

An Efficient ε -BIC to BIC Transformation and Its Application to Black-Box Reduction in Revenue Maximization

Yang Cai*
Yale University, USA
yang.cai@yale.edu

Argyris Oikonomou
Yale University, USA
argyris.oikonomou@yale.edu

Grigoris Velegkas
Yale University, USA
grigoris.velegkas@yale.edu

Mingfei Zhao
Yale University, USA
mingfei.zhao@yale.edu

November 6, 2020

Abstract

We consider the black-box reduction from multi-dimensional revenue maximization to virtual welfare maximization. Cai et al. [12, 13, 14, 15] show a polynomial-time approximation-preserving reduction, however, the mechanism produced by their reduction is only approximately Bayesian incentive compatible (ε -BIC). We provide two new **polynomial time** transformations that convert any ε -BIC mechanism to an exactly BIC mechanism with only a negligible revenue loss.

- Our first transformation applies to any mechanism design setting with *downward-closed outcome space* and only requires *sample access* to the agents' type distributions.
- Our second transformation applies to the *fully general outcome space*, removing the downward-closed assumption, but requires *full access* to the agents' type distributions.

Both transformations only require query access to the original ε -BIC mechanism. Other ε -BIC to BIC transformations for revenue exist in the literature [23, 38, 18] but all require exponential time to run in both of the settings we consider. As an application of our transformations, we improve the reduction by Cai et al. [12, 13, 14, 15] to generate an exactly BIC mechanism.

*Supported by a Sloan Foundation Research Fellowship and the NSF Award CCF-1942583 (CAREER).

1 Introduction

Mechanism design is the study of optimization algorithms with the additional constraint of incentive compatibility. A central theme of algorithmic mechanism design is thus to understand how much this extra constraint hinders our ability to optimize a certain objective efficiently. In the best scenario, one may hope to establish an equivalence between a mechanism design problem and an algorithm design problem, manifested via a *black-box reduction* that converts any algorithm to an incentive compatible mechanism. In this paper, we study the black-box reduction of a central problem in mechanism design: *multi-dimensional revenue maximization*.

The problem description is simple: an auctioneer is selling a collection of items to one or more strategic bidders. We follow the standard Bayesian assumption, that is, each bidder’s type is drawn independently from a distribution known to all other bidders and the auctioneer. The auctioneer’s goal is to design a *Bayesian incentive compatible* (BIC) mechanism that maximizes the *expected revenue*.

In the special case of single-item auction, Myerson provides an elegant characterization of the optimal mechanism. Indeed, Myerson’s solution can be viewed as a black-box reduction from revenue maximization to the algorithmic problem of (virtual) welfare maximization [34]. However, whether the black-box reduction can be extended to multi-dimensional settings remained open after Myerson’s result. Recently, a line of work by Cai et al. [12, 13, 14, 15] showed that there is a *polynomial-time approximation-preserving black-box reduction from multi-dimensional revenue maximization to the algorithmic question of (virtual) welfare optimization*. However, this result still has the following two caveats: (i) the revenue of the mechanism is only guaranteed to be within an additive ε of the optimum; and (ii) the mechanism is only *approximately* Bayesian incentive compatible. Thus, an immediate open problem following their result is whether these two compromises are inevitable. In this paper, we show that approximately Bayesian incentive compatibility is unnecessary through our first main result:

Result I: There is a polynomial-time approximation-preserving black-box reduction from multi-dimensional revenue maximization to the algorithmic question of (virtual) welfare optimization that generates an **exactly Bayesian incentive compatible** mechanism.

Result I is enabled by a **new polynomial time ε -BIC to BIC transformation for revenue**, which is our second main result:

Result II: There is a polynomial-time ε -BIC to BIC transformation that converts any approximately Bayesian incentive compatible mechanism to an exactly Bayesian incentive compatible mechanism with a negligible revenue loss for any downward-closed environment with only *sample access* to the agents’ type distributions.

The transformation is applicable to any downward-closed mechanism design setting.¹ We believe the transformation is of independent interest and would have numerous applications in mechanism design. Indeed, our black-box reduction follows straightforwardly from applying the transformation to the mechanism of Cai et al. [12, 13, 14, 15]. Furthermore, if we are given *full access* to the type distributions of the buyers, we can extend our transformation to the fully general outcome space, removing the downward-closedness assumption. Here is our third main result:

Result III: There exists a polynomial-time ε -BIC to BIC transformation with oracle access to any approximately Bayesian incentive compatible mechanism and full access to the agents’ distributions, outputs an exactly Bayesian incentive compatible mechanism for any *general outcome space* with a negligible revenue loss.

¹Roughly speaking, the setting is downward-closed if the agents have the choice to not participate in the mechanism. See Section 2 for the formal definition.

Note that other ε -BIC to BIC transformations for revenue have been proposed in the literature [23, 38, 18], however, all of the existing transformations require solving a $\#P$ -hard problem repeatedly [28] and therefore cannot be made computationally efficient.

1.1 Our Results and Techniques

We first fix some notations to facilitate our discussion of the results. We consider a general mechanism design environment where there is a set of feasible outcomes denoted by \mathcal{O} . There are n agents, and each agent i has a type t_i drawn from distribution \mathcal{D}_i independently. We use \mathcal{T}_i to denote the support of \mathcal{D}_i , and for every $t_i \in \mathcal{T}_i$, $v_i(t_i, \cdot)$ is a valuation function that maps every outcome to a real number in $[0, 1]$. A mechanism \mathcal{M} consists of an allocation rule $x(\cdot) : \times_{i \in [n]} \mathcal{T}_i \mapsto \Delta(\mathcal{O})$ and a payment rule $p(\cdot) : \times_{i \in [n]} \mathcal{T}_i \mapsto \mathbb{R}^n$. We slightly abuse notation to define $v_i(t_i, x(b)) \equiv \mathbb{E}_{o \sim x(b)} [v_i(t_i, o)]$. If we have query access to \mathcal{M} , then on any query bid profile $b = (b_1, \dots, b_n)$, we receive an outcome $o \sim x(b)$ and payments $p_1(b), \dots, p_n(b)$.

The outcome space \mathcal{O} is called *downward-closed* if each $o \in \mathcal{O}$ can be written as a vector $o = (o_1, \dots, o_n)$ where o_i is the outcome for agent i . And for every $o = (o_1, \dots, o_n) \in \mathcal{O}$, any $o' = (o'_1, \dots, o'_n)$ with $o'_i = o_i$ or $o'_i = \perp$ for every i is also in \mathcal{O} . Here \perp is a null outcome available to each agent i , which represents the option of not participating in the mechanism.

Equipped with the notations, we are ready to discuss our ε -BIC to BIC transformations.

Informal Theorem 1 (ε -BIC to BIC transformation on downward-closed outcome space). *Given sample access to a collection of distributions $(\mathcal{D}_i)_{i \in [n]}$ on a downward-closed outcome space, and query access to an ε -BIC and individually rational (IR) mechanism $\mathcal{M} = (x, p)$ with respect to $\times_{i \in [n]} \mathcal{D}_i$. We can construct another mechanism \mathcal{M}' that is exactly BIC and IR with respect to $\times_{i \in [n]} \mathcal{D}_i$, and its revenue is at most $O(n\sqrt{\varepsilon})$ worse than the revenue of \mathcal{M} . Moreover, for any bid profile $b = (b_1, \dots, b_n)$, \mathcal{M}' computes an outcome $o \in \mathcal{O}$ and payments $p_1(b), \dots, p_n(b)$ in expected running time $\text{poly}(\sum_{i \in [n]} |\mathcal{T}_i|, 1/\varepsilon)$ and makes in expectation at most $\text{poly}(\sum_{i \in [n]} |\mathcal{T}_i|, 1/\varepsilon)$ queries to \mathcal{M} .*

Informal Theorem 2 (ε -BIC to BIC transformation on general outcome space). *Given full access to the collection of distributions $(\mathcal{D}_i)_{i \in [n]}$ on a general outcome space such that $|\text{supp}(\mathcal{D}_i)| \leq m$ for every i , and query access to an ε -BIC and individually rational (IR) mechanism \mathcal{M} with respect to $\times_{i \in [n]} \mathcal{D}_i$. We can construct a mechanism \mathcal{M}' that is exactly BIC and IR with respect to $\times_{i \in [n]} \mathcal{D}_i$. Moreover its revenue is within an additive $O(nm\varepsilon)$ of the revenue of \mathcal{M} . Furthermore, the running time of the constructed mechanism is $\text{poly}(n, m, 1/\varepsilon)$ and the mechanism makes at most $\text{poly}(n, m, 1/\varepsilon)$ queries to \mathcal{M} .*

Previous transformations can produce an \mathcal{M}' with similar guarantees in the downward-closed setting but require $\text{poly}(\prod_{i \in [n]} |\mathcal{T}_i|)$ time to run [23, 38, 18]. In the special case, where there exists symmetry in the agents' type distributions, the transformation can be improved to run in time $\text{poly}(\sum_{i \in [n]} |\mathcal{T}_i|)$, as the interim allocation probabilities and payments of the mechanism \mathcal{M} can be computed efficiently via a polynomial-size LP.² Our result achieves the $\text{poly}(\sum_{i \in [n]} |\mathcal{T}_i|)$ running time without the symmetry assumption.

1.1.1 Our Result for the Downward-Closed Outcome Space

We first discuss our result for the downward-closed outcome space. To illustrate our new ideas, we first briefly review the constructions in the literature. In the heart of all the previous constructions lies the problem called *replica-surrogate matching*.

²If the interim allocation probabilities and payments of \mathcal{M} are given, the edge weights in the Replica-Surrogate matching can be computed efficiently. See the next paragraph for more details.

Replica-Surrogate Matching For each agent i , form a bipartite graph G_i . The left hand side nodes are called replicas, which are types sampled i.i.d. from \mathcal{D}_i . In particular, the true type t_i of agent i is one of the replicas. On the right hand side, the nodes are called surrogates, which are also types sampled from \mathcal{D}_i . The edge between a replica with type $t^{(j)}$ and a surrogate with type $t^{(k)}$ is assigned weight $w_{jk} \equiv \mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [v_i(t^{(j)}, x(t^{(k)}, t_{-i})) - p_i(t^{(k)}, t_{-i})]$ ³, which is the interim utility of agent i when her true type is $t^{(j)}$ but reports $t^{(k)}$ to \mathcal{M} . Compute the maximum weight matching on G_i . The true type t_i selects a surrogate using the matching to compete in \mathcal{M} . Agent i competes in \mathcal{M} using the type of the surrogate she is matched to in the maximum weight matching.

The intuition is that since \mathcal{M} is not BIC, the true type t_i may prefer the outcome and payment from reporting some different type. The matching is set up to allow the true type t_i to pick a more favorable type to compete in \mathcal{M} for it. But why wouldn't the agent misreport in the matching? After all, the edge weights depend on the agent's report. As it turns out, to guarantee incentive compatibility, one needs to find a matching with a *maximal-in-range* algorithm. Namely, the matched surrogate is selected to maximize the agent's induced utility less some cost that only depends on the outcome. It is not hard to verify that the maximum weight matching is indeed maximal-in-range, and therefore the agent has no incentive to lie.

But why does the maximum weight matching take exponential time to find? The problem is that we are not given the edge weights. For each edge, we can only sample from a distribution whose mean is the weight of the edge: Sample t_{-i} from \mathcal{D}_{-i} and compute $v_i(t^{(j)}, x(t^{(k)}, t_{-i})) - p_i(t^{(k)}, t_{-i})$. Even if we assume that we know the distributions $(\mathcal{D}_i)_{i \in [n]}$, it still takes $\text{poly}(\prod_{j \neq i} |\mathcal{T}_j|)$ time to compute the weight of a single edge exactly, which is already an exponential on $\sum_{i \in [n]} |\mathcal{T}_i|$. But why can't we first estimate the edge weights with samples and find the maximum matching using the estimated weights? The issue is that no matter how many samples we take, the empirical mean will be off by some estimation error. The maximal-in-range property is so fragile that even a tiny bit of estimation error can cause the algorithm to violate the property, making the whole mechanism not incentive compatible. See Example 2 below for a more detailed explanation.

Black-box Reduction for Welfare Maximization To overcome the difficulty, we turn to another important problem in mechanism design, black-box reduction for welfare maximization, for inspiration. A line of beautiful results [29, 8, 28, 24] initiated by Hartline and Lucier shows that the mechanism design problem of welfare maximization in the Bayesian setting can be black-box reduced to the algorithmic problem of welfare maximization. The replica-surrogate matching is again the central piece in the reduction. Indeed, the idea of replica-surrogate matching was first proposed by Hartline et al. [27, 28], and later introduced by Daskalakis and Weinberg [23] to the study of ε -BIC to BIC transformation for revenue. The main difference of the two scenarios is the way the edge weights are defined. For welfare maximization, the edge weight between a replica $t^{(j)}$ and a surrogate $t^{(k)}$ is $v_{jk} \equiv \mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [v_i(t^{(j)}, x(t^{(k)}, t_{-i}))]$, namely, the *interim value* for agent i when her true type is $t^{(j)}$ but reports $t^{(k)}$ to \mathcal{M} . We will refer to the one with interim utilities as edge weights the *U-replica-surrogate matching* and the one with interim values as edge weights the *V-replica-surrogate matching*. The main reason that we would like to distinguish the two settings is as follows: in a V-replica-surrogate matching all edge weights are nonnegative, while in a U-replica-surrogate matching the edge weights may be negative. The importance of the presence of negative edges will become clear soon. Obviously, it also takes exponential time to compute the exact maximum weight V-replica-surrogate matching due to the same reason discussed above.

A result by Dughmi et al. [24] shows how to circumvent this barrier for welfare maximization. Their solution has the following two main components: (i) a polynomial time maximal-in-range algorithm to

³The true weight $w_{jk} \equiv \mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [v_i(t^{(j)}, x(t^{(k)}, t_{-i})) - (1 - \eta) \cdot p_i(t^{(k)}, t_{-i})]$ is computed using a discounted price, but we can ignore the difference for now.

solve the *maximum entropy regularized perfect matching* problem; (ii) the *fast exponential Bernoulli race*, a new Bernoulli factory ⁴, that allows them to execute the algorithm in (i) exactly with only sample access to distributions whose means are the edge weights. They use the algorithm to find a maximum entropy regularized V-replica-surrogate matching, and argue that this matching has approximately maximum weight, which allows them to conclude that their new mechanism loses at most a negligible fraction of the welfare.

Why is the mechanism by Dughmi et al. [24] unsuitable? The reason turns out to be subtle. As the U-replica-surrogate matching contains negative edges and the algorithm by Dughmi et al. [24] always returns a perfect matching, some agent types may receive negative utilities from the matching. To guarantee individually rationality, the mechanism must compensate these types. However, due the incentive compatibility constraint, the mechanism must also compensate other agent types. One might think that the overall compensation can be shown to be negligible. Unfortunately, in the following example, we show that the overall compensation can in fact dramatically damage the revenue and may even drive the revenue to 0.

Example 1. Consider the following instance with a single agent and outcome space $\mathcal{O} = \{\perp, o\}$. The agent has two possible types H and L with probability $1 - \sigma$ and σ respectively, where $\sigma \in (0, 1)$ is sufficiently small. The agent’s valuation is: $v(H, o) = 1$, $v(H, \perp) = v(L, \perp) = v(L, o) = 0$. The given mechanism \mathcal{M} chooses outcome o and charges 1 if the agent reports H , and chooses outcome \perp and gives the agent ε if the agent reports L . Clearly, \mathcal{M} is ε -BIC and IR. $\text{REV}(\mathcal{M}) = 1 - \sigma - \sigma\varepsilon$. Note that in the U-Replica-Surrogate matching, the edge between replica with type L and surrogate with type H has negative weight -1 .

Denote ℓ the number of surrogates sampled from the above distribution. Let \mathcal{M}' be the constructed mechanism that always selects a perfect replica-surrogate matching. Denote $p(\cdot)$ the payment function of \mathcal{M}' . Then $p(L) \leq 0$ since \mathcal{M}' is IR. Consider the following two scenarios when the agent has true type H . In the first scenario she reports truthfully her type H and in the second scenario she reports L . With probability $\beta = (1 - \sigma)^\ell$, none of the surrogates has type L . In both scenarios the buyer must be matched to a surrogate with type H in the perfect replica-surrogate matching and has value 1 for the result outcome o . Thus the difference of the agent’s expected value between the two scenarios is at most $1 - \beta$. Since \mathcal{M}' is BIC, we must have $p(H) \leq p(L) + 1 - \beta \leq 1 - \beta$. Therefore $\text{REV}(\mathcal{M}') \leq (1 - \sigma) \cdot (1 - \beta)$. For any fixed ε and ℓ , when $\sigma \rightarrow 0$, the $\beta \rightarrow 1$ and revenue loss goes to 1. Note that our mechanism guarantees revenue loss at most $c\sqrt{\varepsilon}$ for any type distribution, where c is an absolute constant.

Let us take a closer look at what happens in the example above. With high probability, the agent with low type L is matched to a surrogate with high type H in the perfect matching and has negative utility. Thus to satisfy individual rationality, the mechanism has to compensate her. To guarantee incentive compatibility, the mechanism will also have to compensate the agent when her true type is H . However, the total compensation is so large that it essentially drives the revenue of the constructed mechanism to 0. Our main challenge is to enforce both individually rationality and incentive compatibility in the presence of negative edges without sacrificing much of the revenue.

As Example 1 implies, to preserve revenue, it is crucial to avoid matching a replica with a negative weight edge with high probability in the U-replica-surrogate matching. Again the exact weight can not be computed efficiently. One may try to remove the negative edges using samples. However, removing edges based on the empirical means from samples could easily violate the maximal-in-range property. See Example 2.

Example 2. Let N be the number of samples that the algorithm uses to calculate the empirical expectation. Choose $\sigma > 0$ such that $\frac{\sigma}{1-2\sigma} < \frac{1}{N}$. Consider the following example with 1 node on each side. There are

⁴A Bernoulli factory is an algorithm that with sample access to a p -coin to simulate a $f(p)$ -coin. In Section 2.1 we give a brief introduction to Bernoulli factories. We also refer the readers to [30, 35] and the references therein for more details.

two instances. For the first instance, the random variable $\mathcal{F}^{(1)}$ attached to this edge is 1 w.p. 2σ , and $-\frac{\sigma}{1-2\sigma}$ with probability $1 - 2\sigma$. The edge weight $\omega^{(1)} = \sigma$. For the second instance, $\mathcal{F}^{(2)}$ is σ w.p. 1 and $\omega^{(2)} = \sigma$.

For the above example, both instances have the same edge weight and any maximal-in-range allocation will always output the same matching. However in the first instance, with probability $(1 - 2\sigma)^N < 1$ the empirical expectation is negative and the two nodes are not matched. While in the second instance the algorithm will always match the two nodes. Thus the output matching is not maximal-in-range. It is well-known that if the allocation is maximal-in-range, there must exist a payment rule such that the agent is incentive-compatible. Thus the algorithm will violate the incentive-compatibility when applied to the replica-surrogate matching.

Our Solution for the Downward-Closed Outcome Space. Our transformation on downward-closed environments is directly inspired by [24] but differs in several major ways. Our plan is to design an algorithm, for general graphs with arbitrary weights, that satisfies the following two properties:

1. The algorithm produces a distribution of matchings whose expected weight is close to the maximum weight matching.
2. For any LHS node, the expected weight of the edge matched to it is not too negative.

Note that for graphs with positive weights, the algorithm by Dughmi et al. [24] satisfies both properties. When the edges are negative, the first property is not satisfied by their algorithm in general. Even if we only consider U-replica-surrogate matching, their algorithm still violates the second property. Interestingly, we provide a reduction from the case of arbitrary edge weights to the case with only positive edge weights. Indeed, our reduction can be succinctly summarized by the following formula, if an edge has weight w_{jk} , set the new weight by applying the δ -softplus function to w_{jk} ⁵: $\zeta_\delta(w_{jk}) = \delta \cdot \log(\exp(w_{jk}/\delta) + 1)$, where $\delta > 0$ is a parameter of our algorithm. Note that for any value of w_{jk} , $\zeta_\delta(w_{jk})$ is always nonnegative! Moreover, the maximum entropy regularized matching on weights $(\zeta_\delta(w_{jk}))_{j,k}$ can be shown to be close to the maximum weight matching on $(w_{jk})_{j,k}$, and the second property also holds due to nice features of the algorithm and the softplus function. So it seems that we only need to run the algorithm from [24] on the new weights $(\zeta_\delta(w_{jk}))_{j,k}$. An astute reader may have already realized that being able to run the algorithm on $(w_{jk})_{j,k}$ does not imply that one can run the algorithm on $(\zeta_\delta(w_{jk}))_{j,k}$, as we can only sample from distributions whose means are $(w_{jk})_{j,k}$ but not $(\zeta_\delta(w_{jk}))_{j,k}$. One idea is to construct a Bernoulli factory to simulate a $\zeta_\delta(w_{jk})$ -coin using a w_{jk} -coin. To the best of our knowledge, no such construction exists. We take a different approach and make use of a crucial property of the algorithm from [24]. Namely, if we run their algorithm with the same parameter δ , the algorithm only needs to sample from the softmax function over the weights. More specifically, with weights $(w_{jk})_{j,k}$, it suffices to have the ability to sample an edge (j, k) with probability exactly $\frac{\exp(w_{jk}/\delta)}{\sum_{k'} \exp(w_{jk'}/\delta)}$. Despite the fact that we cannot directly sample from distributions with means $(\zeta_\delta(w_{jk}))_{j,k}$, we can indeed sample edge (j, k) with exactly the right probability, as

$$\frac{\exp(\zeta_\delta(w_{jk})/\delta)}{\sum_{k'} \exp(\zeta_\delta(w_{jk'})/\delta)} = \frac{\exp(w_{jk}/\delta) + 1}{\sum_{k'} (\exp(w_{jk'}/\delta) + 1)},$$

which can be sampled efficiently using the fast exponential Bernoulli race given only sample access to distributions with means equal to the original edge weights $(w_{jk})_{j,k}$.

Our second contribution is to show that an approximately maximum U-replica-surrogate matching suffices to guarantee only a small loss in revenue. Previous results [23, 38, 18] only prove the statement for the

⁵The function $\log(\exp(x) + 1)$ is known as the soft plus function.

exactly maximum matching. We provide a more delicate analysis that allows us to extend the statement to approximately maximum matchings. Finally, as the agent may receive negative utility from certain surrogates, we sometimes need to subsidize the agent to ensure individual rationality. Due to the second property of our algorithm, we can argue that the total subsidy is small compared to the revenue. We emphasize that the U-replica-surrogate matching found in our mechanism may not be perfect. Thus the assumption \mathcal{O} being downward-closed is necessary, in order to allow the agent whose true type is unmatched in the matching to receive a \perp outcome. See Mechanism 3 in Section 5 for more details.

1.1.2 Our Result for the General Outcome Space

Our result for the general outcome space is based on the *regularized replica-surrogate fractional assignment mechanism* by Hartline et al. [28]. They use this mechanism to provide a black-box reduction for welfare maximization, but this mechanism only applies to discrete type space and requires full access to all agents' type distributions. We refer the readers to Section 7.1 for details of their mechanism.

The main barrier for applying their approach to transform an ε -BIC mechanism to a BIC mechanism is that their mechanism does not provide any guarantees on the revenue. More specifically, the prices of their mechanism are determined by a set of optimal dual variables of their problem. Although any set of optimal dual variables can guarantee the mechanism to be incentive compatible and individually rational, which is sufficient for welfare maximization, some choices could result in substantial revenue loss, making them unsuitable to preserve revenue. In fact, Example 4 illustrates that for some optimal dual variables, the revenue loss due to negative prices can be very high. Our main contribution is to show an efficient algorithm to find a set of optimal dual variables such that the induced prices only cause negligible loss in the revenue.

1.2 Application to Multi-dimensional Revenue Maximization

We next apply the ε -BIC to BIC transformation to obtain our black-box reduction for revenue maximization. We first introduce the problem formally.

Multi-Dimensional Revenue Maximization (MRM): Given as input n type distributions $\mathcal{D}_1, \dots, \mathcal{D}_n$ and a set of feasible outcomes \mathcal{O} , output a BIC and IR mechanism \mathcal{M} who chooses outcomes from \mathcal{O} with probability 1 and whose expected revenue is optimal relative to any other, possibly randomized, BIC, and IR mechanism with respect to $\mathcal{D} = \times_{i \in [n]} \mathcal{D}_i$.

To state our black-box reduction, we introduce the virtual welfare optimization problem.

Virtual Welfare Optimization (VWO) [15]: Given as input n functions $C_i(\cdot) : \mathcal{T}_i \mapsto \mathbb{R}$ and a set of feasible outcomes \mathcal{O} , output an outcome $o \in \operatorname{argmax}_{x \in \mathcal{O}} \sum_i \sum_{t_i \in \mathcal{T}_i} C_i(t_i) \cdot v_i(t_i, x)$. $C_i(\cdot)$ is considered as the weight function that depends on the agent's real type. We refer to the sum $\sum_{t_i \in \mathcal{T}_i} C_i(t_i) \cdot v_i(t_i, x)$ as agent i 's virtual value for outcome x .

Informal Theorem 3. *When the outcome space is downward-closed, given sample access to distribution $\times_{i=1}^n \mathcal{D}_i$ and oracle access to an α -approximation Algorithm G for VWO, we can construct an exactly BIC and IR mechanism $\mathcal{M} = (x, p)$ with respect to $\times_{i \in [n]} \mathcal{D}_i$, that has expected revenue $\alpha \cdot OPT - O(n\sqrt{\varepsilon})$, where OPT is the optimal revenue over all BIC and IR mechanisms with respect to $\times_{i \in [n]} \mathcal{D}_i$. The running time is poly $\left(\sum_{i \in [n]} |\mathcal{T}_i|, \frac{1}{\varepsilon}, b, rt_G \left(\operatorname{poly} \left(\sum_{i \in [n]} |\mathcal{T}_i|, \frac{1}{\varepsilon}, b \right) \right) \right)$, where $rt_G(\cdot)$ is the running time of G , and b is an upper bound on the bit complexity of $v_i(t_i, o)$ for any agent i , any type t_i , and any outcome o .*

Note that a similar result holds for a general outcome space, given that we have *full access* to distribution $\times_{i=1}^n \mathcal{D}_i$. The modified running time and revenue loss can be found in Section 7.

1.3 Further Related Work

Multi-dimensional revenue maximization has recently received lots of attention from computer scientists. Significant progress has been on the computational front [19, 20, 1, 11, 2, 12, 13, 17, 15, 3, 9, 22, 31]. On the structural front, a family of simple mechanisms, i.e., variants of sequential posted price and two-part tariff mechanisms, have been shown to achieve constant factor approximations of the optimal revenue in quite general settings [5, 39, 38, 16, 21, 18]. ε -BIC to BIC transformation for revenue has been an instrumental tool in obtaining both the computational and structural results [23, 38, 18, 31].

There has also been significant interest in understanding the sample complexity for learning an almost revenue-optimal auction in multi-item settings. Last year, Gonczarowski and Weinberg [26] show that one can learn an almost revenue-optimal ε -BIC mechanism using $\text{poly}(n, m, 1/\varepsilon)$ samples under the item-independence assumption, where n is the number of bidders and m is the number of items. Brustle et al. [10] generalize the result to settings where the item values are drawn from correlated but structured distributions that can be modeled by either Markov random fields or Bayesian Networks. The mechanism they produce is still ε -BIC. Our transformation can certainly convert these mechanisms from [26, 10] into exactly BIC mechanisms, and the transformation requires $\text{poly}\left(\sum_{i \in [n]} |\mathcal{T}_i|, 1/\varepsilon\right)$ many samples. Unfortunately, each $|\mathcal{T}_i|$ is already exponential in m in their settings. The dependence on $|\mathcal{T}_i|$ is unavoidable for us, as our goal is to provide a transformation that is applicable to a general mechanism design setting. Nonetheless, the techniques we develop in this paper may be combined with special structure of the distribution to provide more sample-efficient ε -BIC to BIC transformations.

Recently, Gergatsouli et al. [25] proved that for the case where we have a single buyer with an additive valuation over m independent items and the set of outcomes is downward-closed, an exponential (in m) query complexity is necessary for any black-box reduction for welfare maximization. It will certainly be interesting to see whether a similar lower bound on the sample complexity exists.

1.4 Organization of the Paper

In Section 2, we provide the notations we use throughout the paper. In Section 3, we mention the tools from the literature that are important for our constructions. In Section 4, we provide our new algorithm that solves the entropy regularized matching problem for graphs with arbitrary edge weights assuming the outcome space is downward-closed. In particular, we show the arbitrary edge weight case can be reduced to the nonnegative edge weight case. In Section 5, we describe our ε -BIC to BIC transformation when the outcome space is downward-closed. In Section 6, we show how to use our ε -BIC to BIC transformation to improve the black-box reduction for multi-dimensional revenue maximization. In Section 7 we state our result for the general outcome space. Note that in Sections 4 through 6 we focus on downward-closed outcome spaces, whereas in Section 7 we shift our attention to general outcome spaces.

2 Preliminaries

We specify a general mechanism design setting by the tuple $(n, \mathcal{V}, \mathcal{D}, v, \mathcal{O})$. There are n agents participating in the mechanism. Denote \mathcal{O} the set of all possible outcomes. We consider two types of outcome space.

1. **Downward-Closed Outcome Space:** Each $o \in \mathcal{O}$ can be written as a vector $o = (o_1, \dots, o_n)$ where o_i is the outcome for agent i . We also assume that a null outcome \perp is available to each agent i . One can think of \perp as the option of not participating in the mechanism. Throughout the paper, when we say the outcome space \mathcal{O} is *downward-closed*, then for every $o = (o_1, \dots, o_n) \in \mathcal{O}$, any $o' = (o'_1, \dots, o'_n)$ with $o'_i = o_i$ or $o'_i = \perp$ for every i is also in \mathcal{O} . An example of the downward-closed outcome space

is the combinatorial auction, where the outcome set contains all possible ways to allocate items to agents, and the null outcome represents allocating nothing to the agent. One setting that does not have a downward-closed outcome space is building a public project.

2. **General Outcome Space:** \mathcal{O} is an arbitrary set. This is the most general outcome space, and it can capture settings such as building a public project.

Each agent i has a type t_i from type space \mathcal{V}_i , which is drawn independently from some distribution \mathcal{D}_i . We use $\mathcal{T}_i \subseteq \mathcal{V}_i$ or $\text{supp}(\mathcal{D}_i)$ to denote the support of \mathcal{D}_i . We use \mathcal{D} to denote $\times_{i \in [n]} \mathcal{D}_i$. In the paper we consider discrete type spaces, we assume that every $|\mathcal{T}_i| \leq T$ for some finite T . Note that our results for the downward-closed outcome space can easily be extended to the continuous case using similar techniques as in [24], while our results for general outcome space requires the type space to be discrete. For every $t_i \in \mathcal{V}_i$, $v_i(t_i, \cdot)$ is a valuation function that maps every outcome to a real number in $[0, 1]$. In a downward-closed outcome space, for all agent i and type t_i , $v_i(t_i, o) = 0$ if $o_i = \perp$. Every agent is risk-neutral and has quasi-linear utility.

For any mechanism \mathcal{M} , denote $\text{REV}(\mathcal{M}, \mathcal{D}) = \mathbb{E}_{t \sim \mathcal{D}} \left[\sum_{i \in [n]} p_i(t) \right]$ the expected revenue of \mathcal{M} . We use $\text{REV}(\mathcal{M})$ for short when the agents' distributions and valuation functions are clear. We use the standard definitions of BIC, ε -BIC, IR, and ε -IR:

Bayesian Incentive Compatible (BIC):

$$\mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [v_i(t_i, x(t_i, t_{-i})) - p_i(t_i, t_{-i})] \geq \mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [v_i(t_i, x(t'_i, t_{-i})) - p_i(t'_i, t_{-i})], \quad \forall i \in [n], t_i, t'_i \in \mathcal{T}_i.$$

Individual Rational (IR):

$$\mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [v_i(t_i, x(t_i, t_{-i})) - p_i(t_i, t_{-i})] \geq 0, \quad \forall i \in [n], t_i \in \mathcal{T}_i.$$

ε -BIC:

$$\mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [v_i(t_i, x(t_i, t_{-i})) - p_i(t_i, t_{-i})] \geq \mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [v_i(t_i, x(t'_i, t_{-i})) - p_i(t'_i, t_{-i})] - \varepsilon, \quad \forall i \in [n], t_i, t'_i \in \mathcal{T}_i.$$

ε -IR:

$$\mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [v_i(t_i, x(t_i, t_{-i})) - p_i(t_i, t_{-i})] \geq -\varepsilon, \quad \forall i \in [n], t_i \in \mathcal{T}_i.$$

Coupling between Type Distributions: In order to measure the difference between the two distributions, we will introduce the following definition. Fix every agent i . A *coupling* $c_i(\cdot, \cdot)$ of distribution \mathcal{D}'_i and \mathcal{D}_i is a joint distribution on the probability space $\mathcal{T}'_i \times \mathcal{T}_i$ such that the marginal of c_i coincide with \mathcal{D}'_i and \mathcal{D}_i . In the paper we slightly abuse the notation, denoting $c_i(b)$ a random variable that is distributed according to the conditional distribution of type t_i over \mathcal{T}_i when $t'_i = b$. According to the definition of the coupling, when $t'_i \sim \mathcal{D}'_i$, $c_i(t'_i) \sim \mathcal{D}_i$.

We say v_i is *non-increasing* w.r.t. the coupling c_i if for all $t_i \in \mathcal{T}'_i$, outcome $o \in \mathcal{O}$, and every realized type $c_i(t_i)$, $v_i(t_i, o) \geq v_i(c_i(t_i), o)$. Intuitively, the coupling always maps a ‘‘higher’’ type to a ‘‘lower’’ type. Such coupling is common, for example in a combinatorial auction, rounding agent i 's value for each bundle of items down to the closest multiples of δ can be viewed as such a coupling.

Wasserstein Distance: For any $t_i, t'_i \in \mathcal{V}_i$, let $\text{dist}_i(t_i, t'_i) = \max_{o \in \mathcal{O}} |v_i(t_i, o) - v_i(t'_i, o)|$. The ℓ_∞ -Wasserstein Distance between distribution \mathcal{D}_i and \mathcal{D}'_i w.r.t. dist_i is defined as the smallest expected distance among all couplings. Formally,

$$d_w(\mathcal{D}_i, \mathcal{D}'_i) = \min_{c_i(\cdot, \cdot)} \int \text{dist}_i(t_i, t'_i) dc_i(t_i, t'_i)$$

Finally, we use $\log(\cdot)$ to denote the natural logarithm and Δ^ℓ to denote the set of all distributions over ℓ elements.

Definition 1 (Gibbs Distribution). For any integer ℓ , define the Gibbs distribution $z \in \Delta^\ell$ over ℓ states with temperature β as $z_i = \frac{\exp(E_i/\beta)}{\sum_{i' \in [\ell]} \exp(E_{i'}/\beta)}$ for all $i \in [\ell]$, where E_i is the energy of element i .

Definition 2 (Maximal-in-Range Algorithms). An algorithm is maximal-in-range, if for every $j \in [d\ell]$, there exists a cost function $c(\cdot)$, which may depend on ω_{-j} , such that the allocation $z_j \in \arg\max_{z' \in \mathcal{F}} \sum_{j,k} z'_{jk} \cdot \omega_{jk} - c(z')$ for any ω_j , where \mathcal{F} is a set of all feasible allocations.

2.1 A Brief Introduction to Bernoulli Factories

Suppose we are given a coin with bias μ , can we construct another coin with bias $f(\mu)$ using the original coin? If the answer is yes, then how many flips do we need from the original coin to simulate the new coin? A framework that tackles this problem is called Bernoulli Factories. We refer the reader to [33] for a survey on this topic.

Definition 3 (Keane and O'Brien [30]). Given some function $f : (0, 1) \mapsto (0, 1)$ and black-box access to independent samples of a Bernoulli random variable with bias p , the Bernoulli factory problem is to generate a sample from a Bernoulli distribution with bias $f(p)$.

A useful generalization of the previous model is the following⁶: given sample access to distributions $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_m$ with expectations $\mu_1, \mu_2, \dots, \mu_m \in (0, 1)$, and a function $f : (0, 1)^m \rightarrow \Delta(X)$, where X is a set of feasible outcomes and $\Delta(X)$ is a set of probability distributions over these outcomes, how can we want generate a sample from $f(\mu_1, \dots, \mu_m)$?

Below we state an important result from [24], which we use in this paper. It proposes an algorithm called *Fast Exponential Bernoulli Race* with $X = [m]$. For every $\lambda > 0$, it produce a sample from the Gibbs distribution with temperature $\frac{1}{\lambda}$ and energy μ_i for each outcome i , given only sample access to distributions $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_m$.

Theorem 1. [24] Given any parameter $\lambda > 0$ and sample access to distributions $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_m$ with expectations $\mu_1, \mu_2, \dots, \mu_m \in (0, 1)$, there exists an algorithm that can sample from a Gibbs distribution in Δ^m , where

$$z_i = \frac{\exp(\lambda \mu_i)}{\sum_{j \in [m]} \exp(\lambda \mu_j)},$$

using $O(\lambda^4 m^2 \log(\lambda m))$ samples in expectation.

The *fast exponential Bernoulli race* [24] is a randomized algorithm that allows us to sample from the Gibbs distribution. We use the following result in the rest of our paper.

⁶The model is called *Expectations from Samples* in [24].

Lemma 1. [Fast Exponential Bernoulli Race] For any integer m , any $\delta > 0$, and any $(\alpha_k)_{k \in [m]} \in [0, h]^m$, given sample access to distributions $\mathcal{F}_1, \dots, \mathcal{F}_m$ with expectations $w_1, \dots, w_m \in [-1, 1]$, a sample from the following Gibbs distribution in Δ^m :

$$z_k = \frac{\exp((w_k - \alpha_k)/\delta)}{\sum_{j \in [m]} \exp((w_j - \alpha_j)/\delta)},$$

can be drawn with $\left(\frac{4+h}{\delta}\right)^4 m^2 \log\left(\frac{(4+h)m}{\delta}\right)$ samples from $(\mathcal{F}_k)_{k \in [m]}$ in expectation.

Proof of Lemma 1: First, notice that $(z_k)_{k \in [m]}$ can also be represented as the Gibbs distribution with temperature $\frac{\delta}{h+4}$ and energy $\Omega_k = \frac{w_k - \alpha_k + h + 2}{h+4}$, $k \in [m]$. Note that since $-1 \leq w_k \leq 1$ and $\alpha_k \leq h$, then $0 < \frac{w_k - \alpha_k + h + 2}{h+4} < 1$. Thus, by Theorem 1 with $\lambda = \frac{h+4}{\delta}$, we can generate a sample according to $(z_k)_{k \in [m]}$ with $\left(\frac{4+h}{\delta}\right)^4 m^2 \log\left(\frac{(4+h)m}{\delta}\right)$ samples in expectation. \square

3 Tools from the Literature

3.1 Replica-Surrogate Matching

Now we provide a detailed description of the replica-surrogate matching mechanism used in [23, 38, 18]. For each agent i , the mechanism generates a number of replicas and surrogates from \mathcal{D}_i , and maps the agent's type t_i to one of the surrogates via a maximum weight replica-surrogate matching, and charges the agent the corresponding VCG payment. Then let the matched surrogate participate in the mechanism for the agent. Formally, suppose we are given query access to a mechanism $\mathcal{M} = (x, p)$, we construct a new mechanism \mathcal{M}' using the following two-phase procedure:

Phase 1: Surrogate Selection For each agent i ,

1. Given her reported type $t_i \in \mathcal{D}_i$, create $\ell - 1$ replicas sampled i.i.d. from \mathcal{D}_i and ℓ surrogates sampled i.i.d. from \mathcal{D}_i . The value of ℓ is specified in Lemma 3.
2. Construct a weighted bipartite graph between replicas (and agent i 's true type t_i) and surrogates. The weight between the j -th replica $r^{(j)}$ and the k -th surrogate $s^{(k)}$ is the interim value of agent i when her true type is $r^{(j)}$ but reported $s^{(k)}$ to \mathcal{M} less the interim payment for reporting $s^{(k)}$ multiplied by $(1 - \eta)$:

$$W_i(r^{(j)}, s^{(k)}) = \mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [v_i(r^{(j)}, x(s^{(k)}, t_{-i}))] - (1 - \eta) \cdot \mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [p_i(s^{(k)}, t_{-i})]. \quad (1)$$

3. Treat $W_i(r^{(j)}, s^{(k)})$ as the value of replica $r^{(j)}$ for being matched to surrogate $s^{(k)}$. Run the VCG mechanism among the replicas, that is, compute the maximum weight matching w.r.t. edge weight $W_i(\cdot, \cdot)$ and the corresponding VCG payments. If a replica (or type t_i) is unmatched in the maximum matching, match it to a random unmatched surrogate.

Phase 2: Surrogate Competition Let s_i be the surrogate matched with the agent i 's true type t_i . Run mechanism \mathcal{M} under input $s = (s_1, \dots, s_n)$. Let $o = (o_1, \dots, o_n)$ be a the outcome generated by $x(s)$. If agent i is matched in the maximum matching, her outcome is o_i and her expected payment is $(1 - \eta) \cdot p_i(s)$ plus the VCG payment for winning surrogate s_i in the first phase; Otherwise the agent gets the null outcome \perp and pays 0.

In Figure 1 we illustrate the replica-surrogate scheme.

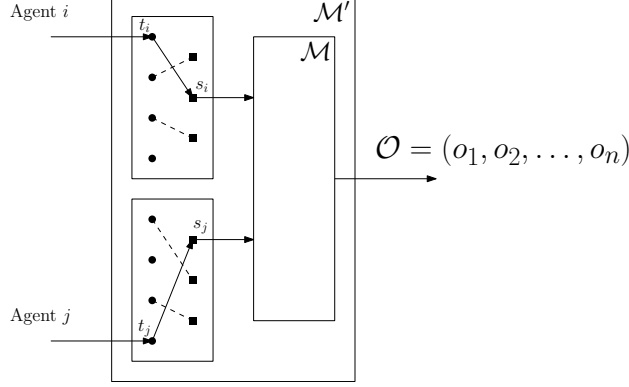


Figure 1: With \bullet we denote the replicas and with \blacksquare the surrogates

Lemma 2. [27, 8, 23, 38, 18] \mathcal{M}' is BIC and IR.

Proof. We prove this in two parts, similarly to [18]. First we argue that the distribution of the surrogate s_i that represents the agent, when the agent reports truthfully, is \mathcal{D}_i . Since we have a perfect matching, an equivalent way of thinking about the process is to draw ℓ replicas, produce the perfect matching (the VCG matching plus the uniform matching between the unmatched replicas and surrogates) and then pick one replica uniformly at random to be the agent. These two processes produce the same joint distribution between replicas, surrogates and the agents i . So we can just argue about the second process of sampling. Since the agent is chosen uniformly at random between the replicas in the second process, the surrogate s_i that represents the agent, will also be chosen uniformly at random between all the surrogates. Thus, the distribution of s_i is \mathcal{D}_i .

We need to argue that for every agent i reporting truthfully is a best response, if every other agent is truthful. In the VCG mechanism, agent i faces a competition with the replicas to win a surrogate. If agent i has type t_i , then her value for winning a surrogate with type s_i in the VCG mechanism is exactly the edge weight

$$W_i(t_i, s_i) = \mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [v_i(t_i, x(s_i, t_{-i}))] - (1 - \eta) \cdot \mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [p_i(s_i, t_{-i})].$$

Clearly, if agent i reports truthfully, the weights on all incident edges between her and all the surrogates will be exactly her value for winning those surrogates. Since agent i is in a VCG mechanism to compete for a surrogate, reporting the true edge weights is a dominant strategy for her, therefore reporting truthfully is also a best response for her assuming the other agents are truthful. It is critical that the other agents are reporting truthfully, otherwise agent i 's value for winning a surrogate with type s_i may be different from the weight on the corresponding edge. □

Moreover, when ℓ is sufficiently large, the revenue of \mathcal{M}' is close to the revenue of \mathcal{M} .

Lemma 3. [23, 38, 18] If \mathcal{M} is an ε -BIC and IR mechanism w.r.t. \mathcal{D} , then for any $\eta \in (0, 1)$ and any $\ell > \frac{T}{\varepsilon^2}$, $\text{REV}(\mathcal{M}', \mathcal{D}) \geq (1 - \eta)\text{REV}(\mathcal{M}, \mathcal{D}) - \Theta(n\varepsilon)/\eta$.

Lemma 3 also follows from a special case of Lemma 23 when $\Delta = 0$ and $d = 1$. The main takeaway of this Lemma is that the above mechanism \mathcal{M}' indeed satisfies the requirement of an ε -BIC to BIC transformation. However, as we discussed in Section 1.1, the mechanism runs in exponential time.

3.2 Online Entropy Regularized Matching

Now we describe the *online entropy regularized matching algorithm* developed by Dughmi et al. [24]. The original application is to find approximately maximum replica-surrogate matching in welfare maximization, but the algorithm is general and can be applied to any *d-to-1 bipartite matching with positive edge weights*.

d-to-1 Matching For every integer ℓ, d , consider the complete bipartite graph between $d\ell$ left hand side nodes (called LHS-nodes) and ℓ right hand side nodes (called RHS-nodes). Let ω_{jk} be the edge weight between LHS-node j and RHS-node k for $j \in [d\ell], k \in [\ell]$. For ease of notation, let $\omega_j = (\omega_{jk})_{k \in [\ell]}$, $\omega = (\omega_j)_{j \in [d\ell]}$, and $\omega_{-j} = (\omega_{j'})_{j' \neq j}$. A matching is called a *d-to-1 matching* if every LHS-node is matched to at most one RHS-node, and every RHS-node is matched to at most d LHS-nodes. A *d-to-1 matching* is called *perfect* if every LHS-node is matched to one RHS-node, and every RHS-node is matched to exactly d LHS-nodes.

In this section, we focus on the case where all edge weights ω are nonnegative, and we refer to this case as the *nonnegative weight d-to-1 matching*. In Section 4, we generalize the results to arbitrary weights.

The optimal *d-to-1 matching* is simply a maximum weight bipartite matching problem. The challenge is that the weights are not given. For every edge (j, k) , we only have sample access to a distribution \mathcal{F}_{jk} whose expectation is ω_{jk} . To the best of our knowledge, none of the algorithms for finding a maximum weight bipartite matching can be implemented exactly with such sample access to the edge weights. Moreover, as we require the replica-surrogate matching mechanism to be incentive compatible, the algorithm should be maximal-in-range. Therefore, finding the maximum weight matching using the empirical means is also not an option, as it violates the maximal-in-range property (see the discussion in Section 1.1).

Dughmi et al. [24] provide a polynomial time maximal-in-range algorithm (Algorithm 1) to compute an approximately maximum weight perfect *d-to-1 matching*. The key idea is to find a “*soft maximum weight matching*” instead of the maximum weight matching by adding an entropy function as a regularizer to the total weight. We summarize the guarantees of Algorithm 1 in Theorem 2. We refer the readers to [24] for intuition behind Algorithm 1. However, to understand this paper, readers can simply treat Theorem 2 as a black box that guarantees that Algorithm 1 is maximal-in-range, and finds approximately maximum expected weight *d-to-1 matching*, with only sample access to the distributions.

Definition 4. Given parameter $\delta > 0$, the (offline) entropy regularized matching program (P) is:

$$\begin{aligned} & \max \sum_{j,k} z_{jk} \cdot \omega_{jk} - \delta \sum_{j,k} z_{jk} \log(z_{jk}) \\ \text{subject to} & \sum_j z_{jk} \leq d, \quad \forall k \in [\ell] \\ & \sum_k z_{jk} = 1, \quad \forall j \in [d\ell] \\ & z_{jk} \in [0, 1], \quad \forall j \in [d\ell], \forall k \in [\ell]. \end{aligned} \tag{2}$$

Lagrangify the constraints $\sum_j z_{jk} \leq d, \forall k \in [\ell]$. The Lagrangian dual of (P) is:

$$L(z, \alpha) = \sum_{j,k} z_{jk} \omega_{jk} - \delta \sum_{j,k} z_{jk} \log(z_{jk}) - \sum_k \alpha_k (d - \sum_j z_{jk}).$$

The following lemma follows from the first-order condition: for any dual variables α , the optimal solution for the Lagrangian is given by a collection of Gibbs distribution $z^* = (z_j^*)_{j \in [d\ell]}$.

Lemma 4. [24] For every dual variables $\alpha \in [0, h]^\ell$, the optimal solution z^* maximizing the Lagrangian

$$L(z, \alpha) \text{ subject to constraints } \sum_k z_{jk}^* = 1, \forall j \in [d\ell] \text{ is: } z_{jk}^* = \frac{\exp\left(\frac{\omega_{jk} - \alpha_k}{\delta}\right)}{\sum_{k' \in [\ell]} \exp\left(\frac{\omega_{jk'} - \alpha_{k'}}{\delta}\right)}, \quad \forall j \in [d\ell], k \in [\ell].$$

If for every edge (j, k) , we are given sample access to a distribution \mathcal{F}_{jk} whose mean is $\omega_{jk} \in [0, 1]$, we can use the fast exponential Bernoulli race [24] to sample from the Gibbs distribution z_j^* for all $j \in [d\ell]$.

In particular, each sample from distribution $z_j^* = (z_{j1}^*, \dots, z_{j\ell}^*)$ only requires in expectation $\text{poly}(h, \ell, 1/\delta)$ many samples from $(\mathcal{F}_{jk})_k$ (Lemma 1).

If the optimal dual variables α^* are known, by complementary slackness, the corresponding z^* in Lemma 4 is the optimal solution of (P) . The gap between the expected weight of z^* and the maximum weight is at most the value of the maximum entropy $\delta \cdot d\ell \log \ell$, so we can simply use the matching sampled according to the distribution z^* . However, as the optimal dual is unknown, the wrong dual variables α may cause a loss of $\sum_k \alpha_k (d - \sum_j z_{jk})$, which may be too large when z is not computed based on the optimal dual variables. To resolve this difficulty, Dughmi et al. [24] introduce the second key idea – *Online Entropy Regularized Matching algorithm* (Algorithm 1). The online algorithm gradually learns a set of dual variables close to the optimum α^* . When the algorithm terminates, it is guaranteed to find a close to optimal solution to program (P) . From Lemma 5, the algorithm is also maximal-in-range for any choice of the parameters δ, η', γ .

Lemma 5. [24] For every j , $\alpha^{(j)}$ and parameter γ , the Gibbs distribution \hat{z}_j (specified in step 4) is maximal-in-range, as

$$\hat{z}_j \in \operatorname{argmax}_{z' \in \Delta_{|K|}} \sum_{k \in K} z'_{jk} \omega_{jk} - \delta \sum_{k \in K} z'_{jk} \log(z'_{jk}) - \sum_{k \in K} \gamma \alpha_k^{(j)} \cdot z'_{jk} \quad 7.$$

Algorithm 1 Online Entropy Regularized Matching with Non-negative Edge Weights (with parameters δ, η', γ)

Require: Sample access to the distribution \mathcal{F}_{jk} whose expectation is ω_{jk} , for every $j \in [d\ell], k \in [\ell]$.

- 1: **for** $j \in [d\ell]$ **do**
 - 2: Let $d_k^{(j-1)}$ be the number of LHS-nodes matched to RHS-node k in the current matching and $K = \{k : d_k^{(j-1)} < d\}$.
 - 3: Set $\alpha^{(j)}$ according to the Gibbs distribution with energy $d_k^{(j-1)}$ for RHS-node $k \in K$ and temperature $1/\eta'$, and $\alpha_k^{(j)} = 0$ for all $k \notin K$.
 - 4: Match LHS-node j to a RHS-node $k \in K$ according to the Gibbs distribution \hat{z}_j over RHS-nodes in K , where the temperature is δ and the energy for matching to a RHS-node $k \in K$ is $(\omega_{jk} - \gamma \alpha_k^{(j)})$. A sample from \hat{z}_j can be generated via the fast exponential Bernoulli race with $\text{poly}(\gamma, \ell, 1/\delta)$ sample from $(\mathcal{F}_{jk})_k$ in expectation (See Lemma 1 for details).
 - 5: **end for**
-

Theorem 2. [24] When $\omega_{jk} \in [0, 1]$ ⁸ for all j, k , Algorithm 1 satisfies the following properties:

1. For any choice of the parameters, it always returns a **perfect** d -to-1 matching.
2. For any choice of the parameters, the algorithm is **maximal-in-range**. The expected running time and sample complexity of Algorithm 1 is $\text{poly}(d, \ell, \gamma, 1/\delta)$.
3. For every $\delta, \eta' > 0$, if $d \geq \ell \log \ell / \eta'^2$ and $\gamma \in \left[\frac{\text{OPT}(P)}{d}, \frac{O(1) \cdot \text{OPT}(P)}{d} \right]$, where $\text{OPT}(P)$ is the optimum of program (P) , the expected value (over the randomness of the Algorithm 1) of $\sum_{j \in [d\ell], k \in [\ell]} \hat{z}_{jk} \omega_{jk} - \delta \sum_{j,k} \hat{z}_{jk} \log(\hat{z}_{jk})$ is at least $(1 - O(\eta')) \cdot \text{OPT}(P)$.

⁷Notice that $\alpha^{(j)}$ only depends on the weights incident to the LHS-nodes 1 to $j - 1$.

⁸The theorem applies to any bounded edge weights $\omega_{jk} \in [0, \mathcal{R}]$. For simplicity we normalize the edge weights to lie between $[0, 1]$.

Moreover, for every $\psi \in (0, 1)$, if we set $\delta = \Theta(\frac{\psi}{\log \ell})$, $\eta' = \Theta(\psi)$, and d and γ satisfy the conditions above, then the expected total weight of the matching output by the algorithm is at most $O(d\ell\psi)$ less than the maximum weight matching.

The only part of Algorithm 1 does not specified is how to choose a γ that is a constant factor approximation to $\frac{\text{OPT}(P)}{d}$. Dughmi et al. [24] show a polynomial time randomized algorithm that produces a γ that falls into $\left[\frac{\text{OPT}(P)}{d}, \frac{O(1) \cdot \text{OPT}(P)}{d}\right]$ with high probability, which suffices to find a close to optimum V-replica-surrogate matching. Please see Appendix A for details.

4 d -to-1 Matching with Arbitrary Edge Weights

To obtain an approximately revenue-preserving ε -BIC to BIC transformation, we need to find a near-optimal U-surrogate-replica matching, where edge weights may be negative. Motivated by this application, we provide a generalization of Theorem 2 to general d -to-1 matchings with arbitrary edge weights. We design a new algorithm (Algorithm 2) with guarantees summarized in Theorem 3.

In Example 1 we point out the issue of directly applying Algorithm 1 to the general d -to-1 matching problem. A tempting way to fix the issue may be to first remove all edges with negative weights then run Algorithm 1. With only sample access to \mathcal{F}_{jk} , one way to achieve this is to remove edges with negative empirical means. In fact, with a sufficiently large number of samples, with high probability, all edges with strictly positive weights will remain and all edges with strictly negative weights will be removed. However, with non-zero probability, some edges will either be kept or removed incorrectly causing the algorithm to violate the maximal-in-range property. See Example 2 for a concrete construction.

An alternative way is to relax the constraint $\sum_k z_{jk} = 1$ to $\sum_k z_{jk} \leq 1$, so the algorithm no longer needs to find a perfect matching. However, Lemma 4 fails to hold as the optimal solution is no longer a Gibbs distribution and it is unclear how to sample efficiently from it with only sample access to \mathcal{F}_{jk} .⁹ A similar attempt is to add a slack variable y to (P) , modifying the constraint $\sum_k z_{jk} = 1$ to $\sum_k z_{jk} + y = 1$. It is equivalent to adding one dummy RHS-node, with weight 0 on every incident edge. Now for every dual variable, the optimal solution for the Lagrangian follows from a Gibbs distribution. However, the program differs from (P) , in particular the new dummy RHS-node has no capacity constraint, and as a result there is no dual variable that corresponds to this dummy node. It is not clear how to modify Algorithm 1 to accommodate the new dummy node and to produce a close to maximum matching.

4.1 Reduction from Arbitrary Weights to Non-Negative Weights

In this section, we provide a reduction from the d -to-1 matching with arbitrary edge weight case to the non-negative edge weight case.

Definition 5. For arbitrary edge weights $(\omega_{jk})_{jk}$ and parameter $\delta > 0$, define the δ -softplus function:

$$\zeta_\delta(\omega_{jk}) = \delta \cdot \log(\exp(\omega_{jk}/\delta) + 1)$$

⁹The issue is that $\sum_k z_{jk}^*$ may be strictly less than 1 and has a complex expression. It is not clear whether we can sample efficiently from z_{jk}^* with only sample access to $(\mathcal{F}_{jk})_{jk}$. Moreover, even if we can sample from the distribution, the guarantees in Theorem 2 may no longer hold.

Consider the entropy regularized matching program (P') w.r.t. weights $(\zeta_\delta(\omega_{jk}))_{jk}$:

$$\begin{aligned}
& \max G(z) = \sum_{j,k} z_{jk} \cdot \zeta_\delta(\omega_{jk}) - \delta \cdot \sum_{j,k} z_{jk} \log(z_{jk}) \\
& \text{subject to } \sum_j z_{jk} \leq d, \quad \forall k \in [\ell] \\
& \quad \quad \quad \sum_k z_{jk} = 1, \quad \forall j \in [d\ell] \\
& \quad \quad \quad z_{jk} \in [0, 1], \quad \forall j \in [d\ell], \forall k \in [\ell].
\end{aligned} \tag{3}$$

Note that $\zeta_\delta(x) > 0$ for any x , so the program (P') is exactly a d -to-1 matching with positive edge weights. We prove that the optimum of (P') is close to the weight of the maximum weight d -to-1 matching (See Lemma 6 and the proof of Theorem 3).

Thus in the rest of this section we will consider approximating the optimum of (P'). Let \hat{z} be the solution produced by Algorithm 1 on (P'). Program (P') is the same as (P) if we substitute the weight ω_{jk} for each LHS-node $r^{(j)}$ and RHS-node $s^{(k)}$ with $\zeta_\delta(\omega_{jk})$. Recall that our main goal is to avoid being matched with negative edges too often. Now for every RHS-node, we construct a dummy 0-RHS-node with weight 0 for all edges incident to it. Let the meta-RHS-node consists of the real RHS-node and the corresponding 0-RHS-node. The weight between the LHS-node j and the meta-RHS-node k is defined as $\zeta_\delta(\omega_{jk})$. We will explain later why the weights are chosen in this way.

Think of the procedure that first executes Algorithm 1 to find a matching between LHS-nodes and meta-RHS-nodes. As a second step, when a LHS-node j is matched to some meta-RHS-node k , we further decide how to match it to the real RHS-node or the 0-RHS-node, according to the following ‘‘softmax’’ program between weight ω_{jk} and 0 (see Figure 2 for an illustration):¹⁰

$$\begin{aligned}
& \max x_{jk}\omega_{jk} - \delta \cdot x_{jk} \log(x_{jk}) - \delta \cdot y_{jk} \log(y_{jk}) \\
& \text{subject to } x_{jk} + y_{jk} = 1 \\
& \quad \quad \quad x_{jk}, y_{jk} \in [0, 1].
\end{aligned} \tag{4}$$

Let (x_{jk}^*, y_{jk}^*) be the optimal solution. One can easily verify that the optimum of the softmax program is equal to $\zeta_\delta(\omega_{jk})$. The two-step procedure finds a d -to-1 matching in the original graph (by removing all edges matched to 0-RHS-nodes). Moreover, when the LHS-node j is matched to the meta-RHS-node k , its expected weight $x_{jk}^*\omega_{jk}$ is at most $O(\delta)$ less than $\zeta_\delta(\omega_{jk})$. Thus by Theorem 2, the two-step procedure that executes Algorithm 1 w.r.t. (P') indeed finds an approximately-optimal d -to-1 matching w.r.t. weights $\zeta_\delta(\omega_{jk})$ and no LHS-node is matched to an edge with too negative weight.

The main issue with the above two-step procedure is that, to execute Algorithm 1 w.r.t. (P'), we will have to sample from a distribution with mean $\zeta_\delta(\omega_{jk})$ with only sample access to the distribution \mathcal{F}_{jk} whose mean is ω_{jk} . To the best of our knowledge, no algorithm exists to sample exactly from such a distribution.

We present our algorithm (Algorithm 2) that solves this issue. The key conceptual idea is to merge the two steps into one. It directly matches LHS-nodes to either real RHS-nodes or 0-RHS-nodes. The reason that this is possible is because the distribution from the combined procedure is again a Gibbs distribution, which allows us to use the fast exponential Bernoulli race to sample directly from it (Observation 2).

We prove a coupling between executing Algorithm 2 over weights $(\omega_{jk})_{jk}$ and executing Algorithm 1 over weights $(\zeta_\delta(\omega_{jk}))_{jk}$, with the same parameters δ, η', γ . Note that both executions are online procedures. Thus the distribution of matching the current LHS-node depends on the previous matching. We carefully prove that the dual variables $\alpha^{(j)}$ and the remaining capacities $(d_k^{(j)})_k$ are the same for every round j . Our result is summarized in Theorem 3.

¹⁰Note that it’s also the entropy regularized matching program between a single LHS-node j and two RHS-nodes (real RHS-node k and 0-RHS-node k).

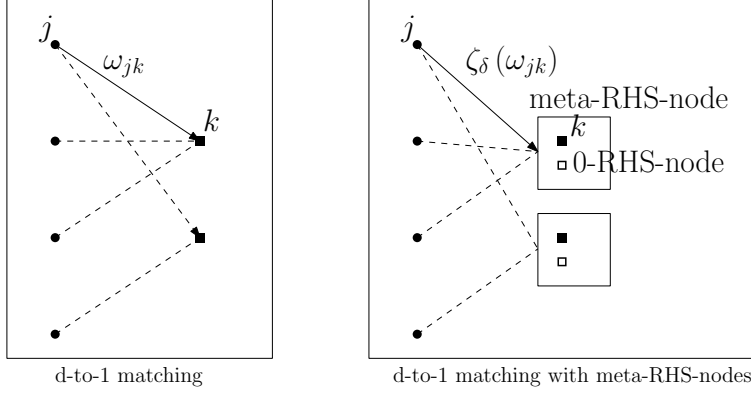


Figure 2: With \bullet we denote the LHS-nodes, with \blacksquare RHS-nodes, with \square we denote the 0-RHS-nodes and with a big rectangle that encloses a \blacksquare and a \square we denote the meta-RHS-nodes.

Theorem 3. When $\omega_{jk} \in [-1, 1]$ ¹¹ for all j, k , Algorithm 2 satisfies the following properties:

1. For any choice of the parameters, dropping all the edges incident to any 0-RHS-nodes in the matching, the algorithm produces a feasible d-to-1 matching (not necessarily perfect).
2. For any choice of the parameters, the algorithm is **maximal-in-range**. The expected running time and sample complexity is $\text{poly}(d, \ell, 1/\delta, \gamma)$.
3. For every $\delta, \eta' > 0$, if $d \geq \ell \log \ell / \eta'^2$ and $\gamma \in \left[\frac{\text{OPT}(P')}{d}, \frac{O(1) \cdot \text{OPT}(P')}{d} \right]$, where $\text{OPT}(P')$ is the optimum of program (P') , then the expected value (over the randomness of the Algorithm 2) of $\sum_{j \in [d\ell], k \in [\ell]} \hat{x}_{jk} \omega_{jk} - \delta \sum_{j,k} \hat{x}_{jk} \log(\hat{x}_{jk}) - \delta \sum_{j,k} \hat{y}_{jk} \log(\hat{y}_{jk})$ is at least $(1 - O(\eta')) \cdot \text{OPT}(P')$.
Moreover, for every $\psi \in (0, 1)$, if we set $\delta = \Theta(\frac{\psi}{\log \ell})$, $\eta' = \Theta(\psi)$, and d and γ satisfy the conditions above, then the expected value of $\sum_{j \in [d\ell], k \in [\ell]} \hat{x}_{jk} \omega_{jk}$, the expected total weight of the matching output by the algorithm (dropping all the edges incident to any 0-RHS-nodes in the matching), has weight at most $O(d\ell\psi)$ less than the maximum weight matching.
4. For every LHS-node j , the expected weight of the edge that matches j is never too much smaller than 0. Formally, for every δ and every $j \in [d\ell]$, $\sum_{k \in [\ell]} \hat{x}_{jk} \omega_{jk} \geq -\delta \cdot \log(2\ell)$.

Remark 1. Property (4) is relevant only when the edge weights may be negative. As discussed in the introduction, this is a crucial property to preserve the revenue in the transformation. Directly applying Algorithm 1 from [24] is insufficient to guarantee this property as shown in Example 1.

To prove Theorem 3, we consider the following auxiliary problem.

Definition 6. For any parameter $\delta > 0$, we define the following auxiliary convex program (P'') :

$$\begin{aligned}
\max \quad & F(x, y) = \sum_{j,k} x_{jk} \omega_{jk} - \delta \cdot \sum_{j,k} (x_{jk} \log(x_{jk}) + y_{jk} \log(y_{jk})) \\
\text{s.t.} \quad & \sum_j (x_{jk} + y_{jk}) \leq d, \quad \forall k \in [\ell] \\
& \sum_k (x_{jk} + y_{jk}) = 1, \quad \forall j \in [d\ell] \\
& x_{jk}, y_{jk} \in [0, 1], \quad \forall j, k
\end{aligned}$$

Let $(\hat{x}_{jk}, \hat{y}_{jk})_{jk}$ be the solution produced by Algorithm 2.

¹¹Again the theorem applies to any bounded edge weights in $[-\mathcal{R}, \mathcal{R}]$. For simplicity we normalize the edge weights to lie in $[-1, 1]$.

Algorithm 2 Online Entropy Regularized Matching with Arbitrary Edge Weights (with parameters δ, η', γ)

Require: Sample access to \mathcal{F}_{jk} whose mean is ω_{jk} , for every j, k .

- 1: For each RHS-node k , add a 0-RHS-node to the bipartite graph with edge weight 0 to every LHS-node. We refer to the k -th original RHS-node the k -th normal-RHS-node.
 - 2: **for** $j \in [d\ell]$ **do**
 - 3: Let $d_k^{(j-1)}$ be the number of LHS-nodes matched to either the k -th normal-RHS-node or the k -th 0-RHS-node in the current matching and $K = \{k : d_k^{(j-1)} < d\}$.
 - 4: Set $\alpha^{(j)}$ according to the Gibbs distribution over RHS-nodes in K , where the energy for any RHS-node $k \in K$ is $d_k^{(j-1)}$ and the temperature is $1/\eta'$. Set $\alpha_k^{(j)} = 0$ for all $k \notin K$.
 - 5: Match LHS-node j to a normal RHS-node (or a 0-RHS-node) $k \in K$ according to the Gibbs distribution over the $2|K|$ RHS-nodes in K , where the temperature is δ and the energy for matching to a normal-RHS-node k is $(\omega_{jk} - \gamma\alpha_k^{(j)})$ and the energy for matching to a 0-RHS-node $k \in K$ is $(-\gamma\alpha_k^{(j)})$. More specifically, match to the normal-RHS-node k with probability $\hat{x}_{jk} = \frac{\exp((\omega_{jk} - \gamma\alpha_k^{(j)})/\delta)}{\sum_{k' \in K} (\exp((\omega_{jk'} - \gamma\alpha_{k'}^{(j)})/\delta) + \exp((-\gamma\alpha_{k'}^{(j)})/\delta))}$ and match to the 0-RHS-node k with probability $\hat{y}_{jk} = \frac{\exp((-\gamma\alpha_k^{(j)})/\delta)}{\sum_{k' \in K} (\exp((\omega_{jk'} - \gamma\alpha_{k'}^{(j)})/\delta) + \exp((-\gamma\alpha_{k'}^{(j)})/\delta))}$. We can generate a sample from (\hat{x}_j, \hat{y}_j) via the fast exponential Bernoulli race with $\text{poly}(\gamma, \ell, 1/\delta)$ sample from $(\mathcal{F}_{jk})_k$ in expectation (See Lemma 1).
 - 6: **end for**
-

Observation 1. For every j , $\alpha^{(j)}$ and parameter γ , match j according to the Gibbs distribution (\hat{x}_j, \hat{y}_j) to the available $2|K|$ RHS-nodes in K ,

$$\hat{x}_{jk} = \frac{\exp((\omega_{jk} - \gamma\alpha_k^{(j)})/\delta)}{\sum_{k \in K} (\exp((\omega_{jk} - \gamma\alpha_k^{(j)})/\delta) + \exp(-\gamma\alpha_k^{(j)}/\delta))}, \hat{y}_{jk} = \frac{\exp((- \gamma\alpha_k^{(j)})/\delta)}{\sum_{k \in K} (\exp((\omega_{jk} - \gamma\alpha_k^{(j)})/\delta) + \exp(-\gamma\alpha_k^{(j)}/\delta))}$$

maximizes

$$\sum_{k \in K} x_{jk} \omega_{jk} - \delta \sum_{k \in K} x_{jk} \log(x_{jk}) - \delta \sum_{k \in K} y_{jk} \log(y_{jk}) - \sum_{k \in K} \gamma \alpha_k^{(j)} \cdot (x_{jk} + y_{jk}),$$

subject to the constraint $\sum_k (x_{jk} + y_{jk}) = 1$.

Observation 2. For every dual variables $\alpha \in [0, h]^\ell$ the optimal solution $\mathbf{x}^*, \mathbf{y}^*$ maximizing the Lagrangian $L((\mathbf{x}, \mathbf{y}), \alpha)$ of program (P'') subject to the constraints $\sum_k (x_{jk} + y_{jk}) = 1, \forall j \in [d\ell]$ is

$$x_{jk}^* = \frac{\exp(\frac{\omega_{jk} - \alpha_k}{\delta})}{\sum_{k'} \left(\exp(\frac{\omega_{jk'} - \alpha_{k'}}{\delta}) + \exp(\frac{-\alpha_{k'}}{\delta}) \right)}, \quad \forall j \in [d\ell], \forall k \in [\ell]$$

$$y_{jk}^* = \frac{\exp(\frac{-\alpha_k}{\delta})}{\sum_{k'} \left(\exp(\frac{\omega_{jk'} - \alpha_{k'}}{\delta}) + \exp(\frac{-\alpha_{k'}}{\delta}) \right)}, \quad \forall j \in [d\ell], \forall k \in [\ell]$$

Hence

$$\frac{x_{jk}^*}{y_{jk}^*} = \exp(\omega_{jk}/\delta), \quad \forall j, k$$

We prove in Lemma 6 that the optimum of (P'') is exactly the same as the optimum of (P') .

Lemma 6. For all $j \in [d\ell]$ and $k \in [\ell]$, if $\frac{x_{jk}}{y_{jk}} = \exp(\omega_{jk}/\delta)$, $x_{jk} + y_{jk} = z_{jk}$, then

$$z_{jk} \cdot \zeta_\delta(\omega_{jk}) - \delta \cdot z_{jk} \log(z_{jk}) = x_{jk} \cdot \omega_{jk} - \delta \cdot x_{jk} \log(x_{jk}) - \delta \cdot y_{jk} \log(y_{jk}). \quad (5)$$

This implies that the optimal objective values of (P') and (P'') are equal.

Proof of Lemma 6:

For every j, k , recall $\zeta_\delta(\omega_{jk}) = \delta \cdot \log(\exp(\omega_{jk}/\delta) + 1)$. Observe that $z_{jk} = (1 + \exp(\omega_{jk}/\delta))y_{jk}$. We have

$$\text{LHS} = z_{jk} \cdot \zeta_\delta(\omega_{jk}) - \delta \cdot z_{jk} \log(z_{jk}) = z_{jk} \cdot \delta \cdot \log\left(\frac{\exp(\omega_{jk}/\delta) + 1}{z_{jk}}\right) = -(\exp(\omega_{jk}/\delta) + 1)y_{jk} \cdot \delta \log(y_{jk})$$

$$\begin{aligned} \text{RHS} &= -\delta \cdot y_{jk} \log(y_{jk}) + x_{jk} \cdot (\omega_{jk} - \delta \log(x_{jk})) \\ &= -\delta \cdot y_{jk} \log(y_{jk}) + \exp(\omega_{jk}/\delta)y_{jk} \cdot (\omega_{jk} - \delta \log(\exp(\omega_{jk}/\delta))) - \delta \log y_{jk} \\ &= -\delta \cdot y_{jk} \log(y_{jk}) \cdot (1 + \exp(\omega_{jk}/\delta)) \end{aligned}$$

Hence, Equation (5) holds. Since the optimal values x_{jk}^*, y_{jk}^* satisfy the requirements by Observation 2, we have that the optimum of (P') is at least as large as the optimum of (P'') . On the other hand, let z^* be the optimal solution of (P'') , we can choose x_{jk}^* and y_{jk}^* so that $x_{jk}^* + y_{jk}^* = z_{jk}^*$ and $\frac{x_{jk}^*}{y_{jk}^*} = \exp(\omega_{jk}/\delta)$. Clearly, $(x_{jk}^*, y_{jk}^*)_{jk}$ is a feasible solution to (P') , therefore the optimum of (P') is at most as large as the optimum of (P'') . Combining the two claims, we prove that (P') and (P'') have the same optimal objective values. \square

Lemma 7. With parameter $\delta \geq 0$, let (x^*, y^*) be the optimal solution of (P'') . The optimum of (P'') , $\sum_{j,k} x_{jk}^* \omega_{jk} - \delta \cdot \sum_{j,k} (x_{jk}^* \log(x_{jk}^*) + y_{jk}^* \log(y_{jk}^*))$, is no smaller than the weight of the maximum weight matching.

Proof. Let x' be the maximum weight matching. It is not hard to see that we can construct a 0–1 vector y' so that (x', y') is a feasible solution of (P'') . As both x' and y' only take values in 0 or 1, the entropy term $-\sum_{j,k} x'_{jk} \log(x'_{jk}) - \sum_{j,k} y'_{jk} \log(y'_{jk}) = 0$. Hence, the optimum of (P'') is at least as large as the weight of the maximum weight matching $\sum_{j,k} x'_{jk} \omega_{jk}$. \square

Proof of Theorem 3: As the algorithm always produces a matching that respects the constraints of (P'') , the first property clearly holds. As the set of available RHS-nodes K and the dual variables $\alpha^{(j)}$ only depend on the first $j - 1$ LHS-nodes but not the LHS-node j , the maximal-in-range property follows from Observation 1. The algorithm runs in $d\ell$ rounds, step 3 and 4 both take $O(\ell)$ time. Step 5 takes expected time $\text{poly}(\gamma, \ell, 1/\delta)$ and $\text{poly}(\gamma, \ell, 1/\delta)$ -many samples from distributions $(\mathcal{F}_{jk})_k$ to complete. Hence, the running time and sample complexity as stated in the second property.

If we execute Algorithm 1 on a d -to-1 matching with weights $(\zeta_\delta(\omega_{jk}))_{jk}$ and Algorithm 2 over weights $(\omega_{jk})_{jk}$ with the same parameters δ, η', γ , we can couple the two executions so that the dual variables $\alpha^{(j)}$ and the remaining capacities $(d_k^{(j)})_k$ are the same for every j . We introduce the new notation $K^{(j)}$ which is exactly the set of available RHS-nodes K in step 2 of both algorithm in round j . Note that $K^{(j)}$ is deterministically determined by $(d_k^{(j-1)})_k$. If $\alpha^{(j)}$ and $K^{(j)}$ are the same in both algorithms for every j , then $\hat{x}_{jk} + \hat{y}_{jk} = \hat{z}_{jk}$ for every $j \in [d\ell]$ and $k \in K^{(j)}$. To verify this, simply observe that

$$\hat{z}_{jk} = \exp\left(\frac{\zeta_\delta(\omega_{jk}) - \gamma \alpha_k^{(j)}}{\delta}\right) = \left(\exp\left(\frac{\omega_{jk}}{\delta}\right) + 1\right) \cdot \exp\left(\frac{-\gamma \alpha_k^{(j)}}{\delta}\right) = \exp\left(\frac{\omega_{jk} - \gamma \alpha_k^{(j)}}{\delta}\right) + \exp\left(\frac{-\gamma \alpha_k^{(j)}}{\delta}\right) = \hat{x}_{jk} + \hat{y}_{jk}.$$

How does the coupling work? We construct it by induction. In the base case where $j = 1$, clearly everything is the same in both algorithms. Suppose the dual variables $\alpha^{(1)}, \dots, \alpha^{(j)}$ and the remaining capacities $(d_k^{(1)})_k, \dots, (d_k^{(j)})_k$ are all the same for the first j rounds, we argue that we can couple the two executions in round $j + 1$ so that $\alpha^{(j+1)}$ and $(d_k^{(j+1)})_k$ remain the same in both algorithms. First, the set $K^{(j+1)}$ is the same, which implies that the dual variables $\alpha^{(j+1)}$ are also the same. Next, Algorithm 1 samples a RHS-node k according to distribution \hat{z}_{j+1} and Algorithm 2 samples a RHS-node according to distribution $(\hat{x}_{j+1}, \hat{y}_{j+1})$. Note that $\hat{x}_{(j+1)k} + \hat{y}_{(j+1)k} = \hat{z}_{(j+1)k}$, so wherever Algorithm 1 matches the LHS-node $j + 1$ to a RHS-node k we match the LHS-node $j + 1$ to the normal RHS-node k with probability $\frac{\hat{x}_{(j+1)k}}{\hat{z}_{(j+1)k}}$ and to the 0-RHS-node with probability $\frac{\hat{y}_{(j+1)k}}{\hat{z}_{(j+1)k}}$. Clearly, this coupling makes sure the new remaining capacities $(d_k^{(j+1)})_k$ also remain the same. Combining the coupling with Lemma 6, we conclude that

$$G(\hat{z}) = \sum_{j,k} \hat{z}_{jk} \cdot \zeta_\delta(\omega_{jk}) - \delta \cdot \sum_{j,k} \hat{z}_{jk} \log(\hat{z}_{jk}) = \sum_{j,k} \hat{x}_{jk} \omega_{jk} - \delta \cdot \sum_{j,k} (\hat{x}_{jk} \log(\hat{x}_{jk}) + \hat{y}_{jk} \log(\hat{y}_{jk})) = F(\hat{x}, \hat{y}).$$

By Theorem 2, the expected value of $G(\hat{z})$ is a $(1 - O(\eta'))$ multiplicative approximation to $\text{OPT}(P')$, if we choose the parameters according to the third property of the statement. Therefore, the expected value of $F(\hat{x}, \hat{y})$ is a $(1 - O(\eta'))$ multiplicative approximation to $\text{OPT}(P')$. Since the optimum of (P'') , $\text{OPT}(P'')$, is the same as $\text{OPT}(P')$ (Lemma 6), the expected value of $F(\hat{x}, \hat{y})$ is also a $(1 - O(\eta'))$ multiplicative approximation to $\text{OPT}(P'')$. Now, invoke Lemma 7, we know that the expected value of $F(\hat{x}, \hat{y})$ is at least a $(1 - O(\eta'))$ multiplicative approximation to the weight of the maximum weight matching, which we denote as OPT . Note that the entropy term $-\delta \cdot \left(\sum_{j,k} \hat{x}_{jk} \log(\hat{x}_{jk}) + \sum_{j,k} \hat{y}_{jk} \log(\hat{y}_{jk}) \right)$ is non-negative and at most $\delta d\ell \log(2\ell)$, hence the expected weight of the matching produced by Algorithm 2, the expected value of $\sum_{j,k} \hat{x}_{jk} \omega_{jk}$, is at least $(1 - O(\eta')) \cdot \text{OPT} - \delta d\ell \log(2\ell)$.

If we choose $\delta = \Theta\left(\frac{\psi}{\log \ell}\right)$, $\eta' = \Theta(\psi)$, then $\delta d\ell \log(2\ell) = \Theta(d\ell\psi)$ and $O(\eta') \cdot \text{OPT} = O(d\ell\psi)$ as $\text{OPT} \leq d\ell$. Thus, the expected weight of the matching produced by Algorithm 2 is within an additive error of $\Theta(d\ell\psi)$ from the weight of the maximum weight matching. This completes our proof for the third property.

Now we are going to prove the final bullet of the theorem. We denote the entropy of (x, y) as:

$$H(x, y) = - \left(\sum_k x_k \log(x_k) + \sum_k y_k \log(y_k) \right)$$

By Observation 1,

$$(\hat{x}_j, \hat{y}_j) = \operatorname{argmax}_{(x_j, y_j)} \sum_k x_{jk} \cdot \omega_{jk} + \delta \cdot H(x_j, y_j) - \sum_k \gamma \alpha_k^{(j)} \cdot (x_{jk} + y_{jk})$$

Therefore the objective of solution $(0, \hat{x}_j + \hat{y}_j)$ is a lower bound on the objective of solution (\hat{x}_j, \hat{y}_j) :

$$\begin{aligned} & \sum_k \hat{x}_{jk} \cdot \omega_{jk} + \delta \cdot H(\hat{x}_j, \hat{y}_j) - \sum_k \gamma \alpha_k^{(j)} \cdot (\hat{x}_{jk} + \hat{y}_{jk}) \\ & \geq \delta \cdot H(0, \hat{x}_j + \hat{y}_j) - \sum_k \gamma \alpha_k^{(j)} \cdot (\hat{x}_{jk} + \hat{y}_{jk}) \end{aligned}$$

Since $H(0, \hat{x}_j + \hat{y}_j) \geq 0$, we can conclude that:

$$\sum_k \hat{x}_{jk} \cdot \omega_{jk} \geq -\delta \cdot H(\hat{x}_j, \hat{y}_j) \geq -\delta \log(2\ell)$$

□

5 ε -BIC to BIC Transformation

In this section, we present our ε -BIC to BIC transformation. In Theorem 4, we prove a more general statement where the given mechanism \mathcal{M} is ε -BIC with respect to $\mathcal{D} = \times_{i \in [n]} \mathcal{D}_i$, while we construct an exactly BIC mechanism \mathcal{M}' with respect to a different distribution $\mathcal{D}' = \times_{i \in [n]} \mathcal{D}'_i$. If $\mathcal{D} = \mathcal{D}'$, the problem is the ε -BIC to BIC transformation problem. We show that the revenue of \mathcal{M}' under \mathcal{D}' decreases gracefully with respect to the Wasserstein Distance of the two distributions. For every i , we denote $d_w(\mathcal{D}_i, \mathcal{D}'_i)$ the ℓ_∞ -Wasserstein Distance of distributions $\mathcal{D}_i, \mathcal{D}'_i$. We slightly abuse notation and let $d_w(\mathcal{D}, \mathcal{D}') = \sum_{i=1}^n d_w(\mathcal{D}_i, \mathcal{D}'_i)$.

Our mechanism works in the following way. After some agent reports her type, we sample $d\ell - 1$ i.i.d. replicas, ℓ i.i.d. “real” surrogates and add ℓ 0-surrogates, for some appropriately chosen parameters d, ℓ . The true type is inserted into a random position in the replicas. We define the weight between a replica $r^{(j)}$ and a “real” surrogate $s^{(i)}$ to be (almost) the expected utility of type $r^{(j)}$ if she reported $s^{(i)}$ to the original mechanism \mathcal{M} . The weights between replicas and 0-surrogates are 0. Then, we run Algorithm 2 in order to get a d -to-1 matching between the replicas and the surrogates. To ensure that this matching is truthful, we impose appropriately selected payments to the agent. This is the first phase of \mathcal{M}' . Now, suppose that the true agent is matched to surrogate $s^{(k)}$. In the second phase of the mechanism, we let $s^{(k)}$ participate in \mathcal{M} along with the matched surrogates of the other agents. If $s^{(k)}$ is a 0-surrogate, then the agent gets the outcome \perp and pays nothing. Otherwise, she gets the outcome that the surrogate gets and pays the same price discounted by a factor of $(1 - \eta)$. We remark that a downward-closed outcome space \mathcal{O} is necessary here to allow a \perp outcome.

A formal description of our mechanism is shown in Mechanism 3. In step 3, the parameter γ is estimated using an approach similar to Dughmi et al. [24]. See Lemma 17 in Appendix A for more details.

How do we compute the payment of Phase 1? Note that if any agent $i' \in [n]$ reports truthfully, then the surrogate $s_{i'}$ who participates for agent i' in Phase 2¹² is exactly drawn from distribution $\mathcal{D}_{i'}$. Therefore, if all the other agents report truthfully, agent i' 's value for winning a normal surrogate s is exactly $W_{i'}(t_{i'}, s)$ and 0 otherwise. In other words, Mechanism 3 is equivalent to a competition among replicas to win surrogates, and the edge weight between a replicas and a surrogate is exactly the replica's value for the surrogate. To show that Mechanism 3 is BIC, it suffices to prove that the payment of Phase 1 incentivizes the replicas to submit their true edge weights. As Algorithm 2 is maximal-in-range, such payment rule indeed exists.

If the true type is the j -th replica, and the reported type t_i induces edge weights $(\omega_{jk})_{k \in [\ell]}$, charge the agent

$$\delta \sum_{k \in K} x_{jk} \log(x_{jk}) + \delta \sum_{k \in K} y_{jk} \log(y_{jk}) + \sum_{k \in K} \gamma \alpha_k^{(j)} \cdot (x_{jk} + y_{jk}), \quad (6)$$

where $x_{jk} = \frac{\exp((\omega_{jk} - \gamma \alpha_k^{(j)})/\delta)}{\sum_{k \in K} (\exp((\omega_{jk} - \gamma \alpha_k^{(j)})/\delta) + \exp(-\gamma \alpha_k^{(j)}/\delta))}$, $y_{jk} = \frac{\exp((- \gamma \alpha_k^{(j)})/\delta)}{\sum_{k \in K} (\exp((\omega_{jk} - \gamma \alpha_k^{(j)})/\delta) + \exp(-\gamma \alpha_k^{(j)}/\delta))}$, and $\alpha^{(j)}$ is the set of dual variables in the j -th iteration of Algorithm 2. Observation 1 implies that the payment rule is BIC. However, direct implementation of the payment requires knowing the edge weights which we

¹²Agent i' may be matched to a 0-surrogate, then $s_{i'}$ is the type of the corresponding normal surrogate.

Mechanism 3 ε -BIC to BIC Transformation (Mechanism \mathcal{M}')

Require: Query access to an IR mechanism $\mathcal{M} = (x, p)$ w.r.t. $\mathcal{D} = \times_{i \in [n]} \mathcal{D}_i$; sample access to the type distribution \mathcal{D}_i and \mathcal{D}'_i for every $i \in [n]$; Parameters $\eta, \eta', \delta, \ell$, and $d \geq 32 \frac{\log(8\eta'^{-1})}{\delta^2 \ell \log^2(\ell)}$.

Phase 1: Surrogate Selection

- 1: **for** $i \in [n]$ **do**
- 2: Sample ℓ surrogates i.i.d. from \mathcal{D}_i . We use \mathbf{s} to denote all surrogates.
- 3: Estimate γ with parameters η' and δ using the algorithm in Lemma 17.
- 4: Agent i reports her type t_i . Create $d\ell - 1$ replicas sampled i.i.d. from \mathcal{D}'_i and insert t_i into the replicas at a uniformly random position. We use \mathbf{r} to denote all the $d\ell$ replicas.
- 5: For each normal surrogate k , also create a 0-surrogate with a special type \diamond . Create a bipartite graph G_i between the $d\ell$ replicas and 2ℓ surrogates. Define the weight between the j -th replica $r^{(j)}$ (t_i is also a replica) and the k -th normal surrogate $s^{(k)}$ using

$$W_i(r^{(j)}, s^{(k)}) = \mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} \left[v_i(r^{(j)}, x(s^{(k)}, t_{-i})) \right] - (1 - \eta) \cdot \mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} \left[p_i(s^{(k)}, t_{-i}) \right].$$

A 0-surrogate has edge weight 0 to every replica, that is $W_i(r^{(j)}, \diamond) = 0$ for all j .

- 6: Run Algorithm 2 on G_i with parameters δ, η' , and γ . For any edge between a replica $r^{(j)}$ and a surrogate $s^{(k)}$, we can sample the edge weight by first sampling t_{-i} from \mathcal{D}_{-i} , then query \mathcal{M} on input $(s^{(k)}, t_{-i})$, and compute $v_i(r^{(j)}, x(s^{(k)}, t_{-i})) - (1 - \eta) \cdot p_i(s^{(k)}, t_{-i})$.
- 7: Suppose the reported type t_i of agent i is matched to the k -th normal surrogate or the k -th 0-surrogate. Let s_i be the type of the k -th normal surrogate.
- 8: Sample λ from $U[0, 1]$ and charge the agent $q_i(t_i, \lambda)$, which is her payment for Phase 1. $q_i(t_i, \lambda)$ is computed via a modified implicit payment (Defintion 7).
- 9: **end for**

Phase 2: Surrogate Competition

- 10: Run mechanism \mathcal{M} on input $s = (s_1, \dots, s_n)$. Let $o = (o_1, \dots, o_n)$ be a random outcome sampled from $x(s)$. If agent i is matched to a normal surrogate in Phase 1, her outcome is o_i and her payment for Phase 2 is $(1 - \eta) \cdot p_i(s)$; otherwise the agent gets the outcome \perp and pays 0 for Phase 2.
-

only have sample access to. We use a procedure called the implicit payment computation [4, 29, 6, 7, 24] to circumvent this difficulty.

Definition 7 (Implicit Payment Computation). *For any fixed parameters δ, η, η' and γ , let $(\omega_{jk})_{jk}$ be the edge weights on a $[d\ell] \times [2\ell]$ size bipartite graph, we use $\mathcal{A}_j(\omega)$ to denote $(\hat{x}_{j1}, \dots, \hat{x}_{j\ell}, \hat{y}_{j1}, \dots, \hat{y}_{j\ell})$, the allocation of the j -th LHS-node/replica to the surrogates computed by Algorithm 2 on the bipartite graph. Now, fix \mathbf{r} and \mathbf{s} , we use $u_i(t_i, (x, y))$ to denote $\sum_{k \in [\ell]} x_k \cdot W_i(t_i, s^{(k)})$. Suppose agent i 's reported type t_i is in position π , that is, $r^{(\pi)} = t_i$. To compute price $q_i(t_i, \lambda)$, let surrogate s' be the surrogate sampled from $\mathcal{A}_\pi(W)$ by Algorithm 2 in step 6, where W is the collection of edge weights in graph G_i as defined in step 5 of Mechanism 3, and we sample a surrogate s'' from $\mathcal{A}_\pi(\lambda W_\pi, W_{-\pi})$, where W_π contains all weights of the edges incident to the π -th replica, and λW_π is simply multiplying each weight in W_π by λ . Then we sample t_{-i} from \mathcal{D}_{-i} , the price $q_i(t_i, \lambda)$ is*

$$\text{weight}_i(t_i, s', t_{-i}) - \text{weight}_i(t_i, s'', t_{-i}) - \sqrt{\delta}(\log(2\ell) + 1),$$

where $\text{weight}_i(t_i, s, t_{-i}) = v_i(t_i, x(s, t_{-i})) - (1 - \eta) \cdot p_i(s, t_{-i})$ if $s \neq \diamond$, otherwise $\text{weight}_i(t_i, s, t_{-i}) = 0$.

In expectation over s' , s'' and t_{-i} ,

$$\mathbb{E}[q_i(t_i, \lambda)] = u_i(t_i, \mathcal{A}_\pi(W)) - u_i(t_i, \mathcal{A}_\pi(\lambda W_\pi, W_{-\pi})) - \sqrt{\delta}(\log(2\ell) + 1),$$

if we also take expectation over λ ,

$$\mathbb{E}_{\lambda \sim U[0,1]}[q_i(t_i, \lambda)] = u_i(t_i, \mathcal{A}_\pi(W)) - \int_0^1 u_i(t_i, \mathcal{A}_\pi(\lambda W_\pi, W_{-\pi})) d\lambda - \sqrt{\delta}(\log(2\ell) + 1).^{13}$$

With Definition 7, our mechanism is fully specified. We proceed to prove that the mechanism is BIC and IR. Our transformation is quite robust. Even if the original mechanism \mathcal{M} is not ε -BIC or the γ estimated in step 3 is not a constant factor approximation of $\frac{\text{OPT}(\omega(\mathbf{r}))}{d}$, the mechanism is still BIC and IR. The proof for truthfulness is similar to the one in Dughmi et al. [24]. However, as our edge weights may be negative, it is more challenging to establish the individually rationality compared to Dughmi et al. [24]. To make sure the mechanism is IR, we sometimes need to use negative payments to subsidize the agents, and at the same time guarantee that the total subsidy is negligible compared to the overall revenue. Note that this is also different from the previous ε -BIC to BIC transformations [23, 38, 18], as they essentially use the VCG mechanism to match surrogates to replicas, their mechanisms are clearly individually rational and use non-negative payments. The proof of Lemma 8 is postponed to Appendix B.

Lemma 8. *For any choice of the parameters ℓ , d , η , η' , δ and any IR mechanism \mathcal{M} , \mathcal{M}' is a BIC and IR mechanism w.r.t. \mathcal{D}' . In particular, we do not require \mathcal{M} to be ε -BIC. Moreover, each agent i 's expected Phase 1 payment $\mathbb{E}[q_i(t_i, \lambda)]$ is at least $-\sqrt{\delta}(\log(2\ell) + 1)$. Finally, on any input bid $b = (b_1, \dots, b_n)$, \mathcal{M}' computes the outcome in expected running time $\text{poly}(d, \ell, 1/\eta', 1/\delta)$ and makes in expectation at most $\text{poly}(d, \ell, 1/\eta', 1/\delta)$ queries to \mathcal{M} .*

We are now ready to present our main result for this section (Theorem 4). To prove the theorem, it suffices to lower bound the revenue of \mathcal{M}' from the second phase due to Lemma 8. In the previous transformations [23, 38, 18], the mechanism computes an exact maximum weight replica-surrogate matching, which allows them to bound the revenue from the second phase directly. Our mechanism only computes an approximately maximum weight replica-surrogate matching. As a result, we need to use a more delicate analysis to lower bound the revenue from Phase 2. We refer the readers to Appendix B.1 for more details about bounding the revenue loss and the complete proof of Theorem 4.

Theorem 4. *Let \mathcal{O} be a downward-closed outcome space. Given sample access to distributions $\mathcal{D} = \times_{i \in [n]} \mathcal{D}_i$ and $\mathcal{D}' = \times_{i \in [n]} \mathcal{D}'_i$, and query access to an ε -BIC and IR mechanism \mathcal{M} w.r.t. distribution \mathcal{D} . We can construct an exactly BIC and IR mechanism \mathcal{M}' w.r.t. distribution \mathcal{D}' , such that*

$$\text{REV}(\mathcal{M}', \mathcal{D}') \geq \text{REV}(\mathcal{M}, \mathcal{D}) - O(n\sqrt{\varepsilon}) - O\left(\sqrt{n \cdot d_w(\mathcal{D}, \mathcal{D}')}\right). \quad (7)$$

On any input bid $b = (b_1, \dots, b_n)$, \mathcal{M}' computes the outcome and payments in expected running time $\text{poly}(n, T', 1/\varepsilon)$ and makes in expectation at most $\text{poly}(n, T', 1/\varepsilon)$ queries to \mathcal{M} , where T'_i is the support of \mathcal{D}'_i and $T' = \max_{i \in [n]} |T'_i|$.

Furthermore, for any coupling $c_i(\cdot)$ between \mathcal{D}_i and \mathcal{D}'_i such that v_i is non-increasing w.r.t. $c_i(\cdot)$ ¹⁴, the error bound can be improved as follows:

¹³The difference between $\mathbb{E}_{\lambda \sim U[0,1]}[q_i(t_i, \lambda)]$ and Equation (6) is indeed a fixed constant, hence our mechanism is BIC.

¹⁴Roughly speaking, v_i is non-increasing w.r.t. a coupling c_i if the coupling always couples a ‘‘higher’’ type to a ‘‘lower’’ type. Namely, for all t_i , outcome $o \in \mathcal{O}$, if the coupling produces type t_i and $c_i(t_i)$, then $v_i(t_i, o) \geq v_i(c_i(t_i), o)$.

$$\text{REV}(\mathcal{M}', \mathcal{D}') \geq \text{REV}(\mathcal{M}, \mathcal{D}) - n\sqrt{\varepsilon} - O\left(n\eta + \frac{n\varepsilon}{\eta}\right) - \frac{\sum_{i \in [n]} \mathbb{E}_{t \sim \mathcal{D}'} [\mathbb{E}_{c_i(t_i)} [v_i(t_i, x'(t)) - v_i(c_i(t_i), x'(t))]]}{\eta}, \quad (8)$$

where $x'(\cdot)$ is the allocation rule of \mathcal{M}' and η can be chosen to be an arbitrary constant in $(0, 1)$.

Inequality (7) is our main result, and provides a strong guarantee in very general settings. Even though the difference between Inequality (8) and (7) seems small, we like to point out that the difference can be substantial sometimes and there were indeed cases where one needed a sharper version similar to Inequality (8). In particular, one common application of bounds similar to Inequality (8) is when the coupling simply rounds values down. For example, the main results in [18, 31] heavily rely on inequalities similar to Inequality (8), and these results may not be possible if only an Inequality (7) type bounds are used.

When $\mathcal{D} = \mathcal{D}'$, $d_w(\mathcal{D}, \mathcal{D}') = 0$, the following corollary states the ε -BIC to BIC transformation.

Corollary 1. *If $\mathcal{D} = \mathcal{D}'$, $\text{REV}(\mathcal{M}', \mathcal{D}) \geq \text{REV}(\mathcal{M}, \mathcal{D}) - O(n\sqrt{\varepsilon})$.*

Another useful corollary is when we choose \mathcal{M} to be the optimal BIC, IR mechanism for \mathcal{D} , we conclude that the optimal revenue under \mathcal{D}' is not far away from the optimal revenue under \mathcal{D} .

Corollary 2. *If $d_w(\mathcal{D}_i, \mathcal{D}'_i) \leq \kappa$ for all $i \in [n]$, let $\text{OPT}(\mathcal{D})$ and $\text{OPT}(\mathcal{D}')$ be the optimal revenue achievable by any BIC and IR mechanism w.r.t. \mathcal{D} and \mathcal{D}' respectively. Then $|\text{OPT}(\mathcal{D}) - \text{OPT}(\mathcal{D}')| \leq O(n \cdot \sqrt{\kappa})$.*

6 Black-box Reduction for Multi-Dimensional Revenue Maximization

In this section, we apply Theorem 4 to the multi-dimensional revenue maximization problem. Cai et al. [15] provide a reduction from MRM to VWO. More formally:

Theorem 5 (Rephrased from Theorem 2 of Cai et al. [15]). *Let \mathcal{O} be a general outcome space. Given the bidders' type distributions $\mathcal{D} = \times_i \mathcal{D}_i$. Let b be an upper bound on the bit complexity of $v_i(t_i, o)$ and $\Pr(t_i)$ for any agent i , any type t_i , and any outcome o , and OPT be the optimal revenue achievable by any BIC and IR mechanisms. We further assume that types are normalized, that is, for each agent i , type t_i and outcome o , $v_i(t_i, o) \in [0, 1]$.*

Given oracle access to an α -approximation algorithm G for VWO with running time $rt_G(x)$, where x is the bit complexity of the input, there is an algorithm that terminates in $\text{poly}(n, T, \frac{1}{\varepsilon}, b, rt_G(\text{poly}(n, T, \frac{1}{\varepsilon}, b)))$ time, and outputs a mechanism with expected revenue $\alpha \text{OPT} - \varepsilon$ that is ε -BIC with probability at least $1 - \exp(-n/\varepsilon)$. Recall that $T = \max_{i \in [n]} |\mathcal{T}_i|$. On any input bid, the mechanism computes the outcome and payments in expected running time $\text{poly}(n, T, \frac{1}{\varepsilon}, b, rt_G(\text{poly}(n, T, \frac{1}{\varepsilon}, b)))$.

We can apply Theorem 4 to the final mechanism produced by Theorem 5 and obtain an exactly BIC mechanism with almost the same revenue.

Theorem 6. *Let \mathcal{O} be a downward-closed outcome space. Given the bidders' type distributions $\mathcal{D} = \times_i \mathcal{D}_i$. Let b be an upper bound on the bit complexity of $v_i(t_i, o)$ and $\Pr(t_i)$ for any agent i , any type t_i , and any outcome o , and OPT be the optimal revenue achievable by any BIC and IR mechanisms. We further assume that types are normalized, that is, for each agent i , type t_i and outcome o , $v_i(t_i, o) \in [0, 1]$.*

*Given oracle access to an α -approximation algorithm G for VWO with running time $rt_G(x)$, where x is the bit complexity of the input, there is an algorithm that terminates in $\text{poly}(n, T, \frac{1}{\varepsilon}, b, rt_G(\text{poly}(n, T, \frac{1}{\varepsilon}, b)))$ time, and outputs an **exactly BIC** and IR mechanism with expected revenue*

$$\text{REV}(\mathcal{M}, \mathcal{D}) \geq \alpha \cdot \text{OPT} - O(n\sqrt{\varepsilon}),$$

where $T = \max_{i \in [n]} |\mathcal{T}_i|$. On any input bid, \mathcal{M} computes the outcome and payments in expected running time $\text{poly}\left(n, T, \frac{1}{\varepsilon}, b, \text{rt}_G\left(\text{poly}\left(n, T, \frac{1}{\varepsilon}, b\right)\right)\right)$.

Proof of Theorem 6: When the mechanism computed by Theorem 5 is ε -BIC, our transformation converts it into a BIC mechanism with at most $O(n\sqrt{\varepsilon})$ less revenue. The important property of our transformation as stated in Lemma 8 is that even if the initial mechanism is not ε -BIC, our transformation still produces an exactly BIC mechanism. In this case, we can still treat the given mechanism as 1-BIC and IR, and use the corresponding revenue guarantees provided by Theorem 4. Since the probability that the mechanism computed by Theorem 5 is not ε -BIC is exponentially small, we can absorb the loss from this exponentially small event in the error term $O(n\sqrt{\varepsilon})$. The time complexity follows from Theorem 4 and 5. \square

Since our Theorem 4 allows us to construct a close to optimal mechanism \mathcal{M}' w.r.t. the type distribution \mathcal{D}'_i , if \mathcal{D}' is not too far away from the distribution \mathcal{D} that \mathcal{M} is designed, we can approximate the optimal revenue even when we only have sample access to the bidders' type distributions. A byproduct of this result is that the running time of our algorithm no longer depends on the bit complexity of the probability that a particular type shows up.

Theorem 7. *Let \mathcal{O} be a downward-closed outcome space. Given **sample access** to bidders' type distributions $\mathcal{D} = \times_i \mathcal{D}_i$. Let b be an upper bound on the bit complexity of $v_i(t_i, o)$ and $\Pr(t_i)$ for any agent i , any type t_i , and any outcome o , and OPT be the optimal revenue achievable by any BIC and IR mechanism. We further assume that types are normalized, that is, for each agent i , type t_i and outcome o , $v_i(t_i, o) \in [0, 1]$.*

*Given oracle access to an α -approximation algorithm G for VWO with running time $\text{rt}_G(x)$, where x is the bit complexity of the input, there is an algorithm that terminates in $\text{poly}\left(n, T, \frac{1}{\varepsilon}, b, \text{rt}_G\left(\text{poly}\left(n, T, \frac{1}{\varepsilon}, b\right)\right)\right)$ time, and outputs an **exactly BIC and IR mechanism** with expected revenue*

$$\text{REV}(\mathcal{M}, \mathcal{D}) \geq \alpha \cdot \text{OPT} - O(n\sqrt{\varepsilon}),$$

where $T = \max_{i \in [n]} |\mathcal{T}_i|$. On any input bid, \mathcal{M} computes the outcome and payments in expected running time $\text{poly}\left(n, T, \frac{1}{\varepsilon}, b, \text{rt}_G\left(\text{poly}\left(n, T, \frac{1}{\varepsilon}, b\right)\right)\right)$.

Proof of Theorem 7: We can create an empirical distribution $\tilde{\mathcal{D}}_i$ for each bidder i , such that $d_{TV}(\mathcal{D}_i, \tilde{\mathcal{D}}_i) \leq \varepsilon', \forall i$, with probability at least $1 - \theta$ using $O\left(\sum_i^n \frac{|\mathcal{T}_i|^2}{\varepsilon'^2} \ln \frac{\sum_{j=1}^n |\mathcal{T}_j|}{\theta}\right)$ samples.

We first consider the case where $d_{TV}(\mathcal{D}_i, \tilde{\mathcal{D}}_i) \leq \varepsilon', \forall i$. Then, $d_w(\mathcal{D}_i, \tilde{\mathcal{D}}_i) \leq d_{TV}(\mathcal{D}_i, \tilde{\mathcal{D}}_i) \leq \varepsilon'$, as the highest value for any outcome is at most 1. Apply Theorem 5 on $\tilde{\mathcal{D}} = \times_{i=1}^n \tilde{\mathcal{D}}_i$ and let $\tilde{\mathcal{M}}$ be the produced mechanism, $\widetilde{\text{OPT}}$ be the optimal revenue achievable by any BIC and IR mechanism w.r.t. $\tilde{\mathcal{D}}$. Clearly, Theorem 5 guarantees that $\text{REV}(\tilde{\mathcal{M}}, \tilde{\mathcal{D}}) \geq \alpha \cdot \widetilde{\text{OPT}} - \varepsilon$. According to Corollary 2,

$$|\text{OPT} - \widetilde{\text{OPT}}| \leq O(n\sqrt{\varepsilon'}).$$

We set $\theta = \varepsilon$ and $\varepsilon' = \varepsilon$, we apply Theorem 4 to $\tilde{\mathcal{M}}$, that is, the replicas are sampled from \mathcal{D} and the surrogates are sampled from $\tilde{\mathcal{D}}$. Let \mathcal{M} be the constructed mechanism, and Theorem 4 guarantees that

$$\text{REV}(\mathcal{M}, \mathcal{D}) \geq \text{REV}(\tilde{\mathcal{M}}, \tilde{\mathcal{D}}) - O(n\sqrt{\varepsilon}) \geq \alpha \cdot \widetilde{\text{OPT}} - O(n\sqrt{\varepsilon}) \geq \alpha \cdot \text{OPT} - O(n\sqrt{\varepsilon}),$$

if $\tilde{\mathcal{M}}$ is a ε -BIC and IR mechanism. Otherwise, we know that $\tilde{\mathcal{M}}$ is a 1-BIC and IR mechanism, and this happens with exponentially small probability according to Theorem 5, so we can absorb the loss from this case in $O(n\sqrt{\varepsilon})$. To sum up, if $d_{TV}(\mathcal{D}_i, \tilde{\mathcal{D}}_i) \leq \varepsilon', \forall i$, then $\text{REV}(\mathcal{M}, \mathcal{D}) \geq \alpha \cdot \text{OPT} - O(n\sqrt{\varepsilon})$.

With probability ε we may get unlucky and $d_{TV}(\mathcal{D}_i, \tilde{\mathcal{D}}_i)$ may be larger than ε for some i . In that case we still construct \mathcal{M} in the same way, and we can apply Theorem 4 by upper bounding $d_w(\mathcal{D}, \tilde{\mathcal{D}})$ by n and treating $\tilde{\mathcal{M}}$ as a 1-BIC and IR mechanism, which shows $\text{REV}(\mathcal{M}, \mathcal{D}) \geq -O(n)$.

Therefore, in expectation of the randomness of the samples used to estimate $\tilde{\mathcal{D}}$,

$$\text{REV}(\mathcal{M}, \mathcal{D}) \geq (1 - \varepsilon) \cdot (\alpha \cdot \text{OPT} - O(n\sqrt{\varepsilon})) - O(n\varepsilon) = \alpha \cdot \text{OPT} - O(n\sqrt{\varepsilon}),$$

as $\text{OPT} \leq n$. Note that even though mechanism \mathcal{M} depends on $\tilde{\mathcal{D}}$ and $\tilde{\mathcal{M}}$, it is always BIC and IR w.r.t. \mathcal{D} . The time complexity follows from Theorem 4 and 5. \square

7 General Outcome Space: Regularized Replica-Surrogate Fractional Assignment

In this section we consider general outcome space \mathcal{O} , removing the assumption that \mathcal{O} is downward-closed. We first demonstrate in Example 3 that only having sample access to the agents' type distributions is not sufficient to transform an ε -BIC mechanism to exactly BIC, with a small revenue loss.

Example 3. Consider the following ε -BIC to BIC transformation instance for a single buyer. Let

$$\mathcal{T} = \{t_H, t_{(l,1)}, \dots, t_{(l,m)}, t_{(s,1)}, \dots, t_{(s,m)}\} \in \mathbb{N},$$

be the set of the buyer's types, $\mathcal{D} = \{\mathcal{D}_{(i,j)}\}_{i,j \in [m]}$, where $\mathcal{D}_{(i,j)}$ is the distribution with support $\{t_H, t_{(l,i)}, t_{(s,j)}\}$ and point-mass probability:

$$\Pr_{t' \sim \mathcal{D}_{(i,j)}} [t' = t] = \begin{cases} 1 - 2\sigma & \text{if } t = t_H \\ \sigma & \text{if } t = t_{(l,i)} \\ \sigma & \text{if } t = t_{(s,j)} \end{cases}$$

The output space is $\mathcal{O} = \{o_H, o_{(l,1)}, \dots, o_{(l,m)}, o_{(s,1)}, \dots, o_{(s,m)}\}$ such that:

$$\begin{aligned} v(t_H, o) &= \begin{cases} 1 & \text{if } o = o_H \\ 0 & \text{o.w.} \end{cases} \\ v(t_{(l,i)}, o) &= \begin{cases} \epsilon & \text{if } o = o_{(s,j')}, \text{ for any } j' \in [m] \\ 0 & \text{o.w.} \end{cases} \\ v(t_{(s,j)}, o) &= \begin{cases} \epsilon & \text{if } o = o_{(l,i')}, \text{ for any } i' \in [m] \\ 0 & \text{o.w.} \end{cases} \end{aligned}$$

For each distribution $\mathcal{D}_{(i,j)} \in \mathcal{D}$ we consider the mechanism $\mathcal{M}_{(i,j)}$: if the buyer reports t_H , the mechanism outputs o_H and charges the buyer 1, if the buyer reports $t_{(l,i)}$, the mechanism outputs $o_{(l,i)}$ and charges the buyer 0, if the buyer reports $t_{(s,j)}$, the mechanism outputs $o_{(s,j)}$ and charges the buyer 0. If the buyer reports anything else, the mechanism outputs o_H and charges the buyer 1. Mechanism $\mathcal{M}_{(i,j)}$ is ε -BIC and IR for distribution $\mathcal{D}_{(i,j)}$. The revenue of \mathcal{M} is $1 - 2\sigma$.

The mechanism designer has oracle access to set \mathcal{T} and faces the following problem: There is an arbitrary distribution from the whole space \mathcal{D} , which is unknown to the designer. The buyer's valuation distribution is realized to be some $\mathcal{D}_{(i,j)} \in \mathcal{D}$, the designer is given sample access to $\mathcal{D}_{(i,j)}$, oracle access to

$\mathcal{M}_{(i,j)}$ ¹⁵ and she needs to output a truly BIC and IR mechanism \mathcal{M}' w.r.t. the buyer's realized distribution that approximately preserves the revenue. Note that $|\mathcal{T}| = 2m + 1$. Therefore the size of each $t \in \mathcal{T}$ is at most $O(\log(m))$.

In Lemma 9, we prove that with proper choice of σ , any BIC and IR mechanism with $\text{poly}(\log(m))$ samples from the type distribution $\mathcal{D}_{(i,j)}$ and queries from $\mathcal{M}_{(i,j)}$ has revenue far less than $\mathcal{M}_{(i,j)}$.

Lemma 9. *For any $\delta > 0$ choose $\sigma < \frac{\delta}{2N}$ and $m \geq \frac{2N}{\delta}$. For any i, j , let \mathcal{M}' be any BIC, IR mechanism w.r.t. $\mathcal{D}_{(i,j)}$, which only uses $N = \text{poly}(\log(m))$ samples from $\mathcal{D}_{(i,j)}$ and queries from $\mathcal{M}_{(i,j)}$ (notice that $\log(m)$ is the size of the input). Then $\text{REV}(\mathcal{M}') < 2\delta + 3\delta \cdot \epsilon$. As δ goes to 0, $\text{REV}(\mathcal{M}')$ goes to 0 while $\text{REV}(\mathcal{M}) = 1 - 2\sigma \geq 1 - \frac{\delta}{N}$ goes to 1.*

Proof. Denote $\mathcal{M}'(t)$ the output of mechanism \mathcal{M}' if the agent reports type t and $p(t)$ the expected payment if the agent reports t . Notice that the outcome of $\mathcal{M}_{(i,j)}$ can only be $o_H, o_{(l,i)}$, or $o_{(s,j)}$. Thus the output of mechanism \mathcal{M}' is also one of these three outcomes. Let $\mathcal{M}'(t)$ for inputs $t_{(l,i)}, t_{(s,j)}$ be the following distributions

$$\mathcal{M}'(t_{(l,i)}) = \begin{cases} o_H & \text{with probability } 1 - p - q \\ o_{(l,i)} & \text{with probability } p \\ o_{(s,j)} & \text{with probability } q \end{cases}$$

$$\mathcal{M}'(t_{(s,j)}) = \begin{cases} o_H & \text{with probability } 1 - p' - q' \\ o_{(s,j)} & \text{with probability } p' \\ o_{(l,i)} & \text{with probability } q' \end{cases}$$

Then by BIC constraint we have that:

$$q \cdot \epsilon - p(t_{(l,i)}) \geq p' \cdot \epsilon - p(t_{(s,j)}) \quad \text{By BIC constraint if agent's true type is } t_{(l,i)} \text{ but reports } t_{(s,j)}.$$

$$q' \cdot \epsilon - p(t_{(s,j)}) \geq p \cdot \epsilon - p(t_{(l,i)}) \quad \text{By BIC constraint if agent's true type is } t_{(s,j)} \text{ but reports } t_{(l,i)}.$$

By the previous inequalities we can infer that:

$$q - p' \geq \frac{p(t_{(l,i)}) - p(t_{(s,j)})}{\epsilon} \geq p - q'$$

Notice that either $p \leq q'$ or $p' \leq q$. The claim holds because if $p(t_{(l,i)}) - p(t_{(s,j)})$ is non-negative, then $q \geq p'$ and if $p(t_{(l,i)}) - p(t_{(s,j)})$ is negative we have that $p \leq q'$. Without loss of generality assume $p' \leq q$.

Now we are going to bound q . Note that since the mechanism designer does not have access to the outcome space, the only way to find outcome $o_{(s,j)}$ is to use $t_{(s,j)}$ as input to the mechanism \mathcal{M} . To do that, the designer must sample $t_{(s,j)}$ either directly from the type space \mathcal{T} or the realized distribution $\mathcal{D}_{(i,j)}$ ¹⁶.

With at most N samples from the buyer's distribution $\mathcal{D}_{(i,j)}$, with probability at least $(1 - \sigma)^N \geq 1 - N\sigma$, none of the sampled types is $t_{(s,j)}$. By assumption, $\sigma \leq \frac{\delta}{2N}$ which implies that with probability at least $1 - \frac{\delta}{2}$ we are not be able to sample type $t_{(s,j)}$, therefore we cannot find output $o_{(s,j)}$ by this method.

Moreover, notice that the output of \mathcal{M} is o_H for all types $\{t_{s,j'}\}_{j \neq j' \in [m]}$. Thus by querying mechanism \mathcal{M} with at most N types $t_{s,j'}$ chosen uniformly at random from $\{t_{(s,j)}\}_{j \in [m]}$, with probability at least

¹⁵In other words the mechanism designer does not know the outcome space \mathcal{O} . It can only put a input type into the mechanism and outputs the returned outcome.

¹⁶Note that if the designer does not sample uniformly at random from the set $\{t_{(s,j)}\}_{j \in [m]}$, since the distribution over \mathcal{D} is unknown, the adversary can make it more difficult for her to sample the right type by picking a different distribution

$(1 - 1/m)^N \geq 1 - \frac{N}{m}$, none of the returned outcomes is $o_{(s,j)}$. By assumption $m \geq \frac{2N}{\delta}$ implies that with probability at least $1 - \frac{\delta}{2}$ none of the returned outcomes is $o_{(s,j)}$.

Hence, for any $\delta > 0$, by taking Union Bound over the two events described above, we cannot identify output $o_{(s,j)}$ with probability at least $1 - \delta$. This implies $p' \leq \delta$. Similarly we can prove that $q' \leq \delta$.

By IR constraints for type $t_{(s,j)}$, we have $q' \cdot \epsilon - p(t_{(s,j)}) \geq 0$. Thus $-p(t_{(s,j)}) \geq -q' \cdot \epsilon$. By $u(t_H)$ we denote the utility of type t_H in mechanism \mathcal{M}' when she reports t_H . Consider the BIC constraint for type t_H when she reports type $t_{(s,j)}$. We have

$$u(t_H) \geq (1 - p' - q') - p(t_{(s,j)}) \geq 1 - p' - q' - q' \cdot \epsilon \geq 1 - 2\delta - \delta \cdot \epsilon$$

Thus $p(t_H) \leq 2\delta + \delta \cdot \epsilon$. Notice that both $p(t_{(s,j)})$ and $p(t_{(l,i)})$ are at most ϵ since the agent's value under these two types are at most ϵ . Thus $\text{REV}(\mathcal{M}') \leq 2\delta + 3\delta \cdot \epsilon$. \square

Throughout this section, we will assume that we have full access to the agents' type distributions: Assume every agent's type distribution is discrete with finite support. We denote the support of the distribution of the k -th agent as $\{t_k^{(i)}\}_{i \in [m_k]}$ and the probability that the k -th agent has type $t_k^{(i)}$ as F_i^k . When there is no confusion about the agent that we are referring to, we may drop the superscript k in F_i^k . The constructed mechanism knows $t_k^{(i)}$ and the exact value F_i^k for every $i \in [m_k]$.

We first present the result in the ideal model, where the edge weights are known exactly. Then we show that if we only estimate the edge weights approximately, we can use an approach similar to [28] to still guarantee the mechanism is BIC and IR.

7.1 Regularized Replica-Surrogate Fractional Assignment

We now present the **Regularized Replica-Surrogate Fractional Assignment (RRSF)** introduced by [28]. There are two main differences between RRSF and the replica-surrogate matching: (i) the replicas and surrogates in RRSF are no longer samples from the agent's type distribution; for each type of the agent, there is exactly one replica and one surrogate of that type in RRSF; (ii) RRSF finds the optimal fractional assignment rather than a maximum weight matching.

Definition 8 (Definition 4.6 of [28]). *Let $\phi(\mathbf{x}) = \frac{1}{2}\gamma\|\mathbf{x}\|_2^2$. For agent $k \in [n]$ and $i, j \in [m_k]$, let W_{ij}^k be any value in $[-1, 1]$. We may drop the superscript when agent k is fixed and clear from context. Consider the following convex program (P^3) with coefficients $\{W_{i,j}\}_{i,j \in [m_k]}$, where the decision variables are $\{q_{i,j}\}_{i,j \in [m_k]}$.*

$$\begin{aligned} & \max \sum_{i,j \in [m_k]} F_i (W_{i,j} q_{i,j} - \phi(\mathbf{q}_i)) \\ & \text{subject to } \sum_j q_{i,j} = 1, \quad \forall i \in [m_k] \\ & \sum_i F_i q_{i,j} = F_j, \quad \forall j \in [m_k] \\ & q_{i,j} \geq 0, \quad \forall i, j \in [m_k] \end{aligned} \tag{9}$$

Let $\mathbf{q}_k^* = \{q_{i,j}^*\}_{i,j \in [m_k]}$ be an optimal solution to the primal problem and $\boldsymbol{\lambda}_k^* = \{\lambda_{i,j}^*\}_{i,j \in [m_k]}$, $\boldsymbol{\mu}_k^* = \{\mu_i^*\}_{i \in [m_k]}$, $\boldsymbol{\pi}_k^* = \{\pi_j^*\}_{j \in [m_k]}$ be any Lagrange multipliers satisfying the KKT condition

1. $F_i \left(W_{i,j} - \frac{\partial \phi(\mathbf{q}_i^*)}{\partial q_{i,j}^*} \right) = \lambda_{i,j}^* + \mu_i^* + F_i \pi_j^*$, $\forall i, j$
2. $\lambda_{i,j}^* \leq 0$, $\forall i, j$
3. $\lambda_{i,j}^* q_{i,j}^* = 0$, $\forall i, j$

7.2 Ideal Model

RRSF Mechanism with parameters ($\mathbf{q}^* = \{q_k^*\}_{k \in [n]}$, $\boldsymbol{\mu}^* = \{\mu_k^*\}_{k \in [n]}$, $\boldsymbol{\pi}^* = \{\pi_k^*\}_{k \in [n]}$): every agent k reports her type $t_k = t_k^{(i)}$. Let s_k be a random surrogate type such that $\Pr[s_k = t_k^{(j)}] = q_{i,j}^*$ for every $j \in [m_k]$.

The RRSF mechanism \mathcal{M}' runs mechanism \mathcal{M} under the random input $s = (s_1, \dots, s_n)$. The outcome for \mathcal{M}' is the random outcome o generated from $x(s)$ and agent k 's expected payment is $p_k(s)$ plus some extra payment \hat{p}_k , where $\hat{p}_k(t_k) = \sum_j \pi_j^* q_{i,j}^* + \phi(q_k^*) - \phi(\mathbf{0}) + \min_\ell \frac{\mu_\ell^*}{F_\ell}$. We call \mathcal{M}' the RRSF mechanism with respect to $\mathbf{W} = \{W_{i,j}^k\}_{k \in [n], i, j \in [m_k]}$ if for every k , the parameters $(q_k^*, \mu_k^*, \pi_k^*)$ are obtained by the convex program (P^3) with coefficients $\{W_{i,j}^k\}_{i, j \in [m_k]}$.

Similar to [28], in our proof each $W_{i,j}^k$ will be chosen to be the expected utility of agent k for the outcome of the mechanism $\mathcal{M} = (x, p)$ when her type is $t_k^{(i)}$ and she is matched to the surrogate of type $t_k^{(j)}$. Formally,

$$W_{i,j}^k = \mathbb{E}_{t_{-k} \sim \mathcal{D}_{-k}} \left[v_k(t_k^{(i)}, x(t_k^{(j)}, t_{-k})) \right] - \mathbb{E}_{t_{-k} \sim \mathcal{D}_{-k}} \left[p_k(t_k^{(j)}, t_{-k}) \right]$$

They proved that the RRSF mechanism with respect to the \mathbf{W} defined above is BIC and IR.

Theorem 8 ([28]). *Given any mechanism \mathcal{M} . For every agent k and $i, j \in [m_k]$, let*

$$W_{i,j}^k = \mathbb{E}_{t_{-k} \sim \mathcal{D}_{-k}} \left[v_k(t_k^{(i)}, x(t_k^{(j)}, t_{-k})) \right] - \mathbb{E}_{t_{-k} \sim \mathcal{D}_{-k}} \left[p_k(t_k^{(j)}, t_{-k}) \right]$$

Then RRSF mechanism \mathcal{M}' defined in Definition 8 is BIC and IR.

As Hartline et al. [28] only care about the welfare of \mathcal{M}' , Theorem 8 suffices. We care about the revenue of \mathcal{M}' , so we need to argue that running \mathcal{M}' does not cause the designer to lose a large fraction of the revenue. $\text{REV}(\mathcal{M}')$ contains two parts: (i) the expected revenue from payments $\{p_k(\cdot)\}_{k \in [n]}$, which is exactly the same as $\text{REV}(\mathcal{M})$; (ii) the expected payments from $\{\hat{p}_k(\cdot)\}_{k \in [n]}$. To prove that $\text{REV}(\mathcal{M}')$ is not too much smaller than $\text{REV}(\mathcal{M})$, we need to prove that the expected payments from $\{\hat{p}_k(\cdot)\}_{k \in [n]}$ are not too negative. We prove this claim with the following sequence of lemmas.

We will prove a more general result for any RRSF mechanism with respect to \mathbf{W} that satisfies: $W_{i,i}^k \geq \max\{W_{i,j}^k - \varepsilon', -\varepsilon''\}$, $\forall k \in [n], i, j \in [m_k]$, for some $\varepsilon', \varepsilon'' \geq 0$. Note that when \mathbf{W} is chosen as in Theorem 8 and \mathcal{M} is ε -BIC and IR, we will have $\varepsilon' = \varepsilon$ and $\varepsilon'' = 0$. The more general result is useful in the proof of the non-ideal model in Section 7.3.

We first prove a structural result about the optimal solution of the convex program (P^3).

Lemma 10. *Fix any agent k . Suppose for all $i, j \in [m_k]$, $W_{i,i}^k \geq \max\{W_{i,j}^k - \varepsilon', -\varepsilon''\}$ holds for some $\varepsilon', \varepsilon'' \geq 0$. Let $\{q_{i,j}^*\}_{i, j \in [m_k]}$ be an optimal solution of the convex program (P^3). Then, it holds that*

$$q_{i,j}^* > 0 \implies W_{i,j} \geq W_{i,i} - m\varepsilon' - \sqrt{2}m\gamma, \quad \forall i, j \in [m_k].$$

Proof. For any type $t_k^{(i)} \in \text{supp}(\mathcal{D}_k)$ such that $q_{i,i}^* = 1$ the statement holds.

Now assume there are some types $t_k^{(i^*)} \neq t_k^{(j^*)}$ such that $q_{i^*,j^*}^* > 0$. A sequence of types $S = \{t_k^{(a_i)}\}_{i=1}^b$ of length b , where $t_k^{(a_i)} \in \text{supp}(\mathcal{D}_k)$, is called a flow sequence if for each $1 \leq i \leq b-1$ it holds that $q_{a_i, a_{i+1}} > 0$. Let V_{j^*} be the set of types (including $t_k^{(j^*)}$) such that there exists some flow sequence that starts with type $t_k^{(j^*)}$ that reaches them. We are going to prove that $t_k^{(i^*)} \in V_{j^*}$.

Note that for each $t_k^{(r)} \in V_{j^*}$, if $q_{r,\ell} > 0$ then $t_k^{(\ell)} \in V_{j^*}$. Thus the set $U_{j^*} = \{t_k^{(\ell)} : \exists t_k^{(r)} \in V_{j^*} \wedge q_{r,\ell} > 0\} \subseteq V_{j^*}$. Since \mathbf{q}^* is feasible, we have

$$\sum_{t_k^{(r)} \in V_{j^*}} F_r = \sum_{t_k^{(r)} \in V_{j^*}} \sum_{\ell \in [m_k]} F_r q_{r,\ell} = \sum_{t_k^{(r)} \in V_{j^*}} \sum_{\ell: q_{r,\ell} > 0} F_r q_{r,\ell} = \sum_{t_k^{(\ell)} \in U_{j^*}} \sum_{t_k^{(r)} \in V_{j^*}} F_r q_{r,\ell}.$$

On the other hand,

$$\sum_{t_k^{(\ell)} \in U_{j^*}} \sum_{t_k^{(r)} \in V_{j^*}} F_r q_{r,\ell} \leq \sum_{t_k^{(\ell)} \in V_{j^*}} \sum_{t_k^{(r)} \in V_{j^*}} F_r q_{r,\ell} \leq \sum_{t_k^{(\ell)} \in V_{j^*}} \sum_{r \in [m_k]} F_r q_{r,\ell} = \sum_{t_k^{(\ell)} \in V_{j^*}} F_\ell.$$

Therefore, we must have $U_{j^*} = V_{j^*}$ and for every $t_k^{(\ell)} \in V_{j^*}$, $t_k^{(r)} \notin V_{j^*}$, $q_{r,\ell} = 0$. Since $q_{i^*,j^*} > 0$, we have $t_k^{(i^*)} \in V_{j^*}$. Let $S_{i^*,j^*} = \{t_k^{a_1}, \dots, t_k^{a_b}\}$ such that $a_1 = j^*$ and $a_b = i^*$ be the shortest flow sequence from $t_k^{(j^*)}$ to $t_k^{(i^*)}$. Then each type appears in S_{i^*,j^*} at most once. Let $f_* = \min_{k \in [b]} F_{a_k} q_{a_k, a_{k+1}}^* \geq 0$ (here $a_{b+1} = a_1$). We consider the following new solution $\hat{\mathbf{q}}$:

$$\hat{q}_{i,j} = \begin{cases} q_{i,j}^* - f_*/F_i, & \text{when there exists } \ell \text{ s.t. } a_\ell = i \text{ and } a_{\ell+1} = j \\ q_{i,i}^* + f_*/F_i & \text{when } i = j \text{ and there exists } \ell \text{ s.t. } a_\ell = i \\ q_{i,j}^* & \text{o.w.} \end{cases}$$

We will verify that $\hat{\mathbf{q}}$ is a feasible solution to RRSF. For the first set of constraints: for every a_ℓ , notice that (i) $\hat{q}_{a_\ell, a_\ell} = q_{a_\ell, a_\ell}^* + f_*/F_{a_\ell}$ and (ii) there exists a unique j_0 such that $\hat{q}_{a_\ell, j_0} = q_{a_\ell, j_0}^* - f_*/F_{a_\ell}$. Thus $\sum_j \hat{q}_{i,j} = \sum_j q_{i,j}^* = 1$. For the second set of constraints: for every $j = a_{\ell+1}$ for some $\ell \in [b]$, notice that there exists a unique $i_0 = a_\ell$ such that $\hat{q}_{i_0, j} = q_{i_0, j}^* - f_*/F_{i_0}$, we have

$$\sum_i F_i \hat{q}_{i,j} = \sum_i F_i q_{i,j}^* - F_{i_0} \cdot \frac{f_*}{F_{i_0}} + F_j \cdot \frac{f_*}{F_j} = \sum_i F_i q_{i,j}^* = F_j$$

For the third set of constraints: for every pair (i, j) , either $\hat{q}_{i,j} \geq q_{i,j}^* \geq 0$ (the second and third case), or $\hat{q}_{i,j} = q_{i,j}^* - \frac{f_*}{F_i} \geq 0$ as $f_* \leq F_i q_{i,j}^*$.

Therefore $\hat{\mathbf{q}}$ is a feasible solution to RRSF. Moreover since \mathbf{q}^* is the optimal feasible solution, its objective should be at least the objective value of $\hat{\mathbf{q}}$. Observe that for every $i \in [m_k]$ and every edge (i, j) with $j \neq i$ the value $\hat{q}_{i,j}$ is not equal to $q_{i,j}^*$ only when there is some ℓ with $a_\ell = i, a_{\ell+1} = j$. The value of this edge drops by f_*/F_i , whereas the value of $q_{i,i}$ increases by f_*/F_i . Denote $V(\mathbf{q}^*), V(\hat{\mathbf{q}})$ the value of the objectives for these two feasible solutions respectively. In the flow sequence $S = S_{i^*,j^*}$, for every $i = a_\ell$ for some $\ell \in [b]$, denote $i' = a_{\ell+1}$. We have

$$\begin{aligned}
0 &\leq V(\mathbf{q}^*) - V(\hat{\mathbf{q}}) \\
&= \sum_i F_i \left(\sum_j W_{i,j} (q_{i,j}^* - \hat{q}_{i,j}) + (\phi(\hat{\mathbf{q}}_i) - \phi(\mathbf{q}_i^*)) \right) \\
&= \sum_{i: t_k^{(i)} \in S_{i^*, j^*}} F_i \left(W_{i,i'} \frac{f^*}{F_i} - W_{i,i} \frac{f^*}{F_i} + (\phi(\hat{\mathbf{q}}_i) - \phi(\mathbf{q}_i^*)) \right) \quad (\hat{\mathbf{q}}_i = \mathbf{q}_i^*, \text{ if } t_k^{(i)} \notin S_{i^*, j^*}) \\
&= f^* \sum_{i: t_k^{(i)} \in S_{i^*, j^*}} \left(W_{i,i'} - W_{i,i} + \frac{F_i}{f^*} (\phi(\hat{\mathbf{q}}_i) - \phi(\mathbf{q}_i^*)) \right) \\
&\leq f^* \sum_{i: t_k^{(i)} \in S_{i^*, j^*}} \left(W_{i,i'} - W_{i,i} + \frac{F_i}{f^*} \nabla \phi(\hat{\mathbf{q}}_i)^T (\hat{\mathbf{q}}_i - \mathbf{q}_i^*) \right) \quad (\phi(\cdot) \text{ is convex}) \\
&\leq f^* \sum_{i: t_k^{(i)} \in S_{i^*, j^*}} \left(W_{i,i'} - W_{i,i} + \frac{F_i}{f^*} \|\nabla \phi(\hat{\mathbf{q}}_i)\|_2 \|\mathbf{q}_i^* - \hat{\mathbf{q}}_i\|_2 \right) \quad (\text{Cauchy-Schwarz Inequality}) \\
&= f^* \sum_{i: t_k^{(i)} \in S_{i^*, j^*}} \left(W_{i,i'} - W_{i,i} + \sqrt{2} \|\nabla \phi(\hat{\mathbf{q}}_i)\|_2 \right) \quad (\|\mathbf{q}_i^* - \hat{\mathbf{q}}_i\|_2^2 = 2 \cdot \left(\frac{f^*}{F_i}\right)^2) \\
&= f^* \sum_{i: t_k^{(i)} \in S_{i^*, j^*}} \left(W_{i,i'} - W_{i,i} + \sqrt{2} \gamma \|\hat{\mathbf{q}}_i\|_2 \right) \quad (\text{the definition of } \phi(\cdot)) \\
&\leq f^* \sum_{i: t_k^{(i)} \in S_{i^*, j^*}} \left(W_{i,i'} - W_{i,i} + \sqrt{2} \gamma \right) \quad (\hat{\mathbf{q}}_i \text{ is a distribution}).
\end{aligned}$$

Since $f^* \geq 0$ we have

$$\sum_{i: t_k^{(i)} \in S_{i^*, j^*}} \left(W_{i,i'} - W_{i,i} + \sqrt{2} \gamma \right) \geq 0.$$

Thus

$$\begin{aligned}
W_{i^*, j^*} &\geq W_{i^*, i^*} - \sqrt{2} \gamma - \sum_{i: t_k^{(i)} \in S_{i^*, j^*} - \{t_k^{(i^*)}\}} \left(W_{i,i'} - W_{i,i} + \sqrt{2} \gamma \right) \\
&\geq W_{i^*, i^*} - \sqrt{2} \gamma - \sum_{i: t_k^{(i)} \in S_{i^*, j^*} - \{t_k^{(i^*)}\}} \left(\varepsilon' + \sqrt{2} \gamma \right) \quad (W_{i,i} \geq W_{i,i'} - \varepsilon' \text{ by assumption}) \\
&\geq W_{i^*, i^*} - m \varepsilon' - \sqrt{2} m \gamma
\end{aligned}$$

□

Equipped with Lemma 10, we proceed to show that the convex program (P^3) has a set of optimal Lagrange multipliers that will make the total expected payments from $\{\hat{p}_k\}_{k \in [n]}$ not too negative. First, we construct a new convex program (P^4) that “removes” all the very negative edge weights from (P^3) , that

is, change every edge (i, j) 's weight to $\max\{W_{i,j}, -m(\varepsilon' + 2\gamma) - \varepsilon''\}$ (see Definition 9). Then, we argue that convex program (P^3) and (P^4) are essentially equivalent. Namely, any optimal solution of (P^4) is also an optimal solution for (P^3) , and there is a straightforward mapping that transforms any optimal Lagrange multipliers of (P^4) to a set of optimal Lagrange multipliers for (P^3) (see Lemma 11). Finally, we show that if we compute $\hat{p}_k(\cdot)$ using any set of the optimal Lagrange multipliers derived from convex program (P^4) , then $\hat{p}_k(\cdot)$ is not too negative (see Lemma 12) The main reason is that using the KKT condition, we can relate $\hat{p}_k(\cdot)$ to the edge weights, and the edge weights in (P^4) are not too small.

Definition 9. Fix an agent k and $\varepsilon' \geq \varepsilon'' \geq 0$. For every i, j , define

$$\hat{W}_{i,j} = \begin{cases} W_{i,j} & \text{if } W_{i,j} \geq -m(\varepsilon' + 2\gamma) - \varepsilon'' \\ -m(\varepsilon' + 2\gamma) - \varepsilon'' & \text{if } W_{i,j} < -m(\varepsilon' + 2\gamma) - \varepsilon'' \end{cases}$$

We solve the following convex program (P^4) where the decision variables are $\{q_{i,j}\}_{i,j \in [m_k]}$.

$$\begin{aligned} \max \quad & \sum_{i,j} F_i \left(\hat{W}_{i,j} q_{i,j} - \phi(\mathbf{q}_i) \right) \\ \text{subject to} \quad & \sum_j q_{i,j} = 1, \quad \forall i \\ & \sum_i F_i q_{i,j} = F_j, \quad \forall j \\ & q_{i,j} \geq 0, \quad \forall i, j \end{aligned} \tag{10}$$

Let $\hat{\mathbf{q}}^*$ be an optimal solution of (P^4) and $\hat{\lambda}^*, \hat{\mu}^*, \hat{\pi}^*$ be any Lagrange multipliers satisfying the KKT conditions:

1. $F_i \left(\hat{W}_{i,j} - \frac{\partial \phi(\hat{\mathbf{q}}_i^*)}{\partial q_{i,j}^*} \right) = \hat{\lambda}_{i,j}^* + \hat{\mu}_i^* + F_i \hat{\pi}_j^*, \quad \forall i, j$
2. $\hat{\lambda}_{i,j}^* \leq 0, \quad \forall i, j$
3. $\hat{\lambda}_{i,j}^* \hat{q}_{i,j}^* = 0, \quad \forall i, j$

Lemma 11. Fix any agent k . Suppose for all $i, j \in [m_k]$, $W_{i,i}^k \geq \max\{W_{i,j}^k - \varepsilon', -\varepsilon''\}$ holds for some $\varepsilon' \geq \varepsilon'' \geq 0$. Let $\hat{\mathbf{q}}^* = \{\hat{q}_{i,j}^*\}_{i,j \in [m_k]}$ be any optimal solution of the convex program (P^4) . Then $\hat{\mathbf{q}}^*$ is also an optimal solution of the convex program (P^3) . Moreover let $\hat{\lambda}^*, \hat{\mu}^*, \hat{\pi}^*$ be any Lagrange multipliers satisfying the KKT conditions w.r.t. the convex program (P^4) . Then there exists $\lambda^* = \{\lambda_{i,j}^*\}_{i,j \in [m_k]}$ such that $(\lambda^*, \mu^* = \hat{\mu}^*, \pi^* = \hat{\pi}^*)$ satisfies the KKT conditions w.r.t. the convex program (P^3) .

Proof. We first prove that $\hat{\mathbf{q}}^*$ is also an optimal solution of the convex program (P^3) . Clearly $\hat{\mathbf{q}}^*$ is a feasible solution of the convex program (P^3) . To argue $\hat{\mathbf{q}}^*$ is an optimal solution, we first prove that $\hat{W}_{i,i} \geq \hat{W}_{i,j} - \varepsilon', \forall i, j \in [m_k]$. For every i , Since $W_{i,i} \geq -\varepsilon''$, $\hat{W}_{i,i} = W_{i,i} \geq -\varepsilon''$. If $W_{i,j} \geq -m(\varepsilon' + 2\gamma) - \varepsilon''$, then $\hat{W}_{i,j} = W_{i,j}$ and we have $\hat{W}_{i,i} \geq \hat{W}_{i,j} - \varepsilon'$ since $W_{i,i} \geq W_{i,j} - \varepsilon'$. On the other hand, suppose $W_{i,j} < -m(\varepsilon' + 2\gamma) - \varepsilon''$. Then $\hat{W}_{i,j} = -m(\varepsilon' + 2\gamma) - \varepsilon'' \leq 0$. $\hat{W}_{i,i} - \hat{W}_{i,j} \geq W_{i,i} \geq -\varepsilon'' \geq -\varepsilon'$.

We apply Lemma 10 to convex program (P^4) . For any i, j such that $\hat{q}_{i,j}^* > 0$, $\hat{W}_{i,j} \geq \hat{W}_{i,i} - m(\varepsilon' + \sqrt{2}\gamma) \geq -m(\varepsilon' + \sqrt{2}\gamma) - \varepsilon''$. Thus it must hold that $\hat{W}_{i,j} = W_{i,j}$ and the solution $\hat{\mathbf{q}}^*$ has the same objective value in (P^3) and (P^4) . Moreover since $\hat{W}_{i,j} \geq W_{i,j}$ for every i, j , the optimal objective value of (P^3) is no larger than the optimal objective value of (P^4) . Thus $\hat{\mathbf{q}}^*$ is an optimal solution of the convex program (P^3) .

For the second part of the statement, consider the following dual variables $\{\lambda_{i,j}^*\}_{i,j \in [m,k]}$:

$$\lambda_{i,j}^* = \begin{cases} \hat{\lambda}_{i,j}^* & \text{if } W_{i,j} \geq -m(\varepsilon' + 2\gamma) - \varepsilon'' \\ \hat{\lambda}_{i,j}^* + F_i(W_{i,j} + m(\varepsilon' + 2\gamma) + \varepsilon'') & \text{o.w.} \end{cases}$$

We now verify that $(\lambda^*, \mu^* = \hat{\mu}^*, \pi^* = \hat{\pi}^*)$ satisfies the KKT conditions w.r.t. the convex program (P^3) . When $\hat{W}_{i,j} \geq -m(\varepsilon' + 2\gamma) - \varepsilon''$ we have that $\lambda_{i,j}^* = \hat{\lambda}_{i,j}^*$ and $\hat{W}_{i,i} = W_{i,i}$. This implies that:

1. $F_i \left(\hat{W}_{i,j} - \frac{\partial \phi(\hat{q}_{i,j}^*)}{\partial \hat{q}_{i,j}^*} \right) = \hat{\lambda}_{i,j}^* + \hat{\mu}_i^* + F_i \hat{\pi}_j^* \Leftrightarrow F_i \left(W_{i,j} - \frac{\partial \phi(\hat{q}_{i,j}^*)}{\partial \hat{q}_{i,j}^*} \right) = \lambda_{i,j}^* + \hat{\mu}_i^* + F_i \hat{\pi}_j^*$
2. $\hat{\lambda}_{i,j}^* \leq 0$
3. $\hat{\lambda}_{i,j}^* \hat{q}_{i,j}^* = 0$

Now consider i, j such that $W_{i,j} < -m(\varepsilon' + 2\gamma) - \varepsilon''$. In this case $\hat{W}_{i,j} = -m(\varepsilon' + 2\gamma) - \varepsilon''$. By Lemma 10, if $\hat{q}_{i,j}^* > 0$, then $\hat{W}_{i,j} \geq \hat{W}_{i,i} - m(\varepsilon' + \sqrt{2}\gamma) \geq -m(\varepsilon' + \sqrt{2}\gamma) - \varepsilon''$. This contradicts with the fact that $W_{i,j} < -m(\varepsilon' + 2\gamma) - \varepsilon''$, so $\hat{q}_{i,j}^* = 0$. We have

1.

$$\begin{aligned} F_i \left(\hat{W}_{i,j} - \frac{\partial \phi(\hat{q}_{i,j}^*)}{\partial \hat{q}_{i,j}^*} \right) &= \hat{\lambda}_{i,j}^* + \hat{\mu}_i^* + F_i \hat{\pi}_j^* \\ \Leftrightarrow F_i \left(-m(\varepsilon' + 2\gamma) - \varepsilon'' - \frac{\partial \phi(\hat{q}_{i,j}^*)}{\partial \hat{q}_{i,j}^*} \right) &= \lambda_{i,j}^* - F_i(W_{i,j} + m(\varepsilon' + 2\gamma) + \varepsilon'') + \hat{\mu}_i^* + F_i \hat{\pi}_j^* \\ \Leftrightarrow F_i \left(W_{i,j} - \frac{\partial \phi(\hat{q}_{i,j}^*)}{\partial \hat{q}_{i,j}^*} \right) &= \lambda_{i,j}^* + \hat{\mu}_i^* + F_i \hat{\pi}_j^* \end{aligned}$$

2. $\hat{\lambda}_{i,j}^* \leq 0$

3. $\hat{\lambda}_{i,j}^* \hat{q}_{i,j}^* = 0$

Thus $(\lambda^*, \mu^* = \hat{\mu}^*, \pi^* = \hat{\pi}^*)$ satisfies the KKT conditions w.r.t. the convex program (P^3) . \square

As illustrated in the following example an arbitrary set of optimal dual variables that satisfy the KKT conditions can cause a big revenue loss.

Example 4. We consider the following instance. There is one agent with two possible types t_L and t_H such that the agent has type t_H with probability $F_{t_H} = 1 - p$ and t_L with probability $F_{t_L} = p$ for some sufficiently small $p > 0$. The outcome space $O = \{o_L, o_H\}$. $v(t_L, o_L) = v(t_H, o_H) = 1$ and $v(t_L, o_H) = v(t_H, o_L) = 0$. The mechanism \mathcal{M} is defined as follows: it returns outcome o_H with input t_H and returns outcome o_L with input t_L . The payment of the mechanism is always $\frac{1}{2}$. It's a BIC mechanism and it holds that $W_{t_L, t_L} = W_{t_H, t_H} = 1/2$ and $W_{t_L, t_H} = W_{t_H, t_L} = -1/2$.

We are going to prove through KKT conditions that for this instance, the following q^* is the optimal solution for Convex Program (P^3) : $q_{t_L, t_L}^* = q_{t_H, t_H}^* = 1$ and $q_{t_H, t_L}^* = q_{t_L, t_H}^* = 0$. Consider the following set of dual variables: $\mu_{t_L}^* = p(-1 + \gamma)$, $\mu_{t_H}^* = 0$, $\pi_{t_L}^* = 3/2 - 2\gamma$, $\pi_{t_H}^* = 1/2 - \gamma$ and $\lambda^*(t_H, t_L) = (1 - p)(-2 + 2\gamma)$, $\lambda^*(t_H, t_H) = \lambda^*(t_L, t_L) = \lambda^*(t_L, t_H) = 0$.

Note that $\gamma < 1$, and $\frac{\partial \phi(\hat{\mathbf{q}}_{t_L}^*)}{\partial q_{t_L, t_L}^*} = \frac{\partial \phi(\hat{\mathbf{q}}_{t_H}^*)}{\partial q_{t_H, t_H}^*} = \gamma$, $\frac{\partial \phi(\hat{\mathbf{q}}_{t_L}^*)}{\partial q_{t_L, t_H}^*} = \frac{\partial \phi(\hat{\mathbf{q}}_{t_H}^*)}{\partial q_{t_H, t_L}^*} = 0$. One can verify that the dual variables $(\mu^*, \pi^*, \lambda^*)$ satisfy the KKT conditions with respect to \mathbf{q}^* , which implies that \mathbf{q}^* is an optimal solution for (P^3) .

Note that $\min\{\mu_{t_L}^*/F_{t_L}, \mu_{t_H}^*/F_{t_H}\} = -1 + \gamma$. According to the payment rule, the payment charged to the first agent if she has type t_L is $3/2 - 2\gamma + \gamma - 0 + (-1 + \gamma) = 1/2$ and the payment charged if she has type t_H is $1/2 - \gamma + \gamma - 0 + (-1 + \gamma) = -1/2 + \gamma$.

Therefore the expected revenue of the RRSF mechanism with parameter $(\mu^*, \pi^*, \lambda^*)$ is $p \cdot 1/2 + (1-p) \cdot (-1/2 + \gamma) = -1/2 + \gamma + p(1 - \gamma)$. When both p and γ go to 0, the expected revenue of the mechanism is close to $-1/2$, which is far from the revenue of \mathcal{M} .

Lemma 12. Given any mechanism \mathcal{M} . Suppose for every agent k and every $i, j \in [m_k]$, $W_{i,i}^k \geq \max\{W_{i,j}^k - \varepsilon', -\varepsilon''\}$ holds for some $\varepsilon' \geq \varepsilon'' \geq 0$. Let $\hat{\mathbf{q}}^* = \{\hat{q}_{i,j}^*\}_{i,j \in [m_k]}$ be any optimal solution of the convex program (P^4) and $\hat{\lambda}^*, \hat{\mu}^*, \hat{\pi}^*$ be any Lagrange multipliers satisfying the KKT conditions w.r.t. the convex program (P^4) . Let $(\lambda^*, \mu^* = \hat{\mu}^*, \pi^* = \hat{\pi}^*)$ be the Lagrange multipliers stated in Lemma 11, which satisfy the KKT conditions w.r.t. the convex program (P^3) . Consider the RRSF mechanism \mathcal{M}' in Definition 8 with optimal solution $\mathbf{q}^* = \hat{\mathbf{q}}^*$ and Lagrange multipliers $(\lambda^*, \mu^* = \hat{\mu}^*, \pi^* = \hat{\pi}^*)$. Then for each agent k and $i \in [m_k]$, $\hat{p}_k(t_k^{(i)}) = \sum_j \pi_j^* q_{i,j}^* + \phi(\mathbf{q}_i^*) - \phi(\mathbf{0}) + \min_\ell \frac{\mu_\ell^*}{F_\ell} \geq -m(\varepsilon' + 2\gamma) - \varepsilon'' - \gamma$. This implies that

$$\text{REV}(\mathcal{M}') \geq \text{REV}(\mathcal{M}) - n(m(\varepsilon' + 2\gamma) + \varepsilon'' + \gamma)$$

Proof. Assume $\ell^* = \operatorname{argmin}_\ell \frac{\hat{\mu}_\ell^*}{F_\ell}$. Note that $\sum_j \hat{q}_{i,j}^* = 1$. If agent k reports type $t_k^{(i)}$ then :

$$\begin{aligned} \hat{p}_k(t_k^{(i)}) &= \sum_j \pi_j^* q_{i,j}^* + \phi(\mathbf{q}_i^*) - \phi(\mathbf{0}) + \min_\ell \frac{\mu_\ell^*}{F_\ell} \\ &= \sum_j \hat{\pi}_j^* q_{i,j}^* + \phi(\hat{\mathbf{q}}_i^*) - \phi(\mathbf{0}) + \sum_j \frac{\hat{\mu}_{\ell^*}^*}{F_{\ell^*}} q_{i,j}^* \\ &= \sum_j q_{i,j}^* \left(\frac{\hat{\mu}_{\ell^*}^*}{F_{\ell^*}} + \hat{\pi}_j^* \right) + \phi(\hat{\mathbf{q}}_i^*) - \phi(\mathbf{0}) \\ &\geq \sum_j q_{i,j}^* \left(\frac{\hat{\mu}_{\ell^*}^*}{F_{\ell^*}} + \hat{\pi}_j^* \right) \end{aligned}$$

By the KKT condition w.r.t to the convex program (P^4) we have that:

$$\begin{aligned} \sum_j \hat{q}_{i,j}^* \left(\frac{\hat{\mu}_{\ell^*}^*}{F_{\ell^*}} + \hat{\pi}_j^* \right) &= \sum_j \hat{q}_{i,j}^* \left(-\frac{\hat{\lambda}_{\ell^*,j}^*}{F_{\ell^*}} + \hat{W}_{\ell^*,j} - \frac{\partial \phi(\hat{\mathbf{q}}_{\ell^*}^*)}{\partial \hat{q}_{\ell^*,j}^*} \right) \\ &\geq \sum_j \hat{q}_{i,j}^* \left(\hat{W}_{\ell^*,j} - \frac{\partial \phi(\hat{\mathbf{q}}_{\ell^*}^*)}{\partial \hat{q}_{\ell^*,j}^*} \right) \quad (\hat{\lambda}_{\ell^*,j}^* \leq 0) \\ &\geq \sum_j \hat{q}_{i,j}^* \left(-m(\varepsilon' + 2\gamma) - \varepsilon'' - \frac{\partial \phi(\hat{\mathbf{q}}_{\ell^*}^*)}{\partial \hat{q}_{\ell^*,j}^*} \right) \quad (\text{Definition 9}) \\ &\geq -m(\varepsilon' + 2\gamma) - \varepsilon'' - \gamma \quad (\text{the definition of } \phi(\cdot) \text{ and } \hat{q}_{i,j}^* \in [0, 1]) \end{aligned}$$

Finally, note that in \mathcal{M}' , the expected revenue from payments $\{p_k(\cdot)\}_{k \in [n]}$ is exactly the same as $\text{REV}(\mathcal{M})$. Thus we have

$$\text{REV}(\mathcal{M}') \geq \text{REV}(\mathcal{M}) - n(m(\varepsilon' + 2\gamma) + \varepsilon'' + \gamma).$$

□

We summarize the result for the ideal model in the following theorem, by choosing \mathbf{W} as in Theorem 8.

Theorem 9. *Let \mathcal{M} be an ε -BIC and IR mechanism. Fix any $\gamma > 0$. Fix any agent k . For every $i, j \in [m_k]$, define $W_{i,j}^k$ as in Theorem 8. Let $\mathbf{q}_k^* = \{q_{i,j}^*\}_{i,j \in [m_k]}$ be the optimal solution of the convex program (P^4) (described in Definition 9) and $\boldsymbol{\mu}_k^* = \{\hat{\mu}_i^*\}_{i \in [m_k]}$, $\boldsymbol{\pi}_k^* = \{\hat{\pi}_j^*\}_{j \in [m_k]}$ be the corresponding optimal Lagrange multipliers. Let mechanism \mathcal{M}' be the RRSF mechanism with parameters $\mathbf{q}^* = \{\mathbf{q}_k^*\}_{k \in [n]}$, $\boldsymbol{\mu}^* = \{\boldsymbol{\mu}_k^*\}_{k \in [n]}$, $\boldsymbol{\pi}^* = \{\boldsymbol{\pi}_k^*\}_{k \in [n]}$ as described in Definition 8. Then \mathcal{M}' is BIC, IR, and*

$$\text{REV}(\mathcal{M}') \geq \text{REV}(\mathcal{M}) - n(m(\varepsilon + 2\gamma) + \gamma).$$

Proof. According to Lemma 11, \mathbf{q}_k^* is also an optimal solution for the convex program (P^3) and $\boldsymbol{\mu}_k^*$ and $\boldsymbol{\pi}_k^*$ are optimal Lagrange multipliers for (P^3) as well. Hence, \mathcal{M}' is BIC and IR by Theorem 8.

Since \mathcal{M} is ε -BIC and IR, we have $W_{i,i}^k \geq \max\{W_{i,j}^k - \varepsilon, 0\}$, for every $k \in [n]$ and $i, j \in [m_k]$. By applying Lemma 12 with $\varepsilon' = \varepsilon$ and $\varepsilon'' = 0$, we have that the revenue of \mathcal{M}' can be at most $n(m(\varepsilon + 2\gamma) + \gamma)$ smaller than the revenue of \mathcal{M} .

□

At the end of Section 7.2, we prove a lemma about RRSF mechanism using the KKT condition of the convex program (P^3). It's useful in the proof of Lemma 14.

Lemma 13. *Let \mathcal{M} be any RRSF mechanism with respect to $\mathbf{W} = \{W_{i,j}^k\}_{k \in [n], i, j \in [m_k]}$. It has parameter $(\mathbf{q}^*, \boldsymbol{\mu}^*, \boldsymbol{\pi}^*)$. Then for each agent k and $i \in [m_k]$, $\sum_{j \in [m_k]} q_{i,j}^* W_{i,j}^k - \hat{p}_k(t_k^{(i)}) \geq 0$.*

Proof. Fix any agent k . Note that $(\mathbf{q}_k^*, \boldsymbol{\mu}_k^*, \boldsymbol{\pi}_k^*)$ satisfies the KKT conditions of (P^3). By the definition of $\hat{p}_k(t_k^{(i)})$, for every $i \in [m_k]$, we have

$$\begin{aligned} \sum_{j \in [m_k]} q_{i,j}^* W_{i,j}^k - \hat{p}_k(t_k^{(i)}) &= \sum_j q_{i,j}^* (W_{i,j}^k - \pi_j^*) + \phi(\mathbf{0}) - \phi(\mathbf{q}_i^*) - \min_{\ell} \frac{\mu_{\ell}^*}{F_{\ell}} \\ &\geq \sum_j q_{i,j}^* (W_{i,j}^k - \pi_j^* - \frac{\mu_i^*}{F_i}) + \phi(\mathbf{0}) - \phi(\mathbf{q}_i^*) \quad (\sum_j q_{i,j}^* = 1) \\ &= \sum_j q_{i,j}^* \left(\frac{\lambda_{i,j}^*}{F_i} + \frac{\partial \phi(\mathbf{q}_i^*)}{\partial q_{i,j}^*} \right) + \phi(\mathbf{0}) - \phi(\mathbf{q}_i^*) \quad (\text{KKT condition 1}) \\ &= \sum_j q_{i,j}^* \frac{\partial \phi(\mathbf{q}_i^*)}{\partial q_{i,j}^*} + \phi(\mathbf{0}) - \phi(\mathbf{q}_i^*) \quad (\text{KKT condition 3}) \\ &= \nabla \phi(\mathbf{q}_i^*)^T \cdot \mathbf{q}_i^* + \phi(\mathbf{0}) - \phi(\mathbf{q}_i^*) \geq 0 \quad (\phi(\cdot) \text{ is convex}) \end{aligned}$$

□

7.3 Non-Ideal Model

In the previous section we showed that, in the ideal model, the RRSF mechanism \mathcal{M}' has only a small revenue loss, if the original mechanism \mathcal{M} is ε -BIC and IR. However, with only sample access to the distributions of the edge weights, these weights can not be computed exactly. In this section, we show that a BIC and IR mechanism with small revenue loss can be constructed, using estimates for the edge weights. The approach is similar to [28]. For any fixed agent k , let $\mathcal{W}_{i,j}^k$ be the utility of agent k on an execution of the mechanism $\mathcal{M} = (x, p)$ when she reports $t_k^{(j)}$ and her true type is $t_k^{(i)}$. More formally,

$$\mathcal{W}_{i,j} = v_k(t_k^{(i)}, x(t_k^{(j)}, t_{-k})) - p_k(t_k^{(j)}, t_{-k}) \quad (11)$$

Note that $\mathcal{W}_{i,j}$ is a random variable over $t_{-k} \sim \mathcal{D}_{-k}$ and the randomness of \mathcal{M} , whose expectation is $W_{i,j}$ as defined in Definition 8.

Definition 10 (Definition 4.9 from [28]). *The Regularized Estimated Replica–Surrogate Fractional Assignment (RERSF) with parameter L , is defined as follows:*

1. Fix any agent k . For every pair of types $t_k^{(i)}, t_k^{(j)} \in \text{supp}(\mathcal{D}_k)$, we define $\widetilde{W}_{i,j}^k$ as the empirical mean with L samples of $\mathcal{W}_{i,j}$. A sample of $\mathcal{W}_{i,j}$ is obtained by drawing a sample $t_{-k} \sim \mathcal{D}_{-k}$, running mechanism \mathcal{M} with input $(t_k^{(j)}, t_{-k})$, and computing the utility in Equation (11) based on the output outcome and payment. Let \mathcal{O}_k be the set of output outcomes from \mathcal{M} among all samples and pairs (i, j) . In total there are $m_k \cdot L$ number of outcomes.
2. Run the RRSF mechanism \mathcal{M}' in Definition 8 with respect to $\widetilde{W} = \{\widetilde{W}_{i,j}^k\}_{k \in [n], i, j \in [m_k]}$.

The following lemma shows that if all agents report truthfully, the RERSF mechanism has a small revenue loss compared to \mathcal{M} .

Lemma 14. *For any $\eta \in (0, 1)$, suppose $L \geq \frac{2}{\eta^2} \ln \left(\frac{2nm^2}{\eta} \right)$. Then the revenue of RERSF, when all agents report truthfully, is at least $\text{REV}(\mathcal{M}) - nm(\varepsilon + 6\gamma + 5\eta)$. Moreover, the expected utility for each agent when she reports truthfully is at least -3η .*

Proof. Note that $\mathcal{W}_{i,j}^k \in [-1, 1]$. By the Hoeffding bound, when $L \geq \frac{2}{\eta^2} \ln \left(\frac{2nm^2}{\eta} \right)$, $\Pr \left[|\widetilde{W}_{i,j}^k - W_{i,j}^k| > \eta \right] \leq \frac{\eta}{nm^2}$. By taking the union bound over all agents, replicas, and surrogates, we have that with probability at least $1 - \eta$, $|\widetilde{W}_{i,j}^k - W_{i,j}^k| \leq \eta$ holds for all agent k and types $t_k^{(i)}, t_k^{(j)} \in \text{supp}(\mathcal{D}_k)$. We refer to the event that the above inequalities hold as a “good” event.

For the first statement, consider the case when a “good” event happens. Since \mathcal{M} is ε -BIC and IR, we have $W_{i,i}^k \geq W_{i,j}^k - \varepsilon$ and $W_{i,i}^k \geq 0$, for every $k \in [n]$ and $i, j \in [m_k]$. Thus

$$\widetilde{W}_{i,i}^k \geq \widetilde{W}_{i,j}^k - \varepsilon - |\widetilde{W}_{i,i}^k - W_{i,i}^k| - |\widetilde{W}_{i,j}^k - W_{i,j}^k| \geq \widetilde{W}_{i,j}^k - \varepsilon - 2\eta$$

Moreover, $\widetilde{W}_{i,i}^k \geq W_{i,i}^k - |\widetilde{W}_{i,i}^k - W_{i,i}^k| \geq W_{i,i}^k - \eta$. By applying Lemma 12 with $\varepsilon' = \varepsilon + 2\eta$ and $\varepsilon'' = \eta$, we have that the expected revenue of RERSF is at most $n(m(\varepsilon + 2\gamma + 2\eta) + \gamma + \eta)$ smaller than the revenue of \mathcal{M} . When a “good” event does not happen, we can apply Lemma 12 with the trivial guarantee $\varepsilon' = \varepsilon'' = 1$. The revenue loss is at most $n(m(1 + 2\gamma) + 1 + \gamma) \leq nm(2 + 3\gamma)$. Thus the expected revenue loss is at most

$$n(m(\varepsilon + 2\gamma + 2\eta) + \gamma + \eta) + nm(2 + 3\gamma) \cdot \eta \leq nm(\varepsilon + 6\gamma + 5\eta)$$

We now prove the second statement. For every agent k , let $t_k^{(i)}$ be her true type. For every $j \in [m_k]$, let $q_{i,j}^*$ be the probability that she is represented by type $t_k^{(j)}$ in the RRSF mechanism (step 2 of the RERSF mechanism). Then her expected utility by reporting truthfully is $\sum_j W_{i,j}^k \cdot q_{i,j}^* - \hat{p}_k(t_k^{(i)})$, where \hat{p}_k is the extra payment in the RRSF mechanism with respect to \tilde{W} . By Lemma 13, we have $\sum_j \tilde{W}_{i,j}^k \cdot q_{i,j}^* - \hat{p}_k(t_k^{(i)}) \geq 0$. We have

$$\sum_j W_{i,j}^k \cdot q_{i,j}^* - \hat{p}_k(t_k^{(i)}) \geq \sum_j \tilde{W}_{i,j}^k \cdot q_{i,j}^* - \hat{p}_k(t_k^{(i)}) - \sum_j q_{i,j}^* |\tilde{W}_{i,j}^k - W_{i,j}^k| \geq - \sum_j q_{i,j}^* |\tilde{W}_{i,j}^k - W_{i,j}^k|$$

If a “good” event happens, $-\sum_j q_{i,j}^* |\tilde{W}_{i,j}^k - W_{i,j}^k| \geq -\eta$. If a “good” event does not happen, we can apply the trivial guarantee $|\tilde{W}_{i,j}^k - W_{i,j}^k| \leq 2$ (since $W_{i,j}^k \in [-1, 1]$) to get $-\sum_j q_{i,j}^* |\tilde{W}_{i,j}^k - W_{i,j}^k| \geq -2$. Hence the expected utility for agent k when she reports truthfully is at least $(-\eta) \cdot 1 + (-2) \cdot \eta = -3\eta$. \square

Note that for every agent k with type $t_k^{(i)}$, her expected utility when represented by a surrogate with type $t_k^{(i)}$, is $W_{i,j}^k$ instead of $\tilde{W}_{i,j}^k$. Thus the RERSF mechanism is not BIC. However, with the number of samples L sufficiently large, one can show that the mechanism is ϵ' -BIC for some small $\epsilon' > 0$. To formally prove this claim, we first introduce the *distinguishability* of two types.

Definition 11 (Definition 3.3 from [28]). *Let $v_k(\cdot, \cdot)$ be the valuation function of agent k . We define the swap-disutility for the types $t_k^{(i)}, t_k^{(j)} \in \text{supp}(\mathcal{D}_k)$ and outcomes $o, o' \in \mathcal{O}$ to be $d(t_k^{(i)}, t_k^{(j)}, o, o') = v(t_k^{(i)}, o) - v(t_k^{(i)}, o') + v(t_k^{(j)}, o') - v(t_k^{(j)}, o)$. The distinguishability of types $t_k^{(i)}, t_k^{(j)}$ is defined as*

$$d(t_k^{(i)}, t_k^{(j)}) = \max_{o, o' \in \mathcal{O}} d(t_k^{(i)}, t_k^{(j)}, o, o')$$

The pair $o, o' \in \mathcal{O}$ that achieves the maximum is called the distinguishing outcome pair for $t_k^{(i)}, t_k^{(j)}$. We call $t_k^{(i)}, t_k^{(j)}$ distinguishable w.r.t. an outcome set \mathcal{O} if they have non-zero distinguishability for some outcome in that set. For every pair of types $t_k^{(i)}, t_k^{(j)} \in \text{supp}(\mathcal{D}_k)$, define the sampled outcome distinguishability $\tilde{d}(t_k^{(i)}, t_k^{(j)})$ to be the distinguishability of types $t_k^{(i)}, t_k^{(j)}$ with respect to the output outcome set \mathcal{O}_k .

Lemma 15 (Lemma 4.7 from [28]). *For any $\gamma > 0$, $\zeta \in (0, 1/2)$, consider the RERSF mechanism with parameter $L > \frac{1}{2}\zeta^{-2} \ln(2m/\zeta)$. Then any agent k with type $t_k^{(i)} \in \text{supp}(\mathcal{D}_k)$ that participates in the mechanism can gain at most $\frac{2\zeta m}{\gamma} \mathbb{E}[\tilde{d}(t_k^{(i)}, t_k^{(j)})]$ utility by reporting type $t_k^{(j)} \in \text{supp}(\mathcal{D}_k)$, where $\tilde{d}(\cdot, \cdot)$ is the sampled outcome distinguishability of $t_k^{(i)}, t_k^{(j)}$. The expectation is taken over the randomness of \mathcal{O}_k .*

In order to make our mechanism BIC, we will combine the above RERSF mechanism with a strictly IC mechanism defined below. In a strictly IC mechanism, for every agent, the utility by reporting her true type is strictly larger the utility by reporting any other types, by some positive value.

Definition 12 ([28]). (Strict IC mechanism) *Fix any agent k . The strictly IC mechanism for agent k , denoted by SIC^k , consists of the following steps:*

1. *Chooses uniformly at random a pair of distinguishable types $t_k^{(i)}, t_k^{(j)} \in \text{supp}(\mathcal{D}_k)$. Let (o, o') be the distinguishing outcome pair for $(t_k^{(i)}, t_k^{(j)})$, where $o, o' \in \mathcal{O}_k$.*
2. *Let $v = v_k(t_k^{(i)}, o) - v_k(t_k^{(i)}, o')$ and $v' = v_k(t_k^{(j)}, o) - v_k(t_k^{(j)}, o')$ and define the price $p = \frac{v+v'}{2}$. Without loss of generality, assume $p \geq 0$, otherwise we can swap o and o' .*

3. The mechanism lets agent k select between the following two options: 1) The mechanism outputs an outcome o with payment p for agent k ; 2) The mechanism outputs an outcome o' with payment 0 for agent k .

Lemma 16 (Lemma 4.4 from [28]). *The SIC^k mechanism for agent k is strictly IC: The utility of agent k by reporting her true type $t_k^{(i)}$ is at least $\frac{\mathbb{E}[\tilde{d}(t_k^{(i)}, t_k^{(j)})]}{2M}$ larger than the utility by reporting another type $t_k^{(j)}$. Here M is the number of distinguishable pairs of types in $\text{supp}(\mathcal{D}_k)$, $\tilde{d}(\cdot, \cdot)$ is the sampled outcome distinguishability of $(t_k^{(i)}, t_k^{(j)})$ w.r.t. \mathcal{O}_k , and the expectation is taken over the randomness of \mathcal{O}_k .*

Now we are ready to construct our mechanism for the non-ideal model.

Definition 13. *For any $\delta > 0$, define mechanism \mathcal{M}' as follows. The mechanism has parameter γ, η, L (from the RERSF mechanism) and $\delta, C > 0$. With probability $1 - \delta$, it runs the RERSF mechanism (with parameter γ, η, L); With probability δ , it picks an agent k uniformly at random and runs SIC^k for agent k . The payment for agent k is defined in Definition 12. Other agents pays 0. At last, the mechanism subsidizes every agent C .*

We wrap everything up in Theorem 10. We prove that with a proper choice of each parameter, mechanism \mathcal{M}' in Definition 13 is BIC and IR. Also the revenue loss of \mathcal{M}' is small.

Theorem 10. *There exist choices for the parameters $\gamma = \varepsilon, \delta = \varepsilon, \eta = \varepsilon, C = 4\varepsilon$ and $L = \text{poly}(n, m, \frac{1}{\varepsilon})$ such that the mechanism \mathcal{M}' in Definition 13 is BIC, IR and $\text{REV}(\mathcal{M}') \geq \text{REV}(\mathcal{M}) - O(nm\varepsilon)$. Mechanism \mathcal{M}' has expected running time $\text{poly}(n, m, 1/\varepsilon, b)$ and makes in expectation at most $\text{poly}(n, m, 1/\varepsilon)$ queries to \mathcal{M} . Here b is an upper bound on the bit complexity of $v_k(t_k^{(i)}, o), F_i^k, o$ for every $k \in [n], i \in [m_k], o \in \mathcal{O}$.*

Proof. Let $\zeta = \frac{\varepsilon^2}{4nm^3}$ and $L = \left\lceil \max \left(\frac{2}{\varepsilon^2} \ln \left(\frac{2nm^2}{\varepsilon} \right), \frac{8m^6n^2}{\varepsilon^4} \ln \left(\frac{8m^4n}{\varepsilon^2} \right) \right) \right\rceil$. We first show that the mechanism is BIC. Fix any agent $k \in [n]$ and her type $t_k^{(i)} \in \text{supp}(\mathcal{D}_k)$. We are going to bound the difference between the utility of reporting truthfully and the utility of reporting type $t_k^{(j)} \in \text{supp}(\mathcal{D}_k)$, for any $j \neq i$. Note that $L > \frac{1}{2}\zeta^{-2} \ln(2m/\zeta)$. By Lemma 15, when \mathcal{M}' executes the RERSF mechanism (with probability $1 - \delta$), the difference is at least $-\frac{2\zeta m}{\gamma} \mathbb{E}[\tilde{d}(t_k^{(i)}, t_k^{(j)})]$. By Lemma 16, when \mathcal{M}' executes SIC^k (with probability δ/n), the difference is at least $\frac{\mathbb{E}[\tilde{d}(t_k^{(i)}, t_k^{(j)})]}{2m^2}$, since the number of distinguishable pairs of types w.r.t. \mathcal{O}_k is at most m^2 . When \mathcal{M}' executes SIC^ℓ for some $\ell \neq k$, the difference is 0 as agent k 's reported type won't affect the output outcome and her payment. Thus the difference between the expected utility when truthfully reporting and misreporting is at least

$$-\frac{2\zeta m}{\gamma} \mathbb{E}[\tilde{d}(t_k^{(i)}, t_k^{(j)})] \cdot (1 - \delta) + \frac{\mathbb{E}[\tilde{d}(t_k^{(i)}, t_k^{(j)})]}{2m^2} \cdot \frac{\delta}{n}$$

The above value is non-negative when $\delta = \gamma = \varepsilon$ and $\zeta = \frac{\varepsilon^2}{4nm^3}$. Thus the mechanism is BIC.

Furthermore, since $L \geq \frac{2}{\eta^2} \ln \left(\frac{2nm^2}{\eta} \right)$, by Lemma 14, the expected utility of agent k , given the fact that \mathcal{M}' executes the RERSF mechanism, is at least $C - 3\varepsilon$. When \mathcal{M}' executes SIC^k , the utility for agent k is at least $-1 + C$. This is because by Definition 12, the payment of the strict IC mechanism is at least -1 and the agent's value for the outcome is non-negative. When \mathcal{M}' executes SIC^ℓ for some $\ell \neq k$, agent k 's utility is at least C , since her value for the outcome is non-negative. Thus agent k 's overall expected utility by reporting her true type $t_k^{(i)}$ is at least

$$(C - 3\varepsilon) \cdot (1 - \delta) + (-1 + C) \cdot \frac{\delta}{n} + C \cdot \frac{\delta(n-1)}{n} \geq 0$$

The above inequality holds when $\delta = \varepsilon$ and $C = 4\varepsilon$. Thus \mathcal{M}' is IR. To bound the revenue loss, note that the payment in mechanism SIC^k is always non-negative. By Lemma 14, we have

$$\text{REV}(\mathcal{M}') \geq (1 - \delta) \cdot (\text{REV}(\mathcal{M}) - nm(\varepsilon + 6\gamma + 5\eta)) \geq \text{REV}(\mathcal{M}) - n(12m + 1)\varepsilon$$

The last inequality follows from the fact that $\text{REV}(\mathcal{M}) \leq n$.

Finally we discuss the running time of \mathcal{M}' and number of queries to \mathcal{M} . For a given input, in step 1 of the RERSF mechanism (see Definition 10), the mechanism makes L queries to \mathcal{M} for every $k \in [n]$ and $i, j \in [m_k]$. In step 2 of RERSF, it makes one query to obtain the output outcome for a given input. The strictly IC mechanism does not make queries to \mathcal{M} . Overall, \mathcal{M}' makes at most $O(mL) = \text{poly}(n, m, 1/\varepsilon)$ queries to \mathcal{M} .

For the running time of \mathcal{M}' , first consider the RERSF mechanism. By [32], the convex program (P^3) and its dual problem can be solved in $\text{poly}(n, m, L, b)$ time. Thus all the parameters in the RERSF is determined in $\text{poly}(n, m, L, b)$ time. One can easily verify that other procedures of RERSF runs in time $\text{poly}(n, m, L, b)$. Thus RERSF runs in $\text{poly}(n, m, 1/\varepsilon, b)$ time.

Now consider the running time of the strictly IC mechanism SIC^k . Note that finding all the distinguishable types with respect to \mathcal{O}_k takes time at most $O(m^4L^2)$, by going over all pairs of types $(t_k^{(i)}, t_k^{(j)})$ and pairs of outcomes (o, o') . Thus SIC^k runs in time $\text{poly}(n, m, 1/\varepsilon)$. Altogether, \mathcal{M}' runs in time $\text{poly}(n, m, 1/\varepsilon, b)$. \square

Similar to Section 6, we can apply Theorem 10 to the multi-dimensional revenue maximization problem and derive the following theorem. The proof of Theorem 11 is analogous to the proof of Theorem 6, following from Theorem 10 and Theorem 5.

Theorem 11. *Let \mathcal{O} be a general outcome space. Given **full access** to agents' type distributions $\mathcal{D} = \times_i \mathcal{D}_i$. Let b be an upper bound on the bit complexity of $v_i(t_i, o)$ and $\Pr(t_i)$ for any agent i , any type t_i , and any outcome o , and OPT be the optimal revenue achievable by any BIC and IR mechanism. We further assume that types are normalized, that is, for each agent i , type t_i and outcome o , $v_i(t_i, o) \in [0, 1]$.*

*Given oracle access to an α -approximation algorithm G for VWO with running time $rt_G(x)$, where x is the bit complexity of the input, there is an algorithm that terminates in $\text{poly}(n, m, \frac{1}{\varepsilon}, b, rt_G(\text{poly}(n, m, \frac{1}{\varepsilon}, b)))$ time, and outputs an **exactly BIC** and IR mechanism with expected revenue*

$$\text{REV}(\mathcal{M}, \mathcal{D}) \geq \alpha \cdot \text{OPT} - O(nm\varepsilon),$$

where $m = \max_{i \in [n]} |\text{supp}(\mathcal{D}_i)|$. On any input bid, \mathcal{M} computes the outcome and payments in expected running time $\text{poly}(n, m, \frac{1}{\varepsilon}, b, rt_G(\text{poly}(n, m, \frac{1}{\varepsilon}, b)))$.

References

- [1] Saeed Alaei. Bayesian Combinatorial Auctions: Expanding Single Buyer Mechanisms to Many Buyers. In *the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2011.
- [2] Saeed Alaei, Hu Fu, Nima Haghpanah, Jason Hartline, and Azarakhsh Malekian. Bayesian Optimal Auctions via Multi- to Single-agent Reduction. In *the 13th ACM Conference on Electronic Commerce (EC)*, 2012.

- [3] Saeed Alaei, Hu Fu, Nima Haghpanah, and Jason D. Hartline. The simple economics of approximately optimal auctions. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 628–637. IEEE Computer Society, 2013.
- [4] Aaron Archer, Christos Papadimitriou, Kunal Talwar, and Éva Tardos. An approximate truthful mechanism for combinatorial auctions with single parameter agents. *Internet Mathematics*, 1(2):129–150, 2004.
- [5] Moshe Babaioff, Nicole Immorlica, Brendan Lucier, and S. Matthew Weinberg. A simple and approximately optimal mechanism for an additive buyer. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 21–30, 2014.
- [6] Moshe Babaioff, Robert Kleinberg, and Aleksandrs Slivkins. Multi-parameter mechanisms with implicit payment computation. In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pages 35–52. ACM, 2013.
- [7] Moshe Babaioff, Robert D Kleinberg, and Aleksandrs Slivkins. Truthful mechanisms with implicit payment computation. *Journal of the ACM (JACM)*, 62(2):10, 2015.
- [8] Xiaohui Bei and Zhiyi Huang. Bayesian Incentive Compatibility via Fractional Assignments. In *the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011.
- [9] Anand Bhalgat, Sreenivas Gollapudi, and Kamesh Munagala. Optimal auctions via the multiplicative weight method. In Michael J. Kearns, R. Preston McAfee, and Éva Tardos, editors, *ACM Conference on Electronic Commerce, EC '13, Philadelphia, PA, USA, June 16-20, 2013*, pages 73–90. ACM, 2013.
- [10] Johanness Brustle, Yang Cai, and Constantinos Daskalakis. Multi-item mechanisms without item-independence: Learnability via robustness. *arXiv preprint arXiv:1911.02146*, 2019.
- [11] Yang Cai and Constantinos Daskalakis. Extreme-Value Theorems for Optimal Multidimensional Pricing. In *the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2011.
- [12] Yang Cai, Constantinos Daskalakis, and S. Matthew Weinberg. An Algorithmic Characterization of Multi-Dimensional Mechanisms. In *the 44th Annual ACM Symposium on Theory of Computing (STOC)*, 2012.
- [13] Yang Cai, Constantinos Daskalakis, and S. Matthew Weinberg. Optimal Multi-Dimensional Mechanism Design: Reducing Revenue to Welfare Maximization. In *the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2012.
- [14] Yang Cai, Constantinos Daskalakis, and S. Matthew Weinberg. Reducing Revenue to Welfare Maximization : Approximation Algorithms and other Generalizations. In *the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2013.
- [15] Yang Cai, Constantinos Daskalakis, and S. Matthew Weinberg. Understanding Incentives: Mechanism Design becomes Algorithm Design. In *the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2013.
- [16] Yang Cai, Nikhil R. Devanur, and S. Matthew Weinberg. A duality based unified approach to bayesian mechanism design. In *the 48th Annual ACM Symposium on Theory of Computing (STOC)*, 2016.

- [17] Yang Cai and Zhiyi Huang. Simple and Nearly Optimal Multi-Item Auctions. In *the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2013.
- [18] Yang Cai and Mingfei Zhao. Simple mechanisms for subadditive buyers via duality. In *the 49th Annual ACM Symposium on Theory of Computing (STOC)*, 2017.
- [19] Shuchi Chawla, Jason D. Hartline, and Robert D. Kleinberg. Algorithmic Pricing via Virtual Valuations. In *the 8th ACM Conference on Electronic Commerce (EC)*, 2007.
- [20] Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Multi-Parameter Mechanism Design and Sequential Posted Pricing. In *the 42nd ACM Symposium on Theory of Computing (STOC)*, 2010.
- [21] Shuchi Chawla and J. Benjamin Miller. Mechanism design for subadditive agents via an ex-ante relaxation. In Vincent Conitzer, Dirk Bergemann, and Yiling Chen, editors, *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, pages 579–596. ACM, 2016.
- [22] Constantinos Daskalakis, Nikhil R. Devanur, and S. Matthew Weinberg. Revenue maximization and ex-post budget constraints. In Roughgarden et al. [37], pages 433–447.
- [23] Constantinos Daskalakis and S. Matthew Weinberg. Symmetries and Optimal Multi-Dimensional Mechanism Design. In *the 13th ACM Conference on Electronic Commerce (EC)*, 2012.
- [24] Shaddin Dughmi, Jason D Hartline, Robert Kleinberg, and Rad Niazadeh. Bernoulli factories and black-box reductions in mechanism design. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 158–169. ACM, 2017.
- [25] Evangelia Gergatsouli, Brendan Lucier, and Christos Tzamos. The complexity of black-box mechanism design with priors. In Anna Karlin, Nicole Immorlica, and Ramesh Johari, editors, *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, pages 869–883. ACM, 2019.
- [26] Yannai A Gonczarowski and S Matthew Weinberg. The sample complexity of up-to- ϵ multi-dimensional revenue maximization. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 416–426. IEEE, 2018.
- [27] Jason Hartline, Robert Kleinberg, and Azarakhsh Malekian. Bayesian incentive compatibility via matchings. In *the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011.
- [28] Jason D Hartline, Robert Kleinberg, and Azarakhsh Malekian. Bayesian incentive compatibility via matchings. *Games and Economic Behavior*, 92:401–429, 2015.
- [29] Jason D. Hartline and Brendan Lucier. Bayesian Algorithmic Mechanism Design. In *the 42nd ACM Symposium on Theory of Computing (STOC)*, 2010.
- [30] MS Keane and George L O’Brien. A bernoulli factory. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 4(2):213–219, 1994.
- [31] Pravesh Kothari, Divyarthi Mohan, Ariel Schwartzman, Sahil Singla, and S Matthew Weinberg. Approximation schemes for a buyer with independent items via symmetries. *arXiv preprint arXiv:1905.05231*, 2019.

- [32] M.K. Kozlov, S.P. Tarasov, and L.G. Khachiyan. The polynomial solvability of convex quadratic programming. *USSR Computational Mathematics and Mathematical Physics*, 20(5):223 – 228, 1980.
- [33] Krzysztof Latuszynski. The bernoulli factory, its extensions and applications. *Proceedings of IWAP*, pages 1–5, 2010.
- [34] Roger B. Myerson. Optimal Auction Design. *Mathematics of Operations Research*, 6(1):58–73, 1981.
- [35] Şerban Nacu, Yuval Peres, et al. Fast simulation of new coins from old. *The Annals of Applied Probability*, 15(1A):93–115, 2005.
- [36] Jean-Charles Rochet. A necessary and sufficient condition for rationalizability in a quasi-linear context. *Journal of mathematical Economics*, 16(2):191–200, 1987.
- [37] Tim Roughgarden, Michal Feldman, and Michael Schwarz, editors. *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*. ACM, 2015.
- [38] Aviad Rubinstein and S. Matthew Weinberg. Simple mechanisms for a subadditive buyer and applications to revenue monotonicity. In Roughgarden et al. [37], pages 377–394.
- [39] Andrew Chi-Chih Yao. An n-to-1 bidder reduction for multi-item auctions and its applications. In *SODA*, 2015.

A Estimating γ : Approximating the Offline Optimum of the Regularized Matching

In this section, we show how to estimate the parameter γ so that it is a constant factor approximation to optimum of program (P'') (see Definition 6) on the replica-surrogate matching in Mechanism 3 with high probability. Importantly, the estimate is completely independent from the agent's reported type. Here is the basic idea. We sample the edge weights between \mathbf{r}' and \mathbf{s} , and use the empirical mean to compute the optimal solution of program (P''). We show that with polynomially many samples, the optimum of (P'') computed based on the empirical means is a constant approximation to the optimum of (P'') on the true edge weights with probability almost 1.

Lemma 17. *For any agent i , given parameters ℓ , δ , η' , and $d \geq 32 \frac{\log(8\eta'^{-1})}{\delta^2 \ell \log^2(\ell)}$, fix \mathbf{s} to be the ℓ surrogates, first draw $d\ell$ fresh samples from \mathcal{D}'_i , which we denote using \mathbf{r}' . We use $\text{OPT}(\omega)$ to denote the optimum of (P'') when the edge weight between the j -th replica/LHS-node and the k -th normal surrogate/normal RHS-node is ω_{jk} . There exists a randomized algorithm based only on \mathbf{r}' and \mathbf{s} that computes a γ that lies in $\left[\frac{2 \cdot \text{OPT}(\omega(\mathbf{r}'))}{d}, \frac{24 \cdot \text{OPT}(\omega(\mathbf{r}'))}{d} \right]$ with probability at least $1 - \eta'/2$, where $\omega_{jk}(\mathbf{r}') = W_i(r'^{(j)}, s^{(k)})$ as defined in Mechanism 3. Moreover, γ is at most $O(\max\{\ell, \delta \ell \log(\ell)\})$ and the algorithm has $\text{poly}(d, \ell, 1/\eta', 1/\delta)$ running time and makes $\text{poly}(d, \ell, 1/\eta', 1/\delta)$ queries to mechanism \mathcal{M} .*

Furthermore, if \mathbf{r} are $d\ell$ i.i.d. samples from \mathcal{D}'_i , then $\text{OPT}(\omega(\mathbf{r}'))$ lies in $\left[\frac{\text{OPT}(\omega(\mathbf{r}))}{2}, \frac{3 \cdot \text{OPT}(\omega(\mathbf{r}))}{2} \right]$ with probability at least $1 - \eta'/2$ over the randomness of \mathbf{r} and \mathbf{r}' , where $\omega_{jk}(\mathbf{r}) = W_i(r^{(j)}, s^{(k)})$. In this case, γ also lies in $\left[\frac{\text{OPT}(\omega(\mathbf{r}))}{d}, \frac{36 \cdot \text{OPT}(\omega(\mathbf{r}))}{d} \right]$ with probability at least $1 - \eta'$.

Proof of Lemma 17: We prove our statement in two steps. In the first step, we show that if we take polynomially many samples, we can obtain a sufficiently accurate estimate of $\omega(\mathbf{r}')_{jk}$ for each edge (j, k) . We prove that the optimum of (P'') on the estimated weights is close to $\text{OPT}(\omega(\mathbf{r}'))$. We use \mathcal{F}_{jk} to denote the distribution of the random variable $v_i(r'^{(j)}, x(s^{(k)}, t_{-i})) - (1 - \eta) \cdot p_i(s^{(k)}, t_{-i})$, where t_{-i} is distributed according to \mathcal{D}_{-i} .

Lemma 18 (adapted from [24]). *For each edge (j, k) between the j -th replica $r'^{(j)}$ and the k -th normal surrogate $s^{(k)}$, if we take $N \geq \frac{2 \log(4\ell^2 d\eta'^{-1})}{\delta^2 \log^2(\ell)}$ samples from \mathcal{F}_{jk} , and use $\hat{\omega}_{jk}(\mathbf{r}')$ to denote the empirical mean of these N samples, then with probability at least $1 - \frac{\eta'}{2}$,*

$$\frac{\text{OPT}(\omega(\mathbf{r}'))}{2} \leq \text{OPT}(\hat{\omega}(\mathbf{r}')) \leq 2\text{OPT}(\omega(\mathbf{r}')).$$

Proof. By the Chernoff bound, we know that $\Pr \left[|\omega_{jk}(\mathbf{r}) - \hat{\omega}_{jk}(\mathbf{r})| \geq \frac{\delta \log(\ell)}{2} \right] \leq \frac{\eta'}{2d\ell^2}$ for each edge (j, k) . Since there are $d\ell^2$ many edges, by the union bound, we have that with probability at least $1 - \frac{\eta'}{2}$ for each edge (j, k) :

$$|\omega_{jk}(\mathbf{r}) - \hat{\omega}_{jk}(\mathbf{r})| \leq \frac{\delta \log(\ell)}{2}$$

Let (x^*, y^*) be the optimal solution of the (P'') with edge weights $(\omega_{jk}(\mathbf{r}'))_{jk}$ and (x^{**}, y^{**}) be the optimal solution of the (P'') with edge weights $(\hat{\omega}_{jk}(\mathbf{r}'))_{jk}$. Then

$$\begin{aligned}
\text{OPT}(\omega(\mathbf{r}')) &= \sum_{j,k} (x_{jk}^* \omega_{jk}(\mathbf{r}') - \delta \cdot (x_{jk}^* \log(x_{jk}^*) + y_{jk}^* \log(y_{jk}^*))) \\
&\geq \sum_{j,k} (x_{jk}^{**} \omega_{jk}(\mathbf{r}') - \delta \cdot (x_{jk}^{**} \log(x_{jk}^{**}) + y_{jk}^{**} \log(y_{jk}^{**}))) \\
&\geq \text{OPT}(\hat{\omega}(\mathbf{r}')) - \frac{d\ell\delta \log(\ell)}{2} \\
&\geq \frac{\text{OPT}(\hat{\omega}(\mathbf{r}'))}{2}
\end{aligned}$$

The last inequality holds since a valid assignment is to set $y_{jk} = 1/\ell$ and $x_{jk} = 0$ for each j, k , which has objective value $\delta d\ell \log(\ell)$.

The other direction can be proved similarly. \square

Let A be the total weight of the maximum weight matching with edge weights $(\hat{\omega}_{jk}(\mathbf{r}'))_{j,k}$. It is clear that A lies in $[\text{OPT}(\hat{\omega}(\mathbf{r}')) - \delta d\ell \log(2\ell), \text{OPT}(\hat{\omega}(\mathbf{r}'))]$. Note that if we set $y_{jk} = 1/\ell$ and $x_{jk} = 0$ for each j, k , the objective of (P'') has value $\delta d\ell \log(\ell)$. Hence, $\max\{A, \delta d\ell \log(\ell)\}$ is guaranteed to lie in $\left[\frac{\text{OPT}(\hat{\omega}(\mathbf{r}'))}{3}, \text{OPT}(\hat{\omega}(\mathbf{r}'))\right]$. If we choose γ to be $\frac{12 \cdot \max\{A, \delta d\ell \log(\ell)\}}{d}$, γ is guaranteed to lie in $\left[\frac{4 \cdot \text{OPT}(\hat{\omega}(\mathbf{r}'))}{d}, \frac{12 \cdot \text{OPT}(\hat{\omega}(\mathbf{r}'))}{d}\right]$. Due to Lemma 18, γ lies in $\left[\frac{2 \cdot \text{OPT}(\omega(\mathbf{r}'))}{d}, \frac{24 \cdot \text{OPT}(\omega(\mathbf{r}'))}{d}\right]$ with probability at least $1 - \eta'/2$. As A can be computed in time $\text{poly}(d, \ell, 1/\eta', 1/\delta)$, γ can also be computed in time $\text{poly}(d, \ell, 1/\eta', 1/\delta)$.

In the second step of the proof, we show that $\text{OPT}(\omega(\mathbf{r}))$ and $\text{OPT}(\omega(\mathbf{r}'))$ are close with high probability. We first need the following Lemma to prove $\text{OPT}(\omega(\mathbf{r}))$ has bounded difference of 2.

Lemma 19. For any $j \in [d\ell]$, any type $r^{(j)}$ and replica profile \mathbf{r} ,

$$\left| \text{OPT}(\omega(\mathbf{r})) - \text{OPT}\left(\omega\left(r^{(j)}, \mathbf{r}^{(-j)}\right)\right) \right| \leq 2,$$

where $\omega_{jk}(r^{(j)}, \mathbf{r}^{(-j)}) = W_i(r^{(j)}, s^{(k)})$ and $\omega_{j'k}(r^{(j)}, \mathbf{r}^{(-j)}) = \omega_{j'k}(\mathbf{r})$ for any $j' \neq j$.

Proof. Let $(x^*, y^*), (x^{**}, y^{**})$ be the optimal solutions under replica profile \mathbf{r} and $(r^{(j)}, \mathbf{r}^{(-j)})$ for (P'') respectively. Then

$$\begin{aligned}
\text{OPT}(\omega(\mathbf{r})) &= \sum_{j' \in [d\ell], k \in [\ell]} (x_{j'k}^* \omega_{j'k}(\mathbf{r}) - \delta \cdot (x_{j'k}^* \log(x_{j'k}^*) + y_{j'k}^* \log(y_{j'k}^*))) \\
&\geq \sum_{j' \in [d\ell], k \in [\ell]} (x_{j'k}^{**} \omega_{j'k}(\mathbf{r}) - \delta \cdot (x_{j'k}^{**} \log(x_{j'k}^{**}) + y_{j'k}^{**} \log(y_{j'k}^{**}))) \\
&\geq \sum_{j' \neq j, k \in [\ell]} (x_{j'k}^{**} \omega_{j'k}(\mathbf{r}) - \delta \cdot (x_{j'k}^{**} \log(x_{j'k}^{**}) + y_{j'k}^{**} \log(y_{j'k}^{**}))) \\
&\quad + \sum_{k \in [\ell]} \left(x_{jk}^{**} \left(\omega_{jk}(r^{(j)}, r^{(-j)}) - 2 \right) - \delta \cdot (x_{jk}^{**} \log(x_{jk}^{**}) + y_{jk}^{**} \log(y_{jk}^{**})) \right) \\
&= \text{OPT}(\omega(r^{(j)}, \mathbf{r}^{(-j)})) - 2
\end{aligned}$$

The last inequality is because both $\omega_{jk}(\mathbf{r})$ and $\omega_{jk}(r^{(j)}, r^{(-j)})$ lie in $[-1, 1]$. The other direction follows similarly. \square

Next, we apply McDiarmid's inequality to the function $\text{OPT}(\omega(\mathbf{r}))$.

Lemma 20. *When $d \geq 32 \frac{\log(8\eta'^{-1})}{\delta^2 \ell \log^2(\ell)}$, if both \mathbf{r} and \mathbf{r}' are collections of $d\ell$ i.i.d. samples from \mathcal{D}'_i , then with probability at least $1 - \frac{\eta'}{2}$,*

$$\frac{1}{2} \text{OPT}(\omega(\mathbf{r})) \leq \text{OPT}(\omega(\mathbf{r}')) \leq \frac{3}{2} \text{OPT}(\omega(\mathbf{r})).$$

The probability is over the randomness of both \mathbf{r} and \mathbf{r}' .

Proof. Due to Lemma 20, we can apply McDiarmid's inequality on the function $\text{OPT}(\omega(\mathbf{r}))$, and we have

$$\Pr_{\mathbf{r}} \left[\left| \text{OPT}(\omega(\mathbf{r})) - \mathbb{E}_{\mathbf{r}}[\text{OPT}(\omega(\mathbf{r}))] \right| \geq \frac{\delta d \ell \log(\ell)}{4} \right] \leq \frac{\eta'}{4}$$

Similarly, we have

$$\Pr_{\mathbf{r}'} \left[\left| \text{OPT}(\omega(\mathbf{r}')) - \mathbb{E}_{\mathbf{r}'}[\text{OPT}(\omega(\mathbf{r}'))] \right| \geq \frac{\delta d \ell \log(\ell)}{4} \right] \leq \frac{\eta'}{4}$$

Hence, with probability at least $1 - \frac{\eta'}{2}$,

$$\left| \text{OPT}(\omega(\mathbf{r})) - \text{OPT}(\omega(\mathbf{r}')) \right| \leq \frac{\delta d \ell \log(\ell)}{2}$$

Since $\delta d \ell \log(\ell)$ is a lower bound on both $\text{OPT}(\omega(\mathbf{r}'))$ and $\text{OPT}(\omega(\mathbf{r}))$ (by setting $y_{jk} = 1/\ell$ and $x_{jk} = 0$ for each j, k) we have that with probability at least $1 - \frac{\eta'}{2}$,

$$\frac{1}{2} \text{OPT}(\omega(\mathbf{r})) \leq \text{OPT}(\omega(\mathbf{r}')) \leq \frac{3}{2} \text{OPT}(\omega(\mathbf{r})).$$

□

Lemma 17 follows from Lemma 18 and 20. □

B Missing Details from Section 5

Proof of Lemma 8:

\mathcal{M}' is BIC: We prove the Bayesian Incentive Compatibility in two parts. The first part is similar to the proof of Lemma 2. We argue that the distribution of the normal surrogate s_i that represents agent i in Phase 2, when the agent i reports truthfully, is \mathcal{D}_i . Note that for any matching Algorithm 2 produces, the k -th normal surrogate and the k -th 0-surrogate together are matched to exactly d replicas for every $k \in [\ell]$. As the $d\ell - 1$ replicas and the agent's type are all drawn from the same distribution \mathcal{D}'_i , we can simply treat all of them as replicas and uniformly choose one to be the agent reported type *after* Algorithm 2 terminates. Therefore, the surrogate s_i that represents the agent, will also be chosen uniformly at random between all the normal surrogates. Thus, the distribution of s_i is \mathcal{D}_i .

If all the other agents report truthfully, agent i 's value for winning a surrogate s is exactly $W_i(t_i, s)$ if her true type is t_i . In other words, under the assumption that all other agents report truthfully, Mechanism 3 for agent i is equivalent to a competition among replicas to win surrogates, and the edge weight between a

replica and a surrogate is exactly the replica's value for winning the surrogate. To show that Mechanism 3 is BIC, it suffices to prove that at any position π ,

$$u_i(t_i, \mathcal{A}_\pi(W(t'_i))) - \mathbb{E}_\lambda[q_i(t'_i, \lambda)]$$

is maximized when the reported type t'_i equals to the true type t_i . Here $W(t'_i)$ is simply the collection of the edge weights when $r^{(\pi)} = t'_i$, and the function $u_i(\cdot)$ is defined in Definition 7. A result by Rochet [36] implies that this is indeed the case. Interested readers can find a modern restatement of the result in Theorem 2.1 of [6]¹⁷.

\mathcal{M}' is IR: The expected utility for agent i with type t_i at position π is

$$u_i(t_i, \mathcal{A}_\pi(W)) - \mathbb{E}_\lambda[q_i(t_i, \lambda)] = \int_0^1 u_i(t_i, \mathcal{A}_\pi(\lambda W_\pi, W_{-\pi})) d\lambda + \sqrt{\delta}(\log 2\ell + 1), \quad (12)$$

where W is the collection of weights in G_i when agent i reports truthfully. We will first prove that for any $\lambda \in [0, 1]$, $u_i(t_i, \mathcal{A}_\pi(\lambda W_\pi, W_{-\pi}))$ is at least $-\frac{\delta \log(2\ell)}{\lambda}$. Denote $H(x, y) = -\sum_k (x_k \log(x_k) + y_k \log(y_k))$ as the entropy for distribution (x, y) .

Let (x''_π, y''_π) be $\mathcal{A}_\pi(\lambda W_\pi, W_{-\pi})$. By Observation 1,

$$(x''_\pi, y''_\pi) = \operatorname{argmax}_{(x_\pi, y_\pi)} \sum_k x_{\pi k} \cdot \lambda W_{\pi k} + \delta \cdot H(x_\pi, y_\pi) - \sum_k \gamma \alpha_k^{(\pi)} \cdot (x_{\pi k} + y_{\pi k})$$

By considering an alternative solution $(0, x''_\pi + y''_\pi)$, we have

$$\begin{aligned} \sum_k x''_{\pi k} \cdot \lambda W_{\pi k} + \delta \cdot H(x''_\pi, y''_\pi) - \sum_k \gamma \alpha_k^{(\pi)} \cdot (x''_{\pi k} + y''_{\pi k}) \\ \geq 0 - \delta \cdot \sum_k (x''_{\pi k} + y''_{\pi k}) \log(x''_{\pi k} + y''_{\pi k}) - \sum_k \gamma \alpha_k^{(\pi)} \cdot (x''_{\pi k} + y''_{\pi k}) \end{aligned}$$

Since $-\sum_k (x''_{\pi k} + y''_{\pi k}) \log(x''_{\pi k} + y''_{\pi k}) \geq 0$,

$$u_i(t_i, \mathcal{A}_\pi(\lambda W_\pi, W_{-\pi})) = \sum_k x''_{\pi k} \cdot W_{\pi k} \geq -\frac{\delta}{\lambda} H(x''_\pi, y''_\pi) \geq \frac{-\delta \log(2\ell)}{\lambda}.$$

Another lower bound for $u_i(t_i, \mathcal{A}_\pi(\lambda W_\pi, W_{-\pi}))$ is -1 , as the mechanism \mathcal{M} is IR, by definition of the edge weight $W_{\pi k} \geq -1$ for all $k \in [\ell]$. Now, we are ready to lower bound the utility

$$\begin{aligned} \text{RHS of Equation (12)} &\geq \int_{\sqrt{\delta}}^1 \frac{-\delta \log(2\ell)}{\lambda} d\lambda + \int_0^{\sqrt{\delta}} -1 d\lambda + \sqrt{\delta}(\log 2\ell + 1) \\ &= -\delta \log(1/\sqrt{\delta}) \log(2\ell) - \sqrt{\delta} + \sqrt{\delta}(\log(2\ell) + 1) \\ &\geq -\sqrt{\delta} \log(2\ell) + \sqrt{\delta} \log(2\ell) \\ &= 0 \end{aligned}$$

Next, we prove that \mathcal{M}' does not lose too much revenue by subsidizing the agents in Phase 1.

¹⁷To apply Theorem 2.1 of [6] to our setting, one should think of each surrogate as an outcome, and the corresponding edge weight as the value for the outcome. In other words, a replica's type is the weights on the incident edges. As the matching is computed by a maximal-in-range algorithm, we can allow the edge weights to be arbitrary numbers, and the induced allocation rule will still be implementable in an incentive compatible way. As a result, we can apply Theorem 2.1 of [6] to our setting. Note that the incentive compatible payment rule it gives is off by an absolute constant compared to our payment rule in Definition 7.

$\mathbb{E}_\lambda[q_i(t'_i, \lambda)]$ is at least $-\sqrt{\delta}(\log(2\ell) + 1)$: It suffices to show that

$$u_i(t_i, \mathcal{A}_\pi(W)) \geq u_i(t_i, \mathcal{A}_\pi(\lambda W_\pi, W_{-\pi}))$$

for any $\lambda \in [0, 1)$. We still use (x''_π, y''_π) to denote $\mathcal{A}_\pi(\lambda W_\pi, W_{-\pi})$ and $(\hat{x}_\pi, \hat{y}_\pi)$ to denote $\mathcal{A}_\pi(W)$.

By Observation 1, both allocations are maximal-in-range for the same dual variables $\alpha^{(\pi)}$. Hence, the following two inequalities are true.

$$\sum_k \hat{x}_{\pi k} W_{\pi k} + \delta \cdot H(\hat{x}_\pi, \hat{y}_\pi) - \sum_k \gamma \alpha_k^{(\pi)} \cdot (x'_{\pi k} + y'_{\pi k}) \geq \sum_k x''_{\pi k} W_{\pi k} + \delta \cdot H(x''_\pi, y''_\pi) - \sum_k \gamma \alpha_k^{(\pi)} \cdot (x''_{\pi k} + y''_{\pi k})$$

$$\sum_k x''_{\pi k} \lambda W_{\pi k} + \delta \cdot H(x''_\pi, y''_\pi) - \sum_k \gamma \alpha_k^{(\pi)} \cdot (x''_{\pi k} + y''_{\pi k}) \geq \sum_k \hat{x}_{\pi k} \lambda W_{\pi k} + \delta \cdot H(\hat{x}_\pi, \hat{y}_\pi) - \sum_k \gamma \alpha_k^{(\pi)} \cdot (\hat{x}_{\pi k} + \hat{y}_{\pi k})$$

Summing up the two inequalities together, we have

$$\sum_k (\hat{x}_{\pi k} - x''_{\pi k}) W_{\pi k} (1 - \lambda) \geq 0.$$

Since $\lambda \in [0, 1)$,

$$u_i(t_i, \mathcal{A}_\pi(W)) - u_i(t_i, \mathcal{A}_\pi(\lambda W_\pi, W_{-\pi})) = \sum_k (\hat{x}_{\pi k} - x''_{\pi k}) W_{\pi k} \geq 0.$$

Finally, we analyze the time and query complexity of the mechanism.

Time and Query Complexity: All steps except Step 3, 6, and 8 clearly has $\text{poly}(d, \ell)$ time and query complexity. According to Lemma 17, Step 3 has $\text{poly}(d, \ell, 1/\eta', 1/\delta)$ time and query complexity. Since γ is guaranteed to be at most $\max\{\ell, \delta \ell \log \ell\}$, Algorithm 2 in Step 6 has time and query complexity $\text{poly}(d, \ell, 1/\delta)$ according to Theorem 3. From Definition 7, it is clear that Step 8 also has time and query complexity at most $\text{poly}(d, \ell, 1/\delta)$. Hence, the mechanism \mathcal{M}' has time and query complexity $\text{poly}(d, \ell, 1/\eta', 1/\delta)$. \square

B.1 Bounding the Revenue Loss

Now we give the proof of Theorem 4. It suffices to lower bound the revenue of \mathcal{M}' from the second phase due to Lemma 8. To bound the revenue loss, we define some new notations.

New Notations: For every agent i , and the corresponding bipartite graph G_i , we define a new bipartite graph \widehat{G}_i whose left hand side nodes are the replicas/LHS-nodes of G_i . For each normal surrogate/RHS-node of G_i , we duplicate it d times to form the set of right hand side nodes of \widehat{G}_i . For the k -th surrogate in G_i , the $(a\ell + k)$ -th surrogate in \widehat{G}_i is one of its copies for all $0 \leq a \leq d - 1$. We do not copy the 0-surrogates to \widehat{G}_i . The edge weights in \widehat{G}_i are still defined using $W_i(\cdot, \cdot)$. Clearly, every d -to-1 matching in G_i corresponds to a 1-to-1 matching in \widehat{G}_i . If replica r is matched to a surrogate s in G_i , simply match r to the first available copy of s in \widehat{G}_i . We use ℓ' to denote $d\ell$, and \widehat{G}_i has $2\ell'$ nodes. When we say the matching in \widehat{G}_i produced by Algorithm 2, we mean the matching in \widehat{G}_i that corresponds to the matching produced by Algorithm 2 in G_i . We follow the convention to use $r^{(j)}$ to denote the type of the j -th replica

and $s^{(k)}$ to denote the type of the k -th surrogate in \widehat{G}_i . We further simplify the notation and use $p_i(t_i)$ to denote $\mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}}[p_i(t_i, t_{-i})]$ for any type $t_i \in \mathcal{T}_i$.

Given the replica profile \mathbf{r} and surrogate profile \mathbf{s} , for any matching $L(\mathbf{r}, \mathbf{s})$ in \widehat{G}_i , we slightly abuse notation to use $W_i(L(\mathbf{r}, \mathbf{s}))$ to denote $\sum_{(r,s) \in L(\mathbf{r}, \mathbf{s})} W_i(r, s)$. When the replica profile \mathbf{r} and surrogate profile \mathbf{s} are clear from context, we simply use $W_i(L)$ to denote the total weight of the matching L . Since the analysis mainly concerns the set of surrogates that are matched in a matching, we use $s \in L(\mathbf{r}, \mathbf{s})$ to denote that the surrogate s is matched in $L(\mathbf{r}, \mathbf{s})$. Let $O(\mathbf{r}, \mathbf{s})$ be the (randomized) matching obtained by Algorithm 2 on \widehat{G}_i , $V(\mathbf{r}, \mathbf{s})$ be the maximum weight matching in \widehat{G}_i .

We first provide a Lemma that relates the expected revenue of \mathcal{M}' to the size of the matchings.

Lemma 21. *Let $\text{REV-SECOND}_i(\mathcal{M}', \mathcal{D}')$ be the expected revenue of \mathcal{M}' from agent i in Phase 2, $\text{REV}_i(\mathcal{M}, \mathcal{D})$ be the expected revenue of \mathcal{M} from agent i ,*

$$\text{REV-SECOND}_i(\mathcal{M}', \mathcal{D}') \geq (1 - \eta) \cdot \mathbb{E}_{\mathbf{r}, \mathbf{s}} \left[\sum_{s^{(k)} \in O(\mathbf{r}, \mathbf{s})} p_i(s^{(k)}) / \ell' \right],$$

and

$$\text{REV}_i(\mathcal{M}, \mathcal{D}) = \mathbb{E}_{\mathbf{s}} \left[\sum_{k \in [\ell']} p_i(s^{(k)}) / \ell' \right]$$

Proof. For every agent i , only when the agent i is matched to a surrogate in $O(\mathbf{r}, \mathbf{s})$, she pays the surrogate price. We can again first sample \mathbf{r} and \mathbf{s} , and run Algorithm 2 on the corresponding graph \widehat{G}_i to find the matching $O(\mathbf{r}, \mathbf{s})$, then choose a replica uniformly at random to be agent i . Since each replica has exactly probability $1/\ell'$ to be agent i , each surrogate in $O(\mathbf{r}, \mathbf{s})$ is selected with probability $1/\ell'$, the expected revenue paid by agent i is exactly $(1 - \eta) \cdot \mathbb{E}_{\mathbf{r}, \mathbf{s}} \left[\sum_{s^{(k)} \in O(\mathbf{r}, \mathbf{s})} \frac{p_i(s^{(k)})}{\ell'} \right]$. The expected payment from agent i in \mathcal{M} is $\mathbb{E}_{t_i \sim \mathcal{D}_i} [p_i(t_i)]$. Since each $s^{(k)}$ is drawn from \mathcal{D}_i , this is exactly the same as $\mathbb{E}_{\mathbf{s}} \left[\sum_{k \in [\ell']} p_i(s^{(k)}) / \ell' \right]$. \square

In Lemma 22, we bound the gap between $\text{REV-SECOND}_i(\mathcal{M}', \mathcal{D}')$ and $\text{REV}_i(\mathcal{M}, \mathcal{D})$. Indeed, we prove a stronger result that holds for any matching $K(\mathbf{r}, \mathbf{s})$ that has close to maximum total weight.

Lemma 22. *Recall that $V(\mathbf{r}, \mathbf{s})$ is the maximum weight matching in \widehat{G}_i . Let*

$$\mathbb{E}_{\mathbf{r}, \mathbf{s}} [W_i(K(\mathbf{r}, \mathbf{s}))] = \mathbb{E}_{\mathbf{r}, \mathbf{s}} [W_i(V(\mathbf{r}, \mathbf{s}))] - \Delta.$$

We have

$$(1 - \eta) \mathbb{E}_{\mathbf{r}, \mathbf{s}} \left[\sum_{s^{(k)} \in K(\mathbf{r}, \mathbf{s})} p_i(s^{(k)}) / \ell' \right] \geq \mathbb{E}_{\mathbf{s}} \left[\sum_{k \in [\ell']} p_i(s^{(k)}) / \ell' \right] - n \left[\eta + \sqrt{\frac{|\mathcal{T}'_i|}{\ell}} + \frac{\varepsilon}{\eta} + \frac{\Delta}{d\ell\eta} \right] - \frac{2}{\eta} d_w(\mathcal{D}_i, \mathcal{D}'_i).$$

To prove Theorem 4, one only needs to choose $K(\mathbf{r}, \mathbf{s})$ to be the matching $O(\mathbf{r}, \mathbf{s})$ produced Algorithm 2, and combine the guarantees in Lemma 8 and 21.

Instead of proving Lemma 22, we prove the following strengthened version of the statement.

Lemma 23. *Let $\mathbb{E}_{\mathbf{r}, \mathbf{s}} [W_i(K(\mathbf{r}, \mathbf{s}))] = \mathbb{E}_{\mathbf{r}, \mathbf{s}} [W_i(V(\mathbf{r}, \mathbf{s}))] - \Delta$. We have*

$$(1 - \eta) \mathbb{E}_{\mathbf{r}, \mathbf{s}} \left[\sum_{s^{(k)} \in K(\mathbf{r}, \mathbf{s})} p_i(s^{(k)}) / \ell' \right] \geq \mathbb{E}_{\mathbf{s}} \left[\sum_{k \in [\ell']} p_i(s^{(k)}) / \ell' \right] - \left(\eta + \sqrt{\frac{|\mathcal{T}'_i|}{\ell}} + \frac{\varepsilon}{\eta} + \frac{\Delta}{d\ell\eta} \right) - \frac{2}{\eta} d_w(\mathcal{D}_i, \mathcal{D}'_i).$$

Moreover, for any coupling $c_i(\cdot)$ that v_i is non-increasing w.r.t. $c_i(\cdot)$, the last term can be improved to

$$-\frac{1}{\eta \ell'} \mathbb{E}_{\mathbf{r}, \mathbf{s}} \left[\sum_{(r^{(j)}, s^{(k)}) \in K(\mathbf{r}, \mathbf{s})} \mathbb{E}_{c_i(\mathbf{r}), t_{-i} \sim \mathcal{D}_{-i}} \left[v_i(r^{(j)}, x(s^{(k)}, t_{-i})) - v_i(c_i(r^{(j)}), x(s^{(k)}, t_{-i})) \right] \right].^{18}$$

Proof of Lemma 23: To prove the statement, we consider an arbitrary coupling $c_i(\cdot, \cdot)$ of distribution \mathcal{D}'_i and \mathcal{D}_i . For every replica $r \in \mathcal{T}'_i$, $c_i(r)$ is a random type from \mathcal{T}_i . For every realization of the types $c_i(\mathbf{r}) = (c_i(r^{(j)}))_{j \in [\ell']}$, we consider the maximal matching that matches a replica $r^{(j)}$ with a surrogate $s^{(k)}$ only if $c_i(r^{(j)}) = s^{(k)}$. We denote the matching as $L(c_i(\mathbf{r}), \mathbf{s})$ and refer to it as the *maximal coupled same-type matching*. In the next Lemma, we argue that in expectation of \mathbf{r}, \mathbf{s} and the realization of $c_i(\mathbf{r})$, the expected size of $L(c_i(\mathbf{r}), \mathbf{s})$ is close to ℓ' .

Lemma 24. *For any \mathbf{r}, \mathbf{s} , and realization of $c_i(\mathbf{r}) = (c_i(r^{(j)}))_{j \in [\ell']}$, let $L(c_i(\mathbf{r}), \mathbf{s})$ be a maximal coupled same-type matching, then*

$$\mathbb{E}_{\mathbf{r}, \mathbf{s}, c_i(\mathbf{r})} [|L(c_i(\mathbf{r}), \mathbf{s})|] \geq \ell' - \sqrt{d |\mathcal{T}'_i| \cdot \ell'}.$$

Proof. To prove the result, we first invoke the following Lemma.

Lemma 25 (Adapted from [27]). *Let \mathbf{r}' be N replicas drawn i.i.d. from distribution \mathcal{D}'_i , and \mathbf{s}' be N surrogates drawn i.i.d. from distribution \mathcal{D}_i . For any coupling $c_i(\cdot)$ between \mathcal{D}'_i and \mathcal{D}_i , the expected cardinality of a maximal matching that only matches a replica r and a surrogate s when $c_i(r) = s$ is at least $N - \sqrt{|\mathcal{T}'_i| \cdot N}$. The expectation is over the randomness of \mathbf{r}', \mathbf{s}' , and the coupling $c_i(\mathbf{r}')$.*

Although we have ℓ' replicas and ℓ' surrogates, we cannot directly apply Lemma 25, as the surrogates are not i.i.d. samples from \mathcal{D}_i . Instead, we partition \widehat{G}_i into d subgraphs. The α -th subgraph contains all replicas $r^{(j)}$ and surrogates $s^{(k)}$ with j and k lie in $[\alpha \ell + 1, (\alpha + 1)\ell]$. If we only consider the α -th subgraph, due to our construction of \widehat{G}_i , the replicas are all sampled i.i.d. from \mathcal{D}'_i and the surrogates are also sampled i.i.d. from \mathcal{D}_i . Therefore, Lemma 25 implies that a maximal coupled same-type matching in the α -th subgraph has expected size at least $\ell - \sqrt{|\mathcal{T}'_i| \ell}$. Since there are d subgraphs, so the expected size of a maximal coupled same-type matching is at least $\ell' - \sqrt{d |\mathcal{T}'_i| \cdot \ell'}$. \square

Now, it suffices to argue that the total payment from surrogates that are in $L(c_i(\mathbf{r}), \mathbf{s})$ but not in $K(\mathbf{r}, \mathbf{s})$ is small. Indeed, when $K(\mathbf{r}, \mathbf{s})$ is the maximum weight matching, one can directly prove the claim. However, $K(\mathbf{r}, \mathbf{s})$ only has approximately maximum weight, and it appears to be difficult to directly compare $K(\mathbf{r}, \mathbf{s})$ with $L(c_i(\mathbf{r}), \mathbf{s})$. Instead, we construct an auxiliary matching based on both $K(\mathbf{r}, \mathbf{s})$ and $L(c_i(\mathbf{r}), \mathbf{s})$. For any \mathbf{r}, \mathbf{s} and realization of types $(c_i(r^{(j)}))_{j \in [\ell']}$, we decompose the union of these two matchings into a set of disjoint alternating paths and cycles. Every surrogate that appears in $L(c_i(\mathbf{r}), \mathbf{s})$ but not in $K(\mathbf{r}, \mathbf{s})$ must be an endpoint of some alternating path. These alternating paths have the following two forms:

- (a). It starts with a surrogate in $L(c_i(\mathbf{r}), \mathbf{s}) \setminus K(\mathbf{r}, \mathbf{s})$ and ends with a surrogate in $K(\mathbf{r}, \mathbf{s}) \setminus L(c_i(\mathbf{r}), \mathbf{s})$ with the form $(s_{(1)}, r_{(1)}, s_{(2)}, r_{(2)}, \dots, r_{(a)}, s_{(a+1)})$.
- (b). It starts with a surrogate in $L(c_i(\mathbf{r}), \mathbf{s}) \setminus K(\mathbf{r}, \mathbf{s})$ and ends with a replica with the form $(s_{(1)}, r_{(1)}, s_{(2)}, r_{(2)}, \dots, s_{(a)}, r_{(a)})$.

¹⁸For the rest of the proof, when we use the notation $\mathbb{E}_{c_i(\mathbf{r})}[\cdot]$, we are taking the expectation over the randomness of the coupling. The $c_i(\mathbf{r}) = (c_i(r^{(j)}))_{j \in [\ell']}$ inside the expectation is the realized type after coupling.

We use \mathcal{P} to denote the set of all alternating paths of form (a) and (b). We construct a new matching $K'(c_i(\mathbf{r}), \mathbf{s})$ as follows: start with the matching $K(\mathbf{r}, \mathbf{s})$, for any alternating path P of form (a) and (b), swap the edges in $K(\mathbf{r}, \mathbf{s})$ with the ones in $L(c_i(\mathbf{r}), \mathbf{s})$, that is, replace all edges in $P \cap K(\mathbf{r}, \mathbf{s})$ with edges in $P \cap L(c_i(\mathbf{r}), \mathbf{s})$. Since all the alternating paths are disjoint, $K'(c_i(\mathbf{r}), \mathbf{s})$ is indeed a matching.

Corollary 3.

$$\mathbb{E}_{\mathbf{r}, \mathbf{s}, c_i(\mathbf{r})} \left[\sum_{s^{(k)} \in K'(c_i(\mathbf{r}), \mathbf{s})} p_i(s^{(k)}) / \ell' \right] \geq \mathbb{E}_{\mathbf{s}} \left[\sum_{k \in [\ell']} p_i(s^{(k)}) / \ell' \right] - \sqrt{\frac{|\mathcal{T}'_i|}{\ell}}.$$

Proof. Fix \mathbf{r}, \mathbf{s} and types $\{c_i(r^{(j)})\}_{j \in [\ell']}$. For any alternating path P of form either (a) or (b), $P \cap L(c_i(\mathbf{r}), \mathbf{s})$ is the same as $P \cap K'(c_i(\mathbf{r}), \mathbf{s})$. For other alternating paths, the matched surrogate in $P \cap L(c_i(\mathbf{r}), \mathbf{s})$ is a subset of $P \cap K(\mathbf{r}, \mathbf{s})$. Thus the number of the matched surrogates in $K'(c_i(\mathbf{r}), \mathbf{s})$ is at least $|L(c_i(\mathbf{r}), \mathbf{s})|$. By Lemma 24, $\mathbb{E}_{\mathbf{r}, \mathbf{s}, c_i(\mathbf{r})} [|\{k : s^{(k)} \notin K'(c_i(\mathbf{r}), \mathbf{s})\}|] \leq \sqrt{d|\mathcal{T}'_i| \cdot \ell'}$. As \mathcal{M} is IR, $p_i(s) \leq 1$ for any surrogate $s \in \mathcal{T}_i$. Therefore,

$$\mathbb{E}_{\mathbf{r}, \mathbf{s}, c_i(\mathbf{r})} \left[\sum_{s^{(k)} \in K'(c_i(\mathbf{r}), \mathbf{s})} p_i(s^{(k)}) / \ell' \right] \geq \mathbb{E}_{\mathbf{s}} \left[\sum_{k \in [\ell']} p_i(s^{(k)}) / \ell' \right] - \sqrt{\frac{|\mathcal{T}'_i|}{\ell}}.$$

□

Equipped with Corollary 3, we only need to compare $K(\mathbf{r}, \mathbf{s})$ with $K'(c_i(\mathbf{r}), \mathbf{s})$.

Lemma 26.

$$\mathbb{E}_{\mathbf{r}, \mathbf{s}} \left[\sum_{s^{(k)} \in K(\mathbf{r}, \mathbf{s})} p_i(s^{(k)}) / \ell' \right] \geq \mathbb{E}_{\mathbf{r}, \mathbf{s}, c_i(\mathbf{r})} \left[\sum_{s^{(k)} \in K'(c_i(\mathbf{r}), \mathbf{s})} p_i(s^{(k)}) / \ell' \right] - \frac{1}{\eta} \left(\varepsilon + \frac{\Delta}{\ell'} \right) - \frac{2}{\eta} d_w(\mathcal{D}_i, \mathcal{D}'_i)$$

Moreover, for any coupling c_i such that v_i is non-increasing w.r.t. c_i , the last term can be improved to $-\frac{1}{\eta} \mathbb{E}_{t \sim \mathcal{D}'} [v_i(t_i, x'(t)) - v_i(c_i(t_i), x'(t))]$.

Proof. Fix any \mathbf{r}, \mathbf{s} and realization of $c_i(\mathbf{r})$. Observe that if we decompose the union of $K(\mathbf{r}, \mathbf{s})$ and $K'(c_i(\mathbf{r}), \mathbf{s})$ into alternating path and cycles, we will end up with many length 2 cycles and all the alternating paths in \mathcal{P} . Hence, we only need to consider the paths in \mathcal{P} .

Consider any $k \in [a]$ if the path has form (a) (or $k \in [a-1]$ if the path has form (b)), note that $c_i(r_{(k)}) = s_{(k)}$, as this is also an edge in the matching $L(c_i(\mathbf{r}), \mathbf{s})$. Since \mathcal{M} is ε -BIC, we have

$$\mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [v_i(c_i(r_{(k)}), x(s_{(k)}, t_{-i}))] - p_i(s_{(k)}) \geq \mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [v_i(c_i(r_{(k)}), x(s_{(k+1)}, t_{-i}))] - p_i(s_{(k+1)}) - \varepsilon,$$

which is equivalent to

$$\begin{aligned} & W_i(r_{(k)}, s_{(k)}) - W_i(r_{(k)}, s_{(k+1)}) \\ & \geq -\varepsilon - \eta \cdot (p_i(s_{(k+1)}) - p_i(s_{(k)})) + \Delta_{i, c_i}(r_{(k)}, s_{(k)}) - \Delta_{i, c_i}(r_{(k)}, s_{(k+1)}) \end{aligned} \quad (13)$$

where $\Delta_{i, c_i}(r, s) = \mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [v_i(r, x(s, t_{-i})) - v_i(c_i(r), x(s, t_{-i}))]$.

By summing up Inequality (13) for each k , we are able to relate the difference of the total weight between $K(\mathbf{r}, \mathbf{s})$ and $K'(c_i(\mathbf{r}), \mathbf{s})$ with the total payment from surrogates in $K(\mathbf{r}, \mathbf{s})$ and $K'(c_i(\mathbf{r}), \mathbf{s})$.

- For any form (a) path,

$$\begin{aligned} & \sum_{k=1}^a (W_i(r(k), s_{(k+1)}) - W_i(r(k), s_{(k)})) \\ & \leq a \cdot \varepsilon + \eta \cdot (p_i(s_{(a+1)}) - p_i(s_{(1)})) - \sum_{k=1}^a (\Delta_{i,c_i}(r(k), s_{(k)}) - \Delta_{i,c_i}(r(k), s_{(k+1)})) \end{aligned}$$

- For any form (b) path

$$\begin{aligned} & \sum_{k=1}^{a-1} (W_i(r(k), s_{(k+1)}) - W_i(r(k), s_{(k)})) - W_i(r(a), s_{(a)}) \\ & \leq (a-1) \cdot \varepsilon + \eta \cdot \sum_{k=1}^{a-1} (p_i(s_{(k+1)}) - p_i(s_{(k)})) - W_i(r(a), s_{(a)}) - \sum_{k=1}^{a-1} (\Delta_{i,c_i}(r(k), s_{(k)}) - \Delta_{i,c_i}(r(k), s_{(k+1)})) \\ & \leq (a-1) \cdot \varepsilon - \eta \cdot p_i(s_{(1)}) - \sum_{k=1}^a \Delta_{i,c_i}(r(k), s_{(k)}) + \sum_{k=1}^{a-1} \Delta_{i,c_i}(r(k), s_{(k+1)}) \end{aligned}$$

The last inequality is because $\eta \cdot p_i(s_{(a)}) - W_i(r(a), s_{(a)}) \leq -\Delta_{i,c_i}(r(a), s_{(a)})$, which is implied by the fact that \mathcal{M} is IR.

To sum up, for any alternating path $P \in \mathcal{P}$,

$$\begin{aligned} & \sum_{(r^{(j)}, s^{(k)}) \in P \cap K(\mathbf{r}, \mathbf{s})} W_i(r^{(j)}, s^{(k)}) - \sum_{(r^{(j)}, s^{(k)}) \in P \cap K'(c_i(\mathbf{r}), \mathbf{s})} W_i(r^{(j)}, s^{(k)}) \\ & \leq |P \cap K(\mathbf{r}, \mathbf{s})| \cdot \varepsilon + \eta \cdot \left[\sum_{s^{(k)} \in P \cap K(\mathbf{r}, \mathbf{s})} p_i(s^{(k)}) - \sum_{s^{(k)} \in P \cap K'(c_i(\mathbf{r}), \mathbf{s})} p_i(s^{(k)}) \right] + \text{DIFF}(P), \end{aligned} \quad (14)$$

where $\text{DIFF}(P) = \sum_{(r^{(j)}, s^{(k)}) \in P \cap K(\mathbf{r}, \mathbf{s})} \Delta_{i,c_i}(r^{(j)}, s^{(k)}) - \sum_{(r^{(j)}, s^{(k)}) \in P \cap K'(c_i(\mathbf{r}), \mathbf{s})} \Delta_{i,c_i}(r^{(j)}, s^{(k)})$. Since $V(\mathbf{r}, \mathbf{s})$ is the maximum weight matching, we have

$$\begin{aligned} & \sum_{P \in \mathcal{P}} \left[\sum_{(r^{(j)}, s^{(k)}) \in P \cap K(\mathbf{r}, \mathbf{s})} W_i(r^{(j)}, s^{(k)}) - \sum_{(r^{(j)}, s^{(k)}) \in P \cap K'(c_i(\mathbf{r}), \mathbf{s})} W_i(r^{(j)}, s^{(k)}) \right] \\ & = W_i(K(\mathbf{r}, \mathbf{s})) - W_i(K'(c_i(\mathbf{r}), \mathbf{s})) \\ & \geq W_i(K(\mathbf{r}, \mathbf{s})) - W_i(V(\mathbf{r}, \mathbf{s})) \end{aligned} \quad (15)$$

Note that if we are using the matching $L(c_i(\mathbf{r}), \mathbf{s})$ instead of $K'(c_i(\mathbf{r}), \mathbf{s})$, we can no longer prove Inequality (15). The reason is quite subtle. It is possible that $L(c_i(\mathbf{r}), \mathbf{s})$ has much higher weight than $K(\mathbf{r}, \mathbf{s})$ on paths in \mathcal{P} , but much smaller weight on the rest alternating path and cycles. In that case, the first equal sign will be replaced by a less equal sign, which makes the inequality meaningless. By comparing to $K'(c_i(\mathbf{r}), \mathbf{s})$, we can avoid this issue.

Combining Inequality (14) and (15), we have

$$\begin{aligned}
& W_i(K(\mathbf{r}, \mathbf{s})) - W_i(V(\mathbf{r}, \mathbf{s})) \\
& \leq \sum_{P \in \mathcal{P}} \left[\sum_{(r^{(j)}, s^{(k)}) \in P \cap K(\mathbf{r}, \mathbf{s})} W_i(r^{(j)}, s^{(k)}) - \sum_{(r^{(j)}, s^{(k)}) \in P \cap K'(c_i(\mathbf{r}), \mathbf{s})} W_i(r^{(j)}, s^{(k)}) \right] \\
& \leq \sum_{P \in \mathcal{P}} \left[|P \cap K(\mathbf{r}, \mathbf{s})| \cdot \varepsilon + \eta \cdot \left[\sum_{s^{(k)} \in P \cap K(\mathbf{r}, \mathbf{s})} p_i(s^{(k)}) - \sum_{s^{(k)} \in P \cap K'(c_i(\mathbf{r}), \mathbf{s})} p_i(s^{(k)}) \right] + \text{DIFF}(P) \right] \\
& \leq \ell' \cdot \varepsilon + \eta \cdot \left[\sum_{s^{(k)} \in K(\mathbf{r}, \mathbf{s})} p_i(s^{(k)}) - \sum_{s^{(k)} \in K'(c_i(\mathbf{r}), \mathbf{s})} p_i(s^{(k)}) \right] + \sum_{P \in \mathcal{P}} \text{DIFF}(P)
\end{aligned}$$

Finally, we take expectation over \mathbf{r} , \mathbf{s} , and $c_i(\mathbf{r})$.

$$\begin{aligned}
& \mathbb{E}_{\mathbf{r}, \mathbf{s}} \left[\sum_{s^{(k)} \in K(\mathbf{r}, \mathbf{s})} p_i(s^{(k)}) \right] - \mathbb{E}_{\mathbf{r}, \mathbf{s}, c_i(\mathbf{r})} \left[\sum_{s^{(k)} \in K'(c_i(\mathbf{r}), \mathbf{s})} p_i(s^{(k)}) \right] \\
& \geq \frac{1}{\eta} \left(-\ell' \cdot \varepsilon + \mathbb{E}_{\mathbf{r}, \mathbf{s}} [W_i(K(\mathbf{r}, \mathbf{s}))] - \mathbb{E}_{\mathbf{r}, \mathbf{s}} [W_i(V(\mathbf{r}, \mathbf{s}))] - \mathbb{E}_{\mathbf{r}, \mathbf{s}, c_i(\mathbf{r})} \left[\sum_{P \in \mathcal{P}} \text{DIFF}(P) \right] \right) \\
& \geq -\frac{1}{\eta} (\ell' \cdot \varepsilon + \Delta) - \frac{1}{\eta} \mathbb{E}_{\mathbf{r}, \mathbf{s}, c_i(\mathbf{r})} \left[\sum_{P \in \mathcal{P}} \text{DIFF}(P) \right]
\end{aligned}$$

For every type r, s , and realized type $c_i(r)$, $\Delta_{i, c_i}(r, s) = \mathbb{E}_{t_{-i} \sim \mathcal{D}_{-i}} [v_i(r, x(s, t_{-i})) - v_i(c_i(r), x(s, t_{-i}))] \in [-\text{dist}_i(r, c_i(r)), \text{dist}_i(r, c_i(r))]$ (recall that $\text{dist}_i(r, c_i(r)) = \max_{o \in \mathcal{O}} |v_i(r, o) - v_i(c_i(r), o)|$). Thus

$$\sum_{P \in \mathcal{P}} \text{DIFF}(P) \leq 2 \sum_{j=1}^{\ell'} \text{dist}_i(r^{(j)}, c_i(r^{(j)})),$$

and

$$\mathbb{E}_{\mathbf{r}, \mathbf{s}, c_i(\mathbf{r})} \left[\sum_{P \in \mathcal{P}} \text{DIFF}(P) \right] \leq 2\ell' d_w(\mathcal{D}_i, \mathcal{D}'_i).$$

Therefore,

$$\mathbb{E}_{\mathbf{r}, \mathbf{s}} \left[\sum_{s^{(k)} \in K(\mathbf{r}, \mathbf{s})} p_i(s^{(k)}) / \ell' \right] \geq \mathbb{E}_{\mathbf{r}, \mathbf{s}, c_i(\mathbf{r})} \left[\sum_{s^{(k)} \in K'(c_i(\mathbf{r}), \mathbf{s})} p_i(s^{(k)}) / \ell' \right] - \frac{1}{\eta} \left(\varepsilon + \frac{\Delta}{\ell'} \right) - \frac{2}{\eta} d_w(\mathcal{D}_i, \mathcal{D}'_i).$$

If v_i is non-increasing w.r.t. c_i , then $\Delta_{i, c_i}(\cdot, \cdot)$ is a non-negative function. Then

$$\sum_{P \in \mathcal{P}} \text{DIFF}(P) \leq \sum_{P \in \mathcal{P}} \sum_{(r^{(j)}, s^{(k)}) \in P \cap K(\mathbf{r}, \mathbf{s})} \Delta_{i, c_i}(r^{(j)}, s^{(k)}) \leq \sum_{(r^{(j)}, s^{(k)}) \in K(\mathbf{r}, \mathbf{s})} \Delta_{i, c_i}(r^{(j)}, s^{(k)}),$$

and

$$\mathbb{E}_{\mathbf{r}, \mathbf{s}, c_i(\mathbf{r})} \left[\sum_{P \in \mathcal{P}} \text{DIFF}(P) \right] \leq \mathbb{E}_{\mathbf{r}, \mathbf{s}} \left[\sum_{(r^{(j)}, s^{(k)}) \in K(\mathbf{r}, \mathbf{s})} c_i(\mathbf{r}), t_{-i} \sim \mathcal{D}_{-i} \left[v_i(r^{(j)}, x(s^{(k)}, t_{-i})) - v_i(c_i(r^{(j)}), x(s^{(k)}, t_{-i})) \right] \right].$$

□

Finally, we are ready to prove Lemma 23. Note that for every $s^{(k)}$, $p_i(s^{(k)}) \leq 1$ since \mathcal{M} is -IR. We have

$$(1 - \eta) \cdot \mathbb{E}_{\mathbf{r}, \mathbf{s}} \left[\sum_{s^{(k)} \in K(\mathbf{r}, \mathbf{s})} p_i(s^{(k)}) / \ell' \right] \geq \mathbb{E}_{\mathbf{r}, \mathbf{s}} \left[\sum_{s^{(k)} \in K(\mathbf{r}, \mathbf{s})} p_i(s^{(k)}) / \ell' \right] - \eta.$$

The lemma follows from Lemma 26 and Corollary 3. □

Proof of Theorem 4: First, by Lemma 21, we can lower bound the revenue of \mathcal{M}' under \mathcal{D}' from agent i in Phase 2 $\text{REV-SECOND}_i(\mathcal{M}', \mathcal{D}')$ by $(1 - \eta) \mathbb{E}_{\mathbf{r}, \mathbf{s}} \left[\sum_{s^{(k)} \in O(\mathbf{r}, \mathbf{s})} p_i(s^{(k)}) / \ell' \right]$, where $O(\mathbf{r}, \mathbf{s})$ is the matching produced by Algorithm 2 on \hat{G}_i . Lemma 21 also provides an equivalent expression for the revenue of \mathcal{M} under \mathcal{D} from agent i : $\text{REV}_i(\mathcal{M}, \mathcal{D}) = \mathbb{E}_{\mathbf{s}} \left[\sum_{k \in [\ell']} p_i(s^{(k)}) / \ell' \right]$.

We choose the parameters according to Theorem 3, that is, for any $\psi \in (0, 1)$, we set $\delta = \Theta(\frac{\psi}{\log \ell})$, $\eta' = \Theta(\psi)$ and $d \geq \frac{\ell \log \ell}{\eta'^2}$. Theorem 3 implies that $\mathbb{E}_{\mathbf{r}, \mathbf{s}} [W_i(O(\mathbf{r}, \mathbf{s}))] = \mathbb{E}_{\mathbf{r}, \mathbf{s}} [W_i(V(\mathbf{r}, \mathbf{s}))] - O(d\ell\psi)$, that is in expectation $O(\mathbf{r}, \mathbf{s})$ has close to maximum weight. We will specify the choice of the other parameters ℓ , η , and ψ later. By Lemma 23, we know that

$$\text{REV-SECOND}_i(\mathcal{M}', \mathcal{D}') \geq \text{REV}_i(\mathcal{M}, \mathcal{D}) - \left(\eta + \sqrt{\frac{|T'_i|}{\ell}} + \frac{\varepsilon}{\eta} + \frac{O(\psi)}{\eta} \right) - \frac{2d_w(\mathcal{D}_i, \mathcal{D}'_i)}{\eta} \quad (16)$$

Combining Inequality (16) with Lemma 8, we can obtain the following lower bound on $\text{REV}(\mathcal{M}', \mathcal{D}')$.

$$\text{REV}(\mathcal{M}', \mathcal{D}') \geq \text{REV}(\mathcal{M}, \mathcal{D}) - \sum_{i \in [n]} \left(\eta + \sqrt{\frac{T'_i}{\ell}} + \frac{\varepsilon}{\eta} + \frac{O(\psi)}{\eta} \right) - \frac{2d_w(\mathcal{D}, \mathcal{D}')}{\eta} - n\sqrt{\delta}(\log(2\ell) + 1) \quad (17)$$

Now we set $\ell = \frac{T'}{\varepsilon}$, $\psi = \frac{\varepsilon^2}{\log \ell}$, and we can choose η to be $O\left(\sqrt{\varepsilon + \frac{d_w(\mathcal{D}, \mathcal{D}')}{n}}\right)$ so that

$$\begin{aligned} \text{REV}(\mathcal{M}', \mathcal{D}') &\geq \text{REV}(\mathcal{M}, \mathcal{D}) - O\left(n \left(\sqrt{\varepsilon} + \sqrt{\varepsilon + \frac{d_w(\mathcal{D}, \mathcal{D}')}{n}} \right)\right) - \Theta(n\varepsilon) \\ &\geq \text{REV}(\mathcal{M}, \mathcal{D}) - O(n\sqrt{\varepsilon}) - O\left(\sqrt{n \cdot d_w(\mathcal{D}, \mathcal{D}')}\right) \end{aligned}$$

Plugging in our choice of the parameters to Lemma 8, we can conclude that both the computational and query complexity of \mathcal{M}' is $\text{poly}(n, T', 1/\varepsilon)$.

If $c_i(\cdot)$ that v_i is non-increasing w.r.t. $c_i(\cdot)$, we can replace the last term $-\frac{2d_w(\mathcal{D}_i, \mathcal{D}'_i)}{\eta}$ in Inequality (16) by

$$-\frac{1}{\eta \ell'} \mathbb{E}_{\mathbf{r}, \mathbf{s}} \left[\sum_{(r^{(j)}, s^{(k)}) \in O(\mathbf{r}, \mathbf{s})} c_i(\mathbf{r}), t_{-i} \sim \mathcal{D}_{-i} \left[v_i(r^{(j)}, x(s^{(k)}, t_{-i})) - v_i(c_i(r^{(j)}), x(s^{(k)}, t_{-i})) \right] \right].$$

Note that this quantity is the same as

$$-\frac{1}{\eta} \mathbb{E}_{t \sim \mathcal{D}'} \left[\mathbb{E}_{c_i(t_i)} [v_i(t_i, x'(t)) - v_i(c_i(t_i), x'(t))] \right].$$

Hence, for any $\eta \in (0, 1)$, we can improve the result to

$$\text{REV}(\mathcal{M}', \mathcal{D}') \geq \text{REV}(\mathcal{M}, \mathcal{D}) - n\sqrt{\varepsilon} - O\left(n\eta + \frac{n\varepsilon}{\eta}\right) - \frac{\sum_{i \in [n]} \mathbb{E}_{t \sim \mathcal{D}'} [\mathbb{E}_{c_i(t_i)} [v_i(t_i, x'(t)) - v_i(c_i(t_i), x'(t))]]}{\eta}.$$

□