In the context of mechanism design with predictions, Xu & Lu (2022) initiates the field by studying classical problems such as auctions, frugal path auctions, scheduling, and facility location. Subsequent work (e.g., Agrawal et al. (2022); Gkatzelis et al. (2022); Balkanski et al. (2022); Istrate & Bonchis (2022); Caragiannis & Kalantzis (2024); Barak et al. (2024)) has expanded this research by addressing a broader range of mechanism design problems with predictions.

Contract Theory. Contract theory is a foundational area of economics with broad applications and a well-established history, tracing back to seminal works such as Holmström (1979) and Grossman & Hart (1992). For an introduction, see the comprehensive book Bolton & Dewatripont (2004) and the overview Caillaud & Hermalin (2000). A major strand of research addresses contract design under imperfect information. Carroll (2015) initiates the study of robust contracts where the principal lacks full knowledge of the agent's action set, highlighting the effectiveness of linear contracts. Further developments on robust contracts under uncertainty are presented in works such as Carroll (2019) and Dütting et al. (2019). Several studies adopt a Bayesian perspective, modeling agent types as drawn from a distribution (e.g., Castiglioni et al. (2021); Guruganesh et al. (2021); Castiglioni et al. (2022); Guruganesh et al. (2023); Alon et al. (2023)).

This paper introduces a novel framework by integrating predictions into contract design. To the best of our knowledge, it is the first study to incorporate predictions into contract theory.

2 Setup

The contract scenario involves a single principal and a single agent. The principal delegates a specific task to the agent and offers payment contingent upon the outcome achieved. Let there be a total of O potential outcomes, with each outcome o associated with a reward r_o for the principal. For the agent, there are N possible actions, where each action i incurs a cost c_i and is characterized by a probability distribution $\mathbf{F}_i$ over the potential outcomes. Specifically, F_{io} represents the probability that action i will result in outcome o. Consequently, the principal's expected reward when the agent chooses action i can be calculated as: $R_i = \mathbb{E}_{o\sim\mathbf{F}_i}[r_o] = \sum_{o\in[O]} F_{io} r_o$. [1]

Contract. A contract specifies a payment scheme $\boldsymbol{t} \in [0,\infty)^O$, which is represented as an O-dimensional vector. This vector outlines a payment $t_o \geq 0$ that the principal transfers to the agent for each possible outcome o. The expected payment associated with the agent choosing action i, denoted T_i, is calculated as: $T_i = \mathbb{E}_{o\sim\mathbf{F}_i}[t_o] = \sum_{o\in[O]} F_{io} t_o$.

Incentivized Action. Given a contract $\boldsymbol{t}$, the agent will select the action that maximizes his own utility. Specifically, the agent's (expected) utility for choosing action i is defined as $U_i = T_i - c_i$. Consequently, the agent will choose the action that maximizes his utility: $i^* = \arg\max_{i\in[N]} U_i = \arg\max_{i\in[N]}\{T_i - c_i\}$. In cases of indifference, where multiple actions result in the same maximum utility, we use the commonly adopted assumption that he will select the action that is most favorable to the principal. If the agent selects action i^* under contract $\boldsymbol{t}$, we say that the contract $\boldsymbol{t}$ incentivizes action i^*, and we refer to i^* as the incentivized action or the best-response action.

[1] Without loss of generality, we assume the expected rewards are ordered such that $0 \leq R_1 \leq ... \leq R_N$.

Profit of the Principal. For the principal, profit is calculated as the difference between the reward associated with the realized outcome and the payment made to the agent. Specifically, given a contract $\boldsymbol{t}$, the principal's profit for a particular outcome o is $r_o - t_o$. The expected profit for the principal, considering the agent's best response, is defined as: $p = R_{i^*} - T_{i^*}$, where i^* is the agent's best-response action towards contract $\boldsymbol{t}$, or to say the incentivized action.

In the subsequent sections of this paper, when we refer to profit or reward, we mean their expected values rather than their deterministic counterparts.

2.1 PREDICTION

In practice, the principal may lack complete information about the true outcome distributions and costs induced by the agent's actions. Instead, we assume that the principal has access to predictions, which allow her to estimate the expected reward and cost for each action. Let $(\hat{\boldsymbol{R}}, \hat{\boldsymbol{c}})$ represent the predictions, where $\hat{R}_i$ and $\hat{c}_i$ denote the predicted reward and cost for action i, respectively.

The Information Structure. The principal observes the reward r_o for each outcome o, but lacks access to the agent's private information, including action costs and outcome distributions. Instead, she relies on predictions of expected rewards and costs to design a contract $\boldsymbol{t}$. The agent, however, knows this private information and chooses a best-response action upon receiving the contract. His decision depends only on the contract and his own information, rather than the principal's predicted reward information.

The following definition of prediction error quantifies the degree to which the predicted values approximate the true reward and cost associated with each action.

Definition 1 (Prediction Error)**.** *We say that the predictions $(\hat{\boldsymbol{R}}, \hat{\boldsymbol{c}})$ have a prediction error of γ if*

$$\gamma = \max_{i \in [N]} \left\{ \frac{\hat{R}_i}{R_i}, \frac{R_i}{\hat{R}_i}, \frac{\hat{c}_i}{c_i}, \frac{c_i}{\hat{c}_i} \right\},$$

where the ratios involving costs are interpreted as follows: $\hat{x}/x = +\infty$ if $x = 0$ and $\hat{x} > 0$, $\hat{x}/x = 1$ if $x = \hat{x} = 0$, and $\hat{x}/x$ takes its usual value otherwise. The same convention applies to $x/\hat{x}$.

The following definition of the approximation ratio illustrates how the performance of an algorithm that utilizes predictions compares to the optimal profit achievable with perfect information.

Definition 2 (Approximation ratio)**.** *The approximation ratio r of an algorithm $\mathbf{A}$ is defined as the factor between the expected profit obtained under the optimal parameter with perfect information and the expected profit achieved by $\mathbf{A}$, using the parameter selected based on predictions.*

With access to predictions, the principal's goal in designing a contract is to maximize her own profit while considering two key aspects:

Consistency. This means that when the predictions are close to the true values (i.e., when γ is close to 1), the principal's profit should be close to the optimal profit that would be achievable under perfect information (i.e., the approximation ratio r approaches 1).

Robustness. Robustness implies that even when the predictions deviate significantly from the true values (i.e., when γ is arbitrary), the principal's profit does not suffer excessively.

Notably, in the context of consistency, our goal is for the algorithm to perform well under highly accurate, rather than perfectly accurate, predictions. This distinction prevents the algorithm from directly outputting the optimal parameter based solely on the predictions.

To facilitate analysis of the contract setting with predictions, we make the following assumptions.

Assumption 1.

A1 *There exists a zero-cost action, denoted as action 1, such that $c_1 = 0$.*

A2 *There are no dominated actions. That is, for any action j, there does not exist an action $i \neq j$ such that $R_i \geq R_j$ and $c_i \leq c_j$ simultaneously.*

A3 *The reward for each outcome is at least 1 and at most R. Without loss of generality, we assume $c_i \leq R$ for all actions.*

A4 *The predictions satisfy the same conditions as the true parameters.*

Assumptions A1 and A2 are standard in contract theory (e.g., Dütting et al. (2019)). Assumption A3, which bounds the rewards, is widely used in algorithms with predictions (e.g., Xu & Lu (2022)) and is crucial for establishing robustness guarantees. The normalization $c_i \leq R$ is without loss of generality, since any action with $c_i > R$ cannot be incentivized by a linear contract and can therefore be removed without affecting the optimal solution. Assumption A4 is also without loss of generality, as the principal can adjust the predictions to satisfy the required conditions without violating the prediction error bounds.

2.2 Linear Contract

A linear contract is a specific type of incentive contract in which the principal commits to providing the agent with a fixed proportion of the principal's own reward. In this contract form, the principal selects a fractional parameter, denoted by α, which represents the proportion of the outcome rewards attributed to the agent. Under this arrangement, if the agent achieves an outcome o, the direct payment is calculated as αr_o. Furthermore, the agent's utility for action i is given by $U_i = \alpha R_i - c_i$. The principal's profit is $p = (1-\alpha)R_{i^*}$, where i^* is the incentivized action under the parameter α.

Why Linear Contracts? Building on the findings from previous works, the linear contract is convenient to employ and robust under uncertainty. For instance, it is well-established in Dütting et al. (2019) that a linear contract can guarantee optimal worst-case profits for the principal when the specific distributions are unknown, and only expected rewards are available. Specifically, even when the expected reward is known, there generally exists a distribution where a linear contract can achieve optimal profit, highlighting the effectiveness of the linear contract framework. Given its robustness and simplicity, we apply the linear contract in the context of this paper.

We now present some fundamental properties of linear contracts. Interested readers can refer to Dütting et al. (2019) for detailed proofs.

Proposition 1. *If $\alpha_1 \geq \alpha_2$, the incentivized action at α_1 will have weakly higher expected reward than the incentivized action at α_2.*

Proposition 1 illustrates the relationship between the expected rewards of incentivized actions at different linear factors. The following concept of a breakpoint is critical for analyzing linear contracts, as it denotes the parameter at which two actions provide the agent with equal utility.

Definition 3 (Breakpoint)**.** *A breakpoint, denoted as $\alpha_{i,j}$, is the parameter value at which the agent is indifferent between taking action i or action j, i.e., $U_i = \alpha_{i,j}R_i - c_i = U_j = \alpha_{i,j}R_j - c_j$ holds. Solving for $\alpha_{i,j}$, we obtain $\alpha_{i,j} = \frac{c_i - c_j}{R_i - R_j}$.*

However, not all breakpoints will result in the agent switching actions. Specifically, there are instances where at a particular breakpoint $\alpha_{i,j}$, neither action i nor action j provides the highest utility for the agent. As a result, only a subset of breakpoints represent the "critical breakpoints", which correspond to transitions in the agent's best response action.

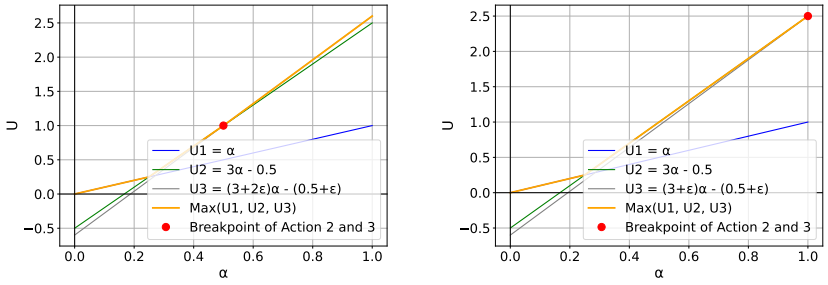
Definition 4 (Critical Breakpoint)**.** *A critical breakpoint is a parameter value at which the agent's best response action changes.*

It is obvious that there are at most $N-1$ critical breakpoints, denoted as $(\alpha_{i_1,i_2}, \alpha_{i_2,i_3}, ..., \alpha_{i_{n-1},i_n})$ where $i_1 = 1, n \leq N-1$. These breakpoints correspond to the transitions in the agent's best response actions. Consequently, for any point α such that $\alpha_{i_{m-1},i_m} \leq \alpha \leq \alpha_{i_m,i_{m+1}}$, the incentivized action for the agent is action i_m. The following proposition underscores the importance of critical breakpoints.

Proposition 2. *The optimal linear contract chooses a parameter from the set consisting of the endpoint $\alpha = 0$ and the critical breakpoints.*

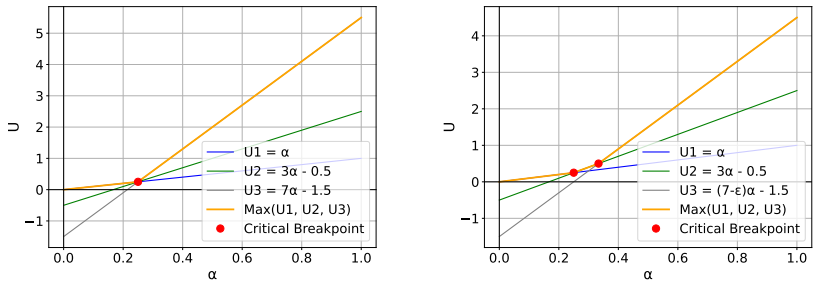
3 The Challenge of Sensitivity in Best Response

This section highlights the challenge of designing optimal linear contracts with predictions. The primary difficulty stems from the sensitivity of the agent's best response to prediction errors, where



(i) Utility calculated based on predictions. (ii) Utility calculated based on ground-truth.

Fig. 1. Substantially different breakpoint even with high accuracy.



(i) Utility calculated based on predictions. (ii) Utility calculated based on ground-truth.

Fig. 2. Different number and order of critical breakpoints even with high accuracy.

even small inaccuracies can significantly alter the agent's chosen actions. To illustrate this sensitivity, we present several figures comparing the agent's utility under predicted and ground-truth scenarios. Since breakpoints are key indicators of the agent's best response, we specifically analyze and compare breakpoints derived from both ground-truth data and predictions.

Figure 1 illustrates that even small prediction errors can lead to substantial deviations in the computed breakpoints. The y-axis represents the agent's utility U, while the x-axis denotes the linear parameter α. The agent selects the action that maximizes utility, depicted by the orange piecewise-linear curve. The breakpoints—locations where the slope of this curve changes—correspond to shifts in the agent's preferred action as α varies. The left panel (Figure (i)) shows results based on predicted values, while the right panel (Figure (ii)) presents the ground truth. In this example, the agent chooses among three actions. Only the predicted reward for action 3 is slightly inaccurate, with an error below $1 + \epsilon$, while predictions for the other actions are exact. Despite this minor error, the predicted breakpoint between actions 2 and 3 is $\hat{\alpha}_{2,3} = 0.5$, while the true breakpoint is $\alpha_{2,3} = 1$.

Figure 2 further illustrates that even when the predicted breakpoints between action pairs are close to the true ones, the overall structure, such as the number and order of critical breakpoints, may still be incorrect. Similar to Figure 1, this figure compares the agent's utility under predicted versus true values. The agent chooses from three actions, with only a slight prediction error in the reward for action 3 (within a factor of $1 + \epsilon$), while predictions for the other actions are accurate. Despite this high level of accuracy, the prediction-based model identifies only a single critical breakpoint, $\hat{\alpha}_{1,3}$, where the agent switches actions. In contrast, under perfect information, two distinct breakpoints, $\alpha_{1,2}$ and $\alpha_{2,3}$, emerge, with each indicating a shift in the agent's optimal action.

The preceding examples demonstrate that even a slight deviation in the predicted reward of a single action can lead to significant issues, despite overall high prediction accuracy. When small errors are introduced across multiple actions, the likelihood of misidentifying breakpoints or incentivized actions increases substantially, compounding the challenge in contract design.

Algorithm 1 Discrete Search

Require: $\eta \geq 0, R, M \in \mathbb{Z}^+$
for $m = 1, 2, ..., M$ **do**#divide $[0, 1 - \frac{1}{R}]$ into M intervals
  label the m-th boundary point α_m, i.e., $\alpha_m \leftarrow \frac{m}{M}(1 - \frac{1}{R})$
end for
for $m = 1, 2, ..., M$ **do**
  for $i = 1, 2, ..., N$ **do** #compute the η-best response action set $\mathcal{I}_m$ at α_m
    if $\alpha_m \hat{R}_i - \hat{c}_i \geq \max_{j \in [N]} \{\alpha_m \hat{R}_j - \hat{c}_j - \eta\}$ **then** insert i into $\mathcal{I}_m$
    end if
  end for
  compute the value of α_m such that $v_m \leftarrow \min_{i \in \mathcal{I}_m} (1 - \alpha_m)\hat{R}_i$
end for
$v_0 \leftarrow \hat{R}_1$
choose the point with highest value, i.e., $\alpha \leftarrow \alpha_m$, where $m = \arg\max_{m=0,...,M} v_m$
return α

4 Discrete Search

In this section, we introduce our first algorithm, Discrete Search, which discretizes the continuous interval of feasible parameters. A key idea is to construct a candidate set of the agent's best responses and evaluate the pessimistic profit at each boundary point. We show that the true best response always lies within this candidate set. Furthermore, the pessimistic profit at a boundary point serves as a lower bound on the true profit, while that at the next boundary point provides an upper bound.

We now proceed to introduce our algorithm in detail, as presented in Algorithm 1. We first partition the interval $[0, 1 - \frac{1}{R}]$ evenly into M segments, denoting the m-th boundary point as α_m; thus, the m-th interval is $(\alpha_{m-1}, \alpha_m]$. At each boundary point α_m, we calculate a candidate set of actions for the best response. Specifically, we incorporate a tolerance margin by including actions whose utilities exceed a certain threshold. We add an action i into the η-best response set $\mathcal{I}_m$ if $\alpha_m \hat{R}_i - \hat{c}_i \geq \max_{j \in [N]} \{\alpha_m \hat{R}_j - \hat{c}_j - \eta\}$. The actions in this η-best response set $\mathcal{I}_m$ are candidates for the best response. Next, we evaluate each boundary point by applying a pessimistic approach, computing the point's value based on the lowest expected reward (or profit). Specifically, we determine the value of the m-th boundary point α_m as $v_m = \min_{i \in \mathcal{I}_m} (1 - \alpha_m)\hat{R}_i$. Finally, we output the point with the highest value.

Now, we demonstrate the performance of the algorithm, focusing on its consistency and robustness.

Theorem 1 (Approximation Ratio)**.** *Given a prediction error γ satisfying $(\gamma - \frac{1}{\gamma})2R \leq \eta$, the approximation ratio r of Algorithm 1 is at most:*

$$r \leq 1/(1 - \frac{R}{M})^2(\frac{1}{\gamma^2} - \frac{2M\eta}{\gamma}), \tag{1}$$

where $M \in \mathbb{Z}^+$ and $M > R$.

Remark 1 (Consistency)**.** *Theorem 1 establishes the consistency of our algorithm. It shows that, by first selecting a sufficiently large M and then choosing η close to 0, the approximation ratio converges to 1 as the prediction accuracy increases, i.e., as γ approaches 1.*

To demonstrate its robustness, we first prove that no algorithm can guarantee an approximation ratio smaller than R in the worst-case scenario. We then show that our algorithm can achieve an approximation ratio of R, thereby confirming its tight robustness.

Theorem 2 (Impossibility Result)**.** *If the predictions are arbitrarily inaccurate, i.e., $\forall \gamma \in \mathbb{R}^+$, then for any $\epsilon > 0$, no algorithm can achieve an approximation ratio less than $R - \epsilon$.*

Theorem 3 (Robustness)**.** *If the predictions are arbitrarily inaccurate, i.e., $\forall \gamma \in \mathbb{R}^+$, the approximation ratio of Algorithm 1 is at most R.*

While Algorithm 1 achieves both consistency and robustness, guaranteeing consistency requires choosing the discretization parameter $M = \Omega(R)$. Consequently, the running time of Algorithm 1 is $O(MN)$, resulting in pseudo-polynomial dependence on R.

Algorithm 2 Pessimistic Breakpoint Search

Require: $\eta \geq 0, R, \sigma > 1, \delta \geq 1$

for $i = 1, 2, ..., N, j = i + 1, ..., N$ **do** #select action pairs with relatively different rewards
  if $\hat{R}_j - \hat{R}_i > (1 - \frac{2}{1-\sigma})\eta\hat{R}_j$ **then** insert (i, j) into $\mathcal{S}$
  end if
end for
for $(i, j) \in \mathcal{S}$ **do**
  compute the predicted breakpoint of action i and j, i.e., $\hat{\alpha}_{i,j} \leftarrow \frac{\hat{c}_j - \hat{c}_i}{\hat{R}_j - \hat{R}_i}$
  set tolerant breakpoint $\beta_{i,j} \leftarrow \min\{\frac{\hat{\eta}_{i,j} + \frac{\eta}{2R}\hat{c}_j}{\hat{\eta}_{i,j} - \frac{\eta}{2R}\hat{\alpha}_{i,j}\hat{R}_j}\hat{\alpha}_{i,j}, 1 - \frac{1}{R}\}$, where $\hat{\eta}_{i,j} = \hat{c}_j - \hat{c}_i$
  for $k = 1, 2, 3..., N$ **do** #compute the η-best response set of $\beta_{i,j}$
    if $\beta_{i,j}\hat{R}_k - \hat{c}_k \geq \max_{m\in[N]}\{\beta_{i,m}\hat{R}_k - \hat{c}_k - \eta\}$ **then** insert k into $\mathcal{I}_{i,j}$
    end if
  end for
  compute the value of $\beta_{i,j}$ such that $v_{i,j} \leftarrow \min_{k\in\mathcal{I}_{i,j}}(1-\beta_{i,j})\hat{R}_k$ if $\beta_{i,j} \neq 1-\frac{1}{R}$ or $v_{i,j} \leftarrow 1$ if $\beta_{i,j} = 1 - \frac{1}{R}$
  consider one more parameter $\beta^1_{i,j} \leftarrow \min\{1 - \frac{1}{\delta} + \frac{\beta_{i,j}}{\delta}, 1 - \frac{1}{R}\}$
  for $k = 1, 2, 3..., N$ **do** #compute the η-best response set of $\beta^1_{i,j}$
    if $\beta^1_{i,j}\hat{R}_k - \hat{c}_k \geq \max_{m\in[N]}\{\beta^1_{i,m}\hat{R}_k - \hat{c}_k - \eta\}$ **then** insert k into $\mathcal{I}^1_{i,j}$
    end if
  end for
  compute the value of $\beta^1_{i,j}$ such that $v^1_{i,j} \leftarrow \max\{\min_{k\in\mathcal{I}^1_{i,j}}(1 - \beta^1_{i,j})\hat{R}_k, 1\}$ if $\beta^1_{i,j} \neq 1 - \frac{1}{R}$ or $v^1_{i,j} \leftarrow 1$ if $\beta^1_{i,j} = 1 - \frac{1}{R}$
end for
$v_0 \leftarrow \hat{R}_1$
set α to be the parameter among $\beta_{i,j}, \beta^1_{i,j}, 0$ for all $(i, j) \in \mathcal{S}$ that has the highest value
return α

5 Pessimistic Breakpoint Search

In this section, we present our second algorithm, the Pessimistic Breakpoint Search. This algorithm demonstrates desirable consistency and robustness, while also operating with polynomial complexity, making it both efficient and reliable for practical applications. For implementation, action pairs in Algorithm 2 are oriented according to the predicted rewards. That is, a pair (i, j) is considered only when $\hat{R}_i < \hat{R}_j$. The ordering of true rewards, i.e., $R_1 \leq R_2 \leq \cdots \leq R_N$, is only used as a notational convention in the analysis and is not required by the algorithm.

Our approach incorporates an efficient breakpoint search. As discussed in Section 3, the computed breakpoints may vary significantly even under highly accurate predictions, making it challenging to identify the true breakpoints. However, we show that when a computed breakpoint deviates substantially from its true value, the expected rewards for the corresponding action pair remain nearly equivalent (see Lemma 1). This insight allows us to discard such action pairs and focus only on those with noticeably different rewards. To address the challenge of selecting the incentivized action at each breakpoint, we further adopt a candidate set of the agent's best responses and perform a pessimistic profit analysis, similar to the strategy used in Algorithm 1.

We proceed to introduce Algorithm 2 in detail and provide a brief overview of the methodology behind the algorithm along with its introduction.

First, it identifies all action pairs (i, j) such that $\hat{R}_j - \hat{R}_i > (1 - \frac{2}{1-\sigma})\eta\hat{R}_j$ and adds these pairs to the set $\mathcal{S}$. This set comprises action pairs with relatively different rewards, ensuring that the computed breakpoints are close to the true breakpoints. Then, for each pair $(i, j) \in \mathcal{S}$, the algorithm calculates the corresponding breakpoint $\hat{\alpha}_{i,j}$ based on predictions. To account for potential errors, we introduce a tolerance threshold, $\beta_{i,j}$, which is set to ensure that the true breakpoint $\alpha_{i,j}$ falls below $\beta_{i,j}$ in most cases (except when $\alpha_{i,j} \geq 1 - \frac{1}{R}$). This approach implies that, at the tolerant threshold $\beta_{i,j}$, action j usually has a higher utility than action i.

However, the breakpoint $\alpha_{i,j}$ (and its loose estimation $\beta_{i,j}$) only guarantee that action j is better than action i for the agent, but not necessarily the best among all available actions. Thus, it remains unknown which action is agent's best response to this tolerant breakpoint $\beta_{i,j}$. Consequently, this uncertainty prevents us from accurately computing the principal's profit at these tolerant breakpoints. To address this, we adopt a similar approach to Algorithm 1. We construct a candidate set for the best response, including all actions whose utilities exceed a specified threshold. Specifically, we add an action k to the η-best response set $\mathcal{I}_{i,j}$ if it satisfies $\beta_{i,j}\hat{R}_k - \hat{c}_k \geq \max_{j\in[N]}\{\beta_{i,j}\hat{R}_k - \hat{c}_k - \eta\}$. Next, we attempt to bound the profit using values. The value of each tolerant breakpoints is determined based on the lowest expected reward (or profit) and is defined as:

$$v_{i,j} = \begin{cases} \min_{k\in\mathcal{I}_{i,j}}(1-\beta_{i,j})\hat{R}_k & \text{if } \beta_{i,j} \neq 1 - \frac{1}{R}, \\ 1 & \text{if } \beta_{i,j} = 1 - \frac{1}{R}. \end{cases}$$

This value roughly serves as a lower bound on the true profit at this tolerant breakpoint. To select a parameter with high profit for the principal, we also seek to establish an upper bound on the profit at these tolerant breakpoints. For any tolerant breakpoint $\beta_{i,j}$, we consider an additional parameter that is slightly larger than $\beta_{i,j}$, defined as $\beta^1_{i,j} = \min\{1 - \frac{1}{\delta} + \frac{\beta_{i,j}}{\delta}, 1 - \frac{1}{R}\}$. For this new parameter, we similarly determine the η-best response set and corresponding value. The only difference is that we set a lower bound of 1 for the value of the newly added parameter, which is crucial for the proof of Lemma 3. Specifically, we define the η-best response set $\mathcal{I}^1_{i,j}$ for $\beta^1_{i,j}$ to include action k such that $\beta^1_{i,j}\hat{R}_k - \hat{c}_k \geq \max_{j\in[N]}\{\beta^1_{i,j}\hat{R}_k - \hat{c}_k - \eta\}$. The value is defined as:

$$v^1_{i,j} = \begin{cases} \max\{\min_{k\in\mathcal{I}^1_{i,j}}(1-\beta^1_{i,j})\hat{R}_k, 1\} & \text{if } \beta_{i,j} \neq 1 - \frac{1}{R}, \\ 1 & \text{if } \beta_{i,j} = 1 - \frac{1}{R}. \end{cases}$$

The value at $\beta^1_{i,j}$ thus serves as a rough lower bound for the profit at $\beta^1_{i,j}$, and more importantly, it provides a rough upper bound on the true profit for the tolerant breakpoint $\beta_{i,j}$ (see Lemma 3).

Finally, the algorithm sets $v_0 = \hat{R}_1$, representing the value at $\alpha = 0$. It then outputs the parameter corresponding to the highest value among all computed values.

Based on the algorithm's design, we can confirm that it operates in polynomial time.

Observation 1 (Polynomial Complexity)**.** *The complexity of Algorithm 2 is polynomial.*

Next, we evaluate the performance of our algorithm from two perspectives: consistency and robustness. Before proceeding with the proofs for consistency, we present the following lemmas.

The first lemma demonstrates the relationship between the tolerant breakpoints and the true breakpoints. Specifically, for the action pair $(i, j) \in \mathcal{S}$, it establishes that the tolerant breakpoint is slightly larger than the true breakpoint.

Lemma 1. *For any action pair $(i, j) \in \mathcal{S}$, if the prediction error γ satisfies $(\gamma - \frac{1}{\gamma})2R \leq \eta$, then the tolerant breakpoint $\beta_{i,j}$ is slightly larger than the true breakpoint $\alpha_{i,j}$ when $\alpha_{i,j} \leq 1 - \frac{1}{R}$. Specifically, the following inequalities hold:*

$$\alpha_{i,j} \leq \beta_{i,j}, 1 - \alpha_{i,j} \leq \sigma^2(1 - \beta_{i,j}). \tag{2}$$

Let $p_{i,j}$ represent the true profit associated with $\beta_{i,j}$ and $p^1_{i,j}$ represent the true profit associated with $\beta^1_{i,j}$. The next two lemmas establish the relationship between the values and the profits for the tolerant breakpoint $\beta_{i,j}$ and the additional parameter $\beta^1_{i,j}$.

Lemma 2. *Under a prediction error γ satisfying $(\gamma - \frac{1}{\gamma})2R \leq \eta$, the following statements hold:*

(1) The value of every tolerant breakpoint $\beta_{i,j}$ does not exceed γ times the true expected profit if $\beta_{i,j} \neq 1 - \frac{1}{R}$, i.e., $v_{i,j} \leq \gamma p_{i,j}$.

(2) The value of every additional parameter $\beta^1_{i,j}$ does not exceed γ times the true expected profit if $v^1_{i,j} \neq 1$, i.e., $v^1_{i,j} \leq \gamma p^1_{i,j}$.

Lemma 3. *Under a prediction error γ satisfying $(\gamma - \frac{1}{\gamma})2R \leq \eta$, the value of $\beta^1_{i,j}$ is at least $\frac{1}{\delta}(\frac{1}{\gamma} - \frac{2\eta}{1-\frac{1}{\delta}})$ times the true expected profit at $\beta_{i,j}$. Specifically, the following inequality holds:*

$$v^1_{i,j} \geq \frac{1}{\delta}(\frac{1}{\gamma} - \frac{2\eta}{1-\frac{1}{\delta}})p_{i,j}.$$

With the above lemmas established, we are now ready to demonstrate the performance of our algorithm.

Theorem 4 (Approximation Ratio)**.** *Given a prediction error γ satisfying $(\gamma - \frac{1}{\gamma})2R \leq \eta$, the approximation ratio r of Algorithm 2 is at most:*

$$r \leq \frac{\sigma^2\gamma^3\delta}{(1-\eta(1-\frac{2}{1-\sigma}))(\frac{1}{\gamma} - \frac{2\eta}{1-\frac{1}{\delta}})}, \tag{3}$$

where $\sigma, \delta \geq 1$.

Proof Sketch. Let p denote the expected profit achieved by our algorithm, and let p^* denote the optimal profit. Additionally, let v represent the value at the output α. Based on the algorithm design and Lemma 2, we establish that $p \geq \frac{1}{\gamma}v$ always holds. We consider three cases:

(i) For the true optimal critical breakpoint $\alpha_{i,j}$, if $(i,j) \in \mathcal{S}$. Given $(i,j) \in \mathcal{S}$, Lemma 1 and Lemma 3 imply that the profit of $\alpha_{i,j}$ is upper bounded by the value at $\beta^1_{i,j}$, i.e.,$p_{i,j} \leq \frac{1}{\frac{1}{\delta}(\frac{1}{\gamma} - \frac{2\eta}{1-\frac{1}{\delta}})}v^1_{i,j}$. Since the algorithm selects the parameter with the highest value and $p \geq \frac{1}{\gamma}v$, the result follows.

(ii) For the true optimal critical breakpoint $\alpha_{i,j}$, if $(i,j) \notin \mathcal{S}$, and $(1,j) \notin \mathcal{S}$. Thus, the expected profit at $\alpha = 0$ can be used to approximate the optimal profit. Since the algorithm selects the parameter with the highest value and $p \geq \frac{1}{\gamma}v$, the conclusion follows.

(iii) For the true optimal critical breakpoint $\alpha_{i,j}$, if $(i,j) \notin \mathcal{S}$, and $(1,j) \in \mathcal{S}$. Assume the critical breakpoints are ordered as $(i_1 = 1, i_2), (i_2, i_3), ..., (i,j)$ and let i_m be the largest index such that $(i_m, j) \in \mathcal{S}$. We show: $\alpha_{i_m,i} \leq \alpha_{i_m,j} \leq \alpha_{i,j}$. Therefore, the incentivized action at $\beta_{i_m,j}$ yields a reward at least as high as that of action i. Since $(i,j) \notin \mathcal{S}$, we approximate the reward of action j using that of action i, and due to $\alpha_{i_m,j} \leq \alpha_{i,j}$, approximate the true optimal profit by the profit at $\alpha_{i_m,j}$. Finally, since $(i_m, j) \in \mathcal{S}$, we apply Lemma 1 and Lemma 3, as in Case (i), to upper bound the profit at $\alpha_{i_m,j}$ and derive the approximation ratio.

Combining all cases and all analysis, we get the conclusion. □

Remark 2 (Consistency)**.** *Based on the above theorem, we can deduce that by selecting σ and δ close to 1, and then η close to 0, the approximation ratio converges to 1 as the prediction error γ approaches 1. This result establishes the consistency guarantee.*

The robustness analysis is similar to that of Algorithm 2. Specifically, according to Theorem 2, no algorithm can guarantee an approximation ratio strictly less than R. However, our algorithm guarantees an approximation ratio of R, as stated in the following theorem.

Theorem 5 (Robustness)**.** *If the predictions are arbitrarily inaccurate, i.e., $\forall\gamma \in \mathbb{R}^+$, the approximation ratio of Algorithm 2 is at most R.*

6 Conclusions

In this paper, we study contract design with predictions, where the principal uses predicted action costs and rewards to design linear contracts under incomplete information. We develop algorithms that achieve both consistency and robustness guarantees. Our first algorithm discretizes the contract parameter space, while our main algorithm exploits the structure of critical breakpoints to obtain polynomial-time complexity while preserving these guarantees.

Open Questions. This work focuses on linear contracts with predictions of action costs and expected rewards. Extending our framework to richer contract classes and other forms of predictive information remains an interesting direction.

AI USE STATEMENT

In this work, we used generative AI tools to assist in the writing of proofs. We have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean or reformat datasets, support qualitative or thematic data analysis, or interpret research results. The tasks related to dataset generation, experimental design, experimental implementation, and experimental analysis are not applicable to this work, as this paper does not contain an experimental component.

We have reviewed all AI-assisted work and verified the correctness and accuracy of the final manuscript. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Priyank Agrawal, Eric Balkanski, Vasilis Gkatzelis, Tingting Ou, and Xizhi Tan. Learning-augmented mechanism design: Leveraging predictions for facility location. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pp. 497–528, 2022.

Tal Alon, Paul Duetting, Yingkai Li, and Inbal Talgam-Cohen. Bayesian analysis of linear contracts. In *Proceedings of the 24th ACM Conference on Economics and Computation*, pp. 66–66, 2023.

Yossi Azar, Debmalya Panigrahi, and Noam Touitou. Discrete-smoothness in online algorithms with predictions. *Advances in Neural Information Processing Systems*, 36, 2024.

Moshe Babaioff, Michal Feldman, and Noam Nisan. Combinatorial agency. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, pp. 18–28, 2006.

Xingjian Bai and Christian Coester. Sorting with predictions. *Advances in Neural Information Processing Systems*, 36:26563–26584, 2023.

Eric Balkanski, Vasilis Gkatzelis, and Xizhi Tan. Strategyproof scheduling with predictions. *arXiv preprint arXiv:2209.04058*, 2022.

Eric Balkanski, Vasilis Gkatzelis, Xizhi Tan, and Cherlin Zhu. Online mechanism design with predictions. *arXiv preprint arXiv:2310.02879*, 2023.

Zohar Barak, Anupam Gupta, and Inbal Talgam-Cohen. Mac advice for facility location mechanism design. *arXiv preprint arXiv:2403.12181*, 2024.

Magnus Berg and Shahin Kamali. Online bin covering with frequency predictions. In *19th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2024)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024.

Patrick Bolton and Mathias Dewatripont. *Contract theory*. MIT press, 2004.

Federico Cacciamani, Martino Bernasconi, Matteo Castiglioni, and Nicola Gatti. Multi-agent contract design beyond binary actions. *arXiv preprint arXiv:2402.13824*, 2024.

Bernard Caillaud and Benjamin E Hermalin. Hidden-information agency. *Unpublished manuscript*, 2000.

Ioannis Caragiannis and Georgios Kalantzis. Randomized learning-augmented auctions with revenue guarantees. *arXiv preprint arXiv:2401.13384*, 2024.

Gabriel Carroll. Robustness and linear contracts. *American Economic Review*, 105(2):536–563, 2015.

Gabriel Carroll. Robustness in mechanism design and contracting. *Annual Review of Economics*, 11 (1):139–166, 2019.

Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Bayesian agency: Linear versus tractable contracts. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pp. 285–286, 2021.

Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Designing menus of contracts efficiently: The power of randomization. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pp. 705–735, 2022.

Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Multi-agent contract design: How to commission multiple agents with individual outcomes. In *Proceedings of the 24th ACM Conference on Economics and Computation*, pp. 412–448, 2023.

Ramiro Deo-Campo Vuong, Shaddin Dughmi, Neel Patel, and Aditya Prasad. On supermodular contracts and dense subgraphs. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 109–132. SIAM, 2024.

Paul Dütting, Tim Roughgarden, and Inbal Talgam-Cohen. Simple versus optimal contracts. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pp. 369–387, 2019.

Paul Dütting, Tomer Ezra, Michal Feldman, and Thomas Kesselheim. Combinatorial contracts. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 815–826. IEEE, 2022.

Paul Dütting, Tomer Ezra, Michal Feldman, and Thomas Kesselheim. Multi-agent contracts. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pp. 1311–1324, 2023.

Paul Dutting, Michal Feldman, and Yoav Gal Tzur. Combinatorial contracts beyond gross substitutes. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 92–108. SIAM, 2024.

Yuval Emek and Michal Feldman. Computing optimal contracts in combinatorial agencies. *Theoretical Computer Science*, 452:56–74, 2012.

Vasilis Gkatzelis, Kostas Kollias, Alkmini Sgouritsa, and Xizhi Tan. Improved price of anarchy via predictions. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pp. 529–557, 2022.

Sumit Goel and Wade Hann-Caruthers. Optimality of weighted contracts for multi-agent contract design with a budget. *arXiv preprint arXiv:2402.15890*, 2024.

Sreenivas Gollapudi and Debmalya Panigrahi. Online algorithms for rent-or-buy with expert advice. In *International Conference on Machine Learning*, pp. 2319–2327. PMLR, 2019.

Sanford J Grossman and Oliver D Hart. An analysis of the principal-agent problem. In *Foundations of Insurance Economics: Readings in Economics and Finance*, pp. 302–340. Springer, 1992.

Guru Guruganesh, Jon Schneider, and Joshua R Wang. Contracts under moral hazard and adverse selection. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pp. 563–582, 2021.

Guru Guruganesh, Jon Schneider, Joshua Wang, and Junyao Zhao. The power of menus in contract design. In *Proceedings of the 24th ACM Conference on Economics and Computation*, pp. 818–848, 2023.

Bengt Holmström. Moral hazard and observability. *The Bell journal of economics*, pp. 74–91, 1979.

Gabriel Istrate and Cosmin Bonchis. Mechanism design with predictions for obnoxious facility location. *arXiv preprint arXiv:2212.09521*, 2022.

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

Bo Peng and Zhihao Gavin Tang. Optimal robust contract design. In *Proceedings of the 25th ACM Conference on Economics and Computation*, pp. 1294–1294, 2024.

Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ml predictions. *Advances in Neural Information Processing Systems*, 31, 2018.

Dhruv Rohatgi. Near-optimal bounds for online caching with machine learned advice. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1834–1845. SIAM, 2020.

Daniel Walton and Gabriel Carroll. A general framework for robust contracting models. *Econometrica*, 90(5):2129–2159, 2022.

Chenyang Xu and Pinyan Lu. Mechanism design with predictions. *arXiv preprint arXiv:2205.11313*, 2022.

A Related Works

Algorithm with Predictions. The study of algorithms with predictions has received significant attention in recent years, with numerous classic online problems and mechanism design challenges being analyzed within this framework. For a comprehensive overview, interested readers can refer to the survey Mitzenmacher & Vassilvitskii (2022).

In the domain of online algorithms with predictions, numerous studies have explored how machine-learned predictions can enhance algorithmic performance. For instance, Purohit et al. (2018) examines two classical problems—ski rental and non-clairvoyant job scheduling—demonstrating provable performance improvements through predictive methods. Similarly, Gollapudi & Panigrahi (2019) addresses the rent-or-buy problem (also known as ski rental), showing that predictive algorithms outperform those designed for adversarial settings. Bai & Coester (2023) investigates sorting from the perspective of learning-augmented algorithms, leveraging potentially inaccurate predictions to improve efficiency. Rohatgi (2020) presents a caching algorithm that utilizes next-arrival predictions, with each element's arrival accompanied by a forecast of its next appearance. Azar et al. (2024) introduces discrete-smoothness as a general framework to quantify consistency and robustness guarantees, applying it to online covering problems such as facility location and set cover. Berg & Kamali (2024) explores the discrete bin covering problem in a predictive context, where the algorithm accesses a vector predicting the frequency of each item size.

In the context of mechanism design with predictions, Xu & Lu (2022) initiates the exploration of this field, addressing problems such as auction design, frugal path auction, scheduling, and facility location. Agrawal et al. (2022) pioneers learning-augmented mechanism design, focusing specifically on the strategic facility location problem. Gkatzelis et al. (2022) investigates a decentralized setting where the mechanism can only indirectly influence agents' decisions, and studied job scheduling and network creation games. Balkanski et al. (2022) focuses on makespan minimization in strategic scheduling, while Istrate & Bonchis (2022) studies the obnoxious facility location problem, where agents prefer to be far from the facility rather than close to it. Caragiannis & Kalantzis (2024) addresses the fundamental problem of designing a truthful single-item auction, focusing on the challenging objective of extracting a large fraction of the highest agent valuation as revenue. Barak et al. (2024) studies the k-facility location mechanism design problem, where the n agents are strategic and may misreport their locations.

Furthermore, Balkanski et al. (2023) initiates the study of online mechanism design with predictions, integrating the fields of online algorithms with predictions and mechanism design with predictions.

Contract Theory. Contract theory is a foundational area of study within economics, applicable to a wide range of contexts and characterized by a rich historical development. It can be traced back to seminal works such as Holmström (1979) and Grossman & Hart (1992). For those seeking an introduction to the subject, comprehensive resources like Bolton & Dewatripont (2004) and the introductory chapters in Caillaud & Hermalin (2000) provide essential insights into the principles and frameworks of contract theory.

Research in contract theory covers a broad spectrum of topics, with one important focus being the complexity and approximation of contracts. Early works, such as Emek & Feldman (2012); Babaioff et al. (2006), apply complexity and approximation methods to the analysis of contracts. More recent studies explore these aspects from different perspectives. For instance, Dütting et al. (2022); Dutting

et al. (2024); Deo-Campo Vuong et al. (2024) examine combinatorial contracts, where the agent selects a subset of actions. Meanwhile, Dütting et al. (2023); Cacciamani et al. (2024); Castiglioni et al. (2023); Goel & Hann-Caruthers (2024) focus on multi-agent scenarios, where joint actions influence the outcome.

An important stream of research focuses on contract design under imperfect information. A pioneering work Carroll (2015) examines the scenario where the principal knows only a subset of the actions available to the agent—while other unknown actions may exist—and demonstrates the robustness of linear contracts. Carroll (2019) provides an overview of recent theories on robust incentive design when the principal lacks complete knowledge of the environment. Dütting et al. (2019) analyzes linear contracts, showing that they can guarantee worst-case performance in ambiguous settings. Walton & Carroll (2022) studies a case where the principal is uncertain about which actions might be taken in response to the contract and seeks to maximize her worst-case payoff. Some works, such as Peng & Tang (2024), consider randomized contracts in scenarios where the principal has limited information about the agent's available actions. Other studies examine Bayesian models in which the agent may belong to different types. For instance, Castiglioni et al. (2021) analyzes a multi-dimensional private type setting, showing that, compared to optimal contracts, linear contracts may suffer a multiplicative loss. Guruganesh et al. (2021) considers a scenario where only the outcome distribution is private, and all types share the same costs. They provide a (tight) bound on the welfare gap between linear contracts and the optimal outcome. Castiglioni et al. (2022) explores the role of randomization in the Bayesian setting, focusing on designing efficient random contract menus. Guruganesh et al. (2023) examines the relative power of different contract classes in the Bayesian context. Alon et al. (2023) demonstrates that linear contracts become approximately optimal under sufficient uncertainty, as the setting becomes less 'pointmass-like' in the Bayesian model.

A key distinction of our work is that, unlike previous studies, it integrates predictions into the framework of contract theory. To the best of our knowledge, this is the first paper to combine contract theory with predictions.

B Omitted Proofs

B.1 Necessity for Rewards Upper Bound and Lower Bound

Theorem 6. *If there is no lower bound on rewards, no algorithm can guarantee a positive profit when predictions are arbitrarily inaccurate. Consequently, no algorithm can guarantee robustness.*

Proof. Consider an instance with fixed predictions $(\hat{\boldsymbol{R}}, \hat{\boldsymbol{c}})$. Based on the predictions, an algorithm computes a linear factor, denoted by α.

For $0 < \alpha < 1$, since the predictions can be arbitrarily inaccurate, there exists an instance where the first true critical breakpoint slightly exceeds α, leading to an incentivized action with an expected reward of R. Yet, the output α is insufficient to incentivize this sole positive-reward action.

We now construct a specific instance that meets the following conditions:

- Let $R_1 = 0$ and $c_1 = 0$.
- For $i = 2, 3, ..., N-1$, set $R_i = c_i$.
- Set $R_N = R$ and $c_N = (\alpha + y)R$, where $y = \frac{1-\alpha}{R}$ and $R \geq 1$.

In this scenario, the optimal linear factor achievable by the principal is $\alpha^* = \frac{(\alpha+y)R}{R} = \alpha + y \geq \alpha$, thus incentivizing the sole positive-reward action and yielding positive profit.

However, at the algorithm's output point α, the incentivized action is action 1, which has a reward of 0. This results in a profit of 0.

If $\alpha = 0$ or $\alpha = 1$, the principal will still obtain a reward of 0.

By combining all these cases, we conclude that no algorithm can guarantee a positive profit when predictions are arbitrarily inaccurate. Therefore, no algorithm can guarantee robustness. □

Theorem 7. *If there is no upper bound of rewards, for any $Q \geq 1$, no algorithm can achieve the approximation ratio less than Q when the predictions are arbitrarily inaccurate. Consequently, no algorithm can guarantee robustness.*

Proof. Consider an instance with fixed predictions $(\hat{\boldsymbol{R}}, \hat{\boldsymbol{c}})$. Based on the predictions, an algorithm computes a linear factor, denoted by α.

If $\alpha = 1$, the algorithm will secure only zero profit.

However, if $0 < \alpha < 1$, since the predictions can be arbitrarily inaccurate, there exists an instance where the first true breakpoint slightly exceeds α, resulting in an incentivized action with an expected reward of $2Q$. Yet, the output α is insufficient to incentivize this high-reward action.

We now construct a specific instance that meets the following conditions:

- Let $R_1 = 1$ and $c_1 = 0$.
- For $i = 2, 3, ..., N-1$, set $R_i = c_i$.
- Set $R_N = 2Q$ and $c_N = (\alpha + y)(2Q - 1)$, where $y = \frac{(1-\alpha)}{Q}$.

In this scenario, the optimal linear factor achievable by the principal is $\alpha^* = \frac{(\alpha+y)(2Q-1)}{2Q-1} = \alpha + y$, resulting in a corresponding profit of $(1 - \alpha - y)2Q$.

However, the algorithm outputs α, at which the incentivized action is action 1 with a reward of 1, yielding a profit of $1 - \alpha$. Consequently, the approximation ratio is:

$$\frac{(1-\alpha-y)}{(1-\alpha)}2Q = \frac{1-\alpha-\frac{(1-\alpha)}{Q}}{1-\alpha}2Q = \frac{(1-\frac{1}{Q})(1-\alpha)}{1-\alpha}2Q = 2Q - 2 \geq Q.$$

If $\alpha = 0$, the principal will obtain a reward of only R_1. In this case, we can construct an instance where $c_N = \frac{(2Q-1)}{2}$ and $R_N = 2Q$, and similarly, set $R_1 = 1$ and $c_1 = 0$. For $i = 2, 3, ..., N-1$, let $R_i = c_i$. The optimal profit for the principal in this scenario is $(1 - \frac{(2Q-1)}{2(2Q-1)})2Q = Q$. Thus, the approximation ratio is Q.

By combining all these cases, we conclude that no algorithm can guarantee an approximation ratio less than Q when predictions are arbitrarily inaccurate. Therefore, no algorithm can guarantee robustness. □

B.2 Proof of Theorem 1

Proof. The true optimal solution clearly lies within the interval $[0, 1 - \frac{1}{R}]$, as any choice outside this interval would yield a lower profit than selecting the linear parameter to be 0.

Assume the true optimal breakpoint α^* lies within the m-th interval, i.e., $\alpha^* \in (\alpha_{m-1}, \alpha_m]$. If $\alpha^* = 0$, we assume that it lies within the first interval, i.e., $m = 1$. Let the action incentivized by α^* be i^*, yielding an expected profit p^*. At the m-th boundary point α_m, the incentivized action j (which may coincide with i^*) offers a weakly higher reward than the action i^*.

We first demonstrate that action j belongs to the η-best response set.

Let i represent the best response action at α_m based on the predicted rewards and costs. According to the accuracy constraints, we obtain the following inequalities:

$$\alpha_m \hat{R}_i - \hat{c}_i \geq \alpha_m \hat{R}_j - \hat{c}_j \geq \alpha_m R_j - c_j - ((\gamma - 1)\alpha_m \hat{R}_j + (1 - \frac{1}{\gamma})\hat{c}_j) \tag{4}$$

$$\geq \alpha_m R_i - c_i - ((\gamma - 1)\alpha_m \hat{R}_j + (1 - \frac{1}{\gamma})\hat{c}_j) \tag{5}$$

$$\geq \alpha_m \hat{R}_i - \hat{c}_i - ((\gamma - 1)\alpha_m (\hat{R}_j + \hat{R}_i) + (1 - \frac{1}{\gamma})(\hat{c}_j + \hat{c}_i)). \tag{6}$$

Equation 4 holds because $\gamma\hat{R}_j \geq R_j$, which implies $\hat{R}_j - R_j \geq (1-\gamma)\hat{R}_j$. Similarly, we can show that $\hat{c_j} - c_j \leq (1-\frac{1}{\gamma})\hat{c}_j$. Equation 6 can be obtained through a similar process. Equation 5 holds because action j represents the true best response.

Given that $(\gamma-1)\alpha_m(\hat{R}_j + \hat{R}_i) + (1-\frac{1}{\gamma})(\hat{c}_j + \hat{c}_i) \leq (\gamma - \frac{1}{\gamma})2R \leq \eta$, we can infer from the above inequalities that: $\alpha_m\hat{R}_i - \hat{c}_i \geq \max_{j\in[N]}\{\alpha_m\hat{R}_j - \hat{c}_j - \eta\}$. This implies that j belongs to the η-best response set.

Let p_m denote the true profit at the m-th boundary point, α_m and let $x = \frac{1}{M}$.

We then demonstrate that:

(a) the value of the m-th boundary point does not exceed γ times the true expected profit, i.e., $v_m \leq \gamma p_m$;

(b) the value of the next boundary point is at least $(1-xR)(\frac{1}{\gamma} - \frac{2\eta}{x})$ times the true expected profit at this boundary point if $m \neq M$, i.e., $v_{m+1} \geq (1-xR)(\frac{1}{\gamma} - \frac{2\eta}{x})p_m$.

To prove statement (a), we rely only on the accuracy constraint and the algorithm's design. We know that the true incentivized action is included within the η-best response set, and the algorithm calculates the value based on the lowest expected reward within this set. Consequently, we have:

$$v_m = \min_{i\in\mathcal{I}_m}(1-\alpha_m)\hat{R}_i \leq (1-\alpha_m)\hat{R}_j \leq (1-\alpha_m)\gamma R_j = \gamma p_m.$$

Thus, we can conclude that $v_m \leq \gamma p_m$, as required.

To prove statement (b), let $\mathcal{J}$ denote the set of actions such that $\hat{R}_j - \hat{R}_i \geq \frac{2\eta}{x}$ for actions i within the η-best response set. Assume k is the largest action such that $k \notin \mathcal{J}$(where k may equal j).

For any action i in $\mathcal{J}$, when calculating the next η-best response set, we have:

$$\begin{aligned}
\alpha_{m+1}\hat{R}_i - \hat{c}_i &= \alpha_m\hat{R}_i - \hat{c}_i + x\hat{R}_i && (7)\\
&\leq \alpha_m\hat{R}_j - \hat{c_j} + \eta + x\hat{R}_i && (8)\\
&= \alpha_{m+1}\hat{R}_j - \hat{c}_j + \eta + x(\hat{R}_i - \hat{R}_j) && (9)\\
&< \alpha_{m+1}\hat{R}_j - \hat{c}_j + \eta - 2\eta && (10)\\
&= \alpha_{m+1}\hat{R}_j - \hat{c}_j - \eta. && (11)
\end{aligned}$$

Equation 8 holds because both actions i and j are included in the η-best response set at the m-th boundary point, α_m. The remaining inequalities can be derived through straightforward computations.

These above inequalities implies that all actions like $i \in \mathcal{J}$ will not be included in the next η-best response set.

Therefore, the value of the next boundary point is at least $(1-\alpha_{m+1})\hat{R}_k$. Moreover, since $k \notin \mathcal{J}$, we can acquire that: $\hat{R}_k \geq \hat{R}_j - \frac{2\eta}{x} \geq \frac{1}{\gamma}R_j - \frac{2\eta}{x}$. Thus, the factor f between the value of the next boundary point and the true expected profit at this boundary point is at least:

$$f \geq \frac{(1-\alpha_{m+1})\hat{R}_k}{(1-\alpha_m)R_j} \geq (1-xR)(\frac{1}{\gamma} - \frac{2\eta}{x}). \tag{12}$$

This inequality holds because $1-\alpha_m \geq \frac{1}{R}$ according to the algorithm's design, and given that the reward for each action is at least 1. Consequently, we arrive at statement (b).

Suppose our algorithm outputs the point α with profit p and value v. We can conclude that $p \geq \frac{1}{\gamma}v$, similar to the proof of statement (a).

If $m = M$, then we have the following inequalities: $p^* \leq (1-\alpha_{M-1})R \leq 1+xR$ and $p \geq \frac{1}{\gamma}v_0 \geq \frac{1}{\gamma}$. Consequently, the approximation ratio is at most $\gamma(1+xR)$.

If $m \neq M$, we have the following chain of inequalities:

$$p \geq \frac{1}{\gamma} v \geq \frac{1}{\gamma} v_{m+1} \tag{13}$$

$$\geq (1 - xR)(\frac{1}{\gamma^2} - \frac{2\eta}{x\gamma}) p_m \tag{14}$$

$$\geq (1 - xR)(\frac{1}{\gamma^2} - \frac{2\eta}{x\gamma}) \frac{1 - \alpha_m}{1 - \alpha} p^* \tag{15}$$

$$\geq (1 - xR)^2 (\frac{1}{\gamma^2} - \frac{2\eta}{x\gamma}) p^*. \tag{16}$$

Equation 13 holds because our algorithm outputs the point with the highest value. Equation 14 is justified by statement (b). Equation 15 holds since the true optimal parameter lies within the m-th interval, and Equation 16 follows from straightforward computation.

Combining all of these results above, we can find that the approximation ratio is at most $1/(1 - xR)^2(\frac{1}{\gamma^2} - \frac{2\eta}{x\gamma})$. □

B.3 Proof of Theorem 2

Proof. Consider an instance with fixed predictions $(\hat{\boldsymbol{R}}, \hat{\boldsymbol{c}})$. Based on the predictions, an algorithm computes a linear factor, say α.

If $\alpha = 1$, the algorithm will only secure zero profit.

However, if $0 < \alpha < 1$, since the predictions can be arbitrarily inaccurate, there exists an instance where the first true critical breakpoint slightly exceeds α, resulting in an incentivized action with an expected reward of R. Yet, the output α is insufficient to incentivize this high-reward action.

We now construct a specific instance that meets the following conditions:

- Let $R_1 = 1$ and $c_1 = 0$.
- For $i = 2, 3, ..., N - 1$, set $R_i = c_i$.
- Set $R_N = R$ and $c_N = (\alpha + y)(R - 1)$, where $y = \frac{\epsilon(1-\alpha)}{R}$.

In this scenario, the optimal linear factor achievable by the principal is $\alpha^* = \frac{(\alpha+y)(R-1)}{R-1} = \alpha + y$, resulting in a corresponding profit of $(1 - \alpha - y)R$.

However, the algorithm outputs α, at which the incentivized action is action 1 with a reward of 1. This yields a profit of $(1 - \alpha)$. Consequently, the approximation ratio is:

$$\frac{(1 - \alpha - y)}{(1 - \alpha)} R = \frac{1 - \alpha - \frac{\epsilon(1-\alpha)}{R}}{1 - \alpha} R = \frac{(1 - \frac{\epsilon}{R})(1 - \alpha)}{1 - \alpha} R = R - \epsilon.$$

If $\alpha = 0$, the principal will obtain a reward of only R_1. In this case, we can construct an instance where $c_N = \frac{\epsilon(R-1)}{R}$ and $R_N = R$, and similarly, set $R_1 = 1$ and $c_1 = 0$. For $i = 2, 3, ..., N - 1$, let $R_i = c_i$. The optimal profit for the principal in this scenario is $(1 - \frac{\epsilon(R-1)}{R(R-1)})R = R - \epsilon$. Thus, the approximation ratio is $R - \epsilon$.

Combining all of these cases, we conclude that no algorithm can guarantee an approximation ratio less than $R - \epsilon$ when predictions are arbitrarily inaccurate. □

B.4 Proof of Theorem 3

Proof. Suppose α^* is the optimal linear factor based on the true rewards and costs and α is the linear factor computed by Algorithm 1. Denote the approximation ratio by r.

We analyze three cases:

(i) $\alpha < \alpha^*$:

Given the upper bound on the expected reward, the expected profit for the optimal linear contract satisfies the inequality $p^* \leq (1-\alpha^*)R$. Similarly, due to the lower bound on the expected reward, the profit at α is at least $1-\alpha$. Consequently, the approximation ratio can be expressed as:

$$r \leq \frac{(1-\alpha^*)R}{(1-\alpha)} \leq R.$$

(ii) $\alpha = \alpha^*$:

We have $r = 1$, meaning the original statement holds trivially.

(iii) $\alpha > \alpha^*$:

Suppose that at α^*, the incentivized action yields a reward of R^*, and at α, the incentivized action has a reward of R'. The action incentivized by α yields a reward that is weakly greater than the reward of the action incentivized by α^*, i.e., $R' \geq R^*$ Therefore, the following inequalities hold:

$$r \leq \frac{(1-\alpha^*)R^*}{(1-\alpha)R'} \leq \frac{1-\alpha^*}{1-\alpha} \leq \frac{1}{1-\alpha}.$$

According to the construction of our algorithm, it select parameter α within the interval $[0, 1-\frac{1}{R}]$. Consequently, we have $r \leq \frac{1}{1-\alpha} \leq R$, which confirms that the original statement is satisfied.

By combining all cases, we have proven the original statement. □

B.5 Proof of Lemma 1

Proof. For $i < j, i, j \in [N]$, let $\eta_{i,j} = c_j - c_i, w_{i,j} = R_j - R_i$, where $\eta_{i,j}$ represents the true cost difference between actions i and j, and $w_{i,j}$ represents the true reward difference between these actions. Similarly, let $\hat{\eta}_{i,j} = \hat{c}_j - \hat{c}_i, \hat{w}_{i,j} = \hat{R}_j - \hat{R}_i$, representing the predicted differences.

With an accuracy factor γ, we have the following bound:

$$|\eta_{i,j} - \hat{\eta}_{i,j}| \leq |\gamma c_j - c_j| + |c_i - \frac{1}{\gamma} c_i| \leq (\gamma - \frac{1}{\gamma}) \min\{\hat{c}_j, c_j\} \leq \frac{\eta}{2R} \min\{\hat{c}_j, c_j\}. \tag{17}$$

Similarly, we can conclude that $|w_{i,j} - \hat{w}_{i,j}| \leq \frac{\eta}{2R} \min\{\hat{R}_j, R_j\}$.

Let $\alpha_{i,j} = \frac{\eta_{i,j}}{w_{i,j}}$ represent the true breakpoint. By simple computation using the above bounds, the estimated $\hat{\alpha}_{i,j} = \frac{\hat{\eta}_{i,j}}{\hat{w}_{i,j}}$ and the true breakpoint satisfy the following condition:

$$n\hat{\alpha}_{i,j} \leq \alpha_{i,j} \leq z\hat{\alpha}_{i,j}, \tag{18}$$

where $z \leq \frac{\hat{\eta}_{i,j} + \frac{\eta}{2R}\hat{c}_j}{\hat{\eta}_{i,j} - \hat{\alpha}_{i,j}\frac{\eta}{2R}\hat{R}_j}$,$n \geq \frac{\hat{\eta}_{i,j} - \frac{\eta}{2R}\hat{c}_j}{\hat{\eta}_{i,j} + \hat{\alpha}_{i,j}\frac{\eta}{2R}\hat{R}_j}$.

Consequently, we know that $\alpha_{i,j} \leq \beta_{i,j}$ if $\alpha_{i,j} \leq 1 - \frac{1}{R}$, based on the design of the algorithm.

Now we proceed to prove that $1 - \alpha_{i,j} \leq \sigma^2(1 - \beta_{i,j})$.

We separate into two cases:

(i) If $\hat{\alpha}_{i,j} \leq 1 - \frac{1}{R}$:

Using the previous results (Equation 18) along with the selection of $\beta_{i,j}$, we acquire that:

$$\frac{1-\alpha_{i,j}}{1-\beta_{i,j}} \leq \frac{1-n\hat{\alpha}_{i,j}}{1-z\hat{\alpha}_{i,j}} \leq \frac{(1-\hat{\alpha}_{i,j})\hat{w}_{i,j} + \frac{\eta}{2R}(\hat{R}_j + \hat{c}_j)}{(1-\hat{\alpha}_{i,j})\hat{w}_{i,j} - \frac{\eta}{2R}(\hat{R}_j + \hat{c}_j)} \cdot \frac{\hat{w}_{i,j} + \frac{\eta}{2R}\hat{R}_j}{\hat{w}_{i,j} - \frac{\eta}{2R}\hat{R}_j}. \tag{19}$$

Since $(i,j) \in \mathcal{S}$, we have $\hat{w}_{i,j} \geq (1-\frac{2}{1-\sigma})\eta\hat{R}_j$. Then by direct computation, we have: $\frac{\hat{w}_{i,j}+\frac{\eta}{2R}\hat{R}_j}{\hat{w}_{i,j}-\frac{\eta}{2R}\hat{R}_j} \leq \frac{(1-\hat{\alpha}_{i,j})\hat{w}_{i,j}+\frac{\eta}{2R}(\hat{R}_j+\hat{c}_j)}{(1-\hat{\alpha}_{i,j})\hat{w}_{i,j}-\frac{\eta}{2R}(\hat{R}_j+\hat{c}_j)}$. The following chain of inequalities holds by simple computation:

$$\frac{(1-\hat{\alpha}_{i,j})\hat{w}_{i,j}+\frac{\eta}{2R}(\hat{R}_j+\hat{c}_j)}{(1-\hat{\alpha}_{i,j})\hat{w}_{i,j}-\frac{\eta}{2R}(\hat{R}_j+\hat{c}_j)} \leq \frac{(1-\hat{\alpha}_{i,j})(1-\frac{2}{1-\sigma})\eta\hat{R}_j+\frac{\eta}{2R}(\hat{R}_j+\hat{c}_j)}{(1-\hat{\alpha}_{i,j})(1-\frac{2}{1-\sigma})\eta\hat{R}_j-\frac{\eta}{2R}(\hat{R}_j+\hat{c}_j)} \tag{20}$$

$$\leq \frac{2(1-\frac{2}{1-\sigma})\frac{\eta}{2R}\hat{R}_j+2\frac{\eta}{2R}\hat{R}_j}{2(1-\frac{2}{1-\sigma})\frac{\eta}{2R}\hat{R}_j-2\frac{\eta}{2R}\hat{R}_j} \leq \sigma. \tag{21}$$

Combining these results, we conclude that $1-\alpha_{i,j} \leq \sigma^2(1-\beta_{i,j})$.

(ii) If $\hat{\alpha}_{i,j} > 1-\frac{1}{R}$:

Then $\beta_{i,j} = 1-\frac{1}{R}$ according to the design of the algorithm. Consequently, we have the following chain of inequalities:

$$\frac{1-\alpha_{i,j}}{1-\beta_{i,j}} \leq \frac{1-n\hat{\alpha}_{i,j}}{1-\beta_{i,j}} \leq R(\frac{(1-\hat{\alpha}_{i,j})\hat{w}_{i,j}+\frac{\eta}{2R}(\hat{R}_j+\hat{c}_j)}{\hat{w}_{i,j}+\frac{\eta}{2R}\hat{R}_j}) \tag{22}$$

$$\leq \frac{2(1-\frac{2}{1-\sigma})\frac{\eta}{2R}\hat{R}_j+2\frac{\eta}{2R}\hat{R}_j}{2(1-\frac{2}{1-\sigma})\frac{\eta}{2R}\hat{R}_j} \leq \sigma. \tag{23}$$

Equation 22 holds due to Equtation 18. Equation 23 is valid because $(i,j) \in \mathcal{S}$, which implies $\hat{w}_{i,j} \geq (1-\frac{2}{1-\sigma})\eta\hat{R}_j$, along with appropriate scaling. Thus, we can also derive the original statement that $1-\alpha_{i,j} \leq \sigma^2(1-\beta_{i,j})$.

Combining all these results above, we arrive at the desired conclusion. □

B.6 Proof of Lemma 2

Proof. The proof follows a similar structure to that of Theorem 1. Specifically, we first demonstrate that the true best response action is included in the η-best response set. Then, except for the specific case we exclude, the value of the parameter is chosen based on a pessimistic assessment, leading us to the conclusion.

We first show that the true best-response action is contained in the corresponding η-best response set. Then, the desired inequalities follow from the pessimistic construction of the values.

For any tolerant breakpoint $\beta_{i,j}$, let k denote the true best-response action under $\beta_{i,j}$. By the definition of η-best response set, it suffices to show that

$$\beta_{i,j}\hat{R}_k - \hat{c}_k \geq \max_{l\in[N]}\{\beta_{i,j}\hat{R}_l - \hat{c}_l - \eta\}.$$

Let l be the predicted best-response action at $\beta_{i,j}$, i.e.,

$$\beta_{i,j}\hat{R}_l - \hat{c}_l \geq \beta_{i,j}\hat{R}_k - \hat{c}_k.$$

Using the prediction error assumption, we have

$$\hat{R}_l - R_l \leq (\gamma-1)\hat{R}_l,$$

and

$$\hat{c}_l - c_l \leq (1-\frac{1}{\gamma})\hat{c}_l.$$

Similarly, the errors for action k satisfy the same bounds. Therefore,

$$(\beta_{i,j}\hat{R}_l - \hat{c}_l) - (\beta_{i,j}\hat{R}_k - \hat{c}_k)$$
$$\leq (\gamma - \frac{1}{\gamma})2R.$$

Hence, the true best-response action belongs to the η-best response set used by the algorithm.

If $\beta_{i,j} \neq 1 - \frac{1}{R}$, by definition,

$$v_{i,j} = \min_{k \in I_{i,j}} (1 - \beta_{i,j}) \hat{R}_k.$$

Since the true best-response action is included in $I_{i,j}$,

$$\begin{aligned} v_{i,j} &\leq (1 - \beta_{i,j}) \hat{R}_k \\ &\leq \gamma (1 - \beta_{i,j}) R_k \\ &= \gamma p_{i,j}. \end{aligned}$$

This proves statement (1).

The proof of statement (2) follows the same argument. Consider the additional parameter $\beta^1_{i,j}$. The corresponding η-best response set is constructed in the same way, and the above argument shows that the true best-response action under $\beta^1_{i,j}$ is also included in $I^1_{i,j}$. Therefore, if $v^1_{i,j} \neq 1$,

$$\begin{aligned} v^1_{i,j} &= \min_{k \in I^1_{i,j}} (1 - \beta^1_{i,j}) \hat{R}_k \\ &\leq (1 - \beta^1_{i,j}) \hat{R}_k \\ &\leq \gamma (1 - \beta^1_{i,j}) R_k \\ &= \gamma p^1_{i,j}. \end{aligned}$$

Therefore, statement (2) also holds. □

B.7 Proof of Lemma 3

Proof. To prove our assertion, we first focus on the case where $\beta^1_{i,j} = 1 - \frac{1}{R}$. In this specific case, we have the inequality: $1 - \beta_{i,j} \leq \frac{\delta}{R}$. Under this condition, the true profit at $\beta_{i,j}$ is at most δ. However, we observe that the value of $\beta^1_{i,j}$ equals to 1. Therefore, the condition indeed holds trivially.

When $p_{i,j} < 1$, the statement holds trivially since $v^1_{i,j} \geq 1$.

If $\beta^1_{i,j} \leq 1 - \frac{1}{R}$ and $p_{i,j} \geq 1$, we obtain $\beta^1_{i,j} - \beta_{i,j} = (1 - \frac{1}{\delta})(1 - \beta_{i,j})$. Let action l be the true best response at $\beta_{i,j}$. Define $x = (1 - \frac{1}{\delta})(1 - \beta_{i,j})$ and let $\mathcal{J}$ represent the set of actions for which $\hat{R}_l - \hat{R}_a \geq \frac{2\eta}{x}$ for actions a within the η-best response set. Assume k is the largest action such that $k \notin \mathcal{J}$(where k may equal l).

Following a similar approach as outlined in Theorem 1, we find that the value of $\beta^1_{i,j}$ is at least $(1 - \beta^1_{i,j}) \hat{R}_k$. Moreover, since $k \notin \mathcal{J}$, we can acquire that:

$$\hat{R}_k \geq \hat{R}_l - \frac{2\eta}{x} \geq \frac{1}{\gamma} R_l - \frac{2\eta}{x}. \tag{24}$$

Thus, the factor f between the value at $\beta^1_{i,j}$ and the true expected profit at $\beta_{i,j}$ satisfies:

$$f \geq \frac{(1 - \beta^1_{i,j}) \hat{R}_k}{(1 - \beta_{i,j}) R_l} \geq \frac{(1 - \beta_{i,j})(\frac{1}{\gamma} R_l - \frac{2\eta}{(1 - \frac{1}{\delta})(1 - \beta_{i,j})})}{\delta (1 - \beta_{i,j})(R_l)} \tag{25}$$

$$\geq \frac{1}{\delta} (\frac{1}{\gamma} - \frac{2\eta}{(1 - \frac{1}{\delta})(1 - \beta_{i,j}) R_l}) \tag{26}$$

$$\geq \frac{1}{\delta} (\frac{1}{\gamma} - \frac{2\eta}{1 - \frac{1}{\delta}}). \tag{27}$$

Equation 25 follows from Equation 24 and by the fact that $\delta(1 - \beta^1_{i,j}) = 1 - \beta_{i,j}$. Equation 27 holds since $p_{i,j} = (1 - \beta_{i,j}) R_l \geq 1$.

Combining all these results, we reach the conclusion. □

B.8 PROOF OF THEOREM 4

Proof. Let p denote the expected profit achieved by our algorithm, and let p^* denote the optimal profit. Additionally, let v represent the value at the output α. The true optimal solution, defined as the breakpoint $\alpha_{i,j}$ between action i and action j, clearly lies within the interval $[0, 1 - \frac{1}{R}]$. Therefore, it trivially follows that $\alpha_{i,j} \leq 1 - \frac{1}{R}$ in the subsequent proofs.

Since $v_0 = \hat{R}_1 \geq 1$, the output value is either strictly greater than 1, or the algorithm outputs $\alpha = 0$, ensuring that $p \geq \frac{1}{\gamma}\hat{R}_1 = \frac{1}{\gamma}v$. Therefore, based on Lemma 2, it follows that $p \geq \frac{1}{\gamma}v$ always holds.

We proceed by examining different cases:

(i) For the true optimal critical breakpoint $\alpha_{i,j}$, if $(i, j) \in \mathcal{S}$.

Since $(i, j) \in \mathcal{S}$ and considering that $\alpha_{i,j} \leq 1 - \frac{1}{R}$ along with Lemma 1, we can deduce the following: $1 - \alpha_{i,j} \leq \sigma^2(1 - \beta_{i,j})$ and $\beta_{i,j} \geq \alpha_{i,j}$. Given that $\beta_{i,j} \geq \alpha_{i,j}$, the incentivized action at $\beta_{i,j}$ has a weakly higher reward than action j. Therefore, we have: $(1 - \beta_{i,j})R_j \leq p_{i,j}$.

Then, according to Lemma 3, we have: $p_{i,j} \leq \frac{1}{\frac{1}{\delta}(\frac{1}{\gamma} - \frac{2\eta}{1-\frac{1}{\delta}})} v^1_{i,j}$.

By combining these results, and considering the structure of our algorithm, we can conclude that:

$$p^* = (1 - \alpha_{i,j})R_j \leq \sigma^2(1 - \beta_{i,j})R_j \tag{28}$$

$$\leq \sigma^2 p_{i,j} \leq \frac{\sigma^2\delta}{(\frac{1}{\gamma} - \frac{2\eta}{1-\frac{1}{\delta}})} v^1_{i,j} \tag{29}$$

$$\leq \frac{\sigma^2\delta}{(\frac{1}{\gamma} - \frac{2\eta}{1-\frac{1}{\delta}})} v \tag{30}$$

$$\leq \frac{\gamma\sigma^2\delta}{(\frac{1}{\gamma} - \frac{2\eta}{1-\frac{1}{\delta}})} p. \tag{31}$$

Equation 30 holds because our algorithm is designed to select the parameter that maximizes the value. Additionally, Equation 31 is valid since $p \geq \frac{1}{\gamma}v$.

(ii) For the true optimal critical breakpoint $\alpha_{i,j}$, if $(i, j) \notin \mathcal{S}$, and $(1, j) \notin \mathcal{S}$.

Since $(1, j) \notin \mathcal{S}$, we have that: $\hat{w}_{1,j} = \hat{R}_j - \hat{R}_1 \leq \eta(1 - \frac{2}{1-\sigma})\hat{R}_j$. From this, it follows that: $\hat{R}_j \leq \frac{1}{1-(1-\frac{2}{1-\sigma})\eta}\hat{R}_1$.

We select the parameter such that $v \geq \hat{R}_1$. Consequently, the true expected profit is at least $p \geq \frac{1}{\gamma}\hat{R}_1$. The approximation ratio r then satisfies:

$$r \leq \frac{R_j}{p} \leq \frac{\gamma\hat{R}_j}{\frac{1}{\gamma}\hat{R}_1} \leq \frac{\gamma^2}{1 - \eta(1 - \frac{2}{1-\sigma})}. \tag{32}$$

(iii) For the true optimal critical breakpoint $\alpha_{i,j}$, if $(i, j) \notin \mathcal{S}$, and $(1, j) \in \mathcal{S}$.

Suppose the true critical breakpoints are ordered as $(i_1 = 1, i_2), (i_2, i_3), ..., (i, j)$. Furthermore, let i_m be the largest index such that $(i_m, j) \in \mathcal{S}$. Since $(i, j) \notin \mathcal{S}$, we have $\hat{w}_{i,j} = \hat{R}_j - \hat{R}_i \leq \eta(1 - \frac{2}{1-\sigma})\hat{R}_j$. This implies that:

$$\hat{R}_j \leq \frac{1}{1 - \eta(1 - \frac{2}{1-\sigma})}\hat{R}_i. \tag{33}$$

We will first show that for actions i_m, i and j, the breakpoints satisfy $\alpha_{i_m,i} \leq \alpha_{i_m,j} \leq \alpha_{i,j}$. This holds because:

$$\alpha_{i,j}R_i - c_i = \alpha_{i,j}R_j - c_j \geq \alpha_{i,j}R_{i_m} - c_{i_m} \tag{34}$$

$$\alpha_{i_m,i}R_{i_m} - c_{i_m} = \alpha_{i_m,i}R_i - c_i \geq \alpha_{i_m,i}R_j - c_j \tag{35}$$

$$\alpha_{i_m,j}R_{i_m} - c_{i_m} = \alpha_{i_m,j}R_j - c_j \tag{36}$$

Using Equation 34, Equation 36, and the fact that $R_{i_m} \leq R_j$, we can deduce that $\alpha_{i_m,j} \leq \alpha_{i,j}$. Similarly, from Equation 35, Equation 36, and again using the fact that $R_{i_m} \leq R_j$,we can establish that $\alpha_{i_m,j} \geq \alpha_{i_m,i}$. Consequently, the original statement holds as intended.

According to the design of the algorithm, we have:

$$p \geq \frac{1}{\gamma} v^1_{i_m,j} \tag{37}$$

$$\geq \frac{1}{\gamma\delta}(\frac{1}{\gamma} - \frac{2\eta}{1-\frac{1}{\delta}})p_{i_m,j} \tag{38}$$

$$\geq \frac{1}{\gamma\delta}(\frac{1}{\gamma} - \frac{2\eta}{1-\frac{1}{\delta}})(1-\beta_{i_m,j})R_i \tag{39}$$

$$\geq \frac{1}{\gamma\sigma^2\delta}(\frac{1}{\gamma} - \frac{2\eta}{1-\frac{1}{\delta}})(1-\alpha_{i_m,j})R_i. \tag{40}$$

Equation 37 holds due to the inequality $p \geq \frac{1}{\gamma}v$ and the fact that Algorithm 2 output the parameter with highest value. Equation 38 follows from Lemma 3. Equation 39 holds because $\alpha_{i_m,i} \leq \alpha_{i_m,j}$, $\alpha_{i_m,j} \leq \alpha_{i,j} \leq 1-\frac{1}{R}$, which implies that $\alpha_{i_m,i} \leq \alpha_{i_m,j} \leq \beta_{i_m,j}$. This indicates that the incentivized action at $\beta_{i_m,j}$ provides a reward that is weakly higher than that of action i. Additionally, Equation 40 is valid because $(i_m, j) \in \mathcal{S}$ and $\alpha_{i,j} \leq 1 - \frac{1}{R}$, as established by Lemma 1.

Thus, the approximation ratio r satisfies:

$$r \leq \frac{(1-\alpha_{i,j})R_j}{p} \leq \frac{\gamma\sigma^2\delta}{(\frac{1}{\gamma} - \frac{2\eta}{1-\frac{1}{\delta}})} \frac{(1-\alpha_{i,j})R_j}{(1-\alpha_{i_m,j})R_i} \tag{41}$$

$$\leq \frac{\sigma^2\gamma^3\delta}{(1-\eta(1-\frac{2}{1-\sigma}))(\frac{1}{\gamma} - \frac{2\eta}{1-\frac{1}{\delta}})}. \tag{42}$$

Equation 41 holds based on Equation 40. Additionally, Equation 42 is valid due to Equation 33, the fact that $\alpha_{i_m,i} \leq \alpha_{i,j}$, and the accuracy constraint.

Combing all cases and all analysis, we get the conclusion. □

B.9 Proof of Theorem 5

Proof of Theorem 5. Suppose α^* is the optimal linear factor based on the true rewards and costs and α is the linear factor computed by Algorithm 2. Denote the approximation ratio by r. We consider the case where $\alpha > \alpha^*$:

Suppose that at α^*, the incentivized action yields a reward of R^*, and at α, the incentivized action has a reward of R'. Following a similar approach to the proof in Theorem 3, we have: $r \leq \frac{(1-\alpha^*)R^*}{(1-\alpha)R'} \leq \frac{1-\alpha^*}{1-\alpha} \leq \frac{1}{1-\alpha}$. According to the algorithm's construction, it also selects the parameter α within the interval $[0, 1-\frac{1}{R}]$. Consequently, we have $r \leq \frac{1}{1-\alpha} \leq R$. The other cases follow straightforwardly from the proof in Theorem 3. □