Sample-Efficient Black-box Reductions in Bayesian Mechanism Design with Independent Items

Arya Maheshwari*[1], S. Matthew Weinberg[2], and Eric Xue[2]

[1]Carnegie Mellon University
[2]Princeton University

April 3, 2026

Abstract

We consider the canonical multi-dimensional mechanism design setting: a seller has m items for sale to n additive bidders, with bidder k's value $v_{k\ell}$ for item $\ell \in [m]$ drawn independently from distribution $D_{k\ell}$ supported on $[0,1]$. The seller begins with a mechanism $\mathcal{M}'$ that is ε-Bayesian Incentive Compatible, and desires a new mechanism $\mathcal{M}$ that is exactly Bayesian Incentive Compatible with minimal loss in revenue. Our main result is a black-box reduction that, given only sample access to each $D_{k\ell}$ and query access to $\mathcal{M}'$, produces an $\mathcal{M}$ with revenue at least that of $\mathcal{M}'$ within an additive $O(n\sqrt{\varepsilon})$, resolving the main open problem left in [GW18]. The main distinction to prior work ([HKM11, DW12, RW18, DHKN17, COVZ19]) is our sample complexity of $\text{poly}(n, m, 1/\varepsilon)$ rather than $\text{poly}(n, \exp(m), 1/\varepsilon)$. A corollary of our main result (together with [GW18]) is an algorithm to find an up-to-ε revenue-optimal mechanism using only $\text{poly}(n, m, 1/\varepsilon)$ samples from each $D_{k\ell}$. We also extend our results to the more general notion of Lipschitz valuations over independent items, which includes additive, unit-demand, and various other natural classes of valuations.

*Majority of work done while at Princeton University. Correspondence to `aryam@cs.cmu.edu`.

Contents

1 Introduction

The revenue-optimal sale of multiple items to multiple bidders by a single seller is a fundamental problem that has become a cornerstone of Economics and Computation since its introduction by seminal work of [CHK07]. In this problem, the seller has a prior distribution over buyers' valuations, and her goal is to design a truthful mechanism that maximizes her expected revenue over the randomness of these valuations.

Due to known intractabilities along several dimensions with arbitrary priors [CM88, BCKW15, HN13], an overwhelming fraction of the literature therefore focuses on the "independent items" regime – the prior draws each bidder's value for each item independently. In this regime, seminal work of [Mye81] characterizes the revenue-optimal truthful mechanism for selling a single item to multiple bidders. With multiple heterogeneous items, works of [CHK07, CHMS10, CMS15, HN17, LY13, BILW20, Yao15, RW18, CDW16, CZ17, EFF+21, JL24] design constant-factor approximations in richer and richer settings.

One direction left unaddressed by these works is how exactly the auctioneer forms this prior, and in particular how much data is necessary to form a sufficiently accurate prior. Seminal work of [CR14] initiates the study of *sample complexity* to address this challenge, and subsequent works of [DHP16, RS16, GN17, Syr17, GHZ19, GJZ21, HHSW21, JLX23, MR16, MR16, BSV16, BSV18, Syr17, TW25] recover a wide range of the aforementioned known-prior results with instead only polynomially-many samples from the prior.

Sample-Efficient Near-Optimal Mechanisms. [GW18] find *near-optimal* (i.e. up to an additive ε, rather than a constant-factor approximation) auctions with multiple heterogeneous items using polynomially-many samples. However, their work has the following shortcoming: all other aforementioned works design *exactly truthful mechanisms* that *suffer error only in revenue guarantees*. [GW18], on the other hand, designs an *approximately truthful mechanism* that *suffers error both in incentive compatibility and in revenue guarantees*. Formally, their output mechanism is ε-Bayesian Incentive Compatible (ε-BIC) and achieves revenue within an additive ε of the revenue-optimal mechanism using $\text{poly}(n, m, 1/\varepsilon)$ samples. Small loss in revenue is unavoidable – without knowing the prior exactly, there is simply no way to guess the precise revenue-optimal auction. Small loss in incentive guarantees, however, may be entirely avoidable. Moreover, even a small loss in incentive guarantees, especially for Bayesian Incentive Compatibility, carries additional cost – if any single bidder executes a profitable deviation, this distorts other bidders' incentives, perhaps creating even larger incentives for those bidders to deviate. The primary open question left open by [GW18] is to design *exactly truthful mechanisms* with comparable revenue guarantees using polynomially-many samples. A corollary of our main result (together with [GW18]) resolves this open problem.

ε-BIC-to-BIC Reductions. The ideal approach to resolve this problem would be an ε-BIC-to-BIC reduction: a procedure to take as input any ε-BIC mechanism M and output an exactly BIC mechanism M' with revenue $\text{poly}(n, m, \varepsilon)$ within that of M. Indeed, such reductions exist in quite broad settings – building from algorithms-to-BIC reductions of [HKM11, BH11, DHKN17], works of [DW12, RW18, COVZ19] design ε-BIC-to-BIC reductions. [DHKN17, COVZ19] moreover require only sample access to the underlying prior. But, all prior works require complexity polynomial in the size of an ε-net of a single bidder's valuation space, which is exponential in the number of items.

[GW18] therefore concretely ask whether a ε-BIC-to-BIC reduction exists with $\text{poly}(n, m, 1/\varepsilon)$

samples in the case of independent items. Our main result resolves this question in the affirmative, and implies the earlier aforementioned resolution after plugging in [GW18]'s ε-BIC mechanism. The remainder of this section overviews our main result and technical highlights in more detail.

Key Conceptual Challenge. On first glance, it may seem surprising that this problem stood open for eight years. After all, we can certainly "learn" a product distribution by learning its marginals, and indeed learning the marginals does suffice to "learn" the distribution. But, typical learning comes with some (often small) error, and so *the key challenge is that exact incentive compatibility requires zero tolerance for error, and even an exponentially small failure probability is insufficient.* Most prior work in sample complexity of mechanism design dodges this challenge by explicitly learning over on *Dominant Strategy Incentive Compatible* mechanisms (where incentive compatibility is a property of the mechanism that holds *for all distributions and requires no learning whatsoever*) [CR14, DHP16, MR16, MR16, RS16, BSV16, Syr17, CD17, BSV18, GHZ19, GJZ21, HHSW21, JLX23, TW25]. But, if we wish to consider near-optimal mechanisms in multi-dimensional settings, we *must consider Bayesian Incentive Compatible mechanisms*,[1] where *Incentive Compatibility itself depends on D and therefore must be "learned"*.

Indeed, only two prior works directly address this challenge, and do so using machinery similar to Bernoulli factories[2] [DHKN17, COVZ19].[3] Our approach certainly leverages their prior machinery, but our key technical contribution is a novel method to protect exact Bayesian Incentive Compatibility despite inevitable low-probability failures in learning.

1.1 Main Result 1: Polynomial Sample Complexity

The main result of this paper is a sample-efficient and revenue-preserving ε-BIC-to-BIC transformation that converts an ε-BIC mechanism $\mathcal{M}'$ for n additive bidders whose values for m items are drawn from the product distribution $D := \times_{k\in[n],\ell\in[m]} D_{k\ell}$ into an exactly BIC mechanism $\mathcal{M}$. Like [GW18], we assume that values are bounded,[4] and our main result extends well beyond additive valuations to the entire class of Lipschitz valuations.[5] The transformation is sample-efficient in that it requires only a polynomial number of samples from D, and revenue-preserving in that the revenue loss is bounded by $n\sqrt{\varepsilon}$.[6] Moreover, the transformation only requires query access to the mechanism $\mathcal{M}'$, that is, we treat $\mathcal{M}'$ as a black-box that takes as input a profile of values and outputs an allocation and payments.

Main Result 1 (Informal, see Theorem 6.1). Given sample access to a bounded product distribution $D := \times_{k,\ell} D_{k\ell}$ for n bidders with Lipschitz valuations and m items, and query access to an ε-BIC mechanism $\mathcal{M}'$ for D, we can construct a mechanism $\mathcal{M}$ that is exactly BIC for D and

---

[1] Due to known separations between the achievable revenue of the optimal DSIC mechanism and the optimal BIC mechanism even in extremely simple settings [Yao17].

[2] A Bernoulli factory is a procedure by which, given *only sample access* to a coin with bias p, one produces sample access to a coin with bias exactly $f(p)$.

[3] Other works are either discussed above, accept a ε-BIC mechanism [HKM11, BH11, GW18, BCWZ17, GHTZ21, BB24], or work in a model where the distribution is precisely known [DW12, RW15].

[4] Bounded valuations is a necessary assumption – with only sample access to an unbounded distribution, no guarantees are possible.

[5] A class of valuation functions over m items is *c-Lipschitz* if it every valuation $v(\cdot)$ can be parameterized by an m dimensional vector $\vec{x}$ and: (a) $|v_{\vec{x}}(S) - v_{\vec{x}}(S \cup \{i\})| \le c$ for all $\vec{x}, S, i$, and (b) $|v_{\vec{x}}(S) - v_{\vec{y}}(S)|$ is at most c times the number of coordinates on which $\vec{x}$ and $\vec{y}$ differ.

[6] This revenue loss term matches that of prior ε-BIC-to-BIC transformations [COVZ19], and captures the natural poly(ε) loss incurred per bidder due to starting with an ε-BIC mechanism.

whose revenue is at most $O(n\sqrt{\varepsilon})$ less than the revenue of $\mathcal{M}'$. That is,

$$\textsc{Rev}(\mathcal{M}; D) \geq \textsc{Rev}(\mathcal{M}', D) - O(n\sqrt{\varepsilon})$$

The sample complexity of this procedure is $\mathbf{poly}(\mathbf{m}, \mathbf{n}, \frac{\mathbf{1}}{\varepsilon})$**.**

Like prior work on ε-BIC-to-BIC transformations for revenue, Theorem 6.1 is in fact more general and allows $\mathcal{M}'$ to be an ε-BIC mechanism for a different distribution D'. In this case, the additional revenue loss is bounded by the distance between D and D' with respect to a natural metric defined over the space of product distributions (see Section 2 for the formal definition of this metric). For simplicity, we focus on the case in which $D' = D$ for now.

Theorem 6.1 resolves [GW18]'s open problem, and implies the following corollary.

Corollary (via Theorem 4 of [GW18]). For all $\varepsilon, \delta > 0$, the sample complexity of learning, with probability $1 - \delta$, an *exactly* BIC mechanism for n bidders with independent bounded values for m items that achieves the optimal revenue (up to an additive ε loss) is $\text{poly}(m, n, 1/\varepsilon, \log(1/\delta))$.

Remark: Computational Complexity. Like [GW18], our reduction is sample-efficient but not computationally-efficient. The source of computational inefficiency stems from the same reason: a ε-net for $[0,1]^m$ still requires $\exp(m)$ points, even if those points are generated from a product distribution. In [GW18], this shows up in the runtime of solving a linear program for the optimal mechanism for an empirical distribution – although that empirical distribution is a product distribution with a concise description, the linear program requires a variable for every element in its support (which remains exponential). In our work, this shows up in the number of "surrogates" needed in order for the "replica-surrogate matching" to guarantee a nearby replica for every possible type (see Section 3 for an overview of the replica-surrogate matching). Both barriers are long-standing and well-established with regards to their respective techniques ([KMS+19, CWX25] discuss barriers to computing optimal mechanisms, and see [HKM11, DW12, DHKN17, COVZ19] for the same barrier in replica-surrogate matchings).

1.2 Main Result 2: Meta-Framework

As an intermediate step in our analysis, we derive a *meta-framework* that unifies the approaches of prior work as well as our own. More specifically, our meta-framework takes as input subroutines ("meta-inputs") and specifies a list of conditions for each subroutine that need to be satisfied for the output of our meta-framework to be a revenue-preserving ε-BIC-to-BIC transformation. In other words, our meta-framework reduces the design of such transformations to the design of specific subroutines. The meta-framework is agnostic to assumptions about the type space and valuations (like the independent items and Lipschitz valuations conditions in Main Result 1) and hence applies to more general mechanism design settings. Moreover, our meta-framework retroactively highlights exactly the innovations of prior work – every black-box reduction based on replica-surrogate matching follows our meta-framework, and so our meta-framework formally codifies this outline. Moreover, future developments on black-box reductions in Bayesian mechanism design can now more cleanly present their innovations, rather than additionally repeating similar (extremely technical) arguments to build the replica-surrogate outline.

Main Result 2 (Informal, see Theorem 5.1) There exists a reduction from designing revenue-preserving ε-BIC-to-BIC transformations to designing the following subroutines: $\textsc{Draw-}\mathcal{S}$, $\textsc{Draw-}\mathcal{R}$, $\textsc{EstMechanism}$, $\textsc{Preprocess}$, $\textsc{Compute-}\alpha$, $\textsc{Match}$, and $\textsc{Payment}$.

1.3 Related Work

Black-box reductions in Bayesian Mechanism Design. Our work follows a long line of black-box reductions in Bayesian Mechanism Design, starting from [HL10] [HKM11, BH11, DW12, CIL12, RW15, DHKN17, COVZ19, GLT20]. Among these, [HKM11] introduces the replica-surrogate matching framework employed by our work and also [DW12, RW15, DHKN17, COVZ19] – we provide further details in Section 3 on the different components provided by these works. The biggest distinction between our work and these prior works is our focus on *improved sample complexity for restricted priors* – all prior reductions make no assumptions on the prior (and therefore require sample complexity exponential in m simply to learn the distribution), whereas our work improves efficiency throughout the entire pipeline for product distributions. [CIL12, GLT20] prove impossibility results on black-box reductions in Bayesian Mechanism Design.

In comparison to our work, [CIL12, GLT20] focuses on *queries to the mechanism* whereas our work focuses on the *samples from* D. Notably, [GLT20]'s exponential query lower bound also follows from the same ε-net challenges that prevent [GW18] and our work from being computationally-efficient. [CFPS21] study ε-BIC-to-BIC transformations that simultaneously preserve welfare exactly and lose revenue parameterized by ε (and the size of each bidder's type space) and use machinery outside of replica-surrogate matching to do so. One key distinction between [CFPS21] to our work and that of [DHKN17, COVZ19] is that [CFPS21] assumes complete knowledge of the distribution and interim payments of the input mechanism, and so there is no question of sample complexity of D nor query complexity. A second key distinction between [CFPS21] (and [HKM11, BH11, DW12, RW15, DHKN17, COVZ19], as noted above) to our work is the size of a single bidder's full type-space as the relevant complexity parameter (which is exponential in m).

Sample Complexity of Bayesian Mechanism Design. Following [CR14], there is an extensive body of works considering the sample complexity of Bayesian Mechanism Design [CR14, DHP16, RS16, BSV16, Syr17, BSV18, GHZ19, GJZ21, HHSW21, JLX23, MR16, MR16, CD17, Syr17, TW25, GW18, BCWZ17, GHTZ21, BB24]. These works largely fall into three categories: (a) sample-complexity of *near-optimal* mechanisms in *single-dimensional* settings [CR14, DHP16, RS16, GN17, Syr17, GHZ19, GJZ21, HHSW21, JLX23], (b) sample-complexity of *approximately-optimal* mechanisms in *multi-dimensional* settings [MR16, MR16, BSV16, BSV18, Syr17, TW25], and (c) *near-optimal* mechanisms in *multi-dimensional* settings [GW18, BCWZ17, GHTZ21, BB24]. Except for (c), all works learn over a structured class of *exactly* incentive compatible mechanisms. However, while optimal single-dimensional mechanisms are perfectly characterized by [Mye81], and simple auctions are approximately optimal in multi-dimensional settings [CHK07, CHMS10, CMS15, HN17, LY13, BILW20, Yao15, RW18, CDW16, CZ17, EFF+21, JL24], no structured class of auctions is known to be near-optimal in multi-dimensional settings,[7] precluding (c) from exactly incentive compatible mechanisms via the same techniques. Through this lens, our work catches the sample complexity of near-optimal multi-dimensional auctions up to that of (a) and (b) by providing a novel method for exact incentive compatibility via ε-BIC-to-BIC reductions.

In a separate direction, [DRY15, AKW14, BGMM18, DZ20] study approximately optimal mechanisms in the data poor regime, where only very few samples are available. [FL20] study the sample complexity of learning approximate equilibria in non-truthful auctions, such as first price auctions.

[7] Moreover, there are several formal barriers noting the lack of structure in optimal multi-dimensional auctions [DDT14, CDP+14, CDO+15, HR15, DDT17].

1.4 Roadmap

We state preliminaries in Section 2. Section 3 overviews replica-surrogate matching, the key technical barriers to a sample-efficient ε-BIC-to-BIC transformation, and the proof outline of our main results. Section 4 standardizes some notation specific to replica-surrogate matching that we use through the paper and overviews some key tools from prior work, such as entropy-regularized matching and Bernoulli factories. Section 5 presents the meta-framework, including the main algorithms for our ε-BIC-to-BIC transformation and the specifications for the meta-inputs they require. Proofs of the incentive and revenue guarantees of the meta-framework are deferred to Section A. Section 6 presents our main polynomial sample complexity result, with implementations of meta-inputs that achieve this result. The analysis of these meta-input implementations is deferred to Section C.

2 Preliminaries

Conventions. All logs are natural logs unless otherwise specified. We use $[N]$ to denote the set $\{1, \dots, N\}$ for $N \in \mathbb{N}$. For a product distribution $D = \times_{k\in[n]} D_k$, we will use $D_{\text{-}k}$ to denote the product $\times_{j\in[n]\setminus\{k\}} D_j$. We use $\Delta(U)$ to denote the set of distributions over a set U. We will sometimes abuse notation and use "set" to refer to a multiset. We will also abuse notation to write the density of a continuous distribution F at point x as $\Pr_{X\sim F}[X = x]$, or simply $\Pr_F[x]$ when the meaning is clear from context.

Mechanism design model. We consider the following standard mechanism design model. There is a single seller and n bidders. Each bidder $k \in [n]$ has a *type* t_k in some type space $\mathcal{T}_k$. Each bidder's type is drawn independently from some distribution D_k supported on $\mathcal{T}_k$. We let $D := \times_{k\in[n]} D_k$ and $\mathcal{T} := \times_{k\in[n]} \mathcal{T}_k$ denote the overall type distribution and space over the bidders. Each bidder has a valuation function $v_k : \mathcal{T}_k \times \mathcal{O} \to [0, 1]$ that maps a type and an *outcome*, in *outcome space* $\mathcal{O}$, to a non-negative value, which we assume to be normalized to $[0, 1]$.

We consider *downward-closed* outcome spaces where each $o \in \mathcal{O}$ can be written as a tuple over bidders $o = (o_1, \dots, o_n)$, and for any $o \in \mathcal{O}$ and any $k \in [n]$, the outcome $(o_1, \dots, o_{k-1}, \perp, o_{k+1}, \dots, o_n)$ is also in $\mathcal{O}$, where $\perp$ denotes a "null" outcome for bidder k. That is, each bidder can always receive nothing without impacting other bidders: for any o with $o_k = \perp$ and any $t_k \in \mathcal{T}_k$, $v_k(t_k, o) = 0$.

An *allocation rule* $\mathcal{A} : \mathcal{T} \to \Delta(\mathcal{O})$ maps a type profile of bidders to a (possibly random) outcome. A *payment rule* $p : \mathcal{T} \to \Delta(\mathbb{R}^n)$ maps a type profile of bidders to a (possibly random) vector of payments. A *mechanism* $\mathcal{M}$ is then a pair $(\mathcal{A}, p)$ of allocation and payment rule. Bidders have quasi-linear utility: for outcome o and payment p_k, the utility of bidder k with type t_k is $u_k := v_k(t_k, o) - p_k$. We will abuse notation and write $v_k(t_k, \mathcal{A}(t))$ to mean the *expected* value $\mathbb{E}_{o\sim\mathcal{A}(t)}[v_k(t_k, o)]$, and similarly $u_k(t_k, \mathcal{M}(t))$ to mean the *expected* utility $\mathbb{E}_{o,p\sim(\mathcal{A}(t),p(t))}[v_k(t_k, o) - p_k]$, except where otherwise noted.

For a mechanism $\mathcal{M} = (\mathcal{A}, p)$, for any bidder k, the *interim value* of type t_1 for type t_2 with respect to bidder distribution D is given by

$$\mathrm{val}^{(k)}(t_1, t_2) := \mathop{\mathbb{E}}_{t_{\text{-}k}\sim D_{\text{-}k}}[v_k(t_1, \mathcal{A}(t_2; t_{\text{-}k}))].$$

Similarly, the *interim payment* for a type t is given by

$$p^{(k)}(t) := \mathop{\mathbb{E}}_{t_{\text{-}k} \sim D_{\text{-}k}} [p_k(t; t_{\text{-}k})]$$

where the expectation is also over the randomness of the payment rule $p(\cdot)$, and the *interim utility* of type t_1 for type t_2 is

$$u^{(k)}(t_1, t_2) := \mathop{\mathbb{E}}_{t_{\text{-}k} \sim D_{\text{-}k}} [u_k(t_1, \mathcal{M}(t_2; t_{\text{-}k})] = \mathrm{val}^{(k)}(t_1, t_2) - p^{(k)}(t_2)$$

The seller's optimization objective is the *revenue* obtained by $\mathcal{M}$ in expectation over distribution D, denoted $\textsc{Rev}(\mathcal{M}; D) := \mathbb{E}_{t \sim D}[\sum_{k \in [n]} p_k(t)]$, where the expectation is again also over the randomness of $p(\cdot)$.[8]

Incentive compatibility and individual rationality. A mechanism $\mathcal{M} = (\mathcal{A}, p)$ can satisfy the following properties:

- Individually Rational (IR): $\mathcal{M}$ is IR if $\forall k \in [n], \forall t_k \in \mathcal{T}_k$,

$$u^{(k)}(t_k, t_k) \geq 0$$

- Bayesian Incentive Compatible (BIC): $\mathcal{M}$ is BIC over a distribution D if $\forall k \in [n], \forall t_a, t_b \in \mathcal{T}_k$,

$$u^{(k)}(t_a, t_a) \geq u^{(k)}(t_a, t_b)$$

- ε-Bayesian Incentive Compatible (ε-BIC): for $\varepsilon \geq 0$, $\mathcal{M}$ is ε-BIC over a distribution D if $\forall k \in [n], \forall t_a, t_b \in \mathcal{T}_k$,

$$u^{(k)}(t_a, t_a) \geq u^{(k)}(t_a, t_b) - \varepsilon$$

Couplings. Let μ and ν be distributions supported on spaces X and Y respectively. A *coupling* γ of μ and ν is a joint distribution supported on $X \times Y$ whose marginals are μ and ν. We denote the set of all couplings between μ and ν by $\Pi(\mu, \nu)$. We write $\gamma(x)$ to denote a random variable distributed according to the conditional distribution of the Y-coordinate given the X-coordinate equals x. Thus, if $x \sim \mu$, $\gamma(x) \sim \nu$.

Furthermore, given couplings $c \in \Pi(\mu, \nu)$ and $c' \in \Pi(\nu, \xi)$, we define their *composition* $c' \circ c \in \Pi(\mu, \xi)$ in the natural way. Formally, for any $x \in X$, the conditional distribution $(c' \circ c)(x)$ is the mixture

$$(c' \circ c)(x) := \mathbb{E}_{y \sim c(x)}\big[c'(y)\big].$$

That is, a sample $(x, z) \sim (c' \circ c)$ is obtained by sampling $x \sim \mu$, then $y \sim c(x)$, then $z \sim c'(y)$.

Let $d : (X \times Y) \to \mathbb{R}_{\geq 0}$ be a cost/distance function. For a coupling $\gamma \in \Pi(\mu, \nu)$, we define the *distance between μ and ν with respect to γ* by $d_\gamma(\mu, \nu) := \mathbb{E}_{(x,y) \sim \gamma}[d(x, y)]$. The *Wasserstein distance* d^W between μ and ν is defined as the smallest distance over all couplings:

$$d^W(\mu, \nu) := \min_{\gamma \in \Pi(\mu, \nu)} d_\gamma(\mu, \nu).$$

[8] Henceforth we will not explicitly mention when expectations are over the randomness of the payment rule as this should be clear from context. In particular we will sometimes abuse notation to treat p as a map from $\mathcal{T} \to [0, 1]^n$, instead of $T \to \Delta([0, 1]^n)$, by taking the expectation implicitly.

Suppose $\mu = \times_{i=1}^N \mu_i$ and $\nu = \times_{i=1}^N \nu_i$ are product distributions over product spaces $X = \times_{i=1}^N X_i$ and $Y = \times_{i=1}^N Y_i$, with marginals $(\mu_i)_i$ and $(\nu_i)_i$, respectively. Given couplings $\gamma_i \in \Pi(\mu_i, \nu_i)$, their *product coupling* is $\Gamma := (\times_{i=1}^N \gamma_i) \in \Pi(\mu, \nu)$. That is, $(x, y) \sim \Gamma$ is obtained by *independently* sampling $(x_i, y_i) \sim \gamma_i$ for each i and setting $x := (x_1, \dots, x_N)$ and $y := (y_1, \dots, y_N)$. We denote the set of all product couplings by $\Pi_{\mathrm{prod}}(\mu, \nu) := \{\times_{i=1}^N \gamma_i \ : \ \gamma_i \in \Pi(\mu_i, \nu_i)\} \subseteq \Pi(\mu, \nu)$. We then define the *product–restricted Wasserstein distance* to be

$$d^W_{\mathrm{prod}}(\mu, \nu) \ := \min_{\Gamma \in \Pi_{\mathrm{prod}}(\mu, \nu)} d_\Gamma(\mu, \nu).$$

We call any minimizer $\Gamma^* \in \arg\min_{\Gamma \in \Pi_{\mathrm{prod}}(\mu,\nu)} d_\Gamma(\mu, \nu)$ an *optimal product coupling.*

Distance between types. For each $k \in [n]$, we consider a distance metric $\mathrm{dist}_k(\cdot, \cdot)$ (on the type space $\mathcal{T}_k \cup \mathcal{T}'_k$) defined by a maximum valuation difference: for any two types t_1, t_2, $\mathrm{dist}_k(t_1, t_2) = \max_{o \in \mathcal{O}} |v_k(t_1, o) - v_k(t_2, o)|$. We say that the types t_1, t_2 are *d-close* if $\mathrm{dist}_k(t_1, t_2) \le d$.

Unless otherwise specified, we will always be referring to the metric $\mathrm{dist}_k(\cdot, \cdot)$ when talking about distances. In particular, (product-restricted) Wasserstein distances between distributions supported on type spaces are assumed to be based on $\mathrm{dist}_k(\cdot, \cdot)$ as the distance function.

2.1 Problem Statement

We now state precisely the ε-BIC-to-BIC transformation problem. We are given query access to a mechanism $\mathcal{M}' = (\mathcal{A}', p')$. $\mathcal{M}'$ achieves some expected revenue $\textsc{Rev}(\mathcal{M}'; D')$ over a bidder distribution $D' := \times_{k \in [n]} D'_k$ (with corresponding type space $\mathcal{T}' := \times_{k \in [n]} \mathcal{T}'_k$) to which we have sample access. Moreover, $\mathcal{M}'$ is IR and ε-BIC over D' for some $\varepsilon \ge 0$. We assume bounded valuations, i.e. $v_k(\cdot, \cdot) \in [0, 1]$ (without loss of generality), and we further assume that payments $p'_k(\cdot)$ are in $[0, 1]$ for the input mechanism $\mathcal{M}'$. We also have a target distribution $D := \times_{k \in [n]} D_k$ (with corresponding type space $\mathcal{T} := \times_{k \in [n]} \mathcal{T}_k$) to which we have sample access.

Our goal is then to design a mechanism $\mathcal{M}$ for D that is IR and (*exactly*) BIC over D, and achieves revenue $\textsc{Rev}(\mathcal{M}; D)$ that approximates $\textsc{Rev}(\mathcal{M}'; D')$.[9] Our metric of interest in terms of efficiency will be the *sample complexity* (from D and D') required to achieve these incentive and revenue guarantees.

We focus our main presentation producing an (Interim) Individually Rational mechanism while assuming payments are bounded in $[0, 1]$. See Section **??** for a reduction to produce an ex-post IR mechanism and to accommodate unbounded payments.

3 Technical Background, Challenges, and Overview

Here, we provide a technical overview of replica-surrogate matchings (Section 3.1), overview the technical contributions of prior work and how they fit into our meta-framework (Section 3.2), overview the technical barriers to improved sample complexity with product distributions (Section 3.3), and finally overview our solution (Section 3.4).

[9]The specific notion of approximation will be an additive $O(\mathrm{poly}(\varepsilon) \cdot n)$ loss, plus a term dependent on the distance between D and D'.

3.1 Replica-Surrogate Matching: Background

Let us first recall the problem at hand: we have black-box access to a mechanism $\mathcal{M}'$ and a distribution D', and all we know is that $\mathcal{M}'$ is ε-BIC for D' and that we're happy with *the revenue* $\mathcal{M}'$ *gets on* D', $\text{Rev}(\mathcal{M}', D')$. The goal of Replica-Surrogate Matching is to somehow use the process of drawing samples from D' and participating in $\mathcal{M}'$ as a black-box.

[HKM11] propose the following framework: rather than have real bidders *directly participate* in $\mathcal{M}'$, have each real bidder instead *purchase a surrogate to represent them* in $\mathcal{M}'$. That is, our mechanism $\mathcal{M}$ provides an interface to each bidder i allowing them to purchase a *surrogate* s_i. Then, after all bidders have chosen their surrogates, $\mathcal{M}$ simply plugs $\vec{s}$ into $\mathcal{M}'$ and selects whatever outcome is selected by $\mathcal{M}'$. This procedure needs three desiderata:

- **Surrogate Stability**. Ultimately, all we know about the revenue of $\mathcal{M}'$ is that we're happy with $\text{Rev}(\mathcal{M}', D')$. Therefore, our reduction better result in surrogates from D' being input to $\mathcal{M}'$ – if the purchased surrogates come from some other distribution entirely, then we don't know anything about the ultimate revenue we get from $\mathcal{M}'$.
- **Incentive Compatibility**. The process by which bidders purchase surrogates must be Bayesian Incentive Compatible.
  - In particular, recall that *bidders have values for outcomes, not surrogates.* Therefore, the "value" of a bidder for a surrogate is the expected value for the outcome that surrogate receives (over the randomness in $\mathcal{M}'$ and over all other bidders drawing their values from D and purchasing their own surrogates).
  - Because we already must have Surrogate Stability, this amounts to: **IC(a)** ensuring Surrogate Stability, **IC(b)** computing the expected outcome that surrogate s achieves in mechanism $\mathcal{M}'$ assuming (a), and **IC(c)** designing an Incentive Compatible mechanism to sell surrogates to bidders assuming (a) and (b).
- **Small Subsidies**. Keeping in mind that surrogates in $\mathcal{M}'$ make payments, bidders may prefer to skip the surrogate purchase entirely (and have this option due to Individual Rationality). To address this, the designer may have to subsidize participation. If these subsidies are too large, it may completely offset the revenue earned by $\mathcal{M}'$.

To get intuition for each property, consider the following (flawed) mechanism: assign to each bidder i a random surrogate from D'_i. Consider also the following payment rules:

- We could give the bidder the surrogate for free, and just give them the outcome of $\mathcal{M}'$. But, this violates Incentive Compatibility (specifically, IC(c)) – perhaps the bidder has negative utility for the outcome of $\mathcal{M}'$, which includes a payment.
- Alternatively, we could refund the bidder the full payment made by the surrogate, and just assign the outcome of $\mathcal{M}'$. But, this violates Small Subsidies – we've now subsidized the bidders' participation with the full revenue earned by $\mathcal{M}'$.
- We could instead compute the maximum negative utility that any bidder in the support of D_i has for being represented by surrogate s, and subsidize the bidder exactly this amount. This may be computationally expensive – it takes an expectation over a product distribution with exponential support *and must be computed exactly!* If we're off by a tiny ε, we wind up

right back where we started with a ε-BIC mechanism. With only black-box access to $\mathcal{M}'$, it is moreover impossible to compute exactly. Moreover, even if we succeed with IC(b), this may still violate Small Subsides.

- We could instead compute the disutility of the bidder's reported type for being represented by surrogate s, and subsidize the bidder exactly this amount (if positive). This again may be computationally expensive to satisfy IC(b). Moreover, it likely violates IC(c) because the bidder can misreport their type to get a different subsidy. And, it may still violate Small Subsidies.

[HKM11] propose the following Replica-Surrogate Matching framework to sell surrogates.

- For each bidder i, draw a set S of surrogates.
- For each bidder i, draw *simulated competition* – a set R of *replicas*.
- Pick a *truthful mechanism* SurrogateMech for Bidder i to participate in against the simulated competition R, in order to purchase a surrogate.
- Design a procedure ComputeEdgeWeights to compute, for all surrogates s, the expected outcome that surrogate s achieves in mechanism $\mathcal{M}'$ assuming Surrogate Stability.

And, the following proof approach:[10]

- DrawSurrogates: ensure the process of drawing S such that a uniformly random element of S is identically distributed to a draw from D'.
- DrawReplicas: ensure process of drawing the simulated competition R, then concatenating t_i at the end is permutation-invariant.
- Observe that Surrogate Stability follows by DrawSurrogates and DrawReplicas.[11]
- Prove that ComputeEdgeWeights is correct, and analyze its sample/computational/query complexity.
- Prove that SurrogateMech is a truthful mechanism, assuming Surrogate Stability and that ComputeEdgeWeights is correct, and analyze its sample/computational/query complexity. This guarantees that the entire procedure is Bayesian Incentive Compatible, and together with the previous step concludes the sample/computational/query complexity analysis.
- Prove that SurrogateMech satisfies Small Subsidies. This is the only step that addresses the revenue loss.

[10]Below, we abuse notation and use the same name to refer both to the process and the fact that the process satisfies the desired (stated) properties.

[11]Proof sketch: By DrawReplicas, the process of drawing R and then t_i and seeing which surrogate t_i selects is identically distributed to first jointly drawing $R \cup \{t_i\}$ and then afterwards picking a uniformly random type to be the real t_i. This is a uniformly-random surrogate in S. By DrawSurrogates, this is identically distributed to a draw from D'_i.

3.2 Replica-Surrogate Matching: Detailed Overview of Prior Work

With the replica-surrogate matching framework in mind, we can now cleanly highlight the key innovations of prior work.

- [HKM11] first propose the framework, focused on approximately preserving *welfare* rather than revenue (and where Small Subsidies therefore captures lost welfare rather than lost revenue). Their implementation has DrawSurrogates and DrawReplicas simply draw surrogates and replicas i.i.d. from D' and D respectively, proposes for ComputeEdgeWeights to estimate expectations by sampling (and accept the sampling error in Bayesian Incentive Compatibility), and for SurrogateMech to be the VCG mechanism. After the (highly non-trivial) work to set up the [HKM11] framework, the only non-trivial step in the proof is Small Subsidies. Here, they use a HighCardinalityMatching argument: as long as we can match most replicas in R to a surrogate in S "within distance ε," then Small Subsidies holds.[12]
- [DW12, RW15] extend the framework to accommodate revenue loss instead of welfare loss. Every step is nearly-identical to [HKM11], except for: (a) a novel proof that the aforementioned HighCardinalityMatching suffices low revenue loss as well, and (b) identifying special cases where ComputeEdgeWeights can compute expectations exactly (and therefore the output mechanism is exactly BIC).
- [DHKN17, COVZ19] replace VCG with an online randomized matching mechanism, and develop novel Bernoulli factories to establish that their mechanism is *exactly* BIC even when ComputeEdgeWeights only provides sample access to the surrogate's outcome rather than a correctly-computed expectation. They also establish that HighCardinalityMatching suffices for Small Subsidies ([DHKN17] for welfare, and [COVZ19] for revenue) in their novel randomized matching mechanism.

In particular, all prior work simply has DrawSurrogates and DrawReplicas draw surrogates/replicas i.i.d., and leverages the same HighCardinalityMatching argument for Small Subsidies. Looking forwards, our meta-framework will formalize both the framework and this proof approach, and our implementation will propose a different DrawSurrogates and DrawReplicas, and also propose a novel SurrogateMech.

3.3 Key Technical Barriers

Before proposing our solution, we first discuss the obvious approaches and resulting technical barriers. First, we remind the reader that there is *zero tolerance for failure probability and sampling error*, and so it is crucial (for example) that Surrogate Stability holds *exactly with probability one.* Second, it is helpful to recall that the issue with simply applying [COVZ19] to product distributions is that it requires $\exp(m)$ samples from D and D' in order to satisfy HighCardinalityMatching.

Efficient surrogate sampling for product distributions. The obvious first proposal to explore is to replace DrawSurrogates with the following efficient procedure: instead of drawing

[12] Proof Sketch: one matching that might get selected by VCG is a maximal matching only considering edges where the replica and surrogate are within distance ε. This matching would preserve welfare up to an additive $n \cdot \varepsilon$, and the VCG matching selected will only get greater welfare.

surrogates i.i.d. from D'_i, draw *marginals for item* j i.i.d. from D'_{ij} and then create S by taking a product over all empirical marginal distributions. This indeed satisfies the necessary property: the process of drawing S, then selecting a uniformly random element of S, is indeed identically distributed to a draw from D'. By (now-)known concentration inequalities, poly$(m, 1/\varepsilon)$ samples from each marginal suffice to form an ε-net of the type space. This obvious idea works, and is part of our eventual solution.

Efficient replica sampling for product distributions. The obvious next step is to attempt the same adjustment for DrawReplicas to draw R. Unfortunately, this no longer works *because the joint distribution over* $(R, \{t_i\})$ *is no longer permutation-invariant.* Indeed, exactly one element of $R \cup \{t_i\}$ has marginals that were not used to form a product with all other marginals (t_i), and therefore this type (and only this type) must be t_i. Backtracking through the replica-surrogate proof outline, this means that the surrogate purchased by t_i under simulated competition is *not* a uniformly-random surrogate and therefore Surrogate Stability fails.

Surrogate Stability certainly *almost* holds with very high probability. This means that the impact on Small Subsidies is likely negligible and not a problem. However, the impact on *other bidders' incentives* is crucial: when Bidder i violates Surrogate Stability, the estimated expected outcome for any surrogate of Bidder $j \neq i$ is incorrectly computed, and therefore the mechanism is not exactly BIC for Bidder j. Again, while the error will be very small with very high probability, the error must be zero with probability one in order to be exactly BIC – the entire goal of this exercise is to eliminate small ε error in the incentive compatibility.

Alternate efficient replica sampling for product distributions. An alternate simple approach might be the following: draw many samples from each marginal D'_{ij} and call these the replica marginals R_{ij}. Then, add to each R_{ij} the value of bidder i for item j to get R'_{ij}. Then, form $R \cup \{t_i\}$ by taking all products of all marginals in R'_{ij}. This procedure is now permutation-invariant (because the marginals of t_i are treated exactly like the replica marginals. Therefore, Surrogate Stability is satisfied, and we can copy/paste the HighCardinalityMatching argument from earlier works. But, we've introduced a new problem: *Bidder i now controls some aspects of the simulated competition, and therefore* SurrogateMech *is no longer truthful!*

Again, Bidder i does not control *much* of the simulated competition (indeed, they only control a $1/\text{poly}(m, 1/\varepsilon)$-fraction of all item-values), but again the purpose of the entire exercise is to have zero incentive error with probability one, and so this also fails.

Despite their simplicity, the above examples capture well the conceptual challenge: (a) it is always hard to have *zero error with probability one* with only sample access, and more specifically (b) leveraging the replica-surrogate framework somehow requires replicas that are (i) completely indistinguishable from Bidder i (for Surrogate Stability) and yet (ii) that Bidder i has no control over whatsoever (for incentive compatibility of SurrogateMech), while (iii) leveraging the fact that D_i is a product distribution to create $\exp(m)$ replicas with poly$(n, m, 1/\varepsilon)$ samples from D_i.

3.4 Overview: Key Ideas

Finally, we overview the key aspect of our solution. As described in previous sections, we leverage much of the existing framework (or its natural extension to product distributions):

- We draw surrogates by first drawing $\text{poly}(n, m, 1/\varepsilon)$ times from each marginal, and then form S by taking all product distributions (natural extension of prior implementations).
- We leverage HighCardinalityMatching to ultimately guarantee Small Subsidies (which requires $\exp(m)$ replicas and surrogates).
- We use the replica-surrogate matching framework and proof approach (and, formalize this as a rigorous meta-framework).
- The missing steps are to define a new mechanism SurrogateMech, define a procedure to draw simulated competition, and to establish incentive compatibility.

We now overview our mechanism (jointly describing the simulated competition and the mechanism for surrogate sale):

- Draw $r = \text{poly}(n, m, 1/\varepsilon)$ *pricing-replicate-marginals* from each D_{ij}. Build the set R_p of *pricing-replicates* by taking the product of all pricing-replicate marginals.
- Compute the optimal dual prices for entropy-regularized matching *for the market of replicas* R_p (and surrogates drawn as described above).[13]
- Draw $r' = \text{poly}(n, m, 1/\varepsilon)$ *buying-replica-marginals* from each D_{ij}. Build the set R_b of *buying-replicates* by taking a product of all *buying-replicate-marginals and marginals of Bidder i.*
- Offer to every buying-replicate marginal (including Bidder i) the option to purchase any distribution over surrogates at the computed dual prices.
- If no surrogate is purchased with probability more than $1 + x$, where $x = 1/\text{poly}(n, m, 1/\varepsilon)$, set $\text{FLAG}_i = \texttt{True}$.
- For each bidder i, select a random surrogate to input to $\mathcal{M}'$ according to their purchased (random) surrogate with probability $1/(1+x)$, and a uniformly random surrogate with probability $x/(1+x)$.
- Finally, for each Bidder i, if $\text{FLAG}_j = \texttt{True}$ for all $j \neq i$, award Bidder i the outcome won by their (random) surrogate. If $\text{FLAG}_j = \texttt{False}$ for any $j \neq i$, award Bidder i nothing and charge them nothing.

Intuitively, something should seem very fishy about the plan above. For one, we just described in Section 3.3 how letting Bidder i influence their simulated competition violates Bayesian Incentive Compatibility, but then went right ahead and let Bidder i influence their competition anyway. For another, we also emphasized several times that we must have *zero error with probability one*, and the above FLAG_is certainly constitute some kind of failure that needs to be accounted for.

Here is the key idea, though (as a proof sketch – see Section C for a rigorous proof). First, it is indeed the case the Bidder i can influence their simulated competition, but *the simulated competition cannot impact the surrogates offered to Bidder i.* Indeed, *only the pricing-replicates influence Bidder i's surrogate menu and Bidder i has no influence over these.* Moreover, *while Bidder i's buying-replicates have influence over* FLAG_i*,* FLAG_i *has no influence over whether Bidder*

[13] We postpone a description of entropy-regularized matching until Section 4.2 as it is substantial and requires several parameters to be set correctly. Moreover, this particular choice is not a key innovation and comes from [DHKN17, COVZ19]. The only important aspect for this outline is that *Bidder i does not influence the prices at all.*

i keeps their surrogate's outcome. Therefore, assuming Surrogate Stability and the correctness of ComputeEdgeWeights, the entire procedure is indeed exactly BIC. Intuitively, the key contribution from separating replicas into pricing-replicas and buying-replicas is so that the ultimate surrogate purchased by Bidder i is identically distributed to that purchased by a buying-replica (which necessarily requires dependence on Bidder i), while maintaining that the prices are set only by pricing-replicas (who are not influenced at all by Bidder i).

Therefore, it remains to argue for Surrogate Stability (what follows is a proof-sketch – see Section C for a rigorous proof). It is indeed the case that FLAG_i might be `False`, because unlikely events happen with non-zero probability when relying on samples. And, when FLAG_i is `False`, we cannot argue Surrogate Stability for Bidder i, because some surrogates are overdemanded. But, *importantly, when* FLAG_i *is* `False`*, no other bidder j gets any outcome no matter what.* Therefore, *the outcome Bidder j receives from surrogate s is conditioned on no* FLAG*s being* `False` *for any other bidders, and conditioned on no* FLAG*s being* `False` *for any other bidders, Surrogate Stability holds for all other bidders!*[14] Intuitively, the key contribution of the FLAGs is to turn a low-probability failure into an additional revenue problem (by throwing out every other bidders' outcome and payment) rather than an incentive problem (if instead we had let other bidders receive outcomes with an improper Bidder i surrogate distribution).

This concludes our technical outline. Section A provides complete proofs of the meta-framework, which essentially formalizes the above approach. Section C provides rigorous proofs of the above proof sketch.

4 Additional Technical Preliminaries

4.1 Replica-Surrogate Matching: Notation

For bidder $k \in [n]$, we refer to the replicas as a multiset $\mathcal{R}_k$, with $R := |\mathcal{R}_k|$, and surrogates as a multiset $\mathcal{S}_k$, with $S := |\mathcal{S}_k|$. In general, the number R of replicas can be any multiple of S, in which case κ-to-1 matchings are considered where $\kappa := \frac{R}{S} \in \mathbb{N}$ (as in [DHKN17, COVZ19] and our algorithm). We consider the replicas to be enumerated in some fixed order $\mathcal{R}_k = (r_1, \dots, r_R)$, and then we will often identify a replica r_i with its index $i \in [R]$. Similarly, we write surrogates as $\mathcal{S}_k = (s_1, \dots, s_S)$ and often identify a surrogate s_j with its index $j \in [S]$.

We will also need to consider edge weights defined with respect to particular distributions other than D', captured formally by the next definition.

Definition 4.1 (Edge weights $\hat{w}_{ij}$ with respect to mechanism $\hat{\mathcal{M}}'$ and $\hat{D}'$)**.** *For a particular bidder k, the edge weight* $\hat{w}(t_a, t_b)$ *from type t_a to t_b, with respect to a mechanism* $\hat{\mathcal{M}}' = (\mathcal{A}', \hat{p}')$ *and a distribution* $\hat{D}' := \times_{k' \in [n]} \hat{D}'_{k'}$, is defined to be the interim utility of t_a for t_b under $\hat{\mathcal{M}}'$ and $\hat{D}'$ with *discount factor* β_k:

$$\hat{w}(t_a, t_b) = \mathop{\mathbb{E}}_{t_{\text{-}k} \sim \hat{D}'_{\text{-}k}} [v_k(t_a, \mathcal{A}'(t_b; t_{\text{-}k})) - (1 - \beta_k) \hat{p}'_k(t_b; t_{-k})],$$

[14] Note that the claim "Surrogate Stability holds conditioned on no FLAGs being `False` for any other bidders" should not feel immediately obvious, but it follows by a similar coupling argument establishing that, under this conditioning, the purchased surrogate is indeed uniformly random (see Section C for a rigorous statement and proof).

where $\hat{D}'_{\text{-}k} = \times_{k' \neq k} \hat{D}'_{k'}$. We will often abbreviate the edge weight $\hat{w}(r_i, s_j)$ *between a replica* r_i *and surrogate* s_j to $\hat{w}_{ij} := \hat{w}(r_i, s_j)$.

4.2 Entropy-Regularized Matching: Background

We turn to a technique introduced by [DHKN17, COVZ19] for executing replica-surrogate matching in the sample access model. The issue with the base approach described in prior sections is that the maximum-weight matching cannot actually be computed given only sample access to D' and query access to $\mathcal{M}'$, because we cannot compute the expectations in the expression for edge weights $w(r_i, s_j)$.[15] In particular, as [COVZ19] discuss, it does not suffice to estimate the expectations in the edge weights by sampling, as even arbitrarily small approximation error derails the delicate argument for exact BIC.

To handle this challenge, [DHKN17] introduce an *entropy regularization* term to the max-weight (many-to-one) matching. The benefit of this entropy-regularized version is that its optimal solution, unlike the exact max-weight matching, can be sampled from given only sample access to edge weight distributions, via a new connection to *Bernoulli factories* [KO94], namely, the *Fast Exponential Bernoulli Race* [DHKN17]. This is the key innovation that enables [DHKN17] and [COVZ19] to obtain black-box reduction results in the sample access model. Meanwhile, it can be shown that using the max-weight entropy-regularized matching still permits an incentive compatibility argument to go through.

We present the technical details of entropy-regularized matching that are relevant to our approach. The definitions in this section will all be in the context of a particular bidder $k \in [n]$. Thus we refer to replicas and surrogates here simply as $\mathcal{R} := \mathcal{R}_k$ and $\mathcal{S} := \mathcal{S}_k$, respectively, and let $R := |\mathcal{R}|$, $S := |\mathcal{S}|$, and $\kappa := \frac{R}{S} \in \mathbb{N}$. We start by defining the program that encodes a max-weight, many-to-one matching.

Definition 4.2 (Max-weight κ-to-1 matching)**.** For replicas $\mathcal{R}$, surrogates $\mathcal{S}$, and edge weights $w = (w_{ij})_{i,j}$ between each replica $r_i \in \mathcal{R}$ and surrogate $s_j \in \mathcal{S}$, the *max-weight* (integral) *κ-to-1 matching* is given by the following program:

$$\begin{aligned}
\max_{(x_{ij})_{i,j}} \quad & \sum_{i \in [R], j \in [S]} x_{ij} w_{ij} \\
\text{s.t.} \quad & \sum_i x_{ij} \leq \kappa \qquad \forall j \in [S], \\
& \sum_j x_{ij} \leq 1 \qquad \forall i \in [R], \\
& x_{ij} \in \{0, 1\}, \qquad \forall i \in [R], j \in [S].
\end{aligned}$$

We will refer to this program as $\mathbf{P}(\mathcal{R}, \mathcal{S}, w)$.

We now define the entropy-regularized matching program from [DHKN17, COVZ19]. In Theorem 4.3 below, we directly state the (partial) *Lagrangian relaxation* of the entropy-regularized matching program where surrogate demand constraints are relaxed, as this is the form that we will make use of. Moreover, we state a version of entropy-regularized matching that includes *dummy*

[15] Recall that there is an expectation over both $t_{\text{-}k} \sim D'_{\text{-}k}$ *and* the randomness of the mechanism $\mathcal{M}'$ (implicit in our notation).

matches as introduced in [COVZ19]: intuitively, the added y_{ij} variables capture an option to match to a dummy version of a surrogate s_j with zero edge weight (while contributing to the entropy term and jth demand constraint).

Definition 4.3 (Entropy-regularized matching [DHKN17, COVZ19])**.** Consider replicas $\mathcal{R}$; surrogates $\mathcal{S}$; edge weights $w = (w_{ij})_{i,j}$ between each replica $r_i \in \mathcal{R}$ and surrogate $s_j \in \mathcal{S}$; and regularization parameter δ. For fixed dual variables $\alpha = (\alpha_j)_{j\in[S]}$, the *Lagrangian relaxation of the entropy regularized matching program*, denoted by $\mathbf{P}'_\delta(\mathcal{R}, \mathcal{S}, w, \alpha)$, is given by the following:

$$
\begin{aligned}
\max_{(x_{ij}, y_{ij})_{i,j}} \quad & \sum_{i\in[R], j\in[S]} x_{ij} \cdot w_{ij} - \delta \sum_{i,j} (x_{ij} \log x_{ij} + y_{ij} \log y_{ij}) + \sum_j \alpha_j \left(\kappa - \sum_i (x_{ij} + y_{ij}) \right) \\
s.t. \quad & \sum_j (x_{ij} + y_{ij}) = 1 \qquad \forall i \in [R], \\
& x_{ij}, y_{ij} \in [0,1], \qquad \forall i \in [R], j \in [S].
\end{aligned}
$$

The optimizer of program $\mathbf{P}'_\delta(\mathcal{R}, \mathcal{S}, w, \alpha)$ is a softmax function of the following form.

Definition 4.4 (Softmax allocation)**.** Given parameter δ, weights $w = (w_j)_{j\in[S]}$, and dual variables $\alpha = (\alpha_j)_{j\in[S]}$ over index set $[S]$, define the *softmax allocation* $(x^*, y^*) = (x^*_j, y^*_j)_{j\in[S]}$ by

$$
x^*_j(w, \alpha) := \frac{\exp(\frac{w_j - \alpha_j}{\delta})}{\sum_{j'\in[S]} \exp\left(\frac{-\alpha_{j'}}{\delta}\right)(1 + \exp(w_{j'}/\delta))} \text{ and } y^*_j(w, \alpha) := \frac{\exp(\frac{-\alpha_j}{\delta})}{\sum_{j'\in[S]} \exp\left(\frac{-\alpha_{j'}}{\delta}\right)(1 + \exp(w_{j'}/\delta))}.
$$

Fact 4.5. For any dual variables $\alpha = (\alpha_j)_j$, the optimizer $(x^*_{ij}(\alpha), y^*_{ij}(\alpha))_{i,j}$ of program $\mathbf{P}'_\delta(\mathcal{R}, \mathcal{S}, w, \alpha)$ for edge weights $w := (w_{ij})_{i,j}$ over $r_i \in \mathcal{R}, s_j \in \mathcal{S}$ is given by $x^*_{ij}(\alpha) := x^*_j(w_i, \alpha)$ and $y^*_{ij}(\alpha) := y^*_j(w_i, \alpha)$ defined according to Theorem 4.4 with $w_i := (w_{ij})_{j\in[S]}$.

Optimal dual variables. For the *optimal* duals $\alpha^* \in \arg\min_\alpha \mathbf{P}'_\delta(\mathcal{R}, \mathcal{S}, w, \alpha)$, it holds for all $j \in [S]$ that $\sum_i \left(x^*_{ij}(\alpha^*) + y^*_{ij}(\alpha^*) \right) = \kappa$. This is seen as follows: since α^* are the optimal duals, we must have that for each j, $\sum_i (x^*_{ij}(\alpha^*) + y^*_{ij}(\alpha^*)) \le \frac{R}{S}$, i.e. no surrogate is overdemanded. But $\sum_j (x^*_{ij}(\alpha^*) + y^*_{ij}(\alpha^*)) = 1$ for each i, so $\sum_i (x^*_{ij}(\alpha^*) + y^*_{ij}(\alpha^*))$ must be exactly $\frac{R}{S}$ for all j.

In a similar vein to Theorem 4.5, we can also observe that the softmax allocation is the optimizer of the following optimization problem specific to a particular replica r_i, viewed as an abstract type $t \in \mathcal{T}_k$ below.

Fact 4.6. Consider surrogates $\mathcal{S}$, duals α, and a type $t \in \mathcal{T}_k$ with edge weights $w_t := (w_{tj})_j$ between t and each surrogate $s_j \in \mathcal{S}$. Then for $x^*_{tj}(\alpha) := x^*_j(w_t, \alpha)$ and $y^*_{tj}(\alpha) := y^*_j(w_t, \alpha)$ as defined in Theorem 4.4, it holds that

$$
(x^*_{tj}(\alpha), y^*_{tj}(\alpha))_j \in \arg\max_{(x_j, y_j)_j} \sum_{j\in[S]} x_j w_{tj} - \delta \sum_j (x_j \log x_j + y_j \log y_j) - \sum_j \alpha_j \cdot (x_j + y_j)
$$

where the optimization is subject to the constraint $\sum_{j\in[S]} (x_j + y_j) = 1$.

Notation. As in Theorem 4.5 and Theorem 4.6, for convenience we will abuse notation to write both $w_{ij} := w(r_i, s_j)$ for a (replica) type r_i and $w_{tj} := w(t, s_j)$ for a type t to denote edge weights depending on the situation, where the weight function $w(\cdot,\cdot)$ in question (usually the one defined in Theorem 4.1) will be clear from context. As an extension of this notation, we will write both x^*_{tj} and x^*_{ij}; y^*_{tj} and y^*_{ij}; and similarly for analogous terms in the future. We will always index replicas with i and surrogates with j so that it is clear which notation is being used.

Finally, we state the Fast Exponential Bernoulli Race, the randomized algorithm introduced by [DHKN17] that draws a sample from the softmax allocation distribution given only sample access to the distributions whose expectations are the edge weights $(w_j)_j$. The version stated below is from Lemma 1 in [COVZ19].

Lemma 4.7 (Fast Exponential Bernoulli Race [DHKN17,COVZ19])**.** *For any integer N, any $\delta > 0$, and any $(\alpha_p)_{p\in[N]} \in [0, A]^N$, given sample access to distributions $\mathcal{F}_1, \ldots, \mathcal{F}_N$ with expectations $w_1, \ldots, w_N \in [-1, 1]$, a sample from the distribution $(\zeta_p)_{p\in[N]}$ defined by*

$$\zeta_p = \frac{\exp((w_p - \alpha_p)/\delta)}{\sum_{p'\in[N]} \exp((w_{p'} - \alpha_{p'})/\delta)}$$

can be drawn with $\left(\frac{4+A}{\delta}\right)^4 N^2 \log\left(\frac{(4+A)N}{\delta}\right)$ samples from $(\mathcal{F}_j)_{j\in[n]}$ in expectation.

In particular, observe that taking $N = 2S$ and considering $w_p = 0$ for S of the N distributions in Theorem 4.7 allows us to recover the form of softmax allocations in Theorem 4.4.

5 Meta-Framework

We now introduce a *meta-framework* that provides a generalized framework for black-box reductions in Bayesian mechanism design. The goal of our meta-framework is to isolate the core structure of replica-surrogate matching necessary for the incentive and revenue guarantees desired in ε-BIC-to-BIC transformations,[16] while abstracting out certain subroutines that can be implemented to obtain different or improved downstream results. In particular, the meta-framework reduces the design of ε-BIC-to-BIC transformations to the design of these specific subroutines (which we call "meta-inputs"). Our main result of polynomial sample complexity will be proved in Section 6 through implementations of these subroutines. We hope that future work on such transformations can focus on designing versions of these abstract subroutines and then plug in to our meta-framework to obtain new results, rather than repeating (sometimes onerous) arguments inherent to replica-surrogate matching.

We describe the meta-inputs in Section 5.1 and then present the main algorithms that define the meta-framework in Section 5.2. We formally state the requirements each meta-input must satisfy for the meta-framework to yield a revenue-preserving ε-BIC-to-BIC transformation in Section 5.3, with the resulting guarantee stated in Section 5.4.

5.1 Meta-Inputs

The meta-framework takes as input six abstract subroutines that capture various components of (a generalized version of) entropy-regularized replica-surrogate matching: Draw-$\mathcal{S}$, Draw-$\mathcal{R}$, Compute-α, Match, Payment, and Preprocess.

[16] Or other objectives than revenue, e.g. welfare, in the literature on black-box reductions in mechanism design.

Each of these meta-inputs is defined in the context of an individual bidder k.[17] We start by giving high-level descriptions of each meta-input's intended input/output behavior below, with detailed requirements in Section 5.3. When we say that a meta-input takes as input D'_k or D_k, we mean that it has *sample access* to that distribution. Similarly, if a procedure takes as input $\mathcal{M}'$, we mean that it has *query access* to $\mathcal{M}'$.

$\textsc{Draw-}\mathcal{S}$ (Input: $D'_k \to$ Output: $\mathcal{S}_k$): The $\textsc{Draw-}\mathcal{S}$ meta-input draws the set of surrogates $\mathcal{S}_k$ to be used in replica-surrogate matching, given sample access to D'_k.

$\textsc{EstMechanism}$ (Input: $\mathcal{M}'$, $(\mathcal{S}_k)_k) \to$ Output: $\hat{\mathcal{M}}'$, $\textsc{Flag}_{\textsc{Est}}$): The $\textsc{EstMechanism}$ meta-input takes as input a mechanism $\mathcal{M}'$ to which it has query access and the sets of sampled surrogates and outputs a mechanism for the uniform distribution over $\times_k \mathcal{S}_k$. The procedure is permitted to raise a flag $\textsc{Flag}_{\textsc{Est}}$ indicate a failure event where the output does not satisfy some desired property.

$\textsc{Draw-}\mathcal{R}$ (Input: D_k, $t_k \to$ Output: $\mathcal{R}_k$, i^*, $\textsc{Flag}_{\mathcal{R}}$): The $\textsc{Draw-}\mathcal{R}$ meta-input draws the set of replicas $\mathcal{R}_k$, given both sample access to D_k and bidder k's reported type t_k. The replicas $\mathcal{R}_k$ must include the bidder's reported type t_k, where the returned value i^* indicates the index of t_k in $\mathcal{R}_k$. The procedure is permitted to raise a flag, indicated by boolean $\textsc{Flag}_{\mathcal{R}}$, to indicate a failure event where the replicas need not satisfy desired conditions.[18]

$\textsc{Compute-}\alpha$ (Input: $(\mathcal{S}_{k'})_{k' \in [n]}$, D_k, $\mathcal{M}' \to$ Output: $\hat{\alpha} = (\hat{\alpha}_j)_{j \in [S]}$): The $\textsc{Compute-}\alpha$ meta-input returns an approximately-optimal dual solution for max-weight entropy-regularized matching between replicas $\mathcal{R}_k$ and surrogates $\mathcal{S}_k$ (Theorem 4.3). Note that the inputs to $\textsc{Compute-}\alpha$ include the surrogates $\mathcal{S}_k$ but *not* the replicas $\mathcal{R}_k$.[19]

$\textsc{Match}$ (Input: r_i, $\hat{\alpha}$, $(\mathcal{S}_{k'})_{k' \in [n]} \to$ Output: s_j or $\circ_j$ for some $j \in [S]$): The $\textsc{Match}$ meta-input samples a surrogate match for the given replica r_i: for all $j \in [S]$, it returns $s_j \in \mathcal{S}_k$ with probability exactly $x^*_{ij}(\hat{\alpha})$ or a corresponding *dummy surrogate* $\circ_j$ with probability $y^*_{ij}(\hat{\alpha})$, for softmax allocation $(x^*_{ij}(\hat{\alpha}), y^*_{ij}(\hat{\alpha}))_j$.

$\textsc{Payment}$ (Input: t_k, $\hat{\alpha}$, $(\mathcal{S}_{k'})_{k' \in [n]} \to$ Output: q_k): The $\textsc{Payment}$ meta-input specifies the payment rule for the surrogate selection procedure: as a function of the duals $\hat{\alpha}$, it outputs a payment q_k to charge a given type t_k to be matched to a surrogate according to $(x^*_{ij}(\hat{\alpha}), y^*_{ij}(\hat{\alpha}))_j$, such that the desired incentive guarantees hold.

$\textsc{Preprocess}$ (Input: $D_k \to$ Output: $\mathcal{G}$): The $\textsc{Preprocess}$ meta-input is an optional subroutine that, given sample access to D_k, permits pre-processing D_k to compute some global state $\mathcal{G}$ at the start of surrogate selection that can be used by subsequent meta-inputs. In particular, $\mathcal{G}$ is an implicit input to each of the subsequent meta-inputs in surrogate selection.[20]

[17] It is assumed that the implementations of these procedures use *independent randomness* across the n bidders.

[18] Allowing and handling a failure event in $\textsc{Draw-}\mathcal{R}$ is necessary due to technical details of our implementation in Section 6.

[19] In particular, $\textsc{Compute-}\alpha$ cannot take as input $\mathcal{R}_k$ (nor the bidder's type t_k) yet the requirements it must satisfy, as stated in Section 5.3, depend directly on $\mathcal{R}_k$. This pinpoints part of the challenge in implementing $\textsc{Compute-}\alpha$ (and a corresponding implementation of $\textsc{Draw-}\mathcal{R}$).

[20] We will need such a preprocessing function in our implementation in Section 6 for technical reasons, to estimate an *effective support* of a discretized version of D_k.

5.2 Main Algorithms

We use the meta-inputs to define our core surrogate selection procedure (the MetaRS algorithm, Algorithm 1), and then the overall mechanism $\mathcal{M}$, presented in Algorithm 2. The primary innovations of these algorithms, relative to existing (entropy-regularized) replica-surrogate matching, can be summarized as follows: (1) decoupling surrogate price-setting from bidder report; (2) randomly dropping matches to scale down surrogate (over)demand; (3) correlated discarding of allocations across bidders under failure events; and (4) computing with respect to *empirical* versions of surrogate distributions.

Intuition. To motivate these innovations, we first recall at a high level the implementation changes we will make to achieve polynomial sample complexity in the independent items setting (the focus of Section 6), which we want the meta-framework to support. In order to obtain the exponentially-many replicas and surrogates needed for revenue-approximation guarantees, we will sample from each of the marginals $D'_{k\ell}$ of bidder k's distribution $D'_k := \times_{\ell \in [m]} D'_{k\ell}$ and take the product set over $[m]$ as the surrogates $\mathcal{S}_k$, and do the same (roughly speaking) for D_k to obtain replicas $\mathcal{R}_k$. We include the bidder's reported type t_k when taking the product, which means the marginals of t_k are part of many of the final replicas $\mathcal{R}_k$. However, this approach leads to an incentives problem: the bidder may now find it in her interest to misreport as a means to manipulate the aggregate demand (via the replicas she influences) and thereby the (market-clearing) prices for surrogates.

This challenge brings us to point (1). Concretely, what this means for entropy-regularized matching (for each bidder k) is that we must somehow decouple determining the surrogate prices, as characterized by dual variables α, from any quantities dependent on the bidder's type t_k. This functionality is encoded via the Compute-α meta-input and its requirements: it must compute (offline, unlike [DHKN17, COVZ19]) an estimate $\hat{\alpha}$ of the optimal duals for the entropy-regularized matching between $\mathcal{R}_k$ and $\mathcal{S}_k$—*without access to* $\mathcal{R}_k$, so that the bidder's report has no influence on this process—such that, with high probability, the market for surrogates approximately clears when replicas r_i are matched according to $(x^*_{ij}(\hat{\alpha}), y^*_{ij}(\hat{\alpha}))$. We will see one implementation of Compute-α that satisfies this (a priori, rather challenging) specification in Section 6, by learning duals on a set of "training replicas."

Still, there are two qualifiers above: the surrogate market for each bidder k clears only *approximately*, and only *with high probability*. Both of these qualifications are problematic for the standard argument for exact BIC in replica-surrogate matching. We handle the fact that the market clears only approximately (i.e., surrogates may be slightly over-demanded based on the returned $\hat{\alpha}$) with point (2). Specifically, for parameter λ such that Compute-α ensures the realized demand for each surrogate will be at most $(1 + \lambda)$ times its capacity $\kappa := \frac{R}{S}$, we un-match replicas with probability λ. Any unmatched replica is then matched to a *dummy surrogate* $\circ_j$ corresponding to a surrogate s_j that is under capacity (where $\circ_j$ contributes to j's demand but indicates the fact that the bidder will not actually receive an allocation in the end, which is possible since the outcome space is downward-closed). Note that the bidder as a replica has no influence whether it will be dropped in this process, so there are no new incentive problems.

We handle low probability failures with point (3): under a failure event for bidder k, we discard the allocations of *all other bidders*. This is perhaps counterintuitive at first blush. The key insight is that when bidder k's surrogate market does not clear, stationarity is violated for bidder k, which invalidates the edge weight expressions in the markets of *all other bidders* $k' \neq k \in [n]$. However, if we always give any other bidder k' a null allocation $\perp$ (and charge 0) under this failure event for k, then k' does not care that bidder k's surrogate's distribution looks incorrect, and importantly, bidder k' cannot influence when this discarding event due to k takes place. Meanwhile, there is no need to discard bidder k's allocation if only her market fails to clear (and doing so may in fact create an incentive for bidder k to misreport). We will formalize this reasoning in the proof that the resulting mechanism is BIC. In our MetaRS algorithm, the failure event for a bidder is encoded through the boolean Flag.[21]

The final point (4) is required to preserve efficient sample complexity. Recall that our implementation in the independent items setting will consider an exponential number of replicas and surrogates (despite constructing them differently). To implement Match (without knowing the edge weights), we will need to run the Fast Exponential Bernoulli Race (FEBR) (Theorem 4.7) for each replica, but notice that for matching even a single replica this requires poly(S) samples from edge weight distributions, which is exponential in m. Our solution is to instead define edge weights (Theorem 4.1) with respect to the *empirical distributions* $(\hat{D}'_{k'})_{k'\in[n]}$, defined as uniform distributions over the realized surrogate draws $\mathcal{S}_{k'}$ for all $k' \in [n]$, rather than $(D'_{k'})_{k'\in[n]}$. Stationarity holds with respect to $\hat{D}'_k$ in the same way as for D'_k in prior work, and the result is that the sampling in the FEBR now incurs no new samples from D'_k.[22]

Algorithm descriptions. We formally present our surrogate selection procedure as the MetaRS algorithm (Algorithm 1), stated in the context of an individual bidder k since it is run separately for each bidder. The overall mechanism $\mathcal{M}$, which uses MetaRS, is given in Algorithm 2.

Algorithm 1 proceeds as follows. First, note that surrogate sets $(\mathcal{S}_1, \dots, \mathcal{S}_n)$ for all bidders and the output of EstMechanism are passed as input to MetaRS, after being drawn and computed externally in Algorithm 2. This is required for edge weights and resulting matching probabilities for bidder k to be set with respect to the product of empirical distributions $\hat{D}'_{\text{-}k} := \times_{k'\neq k}\hat{D}'_{k'}$ over the surrogate sets. The first step of the algorithm is to run Preprocess to obtain global state $\mathcal{G}$, and create the dummy surrogates $\{\circ_j\}_{j\in[S]}$. In addition to our new un-matching process, dummy surrogates $\circ_j$ (and the corresponding allocation probabilities y^*_{ij}) are used for the same reason they are introduced in [COVZ19]: as an alternate zero-weight option to avoid matching along negative-weight edges (see Section 5.3 for more discussion). We then run Compute-α to obtain approximately optimal duals $\hat{\alpha}$, and run Draw-$\mathcal{R}$ to obtain the replicas $\mathcal{R}_k$ and the auxiliary information i^* and Flag$_\mathcal{R}$. The next step is to match replicas to surrogates using Match, dropping a λ fraction (in expectation) of matches by flipping a λ-coin for each replica, where λ is a parameter of the meta-framework (Section 5.3.1). At this point we can mark whether to raise a failure event in bidder k's surrogate selection: if either Flag$_\mathcal{R}$ = `False`, i.e. Draw-$\mathcal{R}$ raised a failure event, or any surrogate is over-demanded even after the $(1-\lambda)$ down-scaling, then we raise by setting Flag = `False`. We then wrap up by matching un-matched replicas to dummy surrogates $\circ_j$ whose indices j are under-capacity, such that, if Flag = `True`, the final matching is a perfect κ-to-1

[21] Throughout the paper, our algorithms will use different types of Flag variables, each of which will be a boolean that indicates some kind of failure event, where a `False` Flag denotes the failure case.

[22] The careful reader may wonder why we need the FEBR at all given that we now fully know the distributions $\hat{D}'_k$ and hence the distribution $\hat{D}'_{\text{-}k}$ in Theorem 4.1. The answer is that we still have only query access to the mechanism $\mathcal{M}' = (\mathcal{A}', p')$, and there are expectations with respect to the randomness of $\mathcal{A}'$ and p' that we cannot compute in Theorem 4.1. As an additional benefit, while computation is not the focus of our work, using the FEBR saves us from computation that is exponential in the *number of bidders* that would otherwise be required even with a fully-known auction.

matching from replica indices $[R]$ to surrogate indices $[S]$.[23] The algorithm returns the surrogate matched to the bidder replica, plus auxiliary information to be used in Algorithm 2.

Algorithm 2 is similar in structure to the overall mechanism in prior replica-surrogate matching work, with two deviations. The first, as mentioned, is that DRAW-$\mathcal{S}$ is run for each bidder at this level, not in the surrogate selection procedure, such that all surrogate sets are passed as input when invoking METARS for each bidder k. The second is the correlated discarding when specifying final allocations and payments. For each bidder k, we set the final allocation to be $\perp$ and total payment to be 0 under the following two events: (1) failure event for some other bidder k', i.e. $\text{FLAG}^{(k')} = \texttt{False}$ for any $k' \neq k$, or (2) bidder k's match was dropped in METARS, i.e. $\text{COIN}^{(k)} = 0$. Otherwise, we charge payment for participating in MATCH using PAYMENT, even if the bidder was matched to a dummy surrogate by MATCH. Only if the bidder was matched to a real surrogate does she receive the resulting allocation o_k and additionally pay $(1 - \beta_k)p'_k$ from running $\mathcal{M}'$ on the profile of surrogates over all bidders, where β_k is a discount factor that is a parameter of the meta-framework.

5.3 Meta-Input Requirements

5.3.1 Parameters.

The meta-framework requirements are stated in terms of the following set of parameters, such that the quantitative results (in particular, the revenue approximation guarantee) of the meta-framework will be a function of these parameters. We give a brief description of each parameter below, and indicate in parentheses the formal context in which it is used, where Conditions refer to the upcoming formal statements of requirements.[24]

- $(\Delta^{(k)}_{\text{COUPLE}})_{k\in[n]}$: Coupling term bound (Condition (B4)).
- S, R: Number of surrogates and replicas, respectively, where it is required that R is a positive multiple of S (Condition (A), (B)).
- λ: Replica-surrogate over-demand allowance (Condition (C1), (C2)).
- δ: Regularization parameter (for program $\mathbf{P}'_\delta(\cdot,\cdot,\cdot,\cdot)$, Theorem 4.3)
- $(\beta_k)_{k\in[n]}$: Payment discount factor for rule p' of $\mathcal{M}'$ (see Algorithm 2).
- d: Maximum valuation distance between two *close* types (Condition (B3)).
- q_{LB}: Lower bound on expected payment q_k for Phase 1 (Condition (E3))
- Δ_{MATCH}: Expected fraction unmatched in high-cardinality matching (Condition (B3))
- Δ_{MWM}: Expected weight difference relative to maximum-weight matching (Condition (C2)).

[23] In terms of demand, the dummy surrogate $\circ_j$ and real surrogate s_j act as a single group, because if a bidder is matched to $\circ_j$ in METARS, then the corresponding s_j will be inputted on its behalf into $\mathcal{M}'$ in Algorithm 2.

[24] As the notation suggests, the values of $\Delta^{(k)}_{\text{COUPLE}}$ and β_k may vary over the bidders $k \in [n]$, as we expect them to depend intrinsically on each distribution D_k/D'_k (as in our implementation). All other parameters are fixed over the bidders.

Algorithm 1: MetaRS for bidder k

Inputs:

1. Sample access to D_k and (query)[a] access to $\hat{\mathcal{M}}'$.
2. Bidder k's reported type t_k.
3. Surrogate sets $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$.
4. Access to random $\textsc{Coin} = \textsc{Bern}(1-\lambda)$ for parameter λ.

Algorithm:

0. Preprocess: obtain state $\mathcal{G} \leftarrow \textsc{Preprocess}(D_k)$ accessible in all future steps.[b]
1. Let $S := |\mathcal{S}_k|$. Create a dummy surrogate $\circ_j$ corresponding to each $s_j \in \mathcal{S}_k$.
2. Set $\hat{\alpha} \leftarrow \textsc{Compute-}\alpha((\mathcal{S}_{k'})_{k' \in [n]}, D_k, \hat{\mathcal{M}}')$.
3. Let t_k be the bidder's type. Set replicas $\mathcal{R}_k$, index i^*, $\textsc{Flag}_{\mathcal{R}} \leftarrow \textsc{Draw-}\mathcal{R}(D_k, t_k)$.
4. For $r_i \in \mathcal{R}_k$ ($i = 1 \to R$, where $R := |\mathcal{R}_k|$):
   (a) Flip $\textsc{Coin}_i := \textsc{Bern}(1-\lambda)$.
   (b) If $\textsc{Coin}_i = 1$, match r_i to surrogates or dummies via $\textsc{Match}(r_i, \hat{\alpha}, (\mathcal{S}_{k'})_{k' \in [n]})$.
   (c) Else (if $\textsc{Coin}_i = 0$) add r_i to set $\textsc{Dummy}(\mathcal{R}_k)$. `// i.e. leave unmatched for now`
5. Set $\textsc{Flag} = \textsc{Flag}_{\mathcal{R}}$ && $(\forall j \in [S], |\{r_i \in \mathcal{R}_k : r_i \text{ matched to } s_j \text{ or } \circ_j\}| \leq \frac{R}{S})$.
6. For $r_i \in \textsc{Dummy}(\mathcal{R}_k)$:
   (a) Match r_i to $\circ_j$ for the smallest j s.t. $|\{r_i \in \mathcal{R}_k : r_i \text{ matched to } s_j \text{ or } \circ_j\}| < \frac{R}{S}$.
7. Let $j^* \in [S]$ be the *surrogate index* matched to r_{i^*}.[c]
   Return $\textsc{Flag}$, $\textsc{Coin}_{i^*}$, s_{j^*}, $\hat{\alpha}$, and `dummy` = `True` if r_{i^*} matched to $\circ_{j^*}$, else `False`.

[a] What kind of access one has to $\hat{\mathcal{M}}'$ depends on the output of EstMechanism.
[b] In particular, $\mathcal{G}$ is an implicit parameter for any meta-input called in MetaRS.
[c] That is, j^* is such that r_{i^*} is either matched to s_{j^*} or to $\circ_{j^*}$.

- $\Delta_{\text{D-S}}, \delta_{\text{D-S}}, \varepsilon_{\text{D-S}}$: Empirical distribution closeness parameters: revenue loss $\Delta_{\text{D-S}}$, incentive loss $\varepsilon_{\text{D-S}}$, failure probability $\delta_{\text{D-S}}$ (Condition (A1))
- $\delta_{\textsc{Demand}}$: Failure probability for replica-surrogate equal-demand (Condition (C1)).
- $\delta_{\text{D-R}}$: Failure probability for $\textsc{Draw-}\mathcal{R}$ (Condition (B1)).

5.3.2 Requirements

We specify in Conditions (A) through (E) below the requirements that must be satisfied for the mechanism $\mathcal{M}$ defined by Algorithm 2 to be BIC and IR and achieve revenue over D approximating that of $\mathcal{M}'$ over D'. The requirements are stated with respect to each individual bidder $k \in [n]$, except where otherwise specified. We first explain some of the less transparent requirements.

Algorithm 2: Full mechanism $\mathcal{M}$ over n bidders

Inputs:

1. Sample access to $D' = \times_k D'_k$ and $D = \times_k D_k$.
2. Reported type t_k of bidder k, for each $k \in [m]$.
3. Query access to input mechanism $\mathcal{M}' = (\mathcal{A}', p')$.

Algorithm:

Phase 1:

1. For each bidder $k \in [n]$, let $\mathcal{S}_k \leftarrow \text{Draw-}\mathcal{S}(D'_k)$.
2. Let $\hat{\mathcal{M}}' = (\hat{\mathcal{A}}', \hat{p}'), \text{Flag}_{\text{Est}} \leftarrow \text{EstMechanism}(\mathcal{M}', \mathcal{S}_1, \dots, \mathcal{S}_n)$.
3. If $\text{Flag}_{\text{Est}} = \texttt{False}$, then $\mathcal{M}$ allocates and charges nothing.
4. Otherwise, for each bidder k, run MetaRS with $\hat{\mathcal{M}}'$, $(\mathcal{S}_1, \dots, \mathcal{S}_n)$, and the rest of the required inputs. Let $\text{Coin}^{(k)}$, $\text{Flag}^{(k)}$, $s^{(k)}$, $\hat{\alpha}^{(k)}$, $\texttt{dummy}^{(k)}$ denote the output for k. Go to Phase 2.

Phase 2:

5. Run $\hat{\mathcal{M}}'$ on $s = (s^{(1)}, \dots, s^{(n)})$. Let $o = (o_1, \dots, o_m)$ and $(\hat{p}'_1, \dots, \hat{p}'_m)$ denote the resulting outcome and payments, randomly drawn from $\mathcal{A}'(s)$ and $\hat{p}'(s)$.
6. For each bidder k:
   - If $\text{Coin}^{(k)} = 0$ or $\exists k' \neq k$ s.t. $\text{Flag}^{(k')} = \texttt{False}$, bidder k gets $\perp$, pays 0 overall.
   - Else: charge bidder k a payment q_k with $\text{Payment}(t_k, \hat{\alpha}^{(k)}, (\mathcal{S}_{k'})_{k' \in [n]})$ for Phase 1. Then, if $\texttt{dummy}^{(k)} = \texttt{False}$ (i.e. bidder k was matched to real surrogate $s^{(k)}$), she gets o_k and pays $(1 - \beta_k) \cdot \hat{p}'_k$ for Phase 2. Else, she gets $\perp$ and pays 0 for Phase 2.

Draw-$\mathcal{S}$. Because edge weights $\hat{w}_{ij}$ are computed with respect to empirical distributions $(\hat{D}'_{k'})_{k' \in [n]}$, our revenue-approximation analysis will relate $\text{Rev}(\mathcal{M}; D)$ to the revenue and incentive loss of $\mathcal{M}'$ over $\hat{D}'$, instead of over D'. The Draw-$\mathcal{S}$ requirement (A1) ensures that surrogates are drawn in such a way that the these terms over $\hat{D}'$ are not far from their counterparts over D', so that our final bounds are expressed based on $\text{Rev}(\mathcal{M}'; D')$ and ε (recall that $\mathcal{M}'$ is ε-BIC over D').

EstMechanism. The purpose of EstMechanism is to allow for additional flexibility in obtaining Condition (A1) if needed. In particular, it is not important that it is $\mathcal{M}'$ that is run on $\hat{D}'$: any mechanism with comparable incentive guarantees and revenue on $\hat{D}'$ as $\mathcal{M}'$ on D' suffices for Condition (A1). Later, we demonstrate how the additional flexibility allows us to obtain stronger guarantees on the output of our framework under weaker assumptions. We suspect that the additional flexibility will be helpful for future work.

Draw-$\mathcal{R}$. Conditions (B2) through (B4) are all conditioned on $\text{Flag}_{\mathcal{R}} = \texttt{True}$, capturing the fact that Draw-$\mathcal{R}$ is allowed to fail with small probability. The replica indistinguishability requirement (B2) is needed to prove BIC/IR via the principle of deferred decisions. In prior work, this requirement is trivially satisfied, as the bidder's type and the rest of the replicas are all i.i.d draws from D_k; in our implementation, the proof is slightly more nuanced. The high-cardinality matching (B3) and coupling distance (B4) requirements will both be used in the proof of revenue approximation similarly to prior work [COVZ19]. It is in the coupling bound $\Delta^{(k)}_{\textsc{Couple}}$ that revenue loss due to the distance between D'_k and D_k will appear.

Compute-α. The approximate equal demand requirement (C1) states that the returned duals $\hat{\alpha}$ must be such that, with high probability, the surrogate market approximately clears. The approximate max-weight requirement (C2) is needed in the proof of revenue approximation similarly to prior work [COVZ19], as part of comparing the matching outputted by MetaRS to the high-cardinality matching guaranteed by (B3).

Payment. An important property of the softmax allocation for a type t implied by Theorem 4.6 is that there exists a payment rule that incentivizes truthful reports to this allocation mechanism (see e.g. Theorem 4.3, Lemma 5.7 in [DHKN17]).[25] Condition (E1) states quantitatively the condition for Payment to be such a payment rule for surrogate selection.

Another challenge in the payment rule is achieving the IR property (Condition (E2)) even when edge weights may be negative, as in our setting. As [COVZ19] discuss, a "rebate" in the payment is required to preserve IR to offset the negative utility from these edges, but too large of a rebate can drive revenue down. [COVZ19] introduce dummy surrogates $\{\circ_j\}_j$ and corresponding allocation variables $(y^*_{tj}(\hat{\alpha}))_j$ for exactly this purpose: to avoid matching along negative edges too often, and thereby ensure that the rebate required is not too large. Condition (E3) encodes the resulting guarantee that the expected payment should not be too small.[26]

A. Draw-$\mathcal{S}(D'_k) \to$ Output: $\mathcal{S}_k$

EstMechanism$(\mathcal{M}', \mathcal{S}_1, \dots, \mathcal{S}_n) \to$ Output: $\hat{\mathcal{M}}' = (\hat{\mathcal{A}}', \hat{p}')$, Flag$_{\text{Est}}$

$\mathcal{S}_k$ is a multiset $\{s_1, \dots, s_S\}$ of size S with each $s_j \in \mathcal{T}'_k$ such that the following holds. Let $\hat{D}'_{k'}$ denote the uniform distribution over $\mathcal{S}_{k'}$ for each bidder $k' \in [n]$ and $\hat{D}' := \times_{k' \in [n]} \hat{D}'_{k'}$. If Flag$_{\text{Est}} = \texttt{True}$, then $\hat{\mathcal{M}}'$ is a mechanism defined on the support of $\hat{D}'$.

(**A1**) **Empirical approximation.** With probability at least $1 - \delta_{\text{D-S}}$ over the randomness of Draw-$\mathcal{S}$ and EstMechanism, the following events hold simultaneously: Flag$_{\text{Est}} = \texttt{True}$, Rev$(\hat{\mathcal{M}}', \hat{D}') \geq$ Rev$(\mathcal{M}'; D') - \Delta_{\text{D-S}}$, and $\hat{\mathcal{M}}'$ is $(\varepsilon + \varepsilon_{\text{D-S}})$-BIC and $(\varepsilon + \varepsilon_{\text{D-S}})$-IR over $\hat{D}'$.

B. Draw-$\mathcal{R}(D_k, t_k) \to$ Output: $\mathcal{R}_k, i^*$, Flag$_{\mathcal{R}}$

Below, let σ denote the random process that consists of Preprocess(D_k); sampling $t_k \sim D_k$; and Draw-$\mathcal{R}(D_k, t_k)$. The outputs $\mathcal{R}_k, i^*$, Flag$_{\mathcal{R}}$ satisfy the following conditions. $\mathcal{R}_k$ is a multiset $\{r_1, \dots, r_R\}$ of size R with each $r_i \in \mathcal{T}_k$, where $R = \kappa \cdot S$ for some $\kappa \in \mathbb{N}$; i^* is an index in $[R]$; and Flag$_{\mathcal{R}}$ is a Boolean, such that:

(**B1**) First, $t_k = r_{i^*}$. Second, $\Pr_\sigma[\text{Flag}_{\mathcal{R}} = \texttt{False}] \leq \delta_{\text{D-R}}$.

[25] In particular, the softmax allocation rule is *maximal-in-range* [DHKN17, COVZ19].

[26] To satisfy all of these conditions simultaneously, we expect that most Payment implementations (like our own in Section 6) will use the procedure from [DHKN17, COVZ19] based on *implicit payments* [AT01, BKS18].

Moreover, for any output $\mathcal{G}$ of PREPROCESS, there exists a distribution $F_{k,\mathcal{G}}$ supported on $\mathcal{T}_k$ and a coupling $c_{k,\mathcal{G}}$ between $F_{k,\mathcal{G}}$ and D'_k[27] such that:

(B2) Replica indistinguishability. For any replica $r_i \in \mathcal{R}_k$, conditioned on $\text{FLAG}_{\mathcal{R}} = \texttt{True}$ and a particular $\mathcal{G}$, the distribution of r_i is $F_{k,\mathcal{G}}$:

$$\forall x, \quad \Pr_{\sigma}[r_i = x \mid \mathcal{G}, \text{FLAG}_{\mathcal{R}} = \texttt{True}] = \Pr_{X \sim F_{k,\mathcal{G}}}[X = x].$$

Further, conditioned on $\text{FLAG}_{\mathcal{R}} = \texttt{True}$, the bidder replica r_{i^*} is distributed identically to a *uniformly random* replica:

$$\forall x, \quad \Pr_{\sigma}[r_{i^*} = x \mid \text{FLAG}_{\mathcal{R}} = \texttt{True}] = \Pr_{\sigma, X \sim \text{UNIF}(\mathcal{R}_k)}[X = x \mid \text{FLAG}_{\mathcal{R}} = \texttt{True}]$$

(B3) High-cardinality matching. Consider fixed $\mathcal{S}_k, \mathcal{G}, \mathcal{R}_k$ with $\text{FLAG}_{\mathcal{R}} = \texttt{True}$, and a fixed realization of replica couples $c_{k,\mathcal{G}}(\mathcal{R}_k) := (c_{k,\mathcal{G}}(r_1), \ldots, c_{k,\mathcal{G}}(r_R))$. Among the κ-to-1 matchings between $\mathcal{R}_k$ and $\mathcal{S}_k$ that match some r_i to some s_j only if $c_{k,\mathcal{G}}(r_i)$ and s_j are d-close, let M denote the matching of maximum cardinality. Then the expected cardinality $|M|$ of M satisfies

$$\mathbb{E}[|M| \ \mid \ \text{FLAG}_{\mathcal{R}} = \texttt{True}] \geq R \cdot (1 - \Delta_{\text{MATCH}}),$$

where the expectation is over the randomness of DRAW-$\mathcal{S}$; σ; and the realization of the couples $c_{k,\mathcal{G}}(\mathcal{R}_k)$, *conditioned on* $\text{FLAG}_{\mathcal{R}} = \texttt{True}$.

(B4) Coupling distance. The expected distance between replicas $\mathcal{R}_k$ and their couples $c_{k,\mathcal{G}}(\mathcal{R}_k)$ satisfies the following, where the expectation is over σ and the realization of $c_{k,\mathcal{G}}(\mathcal{R}_k)$, *conditioned on* $\text{FLAG}_{\mathcal{R}} = \texttt{True}$:

$$\mathbb{E}\left[\sum_{r_i \in \mathcal{R}_k} \text{dist}_k(r_i, c_{k,\mathcal{G}}(r_i)) \mid \text{FLAG}_{\mathcal{R}} = \texttt{True}\right] \leq R \cdot \Delta^{(k)}_{\text{COUPLE}}.$$

For the following requirements, edge weights $\hat{w}$ are defined according to Theorem 4.1 with respect to the distribution $\hat{D}'$ as defined in the DRAW-$\mathcal{S}$ requirement. S and R are as defined in the DRAW-$\mathcal{S}$ and DRAW-$\mathcal{R}$ requirements respectively.

C. COMPUTE-$\alpha\left((\mathcal{S}_{k'})_{k' \in [n]}, D_k, \hat{\mathcal{M}}'\right) \to$ Output: $\hat{\alpha}$

Let σ be the random process that consists of PREPROCESS(D_k), COMPUTE-$\alpha((\mathcal{S}'_k)_{k'}, D_k, \hat{\mathcal{M}}')$, sampling $t_k \sim D_k$, and DRAW-$\mathcal{R}(D_k, t_k)$. Then $\hat{\alpha}$ satisfies the following, where $(x^*_{ij}(\hat{\alpha}), y^*_{ij}(\hat{\alpha}))_{i,j}$ denotes the softmax allocation that optimizes $\mathbf{P}'_\delta(\mathcal{R}_k, \mathcal{S}_k, \hat{w}, \hat{\alpha})$ for replicas $\mathcal{R}_k$, surrogates $\mathcal{S}_k$, edge weights $\hat{w} = (\hat{w}_{ij})_{i,j}$ and duals $\hat{\alpha}$ (Theorem 4.5):

(C1) Approximate equal demand: With probability at least $1 - \delta_{\text{DEMAND}}$ over σ, it holds for

[27] That is, $c_{k,\mathcal{G}}$ is a joint distribution on $\text{supp}(F_{k,\mathcal{G}}) \times \text{supp}(D'_k)$ with marginals $F_{k,\mathcal{G}}$ and D'_k. As before, we write $c_{k,\mathcal{G}}(x)$ to mean a random variable distributed according to the conditional distribution over $\text{supp}(D'_k)$ when the $\text{supp}(F_{k,\mathcal{G}})$ component is x.

some $0 \leq \lambda \leq \frac{1}{4}$ that

$$\forall j \in [S], \sum_{r_i \in \mathcal{R}_k} \left[x^*_{ij}(\hat{\alpha}) + y^*_{ij}(\hat{\alpha})\right] \in (1 \pm \lambda) \cdot \frac{R}{S}$$

(**C2**) **Approximate max-weight**: Let $\textsc{Opt}$ denote the optimum value of the program $\mathbf{P}(\mathcal{R}_k, \mathcal{S}_k, \hat{w})$. Then, for the same λ as in (C1),

$$\mathop{\mathbb{E}}_{\sigma}\left[\textsc{Opt} - \sum_{i \in [R], j \in [S]} (1-\lambda) \cdot x^*_{ij}(\hat{\alpha}) \cdot \hat{w}_{ij}\right] \leq R \cdot \Delta_{\mathrm{MWM}}.$$

D. $\textsc{Match}(r_i, \hat{\alpha}, (\mathcal{S}_{k'})_{k' \in [n]}) \to$ Output: some s_j or $\circ_j$

Let $(x^*_{ij}(\hat{\alpha}),\ y^*_{ij}(\hat{\alpha}))_j$ be the softmax allocation for surrogates $\mathcal{S}_k$, duals $\hat{\alpha}$, and replica r_i with edge weights $(\hat{w}_{ij})_j$ as in Theorem 4.6. Then the output of $\textsc{Match}$ satisfies the following:

(**D1**) The output is equal to $s_j \in \mathcal{S}_k$ with probability exactly $x^*_{ij}(\hat{\alpha})$ and to $\circ_j$ with probability exactly $y^*_{ij}(\hat{\alpha})$, for all $j \in [S]$ (over the internal randomness of $\textsc{Match}$).[28]

E. $\textsc{Payment}(t_k, \hat{\alpha}^{(k)}, (\mathcal{S}_{k'})_{k' \in [n]}) \to$ Output: q_k

Let $(x^*_{j}(\hat{\alpha}),\ y^*_{j}(\hat{\alpha}))_j$ be the softmax allocation for surrogates $\mathcal{S}_k$, duals $\hat{\alpha}$, and type t_k with edge weights $(\hat{w}_{t_k j})_j = (\hat{w}(t_k, s_j))_j$ as in Theorem 4.6. Then q_k satisfies the following, where all expectations are over the internal randomness of $\textsc{Payment}$:

(**E1**) **Incentive Compatibility**: There exists a constant C that does not depend on the report t_k such that

$$\mathbb{E}[q_k] = C + \delta \sum_{j \in [S]} (x^*_j(\hat{\alpha}) \log x^*_j(\hat{\alpha}) + y^*_j(\hat{\alpha}) \log y^*_j(\hat{\alpha})) + \sum_j \alpha_j (x^*_j(\hat{\alpha}) + y^*_j(\hat{\alpha})).$$

(**E2**) **Individual Rationality**: It holds that $\mathbb{E}[q_k] \leq \sum_{j \in [S]} x^*_j(\hat{\alpha}) \cdot \hat{w}_{t_k j}$.

(**E3**) **Lower-Bounded Payments**: It holds that $\mathbb{E}[q_k] \geq q_{\mathrm{LB}}$.

5.4 Meta-Result

The main result implied by the meta-framework (a "meta-result") is stated in Theorem 5.1. Any use of the meta-framework must supply implementations of each meta-input and specify parameter values such that the requirements are satisfied; then, the parameter values determine the revenue approximation loss and sample complexity of the resulting ε-BIC-to-BIC transformation as given by Theorem 5.1. We define the following notation for brevity in expressing the bounds.

- $N(D')$ is the number of samples from D' used by $\textsc{Draw-}\mathcal{S}$, and $N(D)$ is the total number of samples from D used by $\textsc{Preprocess}$, $\textsc{Draw-}\mathcal{R}$, and $\textsc{Compute-}\alpha$.[29] Note that, by construction of allowed inputs, no other meta-input algorithms can incur samples from D/D'.

[28] Moreover, the randomness of $\textsc{Match}$ must be *independent* over all replicas $r_i \in \mathcal{R}_k$ when called iteratively in $\textsc{MetaRS}$ Step 4, and independent of all other meta-inputs.

[29] $N(D')$ is equivalently the maximum number of samples from D'_k, over any k, needed by $\textsc{Draw-}\mathcal{S}$ for bidder k, since one sample from $D' = \times_{k \in [n]} D'_k$ yields a sample from each D'_k. The analogous is true for $N(D)$ and $D = \times_{k \in [n]} D_k$. In our implementation, the number of samples from each D'_k and D_k will be fixed across $k \in [n]$.

- Revenue loss is given by $\Delta_{\text{Rev}} := \Delta_{\text{D-S}} + \sum_{k\in[n]} \Delta^{(k)}_{\text{Rev}}$, where $\Delta^{(k)}_{\text{Rev}}$ is defined as follows:

$$\begin{aligned}\Delta^{(k)}_{\text{Rev}} &:= \beta_k + (n+1)\delta_{\text{Flag}} + \delta_{\text{D-S}} + \Delta_{\text{Match}} - \min(q_{\text{LB}}, 0)\\ &\quad + \frac{1}{\beta_k}\cdot(\Delta_{\text{MWM}} + \varepsilon + \varepsilon_{\text{D-S}} + \delta_{\text{D-S}} + 3d + 2\Delta^{(k)}_{\text{Couple}} + 3\delta_{\text{Flag}})\\ \text{for } \delta_{\text{Flag}} &:= \delta_{\text{Demand}} + \delta_{\text{D-R}} + S\cdot\exp\left(-\tfrac{\lambda^4}{6}\cdot\tfrac{R}{S}\right)\end{aligned}$$

We write out all of the components of the per-bidder revenue loss term $\Delta^{(k)}_{\text{Rev}}$ explicitly to enable easy substitution in future work. However, we expect that most applications (including the derivation of our polynomial sample complexity result) will simply end up bounding these components roughly like the following: $\Delta_{\text{MWM}}, \varepsilon_{\text{D-S}}, \delta_{\text{D-S}}, d, \Delta_{\text{Match}} = O(\varepsilon)$; $q_{\text{LB}} = -O(\varepsilon)$; $\delta_{\text{Flag}} = O\left(\frac{\varepsilon}{n}\right)$; $\Delta_{\text{D-S}} = O(n\varepsilon)$; and $\beta_k = \sqrt{\varepsilon}$. These bounds yield a final revenue loss of $\Delta_{\text{Rev}} = O(n\sqrt{\varepsilon})$, which matches that of the main ε-BIC-to-BIC reduction in [COVZ19].[30]

Theorem 5.1 (Meta-Framework: Main Result)**.** *Suppose we are given sample access to distributions $D' = \times_{k\in[n]} D'_k$ and $D = \times_{k\in[n]} D_k$ of bidder types, and query access to a mechanism $\mathcal{M}'$ that is IR and ε-BIC over D'. Given implementations of the meta-inputs* Draw-$\mathcal{S}$, EstMechanism, Preprocess, Draw-$\mathcal{R}$, Compute-α, Match, *and* Payment, *together with a choice of parameter values (Section 5.3.1), that satisfy the meta-framework requirements (Section 5.3), Algorithm 2 defines an IR and (exactly) BIC mechanism $\mathcal{M}$ over n bidders that achieves revenue* $\text{Rev}(\mathcal{M}; D) \geq \text{Rev}(\mathcal{M}'; D') - \Delta_{\text{Rev}}$ *using $N(D)$ samples from D and $N(D')$ samples from D'.*

The proof of Theorem 5.1 is deferred to Section A, where we separately prove the BIC and IR claim (Section A.1) and the revenue approximation claim (Section A.2).

5.5 Meta-Result: Interim IR to Ex-Post IR Reduction

In this section, we demonstrate how to guarantee that the output $\mathcal{M} = (\mathcal{A}, p)$ of the meta-framework is ex-post IR when we have access to the interim allocations and payments of the mechanism $\hat{\mathcal{M}}'$ output by EstMechanism on the empirical surrogate distribution $\hat{D}'$. A mechanism $\mathcal{M}$ is ex-post IR if for all bidders k and type profiles $t = (t_1, \ldots, t_n) \in \text{supp}(D)$,

$$v_k(t_k, \mathcal{M}(t)) - p(t) \geq 0$$

We could have access to the interim form for various reasons. For example, if querying $\mathcal{M}'$ on a type profile returns an explicit description of the distribution over allocations and payments (rather than a random outcome), then we could simply compute the interim payments since we have *direct access* to the empirical surrogate distribution (and since we are not concerned with computational efficiency). Alternatively, if querying $\mathcal{M}'$ only returns a random outcome, then we could repeatedly query $\mathcal{M}'$ to form an empirical distribution over outcomes for each type profile and attempt to use the mechanism that allocates and charges payments according to these empirical distributions as the output of EstMechanism. In any case, how we have access to the interim form of $\hat{\mathcal{M}}'$ on $\hat{D}'$

[30]The careful reader will notice that we have not specified $\Delta^{(k)}_{\text{Couple}}$ here: in general, when $D_k \neq D'_k$, this term will depend on some notion of distance between D_k and D'_k that becomes part of the revenue loss (and the value of β_k will be modified accordingly), as in our main result (Theorem 6.1) and prior work [COVZ19].

is not the concern of this section. Here, we assume access to interim form and demonstrate how to obtain an ex-post mechanism as a consequence of our meta-framework.

To do so, we apply a standard reduction for turning an interim IR mechanism into an ex-post IR mechanism with no incentive or revenue loss. Note that access to the interim allocations and payments of $\hat{\mathcal{M}}'$ on $\hat{D}'$ means that we know the edge weights $\hat{\omega}$ *exactly*. As a result, we not only know the utility that each bidder has for each surrogate (i.e., the edge weight), but we also know the distribution according to which she matches with her surrogate, e.g., $(x^*_{kj}(\hat{\alpha}^{(k)}),\, y^*_{kj}(\hat{\alpha}^{(k)}))_{j\in\mathcal{S}_k}$, and the expectation of the Phase 1 payment she must make in order to receive her surrogate's outcome, e.g., $\mathbb{E}[q_k]$.

Algorithm 3: Interim IR to ex-post IR reduction

Inputs:

1. Reported type t_k of each bidder k.
2. Interim access to $\hat{\mathcal{M}}' = (\hat{\mathcal{A}}', \hat{p}')$.
3. Edge weights/interim utilities $\hat{\omega}_{kj} = \hat{\omega}_k(t_k, s_j)$ with respect to $\hat{\mathcal{M}}'$ and $\hat{D}'$ (Theorem 4.1) for each bidder k and surrogate $s_j \in \mathcal{S}_k$.
4. Surrogate allocations $(x^*_k, y^*_k) = (x^*_j(\hat{\omega}_{kj}, \hat{\alpha}^{(k)}), y^*_j(\hat{\omega}_{kj}, \hat{\alpha}^{(k)}))_{s_j\in\mathcal{S}_k}$ (Theorem 4.4) for each bidder k.
5. Expected Phase 1 payment $\mathbb{E}[q_k]$ for each bidder k.
6. $\textsc{Coin}^{(k)}$, $\textsc{Flag}^{(k)}$, $s^{(k)}$, $\hat{\alpha}^{(k)}$, $\texttt{dummy}^{(k)}$ denote the output for each bidder k

Algorithm:

1. Run $\hat{\mathcal{M}}'$ on $s = (s^{(1)}, \ldots, s^{(n)})$. Let $o = (o_1, \ldots, o_n)$ denote the resulting outcome, randomly drawn from $\hat{\mathcal{A}}'(s)$.
2. For each bidder k:
   - If $\textsc{Coin}^{(k)} = 0$ or $\exists k' \neq k$ s.t. $\textsc{Flag}^{(k')} = \texttt{False}$, bidder k gets $\perp$, pays 0 overall.
   - Otherwise, if $\texttt{dummy}^{(k)} = \texttt{False}$ (i.e. bidder k was matched to real surrogate $s^{(k)}$), then she gets o_k and pays
   $$v_k(t_k, o_k) \cdot \frac{\mathbb{E}[q_k] + (1-\beta_k)\cdot \sum_{s_j\in\mathcal{S}_k} x^*_{kj} \cdot \mathbb{E}_{s_{-k}\sim \hat{D}'_{-k}}[\hat{p}'_k(s_j, s_{-k})]}{\sum_{s_j\in\mathcal{S}_k} x^*_{kj}\cdot \mathbb{E}_{s_{-k}\sim\hat{D}'_{-k}}[v_k(t_k, \hat{\mathcal{A}}'(s_j, s_{-k})]}$$

Theorem 5.2. *Suppose we are given sample access to distributions $D' = \times_{k\in[n]} D'_k$ and $D = \times_{k\in[n]} D_k$ of bidder types, and query access to a mechanism $\mathcal{M}'$ that is IR and ε-BIC over D'. Suppose we have implementations of the meta-inputs* $\textsc{Draw}$-$\mathcal{S}$, $\textsc{EstMechanism}$, $\textsc{Preprocess}$, $\textsc{Draw}$-$\mathcal{R}$, $\textsc{Compute}$-α, $\textsc{Match}$, *and* $\textsc{Payment}$, *together with a choice of parameter values (Section 5.3.1), that satisfy the meta-framework requirements (Section 5.3). If we have interim access to the output of* $\textsc{EstMechanism}$, *then replacing Phase 2 in Algorithm 2 with Algorithm 3 defines an ex-post IR and (exactly) BIC mechanism $\mathcal{M}$ over n bidders that achieves revenue* $\textsc{Rev}(\mathcal{M}; D) \geq \textsc{Rev}(\mathcal{M}'; D') - \Delta_{\textsc{Rev}}$ *using $N(D)$ samples from D and $N(D')$ samples from D'.*

We defer the proof to Section B. The high level idea, however, is simple: in general, if $\mathcal{M} = (\mathcal{A}, p)$

is an interim IR mechanism for D, then charging

$$v_k(t_k, \mathcal{A}(t_k, t_{-k})) \cdot \frac{\mathbb{E}_{t_{-k}\sim D_{-k}}[p_k(t_k, t_{-k})]}{\mathbb{E}_{t_{-k}\sim D_{-k}}[v_k(t_k, \mathcal{A}(t_k, t_{-k}))]}$$

ex-post means that the interim payment will exactly be $\mathbb{E}_{t_{-k}\sim D_{-k}}[p_k(t_k, t_{-k})]$, so the reduction preserves the incentives and revenue of $\mathcal{M}$ exactly. It is not hard to see the new ex-post payments yield an ex-post IR mechanism since interim IR implies that the interim price over the interim value is at most 1, so the ex-post payment never exceeds the ex-post value.

6 Independent Items: Polynomial Sample Complexity

We are now ready to apply the meta-framework to prove our main result (Theorem 6.1): a sample-efficient and revenue-preserving ε-BIC-to-BIC transformation in the independent items setting.

6.1 Model and Notation

The model in this setting is defined by the following two assumptions on types and valuations. This is the same model as in the prior work of [GW18].

1. **Independent items**: Each bidder's distribution is itself a product distribution over m *items*. That is, for all $k \in [n]$, $D_k = \times_{\ell\in[m]} D_{k\ell}$ and $D'_k = \times_{\ell\in[m]} D'_{k\ell}$. We also assume that each item-marginal $D_{k\ell}$ and $D'_{k\ell}$ is supported on $[0, H]$. In particular, we can decompose the type $t_k \sim D_k$ of bidder k as $(t_{k,1}, \ldots, t_{k,m})$ where the $t_{k,\ell}$'s are independent random variables in $[0, H]$ drawn from $D_{k\ell}$ (and analogously for D'_k).
2. **Lipschitz valuations**: Each bidder k's valuation function is Lipschitz in the following sense: there is some absolute constant L such that for any $k \in [n]$ and any types $t = (t_1, \ldots, t_m), t' = (t'_1, \ldots, t'_m)$ such that t' matches t except for a change of δ on a single coordinate $\ell \in [m]$, then for any $o \in \mathcal{O}$, $|v_k(t, o) - v_k(t', o)| \leq L \cdot \delta$.

For ease of exposition and cleanliness of proofs, we will assume henceforth that $H = 1$ and $L = 1$, and we continue to assume $v_k(\cdot, \cdot) \in [0, 1]$. Extending to the case of arbitrary L and H is simply a matter of updating parameter values to be in terms of these constants. We also define the following notation for convenience, as in [GW18]:

- For any $x \in [0, 1]$, $\lfloor x \rfloor_\eta$ denotes the value of x rounded down to the nearest multiple of η.
- The unit interval discretized to multiples of η is denoted by $\mathcal{I}_\eta := \{\lfloor x \rfloor_\eta : x \in [0, 1]\}$.
- For any distribution F supported on $[0, 1]$, $\lfloor F \rfloor_\eta$ denotes the distribution of $\lfloor x \rfloor_\eta$ for $x \sim F$.

6.2 Meta-Input Implementations: Bounded Ex-Post Payments

We begin by instantiating the meta-framework under the assumption that the ex-post payments of the mechanism $\mathcal{M}'$ for the surrogate distribution are bounded in $[0, 1]$. In this case, EstMechanism need not do anything fancy and can simply return the blackbox mechanism $\mathcal{M}'$: concentration results will imply that $\mathcal{M}'$ achieves good incentive and revenue guarantees on the empirical surrogate distribution $\hat{D}'$ with high probability. We opt for this presentation for a few reasons. The first is that it is conceptually cleaner, and much of the technical difficulty and innovation is already present. The second is that it demonstrates how more advanced techniques such as Bernoulli factories and implicit payments, which play crucial roles in computationally "efficient"[31] ε-BIC-to-BIC reductions [DHKN17, COVZ19], fit into our meta-framework. The third reason is that our approach when ex-post payments are bounded truly treats $\mathcal{M}'$ as a blackbox: to get the ultimate allocation and payments for the bidders, we simply query $\mathcal{M}'$ on their surrogates. Later, we will discuss how to remove the bounded payments assumption at the expense of treating $\mathcal{M}'$ as a blackbox in this sense: while we will get the ultimate allocation by querying $\mathcal{M}'$, we will have to toss out the payment rule of $\mathcal{M}'$ and compute our own payments from scratch. The additional flexibility in Condition (A1) offered by EstMechanism allows us to use the resulting mechanism in place of $\mathcal{M}'$.

To instantiate the meta-framework in this model, we give implementations of the Draw-$\mathcal{S}$, Preprocess, Draw-$\mathcal{R}$, Compute-α, Match, and Payment meta-inputs, all in the context of a particular bidder k, and specify the parameter values for which these implementations satisfy the meta-framework requirements.

Our implementations can be summarized as follows. Roughly speaking, both Draw-$\mathcal{S}$ and Draw-$\mathcal{R}$ are implemented by setting $\mathcal{S}_k$ and $\mathcal{R}_k$ to be a product set of i.i.d samples (including the bidder's type for Draw-$\mathcal{R}$) from the item-marginals $(D'_{k\ell})_{\ell\in[m]}$ and $(D_{k\ell})_{\ell\in[m]}$, respectively. Compute-α computes approximate duals $\hat{\alpha}$ for $(\mathcal{R}_k, \mathcal{S}_k)$ by drawing a set of "training replicas" $\mathcal{R}_k^{\mathrm{Tr}}$ and estimating the optimal duals for $(\mathcal{R}_k^{\mathrm{Tr}}, \mathcal{S}_k)$. Preprocess is used to estimate an *effective support* of a discretized version of each $D_{k\ell}$ that is used by both Draw-$\mathcal{R}$ and Compute-α, to restrict draws from $D_{k\ell}$ to only this effective support, which is needed for market-clearing guarantees to generalize from $\mathcal{R}_k^{\mathrm{Tr}}$ to $\mathcal{R}_k$ in the analysis. Finally, Match is implemented via the Fast Exponential Bernoulli Race, and Payment is implemented via an *implicit payments* procedure like in prior work [DHKN17, COVZ19].

Formal algorithm descriptions and more detailed explanations are given below. We first introduce the following variables as notation, whose meaning will be clarified when we present the meta-input algorithms in which they are used.

- η: Discretization parameter, used to discretize the support of distributions.
- N_P, N_S, N_R: Number of draws per item-marginal in Preprocess, Draw-$\mathcal{S}$, and Draw-$\mathcal{R}$, respectively.
- N_{Edge}: Number of samples used to estimate edge weights as part of Compute-α.
- A: Threshold value for final duals $\hat{\alpha}$ in Compute-α.

We will set the values of these variables, and the values of all parameters of the meta-framework (Section 5.3.1), in the proof of Theorem 6.1. We defer proofs that these algorithms satisfy meta-framework requirements, which comprise the vast majority of the proof of Theorem 6.1, to Section C.

Draw-$\mathcal{S}$ and EstMechanism implementations. The Draw-$\mathcal{S}$ algorithm is simple: it draws N_S i.i.d samples from each item-marginal $D'_{k\ell}$ and returns the product set of these draws. Formally,

[31] Recall that these reductions run in time polynomial in the size of the type space, which can be exponential in the number of items.

for any sets $X_1, \ldots, X_m$, the product set $X = \times_{\ell \in [m]} X_\ell$ is defined as $X := \{(x_1, \ldots, x_m) : x_\ell \in X_\ell, \forall \ell \in [m]\}$. EstMechanism is even simpler: it will simply return the blackbox mechanism $\mathcal{M}'$.

Algorithm 4: Draw-$\mathcal{S}$

Inputs: Sample access to $D'_k = \times_{\ell \in [m]} D'_{k\ell}$. Parameter: N_S.

Algorithm:

1. For each $\ell \in [m]$, sample N_S draws i.i.d from $D'_{k\ell}$ to form multiset $\mathcal{S}_{k\ell}$.
2. Return $\mathcal{S}_k := \times_{\ell \in [m]} \mathcal{S}_{k\ell}$.

Algorithm 5: EstMechanism

Inputs: Query access to $\mathcal{M}' = (\mathcal{A}', p')$.

Algorithm:

1. Return $\mathcal{M}'$.

We prove in Section C.1 that the empirical approximation requirement (Condition (A1)) is satisfied when ex-post payments lie in $[0, 1]$ for the desired revenue and incentive errors $\Delta_{\text{D-S}}, \varepsilon_{\text{D-S}}$ and desired failure probability $\delta_{\text{D-S}}$, by taking N_S to be appropriately poly $\left(n, m, \frac{1}{\varepsilon}\right)$. The proof follows from a concentration inequality for product distributions [BBP17, DHP16] stated in Theorem C.8.

Preprocess implementation. The purpose of our Preprocess algorithm is to learn which "buckets" of each item-marginal $D_{k\ell}$ that have probability mass lower bounded by some appropriate poly $\left(\varepsilon, \frac{1}{n}, \frac{1}{m}\right)$, for use in Draw-$\mathcal{R}$ and Compute-α. To implement this, Preprocess draws N_P i.i.d samples from each $\lfloor D_{k\ell} \rfloor_\eta$,[32] and marks any point in $\mathcal{I}_\eta$ that was sampled at least some threshold τ number of times, where $\mathcal{I}_\eta$ is the additive η-discretization of each item-marginal's support.

We will choose η such that any two types in the same η-bucket on each marginal are sufficiently close in valuation distance $\text{dist}_k(\cdot, \cdot)$ (by Lipschitz valuations) for guarantees in Compute-α to go through. We choose $\tau := \left(\frac{nm}{\varepsilon}\right)^2$ and set N_P to be sufficiently large poly $\left(n, m, \frac{1}{\varepsilon}\right)$ to (a) obtain the appropriate probability mass lower bound on identified buckets (points in $\mathcal{G} = (G_{k\ell})_{\ell \in [m]}$), while (b) ensuring that the total probability mass outside of each $G_{k\ell}$ is upper bounded by another appropriate poly $\left(\varepsilon, \frac{1}{n}, \frac{1}{m}\right)$. While Preprocess itself has no meta-framework requirements, our analysis in Section C.2 will formalize points (a) and (b), using straightforward Chernoff bounds, for use in later analysis.

[32] Note that we can always sample from $\lfloor D_{k\ell} \rfloor_\eta$ by sampling from $D_{k\ell}$ and then rounding down to the nearest integer multiple of η.

Algorithm 6: Preprocess

Inputs: Sample access to $D_k = \times_{\ell \in [m]} D_{k\ell}$. Parameters: N_P, η.

Algorithm:

1. For each $\ell \in [m]$, draw N_P i.i.d samples from $\lfloor D_{k\ell} \rfloor_\eta$ to form multiset P.
2. For each $\ell \in [m]$, let $G_{k,\ell} = \{y \in \mathcal{I}_\eta : \left(\sum_{y' \in P} \mathbb{1}\{y' = y\}\right) \geq (\frac{nm}{\varepsilon})^2\}$.
3. Return $\mathcal{G} \leftarrow (G_{k,\ell})_{\ell \in [m]}$.

The global state $\mathcal{G} = (G_{k\ell})_{\ell \in [m]}$ is used in Draw-$\mathcal{R}$ and Compute-α to restrict replica item-marginal draws from each $D_{k\ell}$ to only these buckets. The inverse-polynomial mass lower bound of each bucket will enable a straightforward proof that a polynomial number of samples from each item-marginal suffices for functions over the training replicas $\mathcal{R}_k^{\mathrm{Tr}}$ vs. the real replicas $\mathcal{R}_k$ to be close *multiplicative* approximations. We define a helper subroutine Draw as a rejection sampling-style procedure to draw a given number N of samples from $D_{k\ell}$ conditioned on $G_{k\ell}$, with its own flag $\textsc{Flag}_{\textsc{Draw}}$ to denote a failure condition if N from $G_{k\ell}$ are not obtained after a bounded number of draws (specifically, $2N$). Draw will be used in both Draw-$\mathcal{R}$ and Compute-α.

Subroutine: Draw(F, N, G)

Inputs: Distribution F with $\mathrm{supp}(F) \subseteq [0, 1]$; set $G \subseteq \mathcal{I}_\eta$; number of samples N. Parameter: η.

1. Draw $2N$ i.i.d from distribution F. Let X denote the multiset of samples.
2. Define multiset $X_\eta = \{x \in X : \lfloor x \rfloor_\eta \in G\}$.
3. If $|X_\eta| \geq N$, return the first N draws in X_η and $\textsc{Flag}_{\textsc{Draw}} \leftarrow$ `True`.
4. Else return the first N draws in X and $\textsc{Flag}_{\textsc{Draw}} \leftarrow$ `False`.

Draw-$\mathcal{R}$ **implementation.** Similarly to Draw-$\mathcal{S}$, the Draw-$\mathcal{R}$ algorithm sets $\mathcal{R}_k$ to the product $\times_{\ell \in [m]} \mathcal{R}_{k\ell}$ of draws $\mathcal{R}_{k\ell}$ from each marginal $D_{k\ell}$. However, rather than drawing i.i.d samples from $D_{k\ell}$, each $\mathcal{R}_{k\ell}$ is obtained by drawing $N_R - 1$ samples via Draw, using the state $\mathcal{G}$ from Preprocess, and then concatenating the item-marginal $t_{k,\ell}$ of the bidder's type. Draw-$\mathcal{R}$ raises a failure event $\textsc{Flag}_{\mathcal{R}} =$ `False` if any Draw raises $\textsc{Flag}_{\textsc{Draw}} =$ `False`, or if any $t_{k,\ell}$ does not lie in a bucket covered by $G_{k\ell}$.

Algorithm 7: DRAW-$\mathcal{R}$

Inputs: Sample access to D_k; bidder type t_k. Global state $\mathcal{G} = (G_{k\ell})_{\ell\in[m]}$. Parameters: N_R, η.

Algorithm:

1. For each $\ell \in [m]$, run DRAW$(D_{k\ell}, N_R - 1, G_{k\ell})$ and let $\mathcal{R}^-_{k\ell}$, FLAG$^{(\ell)}_{\text{DRAW}}$ denote the output.
2. If $\exists \ell$ s.t. FLAG$^{(\ell)}_{\text{DRAW}} =$ `False` or $\lfloor t_{k,\ell} \rfloor_\eta \notin G_{k\ell}$, set FLAG$_{\mathcal{R}} \leftarrow$ `False`, else FLAG$_{\mathcal{R}} \leftarrow$ `True`.
3. Let $\mathcal{R}_{k\ell}$ be the concatenation of $\mathcal{R}^-_{k\ell}$ and $t_{k,\ell}$.
4. Return $\mathcal{R}_k := \times_{\ell\in[m]} \mathcal{R}_{k\ell}$, FLAG$_{\mathcal{R}}$, and the appropriate index i^* of $t_k = (t_{k,1}, \ldots, t_{k,m})$ in $\mathcal{R}_k$.

The result of this more complicated marginal drawing procedure, as we will see in the analysis (Section C.3), is that conditioned on FLAG$_{\mathcal{R}} =$ `True` and a particular $\mathcal{G}$, the replica item-marginals in $\mathcal{R}_{k\ell}$ will be i.i.d according to the distribution $F_{\mathcal{G},\ell}$ defined by $(x \sim D_{k\ell} \mid \lfloor x \rfloor_\eta \in G_{k\ell})$. Thus, replica indistinguishability (Condition (B2)) will remain true, albeit according to a distribution other than $D_{k\ell}$ like in basic replica-surrogate matching. The argument for high-cardinality matching (Condition (B3)) is conceptually similar to that in prior work [HKM11, RW15, COVZ19], but must now be tailored to handle product-based replicas that are not i.i.d samples themselves. The analysis will moreover show that the coupling bound $\Delta^{(k)}_{\text{COUPLE}}$ in Condition (B4) depends on the natural notion of product-restricted Wasserstein distance $d^W_{\text{prod}}(D_k, D'_k)$ between D_k and D'_k.

COMPUTE-α **implementation.** The COMPUTE-α algorithm learns duals $\hat{\alpha}$ for the matching on $(\mathcal{R}_k, \mathcal{S}_k)$—without access to $\mathcal{R}_k$—by instead considering the matching on $(\mathcal{R}^{\text{Tr}}_k, \mathcal{S}_k)$, where $\mathcal{R}^{\text{Tr}}_k$ is a set of "training replicas" drawn independently of $\mathcal{R}_k$. In particular, the algorithm first draws a set of N_R samples from each $D_{k\ell}$ via DRAW and sets $\mathcal{R}^{\text{Tr}}_k$ to be the product set of these draws, like in DRAW-$\mathcal{R}$ except now *excluding* the bidder's type. We would now like to solve for the optimal duals $\arg\min_\alpha \mathbf{P}'_\delta(\mathcal{R}^{\text{Tr}}_k, \mathcal{S}_k, \hat{w}, \alpha)$ on $(\mathcal{R}^{\text{Tr}}_k, \mathcal{S}_k)$; however, we cannot compute the exact edge weights $\hat{w}$ given only query access to $\mathcal{M}'$. To handle this, we instead use estimates $\tilde{w}$ of $\hat{w}$ computed straightforwardly via the ESTEDGES subroutine. As a final step, we shift and threshold the optimal duals $\tilde{\alpha} := \arg\min_\alpha \mathbf{P}'_\delta(\mathcal{R}^{\text{Tr}}_k, \mathcal{S}_k, \tilde{w}, \alpha)$ for technical reasons in the analysis.

Our analysis of COMPUTE-α in Section C.4 will show the returned $\hat{\alpha}$ are indeed approximately optimal for $(\mathcal{R}_k, \mathcal{S}_k)$ (quantified in Condition (C1)). We do this by proving that the softmax allocation demand for each surrogate with duals $\hat{\alpha}$ on $(\mathcal{R}_k, \mathcal{S}_k)$ is close to the demand with the optimal duals $\tilde{\alpha}$ on $(\mathcal{R}^{\text{Tr}}_k, \mathcal{S}_k)$, where the latter term is exactly $\kappa = \frac{R}{S}$. This argument proceeds in a sequence of steps. We show that $\mathcal{R}_k$ and $\mathcal{R}^{\text{Tr}}_k$ (with values rounded to η-multiples) are close as empirical distributions, and so sums over $\mathcal{R}_k$ and $\mathcal{R}^{\text{Tr}}_k$ (specifically, the aggregate demand $\sum_{i\in[R]} x^*_{ij} + y^*_{ij}$) are close. We then show that these sums do not deviate significantly due to the discretization of replica item-marginals to η-multiples; the estimation of edge weights; and the thresholding of the α's. We highlight that the softmax allocation's *smoothness* appears as a crucial property of entropy-regularized matching (relative to max-weight matching) for this argument, ensuring that discretizations incur minimal error. Meanwhile, the approximate max-weight requirement (Condition (C2)) is fairly straightforward, after observing that the entropy-regularization term and the Lagrangian term for demand constraints do not alter the optimal value of the entropy-regularized matching objective too much relative to the exact max-weight matching.

Algorithm 8: COMPUTE-α

Inputs: Sample access to D_k. Query access to $\hat{\mathcal{M}}'$. $(\mathcal{S}_{k'})_{k'\in[n]}$. Global state $\mathcal{G} = (G_{k\ell})_{\ell\in[m]}$. Parameters: $N_R, N_{\text{EDGE}}, \eta, \beta_k, A, \delta$.

Algorithm:

1. For each $\ell \in [m]$, run DRAW$(D_{k\ell}, N_R, G_{k\ell})$ and let $\mathcal{R}'_{k\ell}, \text{FLAG}^{(\ell)'}_{\text{DRAW}}$ denote the output.
2. Set *training replicas* $\mathcal{R}_k^{\text{Tr}} := \times_{\ell\in[m]}\mathcal{R}'_{k\ell}$.
3. Compute estimated edge weights $\tilde{w} = (\tilde{w}_{ij})_{i,j} \leftarrow \text{ESTEDGES}(\mathcal{R}_k^{\text{Tr}}, (\mathcal{S}_{k'})_{k'\in[n]})$.
4. Solve for the optimal duals $\tilde{\alpha} = (\tilde{\alpha}_j)_{j\in[S]} := \arg\min_\alpha \mathbf{P}'_\delta(\mathcal{R}_k^{\text{Tr}}, \mathcal{S}_k, \tilde{w}, \alpha)$.
5. Let $\tilde{\alpha}^0 := \min_{j'} \tilde{\alpha}_{j'}$. Define $\hat{\alpha}_j := \min\{\tilde{\alpha}_j - \tilde{\alpha}^0, A\}$ for each j.
6. Return $\hat{\alpha} = (\hat{\alpha}_j)_{j\in[S]}$.

Subroutine: ESTEDGES

Notation: For each k', let $\hat{D}'_{k'}$ denote the uniform distribution over $\mathcal{S}_{k'}$.

1. For $p = 1 \to N_{\text{EDGE}}$:
   - Draw $t_{\text{-}k} \sim \hat{D}'_{\text{-}k}$. Then, for each $r_i \in \mathcal{R}_k^{\text{Tr}}, s_j \in \mathcal{S}_k$, query $\hat{\mathcal{M}}' = (\mathcal{A}', \hat{p}')$ to set
   $$w_{ij}^{(p)} := v_k(r_i, \mathcal{A}'(s_j; t_{\text{-}k})) - (1-\beta_k)\hat{p}'_k(s_j; t_{\text{-}k}).^{a}$$
2. For each $r_i \in \mathcal{R}_k^{\text{Tr}}, s_j \in \mathcal{S}_k$, return $\tilde{w}_{ij} := \frac{1}{N_{\text{EDGE}}}\sum_{p=1}^{N_{\text{EDGE}}} w_{ij}^{(p)}$.

[a] When we write $v_k(r_i, \mathcal{A}'(s_j; t_{\text{-}k}))$ here, we now mean *sampling* an outcome $o \sim \mathcal{A}'(s_j; t_{\text{-}k})$, not taking an expectation over this randomness like we usually mean.

MATCH **implementation.** The MATCH algorithm is straightforward via the Fast Exponential Bernoulli Race (Theorem 4.7). Note below that for each $s_j \in \mathcal{S}_k$, we can sample from the distribution with mean $\hat{w}_{ij} = \mathbb{E}_{t_{\text{-}k}\sim\hat{D}'_{\text{-}k}}[v_k(r_i, \mathcal{A}'(s_j; t_{\text{-}k})) - (1-\beta_k)\hat{p}'_k(s_j; t_{\text{-}k})]$ by sampling $t_{\text{-}k} \sim \hat{D}'_{\text{-}k}$ and $o \sim \mathcal{A}'(s_j; t_{\text{-}k})$ and returning $v_k(r_i, o) - (1-\beta_k)\hat{p}'_k(s_j; t_{\text{-}k})$.

PAYMENT **implementation.** For PAYMENT, we would like to simply charge type t_k

$$\delta \sum_j (x^*_{t_kj}(\hat{\alpha}) \log x^*_{t_kj}(\hat{\alpha}) + y^*_{t_kj}(\hat{\alpha}) \log y^*_{t_kj}(\hat{\alpha})) + \sum_j \hat{\alpha}_j (x^*_{t_kj}(\hat{\alpha}) + y_{t_kj}(\hat{\alpha}))$$

to satisfy incentive compatibility (Condition (E1), based on Fact 4.6). However, recall that the terms $(x^*_{t_kj}(\hat{\alpha}), y^*_{t_kj}(\hat{\alpha}))_j$ depend on the edge weights $(\hat{w}_{t_kj})_j$, which we cannot compute given query access to $\mathcal{A}'$. To resolve this issue, we use *implicit payment* computation [AT01, BKS18] to implement the payment rule as in [DHKN17, COVZ19], using the Fast Exponential Bernoulli Race to sample from the required softmax allocations in the algorithm below. We will show in Section C.6 that the resulting randomized output q_k satisfies meta-framework requirements.

Algorithm 9: MATCH

Inputs: Replica r_i, duals $\hat{\alpha}$, surrogate sets $(\mathcal{S}_{k'})_{k'\in[n]}$. Parameters: β_k, δ.

For all $k' \in [n]$, let $\hat{D}'_{k'}$ denote the uniform distribution over $\mathcal{S}_{k'}$. Let $\hat{D}' := \times_{k'} \hat{D}'_{k'}$. Let $(\hat{w}_{ij})_j$ be as defined in Theorem 4.1 with respect to $\hat{D}'$. Let $S := |\mathcal{S}_k|$.

Algorithm:

1. Let $N := 2S$. For each $j \in \{S+1, \dots N\}$ (the dummy surrogates), note that $\hat{w}_{ij} = 0$.
2. For each $j \in [N]$: if $j \leq S$, set $\alpha_j = \hat{\alpha}_j$; else $\alpha_j = \hat{\alpha}_{j-S}$.
3. Run the Fast Exponential Bernoulli Race (Theorem 4.7) with δ, $(\alpha_j)_{j\in[N]}$, and distributions with expectations $(\hat{w}_{ij})_{j\in[N]}$.[a] This samples some $I \in [N]$.
4. Return s_j if $I = j \in [S]$, else return $\circ_j$ if $I = S + j$.

[a]The distribution for a real surrogate s_j for $j \in [S]$ is given by sampling $s_{-k} \sim \hat{D}'_{-k}$ and querying $\hat{\mathcal{M}}'$ on the surrogate profile (s_j, s_{-k}). Note that we have sample access to these distributions since we have query access to $\hat{\mathcal{M}}'$, and sampling from $\hat{D}'_{-k}$ requires no additional samples from D'_{-k}. The distribution for a dummy surrogate $\circ_j$ for $j \in \{S+1, \dots, N\}$ is simply the degenerate distribution that is 0 with probability 1 (and hence, mean $\hat{w}_{ij} = 0$).

Algorithm 10: PAYMENT

Inputs: Type t_k, duals $\hat{\alpha}$, surrogate sets $(\mathcal{S}_{k'})_{k'\in[n]}$. Parameters: β_k, δ.

Algorithm: (Implicit payments, [AT01, BKS18])

1. Sample $\lambda \sim U[0,1]$ and $t_{-k} \sim \hat{D}'_{-k}$.
2. Sample $s' \sim (x^*_{t_k}(\hat{\alpha}), y^*_{t_k}(\hat{\alpha}))$ and $s'' \sim (x^\lambda_{t_k}(\hat{\alpha}), y^\lambda_{t_k}(\hat{\alpha}))$ where
$$x^\lambda_{t_k j}(\hat{\alpha}) := \frac{\exp(\frac{\lambda \hat{w}_{t_k j} - \hat{\alpha}_j}{\delta})}{\sum_{j'} \exp\left(\frac{-\hat{\alpha}_{j'}}{\delta}\right)\left(1 + \exp(\lambda \hat{w}_{t_k j'}/\delta)\right)}$$
$$y^\lambda_{t_k j}(\hat{\alpha}) := \frac{\exp(\frac{-\hat{\alpha}_j}{\delta})}{\sum_{j'} \exp\left(\frac{-\hat{\alpha}_{j'}}{\delta}\right)\left(1 + \exp(\lambda \hat{w}_{t_k j'}/\delta)\right)}$$
via Fast Exponential Bernoulli Races (Theorem 4.7).
3. Charge $q_k := W(t_k, s') - W(t_k, s'') - \delta \log 2$, where
$$W(t_a, t_b) := v_k(t_a, \mathcal{A}'(t_b; t_{-k})) - (1 - \beta_k) \cdot \hat{p}'_k(t_b; t_{-k}).^{a}$$

[a]Like in ESTEDGES, here we mean sampling an outcome $o \sim \mathcal{A}'(s_j; t_{-k})$), not taking an expectation over these processes.

6.3 Main Result

Using the above meta-input implementations, we can invoke Theorem 5.1 of the meta-framework to prove the main result of this paper, stated formally in Theorem 6.1.

Theorem 6.1 (Main Result: Polynomial Sample Complexity)**.** *Suppose we are given sample access to distributions $D' = \times_{k\in[n]} D'_k$ and $D = \times_{k\in[n]} D_k$ of bidder types, with n bidders and m items, and query access to a mechanism $\mathcal{M}'$ that is IR and ε-BIC over D' and has ex-post payments bounded in $[0,1]$. Then, under the independent items and Lipschitz valuations assumptions, we can construct a mechanism $\mathcal{M}$ that is IR and (exactly) BIC over D and achieves expected revenue*

$$\textsc{Rev}(\mathcal{M}; D) \geq \textsc{Rev}(\mathcal{M}'; D') - O(n\sqrt{\varepsilon}) - \sum_{k\in[n]} \sqrt{d^W_{\text{prod}}\left(D_k, D'_k\right)}$$

using only poly $\left(n, m, \frac{1}{\varepsilon}\right)$ samples from D and D'. Moreover, if $\mathcal{M}$ assigns an allocation and Phase 2 payment to a bidder, then they are the allocation and payment that $\mathcal{M}'$ assigns to the bidder's surrogate.

Theorem 6.2 below states the revenue loss guaranteed by Theorem 6.1 in the case where $D = D'$. Note that this is the same revenue loss as in the analogous Corollary 1 in [COVZ19] (the difference being that it is now achieved using only poly $\left(n, m, \frac{1}{\varepsilon}\right)$ samples). In particular, Theorem 6.2 affirmatively resolves Open Problem 1 in [GW18].

Corollary 6.2. *When $D = D'$, the mechanism $\mathcal{M}$ guaranteed by Theorem 6.1 achieves revenue*

$$\textsc{Rev}(\mathcal{M}; D) \geq \textsc{Rev}(\mathcal{M}'; D') - O(n\sqrt{\varepsilon}).$$

Proof of Theorem 6.1. We prove the theorem via Theorem 5.1. We first set the values of variables introduced for algorithm descriptions: $\eta = \frac{\varepsilon^3}{m^3 n}$; $N_P = \frac{2}{\eta}\cdot\left(\frac{nm}{\varepsilon}\right)^4 = 2\cdot\frac{m^7 n^5}{\varepsilon^7}$; $N_S = \frac{m^3 n}{\varepsilon^3}$; $N_R = \frac{m^{15} n^9}{\varepsilon^{15}}$; $A = 2\varepsilon + 4 = O(1)$; and $N_{\text{Edge}} = \frac{m^6 n^3}{\varepsilon^9}$.[33] We then choose the values of all parameters from the meta-framework (see Section 5.3.1) as follows. The number of surrogates and replicas are set to $S = {N_S}^m = \left(\frac{m^3 n}{\varepsilon^3}\right)^m$, and $R = {N_R}^m = \left(\frac{m^{15} n^9}{\varepsilon^{15}}\right)^m$, such that S divides R. The remaining parameters are chosen as follows: $\delta = \frac{\varepsilon}{m \log N_S}$; $\beta_k = O\left(\sqrt{\varepsilon + d^W_{\text{prod}}\left(D_k, D'_k\right)}\right)$; $q_{\text{LB}} = -O(\delta)$; $d, \lambda, \Delta_{\text{MWM}}, \Delta_{\text{Match}}, \varepsilon_{\text{D-S}}, \delta_{\text{D-S}} = O(\varepsilon)$; $\Delta_{\text{D-S}} = O(n\varepsilon)$; $\delta_{\text{D-R}}, \delta_{\text{Demand}} = O\left(\frac{\varepsilon}{n}\right)$; and $\Delta^{(k)}_{\text{Couple}} = O\left(\frac{\varepsilon}{n}\right) + d^W_{\text{prod}}\left(D_k, D'_k\right)$. Plugging these parameter values into the expression for Δ_{Rev} (Theorem 5.1) gives the claimed revenue loss.

The sample complexity now follows by inspection of the meta-input implementations. Draw-$\mathcal{S}$ requires N_S samples from D'. For D, N_P samples are required by Preprocess, $2(N_R - 1)$ samples are required by Draw-$\mathcal{R}$, and $2N_R$ samples are required by Compute-α. Thus a total of $N(D') = N_S = \text{poly}\left(n, m, \frac{1}{\varepsilon}\right)$ samples are required from D' and $N(D) = N_P + 4N_R - 2 = \text{poly}\left(n, m, \frac{1}{\varepsilon}\right)$ samples are required from D.

It remains to show that the meta-framework requirements laid out in Section 5.3 are satisfied by the meta-input implementations specified in Algorithms 6 through 10. The proofs of these requirements are technical and deferred to Section C. ☐

6.4 Meta-Input Implementations: Unbounded Ex-Post Payments

In this section, we present a way to remove the assumption that the ex-post payments of the surrogate mechanism $\mathcal{M}'$ are bounded. The implementations of Draw-$\mathcal{S}$, Preprocess, Draw-$\mathcal{R}$,

[33]The specific polynomial dependences on $n, m, \frac{1}{\varepsilon}$ of our variables/parameters are not optimized, as our goal is primarily to demonstrate that some polynomial sample complexity is feasible.

Compute-α, Match, and Payment remain exactly the same as when ex-post payments were bounded. However, instead of simply returning $\mathcal{M}'$ as a blackbox, EstMechanism will estimate the interim allocation rule of $\mathcal{M}'$ and use these estimates to compute a new payment rule $\hat{p}'$ such that the new mechanism $\hat{\mathcal{M}}' = (\mathcal{A}', \hat{p}')$ achieves comparable incentive guarantees and revenue as $\mathcal{M}'$. Given a profile of (surrogate) types t, the mechanism $\hat{\mathcal{M}}'$ queries and allocates according to $\mathcal{A}'(t)$ (the allocation of $\mathcal{M}'$), but instead of charging payments according to $p'(t)$ (the payment of $\mathcal{M}'$), $\hat{\mathcal{M}}'$ charges payments according to $\hat{p}'(t)$. The rest of the ε-BIC-to-BIC reduction then uses the new mechanism $\hat{\mathcal{M}}'$ instead of $\mathcal{M}'$. That is, when the ex-post payments of $\mathcal{M}'$ are unbounded, the output of our reduction matches each bidder to a surrogate and then queries $\hat{\mathcal{M}}'$ (instead of $\mathcal{M}'$ as in the bounded ex-post payments case) to determine the ultimate outcome. Essentially, we toss out the payment rule of the surrogate mechanism $\mathcal{M}'$ in favor of computing our own payment rule for $\mathcal{A}'$ from scratch.

More specifically, EstMechanism estimates the expected value of each surrogate in $\mathcal{S}_k$ for the allocation $\mathcal{A}'$ (when the surrogate distribution of the other bidders is $\hat{D}'_{-k}$) by repeatedly sampling $t_{-k} \sim \hat{D}'_{-k}$ and querying $\mathcal{A}'$. Note that $\hat{D}'_{-k}$ is the empirical surrogate distribution associated with the other bidders (formed by our surrogate samples), so sampling from it does not require any additional samples from D'. Querying $\mathcal{A}'$ also does not require additional samples. To compute $\hat{p}'$, EstMechanism simply writes a linear program using the estimated expected values. See Algorithm 11 for the formal implementation of this sub-routine.

Algorithm 11: EstMechanism

Inputs: Query access to $\mathcal{M}' = (\mathcal{A}', p')$. $(\mathcal{S}_k)_{k\in[n]}$. Parameter: N_S.

For all $k \in [n]$, let $\hat{D}'_k$ denote the uniform distribution over $\mathcal{S}_k$. Let $\hat{D}' := \times_k \hat{D}'_k$.

Algorithm:

1. Compute estimated values $(\hat{v}_k(t_k, t'_k))_{k,t_k,t'_k} \leftarrow \textsc{EstValues}((\mathcal{S}_k)_k)$

$$\begin{aligned}\max_{\hat{p}'} \quad & \mathbb{E}_{t\sim\hat{D}'}[\textstyle\sum_k \hat{p}'_k(t_k)] \\ \text{s.t.} \quad & \hat{v}_k(t_k,t_k) - \hat{p}'_k(t_k) \geq \hat{v}_k(t_k,t'_k) - \hat{p}'_k(t'_k) - (\varepsilon + \varepsilon_{\text{D-S}}) \,\forall\, k \in [n], t_k, t'_k \in \mathcal{S}_k \\ & \hat{v}_k(t_k,t_k) - \hat{p}'_k(t_k) \geq -\varepsilon_{\text{D-S}} \,\forall\, k \in [n], t_k \in \mathcal{S}_k\end{aligned}$$

2. If the above linear program[a] is infeasible, then return $\textsc{Flag}_{\textsc{Est}} = $ `False`.

3. Otherwise, let $\hat{p}'$ denote the optimal solution, and return $\hat{\mathcal{M}}' = (\mathcal{A}', \hat{p}')$ and $\textsc{Flag}_{\textsc{Est}} = $ `True`.

Subroutine: EstValues

1. For $\ell = 1 \to N_S$:
   - Draw $t_{\text{-}k} \sim \hat{D}'_{\text{-}k}$. For each $t_k, t'_k \in \mathcal{S}_k$, query $\mathcal{A}'$ to set
   $$\hat{v}_k^{(\ell)}(t_k, t'_k) := v_k(t_k, \mathcal{A}'(t'_k; t_{\text{-}k})).$$[b]

2. For each $t_k, t'_k \in \mathcal{S}_k$, return $\hat{v}_k(t_k, t'_k) := \frac{1}{N_S}\sum_{p=1}^{N_S} \hat{v}_k^{(\ell)}(t_k, t'_k)$.

[a] The linear program has $n \cdot N_S$ variables and $n \cdot {N_S}^{2m}$ constraints but does not require any additional samples from D' to write.

[b] When we write $v_k(t_k, \mathcal{A}'(t'_k; t_{\text{-}k}))$ here, we now mean *sampling* an outcome $o \sim \mathcal{A}'(t'_k; t_{\text{-}k})$, not taking an expectation over this randomness like we usually mean. Again, note that repeatedly querying $\mathcal{A}'$ does not require additional samples from D'.

We prove in Section D that the empirical approximation requirement (Condition (A1)) is satisfied by $\hat{\mathcal{M}}'$ for the desired revenue and incentive errors $\Delta_{\text{D-S}}, \varepsilon_{\text{D-S}}$ and desired failure probability $\delta_{\text{D-S}}$, by taking N_S to be appropriately $\text{poly}\left(n, m, \frac{1}{\varepsilon}\right)$. At a high level, a concentration inequality for product distributions [BBP17, DHP16] stated in Theorem C.8 implies that the expected value that each surrogate in $\mathcal{S}_k$ has for the allocation output by $\mathcal{A}'$ when the surrogate distribution of the other bidders is $\hat{D}'_{-k}$ and when it is D' is close. Hoeffding then implies that our estimates of the former expected values are close to the actual expected values. Then, since $\mathcal{M}'$ is ε-BIC, the interim payment rule[34] of $\mathcal{M}'$ constitutes a feasible solution to the linear program in EstMechanism with high probability. Meanwhile, since $\mathcal{M}'$ is IR, its interim payments are bounded, so applying Theorem C.8 to the interim payments (not the possibly unbounded ex-post payments) implies that the interim payment rule preserves revenue when the distribution is $\hat{D}'$ instead of D'. Since $\hat{p}'$ is an optimal solution to the linear program, it must obtain at least as much revenue.

Note that the exact specification of the surrogate mechanism plays no role in demonstrating that the remaining implementations satisfy the required conditions of the meta-framework, so we obtain the following extension of Theorem 6.1.

[34] That is, the payment rule that charges $\mathbb{E}_{t_{-k}\sim D'_{-k}}[p'(t_k, t_{-k})]$ when bidder k reports t_k.

Theorem 6.3. *Suppose we are given sample access to distributions $D' = \times_{k\in[n]} D'_k$ and $D = \times_{k\in[n]} D_k$ of bidder types, with n bidders and m items, and query access to a mechanism $\mathcal{M}'$ that is IR and ε-BIC over D'. Then, under the independent items and Lipschitz valuations assumptions, we can construct a mechanism $\mathcal{M}$ that is IR and (exactly) BIC over D and achieves expected revenue*

$$\textsc{Rev}(\mathcal{M}; D) \geq \textsc{Rev}(\mathcal{M}'; D') - O(n\sqrt{\varepsilon}) - \sum_{k\in[n]} \sqrt{d^W_{\text{prod}}\left(D_k, D'_k\right)}$$

using only poly $\left(n, m, \frac{1}{\varepsilon}\right)$ samples from D and D'. Moreover, if $\mathcal{M}$ assigns an allocation to a bidder, then it is the allocation that $\mathcal{M}'$ assigns to the bidder's surrogate.

6.5 Meta-Input Implementations: An Ex-Post IR Output

In this section, we demonstrate how to guarantee that the output $\mathcal{M} = (\mathcal{A}, p)$ of our ε-BIC-to-BIC transformation is ex-post IR. That is, for all bidders k and type profiles $t = (t_1, \ldots, t_n) \in \text{supp}(D)$,

$$v_k(t_k, \mathcal{M}(t)) - p(t) \geq 0.$$

Recall that Theorem 5.2 states that it suffices to implement EstMechanism in a way that allows us to compute the interim allocations and payments of the mechanism it outputs, so we modify EstMechanism as follows. We compute a new allocation rule $\hat{\mathcal{A}}'$ by repeatedly querying the allocation rule $\mathcal{A}'$ of the surrogate mechanism $\mathcal{M}'$ on each type profile from the empirical surrogate distribution. This process yields a multi-set of allocations for each type profile from this distribution. Then, given type profile, $\hat{\mathcal{A}}'$ outputs a uniformly random allocation from the multi-set corresponding to the type profile. That is, $\hat{\mathcal{A}}'$ is an "empirical" allocation rule. To compute a payment rule $\hat{p}'$ such that the mechanism $\hat{\mathcal{M}}' = (\hat{\mathcal{A}}', \hat{p}')$ is near-BIC for $\hat{D}'$ and obtains comparable revenue to $\mathcal{M}'$ on D', we again write a revenue maximization linear program but with the interim allocation rule fixed to be the one given by $\hat{\mathcal{A}}'$.[35] See Algorithm 12 for the formal implementation of this subroutine. Note that because we have explicit descriptions of $\hat{\mathcal{M}}'$ and $\hat{D}'$, we can compute the interim allocations and payments and apply the interim IR to ex-post IR reduction given in Section 5.5. The implementations of Draw-$\mathcal{S}$, Preprocess, Draw-$\mathcal{R}$, Compute-α, Match, and Payment can remain exactly the same as when ex-post payments were bounded.[36] We remark that in exchange for an ex-post IR output, the allocation and payment received by a bidder in $\mathcal{M}$ may no longer correspond to the allocation and payment of her surrogate in $\mathcal{M}'$ (although it will coincide with

[35]If the ex-post payments of $\mathcal{M}'$ are bounded in $[0, 1]$, then one can simply use the empirical payment rule formed by repeatedly querying $\mathcal{A}'$ on each surrogate profile instead of re-computing an entirely new payment rule from scratch.

[36]Because we have interim access to the output $\hat{\mathcal{M}}'$ of EstMechanism, we know the edge weights exactly and can simplify the implementations of Match and Payment. Rather than using Bernoulli factories and implicit payments, Match can simply just sample from the softmax allocation (x^*, y^*) exactly, and Payment can simply just charge the expectation $\mathbb{E}[q_k]$ directly.

the allocation and payment of her surrogate in $\hat{\mathcal{M}}'$).[37]

Algorithm 12: EstMechanism

Inputs: Query access to $\mathcal{M}' = (\mathcal{A}', p')$. $(\mathcal{S}_k)_{k\in[n]}$. Parameter: N_{Query}.

For all $k \in [n]$, let $\hat{D}'_k$ denote the uniform distribution over $\mathcal{S}_k$. Let $\hat{D}' := \times_k \hat{D}'_k$.

Algorithm:

1. Compute the empirical allocation rule $\hat{\mathcal{A}}' \leftarrow \text{EstAllocation}(\mathcal{M}', (\mathcal{S}_k)_k)$

$$\begin{aligned}
\max_{\hat{p}'} \quad & \mathbb{E}_{t\sim\hat{D}'}[\textstyle\sum_k \hat{p}'_k(t_k)] \\
\text{s.t.} \quad & \mathbb{E}_{t_{-k}\sim\hat{D}'_{-k}}[\hat{v}_k(t_k, \hat{\mathcal{A}}'(t_k, t_{-k}))] - \hat{p}'_k(t_k) \geq \mathbb{E}_{t_{-k}\sim\hat{D}'_{-k}}[\hat{v}_k(t_k, \hat{\mathcal{A}}'(t'_k, t_{-k}))] - \hat{p}'_k(t'_k) \\
& \qquad\qquad - (\varepsilon + \varepsilon_{\text{D-S}})\,\forall\, k \in [n], t_k, t'_k \in \mathcal{S}_k \\
& \mathbb{E}_{t_{-k}\sim\hat{D}'_{-k}}[\hat{v}_k(t_k, \hat{\mathcal{A}}'(t_k, t_{-k}))] - \hat{p}'_k(t_k) \geq -\varepsilon_{\text{D-S}}\,\forall\, k \in [n], t_k \in \mathcal{S}_k
\end{aligned}$$

2. If the above linear program[a] is infeasible, then return $\text{Flag}_{\text{Est}} = \texttt{False}$.

3. Otherwise, let $\hat{p}'$ denote the optimal solution, and return $\hat{\mathcal{M}}' = (\hat{\mathcal{A}}', \hat{p}')$ and $\text{Flag}_{\text{Est}} = \texttt{True}$.

Subroutine: EstAllocation

1. For each $t \in \text{supp}(\hat{D}')$,
   - Query $\mathcal{A}'(t)$ N_{Query} times, and let $\hat{\mathcal{A}}'(t)$ denote the uniform distribution over the sampled allocations $(o_\ell)_{\ell\in[N_{\text{Query}}]}$.

2. Return $\hat{\mathcal{A}}'$.

[a] The linear program has $n \cdot N_S$ variables and $n \cdot {N_S}^{2m}$ constraints but does not require any additional samples from D' to write.

We prove in Section E that the empirical approximation requirement (Condition (A1)) is satisfied by $\hat{\mathcal{M}}'$ for the desired revenue and incentive errors $\Delta_{\text{D-S}}, \varepsilon_{\text{D-S}}$ and desired failure probability $\delta_{\text{D-S}}$, by taking N_S and N_{Query} to be appropriately poly $\left(n, m, \frac{1}{\varepsilon}\right)$. The proof is highly reminiscent of the proof that Algorithm 11 satisfies Condition (A1), the main difference being that the former applies Hoeffding to each type profile (that is, we need a bidder's value to concentrate for each type profile), while the latter only applies Hoeffding to each type (we only need a bidder's interim value to concentrate for each type).

Theorem 6.4. *Suppose we are given sample access to distributions $D' = \times_{k\in[n]} D'_k$ and $D = \times_{k\in[n]} D_k$ of bidder types, with n bidders and m items, and query access to a mechanism $\mathcal{M}'$ that is*

[37] Given that we are no longer querying $\mathcal{M}'$ to determine each bidder's ultimate allocation and payment in $\mathcal{M}$, a natural question is whether we even need query access to $\mathcal{M}'$ in the first place. On one hand, if all we care about is revenue and sample complexity (and not computational complexity), then we could simply forget entirely about $\mathcal{M}'$ and instead compute the revenue-optimal mechanism for the empirical surrogate distribution $\hat{D}'$ by solving the revenue maximization linear program and take this mechanism as $\hat{\mathcal{M}}'$. Note that this approach is similar to [GW18]. On the other hand, there might other desirable properties of $\mathcal{M}'$ that could be preserved by sticking as close to its allocation and payment rule as possible. For this reason, we present different implementations of EstMechanism, each resulting in an output mechanism that is further and further from simply querying $\mathcal{M}'$ once the replica-surrogate matching is made.

IR and ε-BIC over D'. Then, under the independent items and Lipschitz valuations assumptions, we can construct a mechanism $\mathcal{M}$ that is ex-post IR and (exactly) BIC over D and achieves expected revenue

$$\text{Rev}(\mathcal{M}; D) \geq \text{Rev}(\mathcal{M}'; D') - O(n\sqrt{\varepsilon}) - \sum_{k \in [n]} \sqrt{d^W_{\text{prod}}\left(D_k, D'_k\right)}$$

using only poly $\left(n, m, \frac{1}{\varepsilon}\right)$ samples from D and D'.

References

[AKW14] Pablo Daniel Azar, Robert Kleinberg, and S. Matthew Weinberg. Prophet inequalities with limited information. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1358–1377, 2014.

[AT01] Aaron Archer and Éva Tardos. Truthful Mechanisms for One-Parameter Agents. In *the 42nd Annual Symposium on Foundations of Computer Science (FOCS)*, 2001.

[BB24] Maria-Florina Balcan and Hedyeh Beyhaghi. New guarantees for learning revenue maximizing menus of lotteries and two-part tariffs, 2024.

[BBP17] Yakov Babichenko, Siddharth Barman, and Ron Peretz. Empirical distribution of equilibrium play and its testing application. *Math. Oper. Res.*, 42(1):15–29, January 2017.

[BCKW15] Patrick Briest, Shuchi Chawla, Robert Kleinberg, and S. Matthew Weinberg. Pricing lotteries. *J. Economic Theory*, 156:144–174, 2015.

[BCWZ17] Johannes Brustle, Yang Cai, Fa Wu, and Mingfei Zhao. Approximating gains from trade in two-sided markets via simple mechanisms. In *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, pages 589–590, 2017.

[BGMM18] Moshe Babaioff, Yannai A. Gonczarowski, Yishay Mansour, and Shay Moran. Are two (samples) really better than one? In *Proceedings of the 2018 ACM Conference on Economics and Computation, Ithaca, NY, USA, June 18-22, 2018*, page 175, 2018.

[BH11] Xiaohui Bei and Zhiyi Huang. Bayesian Incentive Compatibility via Fractional Assignments. In *the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011.

[BILW20] Moshe Babaioff, Nicole Immorlica, Brendan Lucier, and S. Matthew Weinberg. A simple and approximately optimal mechanism for an additive buyer. *J. ACM*, 67(4):24:1–24:40, 2020.

[BKS18] Moshe Babaioff, Robert Kleinberg, and Aleksandrs Slivkins. Multi-parameter mechanisms with implicit payment computation. In *Proceedings of the Fourteenth ACM Conference on Electronic Commerce*, EC '13, page 35–52, New York, NY, USA, 2018. Association for Computing Machinery.

[BSV16] Maria-Florina Balcan, Tuomas Sandholm, and Ellen Vitercik. Sample complexity of automated mechanism design. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 2091–2099, Red Hook, NY, USA, 2016. Curran Associates Inc.

[BSV18] Maria-Florina Balcan, Tuomas Sandholm, and Ellen Vitercik. A general theory of sample complexity for multi-item profit maximization. In *Proceedings of the 2018 ACM Conference on Economics and Computation, Ithaca, NY, USA, June 18-22, 2018*, pages 173–174, 2018.

[Can20] Clément L. Canonne. A short note on learning discrete distributions. *arXiv preprint arXiv:2002.11457*, 2020. Version 3, revised April 8, 2020.

[CD17] Yang Cai and Constantinos Daskalakis. Learning multi-item auctions with (or without) samples. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 516–527, 2017.

[CDO+15] Xi Chen, Ilias Diakonikolas, Anthi Orfanou, Dimitris Paparas, Xiaorui Sun, and Mihalis Yannakakis. On the complexity of optimal lottery pricing and randomized mechanisms. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1464–1479, 2015.

[CDP+14] Xi Chen, Ilias Diakonikolas, Dimitris Paparas, Xiaorui Sun, and Mihalis Yannakakis. The complexity of optimal multidimensional pricing. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1319–1328, 2014.

[CDW16] Yang Cai, Nikhil Devanur, and S. Matthew Weinberg. A duality based unified approach to bayesian mechanism design. In *Proceedings of the 48th ACM Conference on Theory of Computation(STOC)*, 2016.

[CFPS21] Vincent Conitzer, Zhe Feng, David C. Parkes, and Eric Sodomka. Welfare-preserving -bic to bic transformation with negligible revenue loss. In *Web and Internet Economics: 17th International Conference, WINE 2021, Potsdam, Germany, December 14–17, 2021, Proceedings*, page 76–94, Berlin, Heidelberg, 2021. Springer-Verlag.

[CHK07] Shuchi Chawla, Jason D. Hartline, and Robert D. Kleinberg. Algorithmic Pricing via Virtual Valuations. In *the 8th ACM Conference on Electronic Commerce (EC)*, 2007.

[CHMS10] 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.

[CIL12] Shuchi Chawla, Nicole Immorlica, and Brendan Lucier. On the limits of black-box reductions in mechanism design. In *Proceedings of the 44th Symposium on Theory of Computing (STOC)*, 2012.

[CM88] Jacques Cremer and Richard P. McLean. Full extraction of the surplus in bayesian and dominant strategy auctions. *Econometrica*, 56(6):1247–1257, 1988.

[CMS15] Shuchi Chawla, David L. Malec, and Balasubramanian Sivan. The power of randomness in bayesian optimal mechanism design. *Games and Economic Behavior*, 91:297–317, 2015.

[COVZ19] Yang Cai, Argyris Oikonomou, Grigoris Velegkas, and Mingfei Zhao. An efficient ϵ-bic to BIC transformation and its application to black-box reduction in revenue maximization. *CoRR*, abs/1911.10172, 2019.

[CR14] Richard Cole and Tim Roughgarden. The sample complexity of revenue maximization. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 243–252, 2014.

[CWX25] Dimitar Chakarov, S. Matthew Weinberg, and Eric Xue. A qptas for up-to- revenue maximization with multiple constant-demand bidders over independent items. In *Proceedings of the 26th ACM Conference on Economics and Computation*, EC '25, page 946–974, New York, NY, USA, 2025. Association for Computing Machinery.

[CZ17] Yang Cai and Mingfei Zhao. Simple mechanisms for subadditive buyers via duality. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 170–183, 2017.

[DDT14] Constantinos Daskalakis, Alan Deckelbaum, and Christos Tzamos. The Complexity of Optimal Mechanism Design. In *the 25th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2014.

[DDT17] Constantinos Daskalakis, Alan Deckelbaum, and Christos Tzamos. Strong duality for a multiple-good monopolist. *Econometrica*, 85(3):735–767, 2017.

[DHKN17] 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, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 158–169, 2017.

[DHP16] Nikhil R. Devanur, Zhiyi Huang, and Christos-Alexandros Psomas. The sample complexity of auctions with side information. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 426–439, 2016.

[DRY15] Peerapong Dhangwatnotai, Tim Roughgarden, and Qiqi Yan. Revenue maximization with a single sample. *Games and Economic Behavior*, 91:318–333, 2015.

[DW12] Constantinos Daskalakis and S. Matthew Weinberg. Symmetries and optimal multidimensional mechanism design. In *Proceedings of the 13th ACM Conference on Electronic Commerce, EC 2012, Valencia, Spain, June 4-8, 2012*, pages 370–387, 2012.

[DZ20] Constantinos Daskalakis and Manolis Zampetakis. More revenue from two samples via factor revealing sdps. In *Proceedings of the 21st ACM Conference on Economics and Computation*, EC '20, page 257–272, New York, NY, USA, 2020. Association for Computing Machinery.

[EFF$^+$21] Alon Eden, Michal Feldman, Ophir Friedler, Inbal Talgam-Cohen, and S. Matthew Weinberg. A simple and approximately optimal mechanism for a buyer with complements. *Oper. Res.*, 69(1):188–206, 2021.

[FL20] Hu Fu and Tao Lin. Learning utilities and equilibria in non-truthful auctions. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

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

[GHZ19] Chenghao Guo, Zhiyi Huang, and Xinzhi Zhang. Settling the sample complexity of single-parameter revenue maximization. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019.*, pages 662–673, 2019.

[GJZ21] Wenshuo Guo, Michael I. Jordan, and Manolis Zampetakis. Robust learning of optimal auctions. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc.

[GLT20] Evangelia Gergatsouli, Brendan Lucier, and Christos Tzamos. The complexity of black-box mechanism design with priors. *ACM Trans. Econ. Comput.*, 8(4), October 2020.

[GN17] Yannai A. Gonczarowski and Noam Nisan. Efficient empirical revenue maximization in single-parameter auction environments. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 856–868, 2017.

[GW18] Yannai A. Gonczarowski and S. Matthew Weinberg. The sample complexity of up-to-ε multidimensional revenue maximization. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, 2018.

[HHSW21] Yihang Hu, Zhiyi Huang, Yiheng Shen, and Xiangning Wang. Targeting makes sample efficiency in auction design. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, EC '21, page 610–629, New York, NY, USA, 2021. Association for Computing Machinery.

[HKM11] Jason D. Hartline, Robert Kleinberg, and Azarakhsh Malekian. Bayesian incentive compatibility via matchings. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 734–747, 2011.

[HL10] Jason D. Hartline and Brendan Lucier. Bayesian algorithmic mechanism design. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 301–310, 2010.

[HN13] Sergiu Hart and Noam Nisan. The menu-size complexity of auctions. In *the 14th ACM Conference on Electronic Commerce (EC)*, 2013.

[HN17] Sergiu Hart and Noam Nisan. Approximate revenue maximization with multiple items. *J. Economic Theory*, 172:313–347, 2017.

[HR15] Sergiu Hart and Philip J. Reny. Maximizing Revenue with Multiple Goods: Nonmonotonicity and Other Observations. *Theoretical Economics*, 10(3):893–922, 2015.

[JL24] Yaonan Jin and Pinyan Lu. Benchmark-tight approximation ratio of simple mechanism for a unit-demand buyer. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 1251–1259. IEEE, 2024.

[JLX23] Yaonan Jin, Pinyan Lu, and Tao Xiao. Learning Reserve Prices in Second-Price Auctions. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 75:1–75:24, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[KMS+19] Pravesh Kothari, Divyarthi Mohan, Ariel Schvartzman, Sahil Singla, and S. Matthew Weinberg. Approximation schemes for a buyer with independent items via symmetries. In *the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2019.

[KO94] M. S. Keane and George L. O'Brien. A bernoulli factory. *ACM Trans. Model. Comput. Simul.*, 4(2):213–219, 1994.

[LY13] Xinye Li and Andrew Chi-Chih Yao. On revenue maximization for selling multiple independently distributed items. *Proceedings of the National Academy of Sciences*, 110(28):11232–11237, 2013.

[MR16] Jamie Morgenstern and Tim Roughgarden. Learning simple auctions. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 1298–1318, 2016.

[Mye81] Roger B. Myerson. Optimal Auction Design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[RS16] Tim Roughgarden and Okke Schrijvers. Ironing in the dark. In *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, pages 1–18, 2016.

[RW15] Aviad Rubinstein and S. Matthew Weinberg. Simple mechanisms for a subadditive buyer and applications to revenue monotonicity. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*, pages 377–394, 2015.

[RW18] Aviad Rubinstein and S. Matthew Weinberg. Simple mechanisms for a subadditive buyer and applications to revenue monotonicity. *ACM Trans. Economics and Comput.*, 6(3-4):19:1–19:25, 2018.

[Syr17] Vasilis Syrgkanis. A sample complexity measure with applications to learning optimal auctions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5358–5365, Red Hook, NY, USA, 2017. Curran Associates Inc.

[TW25] Yifeng Teng and Yifan Wang. *Learning Optimal Posted Prices for a Unit-Demand Buyer*, page 355. Association for Computing Machinery, New York, NY, USA, 2025.

[Yao15] Andrew Chi-Chih Yao. An n-to-1 bidder reduction for multi-item auctions and its applications. In *the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2015.

[Yao17] Andrew Chi-Chih Yao. Dominant-strategy versus bayesian multi-item auctions: Maximum revenue determination and comparison. In Constantinos Daskalakis, Moshe Babaioff, and Hervé Moulin, editors, *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, pages 3–20. ACM, 2017.

A Analysis: Meta-Framework

For the proofs below, whenever we condition on our surrogate samples $(\mathcal{S}_k)_k$, we also condition on the event that $\textsc{Flag}_{\textsc{Est}} = \texttt{True}$. Recall the notation $\kappa := \frac{R}{S}$ and that for a given bidder k and multiset of surrogates $\mathcal{S}_k$, the distribution $\hat{D}'_k$ is defined as the uniform distribution over $\mathcal{S}_k$.

A.1 Proof of BIC and IR

In this section we will prove our desired incentive guarantees, stated in Theorem A.1.

Lemma A.1 (BIC and IR). *The mechanism $\mathcal{M}$ defined by Algorithm 2 is BIC and IR over the distribution D.*

We start by proving the following claim, which captures the *stationarity* property of replica-surrogate matching [RW15, DHKN17].

Lemma A.2 (Stationarity). *If bidder k reports her type truthfully, then* $\textsc{MetaRS}$ *(Algorithm 1) outputs* $\textsc{Flag} = \texttt{True}$ *for k except with probability at most $\delta_{\textsc{Flag}}$, and conditioned on* $\textsc{Flag} = \texttt{True}$*, the distribution of the surrogate inputted for bidder k into $\hat{\mathcal{M}}'$ in Algorithm 2 is precisely $\hat{D}'_k$.*[38]

Proof of Theorem A.2. We prove the claim in two steps: we first bound the probability of $\textsc{Flag} = \texttt{True}$ and then reason about the distribution of the inputted surrogate conditioned on $\textsc{Flag} = \texttt{True}$.

Step 1: Bounding Flag probability. Consider the $\textsc{MetaRS}$ algorithm for bidder k. By Condition (C1), with probability at least $1 - \delta_{\textsc{Demand}}$, it holds that

$$\forall j \in [S], \sum_{r_i \in \mathcal{R}_k} \left[x^*_{ij}(\hat{\alpha}) + y^*_{ij}(\hat{\alpha})\right] \in (1 \pm \lambda) \cdot \kappa$$

where $(x^*_{ij}(\hat{\alpha}), y^*_{ij}(\hat{\alpha}))$ is the optimizer of $\mathbf{P}'_\delta(\mathcal{R}_k, \mathcal{S}_k, \hat{w}, \hat{\alpha})$. Here the probability is over the randomness of $\textsc{Preprocess}$, $\textsc{Draw-}\mathcal{R}$, $\textsc{Compute-}\alpha$, and drawing $t_k \sim D_k$.

Fix $\mathcal{R}_k$, $\mathcal{S}_k$, $\hat{\alpha}$ such the above condition (call it $\Phi_{\textsc{Demand}}$) holds, and consider the matching process in Step 4 of $\textsc{MetaRS}$.For any $s_j \in \mathcal{S}_k$, let $Z_i^{(j)}$ be the indicator random variable that is 1 iff replica r_i is matched to either surrogate s_j or dummy $\circ_j$ by the end of Step 4, where the randomness is over only the flip of $\textsc{Coin}_i$ and the realization of $\textsc{Match}$ (if $\textsc{Coin}_i = 1$). Observe that the random variables $\{Z_i^{(j)}\}_{i \in [R]}$ are all independent. Moreover, $\mathbb{E}[Z_i^{(j)}] = (1-\lambda)[x^*_{ij}(\hat{\alpha}) + y^*_{ij}(\hat{\alpha})]$. Letting $Z^{(j)} := \sum_{i \in [R]} Z_i^{(j)}$, the expectation $\mu := \mathbb{E}[Z^{(j)}]$ satisfies

$$\mu = \sum_{r_i \in \mathcal{R}_k} \mathbb{E}[Z_i^{(j)}] = (1-\lambda) \left(\sum_{r_i \in \mathcal{R}_k} [x^*_{ij}(\hat{\alpha}) + y^*_{ij}(\hat{\alpha})] \right) \in (1-\lambda) \cdot (1 \pm \lambda) \cdot \kappa$$

by condition $\Phi_{\textsc{Demand}}$. We can then apply the Chernoff bound to conclude that

$$\begin{aligned} \Pr\left[Z^{(j)} > \kappa\right] &\le \Pr\left[Z^{(j)} > (1-\lambda^4) \cdot \kappa\right] \\ &\le \Pr\left[Z^{(j)} > (1+\lambda^2) \cdot \mu\right] \end{aligned}$$

[38] The claims are over the randomness in the draw of bidder k's type from D_k, and the internal randomness of $\textsc{MetaRS}$ and its meta-inputs—in particular, *not* over the randomness of $\textsc{Draw-}\mathcal{S}$.

$$\leq \exp\left(\frac{-\lambda^4}{2+\lambda^2}\cdot\mu\right)$$
$$\leq \exp\left(\frac{-\lambda^4}{6}\cdot\kappa\right)$$

where the last line follows since $\mu \geq (1-2\lambda)\cdot\kappa$ and $\frac{1-2\lambda}{2+\lambda^2} \gg \frac{1}{6}$ since $\lambda \leq \frac{1}{4}$.

Finally, note that $\textsc{Flag} = \texttt{False}$ exactly when either $\textsc{Flag}_{\mathcal{R}} = \texttt{False}$ or there exists j such that $Z^{(j)} > \kappa$. Thus we conclude that, over all randomness of $\textsc{MetaRS}$,

$$\Pr[\textsc{Flag} = \texttt{False}] \leq \Pr[\textsc{Flag}_{\mathcal{R}} = \texttt{False}] + \Pr\left[\exists\, j \text{ such that } Z^{(j)} > \kappa \mid \Phi_{\textsc{Demand}}\right] + \Pr[\neg\Phi_{\textsc{Demand}}].$$

By Condition (B1), $\Pr[\textsc{Flag}_{\mathcal{R}} = \texttt{False}] \leq \delta_{\text{D-R}}$. It follows by a union bound over j that $\Pr[\textsc{Flag} = \texttt{False}] \leq \delta_{\text{D-R}} + \delta_{\textsc{Demand}} + S\cdot\exp\left(\frac{-\lambda^4}{6}\cdot\kappa\right) =: \delta_{\textsc{Flag}}$.

Step 2: Distribution of bidder k's surrogate. It remains to show that conditioned on $\textsc{Flag} = \texttt{True}$, the distribution of the surrogate inputted on behalf of bidder k into $\hat{\mathcal{M}}'$ (in Algorithm 2) is precisely $\hat{D}'_k$. When $\textsc{Flag} = \texttt{True}$, observe that the final matching obtained by the end of $\textsc{MetaRS}$ is a *perfect* κ-to-1 matching, when viewed as a matching from the replica index set $[R]$ to the surrogate index set $[S]$, such that i is matched to j if r_i is matched to s_j or $\circ_j$. Let r_{i^*} be the replica equal to the bidder's report. Recall from Condition (B2) that, over the randomness of $\textsc{Preprocess}$, $t_k \sim D_k$, and $\textsc{Draw-}\mathcal{R}$, $\Pr[r_{i^*} = x \mid \textsc{Flag}_{\mathcal{R}} = \texttt{True}] = \Pr[\textsc{Unif}(\mathcal{R}_k) = x \mid \textsc{Flag}_{\mathcal{R}} = \texttt{True}]$. It is not hard to see by inspection of $\textsc{MetaRS}$ that this remains true over the full randomness of $\textsc{MetaRS}$ and conditioned on $\textsc{Flag} = \texttt{True}$.

In particular, this means that the distribution of the surrogate index $j \in [S]$ matched to the bidder's replica is identical to the distribution of the surrogate index matched to a *uniformly random* replica. Since the matching is perfect, this distribution is the uniform distribution over the surrogate indices. It follows by inspection of Algorithm 2 that the distribution of the surrogate inputted into $\hat{\mathcal{M}}'$ for bidder k is given by the uniform distribution over $\mathcal{S}_k$, i.e. $\hat{D}'_k$. □

We now prove Theorem A.1 using Theorem A.2.

Proof of Theorem A.1. Consider bidder k with true type $t_k \in \mathcal{T}_k$ and consider the mechanism $\mathcal{M}$ defined by Algorithm 2. We must show that

$$\mathop{\mathbb{E}}_{\substack{t_{\text{-}k}\sim D_{\text{-}k}\\ \mathcal{M}}}[u_k(t_k, \mathcal{M}(t_k; t_{\text{-}k}))] \geq 0 \qquad \textbf{(IR)}$$

and that for any $t'_k \in \mathcal{T}_k$,

$$\mathop{\mathbb{E}}_{\substack{t_{\text{-}k}\sim D_{\text{-}k}\\ \mathcal{M}}}[u_k(t_k, \mathcal{M}(t_k; t_{\text{-}k}))] \geq \mathop{\mathbb{E}}_{\substack{t_{\text{-}k}\sim D_{\text{-}k}\\ \mathcal{M}}}[u_k(t_k, \mathcal{M}(t'_k; t_{\text{-}k}))]. \qquad \textbf{(BIC)}$$

For any type $t'_k \in \mathcal{T}_k$, let $\Phi_{\textsc{Keep}}(t'_k)$ denote the event, over the randomness of $t_{\text{-}k} \sim D_{\text{-}k}$ and internal randomness of $\mathcal{M}$, that $\textsc{Coin}^{(k)} = \texttt{True}$ and $\textsc{Flag}^{(k')} = \texttt{True}$ for all $k' \neq k \in [n]$ in Algorithm 2, when bidder k reports t'_k. Since bidder k's utility is 0 under $\neg\Phi_{\textsc{Keep}}(t'_k)$ by Algorithm 2 (Step 5), we can write

$$\mathbb{E}[u_k(t_k, \mathcal{M}(t'_k; t_{\text{-}k}))] = \Pr\left[\Phi_{\textsc{Keep}}(t'_k)\right]\cdot\mathbb{E}[u_k(t_k, \mathcal{M}(t'_k; t_{\text{-}k})) \mid \Phi_{\textsc{Keep}}(t'_k)] + \Pr\left[\neg\Phi_{\textsc{Keep}}(t'_k)\right]\cdot 0$$

$$= \Pr\left[\Phi_{\text{Keep}}(t'_k)\right] \cdot \mathbb{E}[u_k(t_k, \mathcal{M}(t'_k; t_{\text{-}k})) \mid \Phi_{\text{Keep}}(t'_k)]$$

It is straightforward to see, by construction of MetaRS, that event $\Phi_{\text{Keep}}(t'_k)$ in fact does *not* depend on the reported bidder type, i.e. $\Pr[\Phi_{\text{Keep}}(t'_k)] = \Pr[\Phi_{\text{Keep}}(t''_k)]$ for all t'_k, t''_k. Thus it suffices to show that (a) $\mathbb{E}[u_k(t_k, \mathcal{M}(t_k; t_{\text{-}k})) \mid \Phi_{\text{Keep}}] \geq 0$ for IR, and (b) $t_k \in \arg\max_{t'_k \in \mathcal{T}_k} \mathbb{E}[u_k(t_k, \mathcal{M}(t'_k; t_{\text{-}k})) \mid \Phi_{\text{Keep}}]$ for BIC. We will show stronger statements by conditioning on the particular outputs $(\mathcal{S}_{k'})_{k' \in [n]}$ of Draw-$\mathcal{S}$ in Step 1 of Algorithm 2:

$$\mathbb{E}[u_k(t_k, \mathcal{M}(t_k; t_{\text{-}k})) \mid \Phi_{\text{Keep}}, (\mathcal{S}_{k'})_{k' \in [n]}] \geq 0 \qquad \text{(IR)}$$

$$t_k \in \arg\max_{t'_k \in \mathcal{T}_k} \mathbb{E}[u_k(t_k, \mathcal{M}(t'_k; t_{\text{-}k})) \mid \Phi_{\text{Keep}}, (\mathcal{S}_{k'})_{k' \in [n]}]. \qquad \text{(BIC)}$$

For each $k' \neq k \in [n]$, consider the random variable $s^{(k')}$ denoting the surrogate inputted into $\hat{\mathcal{M}}'$ for bidder k' (Algorithm 2, Step 4). Observe that $s^{(k')}$ is independent of the event Φ'_{Keep}, where Φ'_{Keep} is defined as the event that $\text{Coin}^{(k)} = \texttt{True}$ and $\text{Flag}^{(k'')} = \texttt{True}$ for all $k'' \notin \{k, k'\}$, such that $\Phi_{\text{Keep}} = \Phi'_{\text{Keep}} \cap (\text{Flag}^{(k')} = \texttt{True})$. It now follows by stationarity (Theorem A.2) that the distribution of each $s^{(k')}$, over the randomness of other bidders truthfully reporting $t_{\text{-}k} \sim D_{\text{-}k}$ and $\mathcal{M}$, conditioned on Φ_{Keep} and the particular surrogate sets $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$, is precisely $\hat{D}'_{k'}$.

In particular, this means that bidder k's *value* for being matched to a real surrogate $s_j \in \mathcal{S}_k$ (conditioned on the aforementioned events) is exactly given by the edge weight

$$\hat{w}_{t_k, j} = \mathop{\mathbb{E}}_{t_{\text{-}k} \sim \hat{D}'_{\text{-}k}} [v_k(t_k, \hat{\mathcal{A}}'(s_j; t_{-k})) - (1 - \beta_k) \cdot \hat{p}'(s_j; t_{-k})],$$

recalling the procedure for assigning allocation and Phase 2 payment in Algorithm 2 (Step 5). That is, the expectation $\mathbb{E}_{t_{\text{-}k} \sim \hat{D}'_{\text{-}k}}[\cdot]$ correctly captures the fact that the surrogate participating on bidder k' is distributed according to $\hat{D}'_{k'}$, for all $k' \neq k \in [n]$. Additionally, note that bidder k's value for being matched to a dummy surrogate $\circ_j$ is clearly 0 (Step 5).

Finally, observe that conditioned on Φ_{Keep} and sets $(\mathcal{S}_{k'})_{k' \in [n]}$, if bidder k reports some type t'_k, then she is matched to real surrogate s_j with probability $x^*_{t'_k j}(\hat{\alpha})$ and to dummy surrogate $\circ_j$ with probability $y^*_{t'_k j}(\hat{\alpha})$ by Condition (D1), where $x^*_{t'_k j}(\hat{\alpha})$ and $y^*_{t'_k j}(\hat{\alpha})$ are defined as in Theorem 4.6 for surrogates $(\mathcal{S}_{k'})_{k' \in [n]}$, duals $\hat{\alpha}$, and type t'_k. Thus bidder k's utility $\mathbb{E}[u_k(t_k, \mathcal{M}(t'_k; t_{\text{-}k})) \mid \Phi_{\text{Keep}}, (\mathcal{S}_{k'})_{k' \in [n]}]$ for reporting type t'_k is given by $\sum_j x^*_{t'_k j}(\hat{\alpha}) \cdot \hat{w}_{t_k j} - \mathbb{E}[q_k]$[39], where q_k is the Phase 1 payment under Φ_{Keep}. First, Condition (E2) guarantees that this expression is non-negative when $t'_k = t_k$, which proves the IR claim. Next, Condition (E1) guarantees that

$$\begin{aligned}
&\sum_j x^*_{t'_k j}(\hat{\alpha}) \cdot \hat{w}_{t_k j} - \mathbb{E}[q_k] \\
&= \sum_j x^*_{t'_k j}(\hat{\alpha}) \cdot \hat{w}_{t_k j} - \delta \sum_j (x^*_{t'_k j}(\hat{\alpha}) \log x^*_{t'_k j}(\hat{\alpha}) + y^*_{t'_k j}(\hat{\alpha}) \log y^*_{t'_k j}(\hat{\alpha})) - \sum_j \alpha_j (x^*_{t'_k j}(\hat{\alpha}) + y^*_{t'_k j}(\hat{\alpha})) - C.
\end{aligned}$$

Since C is a constant that does not depend on the report t'_k, it now follows from Theorem 4.6 that $t_k \in \arg\max_{t'_k \in \mathcal{T}_k} \mathbb{E}[u_k(t_k, \mathcal{M}(t'_k; t_{\text{-}k})) \mid \Phi_{\text{Keep}}, (\mathcal{S}_{k'})_{k' \in [n]}]$, which proves the BIC claim. ☐

[39] Importantly, while the x^* and y^* are defined in terms of edge weights based on the reported type t'_k, bidder k's actual values for surrogates are always given by the edge weights $\hat{w}_{t_k j}$ based on her true type t_k.

A.2 Proof of Revenue Approximation

We now prove the revenue approximation claim in Theorem 5.1, restated separately below. The revenue loss term Δ_{Rev} is given by $\Delta_{\text{D-S}} + \sum_{k=1}^{n} \Delta_{\text{Rev}}^{(k)}$, where $\Delta_{\text{Rev}}^{(k)}$ depends on the variety of parameters involved in the meta-framework:

$$\begin{aligned}\Delta_{\text{Rev}}^{(k)} &:= \beta_k + (n+1)\delta_{\text{Flag}} + \delta_{\text{D-S}} + \Delta_{\text{Match}} - \min(q_{\text{LB}}, 0) \\ &\quad + \frac{1}{\beta_k} \cdot (\Delta_{\text{MWM}} + 2\varepsilon + 2\varepsilon_{\text{D-S}} + 2\delta_{\text{D-S}} + 3d + 2\Delta_{\text{Couple}}^{(k)} + 3\delta_{\text{Flag}})\end{aligned}$$

Lemma A.3 (Meta-Framework: Revenue Approximation)**.** *The mechanism* $\mathcal{M} = (\mathcal{A}, p)$ *defined by Algorithm 2 achieves revenue* $\text{Rev}(\mathcal{M}; D) \geq \text{Rev}(\mathcal{M}'; D') - \Delta_{\text{Rev}}$.

The proof will be similar in structure to prior revenue approximation proofs for replica-surrogate matching schemes, particularly that of [COVZ19], but the extra machinery in our full mechanism $\mathcal{M}$ (among other changes) will lead to different details.

Recall that the revenue $\text{Rev}(\mathcal{M}^*, D^*)$ of a mechanism $\mathcal{M}^* = (\mathcal{A}^*, p^*)$ over distribution D^* is $\text{Rev}(M^*, D^*) := \mathbb{E}_{t\sim D^*}[\sum_{k\in[n]} p_k^*(t)]$. For our mechanism $\mathcal{M}$, we define $\text{Rev}_k^{(1)}(\mathcal{M}; D)$ and $\text{Rev}_k^{(2)}(\mathcal{M}; D)$ to be the expected contributions from bidder k to the revenue based on Phase 1 and Phase 2 payments, respectively, such that

$$\text{Rev}(\mathcal{M}; D) = \sum_{k\in[n]} \text{Rev}_k^{(1)}(\mathcal{M}; D) + \text{Rev}_k^{(2)}(\mathcal{M}; D).$$

Observe that in Algorithm 2, for the Phase 1 payment, each bidder k is either charged 0 or the output q_k of running Payment. Condition (E3) therefore guarantees that $\text{Rev}_k^{(1)}(\mathcal{M}; D) \geq \min(0, q_{\text{LB}})$. Thus it will suffice to bound

$$\sum_{k\in[n]} \text{Rev}_k^{(2)}(\mathcal{M}; D) \geq \text{Rev}(\mathcal{M}'; D') - (\Delta_{\text{Rev}} + n \cdot \min(0, q_{\text{LB}})).$$

Notation. For much of the proof, we will view as fixed the surrogate sets $\mathcal{S} := (\mathcal{S}_k)_{k\in[n]}$ returned by calling Draw-$\mathcal{S}$ for each bidder k in Algorithm 2 (Step 1). Whenever we write $\text{val}^{(k)}(t_1, t_2)$ in the analysis, we mean the *interim value* with respect to $\hat{\mathcal{M}}' = (\hat{\mathcal{A}}', \hat{p}')$ and the distribution $\hat{D}'$:

$$\text{val}^{(k)}(t_1, t_2) = \mathop{\mathbb{E}}_{t_{\text{-}k}\sim \hat{D}'_{\text{-}k}} [v_k(t_1, \hat{\mathcal{A}}'(t_2, t_{\text{-}k}))].$$

Similarly, by $\hat{p}'^{(k)}(t)$ we mean the *interim payment* according to $\hat{p}'$ *over* $\hat{D}'$, such that

$$\text{Rev}(\hat{\mathcal{M}}'; \hat{D}') = \mathop{\mathbb{E}}_{t\sim \hat{D}'}\left[\sum_{k\in[n]} p_k'(t)\right] = \mathop{\mathbb{E}}_{t\sim \hat{D}'}\left[\sum_{k\in[n]} \hat{p}'^{(k)}(t_k)\right]$$

where $t =: (t_1, \ldots, t_n)$. The contribution from bidder k is $\text{Rev}_k(\hat{\mathcal{M}}'; \hat{D}') := \mathbb{E}_{t_k\sim \hat{D}'_k}[\hat{p}'^{(k)}(t_k)]$. Finally, $\text{Rev}(\mathcal{M}; D)|\mathcal{S}$ denotes the expected revenue of $\mathcal{M}$ over D *conditioned* on the particular draw of surrogate sets $\mathcal{S}$, and $\text{Rev}_k^{(2)}(\mathcal{M}; D)|\mathcal{S}$ is defined analogously.

Fix a particular bidder k, with surrogates $\mathcal{S}_k$ and replicas $\mathcal{R}_k$. We will use Φ_{F}^k to denote the event that the $\textsc{Flag}^{(k)}$ returned by $\textsc{MetaRS}$ is `True`.[40] We will usually view $(\mathcal{R}_k, \mathcal{S}_k)$ as a bipartite graph from replicas to surrogates. Let $O(\mathcal{R}_k, \mathcal{S}_k)$ denote the set of matches from replicas to *real* surrogates obtained by $\textsc{MetaRS}$ (that is, dropping all matches to dummy surrogates). In general, $O(\mathcal{R}_k, \mathcal{S}_k)$ is a set of edges on $(\mathcal{R}_k, \mathcal{S}_k)$ where each replica is the endpoint of at most one edge; conditioned on Φ_{F}^k, O is a *feasible κ-to-1 matching.*

For convenience in the upcoming analysis, we will think of our bipartite graph as having the same number of left and right nodes by imagining $\kappa = \frac{R}{S}$ copies of each surrogate $s \in \mathcal{S}_k$, to obtain the *copied-surrogates* $\bar{\mathcal{S}}_k = (\underbrace{\mathcal{S}_k, \ldots, \mathcal{S}_k}_{\kappa \text{ times}})$. We then view the edges $O(\mathcal{R}_k, \mathcal{S}_k)$ as edges $O(\mathcal{R}_k, \bar{\mathcal{S}}_k)$ on $(\mathcal{R}_k, \mathcal{S}_k)$ in the natural way, where edges to a surrogate $s_j \in \mathcal{S}_k$ are distributed among the copies of s_j in $\bar{\mathcal{S}}_k$ to minimize the maximum degree of any copy of s_j. Observe that conditioned on Φ_{F}^k, $O(\mathcal{R}_k, \bar{\mathcal{S}}_k)$ is a *feasible* 1*-to-*1 *matching.* We henceforth consider the graph $(\mathcal{R}_k, \bar{\mathcal{S}}_k)$ in this section, and so sometimes abbreviate $O := O(\mathcal{R}_k, \bar{\mathcal{S}}_k)$ (and refer to $\bar{\mathcal{S}}_k$ as just "the surrogates $\bar{\mathcal{S}}_k$") when the meaning is clear from context. For any set of edges E on $(\mathcal{R}_k, \bar{\mathcal{S}}_k)$, we will write $s_j \in E$ to mean that a surrogate s_j is the endpoint of an edge in E. In a sum $\sum_{s_j \in E}(\cdots)$, s_j should be enumerated with the appropriate multiplicity if it is the endpoint of multiple edges in E.

Proof of Theorem A.3. The proof is built up through a series of technical lemmas, which we present in steps below. To start, fix the surrogate sets $\mathcal{S} = (\mathcal{S}_k)_{k \in [n]}$ from $\textsc{Draw}$-$\mathcal{S}$ and fix a bidder k.

Step 1. Revenue expressed as surrogate payments.

Claim 1 (Adapted from [COVZ19], Lemma 21)**.** *$\textsc{Rev}_k(\hat{\mathcal{M}}'; \hat{D}')$ and $\textsc{Rev}_k^{(2)}(\mathcal{M}; D)|\mathcal{S}$ satisfy the following claims:*

$$\textsc{Rev}_k(\hat{\mathcal{M}}'; \hat{D}') = \frac{1}{R} \cdot \sum_{s_j \in \bar{\mathcal{S}}_k} \hat{p}'^{(k)}(s_j)$$

$$\textsc{Rev}_k^{(2)}(\mathcal{M}; D)|\mathcal{S} \geq (1 - \beta_k) \mathop{\mathbb{E}}_{\textsc{MetaRS}} \left[\frac{1}{R} \cdot \sum_{s_j \in O(\mathcal{R}_k, \bar{\mathcal{S}}_k)} \hat{p}'^{(k)}(s_j) \mid \Phi_{\mathrm{F}}^k \right] - n \cdot \delta_{\textsc{Flag}}$$

where the expectation is over the draw of bidder k's type from D_k and all internal randomness of $\textsc{MetaRS}$ *and its meta-inputs.*

We reiterate that the surrogates sets $\mathcal{S}$ are viewed as non-random for this claim and proof.

Proof of Claim 1. The claim on $\textsc{Rev}_k(\hat{\mathcal{M}}'; \hat{D}')$ is seen as follows:

$$\begin{aligned}
\textsc{Rev}_k(\hat{\mathcal{M}}'; \hat{D}') &= \mathop{\mathbb{E}}_{t \sim \hat{D}'_k} [\hat{p}'^{(k)}(t_k)] \\
&= \frac{1}{S} \sum_{s_j \in \mathcal{S}_k} \hat{p}'^{(k)}(s_j) && \text{definition of } \hat{D}'_k \\
&= \frac{1}{R} \cdot \sum_{s_j \in \bar{\mathcal{S}}_k} \hat{p}'^{(k)}(s_j) && \text{definition of } \bar{\mathcal{S}}
\end{aligned}$$

[40] The event Φ_{F}^k, as in the proof of stationarity, is defined over the randomness in the draw of bidder k's types and the internal randomness of $\textsc{MetaRS}$ and its meta-inputs, *not* the draw of surrogates $\mathcal{S}_k$.

Next we analyze $\textsc{Rev}_k^{(2)}(\mathcal{M}; D)|\mathcal{S}$. We abbreviate $\textsc{Rev}_{k,\mathcal{S}}^{(2)} := \textsc{Rev}_k^{(2)}(\mathcal{M}; D)|\mathcal{S}$ for convenience. Define event Φ_{F} as the event where $\textsc{Flag}^{(k')} = \texttt{True}$ for *all* bidders $k' \in [n]$, i.e. $\Phi_{\mathrm{F}} = \bigcap_{k' \in [n]} \Phi_{\mathrm{F}}^{k'}$. By Theorem A.2, $\Pr[\Phi_{\mathrm{F}}] \geq 1 - n \cdot \delta_{\textsc{Flag}}$. Thus we can write

$$\begin{aligned}
\textsc{Rev}_{k,\mathcal{S}}^{(2)} &= \Pr\left[\Phi_{\mathrm{F}}\right] \cdot \left(\textsc{Rev}_{k,\mathcal{S}}^{(2)} \mid \Phi_{\mathrm{F}}\right) + \Pr\left[\neg\Phi_{\mathrm{F}}\right] \cdot \left(\textsc{Rev}_{k,\mathcal{S}}^{(2)} \mid \neg\Phi_{\mathrm{F}}\right) \\
&\geq (1 - n \cdot \delta_{\textsc{Flag}}) \left(\textsc{Rev}_{k,\mathcal{S}}^{(2)} \mid \Phi_{\mathrm{F}}\right) && \left(\textsc{Rev}_{k,\mathcal{S}}^{(2)} \mid \neg\Phi_{\mathrm{F}}\right) \geq 0 \\
&\geq \left(\textsc{Rev}_{k,\mathcal{S}}^{(2)} \mid \Phi_{\mathrm{F}}\right) - n \cdot \delta_{\textsc{Flag}} && \left(\textsc{Rev}_{k,\mathcal{S}}^{(2)} \mid \neg\Phi_{\mathrm{F}}\right) \leq 1
\end{aligned}$$

where $\textsc{Rev}_{k,\mathcal{S}}^{(2)} \mid \Phi_{\mathrm{F}}$ denotes the expected contribution from bidder k to $\textsc{Rev}(\mathcal{M}; D)$ in Phase 2, conditioned on $\mathcal{S}$ and Φ_{F}^k, and $\textsc{Rev}_{k,\mathcal{S}}^{(2)} \mid \neg\Phi_{\mathrm{F}}$ is defined analogously.

Consider $\textsc{Rev}_{k,\mathcal{S}}^{(2)} \mid \Phi_{\mathrm{F}}$. Under event Φ_{F}^k, $\textsc{Flag}^{(k')} = \texttt{True}$ for all $k' \in [n]$, which implies the following:

1. For Phase 2 payment, bidder k will pay $(1-\beta_k)$ times the payment under $\hat{\mathcal{M}}'$ of the surrogate she is matched to, if any, under the matching O produced by $\textsc{MetaRS}$.
2. Since bidder k's type can be viewed as a uniformly random replica under $\textsc{Flag}^{(k)} = \texttt{True}$ (by the same reasoning as in the proof of Theorem A.2), each of the surrogates $s_j \in O$ is selected with probability $\frac{1}{R}$ as the surrogate used to calculate the Phase 2 payment for bidder k.
3. For all $k' \neq k$, by Theorem A.2, the surrogate $s^{(k')}$ inputted on behalf of bidder k' into $\hat{\mathcal{M}}'$ (in Step 4 of Algorithm 2) is distributed according to $\hat{D}'_{k'}$.

   In particular, the expected payment of bidder k's inputted surrogate—over all randomness of deciding all *other* bidders' inputted surrogates—is the *interim payment* $\hat{p}'^{(k)}(\cdot)$ over $\hat{D}'$.

Putting these facts together, we can write $\textsc{Rev}_{k,\mathcal{S}}^{(2)} \mid \Phi_{\mathrm{F}}$ as

$$\textsc{Rev}_{k,\mathcal{S}}^{(2)} \mid \Phi_{\mathrm{F}} = \mathbb{E}\left[\frac{1}{R} \cdot \sum_{s_j \in O(\mathcal{R}_k, \bar{\mathcal{S}}_k)} (1 - \beta_k) \cdot \hat{p}'^{(k)}(s_j) \mid \Phi_{\mathrm{F}}\right]$$

where the expectation is over the draw of bidder k's type from D_k and all internal randomness in $\textsc{MetaRS}$ and its meta-inputs. Observe that the expression $\sum_{s_j \in O(\mathcal{R}_k, \bar{\mathcal{S}}_k)} (1 - \beta_k) \cdot \hat{p}'^{(k)}(s_j)$ is independent of $\Phi_{\mathrm{F}}^{k'}$ for all $k' \neq k \in [n]$. In particular, we can conclude

$$\textsc{Rev}_{k,\mathcal{S}}^{(2)} \mid \Phi_{\mathrm{F}} = (1 - \beta_k) \cdot \mathbb{E}\left[\frac{1}{R} \cdot \sum_{s_j \in O(\mathcal{R}_k, \bar{\mathcal{S}}_k)} \hat{p}'^{(k)}(s_j) \mid \Phi_{\mathrm{F}}^k\right]$$

which upon substitution completes the proof. □

Step 2. Construct high-cardinality matching $M^*(\mathcal{R}_k, \bar{\mathcal{S}}_k, c_{k,\mathcal{G}}(\mathcal{R}_k))$. For Steps 2 and 3, we fix the output $\mathcal{G}$ of $\textsc{Preprocess}$, the replicas $\mathcal{R}_k$ from $\textsc{Draw}$-$\mathcal{R}$, and the outputs of the rest of $\textsc{MetaRS}$ *such that event Φ_{F}^k holds.* In particular, we consider the resulting $O := O(\mathcal{R}_k, \bar{\mathcal{S}}_k)$ as a feasible, 1-to-1 matching. Furthermore, we fix a realization of the couples $c_{k,\mathcal{G}}(\mathcal{R}_k) =$

$(c_{k,\mathcal{G}}(r_1), \ldots, c_{k,\mathcal{G}}(r_R))$, where $c_{k,\mathcal{G}}$ is the coupling guaranteed by the meta-framework (Section 5.3) corresponding to $\mathcal{G}$.

Then let $M := M(\mathcal{R}_k, \bar{\mathcal{S}}_k, c_{k,\mathcal{G}}(\mathcal{R}_k))$ be the maximum-cardinality, 1-to-1 matching on $(\mathcal{R}_k, \bar{\mathcal{S}}_k)$ that matches some r_i to some s_j only if $c_{k,\mathcal{G}}(r_i)$ and s_j are d-close. We now define a modified version of M, denoted M^*.

Definition A.4 (Modified matching M^* [COVZ19])**.** Suppose Φ^k_{F} holds, so that O is a matching. Consider the union of matchings O and M, which is a set of alternating paths and cycles, and let $\mathcal{P}$ be the set of alternating paths with an endpoint that is a surrogate matched in M but not in O. Then the matching $M^* := M^*(\mathcal{R}_k, \bar{\mathcal{S}}_k, c_{k,\mathcal{G}}(\mathcal{R}_k))$ is defined by starting from O and then, for each path $P \in \mathcal{P}$, replacing all edges from O in P with the edges from M in P.

We now bound the expected payment of surrogates matched under M^* in terms of $\textsc{Rev}_k(\hat{\mathcal{M}}'; \hat{D}')$[41] and the cardinality $|M|$ of M. It will then suffice to focus our analysis on comparing the matchings O and M^*.

Claim 2 (Adapted from [COVZ19], Corollary 3)**.** *Suppose Φ^k_{F} holds, so that M^* is well-defined. Then*

$$\frac{1}{R} \cdot \sum_{s_j \in M^*} \hat{p}'^{(k)}(s_j) \geq \textsc{Rev}_k(\hat{\mathcal{M}}'; \hat{D}') - (1 - |M|/R).$$

Proof of Claim 2. The key observation is that $|M^*| \geq |M|$. To see this, consider the set $\mathcal{P}$ as defined in Theorem A.4. For any surrogate s matched in M on an alternating path $P \in \mathcal{P}$, s is also matched by M^*. For any surrogate s matched in M^* *outside of* alternating paths in $\mathcal{P}$, observe that s must be matched by O (by definition of $\mathcal{P}$) and hence by M^* as well. Thus M^* matches at least as many surrogates as M.

We can then derive as follows, where $\bar{\mathcal{S}}_k \setminus M^*$ denotes the surrogates left unmatched by M^*.

$$\begin{aligned}
\frac{1}{R} \cdot \sum_{s_j \in M^*} \hat{p}'^{(k)}(s_j) &= \frac{1}{R} \cdot \sum_{s_j \in \bar{\mathcal{S}}_k} \hat{p}'^{(k)}(s_j) - \frac{1}{R} \cdot \sum_{s_j \in \bar{\mathcal{S}}_k \setminus M^*} \hat{p}'^{(k)}(s_j) \\
&\geq \frac{1}{R} \cdot \sum_{s_j \in \bar{\mathcal{S}}_k} \hat{p}'^{(k)}(s_j) - \frac{1}{R} \cdot (R - |M^*|) \cdot 1 && \hat{p}'^{(k)}(\cdot) \leq 1 \\
&\geq \frac{1}{R} \cdot \sum_{s_j \in \bar{\mathcal{S}}_k} \hat{p}'^{(k)}(s_j) - \frac{1}{R} \cdot (R - |M|) && |M^*| \geq |M| \\
&= \textsc{Rev}(\hat{\mathcal{M}}'; \hat{D}') - (1 - |M|/R) && \text{Claim 1}
\end{aligned}$$

□

Step 3. Compare surrogates matched in O and M^*. We now analyze the difference between surrogates matched in O and M^* by reasoning about the alternating paths in $\mathcal{P}$. For an edge between $r_i \in \mathcal{R}_k$ and $s_j \in \bar{\mathcal{S}}_k$, let the weight of that edge be $\hat{w}(r_i, s_j) = \hat{w}_{ij}$ as defined in Theorem 4.1, with respect to the mechanism $\hat{\mathcal{M}}'$ and the distribution $\hat{D}'$, and then let $W(E)$ denote the total weight of a matching E. We also define $\hat{\varepsilon} \geq 0$ to be the smallest value such that $\hat{\mathcal{M}}'$ is $\hat{\varepsilon}$-BIC over $\hat{D}'$.[42]

[41] Note that, once $\mathcal{S}$ is fixed, $\textsc{Rev}_k(\hat{\mathcal{M}}'; \hat{D}')$ is a non-random quantity.

[42] Like $\textsc{Rev}_k(\hat{\mathcal{M}}'; \hat{D}')$, while $\hat{\varepsilon}$ is a random variable over the randomness of $\textsc{Draw}$-$\mathcal{S}$, it is fixed here for fixed $\mathcal{S}$.

Claim 3. *Let* $\textsc{Opt} := \textsc{Opt}(\mathbf{P}(\mathcal{R}_k, \mathcal{S}_k, \hat{w}))$ *denote the weight of the maximum-weight, κ-to-1 matching on* $(\mathcal{R}_k, \mathcal{S}_k)$ *for weights* $\hat{w} = (\hat{w}_{ij})_{i,j}$ *(Theorem 4.2). Then*

$$\frac{1}{R}\cdot\sum_{s_j\in O}\hat{p}'^{(k)}(s_j) \geq \frac{1}{R}\cdot\sum_{s_j\in M^*}\hat{p}'^{(k)}(s_j) - \frac{1}{R\beta_k}\left(\textsc{Opt} - W(O)\right) - \frac{2}{R\beta_k}\left(\sum_{r_i\in\mathcal{R}_k} dist_k(r_i, c_{k,\mathcal{G}}(r_i))\right) - \frac{1}{\beta_k}\cdot(2\hat{\varepsilon}+3d).$$

The proof of Claim 3 follows the same derivation as in [COVZ19] (part of the proof of their Lemma 26).

Proof of Claim 3. Consider an alternating path $P \in \mathcal{P}$ (Definition A.4). P must have one of two forms: **(a)** $(s_{(1)}, r_{(1)}, s_{(2)}, \ldots, r_{(c)}, s_{(c+1)})$, where $s_{(1)} \in M^* \setminus O$ and $s_{(c+1)} \in O \setminus M^*$; or **(b)** $(s_{(1)}, r_{(1)}, s_{(2)}, \ldots, s_{(c)}, r_{(c)})$, where $s_{(1)} \in M^* \setminus O$.[43] In either case, any edge $s_{(i)} \leftrightarrow r_{(i)}$ is an edge in M^* and in M, so in particular, $c_{k,\mathcal{G}}(r_{(i)})$ and $s_{(i)}$ must be d-close.

Consider any $i \in [c]$ for type (a) paths, or $i \in [c-1]$ for type (b) paths. Since $\mathcal{M}'$ is $\hat{\varepsilon}$-BIC over $\hat{D}'$ by Condition (A1),

$$\mathrm{val}^{(k)}(s_{(i)}, s_{(i)}) - \hat{p}'^{(k)}(s_{(i)}) \geq \mathrm{val}^{(k)}(s_{(i)}, s_{(i+1)}) - \hat{p}'^{(k)}(s_{(i+1)}) - \hat{\varepsilon}.$$

Since $\mathrm{dist}_k(c_{k,\mathcal{G}}(r_{(i)}), s_{(i)}) = \max_{o\in O}|v_k(c_{k,\mathcal{G}}(r_{(i)}), o) - v_k(s_{(i)}, o)| \leq d$, for any type t' it holds that $|\mathrm{val}^{(k)}(c_{k,\mathcal{G}}(r_{(i)}), t') - \mathrm{val}^{(k)}(s_{(i)}, t')| \leq d$. Thus

$$\mathrm{val}^{(k)}(c_{k,\mathcal{G}}(r_{(i)}), s_{(i)}) - \hat{p}'^{(k)}(s_{(i)}) \geq \mathrm{val}^{(k)}(c_{k,\mathcal{G}}(r_{(i)}), s_{(i+1)}) - \hat{p}'^{(k)}(s_{(i+1)}) - \hat{\varepsilon} - 2d.$$

Since weight $\hat{w}(r_i, s_j)$ is defined as $\hat{w}(r_i, s_j) = \mathrm{val}^{(k)}(r_i, s_j) - (1-\beta_k)\hat{p}'^{(k)}(s_j)$, we can write

$$\begin{aligned}\hat{w}(r_{(i)}, s_{(i)}) - \hat{w}(r_{(i)}, s_{(i+1)}) \geq -\hat{\varepsilon} - 2d - \beta_k\left(\hat{p}'^{(k)}(s_{(i+1)}) - \hat{p}'^{(k)}(s_{(i)})\right) \\ + \Delta_k(r_{(i)}, s_{(i)}) - \Delta_k(r_{(i)}, s_{(i+1)})\end{aligned}$$

where $\Delta_k(r_i, s_j) := \mathrm{val}^{(k)}(r_i, s_j) - \mathrm{val}^{(k)}(c_{k,\mathcal{G}}(r_i), s_j)$ captures a difference in (interim) values between a replica and its couple. We now sum our weight difference terms over the whole path P. When P is of form (a),

$$\begin{aligned}&\sum_{i=1}^{c}\left(\hat{w}(r_{(i)}, s_{(i+1)}) - \hat{w}(r_{(i)}, s_{(i)})\right) \\ &\leq c\cdot(2d+\hat{\varepsilon}) + \beta_k\left(\hat{p}'^{(k)}(s_{(c+1)}) - \hat{p}'^{(k)}(s_{(1)})\right) + \sum_{i=1}^{c}\left(\Delta_k(r_{(i)}, s_{(i+1)}) - \Delta_k(r_{(i)}, s_{(i)})\right).\end{aligned}$$

When P is of form (b),

$$\sum_{i=1}^{c-1}\left(\hat{w}(r_{(i)}, s_{(i+1)}) - \hat{w}(r_{(i)}, s_{(i)})\right) - \hat{w}(r_{(c)}, s_{(c)})$$

[43] The parentheses in the indices are used to distinguish the indices from previous orderings of $\bar{\mathcal{S}}_k$ or $\mathcal{R}_k$. These indices are used to label the replicas and surrogates along a particular alternating path.

$$\leq (c-1)\cdot(2d+\hat{\varepsilon}) + \beta_k \cdot (\hat{p}'^{(k)}(s_{(c)}) - \hat{p}'^{(k)}(s_{(1)})) - \hat{w}(r_{(c)}, s_{(c)}) + \sum_{i=1}^{c-1} \left(\Delta_k(r_{(i)}, s_{(i+1)}) - \Delta_k(r_{(i)}, s_{(i)})\right)$$

$$\leq d + \hat{\varepsilon} + (c-1)\cdot(2d+\hat{\varepsilon}) - \beta_k \cdot \hat{p}'^{(k)}(s_{(1)}) - \sum_{i=1}^{c} \Delta_k(r_{(i)}, s_{(i)}) + \sum_{i=1}^{c-1} \Delta_k(r_{(i)}, s_{(i+1)})$$

where the final inequality holds because

$$\beta_k \cdot \hat{p}'^{(k)}(s_{(i)}) - \hat{w}(r_{(i)}, s_{(i)}) \leq -\Delta_k(r_{(i)}, s_{(i)}) + d + \hat{\varepsilon},$$

which itself is seen as follows:

$$\begin{aligned}
\hat{w}(r_{(i)}, s_{(i)}) - \beta_k \cdot \hat{p}'^{(k)}(s_{(i)}) &= \mathrm{val}^{(k)}(r_{(i)}, s_{(i)}) - \hat{p}'^{(k)}(s_{(i)}) \\
&= \mathrm{val}^{(k)}(c_{k,\mathcal{G}}(r_{(i)}), s_{(i)}) - \hat{p}'^{(k)}(s_{(i)}) + \Delta_k(r_{(i)}, s_{(i)}) \\
&\geq \mathrm{val}^{(k)}(s_{(i)}, s_{(i)}) - \hat{p}'^{(k)}(s_{(i)}) + \Delta_k(r_{(i)}, s_{(i)}) - d \\
&\geq \Delta_k(r_{(i)}, s_{(i)}) - d - \hat{\varepsilon}.
\end{aligned}$$

The last inequality here holds because $\mathrm{val}^{(k)}(s_{(i)}, s_{(i)}) - \hat{p}'^{(k)}(s_{(i)}) \geq 0$, since $\hat{\mathcal{M}}'$ is $\hat{\varepsilon}$-IR.

Thus we have that for any $P \in \mathcal{P}$,

$$\begin{aligned}
&\sum_{r_{(i)}, s_{(j)} \in P\cap O} \hat{w}(r_{(i)}, s_{(j)}) - \sum_{r_{(i)}, s_{(j)} \in P\cap M^*} \hat{w}(r_{(i)}, s_{(j)}) \\
&\leq d + \hat{\varepsilon} + |P \cap O| \cdot (2d+\hat{\varepsilon}) + \beta_k \cdot \left[\sum_{s_{(j)} \in P\cap O} \hat{p}'^{(k)}(s_{(j)}) - \sum_{s_{(j)} \in P\cap M^*} \hat{p}'^{(k)}(s_{(j)})\right] + \textsc{Diff}(P)
\end{aligned}$$

where $\textsc{Diff}(P) = \sum_{r_{(i)}, s_{(j)} \in P\cap O} \Delta_k(r_{(i)}, s_{(j)}) - \sum_{r_{(i)}, s_{(j)} \in P\cap M^*} \Delta_k(r_{(i)}, s_{(j)})$. Summing over $P \in \mathcal{P}$:

$$\begin{aligned}
&\sum_{P\in\mathcal{P}} \left[\sum_{r_{(i)}, s_{(j)} \in P\cap O} \hat{w}(r_{(i)}, s_{(j)}) - \sum_{r_{(i)}, s_{(j)} \in P\cap M^*} \hat{w}(r_{(i)}, s_{(j)})\right] \\
&\leq R \cdot (3d + 2\hat{\varepsilon}) + \beta_k \cdot \left[\sum_{s_j \in O} \hat{p}'^{(k)}(s_j) - \sum_{s_j \in M^*} \hat{p}'^{(k)}(s_j)\right] + \sum_{P\in\mathcal{P}} \textsc{Diff}(P),
\end{aligned}$$

because the surrogates matched under M^* and O differ only via the paths in $\mathcal{P}$, by construction of M^*. We now observe that the LHS of the above inequality can be bounded in a different way:

$$\begin{aligned}
&\sum_{P\in\mathcal{P}} \left[\sum_{r_{(i)}, s_{(j)} \in P\cap O} \hat{w}(r_{(i)}, s_{(j)}) - \sum_{r_{(i)}, s_{(j)} \in P\cap M^*} \hat{w}(r_{(i)}, s_{(j)})\right] \\
&= W(O) - W(M^*) \\
&\geq W(O(\mathcal{R}_k, \bar{\mathcal{S}}_k)) - \textsc{Opt}
\end{aligned}$$

where $\textsc{Opt} := \textsc{Opt}(\mathbf{P}(\mathcal{R}_k, \mathcal{S}_k, \hat{w}))$. The equality follows again by construction of M^*: since M^* differs from matching O *only* in the edges in the alternating paths $P \in \mathcal{P}$, the weight difference between the matchings is exactly the sum of the weight difference on each path $P \in \mathcal{P}$. The inequality follows because $\textsc{Opt}(\mathbf{P}(\mathcal{R}_k, \mathcal{S}_k, \hat{w}))$ denotes the weight of the *maximum*-weight matching on $(\mathcal{R}_k, \bar{\mathcal{S}}_k)$, which is at least $W(M^*)$.[44]

Straightforward rearranging of the last two expressions yields

$$\frac{1}{R} \cdot \sum_{s_j \in O} \hat{p}'^{(k)}(s_j) \geq \frac{1}{R} \cdot \sum_{s_j \in M'} \hat{p}'^{(k)}(s_j) - \frac{1}{R\beta_k} (\textsc{Opt} - W(O)) - \frac{1}{R\beta_k} \sum_{P \in \mathcal{P}} \textsc{Diff}(P) - \frac{1}{\beta_k} \cdot (2\hat{\varepsilon} + 3d).$$

The claim now follows by observing that $\Delta_k(r_i, s_j) \in [-\mathrm{dist}_k(r_i, c_{k,\mathcal{G}}(r_i)), \mathrm{dist}_k(r_i, c_{k,\mathcal{G}}(r_i))]$ for any s_j, which implies that $\sum_{P \in \mathcal{P}} \textsc{Diff}(P) \leq \sum_{r_i \in \mathcal{R}_k} 2 \cdot \mathrm{dist}_k(r_i, c_{k,\mathcal{G}}(r_i))$. □

Step 4. Take expectations over Step 3 terms. We now prove our main technical lemma on revenue bounds by carefully taking (conditional) expectations and leveraging meta-framework guarantees to bound the terms in Claim 3.

Claim 4 (Analog of [COVZ19], Lemma 23)**.** *Fix surrogate sets $\mathcal{S} := (\mathcal{S}_{k'})_{k' \in [n]}$ and fix a bidder k.*

$$\begin{aligned}\textsc{Rev}_k^{(2)}(\mathcal{M}; D) \mid \mathcal{S} \geq\ & \textsc{Rev}_k(\hat{\mathcal{M}}'; \hat{D}') - \beta_k - (m+1)\delta_{\textsc{Flag}} - \Delta_{\textsc{Match}} \\ & - \frac{1}{\beta_k}\left(3d + 2\hat{\varepsilon} + \Delta_{\textsc{MWM}} + 2\Delta_{\textsc{Couple}}^{(k)} + 3\delta_{\textsc{Flag}}\right)\end{aligned}$$

Proof of Claim 4. Claim 3 tells us the following: fixing $\mathcal{G}$, $\mathcal{R}_k$, and the output $O(\mathcal{R}_k, \bar{\mathcal{S}}_k)$ of $\textsc{MetaRS}$ such that Φ_{F}^k holds and fixing the realization of $c_{k,\mathcal{G}}(\mathcal{R}_k)$, we have

$$\frac{1}{R} \cdot \sum_{s_j \in O} \hat{p}'^{(k)}(s_j) \geq \frac{1}{R} \cdot \sum_{s_j \in M^*} \hat{p}'^{(k)}(s_j) - \frac{1}{R\beta_k} (\textsc{Opt} - W(O)) - \frac{2}{R\beta_k} \left(\sum_{r_i \in \mathcal{R}_k} \mathrm{dist}_k(r_i, c_{k,\mathcal{G}}(r_i)) \right) - \frac{1}{\beta_k} \cdot (2\hat{\varepsilon} + 3d).$$

Note first that $\frac{1}{R} \cdot \sum_{s_j \in M^*} \hat{p}'^{(k)}(s_j) \geq \textsc{Rev}_k(\hat{\mathcal{M}}'; \hat{D}') - (1 - |M|/R)$ (where matching M is as defined in Step 2 above). We now take expectations over the draw of bidder k's type, all internal randomness in $\textsc{MetaRS}$ and the meta-inputs used in it, and the realization of the coupling $c_{k,\mathcal{G}}(\cdot)$:[45]

$$\begin{aligned}\mathbb{E}\left[\frac{1}{R} \cdot \sum_{s_j \in O} \hat{p}'^{(k)}(s_j) \mid \Phi_{\mathrm{F}}^k\right] \geq\ & \textsc{Rev}_k(\hat{\mathcal{M}}'; \hat{D}') - \frac{3d + 2\hat{\varepsilon}}{\beta_k} \\ & - \mathbb{E}\left[(1 - |M|/R) \mid \Phi_{\mathrm{F}}^k\right] \\ & - \frac{1}{R\beta_k} \cdot \mathbb{E}\left[\textsc{Opt} - W(O) \mid \Phi_{\mathrm{F}}^k\right] \\ & - \frac{2}{R\beta_k} \cdot \mathbb{E}\left[\sum_{r_i \in \mathcal{R}_k} \mathrm{dist}_k(r_i, c_{k,\mathcal{G}}(r_i)) \mid \Phi_{\mathrm{F}}^k\right]\end{aligned}$$

[44] Note that the max-weight κ-to-1 matching on $(\mathcal{R}_k, \bar{\mathcal{S}}_k)$ corresponds exactly to the max-weight 1-to-1 matching on $(\mathcal{R}_k, \bar{\mathcal{S}}_k)$.

[45] The choice of coupling $c_{k,\mathcal{G}}$ itself is a random variable over the randomness of $\textsc{Preprocess}$, over which we also take an expectation.

We must now deal with a technical detail: we would like to bound the latter three terms above using Conditions (B3), (B4), and (C2) of the meta-framework, where expectations are not conditioned on event Φ_{F}^{k}. To handle this, we use the following elementary fact.

Fact A.5 (Expectations conditioned on high probability events)**.** Let X be a random variable supported on $[a, b]$ for some $0 \le a \le b \in \mathbb{R}$. Let C be an event that holds with probability at least $1-\delta$, for some $\delta \in [0, 1)$. Then

$$\mathbb{E}[X|C] \le \mathbb{E}[X] + \delta(b-a).$$

Note that it is important that the terms inside each of the conditional expectations on the RHS of the previous expression be *well-defined unconditionally* in order to apply Theorem A.5. The term $\textsc{Opt} - W(O)$ is supported on $[0, 2R]$, since edge weights are in $[-1, 1]$ ($\mathrm{val}^{(k)}(\cdot,\cdot) \in [0,1]$ and $\hat{p}'^{(k)}(\cdot) \in [0,1]$). The term $(1 - |M|/R)$ is supported on $[0, 1]$. The term $\sum_{r_i \in \mathcal{R}_k} \mathrm{dist}_k(r_i, c_{k,\mathcal{G}}(r_i))$ is supported on $[0, R]$.

First, since $\Pr\left[\Phi_{\mathrm{F}}^{k}\right] \ge 1 - \delta_{\textsc{Flag}}$,

$$\mathbb{E}\left[\textsc{Opt} - W(O) \mid \Phi_{\mathrm{F}}^{k}\right] \le \mathbb{E}\left[\textsc{Opt} - W(O)\right] + 2\delta_{\textsc{Flag}} R.$$

Observe that $\mathbb{E}[W(O)] = \mathbb{E}[W(O(\mathcal{R}_k, \mathcal{S}_k))] = \mathbb{E}_{\sigma}[(1-\lambda)x^*_{ij}(\hat{\alpha})\hat{w}_{ij}]$, where the final expectation is only over randomness σ defined by $\textsc{Preprocess}$, $\textsc{Compute-}\alpha$, drawing bidder k's type from D_k, and $\textsc{Draw-}\mathcal{R}$—in particular, this expression is obtained after taking the expectation with respect to the matching process in Step 4 of $\textsc{MetaRS}$. Condition (C2) then implies that $\mathbb{E}[\textsc{Opt} - W(O)] \le R \cdot \Delta_{\mathrm{MWM}}$.

For the other two terms, note that $\Phi_{\mathrm{F}}^{k} = \Phi_{\mathrm{F}\text{-}\mathcal{R}}^{k} \cap E$, where $\Phi_{\mathrm{F}\text{-}\mathcal{R}}^{k}$ is the event that $\textsc{Flag}_{\mathcal{R}} = \texttt{True}$ for bidder k and E is the appropriate event implied by Step 5 in $\textsc{MetaRS}$ (the details of which do not matter here). Clearly $\Pr[E] \ge \Pr\left[\Phi_{\mathrm{F}}^{k}\right] \ge 1 - \delta_{\textsc{Flag}}$. Treating E as the event C in Theorem A.5 then yields

$$\mathbb{E}\left[(1 - |M|/R) \mid \Phi_{\mathrm{F}}^{k}\right] \le \mathbb{E}\left[(1 - |M|/R) \mid \Phi_{\mathrm{F}\text{-}\mathcal{R}}^{k}\right] + \delta_{\textsc{Flag}}$$

$$\mathbb{E}\left[\sum_{r_i \in \mathcal{R}_k} \mathrm{dist}_k(r_i, c_{k,\mathcal{G}}(r_i)) \mid \Phi_{\mathrm{F}}^{k}\right] \le \mathbb{E}\left[\sum_{r_i \in \mathcal{R}_k} \mathrm{dist}_k(r_i, c_{k,\mathcal{G}}(r_i)) \mid \Phi_{\mathrm{F}\text{-}\mathcal{R}}^{k}\right] + \delta_{\textsc{Flag}} R.$$

Condition (B3) implies that $\mathbb{E}\left[(1 - |M|/R) \mid \Phi_{\mathrm{F}\text{-}\mathcal{R}}^{k}\right] \le \Delta_{\textsc{Match}}$. Condition (B4) implies that $\mathbb{E}\left[\sum_{r_i \in \mathcal{R}_k} \mathrm{dist}_k(r_i, c_{k,\mathcal{G}}(r_i)) \mid \Phi_{\mathrm{F}\text{-}\mathcal{R}}^{k}\right] \le R \cdot \Delta^{(k)}_{\textsc{Couple}}$.

Substituting all of these bounds, we conclude that

$$\begin{aligned}
\textsc{Rev}_k^{(2)}(\mathcal{M}; D)|\mathcal{S} &\ge (1-\beta_k)\,\mathbb{E}\left[\frac{1}{R} \cdot \sum_{s_j \in O} \hat{p}'^{(k)}(s_j) \mid \Phi_{\mathrm{F}}^{k}\right] - n\delta_{\textsc{Flag}} && \text{Claim 1} \\
&\ge \mathbb{E}\left[\frac{1}{R} \cdot \sum_{s_j \in O} \hat{p}'^{(k)}(s_j) \mid \Phi_{\mathrm{F}}^{k}\right] - \beta_k - n\delta_{\textsc{Flag}} && \hat{p}'^{(k)}(\cdot) \le 1 \\
&\ge \textsc{Rev}_k(\hat{\mathcal{M}}'; \hat{D}') - \beta_k - (n+1)\delta_{\textsc{Flag}} - \Delta_{\textsc{Match}}
\end{aligned}$$

$$-\frac{1}{\beta_k}\cdot(\Delta_{\mathrm{MWM}}+3d+2\hat{\varepsilon}+2\Delta^{(k)}_{\textsc{Couple}}+3\delta_{\textsc{Flag}}) \qquad \text{via Theorem A.5}$$

□

Step 5. Putting everything together. We are finally ready to conclude Theorem A.3. We start by summing over k and taking expectations over the draw of $\mathcal{S}$ via DRAW-$\mathcal{S}$:

$$\mathop{\mathbb{E}}_{\mathcal{S}}\left[\sum_{k=1}^{n}\textsc{Rev}_k^{(2)}(\mathcal{M};D)|\mathcal{S}\right]\geq \mathop{\mathbb{E}}_{\mathcal{S}}\left[\textsc{Rev}(\hat{\mathcal{M}}';\hat{D}')\right]-\frac{1}{\beta_k}\mathop{\mathbb{E}}_{\mathcal{S}}[2n\hat{\varepsilon}]$$
$$-\sum_{k=1}^{n}\left[\beta_k+(n+1)\delta_{\textsc{Flag}}+\Delta_{\textsc{Match}}+\frac{1}{\beta_k}\cdot(\Delta_{\mathrm{MWM}}+3d+2\Delta^{(k)}_{\textsc{Couple}}+3\delta_{\textsc{Flag}})\right]$$

The LHS is equal to $\sum_{k\in[n]}\textsc{Rev}_k^{(2)}(\mathcal{M};D)$. Next, it is not hard to see that both $\textsc{Rev}(\hat{\mathcal{M}}';\hat{D}')$ and $n\cdot\hat{\varepsilon}$, viewed as a random variables over DRAW-$\mathcal{S}$ and ESTMECHANISM, are supported on $[0,n]$. Condition (A1) then implies, via an application of Theorem A.5, that $\mathbb{E}_{\mathcal{S}}\left[\textsc{Rev}(\hat{\mathcal{M}}';\hat{D}')\right]\geq \textsc{Rev}(\mathcal{M}';D')-\Delta_{\text{D-S}}-n\delta_{\text{D-S}}$ and $\mathbb{E}_{\mathcal{S}}[2n\hat{\varepsilon}]\leq 2n(\varepsilon+\varepsilon_{\text{D-S}})+2n\delta_{\text{D-S}}$.

Substituting in, we have that

$$\sum_{k\in[n]}\textsc{Rev}_k^{(2)}(\mathcal{M};D)\geq \textsc{Rev}(\mathcal{M}';D')-\Delta_{\text{D-S}}-\sum_{k=1}^{n}\Delta_{\textsc{Rev}}{}^{(k),(2)}$$

where

$$\Delta_{\textsc{Rev}}{}^{(k),(2)}=\beta_k+(n+1)\delta_{\textsc{Flag}}+\delta_{\text{D-S}}+\Delta_{\textsc{Match}}+\frac{1}{\beta_k}\cdot(\Delta_{\mathrm{MWM}}+2\varepsilon+2\varepsilon_{\text{D-S}}+2\delta_{\text{D-S}}+3d+2\Delta^{(k)}_{\textsc{Couple}}+3\delta_{\textsc{Flag}}).$$

By our earlier observation via Condition (E3) this implies the claim in Theorem A.3. □

B Extension: Interim IR to Ex-Post IR Reduction

In this section, we prove Theorem 5.2, which we restate here.

Theorem B.1. *Suppose we are given sample access to distributions $D'=\times_{k\in[n]}D'_k$ and $D=\times_{k\in[n]}D_k$ of bidder types, and query access to a mechanism $\mathcal{M}'$ that is IR and ε-BIC over D'. Suppose we have implementations of the meta-inputs* DRAW-$\mathcal{S}$, ESTMECHANISM, PREPROCESS, DRAW-$\mathcal{R}$, COMPUTE-α, MATCH, *and* PAYMENT, *together with a choice of parameter values (Section 5.3.1), that satisfy the meta-framework requirements (Section 5.3). If we have interim access to the output of* ESTMECHANISM, *then replacing Phase 2 in Algorithm 2 with Algorithm 3 defines an ex-post IR and (exactly) BIC mechanism $\mathcal{M}$ over n bidders that achieves revenue $\textsc{Rev}(\mathcal{M};D)\geq\textsc{Rev}(\mathcal{M}';D')-\Delta_{\textsc{Rev}}$ using $N(D)$ samples from D and $N(D')$ samples from D'.*

Proof. Let $\mathcal{M}^-=(\mathcal{A}^-,p^-)$ denote the output of Algorithm 2, and let $\mathcal{M}=(\mathcal{A},p)$ denote the output of Algorithm 2 but with Phase 2 replaced with Algorithm 3. Condition on the output $(\mathcal{S}_k)_k$ of DRAW-$\mathcal{S}$ and the event that $\textsc{Flag}_{\textsc{Est}}=$ `True`. Let $\Phi_{\textsc{Keep}}$ denote the event, over the randomness of $t_{\text{-}k}\sim D_{\text{-}k}$ and internal randomness of METARS, that $\textsc{Coin}^{(k)}=$ `True` and $\textsc{Flag}^{(k')}=$ `True` for all $k' \neq k \in [n]$ in Algorithm 2 (note that these events are independent of bidder k's report). Letting $\hat{\mathcal{M}}' = (\hat{\mathcal{A}}', \hat{p}')$ denote the output of EstMechanism, we have that

$$\begin{aligned}
p_k^-(t) &= ((1-\beta_k)\cdot \hat{p}'_k(s^{(k)}, s^{(-k)})\cdot 1(\neg\texttt{dummy}^{(k)}) + q_k)\cdot 1(\Phi_{\text{Keep}})\\
p_k(t) &= v_k(t_k, \hat{\mathcal{A}}'(s^{(k)}, s^{(-k)}))\cdot \frac{\mathbb{E}[q_k]+(1-\beta_k)\cdot\sum_{s_j\in\mathcal{S}_k} x^*_{kj}\cdot\mathbb{E}_{s_{-k}\sim\hat{D}'_{-k}}[\hat{p}'_k(s_j,s_{-k})]}{\sum_{s_j\in\mathcal{S}_k} x^*_{kj}\cdot\mathbb{E}_{s_{-k}\sim\hat{D}'_{-k}}[v_k(t_k,\hat{\mathcal{A}}'(s_j,s_{-k}))]}\cdot 1(\Phi_{\text{Keep}}, \neg\texttt{dummy}^{(k)})\\
u_k(t_k, \mathcal{M}^-(t'_k, t_{-k})) &= v_k(t_k, \hat{\mathcal{A}}'(s^{(k)}, s^{(-k)}))\cdot 1(\Phi_{\text{Keep}}, \neg\texttt{dummy}^{(k)}) - p_k^-(t'_k, t_{-k})\\
u_k(t_k, \mathcal{M}(t'_k, t_{-k})) &= v_k(t_k, \hat{\mathcal{A}}'(s^{(k)}, s^{(-k)}))\cdot 1(\Phi_{\text{Keep}}, \neg\texttt{dummy}^{(k)}) - p_k(t'_k, t_{-k})
\end{aligned}$$

where $s^{(k)}$ is the surrogate matched to bidder k according to MetaRS. Since $\mathcal{M}^-$ satisfies Condition (E2), we have that

$$\textstyle\sum_{s_j\in S_k} x^*_{kj}\cdot\mathbb{E}_{s_{-k}\sim\hat{D}'_{-k}}[v_k(t_k,\hat{\mathcal{A}}'(s_j,s_{-k}) - (1-\beta_k)\cdot\hat{p}'_k(s_j,s_{-k})] = \sum_{s_j\in S_k} x^*_{kj}\cdot\hat{w}_{kj} \geq \mathbb{E}[q_k],$$

so $u_k(t_k, \mathcal{M}(t)) \geq 0$ for all type profiles $t \in \text{supp}(D)$. Moreover, since $\mathcal{M}^-$ is BIC and achieves the desired revenue guarantee, to show that $\mathcal{M}$ also satisfies these properties, it suffices to show that the interim payment from each bidder k coincides between $\mathcal{M}^-$ and $\mathcal{M}$. Recall that we are conditioning on $(\mathcal{S}_k)_k$ and the event that $\text{Flag}_{\text{Est}} = \texttt{True}$.

$$\begin{aligned}
\mathbb{E}_{t_{-k}\sim D_{-k}}[p_k^-(t_k,t_{-k})] &= \mathbb{E}_{t_{-k}\sim D_{-k}}[(1-\beta_k)\cdot\hat{p}'_k(s^{(k)},s^{(-k)})\cdot 1(\neg\texttt{dummy}^{(k)}) + q_k \mid \Phi_{\text{Keep}}]\cdot\Pr[\Phi_{\text{Keep}}]\\
&= (1-\beta_k)\cdot\mathbb{E}_{s^{(-k)}\sim\hat{D}'_{-k}}[\hat{p}'_k(s^{(k)},s^{(-k)})\cdot 1(\neg\texttt{dummy}^{(k)})]\cdot\Pr[\Phi_{\text{Keep}}]\\
&\quad + \mathbb{E}[q_k]\cdot\Pr[\Phi_{\text{Keep}}] && \text{(Theorem A.2)}\\
&= ((1-\beta_k)\cdot\textstyle\sum_{s_j\in\mathcal{S}_k} x^*_{kj}\cdot\mathbb{E}_{s^{(-k)}\sim\hat{D}'_{-k}}[\hat{p}'_k(s_j,s^{(-k)})] + \mathbb{E}[q_k])\cdot\Pr[\Phi_{\text{Keep}}]\\
\mathbb{E}_{t_{-k}\sim D_{-k}}[p_k(t_k,t_{-k})] &= \mathbb{E}_{t_{-k}\sim D_{-k}}\Bigg[v_k(t_k,\hat{\mathcal{A}}'(s^{(k)},s^{(-k)}))\cdot\frac{\mathbb{E}[q_k]+(1-\beta_k)\cdot\sum_{s_j\in\mathcal{S}_k} x^*_{kj}\cdot\mathbb{E}_{s_{-k}\sim\hat{D}'_{-k}}[\hat{p}'_k(s_j,s_{-k})]}{\sum_{s_j\in\mathcal{S}_k} x^*_{kj}\cdot\mathbb{E}_{s_{-k}\sim\hat{D}'_{-k}}[v_k(t_k,\hat{\mathcal{A}}'(s_j,s_{-k}))]}\\
&\qquad\cdot 1(\neg\texttt{dummy}^{(k)}) \mid \Phi_{\text{Keep}}\Bigg]\cdot\Pr[\Phi_{\text{Keep}}]\\
&= \frac{\mathbb{E}[q_k]+(1-\beta_k)\cdot\sum_{s_j\in\mathcal{S}_k} x^*_{kj}\cdot\mathbb{E}_{s_{-k}\sim\hat{D}'_{-k}}[\hat{p}'_k(s_j,s_{-k})]}{\sum_{s_j\in\mathcal{S}_k} x^*_{kj}\cdot\mathbb{E}_{s_{-k}\sim\hat{D}'_{-k}}[v_k(t_k,\hat{\mathcal{A}}'(s_j,s_{-k}))]}\\
&\quad\cdot\mathbb{E}_{s^{(-k)}\sim\hat{D}'_{-k}}[v_k(t_k,\hat{\mathcal{A}}'(s^{(k)},s^{(-k)}))\cdot 1(\neg\texttt{dummy}^{(k)})]\cdot\Pr[\Phi_{\text{Keep}}] && \text{(Theorem A.2)}\\
&= \frac{\mathbb{E}[q_k]+(1-\beta_k)\cdot\sum_{s_j\in\mathcal{S}_k} x^*_{kj}\cdot\mathbb{E}_{s_{-k}\sim\hat{D}'_{-k}}[\hat{p}'_k(s_j,s_{-k})]}{\sum_{s_j\in\mathcal{S}_k} x^*_{kj}\cdot\mathbb{E}_{s_{-k}\sim\hat{D}'_{-k}}[v_k(t_k,\hat{\mathcal{A}}'(s_j,s_{-k}))]}\\
&\quad\cdot\textstyle\sum_{s_j\in\mathcal{S}_k} x^*_{kj}\cdot\mathbb{E}_{s_{-k}\sim\hat{D}'_{-k}}[v_k(t_k,\hat{\mathcal{A}}'(s_j,s_{-k}))]\cdot\Pr[\Phi_{\text{Keep}}]\\
&= (\mathbb{E}[q_k] + (1-\beta_k)\cdot\textstyle\sum_{s_j\in\mathcal{S}_k} x^*_{kj}\cdot\mathbb{E}_{s_{-k}\sim\hat{D}'_{-k}}[\hat{p}'_k(s_j,s_{-k})])\cdot\Pr[\Phi_{\text{Keep}}]
\end{aligned}$$

□

C Analysis: Main Result

C.1 Draw-$\mathcal{S}$ and EstMechanism

Condition (A1).

We prove the condition via Theorem C.1 below. In particular, note that Condition (A1) follows by setting $\varepsilon' := \frac{1}{2}\varepsilon_{\text{D-S}} = O(\varepsilon)$, $\Delta := \Delta_{\text{D-S}} = O(n\varepsilon)$, and $N := N_S = \frac{m^3 n}{\varepsilon^3}$ in Theorem C.1, for which the failure probability comes out to $O(\varepsilon) = \delta_{\text{D-S}}$.

Lemma C.1. *Consider mechanism $\mathcal{M}'$ over $D' = \times_{k\in[n]} D'_k$, where each $D'_k = \times_{\ell\in[m]} D'_{k\ell}$. For each bidder k and item ℓ, draw N samples from $D'_{k\ell}$ and define $\hat{D}'_{k\ell}$ to be the uniform distribution over these samples. Let $\hat{D}'_k = \times_\ell \hat{D}'_{k\ell}$ and $\hat{D}' = \times_k \hat{D}'_k$. If ex-post payments are bounded in $[0,1]$, then with probability at least*

$$1 - \frac{8nN^{2m}}{\varepsilon'} \exp\left(-\frac{\varepsilon'^2 N}{8}\right) - \frac{4n}{\Delta} \exp\left(-\frac{\Delta^2 N}{8n^2}\right),$$

$\mathcal{M}'$ is an $(\varepsilon + 4\varepsilon')$-BIC and $2\varepsilon'$-IR mechanism for bidders with types drawn from $\hat{D}'$ whose revenue from these bidders is at most Δ less than its revenue from bidders whose types are drawn from D'.

The proof uses a concentration result due to Babichenko et al. [BBP17], stated in Theorem C.8.

Proof. Since $\mathcal{M}'$ is ε-BIC and IR over D', we have that

$$\begin{aligned}
&\mathbb{E}_{t_{\text{-}k}\sim D'_{\text{-}k}}[v_k(t_k, \mathcal{A}'(t)) - p'_k(t)] \geq \mathbb{E}_{t_{\text{-}k}\sim D'_{\text{-}k}}[v_k(t_k, \mathcal{A}'(t'_k, t_{\text{-}k})) - p'_k(t'_k, t_{\text{-}k})] - \varepsilon \\
&\mathbb{E}_{t_{\text{-}k}\sim D'_{\text{-}k}}[v_k(t_k, \mathcal{A}'(t)) - p'_k(t)] \geq 0
\end{aligned}$$

for all bidders k and types $t_k, t'_k \in \text{supp}(D'_k)$.

Recall that valuations $v_k(\cdot,\cdot)$ and payments $p'_k(\cdot)$ are both in $[0,1]$. Theorem C.8 then implies that for each bidder k and types $t_k, t'_k \in \text{supp}(\hat{D}'_k)$,

$$\begin{aligned}
&\Pr(|\mathbb{E}_{t_{\text{-}k}\sim \hat{D}'_{\text{-}k}}[v_k(t_k, \mathcal{A}'(t'_k, t_{-k}))] - \mathbb{E}_{t_{\text{-}k}\sim D'_{\text{-}k}}[v_k(t_k, \mathcal{A}'(t'_k, t_{-k}))]| > \varepsilon') \leq \frac{4}{\varepsilon'} \exp\left(-\frac{\varepsilon'^2 N}{8}\right) \\
&\Pr(|\mathbb{E}_{t_{\text{-}k}\sim \hat{D}'_{\text{-}k}}[p'_k(t'_k, t_{-k})] - \mathbb{E}_{t_{\text{-}k}\sim D'_{\text{-}k}}[p'_k(t'_k, t_{-k})]| > \varepsilon') \leq \frac{4}{\varepsilon'} \exp\left(-\frac{\varepsilon'^2 N}{8}\right)
\end{aligned}$$

Via union bound, the probability that there exists a bidder k and types $t_k, t'_k \in \text{supp}(\hat{D}'_k)$ for which the above events do not hold is at most

$$\frac{8nN^{2m}}{\varepsilon'} \exp\left(-\frac{\varepsilon'^2 N}{8}\right)$$

Thus, with probability at least 1 less the above quantity, $\mathcal{M}'$ is an $(\varepsilon + 4\varepsilon')$-BIC and $2\varepsilon'$-IR mechanism for bidders whose types are drawn from $\hat{D}'$ instead.

It remains to bound the revenue loss. Since each $p'_k(\cdot) \in [0,1]$, it follows from Theorem C.8 that

$$\Pr(|\mathbb{E}_{t\sim\hat{D}'}[\textstyle\sum_k p'_k(t)] - \mathbb{E}_{t\sim D'}[\sum_k p'_k(t)]| > \Delta) \leq \frac{4n}{\Delta} \exp\left(-\frac{\Delta^2 N}{8n^2}\right).$$

This completes the proof since $\textsc{Rev}(\mathcal{M}', \hat{D}') = \mathbb{E}_{t\sim\hat{D}'}[\sum_k p'_k(t)]$ and $\textsc{Rev}(\mathcal{M}'; D') = \mathbb{E}_{t\sim D'}[\sum_k p'_k(t)]$. $\square$

C.2 Preprocess

For the remaining proofs of meta-input requirements, we consider a fixed bidder k. Though we do not have a Meta-Framework condition on Preprocess, we prove for later use the following lemma about the probability mass covered by the returned points $(G_{k\ell})_{\ell\in[m]}$.

Lemma C.2 (Preprocess concentration). *With probability at least $1-\exp(-\frac{nm}{\varepsilon})$ over the randomness of* Preprocess*, the output $\mathcal{G}=(G_{k\ell})_{\ell\in[m]}$ satisfies that, for all $\ell\in[m]$,*

$$\Pr_{z\sim\lfloor D_{k\ell}\rfloor_\eta}[z\notin G_{k\ell}]\le\frac{\varepsilon}{nm},\ \textit{and}$$

$$\forall y\in G_{k\ell},\ \Pr_{z\sim\lfloor D_{k\ell}\rfloor_\eta}[z=y]\ge\eta\cdot\Big(\frac{\varepsilon}{nm}\Big)^3=\frac{\varepsilon^6}{m^6n^4}.$$

Proof of Theorem C.2. Recall that $\eta=\frac{\varepsilon^3}{m^3n}$ and $N_P=\frac{2}{\eta}\cdot\left(\frac{nm}{\varepsilon}\right)^4$. We define probability thresholds $p_A:=\frac{\eta}{2}\cdot\frac{\varepsilon}{nm}$ and $p_B:=\eta\cdot\left(\frac{\varepsilon}{nm}\right)^3$. Consider any $\ell\in[m]$. Let $\tau:=\left(\frac{nm}{\varepsilon}\right)^2$ denote the number of times a point must appear to be included in $G_{k\ell}$ in Preprocess.

First, consider any $y\in\mathcal{I}_\eta$ such that $\Pr_{\lfloor D_{k\ell}\rfloor_\eta}[y]\ge p_A$. Consider the multiset P in Preprocess and let N_y denote the number of times y appears in P. Then $\mathbb{E}[N_y]\ge p_A\cdot N_P=\frac{nm}{\varepsilon}\cdot\tau$. Thus it follows via a Chernoff bound that

$$\begin{aligned}\Pr[y\notin G_{k\ell}]&=\Pr[N_y<\tau]\\&=\Pr\left[N_y<\frac{\varepsilon}{nm}\cdot\mathbb{E}[N_y]\right]\\&\le\exp\left(-\frac{1}{4}\left(\frac{nm}{\varepsilon}\right)^2\right)\end{aligned}$$

Clearly there are at most $|\mathcal{I}_\eta|$ points y such that $\Pr_{\lfloor D_{k\ell}\rfloor_\eta}[y]\ge p_A$. By union bound over the points in $\mathcal{I}_\eta$, this implies that except with probability at most $\frac{1}{2n}\exp(-\frac{nm}{\varepsilon})$, all $y\in\mathcal{I}_\eta$ with $\Pr_{\lfloor D_{k\ell}\rfloor_\eta}[y]\ge p_A$ are in $G_{k\ell}$ (by choice of $\eta\ll\exp(\frac{nm}{\varepsilon})$). Under this event it follows that $\Pr_{z\sim\lfloor D_{k\ell}\rfloor_\eta}[z\notin G_{k\ell}]\le p_A\cdot|\mathcal{I}_\eta|\le\frac{\varepsilon}{nm}$. By a union bound over $\ell\in[m]$, this implies that except with probability at most $\frac{1}{2}\exp(-\frac{nm}{\varepsilon})$, it holds for all $\ell\in[m]$ that $\Pr_{z\sim\lfloor D_{k\ell}\rfloor_\eta}[z\notin G_{k\ell}]\le\frac{\varepsilon}{nm}$.

Next consider any $y\in\mathcal{I}_\eta$ such that $\Pr_{\lfloor D_{k\ell}\rfloor_\eta}[y]<p_B$. Let N_y denote the number of times y appears in P. Then $\mathbb{E}[N_y]<\frac{nm}{\varepsilon}$. It follows via a Chernoff bound that

$$\begin{aligned}\Pr[y\in G_{k\ell}]&=\Pr[N_y\ge\tau]\\&=\Pr\left[N_y\ge\left(\left(\frac{nm}{\varepsilon}\right)^2\cdot\frac{1}{\mathbb{E}[N_y]}\right)\cdot\mathbb{E}[N_y]\right]\\&\le\exp\left(-\frac{1}{2}\left(\frac{nm}{\varepsilon}\right)^2\right)\end{aligned}$$

Union bounding over all points in $\mathcal{I}_\eta$, we have that except with probability $\frac{1}{2n}\exp(-\frac{nm}{\varepsilon})$, *no* $y\in\mathcal{I}_\eta$ with $\Pr_{\lfloor D_{k\ell}\rfloor_\eta}[y]<p_B$ is in $G_{k\ell}$. This implies the claim after union bounding over $\ell\in[m]$. □

C.3 Draw-$\mathcal{R}$

Condition (B1).

Proof. It is immediate that the returned set $\mathcal{R}_k$ contains the bidder type t_k, specifically, $r_{i^*} = t_k$. We now bound the probability that $\text{FLAG}_{\mathcal{R}} = \texttt{False}$. First, it is straightforward to see via a Chernoff bound that $\Pr\left[\text{FLAG}^{\ell}_{\text{DRAW}} = \texttt{False}\right] = \exp(-\Omega(N_R))$ and so

$$\Pr\left[\exists\ \ell \text{ s.t. } \text{FLAG}^{\ell}_{\text{DRAW}} = \texttt{False}\right] = m \cdot \exp(-\Omega(N_R)).$$

Next, it follows from Theorem C.2 and a union bound that except with probability at most $\exp\left(-\frac{nm}{\varepsilon}\right)$ (over the randomness of PREPROCESS),

$$\Pr_{t_k \sim D_k}[\exists\ \ell \text{ s.t. } \lfloor t_\ell \rfloor_\eta \notin G_{k\ell}] \leq \frac{\varepsilon}{n}.$$

Putting everything together, we have by choice of N_R that

$$\begin{aligned}
\delta_{\text{D-R}} &= \Pr\left[\text{FLAG}_{\mathcal{R}} = \texttt{False}\right] \\
&\leq m \cdot \exp(-\Omega(N_R)) + \exp\left(-\frac{nm}{\varepsilon}\right) + \frac{\varepsilon}{n} \\
&= O\left(\frac{\varepsilon}{n}\right)
\end{aligned}$$

as desired. □

For Conditions (B2) – (B4), we define distributions $F_{\mathcal{G}} := F_{k,\mathcal{G}}$ and the couplings $c_{\mathcal{G}} := c_{k,\mathcal{G}}$ for any output $\mathcal{G} = (G_{k\ell})_\ell$ of PREPROCESS as follows. We drop the subscript k in this notation for brevity throughout the analysis.

- **Distribution:** For each $\ell \in [m]$, let $F_{\mathcal{G},\ell}$ be the distribution of $x \sim D_{k\ell}$ conditioned on $\lfloor x \rfloor_\eta \in G_{k\ell}$. Then define distribution $F_{\mathcal{G}} := \times_{\ell \in [m]} F_{\mathcal{G},\ell}$.

- **Coupling:** Let c denote an optimal product coupling between D_k and D'_k, i.e.

$$c \in \arg\min_{\Gamma \in \Pi_{\text{prod}}(D_k, D'_k)} d_\Gamma(D_k, D'_k),$$

  for distance function $d(\cdot,\cdot) = \text{dist}_k(\cdot,\cdot)$. For each ℓ, let c_ℓ denote the coupling between $D_{k\ell}$ and $D'_{k\ell}$ induced by c, such that $c = \times_{\ell=1}^m c_\ell$. Next, for all ℓ, we define a coupling $\bar{c}_{\mathcal{G},\ell}$ between $F_{\mathcal{G},\ell}$ and $D_{k\ell}$, where a draw $(x, y) \sim \bar{c}_{\mathcal{G},\ell}$ is defined as follows. Let $p(G_{k\ell}) := \Pr_{y \sim D_{k\ell}}[\lfloor y \rfloor_\eta \in G_{k\ell}]$. First, draw $x \sim F_{\mathcal{G},\ell}$. Independently flip $Z = \text{BERN}(p(G_{k\ell}))$. If $Z = 1$, then set $y = x$; else, sample y from the distribution of $y' \sim D_{k\ell}$ conditioned on $\lfloor y' \rfloor_\eta \notin G_{k\ell}$.

  We then define $c_{\mathcal{G},\ell}$ to be the *composed* coupling $c_{\mathcal{G},\ell} := c_\ell \circ \bar{c}_{\mathcal{G},\ell}$. That is, to sample $(x, z) \sim c_{\mathcal{G},\ell}$, first draw $x \sim F_{\mathcal{G},\ell}$, then draw $y \sim \bar{c}_{\mathcal{G},\ell}(x)$, and then draw $z \sim c(y)$. Finally, we define $c_{\mathcal{G}}$ as the product coupling $c_{\mathcal{G}} := \times_{\ell \in [m]} c_{\mathcal{G},\ell}$ over the couplings $(c_{\mathcal{G},\ell})_\ell$.

  It is straightforward to verify that $\bar{c}_{\mathcal{G},\ell}$ is a valid coupling between $F_{\mathcal{G},\ell}$ and $D_{k\ell}$, which implies by composition that $c_{\mathcal{G},\ell}$ is a valid coupling between $F_{\mathcal{G},\ell}$ and $D'_{k\ell}$, and so $c_{\mathcal{G}}$ is a valid coupling between $F_{\mathcal{G}}$ and D'_k.

The following claim will be useful for proving multiple conditions.

Claim 5. *Conditioned on $\mathcal{G}$ and $\text{FLAG}_{\mathcal{R}} = \texttt{True}$, for each $\ell \in [m]$, the elements of the multiset $\mathcal{R}_{k\ell}$ in DRAW-$\mathcal{R}$ are i.i.d samples from $F_{\mathcal{G},\ell}$ (over randomness of $t_k \sim D_k$ and internal randomness of DRAW-$\mathcal{R}$).*

Proof. Condition on $(G_{k\ell})_\ell$ and $\textsc{Flag}_{\mathcal{R}} = \texttt{True}$. Fix any $\ell \in [m]$. It follows by inspection of $\textsc{Draw}$ that the elements of $\mathcal{R}^-_{k\ell}$ are i.i.d samples from $F_{\mathcal{G},\ell}$, since these are the first $N := N_R - 1$ draws x such that $\lfloor x \rceil_\eta \in G_{k\ell}$ out of the $2N$ i.i.d draws from $D_{k\ell}$. Similarly, it is clear by inspection of $\textsc{Draw-}\mathcal{R}$ the distribution of $t_{k\ell}$ is equal to $F_{\mathcal{G},\ell}$, independently of each element in $\mathcal{R}^-_{k\ell}$. ☐

Condition (B2): Replica indistinguishability.

Proof. Fix output $\mathcal{G} = (G_{k\ell})_{\ell\in[m]}$ of $\textsc{Preprocess}$. Conditioned on $\mathcal{G}$ and $\textsc{Flag}_{\mathcal{R}} = \texttt{True}$, the elements of $\mathcal{R}_{k\ell}$ are i.i.d samples from $F_{\mathcal{G},\ell}$ by Claim 5. $\textsc{Draw-}\mathcal{R}$ then returns $\mathcal{R}_k := \times_{\ell\in[m]}\mathcal{R}_{k\ell}$. Since the sets $\mathcal{R}_{k\ell}$ are independent (conditioned on $\mathcal{G}$) over ℓ, each $r_i \in \mathcal{R}_k$ is distributed according to $\times_{\ell\in[m]}F_{\mathcal{G},\ell} = F_{\mathcal{G}}$. Moreover, because the samples in each $\mathcal{R}_{k\ell}$ are i.i.d for all $\ell \in [m]$, and because $\mathcal{R}_k$ is formed as $\mathcal{R}_k := \times_\ell \mathcal{R}_{k\ell}$, the distribution of any replica r_i in $\mathcal{R}_k$ is equivalently expressed as a uniform random variable over $\mathcal{R}_k$. In particular, this is true of the bidder's type t_k, i.e. the replica r_{i^*}, as desired. Observe that we can uncondition on $\mathcal{G}$ for this latter claim in this condition. ☐

Condition (B3): High-cardinality matching.

For convenience in the proof, we imagine copying the surrogates $\mathcal{S}_k$ κ times, to form the *copied surrogates* $\bar{\mathcal{S}}_k$. We can accomplish this by creating $\frac{N_R}{N_S}$ copies of each surrogate marginal multiset $\mathcal{S}_{k\ell}$ (defined in $\textsc{Draw-}\mathcal{S}$) to form the *copied surrogate marginals* $\bar{\mathcal{S}}_{k\ell}$ (a multiset of size N_R), which we think of as the concatenation of $\mathcal{S}_{k\ell}$ $\frac{N_R}{N_S}$ times: $\bar{\mathcal{S}}_{k\ell} = (\mathcal{S}_{k\ell}, \mathcal{S}_{k\ell}, \ldots, \mathcal{S}_{k\ell})$. We then define $\bar{\mathcal{S}}_k := \times_{\ell\in[m]}\bar{\mathcal{S}}_{k\ell}$. The purpose of this copying is to reframe any κ-to-1 matching from $\mathcal{R}_k$ to $\mathcal{S}_k$ as a 1-to-1 matching from $\mathcal{R}_k$ to $\bar{\mathcal{S}}_k$ in the natural way.[46]

Proof. Consider the surrogates $\mathcal{S}_k$ outputted by $\textsc{Draw-}\mathcal{S}$ and corresponding copied surrogates $\bar{\mathcal{S}}_k$; the output $\mathcal{G}$ of $\textsc{Preprocess}$ and corresponding coupling $c_{\mathcal{G}}(\cdot)$; and the replicas $\mathcal{R}_k$ outputted by $\textsc{Draw-}\mathcal{R}$. Condition on the event $\textsc{Flag}_{\mathcal{R}} = \texttt{True}$ and on the particular global state $\mathcal{G}$.

We consider the following procedure for realizing the couples $c_{\mathcal{G}}(\mathcal{R}_k)$. Recall that $\mathcal{R}_k := \times_{\ell\in[m]}\mathcal{R}_{k\ell}$ in $\textsc{Draw-}\mathcal{R}$ and that $c_{\mathcal{G}} := \times_{\ell\in[m]}c_{\mathcal{G},\ell}$. To draw $c_{\mathcal{G}}(\mathcal{R}_k)$, draw independently for each $\ell \in [m]$ *marginal couples* $c_{\mathcal{G},\ell}(\mathcal{R}_{k\ell}) = (c_{\mathcal{G},\ell}(x_1), \ldots, c_{\mathcal{G},\ell}(x_{N_R}))$, $\forall x_i \in \mathcal{R}_{k\ell}$, where each marginal couple $c_{\mathcal{G},\ell}(x_i)$ is drawn independently. Then return the product set $\times_{\ell\in[m]}c_{\mathcal{G},\ell}(\mathcal{R}_{k\ell})$ as $c_{\mathcal{G}}(\mathcal{R}_k)$.

To verify that this is a valid way to realize the coupling $c_{\mathcal{G}}(\mathcal{R}_k)$, fix any $r_i = (r_{i,1}, \ldots, r_{i,n}) \in \mathcal{R}_k$. Observe that the couple $c_{\mathcal{G}}(r_i)$ is obtained by independently drawing the marginal couples $c_{\mathcal{G},\ell}(r_{i,1}), \ldots, c_{\mathcal{G},\ell}(r_{i,m})$ and setting $c_{\mathcal{G}}(r_i) = (c_{\mathcal{G},\ell}(r_{i,1}), \ldots, c_{\mathcal{G},\ell}(r_{i,m}))$ when the product set is formed. This is what drawing $c_{\mathcal{G}}(r_i)$ means for product coupling $c_{\mathcal{G}}$.[47]

We want to argue that the expected size of the maximal (1-to-1) matching M between d-close replica couples $c_{\mathcal{G}}(\mathcal{R}_k)$ and copied surrogates $\bar{\mathcal{S}}_k$ is large. To do this, we will apply the following technical lemma introduced for a similar high-cardinality matching argument in Hartline et al. [HKM11], adapted below.

Lemma C.3 (High-cardinality matching on i.i.d vertices [HKM11])**.** *Consider a bipartite graph (A, B) where the vertex sets have size $|A| = |B| = N$, and each vertex is associated with a value drawn i.i.d from some distribution Q. Suppose $\mathcal{B}$ is a partition of supp(Q) of size $b := |\mathcal{B}|$. Let M denote the maximum-cardinality matching between A and B that only matches nodes with realized*

[46] This procedure is only imagined for analysis and does not actually need to be run.

[47] Note that the realization of $c_{\mathcal{G}}(\mathcal{R}_k)$ is not required anywhere to be independent *across the replicas*; indeed, the realization given here is compatible with the proof of Condition (B4).

values in the same partite set of $\mathcal{B}$, and let $|M| \in [N]$ denote the size of this matching. Then $\mathbb{E}_Q[N - |M|] \leq \sqrt{bN}$.

Unfortunately, we cannot apply Theorem C.3 directly on $A := c_{\mathcal{G}}(\mathcal{R}_k)$ and $B := \bar{\mathcal{S}}_k$ because the replicas and surrogates are *not* i.i.d draws. However, the underlying replica marginal couples and surrogate marginals *are* i.i.d draws. In particular, observe that conditioned on $\mathcal{G}$ and $\textsc{Flag}_{\mathcal{R}} = \texttt{True}$, for each $\ell \in [m]$, the replica marginal couples $c_{\mathcal{G},\ell}(\mathcal{R}_{k\ell})$ are i.i.d draws from $D'_{k\ell}$, as the elements of $\mathcal{R}_{k\ell}$ are i.i.d draws from $F_{\mathcal{G},\ell}$ by Claim 5. Moreover, for each $\ell \in [m]$, the surrogate marginals $\mathcal{S}_{k\ell}$ are i.i.d draws from $D'_{k\ell}$ by construction in $\textsc{Draw}$-$\mathcal{S}$, and this remains true even conditioned on $\mathcal{G}$ and $\textsc{Flag}_{\mathcal{R}} = \texttt{True}$, as these events are independent of the $\textsc{Draw}$-$\mathcal{S}$ random process.

We will thus proceed by applying Theorem C.3 on the relevant marginals, partitioned into sets of close marginal types, and then considering a replica couple as matched to a surrogate iff each of their marginals are matched. Note that in order to apply the lemma on independent draws, we must consider the copied surrogate marginals $\bar{\mathcal{S}}_{k\ell}$ in blocks, one copy of $\mathcal{S}_{k\ell}$ at a time.

Hence we define M as follows. Fix the replicas $\mathcal{R}_k$, surrogates $\mathcal{S}_k$, and realized replica couples $c_{\mathcal{G}}(\mathcal{R}_k)$. Fix some $\ell \in [m]$. Split the replica marginal couples $A_\ell := c_{\mathcal{G},\ell}(\mathcal{R}_{k\ell})$ into $h := \frac{N_R}{N_S}$ blocks $A_{\ell,1}, \ldots, A_{\ell,h}$, where each $|A_{\ell,i}| = N_S$. Split the copied surrogate marginals $B := \bar{\mathcal{S}}_{k\ell}$ into h blocks $B_{\ell,1}, \ldots, B_{\ell,h}$ (such that each $B_{\ell,i}$ is a copy of $\mathcal{S}_{k\ell}$). Define $d' := \frac{d}{m}$. Consider the partition $\mathcal{B}$ of $[0,1]$ defined by

$$\mathcal{B} := \{[0, d'), [d', 2d'), \ldots, [(b-1) \cdot d', 1]\},$$

of size $b := |\mathcal{B}| \leq \frac{1}{d'} + 1$. For each i, let $M_{\ell,i}$ denote the maximum-cardinality (1-to-1) matching between $A_{\ell,i}$ and $B_{\ell,i}$ that matches some $x \in A_{\ell,i}$ and $y \in B_{\ell,i}$ iff they are in the same interval (partite set) in $\mathcal{B}$ (noting that $x, y \in [0,1]$). Then define M_ℓ to be the matching on (A_ℓ, B_ℓ) obtained as the union of the matchings $M_{\ell,i}$ over all $i \in [h]$. Finally, let M be the matching between the replica couples $c_{\mathcal{G}}(\mathcal{R}_k)$ and surrogates $\mathcal{S}_k$ that matches some $c_{\mathcal{G}}(r_i) = (c_{\mathcal{G},\ell}(x_1), \ldots, c_{\mathcal{G},\ell}(x_m))$ and some $s_j = (y_1, \ldots, y_m)$ iff each $c_{\mathcal{G},\ell}(x_\ell)$ and y_ℓ are matched under M_ℓ, $\forall \ell$.

Observe that the resulting M is indeed a 1-to-1 matching. Moreover, any matched $c_{\mathcal{G}}(r_i)$ and s_j are d-close, as

$$\mathrm{dist}_k(c_{\mathcal{G}}(r_i), s_j) = \max_{o \in O} |v_k(c_{\mathcal{G}}(r_i), o) - v_k(s_j, o)| \leq m \cdot d' \leq d,$$

which follows by the Lipschitz-ness of valuations and because the values in each coordinate of $c_{\mathcal{G}}(r_i)$ and s_j are within d' of each other by construction of $\mathcal{B}$ and M_ℓ.

Finally, we bound the cardinality of M in expectation (conditioned on $\textsc{Flag}_{\mathcal{R}} = \texttt{True}$ and the particular draw $\mathcal{G}$). In the derivation below, we use the fact that for each $\ell \in [m]$ and $i \in [h]$, the sets $A_{\ell,i}$ and $B_{\ell,i}$ each consist of N_S i.i.d draws from $D'_{k\ell}$ in order to apply Theorem C.3.

$$\begin{aligned}
\mathbb{E}\left[|M| \;\middle|\; \mathcal{G}, \textsc{Flag}_{\mathcal{R}} = \texttt{True}\right] &= \mathbb{E}\left[\Pi_{\ell=1}^m |M_\ell| \;\middle|\; \mathcal{G}, \textsc{Flag}_{\mathcal{R}} = \texttt{True}\right] && \text{by construction of } M \\
&= \mathbb{E}\left[\Pi_{\ell=1}^m \left(\sum_{i=1}^h |M_{\ell,i}|\right) \;\middle|\; \mathcal{G}, \textsc{Flag}_{\mathcal{R}} = \texttt{True}\right] && \text{by construction of } M_\ell \\
&= \Pi_{\ell=1}^m \left(\sum_{i=1}^h \mathbb{E}\left[|M_{\ell,i}| \;\middle|\; \mathcal{G}, \textsc{Flag}_{\mathcal{R}} = \texttt{True}\right]\right) && \text{by independence over } \ell
\end{aligned}$$

$$
\begin{aligned}
&\geq \Pi_{\ell=1}^{m}\left(\sum_{i=1}^{h} N_S\left(1-\sqrt{\frac{b}{N_S}}\right)\right) && \text{by Theorem C.3}\\
&= (N_R)^m \cdot \left(1-\sqrt{\frac{b}{N_S}}\right)^m\\
&\geq R \cdot \left(1-\sqrt{\frac{2m}{d}\cdot\frac{1}{N_S}}\right)^m\\
&\geq R \cdot (1-\Delta_{\textsc{Match}})
\end{aligned}
$$

where the last line follows by the choices of $\Delta_{\textsc{Match}} = O(\varepsilon)$, $N_S = \frac{m^3 n}{\varepsilon^3}$, and $d = O(\varepsilon)$. The expectation is the randomness of $t_k \sim D_k$; internal randomness of Draw-$\mathcal{S}$, Preprocess, and Draw-$\mathcal{R}$; and the realization of the coupling $c_{\mathcal{G}}(\mathcal{R}_k)$. [check the randomness claims]

In particular, this implies that the expected size $\mathbb{E}[|M^*| \mid \textsc{Flag}_{\mathcal{R}} = \texttt{True}]$ of the *maximum-cardinality* d-close matching $|M^*|$ on $\mathcal{R}_k$ and $\bar{\mathcal{S}}_k$, conditioned on $\textsc{Flag}_{\mathcal{R}} = \texttt{True}$, is at least $\mathbb{E}[|M| \mid \textsc{Flag}_{\mathcal{R}} = \texttt{True}] = \mathbb{E}_{\mathcal{G}}[\mathbb{E}[|M| \mid \mathcal{G}, \textsc{Flag}_{\mathcal{R}} = \texttt{True}]] \geq R(1-\Delta_{\textsc{Match}})$. □

Condition (B4).

Proof. Recall that for a fixed $\mathcal{G}$, the coupling $c_{\mathcal{G}}$ is product coupling $c_{\mathcal{G}} := \times_{\ell\in[m]} c_{\mathcal{G},\ell}$, where each $c_{\mathcal{G},\ell}$ is the composed coupling $c_{\mathcal{G},\ell} := c_\ell \circ \bar{c}_{\mathcal{G},\ell}$. First, notice that the product and composition operations on couplings (as defined in Section 2) commute: that is, we can write $c_{\mathcal{G}} = c \circ \bar{c}_{\mathcal{G}}$, where c is the product coupling $c := \times_{\ell\in[m]} c_\ell$ between D_k and D'_k, and $\bar{c}_{\mathcal{G}}$ is the product couplings $\bar{c}_{\mathcal{G}} := \times_{\ell\in[m]} \bar{c}_{\mathcal{G},\ell}$ between $F_{\mathcal{G}}$ and D_k.

We condition on $\textsc{Flag}_{\mathcal{R}} = \texttt{True}$ in the rest of this analysis; for brevity, we use $\Phi_{\text{F-}\mathcal{R}}$ to denote the event $\textsc{Flag}_{\mathcal{R}} = \texttt{True}$. By the above observation, we can write

$$
\begin{aligned}
\mathbb{E}\left[\sum_{r_i\in\mathcal{R}_k} \mathrm{dist}_k(r_i, c_{\mathcal{G}}(r_i)) \,\middle|\, \Phi_{\text{F-}\mathcal{R}}\right] &= \mathop{\mathbb{E}}_{\mathcal{G}}\left[\sum_{i\in R} \mathbb{E}\left[\mathrm{dist}_k(r_i, c(\bar{c}_{\mathcal{G}}(r_i))) \mid \mathcal{G}, \Phi_{\text{F-}\mathcal{R}}\right] \,\middle|\, \Phi_{\text{F-}\mathcal{R}}\right]\\
&\leq \mathop{\mathbb{E}}_{\mathcal{G}}\left[\sum_{i\in R} \mathbb{E}\left[\mathrm{dist}_k(r_i, \bar{c}_{\mathcal{G}}(r_i)) \mid \mathcal{G}, \Phi_{\text{F-}\mathcal{R}}\right] \,\middle|\, \Phi_{\text{F-}\mathcal{R}}\right]\\
&\quad + \mathop{\mathbb{E}}_{\mathcal{G}}\left[\sum_{i\in R} \mathbb{E}\left[\mathrm{dist}_k(\bar{c}_{\mathcal{G}}(r_i), c(\bar{c}_{\mathcal{G}}(r_i))) \mid \mathcal{G}, \Phi_{\text{F-}\mathcal{R}}\right] \,\middle|\, \Phi_{\text{F-}\mathcal{R}}\right].
\end{aligned}
$$

The unlabeled expectations are over the draw $t_k \sim D_k$ and randomness in Preprocess, Draw-$\mathcal{R}$, and the realization of $c_{k,\mathcal{G}}(\mathcal{R}_k)$. We use the law of total expectation and linearity of expectation in the first line, and the Triangle Inequality in the second, noting that $\mathrm{dist}_k(\cdot,\cdot)$ is a metric.

We now bound the two terms on the final RHS separately, starting with the second term. For arbitrary $i \in [R]$, by Condition (B2), the distribution of r_i (conditioned on $\mathcal{G}$, $\textsc{Flag}_{\mathcal{R}} = \texttt{True}$) is equal to $F_{\mathcal{G}}$ and so the distribution of $c_{\mathcal{G}}(r_i)$ is equal to D_k. Thus we have that

$$
\mathbb{E}\left[\mathrm{dist}_k(\bar{c}_{\mathcal{G}}(r_i), c(\bar{c}_{\mathcal{G}}(r_i))) \mid \mathcal{G}, \Phi_{\text{F-}\mathcal{R}}\right] = \mathop{\mathbb{E}}_{t\sim D_k}\left[\mathrm{dist}_k(t, c(t)) \mid \mathcal{G}, \Phi_{\text{F-}\mathcal{R}}\right]
$$

$$
\begin{aligned}
&= \mathop{\mathbb{E}}_{t\sim D_k}\left[\mathrm{dist}_k(t, c(t))\right] \\
&= d_c(D_k, D'_k) \\
&= d^W_{\mathrm{prod}}\left(D_k, D'_k\right)
\end{aligned}
$$

where the last line follows by choice of c. Thus

$$
\mathop{\mathbb{E}}_{\mathcal{G}}\left[\sum_{i\in R}\mathbb{E}\left[\mathrm{dist}_k(\bar{c}_{\mathcal{G}}(r_i), c(\bar{c}_{\mathcal{G}}(r_i))) \mid \mathcal{G}, \Phi_{\text{F-}\mathcal{R}}\right] \;\middle|\; \Phi_{\text{F-}\mathcal{R}}\right] = R\cdot d^W_{\mathrm{prod}}\left(D_k, D'_k\right).
$$

Now consider the first term, $\sum_{i\in[R]}\mathbb{E}_{\mathcal{G}}\left[\mathbb{E}\left[\mathrm{dist}_k(r_i, \bar{c}_{\mathcal{G}}(r_i)) \mid \mathcal{G}, \Phi_{\text{F-}\mathcal{R}}\right] \mid \Phi_{\text{F-}\mathcal{R}}\right]$. We begin by defining the event Good over the random process Preprocess to be the event where $\mathcal{G}$ satisfies the guarantees stated in Theorem C.2. In particular, Theorem C.2 tells us that $\Pr[\textsc{Good}] \geq 1-\exp(-\frac{nm}{\varepsilon})$, where the probability is over the draw $\mathcal{G} \leftarrow \textsc{Preprocess}$.

Consider arbitrary $i \in [R]$. We start as follows:

$$
\begin{aligned}
\mathop{\mathbb{E}}_{\mathcal{G}}\left[\mathbb{E}\left[\mathrm{dist}_k(r_i, \bar{c}_{\mathcal{G}}(r_i)) \mid \mathcal{G}, \Phi_{\text{F-}\mathcal{R}}\right] \mid \Phi_{\text{F-}\mathcal{R}}\right] &= \Pr[\textsc{Good}|\Phi_{\text{F-}\mathcal{R}}]\cdot\mathbb{E}\left[\mathrm{dist}_k(r_i, \bar{c}_{\mathcal{G}}(r_i)) \mid \textsc{Good}, \Phi_{\text{F-}\mathcal{R}}\right] \\
&\quad + (1-\Pr[\textsc{Good}|\Phi_{\text{F-}\mathcal{R}}])\cdot\mathbb{E}\left[\mathrm{dist}_k(r_i, \bar{c}_{\mathcal{G}}(r_i)) \mid (\neg\textsc{Good}), \Phi_{\text{F-}\mathcal{R}}\right] \\
&\leq \mathbb{E}\left[\mathrm{dist}_k(r_i, \bar{c}_{\mathcal{G}}(r_i)) \mid \textsc{Good}, \Phi_{\text{F-}\mathcal{R}}\right] + (1-\Pr[\textsc{Good} \mid \Phi_{\text{F-}\mathcal{R}}]) \\
&\leq \mathbb{E}\left[\mathrm{dist}_k(r_i, \bar{c}_{\mathcal{G}}(r_i)) \mid \textsc{Good}, \Phi_{\text{F-}\mathcal{R}}\right] + O\left(\frac{\varepsilon}{n}\right)
\end{aligned}
$$

The second line follows by bounding $\Pr[\textsc{Good} \mid \Phi_{\text{F-}\mathcal{R}}] \leq 1$ and $\mathbb{E}\left[\mathrm{dist}_k(r_i, \bar{c}_{\mathcal{G}}(r_i)) \mid (\neg\textsc{Good}), \Phi_{\text{F-}\mathcal{R}}\right] \leq 1$ since $\mathrm{dist}_k(\cdot,\cdot) \in [0,1]$ always. The third line follows by bounding $(1-\Pr[\textsc{Good}|\Phi_{\text{F-}\mathcal{R}}]) \leq (1-\Pr[\textsc{Good}\cap\Phi_{\text{F-}\mathcal{R}}]) \leq (1-\Pr[\textsc{Good}]) + (1-\Pr[\Phi_{\text{F-}\mathcal{R}}]) = O\left(\frac{\varepsilon}{n}\right)$, using Theorem C.2 and Condition (B1) (and the choice of $\delta_{\text{D-R}}$).

It remains to bound $\mathbb{E}\left[\mathrm{dist}_k(r_i, \bar{c}_{\mathcal{G}}(r_i)) \mid \textsc{Good}, \Phi_{\text{F-}\mathcal{R}}\right]$, which we write as

$$
\mathop{\mathbb{E}}_{\mathcal{G}}\left[\mathbb{E}\left[\mathrm{dist}_k(r_i, \bar{c}_{\mathcal{G}}(r_i)) \mid \mathcal{G}, \textsc{Good}, \Phi_{\text{F-}\mathcal{R}}\right] \mid \textsc{Good}, \Phi_{\text{F-}\mathcal{R}}\right].
$$

Fix a particular $\mathcal{G} = (G_{k\ell})_\ell$. Recall the construction of coupling $\bar{c}_{\mathcal{G}} = \times_{\ell\in[m]}\bar{c}_{\mathcal{G},\ell}$, where we define $p(G_{k\ell}) := \Pr_{y\sim D_{k\ell}}[\lfloor y\rceil_\eta \in G_{k\ell}]$. The key observation about $\bar{c}_{\mathcal{G}}$ is the following: for a coupled draw $(x,y)\sim\bar{c}_{\mathcal{G}}$, except with probability at most $\sum_{\ell\in[m]}(1-p(G_{k\ell}))$ (via union bound), $y = x$; in this case, $\mathrm{dist}_k(x,y) = 0$. It follows that

$$
\mathbb{E}\left[\mathrm{dist}_k(r_i, \bar{c}_{\mathcal{G}}(r_i)) \mid \mathcal{G}, \textsc{Good}, \Phi_{\text{F-}\mathcal{R}}\right] \leq \mathbb{E}\left[\sum_{\ell\in[m]}(1-p(G_{k\ell})) \mid \mathcal{G}, \textsc{Good}, \Phi_{\text{F-}\mathcal{R}}\right]
$$

The statement of Theorem C.2 tells us that conditioned on event Good, $1-p(G_{k\ell}) \leq \frac{\varepsilon}{nm}$ for all $\ell\in[m]$. Thus $\mathbb{E}\left[\mathrm{dist}_k(r_i, \bar{c}_{\mathcal{G}}(r_i)) \mid \mathcal{G}, \textsc{Good}, \Phi_{\text{F-}\mathcal{R}}\right] \leq \frac{\varepsilon}{n}$. Putting everything together, we can bound our original first term as

$$
\sum_{i\in[R]}\mathop{\mathbb{E}}_{\mathcal{G}}\left[\mathbb{E}\left[\mathrm{dist}_k(r_i, \bar{c}_{\mathcal{G}}(r_i)) \mid \mathcal{G}, \Phi_{\text{F-}\mathcal{R}}\right] \mid \Phi_{\text{F-}\mathcal{R}}\right] \leq R\cdot\left(\frac{\varepsilon}{n}\right).
$$

Summing up, we conclude that

$$\mathbb{E}\left[\sum_{r_i \in \mathcal{R}_k} \mathrm{dist}_k(r_i, c_{k,\mathcal{G}}(r_i)) \;\middle|\; \Phi_{\text{F-}\mathcal{R}}\right] \le R \cdot \left(d^W_{\text{prod}}\left(D_k, D'_k\right) + O\left(\frac{\varepsilon}{n}\right)\right) = R \cdot \Delta^{(k)}_{\textsc{Couple}}.$$

□

C.4 Compute-α

Condition (C1): Approximate equal demand. To prove this claim we will condition on various high-probability events over the course of the analysis. At the end, we will tally up the failure probabilities into $\delta_{\textsc{Demand}}$. First, condition on the event (over the randomness of Preprocess) that the output $(G_{k\ell})_\ell$ of Preprocess satisfies the claims in Theorem C.2; this event occurs except w.p. at most $\exp(-\frac{nm}{\varepsilon})$. We fix $(G_{k\ell})_\ell$ for the remainder of the proof. Furthermore, we define $\varepsilon_0 := \frac{\varepsilon^7}{m^7 n^4}$, $\delta_0 := \frac{\varepsilon}{nm}$, and $\Delta_{\textsc{Edge}} := \frac{\varepsilon^3}{m^2 n}$.

<u>Step 1: Discretized $\mathcal{R}_k$ and $\mathcal{R}_k^{\text{Tr}}$ are close (as empirical distributions).</u>

[check reasoning]Let $F_\ell := (\lfloor D_{k\ell}\rfloor_\eta \mid G_{k\ell})$ denote the distribution of a sample drawn from $\lfloor D_{k\ell}\rfloor_\eta$ conditioned on membership in $G_{k\ell}$. Consider the sets $(\mathcal{R}_{k\ell})_{\ell\in[m]}$ from Draw-$\mathcal{R}$ and sets $(\mathcal{R}'_{k\ell})_\ell$ from Compute-α. $\mathcal{R}^{\text{Tr}}_{k\ell}$ Let $\lfloor\mathcal{R}_{k\ell}\rfloor_\eta$ and $\lfloor\mathcal{R}'_{k\ell}\rfloor_\eta$ denote the sets $\mathcal{R}_{k\ell}$ and $\mathcal{R}'_{k\ell}$ with each element x rounded down to $\lfloor x\rfloor_\eta$, respectively. Condition on $\textsc{Flag}^{(\ell)'}_{\textsc{Draw}} = \texttt{True}$ $\forall \ell \in [m]$ in Compute-α, and $\textsc{Flag}_{\mathcal{R}} = \texttt{True}$ in Draw-$\mathcal{R}$; this event occurs except w.p. at most $\delta_{\text{D-R}} + m\cdot\exp(-\Omega(N_R)) = O\left(\frac{\varepsilon}{n}\right)$. Then observe via Claim 5 (and considering the additional rounding in F_ℓ) that conditioned on $\textsc{Flag}_{\mathcal{R}} = \texttt{True}$, the distribution of each element $x \in \lfloor\mathcal{R}_{k\ell}\rfloor_\eta$ is given by F_ℓ, for all $\ell \in [m]$. Similarly, conditioned on $\textsc{Flag}^{(\ell)'}_{\textsc{Draw}} = \texttt{True}$ for each $\ell \in [m]$, the distribution of each element $x' \in \lfloor\mathcal{R}'_{k\ell}\rfloor_\eta$ is also given by $\mathcal{F}_\ell$, for all $\ell \in [m]$.

Let $\tilde{D}_{k\ell}$ denote the uniform distribution over $\lfloor\mathcal{R}_{k\ell}\rfloor_\eta$, for each $\ell \in [m]$. That is, $\tilde{D}_{k\ell}$ is the empirical distributions defined by N_R draws from F_ℓ (with $\mathrm{supp}(F_\ell) \subseteq G_{k\ell}$). We now apply a standard result on learning discrete distributions (Theorem C.7). Since $N_R = \frac{m^{15}n^9}{\varepsilon^{15}} \ge \log(\frac{4}{\eta\cdot\delta_0})/\varepsilon_0{}^2$, we have that except with probability at most δ_0, for all $y \in \mathrm{supp}(F_\ell)$,

$$\left|\Pr_{\tilde{D}_{k\ell}}[y] - \Pr_{F_\ell}[y]\right| < \varepsilon_0$$

For all $y' \in G_{k\ell}$, Theorem C.2 tells us that $\Pr_{y\sim F_\ell}[y = y'] \ge \frac{\varepsilon^6}{m^6 n^4}$ (which also means $\mathrm{supp}(\mathcal{F}_\ell) = G_{k\ell}$).[48] Since $\varepsilon_0 \le \frac{\varepsilon}{m}\cdot\frac{\varepsilon^6}{m^6 n^4}$, the above bound can be written as

$$\Pr_{\tilde{D}_{k\ell}}[y] \in \left(1 \pm \frac{\varepsilon}{m}\right)\cdot \Pr_{F_\ell}[y].$$

Now let $F = \times_{\ell\in[m]} F_\ell$. Let $\tilde{D}_k$ be the uniform distribution over the product set $\times_{\ell\in[m]}\lfloor\mathcal{R}_{k\ell}\rfloor_\eta$, which is the set $\mathcal{R}_k$ with the coordinates of elements rounded down to the nearest η multiple.

[48]Indeed, note that $\Pr_{y\sim(\lfloor D_{k\ell}\rfloor_\eta | G_{k\ell})}[y = y'] \ge \Pr_{y\sim\lfloor D_{k\ell}\rfloor_\eta}[y = y']$.

Notice that $\tilde{D}_k = \times_{\ell\in[m]}\tilde{D}_{k\ell}$. Via union bound, we have that except with probability at most $m\delta_0$,

$$\Pr_{\tilde{D}_k}[y] \in (1 \pm 2\varepsilon) \cdot \Pr_F[y].$$

The same reasoning applies to the sets $\lfloor \mathcal{R}'_{k\ell} \rfloor_\eta$, $\forall \ell \in [m]$. In particular, letting $\tilde{D}_k^{\mathrm{Tr}}$ be the uniform distribution over the product set $\times_{\ell\in[m]}\lfloor \mathcal{R}'_{k\ell} \rfloor_\eta$, we have that except with probability at most $m\delta_0$,

$$\Pr_{\tilde{D}_k^{\mathrm{Tr}}}[y] \in (1 \pm 2\varepsilon) \cdot \Pr_F[y]$$

This implies that except with probability at most $2m\delta_0$, it holds that for all $y \in \mathrm{supp}(F) = \mathrm{supp}(\tilde{D}_k) = \mathrm{supp}(\tilde{D}_k^{\mathrm{Tr}})$, $\Pr_{\tilde{D}_k}[y] \in (1 \pm 6\varepsilon) \Pr_{\tilde{D}_k^{\mathrm{Tr}}}[y]$. Condition on this event (referred to as $(\star)$) henceforth.

<u>Step 2: Relate sum of $[x^*_{ij} + y^*_{ij}]$ over $\mathcal{R}_k$ and $\mathcal{R}_k^{\mathrm{Tr}}$.</u>

Let $\hat{D}_k$ and $\hat{D}_k^{\mathrm{Tr}}$ denote the uniform distributions over $\mathcal{R}_k$ and $\mathcal{R}_k^{\mathrm{Tr}}$ respectively. Note that we can write $\hat{D}_k := \times_{\ell\in[m]}\hat{D}_{k\ell}$ and $\hat{D}_k^{\mathrm{Tr}} := \times_{\ell\in[m]}\hat{D}_{k\ell}^{\mathrm{Tr}}$, where each $\hat{D}_{k\ell}$ is the uniform distribution over $\mathcal{R}_{k\ell}$ and $\hat{D}_{k\ell}^{\mathrm{Tr}}$ is the uniform distribution over $\mathcal{R}'_{k\ell}$. Moreover, $\tilde{D}_{k\ell} = \lfloor \hat{D}_{k\ell} \rfloor_\eta$ and $\tilde{D}_{k\ell}^{\mathrm{Tr}} = \lfloor \hat{D}_{k\ell}^{\mathrm{Tr}} \rfloor_\eta$, where $\tilde{D}_{k\ell}^{\mathrm{Tr}}$ is the uniform distribution over $\lfloor \mathcal{R}'_{k\ell} \rfloor_\eta$. As always we have $S := |\mathcal{S}_k|$ and $R := |\mathcal{R}_k| = |\mathcal{R}_k^{\mathrm{Tr}}|$.

For any $(\alpha_j)_j$, for a type t, we define

$$z^*_{tj}(\alpha) := x^*_{tj}(\alpha) + y^*_{tj}(\alpha) = \frac{\exp\left(-\frac{\alpha_j}{\delta}\right) \cdot \left(1 + \exp\left(\frac{\hat{w}_{tj}}{\delta}\right)\right)}{\sum_{j'\in[S]} \exp\left(-\frac{\alpha_{j'}}{\delta}\right) \cdot \left(1 + \exp\left(\frac{\hat{w}_{tj'}}{\delta}\right)\right)},$$

for all $j \in [S]$. Here $\hat{w}_{tj}$ is defined according to Theorem 4.1 with respect to distribution $\hat{D}' := \times_{k'\in[n]}\hat{D}'_{k'}$, where each $\hat{D}'_{k'}$ is the empirical distribution over $\mathcal{S}_{k'}$.

Consider the duals $\hat{\alpha}$ returned by COMPUTE-α. Fix any surrogate $s_j \in \mathcal{S}_k$. For any $t \in [0,1]^m$, we will use the notation $\lfloor t \rfloor_\eta$ to mean the vector $(\lfloor t_\ell \rfloor_\eta)_{\ell\in[m]}$. We start by bounding the difference between $z^*_{tj}(\hat{\alpha})$ and $z^*_{t'j}(\hat{\alpha})$ for any two types t, t' s.t. $\lfloor t \rfloor_\eta = \lfloor t' \rfloor_\eta$. Since valuations are Lipschitz, $\mathrm{dist}_k(t,t') \le m \cdot \eta \Rightarrow |\hat{w}_{tj}(\hat{\alpha}) - \hat{w}_{t'j}(\hat{\alpha})| \le m \cdot \eta$. Then we have

$$\begin{aligned}
z^*_{t'j}(\hat{\alpha}) &= \frac{\exp\left(-\frac{\hat{\alpha}_j}{\delta}\right) \cdot \left(1 + \exp\left(\frac{\hat{w}_{t'j}}{\delta}\right)\right)}{\sum_{j'} \exp\left(-\frac{\hat{\alpha}_{j'}}{\delta}\right) \cdot \left(1 + \exp\left(\frac{\hat{w}_{t'j'}}{\delta}\right)\right)} \\
&\le \frac{\exp\left(-\frac{\hat{\alpha}_j}{\delta}\right) \cdot \left(1 + \exp\left(\frac{\hat{w}_{tj}+m\eta}{\delta}\right)\right)}{\sum_{j'} \exp\left(-\frac{\hat{\alpha}_{j'}}{\delta}\right) \cdot \left(1 + \exp\left(\frac{\hat{w}_{tj'}-m\eta}{\delta}\right)\right)} \\
&\le \exp\left(\frac{2m\eta}{\delta}\right) \cdot \frac{\exp\left(-\frac{\hat{\alpha}_j}{\delta}\right) \cdot \left(1 + \exp\left(\frac{\hat{w}_{tj}}{\delta}\right)\right)}{\sum_{j'} \exp\left(-\frac{\hat{\alpha}_{j'}}{\delta}\right) \cdot \left(1 + \exp\left(\frac{\hat{w}_{tj'}}{\delta}\right)\right)} \\
&= \exp\left(\frac{2m\eta}{\delta}\right) \cdot z^*_{tj}(\hat{\alpha}).
\end{aligned}$$

We can now relate the expectation of $z^*_{tj}(\hat{\alpha})$ over $t \sim \hat{D}_k$ and $t \sim \hat{D}_k^{\mathrm{Tr}}$. Below, we denote

$U(t) := \{t' \in [0,1]^m : \lfloor t' \rfloor_\eta = \lfloor t \rfloor_\eta\}$ for $t \in [0,1]^m$.

$$\begin{aligned}
\mathop{\mathbb{E}}_{t\sim\hat{D}_k}[z^*_{tj}(\hat{\alpha})] &= \sum_{t\in\mathrm{supp}(\hat{D}_k)} \Pr_{\hat{D}_k}[t] \cdot z^*_{tj}(\hat{\alpha}) \\
&\leq \sum_{t\in\mathrm{supp}(\hat{D}_k)} \Pr_{\hat{D}_k}[t] \cdot \max_{t'\in U(t)} z^*_{t'j}(\hat{\alpha}) \\
&= \sum_{y\in\mathrm{supp}(F)} \Pr_{t\sim\hat{D}_k}[\lfloor t \rfloor_\eta = y] \cdot \max_{t'\in U(y)} z^*_{t'j}(\hat{\alpha}) \\
&= \sum_{y\in\mathrm{supp}(F)} \Pr_{\tilde{D}_k}[y] \cdot \max_{t'\in U(y)} z^*_{t'j}(\hat{\alpha}) \\
&\leq \sum_{y\in\mathrm{supp}(F)} (1+6\varepsilon) \Pr_{\tilde{D}_k^{\mathrm{Tr}}}[y] \cdot \max_{t'\in U(y)} z^*_{t'j}(\hat{\alpha}) \\
&\leq (1+6\varepsilon) \sum_{t\in\mathrm{supp}(\hat{D}_k^{\mathrm{Tr}})} \Pr_{\hat{D}_k^{\mathrm{Tr}}}[t] \cdot \max_{t'\in U(t)} z^*_{t'j}(\hat{\alpha}) \\
&\leq (1+6\varepsilon) \sum_{t\in\mathrm{supp}(\hat{D}_k^{\mathrm{Tr}})} \Pr_{\hat{D}_k^{\mathrm{Tr}}}[t] \cdot \exp\left(\frac{2m\eta}{\delta}\right) \cdot z^*_{tj}(\hat{\alpha}) \\
&= (1+6\varepsilon) \cdot \exp\left(\frac{2m\eta}{\delta}\right) \cdot \mathop{\mathbb{E}}_{t\sim\hat{D}_k^{\mathrm{Tr}}}[z^*_{tj}(\hat{\alpha})]
\end{aligned}$$

The fifth line follows by $(\star)$, and the penultimate line follows by the bound on $z^*_{tj}(\hat{\alpha})$ vs $z^*_{t'j}(\hat{\alpha})$ for $\lfloor t \rfloor_\eta = \lfloor t' \rfloor_\eta$. An analogous derivation in the opposite direction, using $\min_{t'\in U(t)} z^*_{t'j}(\hat{\alpha})$ instead of $\max_{t'\in U(t)} z^*_{t'j}(\hat{\alpha})$, yields $\mathbb{E}_{t\sim\hat{D}_k}[z^*_{tj}(\hat{\alpha})] \geq (1-6\varepsilon) \cdot \exp\left(-\frac{2m\eta}{\delta}\right) \cdot \mathbb{E}_{t\sim\hat{D}_k^{\mathrm{Tr}}}[z^*_{tj}(\hat{\alpha})]$. Interpreting the expectations as sums over $\mathcal{R}_k$ and $\mathcal{R}_k^{\mathrm{Tr}}$, we can write the above as follows: for any $s_j \in \mathcal{S}_k$,

$$\sum_{r_i\in\mathcal{R}_k} z^*_{ij}(\hat{\alpha}) \in \left[(1\pm 6\varepsilon) \cdot \exp\left(\pm\frac{2m\eta}{\delta}\right)\right]^{49} \cdot \sum_{r_i\in\mathcal{R}_k^{\mathrm{Tr}}} z^*_{ij}(\hat{\alpha}).$$

<u>Step 3: Relate $\sum_{r_i\in\mathcal{R}_k^{\mathrm{Tr}}} z^*_{ij}(\hat{\alpha})$ with analogous estimated edge weight terms.</u>

Recall the estimated edge weights $\tilde{w} = (\tilde{w}_{ij})_{i,j}$ outputted by the subroutine EstEdges. Define

$$\tilde{z}_{ij}(\hat{\alpha}) = \tilde{x}_{ij}(\hat{\alpha}) + \tilde{y}_{ij}(\hat{\alpha}) := \frac{\exp\left(-\frac{\hat{\alpha}_j}{\delta}\right) \cdot \left(1 + \exp\left(\frac{\tilde{w}_{ij}}{\delta}\right)\right)}{\sum_{p\in[S]} \exp\left(-\frac{\hat{\alpha}_p}{\delta}\right) \cdot \left(1 + \exp\left(\frac{\tilde{w}_{ip}}{\delta}\right)\right)},$$

i.e. the same form as $z^*_{ij}(\hat{\alpha})$ except replacing w_{ij} with $\tilde{w}_{ij}$. By Claim 6, except with probability at most $\exp(-\frac{nm}{\varepsilon})$, it holds that $\forall r_i \in \mathcal{R}_k^{\mathrm{Tr}}, s_j \in \mathcal{S}_k$, $|\tilde{w}_{ij} - \hat{w}_{ij}| \leq \Delta_{\textsc{Edge}}$. Condition on this event. It is then straightforward to see, like in previous derivations, that $\forall i,j$, $\tilde{z}_{ij}(\hat{\alpha}) \in \exp(\pm\frac{2\Delta_{\textsc{Edge}}}{\delta}) z^*_{ij}(\hat{\alpha})$. This implies that for all $s_j \in \mathcal{S}_k$,

[49] We use this notation to mean the interval $\left[(1-6\varepsilon)\cdot\exp\left(-\frac{2m\eta}{\delta}\right), (1+6\varepsilon)\cdot\exp\left(\frac{2m\eta}{\delta}\right)\right]$.

$$\sum_{r_i \in \mathcal{R}_k^{\mathrm{Tr}}} z^*_{ij}(\hat{\alpha}) \in \exp\left(\pm \frac{2\Delta_{\textsc{Edge}}}{\delta}\right) \cdot \sum_{r_i \in \mathcal{R}_k^{\mathrm{Tr}}} \tilde{z}_{ij}(\hat{\alpha}).$$

<u>Step 4: Relate $\tilde{z}_{ij}(\hat{\alpha})$ and $\tilde{z}_{ij}(\tilde{\alpha})$.</u>

First note that the shift $\tilde{\alpha}_j \to \tilde{\alpha}_j - \tilde{\alpha}^0$ for all j has no effect on $\tilde{z}_{ij}(\tilde{\alpha})$: that is, $\tilde{z}_{ij}(\tilde{\alpha}) = z^*_{ij}(\tilde{\alpha} - \tilde{\alpha}^0)$. Thus we consider $\tilde{\alpha}$ to be such that $\min_{j'} \tilde{\alpha}_{j'} = 0$ WLOG. Next consider the truncation at A: observe that for any j,

$$\exp\left(-\frac{\tilde{\alpha}_j}{\delta}\right) \le \exp\left(-\frac{\hat{\alpha}_j}{\delta}\right) \le \exp\left(-\frac{A}{\delta}\right) + \exp\left(-\frac{\tilde{\alpha}_j}{\delta}\right).$$

Denote $C_j := 1 + \exp\left(\frac{\tilde{w}_{ij}}{\delta}\right)$ for brevity. Then, for any $j \in [S]$, it is a straightforward derivation to see that

$$\tilde{z}_{ij}(\hat{\alpha}) \le \frac{\exp\left(-\frac{A}{\delta}\right) \cdot C_j}{\sum_{p \in [S]} \exp\left(-\frac{\tilde{\alpha}_p}{\delta}\right) \cdot C_p} + \tilde{z}_{ij}(\tilde{\alpha}).$$

Let us analyze the RHS. Note that $C_j \ll \exp(\frac{2}{\delta})$, since $\tilde{w}_{ij} \le 1$, and that $A = 2\delta \log S + 4 = 2\varepsilon + 4 \ge 4$. Then the numerator $\exp\left(-\frac{A}{\delta}\right) \cdot C_j$ is at most $\exp\left(-\frac{A}{2\delta}\right)$. Furthermore, recall that there exists some p s.t. $\tilde{\alpha}_p = 0$ without loss of generality, so the denominator of the fraction in the RHS is at least 1. This means that $\tilde{z}_{ij}(\hat{\alpha}) \le \exp\left(-\frac{A}{2\delta}\right) + \tilde{z}_{ij}(\tilde{\alpha}) \le \frac{1}{S} \cdot \exp\left(-\frac{2}{\delta}\right) + \tilde{z}_{ij}(\tilde{\alpha})$, and so $\sum_{r_i \in \mathcal{R}_k^{\mathrm{Tr}}} \tilde{z}_{ij}(\hat{\alpha}) \le \frac{R}{S} \cdot \exp\left(-\frac{2}{\delta}\right) + \sum_{r_i \in \mathcal{R}_k^{\mathrm{Tr}}} \tilde{z}_{ij}(\tilde{\alpha})$.

To derive a corresponding lower bound, we start as follows:

$$\begin{aligned}
\sum_{p \in [S]} \left(\exp\left(-\frac{A}{\delta}\right) + \exp\left(-\frac{\tilde{\alpha}_p}{\delta}\right)\right) \cdot C_p &\le \sum_p \left(\exp\left(-\frac{A}{2\delta}\right) + \exp\left(-\frac{\tilde{\alpha}_p}{\delta}\right) \cdot C_p\right) \\
&\le \left(1 + S \cdot \exp\left(-\frac{A}{2\delta}\right)\right) \cdot \sum_p \exp\left(-\frac{\tilde{\alpha}_p}{\delta}\right) \cdot C_p \\
&= \left(1 + \exp\left(-\frac{2}{\delta}\right)\right) \cdot \sum_p \exp\left(-\frac{\tilde{\alpha}_p}{\delta}\right) \cdot C_p \\
&\le \frac{1}{\left(1 - \exp\left(-\frac{2}{\delta}\right)\right)} \cdot \sum_p \exp\left(-\frac{\tilde{\alpha}_p}{\delta}\right) \cdot C_p
\end{aligned}$$

where the second line follows since $\sum_p \exp\left(-\frac{\tilde{\alpha}_p}{\delta}\right) \cdot C_p \ge 1$.

Thus we can derive [check]

$$\begin{aligned}
\tilde{z}_{ij}(\hat{\alpha}) &\ge \frac{\exp\left(-\frac{\tilde{\alpha}_j}{\delta}\right) \cdot C_j}{\sum_p \left(\exp\left(-\frac{A}{\delta}\right) + \exp\left(-\frac{\tilde{\alpha}_p}{\delta}\right)\right) \cdot C_p} \\
&\ge \left(1 - \exp\left(-\frac{2}{\delta}\right)\right) \cdot \frac{\exp\left(-\frac{\tilde{\alpha}_j}{\delta}\right) \cdot C_j}{\sum_p \exp\left(-\frac{\tilde{\alpha}_p}{\delta}\right) \cdot C_p}
\end{aligned}$$

$$= \left(1 - \exp\left(-\frac{2}{\delta}\right)\right) \cdot \tilde{z}_{ij}(\tilde{\alpha})$$

and so $\sum_{r_i \in \mathcal{R}_k^{\mathrm{Tr}}} \tilde{z}_{ij}(\hat{\alpha}) \geq \left(1 - \exp\left(-\frac{2}{\delta}\right)\right) \cdot \sum_{r_i \in \mathcal{R}_k^{\mathrm{Tr}}} \tilde{z}_{ij}(\tilde{\alpha})$. We will use these bounds shortly.

Step 5. $\sum_{r_i \in \mathcal{R}_k^{\mathrm{Tr}}} \tilde{z}_{ij}(\tilde{\alpha}) = \frac{R}{S}$.

At last we can make the crucial observation about $\sum_{r_i \in \mathcal{R}_k^{\mathrm{Tr}}} \tilde{z}_{ij}(\tilde{\alpha})$: recall that $\tilde{\alpha}$ is computed as the *optimal* duals $\tilde{\alpha} := \arg\min_\alpha \mathbf{P}'_\delta(\mathcal{R}_k^{\mathrm{Tr}}, \mathcal{S}_k, \tilde{w}, \alpha)$, and $(\tilde{x}_{ij}, \tilde{y}_{ij})_{ij}$ is the optimizer of $\mathbf{P}'_\delta(\mathcal{R}_k^{\mathrm{Tr}}, \mathcal{S}_k, \tilde{w}, \tilde{\alpha})$. This implies that

$$\sum_{r_i \in \mathcal{R}_k^{\mathrm{Tr}}} \tilde{z}_{ij}(\tilde{\alpha}) = \sum_{r_i \in \mathcal{R}_k^{\mathrm{Tr}}} (\tilde{x}_{ij}(\tilde{\alpha}) + \tilde{y}_{ij}(\tilde{\alpha})) = \frac{R}{S}$$

for all j.

Putting the steps together. We can put the final claims of Steps 1 through 5 together to obtain the following upper bound (UB) and lower bound (LB): for all $j \in [S]$, it holds that

$$\mathbf{UB}: \sum_{r_i \in \mathcal{R}_k} z^*_{ij}(\hat{\alpha}) \leq (1 + 6\varepsilon) \cdot \left(1 + \exp\left(-\frac{2}{\delta}\right)\right) \cdot \exp\left(\frac{2m\eta + 2\Delta_{\mathrm{EDGE}}}{\delta}\right) \cdot \frac{R}{S}$$

$$\mathbf{LB}: \sum_{r_i \in \mathcal{R}_k} z^*_{ij}(\hat{\alpha}) \geq (1 - 6\varepsilon) \cdot \left(1 - \exp\left(-\frac{2}{\delta}\right)\right) \cdot \exp\left(-\frac{2m\eta + 2\Delta_{\mathrm{EDGE}}}{\delta}\right) \cdot \frac{R}{S}$$

From the UB, we see that for any λ s.t. $1 + \lambda \geq (1 + 6\varepsilon) \cdot \left(1 + \exp\left(-\frac{2}{\delta}\right)\right) \cdot \exp\left(\frac{2m\eta + 2\Delta_{\mathrm{EDGE}}}{\delta}\right)$, we have $\sum_{r_i \in \mathcal{R}_k} x^*_{ij}(\hat{\alpha}) + y^*_{ij}(\hat{\alpha}) \leq (1 + \lambda) \cdot \frac{R}{S}$. Observe that for any such λ, $1 - \lambda \leq (1 - 6\varepsilon) \cdot \left(1 - \exp\left(-\frac{2}{\delta}\right)\right) \cdot \exp\left(-\frac{2m\eta + 2\Delta_{\mathrm{EDGE}}}{\delta}\right)$.[50] This means that we also have the lower bound $\sum_{r_i \in \mathcal{R}_k} x^*_{ij}(\hat{\alpha}) + y^*_{ij}(\hat{\alpha}) \geq (1 - \lambda) \cdot \frac{R}{S}$ for any such λ.

Substituting $\delta = \varepsilon \cdot \left(m \log \frac{m^3 n}{\varepsilon^3}\right)^{-1}$, $\eta = \frac{\varepsilon^3}{m^3 n}$, and $\Delta_{\mathrm{EDGE}} = \frac{\varepsilon^3}{m^2 n}$, it is not hard to see that both $\left(1 + \exp\left(-\frac{2}{\delta}\right)\right) = 1 + O(\varepsilon)$ and $\exp\left(\frac{2m\eta + 2\Delta_{\mathrm{EDGE}}}{\delta}\right) = 1 + O(\varepsilon)$, which implies that

$$(1 + 6\varepsilon) \cdot \left(1 + \exp\left(-\frac{2}{\delta}\right)\right) \cdot \exp\left(\frac{2m\eta + 2\Delta_{\mathrm{EDGE}}}{\delta}\right) = 1 + O(\varepsilon).$$

Thus choosing $\lambda = O(\varepsilon)$ suffices to satisfy the claim.

Tallying up the failure probabilities of the events we conditioned on yields

$$\exp\left(-\Omega\left(\frac{nm}{\varepsilon}\right)\right) + O\left(\frac{\varepsilon}{n}\right) + 2m\delta_0 = O\left(\frac{\varepsilon}{n}\right),$$

so we can choose $\delta_{\mathrm{DEMAND}} = O\left(\frac{\varepsilon}{n}\right)$. This completes the proof of Condition (C1).

Condition (C2): Approximate max-weight. Let $R := |\mathcal{R}_k|, S := |\mathcal{S}_k|, \kappa := \frac{R}{S}$. Consider program $\mathbf{P}'_\delta(\mathcal{R}_k, \mathcal{S}_k, \hat{w}, \hat{\alpha})$ and let $\textsc{Opt}(\mathbf{P}')$ denote $\textsc{Opt}(\mathbf{P}'_\delta(\mathcal{R}_k, \mathcal{S}_k, \hat{w}, \hat{\alpha}))$. Let $\textsc{Opt}(\mathbf{P})$ denote

[50] This follows from observing that $(1+6\varepsilon) \cdot \left(1 + \exp\left(-\frac{2}{\delta}\right)\right) \cdot \exp\left(\frac{2m\eta + 2\Delta_{\mathrm{EDGE}}}{\delta}\right) \geq \left(1 + 6\varepsilon + \exp\left(-\frac{2}{\delta}\right) + \frac{2m\eta + 2\Delta_{\mathrm{EDGE}}}{\delta}\right)$, while $\left(1 - 6\varepsilon - \exp\left(-\frac{2}{\delta}\right) - \frac{2m\eta + 2\Delta_{\mathrm{EDGE}}}{\delta}\right) \leq (1 - 6\varepsilon) \cdot \left(1 - \exp\left(-\frac{2}{\delta}\right)\right) \cdot \exp\left(-\frac{2m\eta + 2\Delta_{\mathrm{EDGE}}}{\delta}\right)$.

$\textsc{Opt}(\mathbf{P}(\mathcal{R}_k, \mathcal{S}_k, \hat{w}))$. We first show that $\textsc{Opt}(\mathbf{P}') \geq \textsc{Opt}(\mathbf{P})$.[51]

Let $(x^P_{ij})_{i,j}$ denote the optimizer of the program $\mathbf{P}(\mathcal{R}_k, \mathcal{S}_k, \hat{w})$, that is, a max-weight matching on $(\mathcal{R}_k, \mathcal{S}_k)$. Observe that $(x^P_{ij})_{i,j}$ can be *extended* to a feasible 0-1 solution $(x^P_{ij}, y^P_{ij})_{i,j}$ to the entropy-regularized matching program given by $\mathbf{P}'_\delta(\mathcal{R}_k, \mathcal{S}_k, \hat{w}, \hat{\alpha})$, such that for all j, $\sum_i x^P_{ij} + y^P_{ij} = \kappa$.[52] Since $(x^P_{ij}, y^P_{ij})_{i,j}$ is a $\{0,1\}$-solution, the objective value of this feasible solution to $\mathbf{P}'_\delta(\mathcal{R}_k, \mathcal{S}_k, \hat{w}, \hat{\alpha})$ is exactly $\sum_{i,j} x^P_{ij}\hat{w}_{ij} = \textsc{Opt}(\mathbf{P})$. Thus $\textsc{Opt}(\mathbf{P}')$ is at least $\textsc{Opt}(\mathbf{P})$.

Now consider the optimizer $(x^*_{ij}(\hat{\alpha}), y^*_{ij}(\hat{\alpha}))_{i,j}$ of $\mathbf{P}'_\delta(\mathcal{R}_k, \mathcal{S}_k, \hat{w}, \hat{\alpha})$, and the resulting objective value $\textsc{Opt}(\mathbf{P}')$, which is given by

$$\sum_{i,j} x^*_{ij}(\hat{\alpha}) w_{ij} - \delta \sum_{i,j} (x^*_{ij}(\hat{\alpha}) \log x^*_{ij}(\hat{\alpha}) + y^*_{ij}(\hat{\alpha}) \log y^*_{ij}(\hat{\alpha})) + \sum_j \hat{\alpha}_j \left(\kappa - \sum_i (x^*_{ij}(\hat{\alpha}) + y^*_{ij}(\hat{\alpha})) \right).$$

Observe the following two bounds:

1. The term $-\delta \sum_{i,j}(x^*_{ij}(\hat{\alpha}) \log x^*_{ij}(\hat{\alpha}) + y^*_{ij}(\hat{\alpha}) \log y^*_{ij}(\hat{\alpha}))$ is upper bounded by $\delta R \log(2S)$, by the maximum entropy of a distribution (over a finite support of size $2S$).

2. By Condition (C1), we have that except with probability at most $\delta_{\textsc{Demand}}$ (over the randomness of $\textsc{Preprocess}$, $\textsc{Compute}$-α, $\textsc{Draw}$-$\mathcal{R}$, and $t_k \sim D_k$),

$$\sum_{r_i \in \mathcal{R}_k} x^*_{ij}(\hat{\alpha}) + y^*_{ij}(\hat{\alpha}) \geq (1 - \lambda)\kappa.$$

   Under this event (call it E), the term $\sum_j \hat{\alpha}_j \left(\kappa - \sum_i (x^*_{ij}(\hat{\alpha}) + y^*_{ij}(\hat{\alpha}))\right)$ is at most $\lambda \cdot A \cdot R$.

Notice that $\sum_{i,j}(1-\lambda) x^*_{ij}(\hat{\alpha})\hat{w}_{ij}$ is always in the range $[-R, R]$, since $\hat{w}_{ij} \in [-1, 1]$ (by bounded valuations and payments) and since $\sum_j x^*_{ij} \leq 1$. Hence

$$\begin{aligned}
\mathbb{E}[\sum_{i,j}(1-\lambda)x^*_{ij}(\hat{\alpha})\hat{w}_{ij}] &= \Pr[E]\, \mathbb{E}[\sum_{i,j}(1-\lambda)x^*_{ij}(\hat{\alpha})\hat{w}_{ij} | E] + \Pr[\neg E]\, \mathbb{E}[\sum_{i,j}(1-\lambda)x^*_{ij}(\hat{\alpha})\hat{w}_{ij} | \neg E] \\
&\geq (1 - \delta_{\textsc{Demand}})\, \mathbb{E}[\sum_{i,j}(1-\lambda)x^*_{ij}(\hat{\alpha})\hat{w}_{ij} | E] - \delta_{\textsc{Demand}} \cdot R \\
&\geq \mathbb{E}[\sum_{i,j} x^*_{ij}(\hat{\alpha})\hat{w}_{ij} | E] - 2 \cdot \delta_{\textsc{Demand}} \cdot R - \lambda \cdot R \\
&\geq \mathbb{E}[\textsc{Opt}(\mathbf{P}') - \delta R \log(2S) - \lambda \cdot A \cdot R] - 2 \cdot \delta_{\textsc{Demand}} \cdot R - \lambda \cdot R \\
&\geq \mathbb{E}[\textsc{Opt}(\mathbf{P})] - R \cdot (\delta \log(2S) + \lambda \cdot A + 2\delta_{\textsc{Demand}} + \lambda) \\
&\geq \mathbb{E}[\textsc{Opt}(\mathbf{P})] - O(R\varepsilon)
\end{aligned}$$

where the last line follows since $\delta = \varepsilon \log S$, $\lambda = O(\varepsilon)$, $A = O(1)$, and $\delta_{\textsc{Demand}} = O\left(\frac{\varepsilon}{n}\right)$. Thus our choice of $\Delta_{\text{MWM}} = O(\varepsilon)$ proves Condition (C2).

[51] This is similar to Lemma 7 in [COVZ19], and the proof is similar.

[52] To see this, iterate through $i : 1 \to R$ and do the following: if $x^P_{ij} = 0$ for all j, then set $y^P_{i\ell} = 1$ for a surrogate s_ℓ with minimum value of $\sum_{i' \in [R]} x^P_{i'j} + \sum_{i' < i} y^P_{i'j}$ over all surrogates, and all other $y^P_{ij} = 0$; else, if $\exists \ell$ s.t. $x^P_{i\ell} = 1$, set $y^P_{ij} = 0$ for all j. Since x^P satisfies $\sum_i x^P_{ij} \leq \kappa$ for all j, and there are $R = \kappa \cdot S$ replicas, it is easy to see that this iteration will ensure $\sum_i x^P_{ij} + y^P_{ij} = \kappa$ for all j.

C.5 Match

Condition (D1): The claim follows immediately from the Fast Exponential Bernoulli Race (Theorem 4.7). In particular, Theorem 4.7 implies that the implementation of MATCH outputs s_j with probability

$$\frac{\exp((\hat{w}_{ij} - \hat{\alpha}_j)/\delta)}{\sum_{j' \in [S]} \exp(-\hat{\alpha}_{j'}/\delta) \cdot (1 + \exp(\hat{w}_{ij'}/\delta)}$$

which is exactly $x^*_{ij}(\hat{\alpha})$ (with respect to weights $\hat{w}_{ij}$), and returns $\circ_j$ with probability

$$\frac{\exp(-\hat{\alpha}_j/\delta)}{\sum_{j' \in [S]} \exp(-\hat{\alpha}_{j'}/\delta) \cdot (1 + \exp(\hat{w}_{ij'}/\delta)}$$

which is exactly $y^*_{ij}(\hat{\alpha})$ (with respect to weights $\hat{w}_{ij}$), recalling Theorem 4.6 and Theorem 4.4. Moreover, note that running the Fast Exponential Bernoulli Race will use independent samples (fresh randomness) every time it is called, across the replicas $\mathcal{R}_k$.

C.6 Payment

Conditions (E1) - (E3): Conditions (E1), (E2), and (E3) are proved directly via Theorem C.4, Theorem C.5, and Theorem C.6, respectively, below. To see that Condition (E1) follows from Theorem C.4, note that the term $\delta \cdot \log \left(\sum_{j \in [S]} \exp \left(\frac{-\hat{\alpha}_j}{\delta} \right) \right)$ is a constant that does not depend on t_k. For brevity throughout the proofs, we abbreviate $x^*_j := x^*_{t_k j}$, $y^*_j := y^*_{t_k j}$, $x^\lambda_j := x^\lambda_{t_k j}$, $y^\lambda_j := y^\lambda_{t_k j}$, and $\hat{w}_j := \hat{w}_{t_k j}$.

Lemma C.4.

$$\mathop{\mathbb{E}}_{\lambda, s', s'', t_{-k}, \mathcal{A}', p'} [q_k] = \delta \sum_{j \in [S]} (x^*_j(\hat{\alpha}) \log x^*_j(\hat{\alpha}) + y^*_j(\hat{\alpha}) \log y^*_j(\hat{\alpha})) + \sum_{j \in [S]} \hat{\alpha}_j (x^*_j(\hat{\alpha}) + y^*_j(\hat{\alpha}))$$
$$+ \delta \cdot \log \left(\sum_{j \in [S]} \exp \left(\frac{-\hat{\alpha}_j}{\delta} \right) \right)$$

Proof.

$$\begin{aligned}
\mathop{\mathbb{E}}_{\lambda, s', s'', t_{-k}, \mathcal{A}', p'} [q_k] &= \mathop{\mathbb{E}}_{s', t_{-k}, \mathcal{A}', p'} [W_k(t_k, s', t_{-k})] - \mathop{\mathbb{E}}_{\lambda, s'', t_{-k}, \mathcal{A}', p'} [W_k(t_k, s'', t_{-k})] - \delta \log 2 \\
&= \sum_{j \in [S]} x^*_j(\hat{\alpha}) \cdot \hat{w}_j - \int_0^1 \sum_{j \in [S]} x^\lambda_j(\hat{\alpha}) \cdot \hat{w}_j \, \mathrm{d}\lambda - \delta \log 2 \\
&= \sum_{j \in [S]} x^*_j(\hat{\alpha}) \cdot \hat{w}_j - \delta \cdot \left[\log \left(\sum_j \exp \left(\frac{-\hat{\alpha}_j}{\delta} \right) (1 + \exp(\lambda \hat{w}_j/\delta)) \right) \right]_0^1 - \delta \log 2 \\
&= \sum_{j \in [S]} x^*_j(\hat{\alpha}) \cdot \hat{w}_j - \delta \cdot \log \left(\sum_j \exp \left(\frac{-\hat{\alpha}_j}{\delta} \right) (1 + \exp(\hat{w}_j/\delta)) \right) \\
&\quad + \delta \cdot \log \left(\sum_j \exp \left(\frac{-\hat{\alpha}_j}{\delta} \right) \right)
\end{aligned}$$

$$
\begin{aligned}
&= \delta \cdot \sum_{j\in[S]} x_j^*(\hat{\alpha}) \cdot \log \left(\frac{\exp\left(\frac{\hat{w}_j}{\delta}\right)}{\sum_{j'} \exp\left(\frac{-\hat{\alpha}_{j'}}{\delta}\right) \left(1+\exp(\hat{w}_{j'}/\delta)\right)} \right) \\
&\quad - \delta \cdot \log \left(\sum_j \exp\left(\frac{-\hat{\alpha}_j}{\delta}\right) (1+\exp(\hat{w}_j/\delta)) \right) \sum_j y_j^*(\hat{\alpha}) \\
&\quad + \delta \cdot \log \left(\sum_j \exp\left(\frac{-\hat{\alpha}_j}{\delta}\right) \right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \left(\textstyle\sum_j (x_j^*(\hat{\alpha}) + y_j^*(\hat{\alpha})) = 1\right) \\
&= \delta \cdot \sum_{j\in[S]} x_j^*(\hat{\alpha}) \log x_j^*(\hat{\alpha}) + \sum_j \hat{\alpha}_j x_j^*(\hat{\alpha}) + \delta \cdot \log \left(\sum_j \exp\left(\frac{-\hat{\alpha}_j}{\delta}\right) \right) \\
&\quad - \delta \cdot \log \left(\sum_j \exp\left(\frac{-\hat{\alpha}_j}{\delta}\right) (1+\exp(\hat{w}_j/\delta)) \right) \sum_j y_j^*(\hat{\alpha})
\end{aligned}
$$

To conclude, note that

$$
\begin{aligned}
\delta \sum_j y_j^*(\hat{\alpha}) \log y_j^*(\hat{\alpha}) &= \delta \sum_j y_j^*(\hat{\alpha}) \log \left(\frac{\exp\left(\frac{-\hat{\alpha}_j}{\delta}\right)}{\sum_{j'} \exp\left(\frac{-\hat{\alpha}_{j'}}{\delta}\right) \left(1+\exp(\hat{w}_{j'}/\delta)\right)} \right) \\
&= -\sum_j \hat{\alpha}_j \cdot y_j^*(\hat{\alpha}) - \delta \cdot \log \left(\sum_j \exp\left(\frac{-\hat{\alpha}_j}{\delta}\right) (1+\exp(\hat{w}_j/\delta)) \right) \sum_j y_j^*(\hat{\alpha})
\end{aligned}
$$

$\square$

Lemma C.5. $\sum_{j\in[S]} x_j^*(\hat{\alpha}) \cdot \hat{w}_j - \mathbb{E}_{\lambda,s',s'',t_{-k},\mathcal{A}',p'}[q_k] \geq 0$

Proof.

$$
\begin{aligned}
&\sum_{j\in[S]} x_j^*(\hat{\alpha}) \cdot \hat{w}_j - \mathop{\mathbb{E}}_{\lambda,s',s'',t_{-k},\mathcal{A}',p'}[q_k] \\
&\qquad = \sum_{j\in[S]} x_j^*(\hat{\alpha}) \cdot \hat{w}_j - \mathop{\mathbb{E}}_{s',t_{-k},\mathcal{A}',p'}[W_k(t_k, s', t_{-k})] + \mathop{\mathbb{E}}_{\lambda,s'',t_{-k},\mathcal{A}',p'}[W_k(t_k, s'', t_{-k})] + \delta \log 2 \\
&\qquad = \int_0^1 \sum_{j\in[S]} x_j^\lambda(\hat{\alpha}) \cdot \hat{w}_j \,\mathrm{d}\lambda + \delta \log 2 \\
&\qquad = \delta \cdot \left[\log \left(\sum_j \exp\left(\frac{-\hat{\alpha}_j}{\delta}\right) (1+\exp(\lambda\hat{w}_j/\delta)) \right) \right]_0^1 + \delta \log 2 \\
&\qquad = \delta \cdot \log \left(\sum_j \exp\left(\frac{-\hat{\alpha}_j}{\delta}\right) (1+\exp(\hat{w}_j/\delta)) \right) - \delta \cdot \log \left(\sum_j \exp\left(\frac{-\hat{\alpha}_j}{\delta}\right) \right) \\
&\qquad \geq 0
\end{aligned}
$$

□

Lemma C.6. $\mathbb{E}_{\lambda,s',s'',t_{-k},\mathcal{A}',p'}[q_k] \geq -\delta \log 2$

Proof. Since

$$\mathop{\mathbb{E}}_{\lambda,s',s'',t_{-k},\mathcal{A}',p'}[q_k] = \sum_{j\in[S]} x_j^*(\hat{\alpha}) \cdot \hat{w}_j - \int_0^1 \sum_{j\in[S]} x_j^\lambda(\hat{\alpha}) \cdot \hat{w}_j \, \mathrm{d}\lambda - \delta \log 2$$

it suffices to show that $\sum_{j\in[S]}(x_j^*(\hat{\alpha}) - x_j^\lambda(\hat{\alpha}))\cdot\hat{w}_j \geq 0$ for all $\lambda \in (0,1)$. Toward this end, note that by Fact 4.6,

$$\sum_{j\in[S]} x_j^*(\hat{\alpha})\hat{w}_j + \delta \sum_j (x_j^*(\hat{\alpha})\log x_j^*(\hat{\alpha}) + y_j^*(\hat{\alpha})\log y_j^*(\hat{\alpha})) - \sum_j \hat{\alpha}_j(x_j^*(\hat{\alpha}) + y_j^*(\hat{\alpha}))$$
$$\geq \sum_{j\in[S]} x_j^\lambda(\hat{\alpha})\hat{w}_j + \delta \sum_j (x_j^\lambda(\hat{\alpha})\log x_j^\lambda(\hat{\alpha}) + y_j^\lambda(\hat{\alpha})\log y_j^\lambda(\hat{\alpha})) - \sum_j \hat{\alpha}_j(x_j^\lambda(\hat{\alpha}) + y_j^\lambda(\hat{\alpha}))$$
$$\sum_{j\in[S]} x_j^\lambda(\hat{\alpha})\lambda\hat{w}_j + \delta \sum_j (x_j^\lambda(\hat{\alpha})\log x_j^\lambda(\hat{\alpha}) + y_j^\lambda(\hat{\alpha})\log y_j^\lambda(\hat{\alpha})) - \sum_j \hat{\alpha}_j(x_j^\lambda(\hat{\alpha}) + y_j^\lambda(\hat{\alpha}))$$
$$\geq \sum_{j\in[S]} x_j^*(\hat{\alpha})\lambda\hat{w}_j + \delta \sum_j (x_j^*(\hat{\alpha})\log x_j^*(\hat{\alpha}) + y_j^*(\hat{\alpha})\log y_j^*(\hat{\alpha})) - \sum_j \hat{\alpha}_j(x_j^*(\hat{\alpha}) + y_j^*(\hat{\alpha}))$$

Adding these two inequalities and simplifying yields

$$\sum_{j\in[S]} (x_j^*(\hat{\alpha}) - x_j^\lambda(\hat{\alpha})) \cdot \hat{w}_j \cdot (1-\lambda) \geq 0$$

which implies the desired inequality for all $\lambda \in (0,1)$. □

C.7 Helper Technical Lemmas

The following technical lemmas are used in the proofs of meta-input requirements.

Claim 6 (Estimated edge weights $\tilde{w}_{ij}$)**.** *Except with probability at most* $\exp\left(-\frac{nm}{\varepsilon}\right)$, $\forall r_i \in \mathcal{R}_k^{\mathrm{Tr}}, s_j \in \mathcal{S}_k$, $|\tilde{w}_{ij} - \hat{w}_{ij}| \leq \Delta_{\text{EDGE}}$, *where* $\tilde{w}_{ij}$ *is the output of* $\textsc{EstEdges}(\mathcal{R}_k^{\mathrm{Tr}}, (\mathcal{S}_{k'})_{k'\in[n]})$ *and* $\hat{w}_{ij}$ *is defined according to Theorem 4.1 with respect to distribution* $\hat{D}' := \times_{k'\in[n]}\hat{D}'_{k'}$, *where each* $\hat{D}'_{k'}$ *is the empirical distribution over* $\mathcal{S}_{k'}$.

Proof. Recall that $\Delta_{\text{EDGE}} = \frac{\varepsilon^3}{m^2 n}$ and $N_{\text{EDGE}} = \frac{m^6 n^3}{\varepsilon^9} = (\Delta_{\text{EDGE}})^{-3}$. Consider a particular r_i, s_j. Observe that $w_{ij}^{(p)}$, for $p \in [N_{\text{EDGE}}]$, in $\textsc{EstEdges}$ is are independent random variables each with mean $\mathbb{E}[w_{ij}^{(p)}] = \hat{w}_{ij}$ (over the randomness of the sample $t_{\text{-}k} \sim \hat{D}'_{\text{-}k}$, and of $\mathcal{A}'$ and p'). Furthermore, note that each $w_{ij}^{(p)} \in [-1,1]$ (by the bounded range of valuations and payments). It follows by a Hoeffding bound that

$$\Pr\left[|\tilde{w}_{ij} - \hat{w}_{ij}| \geq \Delta_{\text{EDGE}}\right] \leq 2\cdot\exp(-N_{\text{EDGE}} \cdot {\Delta_{\text{EDGE}}}^2) \leq \frac{1}{R}\cdot\frac{1}{S}\exp\left(-\frac{nm}{\varepsilon}\right)$$

by definition of $\tilde{w}_{ij} := \frac{1}{N_{\text{EDGE}}}\sum_{p=1}^{N_{\text{EDGE}}} w_{ij}^{(p)}$ and choice of N_{EDGE}. A union bound over all $r_i \in \mathcal{R}_k^{\mathrm{Tr}}$, $s_j \in \mathcal{S}_k$ gives the claim.

□

Lemma C.7 (Learning discrete distributions [Can20]). *Let F be a discrete distribution over some finite support of size N. Then for any $\varepsilon, \delta > 0$, the empirical distribution $\hat{F}$ defined by drawing $T \geq \frac{\log(2N)+\log(1/\delta)}{2\varepsilon^2}$ independent samples from F satisfies the following: with probability at least $1-\delta$, it holds that for all $y \in supp(F)$, $|p_{\hat{F}}(y) - p_F(y)| < \varepsilon$.*

Lemma C.8 (Babichenko et al. [BBP17], Devanur et al. [DHP16]). *Let $D = \times_i D_i$ be a product distribution. For each i, draw N samples from D_i and define $\bar{D}_i$ to be the uniform distribution over these samples. Let $\bar{D} = \times_i \bar{D}_i$ denote the product distribution whose marginals are given by $\bar{D}_i$. For all $\varepsilon > 0$ and $f : \mathrm{supp}(D) \to [0, H]$, we have that*

$$\Pr(|\mathbb{E}_{\bar{D}}[f] - \mathbb{E}_D[f]| > \varepsilon) \leq \tfrac{4H}{\varepsilon} \exp\left(-\tfrac{\varepsilon^2 N}{8H^2}\right)$$

D Extension: Unbounded Ex-Post Payments

Condition (A1) follows by setting $\varepsilon' := \frac{1}{2}\varepsilon_{\text{D-S}} = O(\varepsilon)$, $\Delta := \Delta_{\text{D-S}} = O(n\varepsilon)$, and $N := N_S = \frac{m^3 n}{\varepsilon^3}$ in Theorem D.1, for which the failure probability comes out to $O(\varepsilon) = \delta_{\text{D-S}}$.

Lemma D.1. *Consider mechanism $\mathcal{M}' = (\mathcal{A}', p')$ over $D' = \times_{k\in[n]} D'_k$ where each $D'_k = \times_{\ell\in[m]} D'_{k\ell}$. For each bidder k and item ℓ, draw N samples from $D'_{k\ell}$ and define $\hat{D}'_{k\ell}$ to be the uniform distribution over these samples. Let $\hat{D}'_k = \times_\ell \hat{D}'_{k\ell}$ and $\hat{D}' = \times_k \hat{D}'_k$. Let $\hat{M}' = (\mathcal{A}', \hat{p})$ and Flag_{Est} denote the output of $\text{EstMechanism}(\mathcal{M}', \hat{D}')$. With probability at least*

$$1 - \tfrac{8nN^{2m}}{\varepsilon'} \exp\left(-\tfrac{\varepsilon'^2 N}{16}\right) - 2nN^{2m} \exp(-2\varepsilon'^2 \cdot N) - \tfrac{4n}{\Delta} \exp\left(-\tfrac{\Delta^2 N}{8n^2}\right),$$

$\text{Flag}_{\text{Est}} =$ `True` and $\hat{\mathcal{M}}'$ is an $(\varepsilon + 6\varepsilon')$-BIC and $4\varepsilon'$-IR mechanism for bidders with types drawn from $\hat{D}'$ whose revenue from these bidders is at most Δ smaller than the revenue of $\mathcal{M}'$ from bidders whose types are drawn from D'.

Proof. Recall that valuations $v_k(\cdot,\cdot)$ are in $[0,1]$, so by Theorem C.8,

$$\Pr(|\mathbb{E}_{t_{\text{-}k}\sim\hat{D}'_{\text{-}k}}[v_k(t_k, \mathcal{A}'(t'_k, t_{-k}))] - \mathbb{E}_{t_{\text{-}k}\sim D'_{\text{-}k}}[v_k(t_k, \mathcal{A}'(t'_k, t_{-k}))]| > \varepsilon') \leq \tfrac{4}{\varepsilon'} \exp\left(-\tfrac{\varepsilon'^2 N}{8}\right)$$

for each bidder k and types $t_k, t'_k \in \mathrm{supp}(\hat{D}'_k)$. Moreover, by Hoeffding,

$$\Pr(|\hat{v}_k(t_k, t'_k) - \mathbb{E}_{t_{\text{-}k}\sim\hat{D}'_{\text{-}k}}[v_k(t_k, \mathcal{A}'(t'_k, t_{-k}))]| > \varepsilon') \leq 2\exp(-2\varepsilon'^2 \cdot N)$$

Via union bound, the probability that the above two events hold for each bidder k and each pair of types $t_k, t'_k \in \mathrm{supp}(\hat{D}'_k)$ simultaneously is at least

$$1 - \tfrac{8nN^{2m}}{\varepsilon'} \exp\left(-\tfrac{\varepsilon'^2 N}{16}\right) - 2nN^{2m} \exp(-2\varepsilon'^2 \cdot N).$$

For the remainder of the proof, we condition on these events.

Conditioned on these events, the interim payment of $\mathcal{M}'$

$$p'_k(t_k) = \mathbb{E}_{t_{-k}\sim D'_{-k}}[p'_k(t_k, t_{-k})]$$

is a feasible solution to the linear program in EstMechanism

$$\begin{aligned} \max_{\hat{p}'} \quad & \mathbb{E}_{t\sim\hat{D}'}[\textstyle\sum_k \hat{p}'_k(t_k)] \\ \text{s.t.} \quad & \hat{v}_k(t_k,t'_k) - \hat{p}'_k(t_k) \geq \hat{v}_k(t_k,t'_k) - \hat{p}'_k(t'_k) - (\varepsilon+4\varepsilon') \, \forall\, k\in[n], t_k, t'_k \in \operatorname{supp}(\hat{D}'_k) \\ & \hat{v}_k(t_k,t'_k) - \hat{p}'_k(t_k) \geq -2\varepsilon' \, \forall\, k\in[n], t_k \in \operatorname{supp}(\hat{D}'_k) \end{aligned}$$

since $\mathcal{M}'$ is ε-BIC and IR over D', so

$$\begin{aligned} &\mathbb{E}_{t_{\text{-}k}\sim D'_{\text{-}k}}[v_k(t_k,\mathcal{A}'(t)) - p'_k(t)] \geq \mathbb{E}_{t_{\text{-}k}\sim D'_{\text{-}k}}[v_k(t_k,\mathcal{A}'(t'_k,t_{\text{-}k})) - p'_k(t'_k,t_{\text{-}k})] - \varepsilon \\ &\mathbb{E}_{t_{\text{-}k}\sim D'_{\text{-}k}}[v_k(t_k,\mathcal{A}'(t)) - p'_k(t)] \geq 0 \end{aligned}$$

for all bidders k and types $t_k, t'_k \in \operatorname{supp}(D'_k)$. It follows that $\text{Flag}_{\text{Est}} = \texttt{True}$ and that $\hat{\mathcal{M}}' = (\mathcal{A}', \hat{p}')$, which charges $\hat{p}'_k(t_k)$ to bidder k whenever she reports t_k (regardless of what the other bidders report), is an $(\varepsilon+6\varepsilon')$-BIC nad $4\varepsilon'$-IR mechanism for $\hat{D}'$.

It remains to bound the revenue loss. Since the interim payment $p'_k(t_k) \in [0,1]$ (as $\mathcal{M}'$ is IR for D'), it follows from Theorem C.8 that

$$\Pr(|\mathbb{E}_{t\sim\hat{D}'}[\textstyle\sum_k p'_k(t_k)] - \mathbb{E}_{t\sim D'}[\sum_k p'_k(t_k)]| > \Delta) \leq \frac{4n}{\Delta}\exp\left(-\frac{\Delta^2 N}{8n^2}\right).$$

Moreover, $\hat{p}'$ is an optimal solution to the linear program, while the interim payment rule p' is a feasible solution, so

$$\mathbb{E}_{t\sim\hat{D}'}[\textstyle\sum_k \hat{p}'_k(t_k)] \geq \mathbb{E}_{t\sim\hat{D}'}[\sum_k p'_k(t_k)]$$

The result follows since $\text{Rev}(\hat{\mathcal{M}}', \hat{D}') = \mathbb{E}_{t\sim\hat{D}'}[\sum_k \hat{p}'_k(t_k)]$ and $\text{Rev}(\mathcal{M}'; D') = \mathbb{E}_{t\sim D'}[\sum_k p'_k(t_k)]$. □

E Extension: An Ex-Post IR Output

Condition (A1) follows by setting $\varepsilon' := \frac{1}{2}\varepsilon_{\text{D-S}} = O(\varepsilon)$, $\Delta := \Delta_{\text{D-S}} = O(n\varepsilon)$, $N := N_S = \frac{m^3 n}{\varepsilon^3}$, and $N_{\text{Query}} := (mn/\varepsilon)^3$ in Theorem E.1, for which the failure probability comes out to $O(\varepsilon) = \delta_{\text{D-S}}$.

Lemma E.1. *Consider mechanism $\mathcal{M}' = (\mathcal{A}', p')$ over $D' = \times_{k\in[n]} D'_k$ where each $D'_k = \times_{\ell\in[m]} D'_{k\ell}$. For each bidder k and item ℓ, draw N samples from $D'_{k\ell}$ and define $\hat{D}'_{k\ell}$ to be the uniform distribution over these samples. Let $\hat{D}'_k = \times_\ell \hat{D}'_{k\ell}$ and $\hat{D}' = \times_k \hat{D}'_k$. Let $\hat{M}' = (\hat{\mathcal{A}}', \hat{p})$ and* Flag_{Est} *denote the output of* EstMechanism$(\mathcal{M}', \hat{D}')$. *With probability at least*

$$1 - \frac{8nN^{2m}}{\varepsilon'}\exp\left(-\frac{\varepsilon'^2 N}{16}\right) - 2nN^{m\cdot(n+1)}\exp(-2\varepsilon'^2 \cdot N_{\text{Query}}) - \frac{4n}{\Delta}\exp\left(-\frac{\Delta^2 N}{8n^2}\right),$$

$\text{Flag}_{\text{Est}} = \texttt{True}$ *and $\hat{\mathcal{M}}'$ is an $(\varepsilon+4\varepsilon')$-BIC mechanism for bidders with types drawn from $\hat{D}'$ whose revenue from these bidders is at most Δ smaller than the revenue of $\mathcal{M}'$ from bidders whose types are drawn from D'.*

Proof. Recall that valuations $v_k(\cdot,\cdot)$ are in $[0,1]$, so by Theorem C.8,

$$\Pr(|\mathbb{E}_{t_{\text{-}k}\sim\hat{D}'_{\text{-}k}}[v_k(t_k,\mathcal{A}'(t'_k,t_{-k}))] - \mathbb{E}_{t_{\text{-}k}\sim D'_{\text{-}k}}[v_k(t_k,\mathcal{A}'(t'_k,t_{-k}))]| > \varepsilon') \leq \frac{4}{\varepsilon'}\exp\left(-\frac{\varepsilon'^2 N}{8}\right)$$

for each bidder k and types $t_k, t'_k \in \text{supp}(\hat{D}'_k)$. Moreover, by Hoeffding,

$$\Pr(|\mathbb{E}[v_k(t_k, \hat{\mathcal{A}}'(t'_k, t_{-k}))] - \mathbb{E}[v_k(t_k, \mathcal{A}'(t'_k, t_{-k}))]| > \varepsilon') \leq 2\exp(-2\varepsilon'^2 \cdot N_{\textsc{Query}})$$

for each bidder k, $t_k, t'_k \in \mathcal{S}_k$, and $t_{-k} \in \mathcal{S}_{-k}$. Via union bound, the probability that

$$|\mathbb{E}_{t_k \sim \hat{D}'_{-k}}[v_k(t_k, \hat{\mathcal{A}}'(t'_k, t_{-k}))] - \mathbb{E}_{t_{-k} \sim D'_{-k}}[v_k(t_k, \mathcal{A}'(t'_k, t_{-k}))]| \leq 2\varepsilon'$$

for all bidders k and types $t_k, t'_k \in \mathcal{S}_k$ simultaneously is at least

$$1 - \tfrac{8nN^{2m}}{\varepsilon'} \exp\left(-\tfrac{\varepsilon'^2 N}{16}\right) - 2nN^{m\cdot(n+1)} \exp(-2\varepsilon'^2 \cdot N_{\textsc{Query}}).$$

For the remainder of the proof, we condition on this event.

Conditioned on this event, the interim payment of $\mathcal{M}'$

$$p'_k(t_k) = \mathbb{E}_{t_{-k} \sim D'_{-k}}[p'_k(t_k, t_{-k})]$$

is a feasible solution to the linear program in EstMechanism

$$\begin{aligned}
\max_{\hat{p}'} \quad & \mathbb{E}_{t \sim \hat{D}'}[\textstyle\sum_k \hat{p}'_k(t_k)] \\
\text{s.t.} \quad & \mathbb{E}_{t_k \sim \hat{D}'_{-k}}[v_k(t_k, \hat{\mathcal{A}}'(t))] - \hat{p}'_k(t_k) \geq \mathbb{E}_{t_k \sim \hat{D}'_{-k}}[v_k(t_k, \hat{\mathcal{A}}'(t'_k, t_{-k}))] - \hat{p}'_k(t'_k) \\
& \qquad\qquad\qquad\qquad\qquad\qquad - (\varepsilon + 4\varepsilon') \,\forall\, k \in [n], t_k, t'_k \in \text{supp}(\hat{D}'_k) \\
& \mathbb{E}_{t_k \sim \hat{D}'_{-k}}[v_k(t_k, \hat{\mathcal{A}}'(t))] - \hat{p}'_k(t_k) \geq -2\varepsilon' \,\forall\, k \in [n], t_k \in \text{supp}(\hat{D}'_k)
\end{aligned}$$

since $\mathcal{M}'$ is ε-BIC and IR over D', so

$$\begin{aligned}
&\mathbb{E}_{t_{\text{-}k} \sim D'_{\text{-}k}}[v_k(t_k, \mathcal{A}'(t)) - p'_k(t)] \geq \mathbb{E}_{t_{\text{-}k} \sim D'_{\text{-}k}}[v_k(t_k, \mathcal{A}'(t'_k, t_{\text{-}k})) - p'_k(t'_k, t_{\text{-}k})] - \varepsilon \\
&\mathbb{E}_{t_{\text{-}k} \sim D'_{\text{-}k}}[v_k(t_k, \mathcal{A}'(t)) - p'_k(t)] \geq 0
\end{aligned}$$

for all bidders k and types $t_k, t'_k \in \text{supp}(D'_k)$. It follows that $\textsc{Flag}_{\textsc{Est}} = $ `True` and that $\hat{\mathcal{M}}' = (\hat{\mathcal{A}}', \hat{p}')$, which charges $\hat{p}'_k(t_k)$ to bidder k whenever she reports t_k (regardless of what the other bidders report), is an $(\varepsilon + 4\varepsilon')$-BIC and $2\varepsilon'$-IR mechanism for $\hat{D}'$.

It remains to bound the revenue loss. Since the interim payment $p'_k(t_k) \in [0, 1]$ (as $\mathcal{M}'$ is IR for D'), it follows from Theorem C.8 that

$$\Pr(|\mathbb{E}_{t \sim \hat{D}'}[\textstyle\sum_k p'_k(t_k)] - \mathbb{E}_{t \sim D'}[\sum_k p'_k(t_k)]| > \Delta) \leq \tfrac{4n}{\Delta} \exp\left(-\tfrac{\Delta^2 N}{8n^2}\right).$$

Moreover, $\hat{p}'$ is an optimal solution to the linear program, while the interim payment rule p' is a feasible solution, so

$$\mathbb{E}_{t \sim \hat{D}'}[\textstyle\sum_k \hat{p}'_k(t_k)] \geq \mathbb{E}_{t \sim \hat{D}'}[\sum_k p'_k(t_k)]$$

The result follows since $\textsc{Rev}(\hat{\mathcal{M}}', \hat{D}') = \mathbb{E}_{t \sim \hat{D}'}[\sum_k \hat{p}'_k(t_k)]$ and $\textsc{Rev}(\mathcal{M}'; D') = \mathbb{E}_{t \sim D'}[\sum_k p'_k(t_k)]$. □