Single-Sample Prophet Inequalities: A Combinatorial to Single-Item Reduction

Shuchi Chawla
The University of Texas at Austin
shuchi@cs.utexas.edu

Trung Dang
The University of Texas at Austin
dddtrung@cs.utexas.edu

Abstract

We study single-sample prophet inequalities for online combinatorial allocation. Our main contribution is a general reduction from combinatorial to single-item prophet inequalities for valuation classes admitting suitable supporting prices. The reduction uses a free-disposal value to separate buyer-side combinatorial constraints from item-side supply constraints, yielding a modular framework that applies in the stronger Game of Googol model. This framework yields a $\frac{1}{6\sqrt{3}} \approx \frac{1}{10.4}$-competitive single-sample prophet inequality and a $(\beta_{k-1}/4)$-competitive k-sample prophet inequality for XOS valuations and $k \geq 2$, where β_k is the competitive ratio of a k-sample single-item prophet inequality, improving upon the work of [DKL+24]. Both results extend directly to divisible resources with continuous capped-XOS valuations, as well as to multi-unit resources and valuations. Along the way, we obtain new results for online free disposal and an optimal single-sample prophet inequality for fractional knapsack in the Game of Googol model.

Contents

1 Introduction **1**
- 1.1 Technical Overview and Results . . . 2
- Acknowledgments and AI Disclosure . . . 5

2 Preliminaries **5**
- 2.1 Online Combinatorial Allocation, Prophet Inequalities, and the Game of Googol . . . 5
- 2.2 Assumptions on Valuation Functions . . . 7
- 2.3 Free-Disposal Value . . . 9
- 2.4 Single-Item Prophet Inequality . . . 10

3 A General Reduction from Combinatorial to Single-Item PI **10**
- 3.1 Reducing $(k+1)$-SCombPI to k-SPI . . . 11
- 3.2 Constructing a Competitive Buyer-Local Provisional-Allocation Algorithm . . . 13
- 3.3 From Two Samples to One Sample . . . 15
- 3.4 Computational Considerations . . . 17

4 Random-Order Free-Disposal Algorithms **17**
- 4.1 Indivisible-Item Free-Disposal Algorithm . . . 17
- 4.2 Divisible-Item Free-Disposal Algorithms . . . 20
  - 4.2.1 A Continuous Extension of Buyer-Wise Greedy . . . 21
  - 4.2.2 A $(1-1/e)$-Competitive Algorithm . . . 24

5 Single-Sample Single-Item Prophet Inequalities **27**
- 5.1 Single-Sample Prophet Inequality for an Indivisible Item . . . 27
- 5.2 Single-Sample Prophet Inequality for Fractional Knapsack . . . 27
- 5.3 Single-Sample Prophet Inequality for a Concave Divisible Item . . . 31

6 Applications **32**

7 Related Work **33**

A Extension to Subadditive Valuations **36**

B Deferred Proofs from Section 2 **37**

C Deferred Proofs from Section 3 **37**

D Deferred Proofs from Section 4 **39**

E Deferred Proofs from Section 5 **42**

F Deferred Proofs from Section 6 **43**

1 Introduction

Can samples be as informationally powerful as the entire distribution? Single-sample prophet inequalities push this question to its informational extreme. An online algorithm faces an input drawn from an unknown distribution and must make irrevocable decisions as the instance is revealed. To guide these decisions, it receives just one independent sample from the underlying distribution. The central question is whether this tiny amount of information can nevertheless support guarantees comparable to those achievable when the full distribution is known.

In a striking result, Rubinstein, Wang, and Weinberg [RWW20] showed that, for the classical single-item prophet inequality, the answer is yes. In this problem, the algorithm observes n nonnegative values drawn independently from possibly different distributions and presented in an adversarial order. It must immediately accept or reject each value, accepting at most one, with the goal of maximizing the expected value selected. Even with full knowledge of the distributions, no online algorithm can guarantee more than half the expected hindsight optimum in the worst case. Rubinstein et al. showed that this optimal factor-2 guarantee can be matched using just one sample from each distribution. The algorithm is remarkably simple: set a threshold equal to the largest sampled value, and accept the first real value that exceeds it.

Subsequent work has demonstrated the power of a single sample in a variety of online settings, including multi-unit selection, matching, and selection under several classes of matroid constraints [CDF+22, KNR22, PS25, NW26]. Many of these results concern *single-dimensional* selection, where each arriving candidate has a scalar value and the algorithm decides whether to accept or reject it subject to feasibility constraints. In *multidimensional* allocation, the algorithm must additionally decide how to serve each candidate, for example, which subset of items to allocate to a buyer whose value depends on the entire bundle. Single-sample guarantees are known for a few such settings, including unit-demand and budget-additive valuations [CDF+22], and Dütting et al. [DKL+24] recently extended this line of work to general XOS (fractionally subadditive) valuations. These results, however, leave open whether single-sample guarantees for combinatorial allocation can be derived systematically from their single-item counterparts, separating the challenges of combinatorial allocation from those of limited distributional information.

In this paper, we develop **a general reduction from combinatorial to single-item single-sample prophet inequalities** for valuations admitting suitable supporting prices. We formulate a free-disposal objective and introduce buyer-locality to separate buyer-side combinatorial structure from item-side supply constraints while preserving the independence needed by single-item prophet inequalities. This decomposition allows online free-disposal algorithms and single-item prophet inequalities to serve as independently replaceable components. Moreover, buyer-locality lets us exploit random-order guarantees for the free-disposal component even when real buyers arrive adversarially. The resulting framework yields substantially improved guarantees for XOS valuations and extends to divisible resources.

To understand our reduction, consider combinatorial allocation with unit-demand buyers. There are m items and n buyers arriving online. Each buyer has a value for every item, with her m-dimensional value vector drawn from an arbitrary joint distribution. A tempting approach is to treat the items independently, running m single-item single-sample prophet inequalities, one for each item. But treating items independently ignores the fact that a unit-demand buyer derives value from at most one item. A single buyer may simultaneously have high values for many items and hence be competitive in many of these prophet inequalities. If she participates in all of them, she may block other buyers from receiving several items, even though, being unit-demand, she ultimately derives value from only one. To avoid this loss, we must first decide which competition—that is, which item-specific prophet inequality—each buyer will participate in.

Our reduction formalizes precisely this separation. In the first stage, we choose a buyer-feasible *provisional allocation*: for a unit-demand buyer, this amounts to selecting a single candidate item or a single competition to participate in. At this stage we ignore supply constraints, so many buyers may provisionally select the same item. The right objective for making these choices is a *free-disposal* objective: for each item, we count only the most valuable provisional claim, since all the others may eventually be discarded. The second stage then runs a single-item prophet inequality independently for each item to resolve these competing claims. More generally, supporting prices extend this same decomposition beyond unit-demand valuations: the free-disposal algorithm handles the multidimensional, buyer-side constraints, while the single-item prophet inequalities enforce the item-side supply constraints. This decomposition introduces a further challenge: choosing a buyer's provisional allocation based on earlier buyers' realized valuations can create dependencies that invalidate the single-item prophet guarantees. We address this by using a coordination sample to determine buyers' provisional allocations without using other buyers' realized valuations, a property we call buyer locality.

Our framework gives a substantially sharper understanding of the single-sample prophet inequality for XOS valuations of Dütting et al. [DKL+24], who first obtained a constant-factor competitive ratio of 1/576 for this problem. Viewed through the free-disposal framework, essentially the same algorithm admits a simpler, modular analysis, improving the competitive ratio to 1/24. More importantly, this viewpoint exposes further opportunities for improvement, both in the underlying free-disposal algorithm and in the analysis of the reduction. Exploiting these opportunities improves the competitive ratio further to $1/(6\sqrt{3}) \approx 1/10.4$. Beyond this improvement, the framework enables extensions to broader multidimensional settings: the same guarantee holds for continuous capped-XOS valuations over divisible resources, with fractional knapsack replacing single-item selection as the underlying prophet problem. All our guarantees hold in the stronger Game of Googol model. These results illustrate how the reduction accommodates different resource models by cleanly separating the combinatorial allocation problem from the single-resource competition.

Table 1 summarizes the competitive ratios we achieve using a single sample in each setting.

1.1 Technical Overview and Results

Linearization of values. At a high level, our reduction from combinatorial to single-dimensional settings maps each buyer's combinatorial valuation function v_i to a linear valuation represented by the vector $\boldsymbol{q}_i$. The j-th coordinate $q_{i,j}$ specifies the value with which buyer i enters the competition for item j. Each item is ultimately allocated to at most one of the buyers competing for it, and the resulting linear objective provides a lower bound on the buyers' true welfare once the item-wise allocations are combined. Thus, the linearization aims to serve two purposes: it determines which item-wise competitions each buyer participates in, and it decouples the objectives for these competitions while preserving a lower bound on the welfare of the resulting allocation.

We instantiate the linearization using supporting prices. For example, for an XOS valuation v, every allocation $\boldsymbol{y}$ admits a supporting-price vector $\boldsymbol{q}_v(\boldsymbol{y})$ such that

$$v(\boldsymbol{y}) = \langle \boldsymbol{q}_v(\boldsymbol{y}), \boldsymbol{y} \rangle, \qquad \text{and} \qquad v(\boldsymbol{x}) \geq \langle \boldsymbol{q}_v(\boldsymbol{y}), \boldsymbol{x} \rangle$$

for every $\boldsymbol{x} \leq \boldsymbol{y}$. The allocation $\boldsymbol{y}_i$ at which $\boldsymbol{q}_i$ supports buyer i's valuation serves as the anchor for this linearization: the final allocation to buyer i is obtained by trimming $\boldsymbol{y}_i$. We refer to $\boldsymbol{y}_i$ as the buyer's *provisional allocation*.

The free-disposal value. We cast the choice of linearizations as an online optimization problem. As each buyer arrives, we choose a provisional allocation $\boldsymbol{y}_i$ and the corresponding supporting-price

Setting	Previous single-sample guarantee	This work (Game of Googol model)
Indivisible items: XOS	1/576 [DKL+24]	$1/(6\sqrt{3}) \approx 1/10.39 \approx 0.0962$ (Theorem 6.1)
Indivisible items: monotone subadditive	$1/(576H_m)$, implied by [DKL+24, Dob07, BR11]	$1/(6\sqrt{3}H_m)$ (Theorems 6.1 and A.2)
Divisible items: continuous capped-XOS	N/A	$1/(6\sqrt{3}) \approx 0.0962$ (Theorem 6.2)
Multi-unit items: XOS, at most one copy per buyer per type	1/576 via separate copies [DKL+24]	$1/(6\sqrt{3}) \approx 0.0962$ (Theorem 6.3)
One divisible item: fractional knapsack	1/2 for independent draws [NRS26]	1/2, optimal (Theorem 5.2)
One divisible item: monotone concave valuations	N/A	$(1 - 1/e)/2 - \varepsilon$ (Theorem 5.8)

Table 1: Single-sample guarantees for adversarial arrivals. Here $H_m = \sum_{j=1}^{m} 1/j$ denotes the m-th harmonic number and $\varepsilon > 0$ can be arbitrarily small. All competitive ratios are stated as < 1, so larger is better. All our guarantees hold in the Game of Googol model. The indivisible and multi-unit XOS algorithms use polynomially many demand and supporting-price queries. For continuous capped-XOS valuations over divisible items, the free-disposal optimization has an efficient implementation under the additional concavity and oracle assumptions specified in Section 4. The subadditive guarantee follows from the standard H_m-approximation by XOS valuations [Dob07, BR11].

vector $\boldsymbol{q}_i$ at $\boldsymbol{y}_i$. The provisional allocations satisfy the buyer-side constraints, but may conflict with one another on the supply side. The goal is to ensure that, after resolving these conflicts, the supporting-price value that can be retained is comparable to the hindsight-optimal social welfare. We call this retained value the *free-disposal value*.[1]

It remains to understand whether the free-disposal problem itself admits good online algorithms. We show that, for the valuation classes we consider, its objective can be represented as an online monotone submodular maximization problem subject to a simple partition matroid constraint. In particular, the familiar buyer-wise greedy algorithm for XOS valuations [DNS05] is exactly the greedy algorithm for this formulation, giving a constant-factor guarantee for the free-disposal value. This provides the first component of our reduction; the second component consists of the item-wise single-dimensional prophet inequalities.

The role of samples. In our reduction, samples serve two distinct purposes: coordinating provisional allocations and supplying the inputs used by the item-wise prophet inequalities. Although the underlying worst-case free-disposal algorithm needs no distributional information, its choices

[1]We use the term *free disposal* in the same spirit as Feldman et al. [FKM+09] and Charikar et al. [CHN14]. In [FKM+09], in particular, items arrive online, so item-side supply constraints are automatically respected, while buyer-side demand constraints may be violated and excess allocation is discarded. Our setting reverses the roles of the two sides: buyers arrive online, so we enforce buyer-side constraints when choosing provisional allocations, while allowing item-side supply constraints to be violated; free disposal then trims the provisional allocations back to supply feasibility.

generally depend on earlier buyers' realized valuations. Feeding these choices directly into the item-wise prophet inequalities would introduce dependencies across buyers and invalidate their guarantees. We avoid this difficulty using an additional *coordination sample*. We run the free-disposal algorithm on this sample and record its state immediately before processing each buyer. When real buyer i arrives, we return the provisional allocation and supporting prices that the algorithm would produce if i's real valuation replaced her sampled valuation at that recorded state. Crucially, this counterfactual execution does not use any other buyer's real valuation. Conditional on the coordination sample and shared internal randomness, each buyer's output depends only on her own valuation. We call this property *buyer-locality*; it preserves independence across buyers, while allowing dependencies across items belonging to the same buyer.

Combining these components, an α-competitive free-disposal algorithm and a β_k-competitive k-sample single-item prophet inequality yield an $\alpha\beta_k/2$-competitive $(k+1)$-sample combinatorial prophet inequality for any $k \geq 1$ (Theorem 3.1).

Random-order free disposal. Buyer-locality also separates the real arrival order of buyers from the order over sampled values used by the free-disposal simulation. This allows us to process the coordination sample in a uniformly random order, even when the real buyers arrive adversarially. Doing so yields a further improvement in the competitive ratio.

Obtaining a truly single-sample PI. Finally, we adapt the sample-splitting idea of Dütting et al. [DKL+24] to our reduction: a single sample is randomly divided between the coordination and competition roles, allowing us to construct a single-sample combinatorial prophet inequality from a single-sample single-item one. Our contribution is a sharper analysis through a notion of smooth competitiveness, which measures how much free-disposal value an algorithm retains when only a random subset of buyers participates. Tracking this value as a function of the sampling probability yields a stronger guarantee for the reduction, even with a balanced split. Optimizing the split probability gives a further improvement, leading to the guarantees reported above.

Game of Googol setting. All of our prophet guarantees hold in the stronger Game of Googol (GoG) model [CCES20]. In this setting, there is no underlying distribution over instances. Instead, for each buyer, an adversary chooses $k+1$ valuation functions, which are then randomly assigned to k sample positions and one real position. The samples are revealed to the algorithm before the process begins, while the real valuations arrive online in adversarial order. Since the GoG model is stronger than the standard sample-based setting, every guarantee we establish in the GoG model immediately applies to the latter as well.

Divisible items and multi-unit items. The key technical ingredient underlying our results is the existence of one-sided supporting prices. For indivisible-item settings, this condition characterizes XOS valuations. For divisible items, it captures the strictly larger class of *capped-XOS* valuations. A capped-XOS valuation is a maximum of capped-additive clauses of the form

$$h_{a,c}(x) = \sum_j a_j \min\{x_j, c_j\}, \qquad a_j \geq 0, \quad c_j \in [0,1].$$

In the divisible-item setting, for each item j, buyer i's provisional allocation contributes an amount $y_{i,j}$ at density $q_{i,j}$. Our continuous free-disposal algorithm for continuous valuations in this class has the same smooth guarantee as its indivisible counterpart. Resolving supply competition, on the other hand, is equivalent to solving a fractional knapsack with capacity one. We show that this

problem admits the same optimal 1/2 single-sample prophet inequality in the GoG setting as the classical single-choice counterpart. Consequently, we obtain the same competitive ratio in both settings. Finally, our approach for divisible items naturally extends to the ℓ-unit setting as well via an appropriate reduction to the $1/\ell$ grid.

Other notable results. Our results require several other new ingredients that are of independent interest. First, we prove an optimal 1/2 single-sample prophet inequality for fractional knapsack in the Game of Googol model. The algorithm is the random-quantile threshold rule of Nuti et al. [NRS26]. Our contribution is an analysis that also handles the sample-real dependence in the Game of Googol model. Second, beyond the greedy algorithms described above, we give a $(1 - 1/e)$-competitive free-disposal algorithm for concave DR-submodular valuations over divisible items, even under adversarial arrivals. Finally, for a single divisible item with monotone concave valuations, we give a separate algorithm based on discretizing marginal values with a competitive ratio approaching 0.316.

Organization of the paper

Section 2 introduces the model, the classes of valuations we study, and the free-disposal value. We present the main reduction from $(k+1)$-sample combinatorial to k-sample single-item prophet inequalities in Section 3, along with the two-sample-to-one-sample improvement to obtain a single-sample combinatorial prophet inequality. Section 4 develops the required free-disposal algorithms for indivisible and divisible resources. Section 5 studies the corresponding single-item single-sample prophet inequalities, including fractional knapsack and concave divisible valuations. Finally, Section 6 combines these ingredients to obtain our main results; additional proofs and extensions appear in the appendices. Related work is discussed in Section 7.

Acknowledgements and AI Disclosure

We are grateful to Sahil Singla for comments and suggestions on an earlier draft of this work. The authors used ChatGPT 5.6 Pro to assist with writing and completing the proofs. The authors have validated all details and take full responsibility for the content of this paper.

2 Preliminaries

2.1 Online Combinatorial Allocation, Prophet Inequalities, and the Game of Googol

In the *Combinatorial Allocation* problem, there are n buyers and m items. We use $A \subseteq [0,1]$ to denote the feasible allocation set for a single item, so $A^m \subseteq [0,1]^m$ is the space of allocation vectors for an individual buyer. Each buyer $i \in [n]$ has a valuation function $v_i : A^m \to \mathbb{R}_{\geq 0}$ that maps a feasible allocation vector to a value. The function v_i is drawn independently from an underlying distribution $\mathcal{D}_i$. Let $\mathcal{D} = \times_i \mathcal{D}_i$. Our goal is to find a partition of the m items into allocations for each buyer that maximizes the social welfare, that is, the sum of the values received by each buyer.

The optimal social welfare (a.k.a. hindsight optimum) is given by[2]

$$\mathsf{Opt}(\mathcal{D}) := \mathbb{E}_{\boldsymbol{v}\sim\mathcal{D}}[\mathsf{Opt}(\boldsymbol{v})] := \mathbb{E}\left[\max_{\substack{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\in A^m \\ \sum_{i=1}^n \boldsymbol{x}_i\le \mathbf{1}}} \sum_{i=1}^n v_i(\boldsymbol{x}_i)\right].$$

We call the constraint $\boldsymbol{x}_i \in A^m$ the demand constraint as it is a buyer constraint, and $\sum_{i=1}^n \boldsymbol{x}_i \le \mathbf{1}$ the supply constraint as it is an item constraint.

Online allocation and competitive ratio. In the online setting, buyers arrive sequentially. Upon the arrival of buyer i, the valuation v_i is revealed to the algorithm, and the algorithm must irrevocably allocate a vector $\boldsymbol{x}_i \in A^m$ to the buyer. For feasibility, the algorithm must satisfy $\sum_i \boldsymbol{x}_i \le \mathbf{1}$. We write $\mathsf{Alg}(\boldsymbol{v}) := \sum_{i=1}^n v_i(\boldsymbol{x}_i)$ for the reward of the algorithm on instance $\boldsymbol{v}$. We say that an algorithm is α-*competitive* for some $\alpha \in [0,1]$ if, for every product distribution $\mathcal{D}$, we have

$$\mathsf{Alg}(\mathcal{D}) := \mathbb{E}_{\boldsymbol{v}\sim\mathcal{D}}[\mathsf{Alg}(\boldsymbol{v})] \ge \alpha \cdot \mathsf{Opt}(\mathcal{D}).$$

The optimal algorithm is also known as the *prophet*, and in the setting where the distribution $\mathcal{D}$ is known to the online algorithm, an α-competitive algorithm is also called a *Combinatorial Prophet Inequality* (henceforth, CombPI).

Sample-based prophet inequalities. In the sample-based setting, the distribution $\mathcal{D}$ is unknown to the algorithm, but the algorithm is supplied with k samples drawn i.i.d. from $\mathcal{D}$ at the beginning of the online process. The algorithm may use these samples to coordinate online allocations across buyers. Note that each sample is a full instance composed of valuation functions for each buyer. We use $\boldsymbol{s}^{(j)} = (s_1^{(j)}, \ldots, s_n^{(j)}) \sim \mathcal{D}$ for $j \in [k]$ to denote the j-th sample (with $s_i^{(j)} \sim \mathcal{D}_i$ denoting buyer i's valuation in sample j), and $\boldsymbol{r} \sim \mathcal{D}$ to denote the real instance over which the algorithm is run and evaluated.

Game of Googol. Online allocation algorithms in the sample-based setting often exploit the symmetry between the samples $\boldsymbol{s}^{(j)}$ and the real instance $\boldsymbol{r}$, as both are drawn independently from the same underlying distribution $\mathcal{D}$. The Game of Googol (GoG) setting makes this symmetry explicit without requiring the existence of an underlying distribution. In this setting, an adversary picks $k+1$ instances $\boldsymbol{v}^{(1)}, \cdots, \boldsymbol{v}^{(k+1)}$ where $\boldsymbol{v}^{(j)} = (v_i^{(j)})_{i\in[n]}$ for $j \in [k+1]$. For every buyer $i \in [n]$, the $k+1$ valuation functions $v_i^{(1)}, \cdots, v_i^{(k+1)}$ are then independently and uniformly permuted and assigned to each of k sample valuations $s_i^{(1)}, \cdots, s_i^{(k)}$ and one real valuation r_i. The algorithm is presented with the k samples $\boldsymbol{s}^{(1)}, \cdots, \boldsymbol{s}^{(k)}$ where $\boldsymbol{s}^{(j)} = (s_i^{(j)})_{i\in[n]}$. It is then run over the instance $\boldsymbol{r} = (r_i)_{i\in[n]}$ in an online fashion. The algorithm's performance is compared against the hindsight-optimal solution over $\boldsymbol{r}$.

We say that an algorithm achieves an α-competitive ratio in the Game of Googol setting if

$$\mathbb{E}_{\boldsymbol{s},\boldsymbol{r}}[\mathsf{Alg}(\boldsymbol{s},\boldsymbol{r})] = \mathbb{E}\left[\sum_{i=1}^n r_i(\boldsymbol{x}_i)\right] \ge \alpha \cdot \mathbb{E}[\mathsf{Opt}(\boldsymbol{r})],$$

where the expectations are taken over the algorithm's internal randomness and the random assignment of the $k+1$ valuation functions $v_i^{(1)}, \cdots, v_i^{(k+1)}$ to the sample and real valuations for each $i \in [n]$.

[2]We use $\mathbf{1}$ to denote the all-ones vector 1^m.

Note that the GoG setting is stronger than the sample-based setting in that an algorithm that is α-competitive in the former setting is also α-competitive in the latter. This work focuses primarily on the GoG setting.

We use the term *k-Sample Combinatorial Prophet Inequality* (henceforth, k-SCombPI) to denote a competitive online algorithm in the GoG setting. The special case of $k = 1$ is called a *Single-Sample Combinatorial Prophet Inequality* (henceforth, SSCombPI).

2.2 Assumptions on Valuation Functions

In this work, we consider two main settings for the possible allocations to each buyer:

1. **Indivisible items:** Each buyer can receive a discrete subset of the items, i.e., $A = \{0, 1\}$.
2. **Divisible items:** Each buyer can receive a fractional allocation of the items, i.e., $A = [0, 1]$.

Let v be any valuation function considered. Throughout this paper, we assume that v is *normalized* and *monotone*, i.e., $v(\mathbf{0}) = 0$ and $v(\boldsymbol{x}) \leq v(\boldsymbol{x}')$ if $\boldsymbol{x} \leq \boldsymbol{x}'$.[3] The normalization condition is without loss of generality, as an algorithm that is α-competitive for the normalized valuation $v'(\boldsymbol{x}) = v(\boldsymbol{x}) - v(\mathbf{0})$ is also α-competitive for v. We also make the following crucial assumption on v, which underlies all of our techniques.

Definition 1 (Supporting prices)**.** A valuation function $v : A^m \to \mathbb{R}_{\geq 0}$ (for $A = \{0, 1\}$ or $[0, 1]$) is said to have *supporting prices* if, for any allocation $\boldsymbol{x} \in A^m$, there exists a vector $\boldsymbol{q}_v(\boldsymbol{x}) \in \mathbb{R}^m_{\geq 0}$ such that

- $\langle \boldsymbol{q}_v(\boldsymbol{x}), \boldsymbol{x} \rangle = v(\boldsymbol{x})$.
- For all $\boldsymbol{x}' \in A^m$ with $\boldsymbol{x}' \leq \boldsymbol{x}$, $\langle \boldsymbol{q}_v(\boldsymbol{x}), \boldsymbol{x}' \rangle \leq v(\boldsymbol{x}')$.

When v_i or v is clear from context, we will simply use $\boldsymbol{q}_i(\boldsymbol{x})$ or $\boldsymbol{q}(\boldsymbol{x})$.

Observe that this definition is different from a standard definition of supporting prices, which requires the inequality $\langle \boldsymbol{q}_v(\boldsymbol{x}), \boldsymbol{x}' \rangle \leq v(\boldsymbol{x}')$ to hold for *all* $\boldsymbol{x}'$ and not just those component-wise smaller than $\boldsymbol{x}$. For indivisible items, the two definitions are equivalent, but for divisible items, the one-sided definition used here is weaker (more permissive) as discussed below.

Whenever computational efficiency is discussed, we assume access to a supporting-price oracle that, given v and $\boldsymbol{x}$, returns a valid supporting-price vector $\boldsymbol{q}_v(\boldsymbol{x})$ in polynomial time.

Indivisible items. In the indivisible-item setting, the supporting-price condition above is equivalent to the statement that v is an *XOS* valuation function.

Definition 2 (XOS valuation function [Fei09])**.** A valuation function $v : 2^{[m]} \to \mathbb{R}_{\geq 0}$ is *XOS* if there exist vectors $\boldsymbol{a}_k = (a_{j,k})_{j \in [m]} \in \mathbb{R}^m_{\geq 0}$ for $k \in [K]$ such that, for all $S \subseteq [m]$,

$$v(S) = \max_{k \in [K]} \sum_{j \in S} a_{j,k}$$[4].

We also consider two special subclasses of XOS valuations, namely *submodular* and *unit-demand* valuations.

[3] For vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$, we use $\boldsymbol{x} \leq \boldsymbol{x}'$ to denote the component-wise inequality $x_j \leq x'_j$ for all $j \in [m]$.

[4] Throughout the paper, we take the maximum of an empty family of nonnegative values to be 0.

Definition 3 (Submodular valuation function)**.** A valuation function $v : 2^{[m]} \to \mathbb{R}_{\geq 0}$ is *submodular* if for any $S \subseteq T \subseteq [m]$ and $j \in [m]$,

$$v(S \cup \{j\}) - v(S) \geq v(T \cup \{j\}) - v(T).$$

Definition 4 (Unit-demand valuation function)**.** A valuation function $v : 2^{[m]} \to \mathbb{R}_{\geq 0}$ is *unit-demand* if there exists a vector $\boldsymbol{v} \in \mathbb{R}^m_{\geq 0}$ such that, for any $S \subseteq [m]$,

$$v(S) = \max_{j \in S} v_j.$$

Divisible items. In the divisible-item setting, the supporting-price assumption is equivalent to the following notion of a *capped-XOS valuation function*. We prove this equivalence in Appendix B.

Definition 5 (Capped-XOS valuation function)**.** A *capped-additive clause* is a function of the form

$$h_{\boldsymbol{a},\boldsymbol{c}}(\boldsymbol{x}) = \sum_{i=1}^{m} a_i \min\{x_i, c_i\},$$

where $\boldsymbol{a} \in \mathbb{R}^m_{\geq 0}$ and $\boldsymbol{c} \in [0,1]^m$. A valuation v is *capped-XOS* if there is a family $\{h_\ell\}_{\ell \in L}$ of capped-additive clauses such that for all $\boldsymbol{x} \in [0,1]^m$,

$$v(\boldsymbol{x}) = \sup_{\ell \in L} h_\ell(\boldsymbol{x}).$$

In the case that L is infinite, we require that the supremum is attained for all $\boldsymbol{x} \in [0,1]^m$.

Proposition 2.1. *A normalized, monotone valuation $v : [0,1]^m \to \mathbb{R}_{\geq 0}$ admits a supporting-price vector $\boldsymbol{q}(\boldsymbol{x})$ at every $\boldsymbol{x} \in [0,1]^m$ if and only if v is capped-XOS.*

We also consider a specific subclass of capped-XOS valuations, called *DR-submodular* valuations. This subclass has been studied in previous work on prophet inequalities for divisible items [OSS26]. In addition to having supporting prices, such valuations have the desirable property that their supporting-price map is continuous.

Definition 6 (DR-submodular valuation function [BLKB17, BBK20])**.** A twice-differentiable valuation function $v : \mathbb{R}^m_{\geq 0} \to \mathbb{R}_{\geq 0}$ is *DR-submodular* if it satisfies one of the following three equivalent conditions:

1. $v(\boldsymbol{y} + \boldsymbol{z}) - v(\boldsymbol{y}) \geq v(\boldsymbol{x} + \boldsymbol{z}) - v(\boldsymbol{x})$ for $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^m_{\geq 0}$ and $\boldsymbol{y} \leq \boldsymbol{x}$.
2. For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m_{\geq 0}$ where $\boldsymbol{x} \leq \boldsymbol{y}$, we have $\nabla v(\boldsymbol{x}) \geq \nabla v(\boldsymbol{y})$.
3. $\frac{\partial^2}{\partial x_i \partial x_j} v(\boldsymbol{x}) \leq 0$ for all $\boldsymbol{x} \in \mathbb{R}^m_{\geq 0}$ and $i, j \in [m]$.

Proposition 2.2 (Lemma 4.2 of [OSS26])**.** *For a DR-submodular function v, define the Aumann–Shapley [AS74] vector $\boldsymbol{q}(\boldsymbol{x})$ for all $\boldsymbol{x} \in [0,1]^m$ by*

$$\boldsymbol{q}(\boldsymbol{x}) := \int_0^1 \nabla v(t\boldsymbol{x})\, dt.$$

Then $\boldsymbol{q}$ is a valid continuous supporting-price map for v.

2.3 Free-Disposal Value

Under the assumption that valuation functions have supporting prices, we can also consider the *free-disposal value* over the allocations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. Intuitively, for allocation vectors that do not satisfy the supply constraint $\sum_i \boldsymbol{x}_i \leq \mathbf{1}$, the free-disposal value is the best "trimming" of the allocations that satisfies the constraint. The trimming is evaluated with respect to the supporting prices of the original allocations. Formally, we define the objective as follows.

Definition 7 (Free-disposal value)**.** For valuation functions $v_1, \ldots, v_n$ with supporting-price maps $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$ and provisional allocation vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in A^m$ that do not necessarily satisfy the supply constraint, the *free-disposal value* is defined by the following optimization program.

$$\mathsf{FD}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) := \max_{\substack{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n \in A^m \\ \boldsymbol{y}_i \leq \boldsymbol{x}_i \ \forall i \in [n] \\ \sum_{i=1}^n \boldsymbol{y}_i \leq \mathbf{1}}} \sum_{i=1}^n \langle \boldsymbol{q}_i(\boldsymbol{x}_i), \boldsymbol{y}_i \rangle.$$

Competitiveness in the free-disposal setting. The input to an online algorithm in the free-disposal setting is the same as the input in a sample-based or GoG combinatorial prophet inequality. However, the algorithm's objective is to find (supply-infeasible) provisional allocations maximizing the free-disposal value. We say that an algorithm in this setting is α-competitive if the free-disposal value of its provisional allocation is α-competitive with respect to the hindsight optimum Opt defined previously.

Per-item decomposition. We observe that the free-disposal value is *decomposable* into multiple per-item *hindsight optima*.

Observation 2.3. *For any valuation functions $v_1, \ldots, v_n$ with supporting prices $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$, and allocation vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in A^m$, in the divisible-item setting, we have*

$$\mathsf{FD}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{j=1}^m \mathsf{Opt}\left(\{(q_{i,j}(\boldsymbol{x}_i), x_{i,j})\}_{i \in [n]}\right) = \sum_{j=1}^m \mathsf{FracKnap}\left(\{(q_{i,j}(\boldsymbol{x}_i), x_{i,j})\}_{i \in [n]}\right).$$

Here $\mathsf{FracKnap}\left(\{(q_{i,j}(\boldsymbol{x}_i), x_{i,j})\}_{i \in [n]}\right)$ is the value of the optimum fractional knapsack of size 1 with n items, where item i has density $q_{i,j}(\boldsymbol{x}_i)$ and weight $x_{i,j}$. If the items are indivisible, then we have

$$\mathsf{FD}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{j=1}^m \mathsf{Opt}\left(\{(q_{i,j}(\boldsymbol{x}_i), x_{i,j})\}_{i \in [n]}\right) = \sum_{j=1}^m \max_{i : x_{i,j} = 1} q_{i,j}(\boldsymbol{x}_i).$$

We also observe that the free-disposal value admits the following per-item *layer-cake representation*.

Proposition 2.4. *For valuation functions $v_1, \ldots, v_n$ with supporting-price maps $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$, and allocation vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in A^m$, we have*

$$\mathsf{FD}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{j=1}^m \int_0^\infty \min\left\{1, \sum_{i=1}^n x_{i,j} \cdot \mathbf{1}[q_{i,j}(\boldsymbol{x}_i) \geq w]\right\} dw.$$

2.4 Single-Item Prophet Inequality

We also consider the *single-item online allocation problem* in this work. This is a special case of the general combinatorial allocation problem in which $m = 1$. In particular, we focus on a variant of the problem called the *uniform-reward limited-demand variant*. In this variant, a buyer's valuation is a pair (q_i, x_i), where $q_i \in \mathbb{R}_{\geq 0}$ is the buyer's unit reward for this item, while $x_i \in A$ is the buyer's demand for the item. The algorithm is given these pairs sequentially and, at each step, must irrevocably choose an allocation x'_i for buyer i, where $x'_i \in A$ and $0 \leq x'_i \leq x_i$. As usual, the algorithm must satisfy supply feasibility, $\sum_{i=1}^{n} x'_i \leq 1$, while its reward is $\sum_{i=1}^{n} q_i x'_i$.

We note that in the indivisible case, this variant is exactly the classical prophet inequality, with the caveat that a buyer may not arrive (corresponding to $x_i = 0$). In the divisible case, this variant is the *prophet inequality for fractional knapsack*, recently studied by [NRS26] in the independent-draw single-sample setting. In both cases, a 1/2-competitive algorithm is known, and this ratio is tight.

For both cases, we will use SSPI to denote the single-item Game of Googol problem or an algorithm for that setting, and $\mathsf{Opt}(\{(q_i, x_i)\}_{i\in[n]})$ to denote the hindsight optimum. Similarly, we use k-SPI to denote the single-item k-*sample* Game of Googol problem or an algorithm for that setting. By a layer-cake representation,

$$\mathsf{Opt}(\{(q_i, x_i)\}_{i\in[n]}) = \int_0^\infty \min\left\{1, \sum_{i=1}^{n} x_i \cdot \mathbf{1}[q_i \geq w]\right\} dw.$$

3 A General Reduction from Combinatorial to Single-Item PI

In this section, we establish our general reduction from combinatorial allocation in the GoG setting to single-item allocation in the same setting. The reduction separates the two sources of feasibility in a combinatorial allocation problem: constraints on what may be allocated to each buyer, and supply constraints on each item.

When a buyer arrives, the algorithm first chooses a buyer-feasible *provisional allocation* together with supporting prices. The provisional allocation enforces the buyer-side combinatorial constraint—for example, a unit-demand buyer is provisionally assigned at most one item—while the supporting prices assign additive value to the items and certify their value under the original valuation. This additivity allows the remaining problem to decompose across items.

The reduction then uses a single-item prophet inequality for each item to resolve conflicts among the buyers provisionally assigned that item. These per-item algorithms enforce the supply constraints and approximate the *free-disposal value*, namely the maximum total supporting-price value that can be retained from the provisional allocation after resolving item conflicts. The free-disposal value therefore serves as the bridge between the two stages: the first stage seeks provisional allocations whose free-disposal value approximates the true combinatorial optimum, and the second converts that value into an ex-post feasible allocation.

At a high level, these two components interact online as follows. When buyer i arrives:

1. The online free-disposal algorithm observes v_i and produces a buyer-feasible provisional allocation $\boldsymbol{x}_i$, together with a corresponding supporting-price vector $\boldsymbol{q}_i$.

2. For each item $j \in [m]$, the j-th single-item prophet inequality receives the provisional allocation $x_{i,j}$ and its supporting price $q_{i,j}(\boldsymbol{x}_i)$, and decides whether buyer i receives item j in the final allocation.

If the free-disposal value $\mathsf{FD}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ is α-competitive with $\mathsf{Opt}(v_1, \ldots, v_n)$, and, for every item j, the corresponding single-item prophet inequality is β-competitive with $\mathsf{Opt}\left(\{(q_{i,j}(\boldsymbol{x}_i), x_{i,j})\}_{i\in[n]}\right)$, then the two stages compose to give an $\alpha\beta$-competitive combinatorial prophet inequality.

There is, however, one important subtlety in this composition. To apply a single-item prophet inequality separately to each item, its inputs across buyers must retain the required independence. Consequently, the provisional allocation produced for buyer i cannot depend on the realized valuations of the other buyers. We capture this requirement through *buyer-locality*. Formally, conditional on the shared sample, a provisional-allocation algorithm is buyer-local if the provisional allocation $\boldsymbol{x}_i$ and its supporting prices $\boldsymbol{q}_i$ depend only on buyer i's valuation v_i (and any private randomness associated with buyer i), and not on $\boldsymbol{v}_{-i}$. Hence, conditional on the sample, the induced inputs to each single-item prophet inequality remain independent across buyers.

At first sight, buyer-locality makes it difficult to construct good provisional allocations: each buyer must be treated without knowing which items other buyers will value or provisionally demand, even though the quality of the free-disposal value depends on how these demands interact across buyers. This is precisely where a sample from the underlying distribution proves useful. The sample provides a common source of information that can be used to coordinate the buyer-local allocation rules across buyers. Thus, while the provisional allocation for buyer i cannot depend on the realized valuations of other buyers, it may depend on the shared sample, which summarizes the distributional environment and guides all buyers' provisional allocations in a consistent way. Conditional on this sample, the decisions remain buyer-local and preserve the independence needed by the single-item prophet inequalities.

The main result of this section is the following theorem for the k-sample Game of Googol setting, which we prove in Sections 3.1 and 3.2. In Section 3.3 we provide an improved reduction for the single-sample special case from the SSCombPI to the SSPI.

Theorem 3.1. *If* FDAlg *is an α-competitive random-order free-disposal algorithm while k-*SPI *is a β_k-competitive k-sample single-item allocation algorithm, then Algorithm 1 using Algorithm 2 as a subroutine provides an $(\alpha\beta_k/2)$-competitive $(k+1)$-*SCombPI.

3.1 Reducing $(k+1)$-SCombPI to k-SPI

We instantiate the general reduction in the $(k+1)$-Sample Combinatorial Prophet Inequality setting. Of the $k+1$ samples provided, we designate one as a *coordination sample* and denote it by $\boldsymbol{s}^c$. The remaining k samples, denoted $\boldsymbol{s}^1, \cdots, \boldsymbol{s}^k$, are used for the underlying k-SPIs. The reduction uses two intermediate algorithms. The first, which we denote by PAlloc, is a single-sample provisional-allocation algorithm that uses the coordination sample to convert the buyers' valuations into a provisional allocation. The second, which we denote by k-SPI, is the k-sample prophet inequality. We run m instantiations of this algorithm, one for each item. Each takes k single-dimensional-valuation samples and item-specific provisional allocations and returns the final allocation for the buyer.

In order to preserve the symmetry of the samples and the real instance in the GoG setting of the k-SPIs, each of the k samples $\boldsymbol{s}^t$ is itself transformed first into a provisional allocation by PAlloc using the same coordination sample $\boldsymbol{s}^c$.

The algorithm is formally described in pseudocode as Algorithm 1. Before analyzing it, we define what it means for a provisional-allocation algorithm to be buyer-local.

Definition 8 (Buyer-local provisional-allocation algorithm)**.** A provisional-allocation algorithm PAlloc takes as input a coordination sample vector $\boldsymbol{s}^c$ and a real valuation vector $\boldsymbol{r}$, and outputs, for every buyer $i \in [n]$, a provisional allocation $\boldsymbol{x}_i$ together with a corresponding supporting-price

vector $\boldsymbol{q}_i$. We say that PAlloc is *buyer-local* if, conditioned on $\boldsymbol{s}^c$ and on any randomness shared across buyers, the pair $(\boldsymbol{x}_i, \boldsymbol{q}_i)$ depends only on r_i and on randomness local to buyer i.

Equivalently, after fixing $\boldsymbol{s}^c$ and the shared randomness, PAlloc can be implemented by algorithms $\mathsf{PAlloc}_1, \ldots, \mathsf{PAlloc}_n$, where PAlloc_i receives only r_i and produces $(\boldsymbol{x}_i, \boldsymbol{q}_i)$. In particular, PAlloc admits an online implementation when the valuations $r_1, \ldots, r_n$ arrive sequentially. Whenever PAlloc or PAlloc_i is used as an allocation-valued map below, we mean its allocation component; the associated supporting prices are computed using the same fixed supporting-price map for each valuation.

Algorithm 1 Combinatorial to Single-Item Prophet Inequalities Reduction

Input:

- Sample valuation vectors $(\boldsymbol{s}^c, \boldsymbol{s}^1, \ldots, \boldsymbol{s}^k)$, where $\boldsymbol{s}^c$ is the coordination sample,
- Arriving buyers with valuations $\boldsymbol{r}$,
- A buyer-local provisional-allocation algorithm PAlloc, and
- A k-sample single-item Game of Googol algorithm k-SPI for the appropriate setting A.

Output: Feasible allocation vectors $\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_n$.

Initialize m independent instances of k-SPI as $k\text{-}\mathsf{SPI}_1, \ldots, k\text{-}\mathsf{SPI}_m$.
Fix any internal randomness of PAlloc and use the same coin flips for all of its oracle calls.
for sample $t \in [k]$ **do**
  Let $\boldsymbol{x}_1^t, \ldots, \boldsymbol{x}_n^t \leftarrow \mathsf{PAlloc}(\boldsymbol{s}^c, \boldsymbol{s}^t)$.
  Let $\boldsymbol{q}_1^t, \ldots, \boldsymbol{q}_n^t$ be the supporting prices of the allocation vectors, i.e., $\boldsymbol{q}_i^t \leftarrow \boldsymbol{q}_{s_i^t}(\boldsymbol{x}_i^t)$.
  ▷ *Assemble the t-th sample for the k-SPIs.*
  for item $j \in [m]$ **do**
    Extract $(q_{i,j}^t, x_{i,j}^t)$ for each buyer $i \in [n]$ from the above.
    Let $(q_{1,j}^t, x_{1,j}^t), (q_{2,j}^t, x_{2,j}^t), \ldots, (q_{n,j}^t, x_{n,j}^t)$ be the t-th sample for k-SPI_j.
  end for
end for
  ▷ *Extract the per-item allocation from each k-SPI.*
for buyer $i \in [n]$ **do**
  $\widehat{\boldsymbol{x}}_i \leftarrow \mathsf{PAlloc}_i(\boldsymbol{s}^c, r_i)$. ▷ PAlloc *needs to be buyer-local for this to be well-defined.*
  Let $\widehat{\boldsymbol{q}}_i \leftarrow \boldsymbol{q}_{r_i}(\widehat{\boldsymbol{x}}_i)$ be the supporting price of this allocation vector.
  for item $j \in [m]$ **do**
    Send $(\widehat{q}_{i,j}, \widehat{x}_{i,j})$ as the real valuation of buyer i to k-SPI_j, and receive an allocation $x'_{i,j} \le \widehat{x}_{i,j}$.
  end for
  Aggregate the per-item allocations into $\boldsymbol{x}'_i$ and allocate it to buyer i.
end for

We measure the performance of a buyer-local algorithm through the free-disposal value in the Game of Googol model.

Definition 9 (Free-disposal competitiveness)**.** A buyer-local provisional-allocation algorithm PAlloc is *α-competitive under the free-disposal value* if the following holds for every collection of valuation pairs $\{v_{i,1}, v_{i,2}\}_{i\in[n]}$. Independently for each buyer i, let (s_i^c, r_i) be a uniformly random ordering of $(v_{i,1}, v_{i,2})$. Then

$$\mathbb{E}_{\boldsymbol{s}^c, \boldsymbol{r}}[\mathsf{FD}(\mathsf{PAlloc}(\boldsymbol{s}^c, \boldsymbol{r}))] \ge \alpha \cdot \mathbb{E}_{\boldsymbol{r}}[\mathsf{Opt}(\boldsymbol{r})],$$

where the expectation is over the independent random orderings and any internal randomness of PAlloc.

We observe that Algorithm 1 always outputs a feasible allocation, since for all j we have $\sum_{i=1}^{n} x'_{i,j} \leq 1$ by the guarantee of k-SPI_j. We show that if PAlloc produces allocation vectors with a good free-disposal value in the Game of Googol setting, then Algorithm 1 is a good $(k+1)$-sample combinatorial prophet inequality. The proof is a straightforward application of the per-item decomposition property of the free-disposal value, and we defer it to Appendix C.

Lemma 3.2. *Suppose that* PAlloc *is buyer-local and* α*-competitive under the free-disposal value, while* k-SPI *is* β*-competitive. Then, Algorithm 1 is* $\alpha\beta$*-competitive.*

3.2 Constructing a Competitive Buyer-Local Provisional-Allocation Algorithm

We are now tasked with designing a buyer-local free-disposal algorithm PAlloc in the single-sample GoG setting. Let $\boldsymbol{s}$ denote the sample vector and $\boldsymbol{r}$ the real valuation vector. Our approach is to transform a worst-case competitive online free-disposal algorithm into a buyer-local algorithm, using the coordination sample to exploit the symmetry between sample and real valuations.

Let FDAlg be an online free-disposal algorithm such that, on every input instance $\boldsymbol{v}$, the free-disposal value of its provisional allocation is competitive with $\mathsf{Opt}(\boldsymbol{v})$. We construct PAlloc from n coupled executions of FDAlg, one for each buyer. In the execution associated with buyer i, we run FDAlg on the hybrid instance obtained from $\boldsymbol{s}$ by replacing s_i with the real valuation r_i; all n executions use the same internal randomness. The provisional allocation (and supporting prices) assigned to buyer i in this execution are returned by PAlloc_i. By construction, PAlloc_i depends on the real instance only through r_i, and hence PAlloc is buyer-local.

Buyer-locality also separates the real arrival order from the order used within the simulated executions of FDAlg. The components PAlloc_i may be invoked in whatever order the real buyers arrive, while the hybrid instance supplied to FDAlg may be processed according to any ordering chosen by the reduction. We can therefore choose this internal ordering to obtain the strongest available guarantee for FDAlg. In particular, since FDAlg enjoys a better competitive ratio under random arrival order than under adversarial order, we process the hybrid instances in a uniformly random order and inherit this improved guarantee in the reduction.

Algorithm 2 Producing a Buyer-Local Free-Disposal Algorithm

Input:

- Sample valuations $\boldsymbol{s}$,
- Arriving buyers with valuations $\boldsymbol{r}$, and
- A random-order free-disposal algorithm FDAlg.

Output: A buyer-local algorithm PAlloc.

1: Let π be a random permutation of $\{1, \ldots, n\}$, and let $\Sigma^{(0)}$ be the initialization state of FDAlg including all internal coin flips.
2: **for** t from 1 to n **do**
3: Let $i \leftarrow \pi_t$ be the current buyer index.
4: Record $\Sigma_i \leftarrow \Sigma^{(t-1)}$ as the state of the algorithm before s_i is processed.
5: $(\boldsymbol{z}_i, \Sigma^{(t)}) \leftarrow \mathsf{FDAlg}(\Sigma^{(t-1)}, s_i)$. ▷ *$\boldsymbol{z}_i$ is used in analysis only*
6: **end for**
7: For every buyer $i \in [n]$, let $\mathsf{PAlloc}_i(\boldsymbol{s}, r_i)$ return the allocation returned by $\mathsf{FDAlg}(\Sigma_i, r_i)$.
8: **return** $\mathsf{PAlloc} \leftarrow (\mathsf{PAlloc}_1, \ldots, \mathsf{PAlloc}_n)$.

We now show that PAlloc inherits the competitive ratio of FDAlg within a factor of 2.

Lemma 3.3. *Suppose* FDAlg *is any random-order free-disposal algorithm. Fix* $(v_{i,1}, v_{i,2})$ *for all buyers* i*, and suppose* $\boldsymbol{s}$ *and* $\boldsymbol{r}$ *are sampled according to the Game of Googol from these fixed valuations. Then* PAlloc *from Algorithm 2 is a buyer-local algorithm, and*

$$\mathbb{E}[\mathsf{FD}(\mathsf{PAlloc}(\boldsymbol{s}, \boldsymbol{r}))] \geq \frac{1}{2} \cdot \mathbb{E}[\mathsf{FD}(\mathsf{FDAlg}(\boldsymbol{r}))].$$

Proof. PAlloc is buyer-local by construction, so we will only prove the second claim. Fix the random arrival order π and the internal randomness of FDAlg. By relabeling the buyers, we assume that π is the identity permutation. Let $\boldsymbol{z}_i = \mathsf{FDAlg}(\Sigma_i, s_i)$ denote the allocation made by FDAlg on $\boldsymbol{s}$ to agent i, and $\boldsymbol{x}_i = \mathsf{FDAlg}(\Sigma_i, r_i)$ the provisional allocation made to i by $\mathsf{PAlloc}_i(\boldsymbol{s}, r_i)$.

Let $q_{v,j}(\boldsymbol{x})$ denote the j-th coordinate of $\boldsymbol{q}_v(\boldsymbol{x})$. For some agent $k \in [n]$, consider the free-disposal value obtained by the partial allocation $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$:

$$\mathsf{FD}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) = \sum_{j=1}^{m} \int_0^\infty \underbrace{\min\left\{1, \sum_{i=1}^{k} x_{i,j} \cdot \mathbf{1}[q_{r_i,j}(\boldsymbol{x}_i) \geq w]\right\}}_{y_{j,w}(\boldsymbol{x}_{1..k})} dw.$$

To prove the lemma, it suffices to show that for every value $w \geq 0$ and every initial state $\Sigma^{(0)}$, at the end of the algorithm we have

$$\mathbb{E}[y_{j,w}(\boldsymbol{x}_{1..n})] \geq \frac{1}{2} \cdot \mathbb{E}[y_{j,w}(\boldsymbol{z}_{1..n})],$$

where the expectation is taken over the random assignment of $v_{i,1}$ and $v_{i,2}$ to s_i and r_i for all $i \in [n]$. Then integrating over w and summing over j provides $\mathbb{E}[\mathsf{FD}(\mathsf{PAlloc}(\boldsymbol{s}, \boldsymbol{r}))] \geq \frac{1}{2} \cdot \mathbb{E}[\mathsf{FD}(\mathsf{FDAlg}(\boldsymbol{s}))]$, and the lemma follows by the symmetry of $\boldsymbol{s}$ and $\boldsymbol{r}$.

We prove the claim by showing

$$\mathbb{E}[y_{j,w}(\boldsymbol{x}_{1..n})] \geq \frac{1}{2} \cdot \mathbb{E}[y_{j,w}(\boldsymbol{x}_{1..n} \cup \boldsymbol{z}_{1..n})] \geq \frac{1}{2} \cdot \mathbb{E}[y_{j,w}(\boldsymbol{z}_{1..n})],$$

where $y_{j,w}(\boldsymbol{x}_{1..n} \cup \boldsymbol{z}_{1..n}) := \min\{1, \sum_{i=1}^{n} (x_{i,j} \cdot \mathbf{1}[q_{r_i,j}(\boldsymbol{x}_i) \geq w] + z_{i,j} \cdot \mathbf{1}[q_{s_i,j}(\boldsymbol{z}_i) \geq w])\}$ can be seen as the "merged" free-disposal contribution of item j at weight level w.

Fix any partial realization of $s_1, \ldots, s_{k-1}$ for some $k \in [n]$, which also fixes $\Sigma^{(k-1)}$ to be the state of FDAlg before the k-th buyer. We now show that

$$\mathbb{E}[y_{j,w}(\boldsymbol{x}_{1..k}) - y_{j,w}(\boldsymbol{x}_{1..k-1}) \mid \boldsymbol{s}_{1..k-1}] \geq \frac{1}{2}\left(y_{j,w}(\boldsymbol{x}_{1..k} \cup \boldsymbol{z}_{1..k}) - y_{j,w}(\boldsymbol{x}_{1..k-1} \cup \boldsymbol{z}_{1..k-1})\right) \tag{1}$$

where the RHS is deterministic conditional on $s_1, \ldots, s_{k-1}$ because it does not depend on the random assignment of s_k and r_k to the two k-th valuations.

The claim follows immediately from the concavity of $f(x) = \min\{1, x\}$. Let

$$\begin{aligned}
A &= \sum_{i=1}^{k-1} x_{i,j} \cdot \mathbf{1}[q_{r_i,j}(\boldsymbol{x}_i) \geq w], \\
B &= \sum_{i=1}^{k-1} \left(x_{i,j} \cdot \mathbf{1}[q_{r_i,j}(\boldsymbol{x}_i) \geq w] + z_{i,j} \cdot \mathbf{1}[q_{s_i,j}(\boldsymbol{z}_i) \geq w]\right), \\
c &= u_{k,j} \cdot \mathbf{1}[q_{v_{k,1},j}(\boldsymbol{u}_k) \geq w] \quad \text{where } \boldsymbol{u}_k = \mathsf{FDAlg}(\Sigma^{(k-1)}, v_{k,1}),
\end{aligned}$$

$$d = u'_{k,j} \cdot \mathbf{1}[q_{v_{k,2},j}(\boldsymbol{u}'_k) \geq w] \quad \text{where } \boldsymbol{u}'_k = \mathsf{FDAlg}(\Sigma^{(k-1)}, v_{k,2}),$$

then $0 \leq A \leq B$ and $c, d \geq 0$. Furthermore,

$$\mathbb{E}[y_{j,w}(\boldsymbol{x}_{1..k}) - y_{j,w}(\boldsymbol{x}_{1..k-1}) \mid \boldsymbol{s}_{1..k-1}] = \frac{1}{2}\left(f(A+c) - f(A) + f(A+d) - f(A)\right)$$

while

$$y_{j,w}(\boldsymbol{x}_{1..k} \cup \boldsymbol{z}_{1..k}) - y_{j,w}(\boldsymbol{x}_{1..k-1} \cup \boldsymbol{z}_{1..k-1}) = f(B+c+d) - f(B).$$

Finally, by the concavity of $f(x)$, $f(B+c+d) - f(B) = f(B+c+d) - f(B+d) + f(B+d) - f(B) \leq f(A+c) - f(A) + f(A+d) - f(A)$. This completes the proof of Equation (1). □

We then have the following corollary.

Corollary 3.4. *If* FDAlg *is an α-competitive random-order free-disposal algorithm, then* PAlloc *as constructed above is an $\alpha/2$-competitive buyer-local algorithm.*

By combining with Lemma 3.2, we obtain Theorem 3.1.

3.3 From Two Samples to One Sample

Finally, we turn Algorithm 1 with $k = 1$, which uses two samples $(\boldsymbol{s}^c, \boldsymbol{s}^1)$, into an algorithm that uses only one sample $\boldsymbol{s}'$. Such a reduction is already known from [DKL+24], which proves that any k-SCombPI algorithm can be turned into an SSCombPI algorithm at a loss factor of $k+1$. For Algorithm 1, this result translates to a loss factor of 3 in the competitive ratio, which amounts to an $(\alpha\beta/6)$-competitive SSCombPI algorithm given an α-competitive free-disposal algorithm and a β-competitive SSPI algorithm. We show that for our specific framework, we can reduce the incurred loss to a factor of 2 or less.

The main idea here is to split the buyers into two types: the *coordinating* and the *competition* buyers. In particular, suppose each buyer $i \in [n]$ has a sample valuation s'_i and real valuation r'_i. Then we construct the two-sample instance $(\boldsymbol{s}^c, \boldsymbol{s}^1, \boldsymbol{r})$ to feed into Algorithm 1 as follows. Fix a parameter $q \in [0,1]$ and for each buyer i independently:

1. With probability q, buyer i plays the role of a coordinating buyer. In particular, $(s^c_i, s^1_i, r_i) \leftarrow (s'_i, 0, 0)$.
2. With probability $1-q$, buyer i is a competition buyer. These buyers directly compete in the free-disposal algorithm. In particular, $(s^c_i, s^1_i, r_i) \leftarrow (0, s'_i, r'_i)$.

For completeness, we provide the pseudocode of the algorithm in Algorithm 3.

We now prove a guarantee for Algorithm 3 assuming black-box guarantees for FDAlg and SSPI. For SSPI, we simply assume that the algorithm is β-competitive. For FDAlg, we assume that it has a *smooth competitive ratio* that depends on the fraction of buyers that get zeroed out. Formally,

Definition 10 (Smooth competitive ratio)**.** Fix any (random-order) online free-disposal allocation algorithm FDAlg. Let $\Gamma : [0,1] \to [0,1]$ be any increasing function. Then, we say FDAlg has a *smooth competitive ratio* Γ if for any valuation vector $\boldsymbol{v}$ on n buyers, and for any $q \in [0,1]$, the following is true.

Algorithm 3 Single-Sample Combinatorial Online Allocation

Input:

- Sample valuations $\boldsymbol{s}'$ and arriving buyers with valuations $\boldsymbol{r}'$,
- A random-order free-disposal allocation algorithm FDAlg,
- A single-sample single-item Game of Googol algorithm SSPI for the appropriate setting A, and
- A parameter $q \in [0,1]$.

Output: Feasible allocation vectors $\boldsymbol{x}'_1, \dots, \boldsymbol{x}'_n$.

Independently draw $H_i \sim \text{Bernoulli}(q)$ for every buyer $i \in [n]$, where $H_i = 1$ means that buyer i is coordinating and $H_i = 0$ means that buyer i is competing.
for buyer $i \in [n]$ **do**
  if $H_i = 1$ **then** $(s_i^c, s_i^1) \leftarrow (s'_i, 0)$ **else** $(s_i^c, s_i^1) \leftarrow (0, s'_i)$.
end for
Construct the buyer-local algorithm PAlloc from $\boldsymbol{s}^c$ and FDAlg as in Algorithm 2.
Initialize Algorithm 1 with $k = 1$, sample valuations $(\boldsymbol{s}^c, \boldsymbol{s}^1)$, buyer-local algorithm PAlloc, and single-item algorithm SSPI.
for each arriving buyer i with valuation r'_i **do**
  Let $r_i \leftarrow 0$ if $H_i = 1$, and let $r_i \leftarrow r'_i$ otherwise.
  Send r_i to Algorithm 1 and receive its final allocation $\boldsymbol{x}'_i$.
  if $H_i = 1$ **then** allocate $\boldsymbol{x}'_i \leftarrow \mathbf{0}$ **else** allocate $\boldsymbol{x}'_i$.
  Let $\boldsymbol{x}_i \leftarrow \mathsf{PAlloc}_i(\boldsymbol{s}^c, r_i)$ be the provisional allocation used inside Algorithm 1, and let $\boldsymbol{z}_i \leftarrow \mathsf{PAlloc}_i(\boldsymbol{s}^c, s_i^c)$ be the allocation from the sample run of FDAlg. ▷ *Only for analysis*
end for

Randomly construct the valuation vector $\boldsymbol{v}'$ as follows: for every buyer i independently, $v'_i = v_i$ with probability q and $v'_i = 0$ otherwise. Run FDAlg on $\boldsymbol{v}'$ and obtain allocation vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$. Then,

$$\mathbb{E}[\mathsf{FD}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)] \geq \Gamma(q) \cdot \mathsf{Opt}(v_1, \dots, v_n)$$

where the left-hand side is the expected free-disposal value of $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ evaluated on $\boldsymbol{v}'$, and the expectation is taken over $\boldsymbol{v}'$ and the internal randomness of FDAlg.

In particular, $\Gamma(1)$ is a competitive guarantee of FDAlg. For the random-order case, the following viewpoint is equivalent. Assume that the algorithm simply skips over zero valuations, and let $n' \sim \text{Binom}(n, q)$. Then the expected free-disposal value from the first n' buyers of the random order π is at least $\Gamma(q)$ of the hindsight optimum.

Fix the valuations $v_{i,1}, v_{i,2}$ for all buyers i, and assume $\boldsymbol{s}'$ and $\boldsymbol{r}'$ are drawn in a Game of Googol fashion from these valuations. We first show the following technical lemma.

Lemma 3.5. *For every $q \in [0,1]$, $\mathbb{E}[\mathsf{FD}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)] \geq (1-q)\mathbb{E}[\mathsf{FD}(\boldsymbol{z}_1, \dots, \boldsymbol{z}_n)]$.*

This inequality is the analogue of Lemma 3.3 promised above. Indeed, when $q = 1/2$, its instancewise inequality is the same real-versus-sample replay comparison as Lemma 3.3, where the replayed allocations $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ have at least half the free-disposal value of the full sample allocations $\boldsymbol{z}_1, \dots, \boldsymbol{z}_n$. Here the real side consists of the competing buyers and the sample side consists of the coordinating buyers; Lemma 3.5 extends the factor $1/2$ to the biased factor $1-q$. We defer its proof to Appendix C.

Noting that $\mathbb{E}[\mathsf{FD}(\boldsymbol{z}_1,\dots,\boldsymbol{z}_n)] \geq \Gamma(q)\cdot\mathbb{E}[\mathsf{Opt}(s'_1,\dots,s'_n)]$ by the smooth competitive ratio definition of FDAlg, we get the following guarantee for Algorithm 3.

Theorem 3.6. *If* FDAlg *has a smooth competitive ratio of* Γ*, while* SSPI *is* β*-competitive, then for every* $q\in[0,1]$*, Algorithm 3 is a* $\beta(1-q)\Gamma(q)$*-competitive* SSCombPI *algorithm.*

Proof. Fix any $q\in[0,1]$. By the smooth competitive ratio of FDAlg combined with Lemma 3.5, we have

$$\begin{aligned}\mathbb{E}[\mathsf{FD}(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)] &\geq (1-q)\Gamma(q)\cdot\mathbb{E}[\mathsf{Opt}(s'_1,\dots,s'_n)]\\ &= (1-q)\Gamma(q)\cdot\mathbb{E}[\mathsf{Opt}(r'_1,\dots,r'_n)].\end{aligned}$$

Condition on the role coins $H_1,\dots,H_n$, the coordination sample $\boldsymbol{s}^c$, and the internal randomness of PAlloc. For a competing buyer, $s_i^c=0$ reveals no information about the random ordering of $(v_{i,1},v_{i,2})$ into (s_i^1,r_i); for a coordinating buyer, $(s_i^1,r_i)=(0,0)$. Thus these pairs remain independent Game of Googol pairs under this conditioning. As in the proof of Lemma 3.2, the per-item guarantee and the supporting-price property give $\mathbb{E}[\sum_i r_i(\boldsymbol{x}'_i)]\geq\beta\mathbb{E}[\mathsf{FD}(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)]$. Since $r'_i\geq r_i$ pointwise and coordinating buyers receive zero, this also lower-bounds the expected welfare on the original real instance $\boldsymbol{r}'$, proving the theorem. □

To apply this theorem, whenever the maximum is attained, choosing $q\in\arg\max_{u\in[0,1]}(1-u)\Gamma(u)$ yields the guarantee $\beta\max_{u\in[0,1]}(1-u)\Gamma(u)$. As an immediate observation, since we can naively bound $\Gamma(q)\geq\alpha q$ where α is the competitive ratio of FDAlg, we immediately get the following corollary.

Corollary 3.7. *If* FDAlg *is an* α*-competitive random-order free-disposal algorithm and* SSPI *is* β*-competitive, then with* $q=1/2$*, Algorithm 3 is an* $(\alpha\beta/4)$*-competitive* SSCombPI *algorithm.*

3.4 Computational Considerations

Given oracle access to the FDAlg, k-SPI, and SSPI algorithms, Algorithms 1, 2, and 3 run in polynomial time. Therefore, we immediately obtain efficient $(k+1)$-SCombPI and SSCombPI algorithms, assuming that the underlying FDAlg, k-SPI, and SSPI algorithms can be implemented efficiently.

4 Random-Order Free-Disposal Algorithms

In this section, we provide random-order free-disposal algorithms for several classes of valuations. In particular, for the indivisible setting, we show that the greedy algorithm [DNS05] obtains a competitive ratio strictly greater than $1/2$ under random buyer arrivals. Notably, we show unconditionally that $1/2$ is the best competitive ratio for randomized online free-disposal algorithms under *adversarial* XOS-buyer arrivals, regardless of computation time. This result demonstrates the ability of our framework to leverage random order in single-sample prophet inequalities. We also provide several first results on free-disposal algorithms for the divisible case.

4.1 Indivisible-Item Free-Disposal Algorithm

Consider the following buyer-wise greedy algorithm (Algorithm 4), first studied by [DNS05] in the context of welfare maximization for XOS buyers and then used extensively in [DKL+24] for the first

Algorithm 4 Buyer-Wise Greedy [DNS05, DKL+24]

Input: Arriving buyers with valuations $\boldsymbol{v}$.
Output: Allocation subsets $S_1, \ldots, S_n$ that need not be feasible.

```
Initialize p^(0) ← 0.
for buyer i ∈ [n] do
    S_i ← arg max_{S⊆[m]} (v_i(S) − p^(i−1)(S)).
    for item j ∈ [m] do
        p_j^(i) ← q_{i,j}(S_i) if j ∈ S_i, else p_j^(i) ← p_j^(i−1).
    end for
    Allocate S_i to buyer i.
end for
```

single-sample combinatorial prophet inequality for XOS buyers. Here, for a set $S \subseteq [m]$, we use the notation $\boldsymbol{p}(S)$ to denote $\sum_{j \in S} p_j$.

We prove that Algorithm 4 achieves a competitive ratio of 0.5096 and a smooth competitive ratio of $\Gamma(q) = q - q^2/2$, under the free-disposal value in the random-order buyer-arrival setting. The competitive ratio is a strict improvement over the ratio of $1/2$ achieved by the same algorithm under an adversarial arrival order [DKL+24, DNS05]. In fact, we also show that no randomized online algorithm has a competitive ratio greater than $1/2$ under the free-disposal value for an adversarial arrival order, regardless of computation time.

Theorem 4.1. *For buyers with XOS valuations that arrive in a uniformly random order, Algorithm 4 is* 0.5096*-competitive and* Γ*-smooth-competitive under the free-disposal value, where* $\Gamma(q) = q - q^2/2$. *Furthermore, the algorithm runs in polynomial time and uses polynomially many demand and supporting-price queries.*

Theorem 4.2. *For buyers with XOS valuations that arrive in an adversarial order, no randomized online algorithm can achieve a competitive ratio of* $1/2 + \epsilon$ *under the free-disposal value for any constant* $\epsilon > 0$*, regardless of computation.*

We defer the proof of the hardness result to Appendix D and focus on the algorithmic side in this section. Our proof is a reduction to online monotone submodular maximization over a simple partition matroid. This problem was studied by [BFFG20] as a generalization of online monotone submodular maximization. We use the following result as a black box.

Theorem 4.3. *Let* $\mathcal{N}$ *be a finite ground set partitioned into nonempty, pairwise disjoint parts* $P_1, \ldots, P_\ell$*, and let* $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ *be a nonnegative monotone submodular function. A set* $S \subseteq \mathcal{N}$ *is feasible if* $|S \cap P_i| \leq 1$ *for every* $i \in [\ell]$*, and the objective is to maximize* $f(S)$ *over all such sets. The parts are revealed one at a time in an order given by a uniformly random permutation* π *of* $[\ell]$*. Upon seeing* $P_{\pi(t)}$*, the algorithm must immediately and irrevocably choose an element from this part.*

Starting with $A_0 \leftarrow \emptyset$*, consider the greedy choices*

$$u_t \leftarrow \operatorname*{arg\,max}_{u \in P_{\pi(t)}} f(u \mid A_{t-1}), \qquad A_t \leftarrow A_{t-1} \cup \{u_t\},$$

where $f(u \mid A) := f(A \cup \{u\}) - f(A)$ *and ties are broken arbitrarily. Then*

$$\mathbb{E}_\pi\big[f(A_\ell)\big] \geq 0.5096 \cdot \max_{\substack{S \subseteq \mathcal{N} \\ |S \cap P_i| \leq 1}} f(S).$$

In other words, random-order greedy is 0.5096*-competitive for maximizing f subject to the simple partition matroid constraint [BFFG20, Theorem 1]. Furthermore, random-order greedy is Γ-smooth-competitive as well, where $\Gamma(q) = q - q^2/2$ [BFFG20, essentially Lemma 3.4].*

In [BFFG20], the 0.5096 competitive ratio is stated as is, while the smooth competitive ratio is stated for the free-disposal value of exactly the first q fraction of arriving buyers, which is not exactly our definition of smooth competitive ratio. We therefore prove the independently pruned version here in Lemma D.1, as well as an extension where the algorithm selects an approximately greedy choice at every step.

We begin the reduction. For every buyer i and bundle $S \subseteq [m]$, let $e_{i,S} = (i, S)$ denote a configuration. Let

$$\mathcal{N} = \{e_{i,S} : i \in [n],\ S \subseteq [m]\}$$

and

$$P_i = \{e_{i,S} : S \subseteq [m]\}.$$

The sets $P_1, \ldots, P_n$ define a simple partition matroid: a feasible partial allocation contains at most one element from each P_i. For $\mathcal{I} \subseteq \mathcal{N}$, define

$$f(\mathcal{I}) := \sum_{j=1}^{m} \max_{e_{i,S} \in \mathcal{I}, S \ni j} q_{i,j}(S).$$

Observe that the partial allocations $(S_i)_{i \in I}$ for $I \subseteq [n]$ have a one-to-one mapping to a feasible subset $\mathcal{I} = \{e_{i,S_i} : i \in I\}$ of $\mathcal{N}$. Furthermore, $f(\mathcal{I})$ is exactly the free-disposal value of the encoded partial allocation. Therefore, finding good free-disposal allocations is equivalent to finding a feasible set with a good f value.

We first prove that f is submodular.

Lemma 4.4. *The function f is normalized, nonnegative, monotone, and submodular.*

Proof. We prove submodularity here. For every item j, define

$$p_j(\mathcal{I}) := \max_{e_{i,S} \in \mathcal{I}, S \ni j} q_{i,j}(S)$$

to be the contribution of item j to f. If $\mathcal{I} \subseteq \mathcal{J}$, then $p_j(\mathcal{I}) \le p_j(\mathcal{J})$ for every j. Consequently, the marginal contribution of $e_{i,S}$ with $e_{i,S} \notin \mathcal{J}$ is

$$f(e_{i,S} \mid \mathcal{I}) = \sum_{j \in S} (q_{i,j}(S) - p_j(\mathcal{I}))^+ \ge \sum_{j \in S} (q_{i,j}(S) - p_j(\mathcal{J}))^+ = f(e_{i,S} \mid \mathcal{J}),$$

which is the diminishing-returns definition of submodularity. If $e_{i,S} \in \mathcal{J}$, its marginal on $\mathcal{J}$ is 0, so monotonicity gives the required inequality. □

It remains to show that the greedy algorithm for the configuration instance behaves exactly like Algorithm 4 on the original instance. Suppose that buyers $1, \ldots, i-1$ have been processed. Let $\mathcal{I}_{i-1}$ contain the encoded partial allocation, and let $p_j = p_j(\mathcal{I}_{i-1})$. For a candidate bundle S, define its buyer-wise utility and marginal contribution by

$$U_i(S) := v_i(S) - \boldsymbol{p}(S) = \sum_{j \in S} (q_{i,j}(S) - p_j)\,,$$
$$\Delta_i(S) := f(e_{i,S} \mid \mathcal{I}_{i-1}) = \sum_{j \in S} (q_{i,j}(S) - p_j)^+\,.$$

The following lemma shows that every maximizer of U_i is also a maximizer of Δ_i. Hence, Algorithm 4 is precisely greedy on the configuration instance.

Lemma 4.5. *We have* $\max_{S\subseteq[m]} U_i(S) = \max_{S\subseteq[m]} \Delta_i(S)$. *Moreover, every maximizer of* U_i *is also a maximizer of* Δ_i.

Proof. Clearly, $U_i(S) \le \Delta_i(S)$. For the reverse direction, fix S and let $S' = \{j \in S : q_{i,j}(S) > p_j\}$. Applying the supporting-price property to $S' \subseteq S$ gives

$$U_i(S') = v_i(S') - \boldsymbol{p}(S') \ge \sum_{j\in S'} (q_{i,j}(S) - p_j) = \Delta_i(S).$$

Taking the maximum on both sides gives $\max_S U_i(S) \ge \max_S \Delta_i(S)$, while the pointwise inequality gives the reverse inequality. If S_i maximizes U_i, then

$$U_i(S_i) \le \Delta_i(S_i) \le \max_S \Delta_i(S) = \max_S U_i(S) = U_i(S_i),$$

so all inequalities are equalities and S_i also maximizes the marginal contribution. Moreover, $U_i(S_i) = \Delta_i(S_i)$ implies $q_{i,j}(S_i) \ge p_j^{(i-1)}$ for every $j \in S_i$, so the price update maintains $p_j^{(i)} = p_j(\mathcal{I}_i)$. □

Proof of Theorem 4.1. A uniformly random buyer order is a uniformly random order of the parts P_i. By Lemma 4.4, the configuration objective is monotone submodular; by Lemma 4.5, Algorithm 4 is the greedy algorithm for this objective. Thus, the 0.5096 guarantee follows from Theorem 4.3. For the smooth guarantee, independently zeroing each buyer with probability $1-q$ is equivalent to independently omitting the corresponding part P_i: a zero-valued buyer can select only a bundle of currently zero-priced items and therefore changes neither the price vector nor the free-disposal value. Hence, Lemma D.1 applies. In both cases, f's optimum is exactly the best free-disposal value of any allocation, which is at least $\mathsf{Opt}(v_1, \dots, v_n)$. □

4.2 Divisible-Item Free-Disposal Algorithms

In this section, we design random-order free-disposal algorithms for divisible items. We present two algorithms under different assumptions on the valuations, where a stronger assumption on the underlying valuations leads to a better final competitive ratio.

Theorem 4.6. *For buyers with continuous capped-XOS valuations, Algorithm 5 is* 0.5096*-competitive and* Γ*-smooth-competitive under the free-disposal value when buyers arrive in a uniformly random order, where* $\Gamma(q) = q - q^2/2$. *Furthermore, if the valuations are also concave, then for any* $\epsilon > 0$, *the algorithm can be implemented in* $\mathrm{poly}(n, m, \log(1/\epsilon))$ *time using* $\mathrm{poly}(n, m, \log(1/\epsilon))$ *value, first-order (supergradient), and supporting-price queries, up to an additive* ϵ *loss in the competitive ratios.*

Theorem 4.7. *For buyers with DR-submodular and concave valuations, Algorithm 6 is* $(1-1/e) \approx$ 0.6321*-competitive under the free-disposal value, even under an adversarial buyer arrival order.*

Before describing the algorithms, we define a quantity that will help us decompose the free-disposal value. For any allocation vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$, item j, and weight level w, let $y_j(w)$ be the fraction of item j allocated at supporting prices whose value is at least w. Formally,

$$y_j(w) = \min\left\{1, \sum_{i=1}^{n} x_{i,j} \cdot \mathbf{1}[q_{i,j}(\boldsymbol{x}_i) \ge w]\right\}.$$

Then, the free-disposal value is exactly

$$\mathsf{FD}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \sum_{j=1}^{m} \int_0^\infty y_j(w)\, dw.$$

We then let $y_j = \int_0^\infty y_j(w)\,dw$ be the contribution of item j to the free-disposal value. We also observe that the same definition can be extended to partial allocations, i.e., those for which only some of the $\boldsymbol{x}_i$ have been determined.

4.2.1 A Continuous Extension of Buyer-Wise Greedy

We introduce the continuous extension of the buyer-wise greedy algorithm (Algorithm 4) as follows. Observe that in the indivisible-item case, $p_j^{(i)}$ is exactly the contribution of item j to the free-disposal value at step i, since the supporting-price property ensures that $p_j^{(i)}$ increases whenever it is updated. In other words, the set S_i that we choose at step i in Algorithm 4 is exactly the set that increases the free-disposal value the most.

Following this viewpoint, we state the continuous buyer-wise greedy algorithm as Algorithm 5.

Algorithm 5 Continuous Buyer-Wise Greedy

Input: Arriving buyers with continuous capped-XOS valuations $\boldsymbol{v}$.
Output: Allocation vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ that need not be feasible.

1: **for** buyer $i \in [n]$ **do**
2: Choose $\boldsymbol{x}_i \in [0,1]^m$ maximizing the surrogate objective Φ_i from Claim 4.9, using a fixed deterministic tie-breaking rule.
3: Allocate $\boldsymbol{x}_i$ to buyer i.
4: **end for**

The proof idea of Theorem 4.6 is similar to that of Theorem 4.1; that is, it is essentially a reduction to online monotone submodular maximization over a partition matroid. However, we cannot directly follow the approach used in the indivisible-item case, since there are infinitely many possible choices for $\boldsymbol{x}_i$ at any point, making the reduced instance infinite and potentially invalidating the use of Theorem 4.3. Instead, we construct a finite instance using *only the possible outcomes for $\boldsymbol{x}_i$ produced by running Algorithm 5, together with the hindsight-optimal choice of $\boldsymbol{x}_i$*. There are at most $n! + 1$ such choices for $\boldsymbol{x}_i$, so the instance is finite.

Formally, we first fix $v_1, \ldots, v_n$, and let $\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_n^*$ be the optimal allocations, which exist by continuity of the valuations and compactness of the feasible allocation space. For every buyer i, we record the set

$$O_i = \{\boldsymbol{x}_i \text{ produced by Algorithm 5 over all possible buyer orders}\} \cup \{\boldsymbol{x}_i^*\}.$$

We will only consider the configurations $e_{i,\boldsymbol{x}_i}$ for $\boldsymbol{x}_i \in O_i$, and there are finitely many of them. As in the indivisible case, let

$$\mathcal{N} = \{e_{i,\boldsymbol{x}_i} : i \in [n], \boldsymbol{x}_i \in O_i\}$$

and

$$P_i = \{e_{i,\boldsymbol{x}_i} : \boldsymbol{x}_i \in O_i\}.$$

We now define the reward function f as follows. For $\mathcal{I} \subseteq \mathcal{N}$, let

$$f(\mathcal{I}) := \mathsf{FD}(\{\boldsymbol{x}_i : e_{i,\boldsymbol{x}_i} \in \mathcal{I}\}).$$

The crucial observation here is that any possible output of Algorithm 5 is encoded as a feasible subset $\mathcal{I} \subseteq \mathcal{N}$, with the free-disposal value being exactly $f(\mathcal{I})$. Similarly, since $\boldsymbol{x}_i^*$ is included in O_i, the optimum allocation is also encoded as a feasible subset. Finally, due to the greedy nature of

Algorithm 5, we can still show that the behavior of the greedy algorithm from Theorem 4.3 on the reduced instance is similar to that of Algorithm 5 for the same buyer arrival order.

Again, we first prove that f is submodular.

Lemma 4.8. *The function f is normalized, nonnegative, monotone, and submodular.*

Proof. We prove submodularity here. By definition,

$$f(\mathcal{I}) = \sum_{j=1}^{m} \int_0^\infty \underbrace{\min\left\{1, \sum_{e_{i,\boldsymbol{x}_i} \in \mathcal{I}} x_{i,j} \cdot \mathbf{1}[q_{i,j}(\boldsymbol{x}_i) \geq w]\right\}}_{y_{j,w}(\mathcal{I})} dw.$$

If we can prove that $y_{j,w}(\mathcal{I})$ is submodular for all $j \in [m], w \geq 0$, then we are done. This is straightforward to show: for $\mathcal{I} \subseteq \mathcal{J}$, we have $y_{j,w}(\mathcal{I}) \leq y_{j,w}(\mathcal{J})$, so the marginal contribution of $e_{i,\boldsymbol{x}_i}$ with $e_{i,\boldsymbol{x}_i} \notin \mathcal{J}$ is

$$\begin{aligned} y_{j,w}(e_{i,\boldsymbol{x}_i} \mid \mathcal{I}) &= \min\{x_{i,j} \cdot \mathbf{1}[q_{i,j}(\boldsymbol{x}_i) \geq w], 1 - y_{j,w}(\mathcal{I})\} \\ &\geq \min\{x_{i,j} \cdot \mathbf{1}[q_{i,j}(\boldsymbol{x}_i) \geq w], 1 - y_{j,w}(\mathcal{J})\} = y_{j,w}(e_{i,\boldsymbol{x}_i} \mid \mathcal{J}) \end{aligned}$$

which is the diminishing-returns definition of submodularity. If $e_{i,\boldsymbol{x}_i} \in \mathcal{J}$, its marginal on $\mathcal{J}$ is 0, so monotonicity gives the required inequality. □

Combining this lemma with Theorem 4.3 gives us the desired competitive ratio. Regarding runtime, however, even with the concavity of v_i, the increment in the free-disposal value is not necessarily concave, so it is not obvious how to find the greedy choice $\boldsymbol{x}_i$.

To this end, observe the following: when we add the vector $\boldsymbol{x}_i$ to the partial allocation, the free-disposal value is updated to include a fraction $\boldsymbol{c} \leq \boldsymbol{x}_i$ of $\boldsymbol{q}_i = \boldsymbol{q}_i(\boldsymbol{x}_i)$, while the previous partial allocation is *truncated* to have total weight $\mathbf{1} - \boldsymbol{c}$. In particular, for any $\boldsymbol{c} \in [0,1]^m$, if we define the *truncated free-disposal value*

$$\mathsf{FD}_{\boldsymbol{c}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}) := \sum_{j=1}^{m} \int_0^\infty \min\left\{c_j, \sum_{i'=1}^{i-1} x_{i',j} \cdot \mathbf{1}[q_{i',j}(\boldsymbol{x}_{i'}) \geq w]\right\} dw$$

then the observation above shows that

$$\mathsf{FD}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i) = \max_{\mathbf{0} \leq \boldsymbol{c} \leq \boldsymbol{x}_i} \langle \boldsymbol{q}_i(\boldsymbol{x}_i), \boldsymbol{c} \rangle + \mathsf{FD}_{\mathbf{1}-\boldsymbol{c}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}).$$

We now claim that, for the greedy choice of $\boldsymbol{x}_i$, we include the fraction $\boldsymbol{x}_i$ of $\boldsymbol{q}_i$ in the new free-disposal value while truncating the previous partial allocation to weight $\mathbf{1} - \boldsymbol{x}_i$. This claim follows intuitively from the supporting-price property: if the objective used only a fraction $\boldsymbol{c} \neq \boldsymbol{x}_i$ of $\boldsymbol{q}_i$, then choosing $\boldsymbol{c}$ instead of $\boldsymbol{x}_i$ would yield at least as much value because $\boldsymbol{c} \leq \boldsymbol{x}_i$. We formalize this claim as follows.

Claim 4.9. *Define the function $\Phi_i : [0,1]^m \to \mathbb{R}_{\geq 0}$ by*

$$\Phi_i(\boldsymbol{x}) := v_i(\boldsymbol{x}) + \mathsf{FD}_{\mathbf{1}-\boldsymbol{x}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}).$$

If v_i is continuous, then Φ_i attains its maximum on $[0,1]^m$. Then, any maximizer of Φ_i also maximizes the increment in the free-disposal value.

Proof. For fixed previous allocations, $\boldsymbol{c} \mapsto \mathsf{FD}_{\boldsymbol{c}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1})$ is continuous and piecewise linear, since it is a sum of finite fractional-knapsack values as functions of their capacities. Hence, if v_i is continuous, so is Φ_i; compactness of $[0,1]^m$ then guarantees a maximizer.

First, note that for all $\boldsymbol{x}$,

$$\begin{aligned}\mathsf{FD}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \boldsymbol{x}) &= \max_{\mathbf{0} \le \boldsymbol{c} \le \boldsymbol{x}} \langle \boldsymbol{q}_i(\boldsymbol{x}), \boldsymbol{c} \rangle + \mathsf{FD}_{\mathbf{1}-\boldsymbol{c}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}) \\ &\ge \langle \boldsymbol{q}_i(\boldsymbol{x}), \boldsymbol{x} \rangle + \mathsf{FD}_{\mathbf{1}-\boldsymbol{x}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}) \\ &= v_i(\boldsymbol{x}) + \mathsf{FD}_{\mathbf{1}-\boldsymbol{x}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}) = \Phi_i(\boldsymbol{x}).\end{aligned}$$

Let $\widehat{\boldsymbol{x}}_i$ be a maximizer of Φ_i. By the pointwise bound above,

$$\Phi_i(\widehat{\boldsymbol{x}}_i) \le \mathsf{FD}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \widehat{\boldsymbol{x}}_i). \tag{2}$$

Now fix an arbitrary $\boldsymbol{x}'_i \in [0,1]^m$, and let $\boldsymbol{c}$ attain the maximum in the decomposition of $\mathsf{FD}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \boldsymbol{x}'_i)$. Such a $\boldsymbol{c}$ exists because this is a continuous fractional-knapsack objective on the compact box $\mathbf{0} \le \boldsymbol{c} \le \boldsymbol{x}'_i$. Then

$$\begin{aligned}&\mathsf{FD}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \boldsymbol{x}'_i) \\ &\quad = \langle \boldsymbol{q}_i(\boldsymbol{x}'_i), \boldsymbol{c} \rangle + \mathsf{FD}_{\mathbf{1}-\boldsymbol{c}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}) \\ &\quad \le v_i(\boldsymbol{c}) + \mathsf{FD}_{\mathbf{1}-\boldsymbol{c}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}) = \Phi_i(\boldsymbol{c}) \le \Phi_i(\widehat{\boldsymbol{x}}_i),\end{aligned} \tag{3}$$

where the first inequality is the supporting-price property for $\boldsymbol{c} \le \boldsymbol{x}'_i$, and the last inequality uses the maximality of $\widehat{\boldsymbol{x}}_i$. Combining Equations (2) and (3) gives $\mathsf{FD}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \boldsymbol{x}'_i) \le \mathsf{FD}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \widehat{\boldsymbol{x}}_i)$ for every $\boldsymbol{x}'_i$. Thus, $\widehat{\boldsymbol{x}}_i$ is a valid greedy choice. □

We are finally ready to prove Theorem 4.6.

Proof of Theorem 4.6. Map any buyer arrival order to the corresponding arrival order of the parts P_i. The outputs of Algorithm 5 and the greedy algorithm of Theorem 4.3 are then the same. Indeed, when buyer i arrives, Algorithm 5 chooses the vector $\boldsymbol{x}_i$ that increases the free-disposal value the most among all options in $[0,1]^m$. This vector is also the greedy choice among all options in O_i and is chosen by the partition-matroid greedy algorithm when ties are resolved to match Algorithm 5. Thus, the expected free-disposal value of Algorithm 5 is exactly the expected value of greedy on f. Furthermore, since $\boldsymbol{x}^*_i \in O_i$ for all i, the optimal value of f is at least the hindsight optimum $\mathsf{Opt}(v_1, \ldots, v_n)$. The competitive and smooth competitive guarantees of Algorithm 5 then follow from Theorem 4.3.

For the smooth guarantee, independently zeroing buyers is equivalent to omitting their corresponding parts by reasoning similar to that in the proof of Theorem 4.1. The retained buyers appear in a uniformly random relative order. Moreover, every choice made in such a pruned execution belongs to O_i, since its retained-buyer prefix can be realized in a full order by placing the omitted buyers afterward. Therefore, Lemma D.1 gives $\Gamma(q) = q - q^2/2$.

It remains to establish the runtime guarantee under the additional assumption that all the v_i are concave. By Claim 4.9, it is sufficient to find a maximizer of Φ_i. Furthermore, observe that $\mathsf{FD}_{\boldsymbol{c}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1})$ is concave in $\boldsymbol{c}$, so $\Phi_i(\boldsymbol{x})$ is also concave. Finally, for a fixed j, the map

$$w \mapsto \min \left\{ c_j, \sum_{i'=1}^{i-1} x_{i',j} \cdot \mathbf{1}[q_{i',j}(\boldsymbol{x}_{i'}) \ge w] \right\}$$

is a step function with at most n breakpoints. Therefore, assuming value and first-order (supergradient) oracles for v, we can also evaluate Φ_i (and its supergradients). Let $M_i := \max_{i' \le i} v_{i'}(\mathbf{1}) \le \mathsf{Opt}(\boldsymbol{v})$. If $M_i = 0$, then $\Phi_i \equiv 0$ by monotonicity and $\boldsymbol{x}_i = \mathbf{0}$ is an exact maximizer; otherwise, $0 \le \Phi_i(\boldsymbol{x}) \le nM_i$ on $[0,1]^m$, and we apply the ellipsoid method [GLS88] to maximize Φ_i within additive accuracy $\delta_i := \frac{\epsilon}{Cn^2} M_i$ in $\mathrm{poly}(n, m, \log(1/\epsilon))$ time, where $C > 0$ is a sufficiently large universal constant. By Equations (2) and (3), this yields an additive δ_i-approximate greedy step on f with $\delta_i \le \frac{\epsilon}{Cn^2}\mathsf{Opt}(\boldsymbol{v})$, which by Lemma D.1 degrades the competitive and smooth-competitive guarantees by at most $O(n \max_i \delta_i) \le \epsilon \cdot \mathsf{Opt}(\boldsymbol{v})$. □

4.2.2 A $(1 - 1/e)$-Competitive Algorithm

Let us consider the following primal LP for the hindsight optimum. Here, the optimum allocates to buyer i a measure μ_i over the possible allocation vectors $\boldsymbol{x} \in [0,1]^m$.

$$\begin{array}{lll}
\text{maximize} & \displaystyle\sum_{i=1}^{n} \int_{\boldsymbol{x}\in[0,1]^m} v_i(\boldsymbol{x})\, d\mu_{i,\boldsymbol{x}} & \\
\text{s.t.} & \displaystyle\int_{\boldsymbol{x}\in[0,1]^m} d\mu_{i,\boldsymbol{x}} \le 1 & \forall i \in [n] \\
& \displaystyle\sum_{i=1}^{n} \int_{\boldsymbol{x}\in[0,1]^m} x_j\, d\mu_{i,\boldsymbol{x}} \le 1 & \forall j \in [m] \\
& d\mu_{i,\boldsymbol{x}} \ge 0 &
\end{array}$$

Its dual is

$$\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{i=1}^{n} \beta_i + \sum_{j=1}^{m} \alpha_j & \\
\text{s.t.} & \beta_i + \langle \boldsymbol{x}, \boldsymbol{\alpha} \rangle \ge v_i(\boldsymbol{x}) & \forall \boldsymbol{x} \in [0,1]^m, i \in [n] \\
& \beta_i, \alpha_j \ge 0 &
\end{array}$$

We will use the online primal-dual framework to design and analyze our algorithms. To facilitate the analysis, we consider the following modification to the primal and dual variables.

- For each item j and each weight level w, we define a split-dual variable $\alpha_j(w)$, and we maintain the invariant throughout the algorithm that

$$\alpha_j = \int_0^\infty \alpha_j(w)\, dw.$$

  Therefore, we will only specify the updates to the split-dual variables $\alpha_j(w)$.

We provide Algorithm 6 below, for the function $f(x) = \frac{e^{x-1} - 1/e}{\gamma}$ and $\gamma = 1 - 1/e$. It is straightforward to verify that f is increasing and continuous, that $f(0) = 0$, $f(1) = 1$, and $f'(x) + 1 - f(x) = 1/\gamma$ for all $x \in [0,1]$. We want to prove that the algorithm is γ-competitive. Following the online primal-dual analysis framework, we shall prove the following two claims.

- **Approximate Reverse Weak Duality.** At every iteration, the primal increment is at least γ times the dual increment.

Algorithm 6 Free-Disposal Algorithm for Divisible Items

Input: Arriving buyers with valuations $\boldsymbol{v}$.
Output: Allocation vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ that need not be feasible.

1: Initialize $y_j^{(0)}(w) \leftarrow 0$ and $\alpha_j^{(0)}(w) \leftarrow f(y_j^{(0)}(w)) = 0$ for all $j \in [m]$ and $w \geq 0$.
2: **for** buyer $i \in [n]$ **do**
3: Carry forward $y_j^{(i)}(w) \leftarrow y_j^{(i-1)}(w)$ and $\alpha_j^{(i)}(w) \leftarrow \alpha_j^{(i-1)}(w)$ for all j, w.
4: Let $\boldsymbol{x}_i \in [0,1]^m$ be the vector that satisfies Condition 1.
5: **for** item $j \in [m]$ and all $w \in [0, q_{i,j}(\boldsymbol{x}_i)]$ **do**
6: $y_j^{(i)}(w) \leftarrow \min\{1, y_j^{(i-1)}(w) + x_{i,j}\}$.
7: $\alpha_j^{(i)}(w) \leftarrow f(y_j^{(i)}(w))$.
8: **end for**
9: $\beta_i \leftarrow v_i(\boldsymbol{x}_i) - \langle \boldsymbol{x}_i, \boldsymbol{\alpha}^{(i)} \rangle$.
10: Allocate $\boldsymbol{x}_i$ to buyer i.
11: **end for**

Condition 1 (Fixed-point condition)**.** For any $\boldsymbol{x} \in [0,1]^m$, let $p_j(\boldsymbol{x})$ be the hypothetical resulting value of $\alpha_j^{(i)} = \int_0^\infty \alpha_j^{(i)}(w)\, dw$ had we chosen $\boldsymbol{x}$ at Line 4. Then, we require

$$\boldsymbol{x} \in \underbrace{\arg\max_{\boldsymbol{x}' \in [0,1]^m} v_i(\boldsymbol{x}') - \langle \boldsymbol{x}', \boldsymbol{p}(\boldsymbol{x}) \rangle}_{\varphi(\boldsymbol{x})}.$$

- **Dual Feasibility.** At the end, the dual program is feasible.

Before proving that Condition 1 always admits a feasible fixed point, we first prove that the algorithm satisfies both approximate reverse weak duality and dual feasibility, assuming that $\boldsymbol{x}_i$ is well-defined.

Lemma 4.10. *At every iteration, the primal increment is at least γ times the dual increment.*

Proof. At iteration i, the primal increase is exactly

$$\sum_{j=1}^m \int_0^\infty \left(y_j^{(i)}(w) - y_j^{(i-1)}(w) \right) dw = \sum_{j=1}^m \int_0^\infty \int_{y_j^{(i-1)}(w)}^{y_j^{(i)}(w)} 1 \, dt \, dw.$$

On the dual side, since we maintain $\alpha_j^{(i)}(w) = f(y_j^{(i)}(w))$ throughout, the increase in $\boldsymbol{\alpha}$ is exactly

$$\begin{aligned}
&\sum_{j=1}^m \int_0^\infty \left(\alpha_j^{(i)}(w) - \alpha_j^{(i-1)}(w) \right) dw \\
&= \sum_{j=1}^m \int_0^\infty \left(f(y_j^{(i)}(w)) - f(y_j^{(i-1)}(w)) \right) dw \\
&= \sum_{j=1}^m \int_0^\infty \int_{y_j^{(i-1)}(w)}^{y_j^{(i)}(w)} f'(t) \, dt \, dw
\end{aligned}$$

while the increase in $\boldsymbol{\beta}$ is

$$\begin{aligned}
\beta_i &= v_i(\boldsymbol{x}_i) - \langle \boldsymbol{x}_i, \boldsymbol{\alpha}^{(i)} \rangle \\
&= \sum_{j=1}^{m} x_{i,j} \left(q_{i,j}(\boldsymbol{x}_i) - \int_0^{\infty} f(y_j^{(i)}(w))\, dw \right) \\
&\le \sum_{j=1}^{m} \int_0^{q_{i,j}(\boldsymbol{x}_i)} x_{i,j}(1 - f(y_j^{(i)}(w)))\, dw \\
&\le \sum_{j=1}^{m} \int_0^{q_{i,j}(\boldsymbol{x}_i)} \int_{y_j^{(i-1)}(w)}^{y_j^{(i)}(w)} (1 - f(t))\, dt\, dw \\
&\le \sum_{j=1}^{m} \int_0^{\infty} \int_{y_j^{(i-1)}(w)}^{y_j^{(i)}(w)} (1 - f(t))\, dt\, dw
\end{aligned}$$

where for the second inequality, if $y_j^{(i)}(w) = 1$, then $1 - f(y_j^{(i)}(w)) = 0$; otherwise, if $y_j^{(i)}(w) < 1$, then we have $y_j^{(i)}(w) - y_j^{(i-1)}(w) = x_{i,j}$, and $1 - f(t)$ is a decreasing function of t.

Finally, since for all $t \in [0,1]$ we have $1 - f(t) + f'(t) = 1/\gamma$ by construction, the lemma follows. □

Lemma 4.11. *At the end, we have $\beta_i + \langle \boldsymbol{x}, \boldsymbol{\alpha}^{(n)} \rangle \ge v_i(\boldsymbol{x})$ for all $\boldsymbol{x} \in [0,1]^m$ and $i \in [n]$.*

Proof. Since f is an increasing function and $y_j^{(i)}(w)$ is non-decreasing in i, $\boldsymbol{\alpha}^{(i)}$ is also non-decreasing in i. Therefore, it suffices to prove that $\beta_i + \langle \boldsymbol{x}, \boldsymbol{\alpha}^{(i)} \rangle \ge v_i(\boldsymbol{x})$ for all $\boldsymbol{x} \in [0,1]^m$. This is immediately implied by Condition 1, since

$$\beta_i = v_i(\boldsymbol{x}_i) - \langle \boldsymbol{x}_i, \boldsymbol{\alpha}^{(i)} \rangle = \max_{\boldsymbol{x}' \in [0,1]^m} v_i(\boldsymbol{x}') - \langle \boldsymbol{x}', \boldsymbol{\alpha}^{(i)} \rangle \ge v_i(\boldsymbol{x}) - \langle \boldsymbol{x}, \boldsymbol{\alpha}^{(i)} \rangle. \qquad \square$$

We now prove the fixed-point condition, which implies the algorithm is well-defined. The proof uses the assumption that the valuations are DR-submodular (in particular, the supporting-price map is continuous) and concave.

Lemma 4.12. *For any valid state of the algorithm before Line 4, there exists an $\boldsymbol{x} \in [0,1]^m$ that satisfies Condition 1.*

Proof. We prove this via the Kakutani fixed-point theorem applied to φ. To this end, we first show that the map $\boldsymbol{x} \mapsto \boldsymbol{p}(\boldsymbol{x})$ is continuous. By definition of the algorithm, this map is given coordinatewise by

$$p_j(\boldsymbol{x}) = \alpha_j^{(i-1)} + \int_0^{q_{i,j}(\boldsymbol{x})} f\left(\min\{1, y_j^{(i-1)}(w) + x_j\}\right) - f(y_j^{(i-1)}(w))\, dw.$$

Since v_i is DR-submodular, Proposition 2.2 implies that $q_{i,j}(\boldsymbol{x})$ is continuous; it is bounded by continuity on the compact domain $[0,1]^m$. Therefore, the map $\boldsymbol{x} \mapsto \boldsymbol{p}(\boldsymbol{x})$ is continuous, as is the map $(\boldsymbol{x}, \boldsymbol{x}') \mapsto v_i(\boldsymbol{x}') - \langle \boldsymbol{x}', \boldsymbol{p}(\boldsymbol{x}) \rangle$.

Next, note that

- The domain $[0,1]^m$ of $\boldsymbol{x}$ is a nonempty, compact, and convex subset of $\mathbb{R}^m$,
- $\varphi(\boldsymbol{x})$ is convex and nonempty, as the function $v_i(\boldsymbol{x}') - \langle \boldsymbol{x}', \boldsymbol{p}(\boldsymbol{x}) \rangle$ is finite and concave in $\boldsymbol{x}'$ due to the finiteness and concavity of v_i, so the maximizers exist and form a convex set, and

- $\varphi(\boldsymbol{x})$ has a closed graph due to the continuity of the map $(\boldsymbol{x}, \boldsymbol{x}') \mapsto v_i(\boldsymbol{x}') - \langle \boldsymbol{x}', \boldsymbol{p}(\boldsymbol{x}) \rangle$.

Therefore, by the Kakutani fixed-point theorem, φ has a fixed point. □

Proof of Theorem 4.7. This follows directly from Lemmas 4.10 to 4.12. □

5 Single-Sample Single-Item Prophet Inequalities

In this section, we provide SSPI algorithms for both the divisible- and indivisible-item cases, each admitting the optimal competitive ratio of 1/2. We also consider the problem of the *single-sample prophet inequality for a concave divisible item*, which is an important generalization of the single-sample prophet inequality for fractional knapsack.

5.1 Single-Sample Prophet Inequality for an Indivisible Item

For the indivisible-item case, the algorithm for the single-sample prophet inequality is a classical result of [RWW20]. For completeness, we include a modification of their algorithm for our variant. Recall that a buyer is characterized by a pair (q_i, x_i), where $q_i \in \mathbb{R}_{\geq 0}$ is the buyer's value for the item and $x_i \in \{0, 1\}$ indicates whether the buyer arrives.

Algorithm 7 Single-Sample Prophet Inequality for an Indivisible Item [RWW20]

Input: Sample valuations $\boldsymbol{s}$ and arriving buyers with real valuations $\boldsymbol{r}$.
Output: Feasible allocations $x_1, \ldots, x_n$.
1: Let $\tau \leftarrow \max_{x(s_i)=1} q(s_i)$.
2: Accept the first real buyer i for whom $x(r_i) = 1$ and $q(r_i) > \tau$.

We state the following result without proof; it is a simple modification of the corresponding result proved in [RWW20].

Theorem 5.1 ([RWW20])**.** *Algorithm 7 is a* 1/2*-competitive polynomial-time* SSPI *for the indivisible-item case. Furthermore, this is the best competitive ratio achievable by any algorithm.*

5.2 Single-Sample Prophet Inequality for Fractional Knapsack

For the divisible-item case, we consider the *single-sample prophet inequality for fractional knapsack*. Recall that each buyer i is represented by a knapsack item (q_i, x_i), where q_i is the *density* of the item and x_i is its weight. Upon the arrival of item i, an algorithm must immediately choose a fraction $x_i' \leq x_i$ of the item while ensuring that the capacity-1 knapsack is not overfilled: $\sum_{i=1}^n x_i' \leq 1$. Finally, the algorithm's reward is $\sum_{i=1}^n x_i' q_i$. The benchmark is the offline optimal fractional knapsack, and it is well known that the greedy algorithm in decreasing order of density achieves this benchmark. For ease of notation, for a knapsack item u, let $q(u)$ and $x(u)$ be the density and weight of the item, respectively.

We prove that, in the single-sample setting, Algorithm 8 achieves a competitive ratio of 1/2 with respect to the hindsight optimum. This result has already been proved in [NRS26], but for the model in which s_i and r_i are drawn i.i.d. from a distribution $\mathcal{D}_i$. Their proof does not apply in the Game of Googol setting, so we prove the theorem from scratch.

Theorem 5.2. *Algorithm 8 is a* 1/2*-competitive polynomial-time* SSPI *for the divisible-item case. Furthermore, this is the best competitive ratio achievable by any algorithm.*

Algorithm 8 Single-Sample Prophet Inequality for Fractional Knapsack [NRS26]

Input: Sample knapsack items $\boldsymbol{s}$ and arriving knapsack items $\boldsymbol{r}$.
Output: Feasible knapsack allocations $x'_1, \ldots, x'_n$.

Compute the optimal fractional knapsack of $\boldsymbol{s}$; let the resulting knapsack be K_s.
Let τ_s be the density of a uniformly random quantile in K_s.
Let $R \leftarrow 1$.
for item $i \in [n]$ **do**
    if $q(r_i) > \tau_s$ **then**
        Accept the maximum feasible fraction of r_i, i.e., $x'_i \leftarrow \min\{R, x(r_i)\}$.
        $R \leftarrow R - x'_i$.
    end if
end for
Let K_r be the optimal fractional knapsack of $\boldsymbol{r}$, and τ_r be the density of a uniformly random quantile in K_r. ▷ *Only for analysis.*

We will not prove the optimality statement, as it follows from a standard argument for the single-item prophet inequality. For the rest of this section, we assume that $\sum_{i=1}^n x(s_i) \geq 1$ and $\sum_{i=1}^n x(r_i) \geq 1$ always hold. This assumption is without loss of generality, as one can insert a dummy item of weight 1 and density 0 at the end of both the sample and real profiles without affecting the algorithm. We further assume that all items in $\boldsymbol{s}$ and $\boldsymbol{r}$ have different densities. Otherwise, the algorithm and analysis still apply under a suitable tie-breaking rule, such as an infinitesimal random perturbation of the densities.

For any $t \geq 0$, let $K_r(t)$ be the fraction of the prophet knapsack for $\boldsymbol{r}$ whose density is larger than t, and let $A(t)$ be the fraction of the algorithm's output whose density is larger than t. We define $K_s(t)$ analogously for the optimal fractional knapsack of $\boldsymbol{s}$. If we can prove that $\mathbb{E}[A(t)] \geq \frac{1}{2} \cdot \mathbb{E}[K_r(t)]$ for almost all $t \geq 0$, then we are done, as

$$\mathsf{Alg} = \int_0^\infty \mathbb{E}[A(t)]\, dt \geq \frac{1}{2} \int_0^\infty \mathbb{E}[K_r(t)]\, dt = \frac{1}{2} \cdot \mathsf{Opt}.$$

Fix any $t \geq 0$. By the definition of τ_r and the greedy algorithm that computes the optimal fractional knapsack, we have the following observation.

Observation 5.3. *For any realization of $\boldsymbol{r}$, we have $K_r(t) = \Pr[\tau_r > t \mid \boldsymbol{r}]$. Furthermore, this is exactly $\min\{H, 1\}$, where $H := \sum_{i=1}^n x(r_i)\mathbf{1}[q(r_i) > t]$ is the total weight of the items in $\boldsymbol{r}$ whose density is above t.*

By symmetry, we now know that $\mathbb{E}[K_r(t)] = \Pr[\tau_s > t]$. We first decompose $\mathbb{E}[A(t)]$ into two cases.

Case $\tau_s \geq t$. Here, every portion of the knapsack accepted by the algorithm has density above $\tau_s \geq t$, so

$$\mathbb{E}[A(t) \cdot \mathbf{1}[\tau_s \geq t]] = \mathbb{E}[K_r(\tau_s) \cdot \mathbf{1}[\tau_s \geq t]].$$

Case $\tau_s < t$. The danger here is that the algorithm may accept too many fractions with medium density, thereby blocking items with high density. Therefore, if we define

$$M := \sum_{i=1}^{n} x(r_i) \cdot \mathbf{1}[\tau_s < q(r_i) \le t], \qquad H := \sum_{i=1}^{n} x(r_i) \cdot \mathbf{1}[q(r_i) > t]$$

then the fraction of items with density above t accepted in this case is at least

$$\begin{aligned} \min\{H, (1-M)^+\} &= (1-M)^+ - (1-M-H)^+ \\ &= (1-M)^+ - (1-\min\{M+H, 1\}) \\ &= (1-M)^+ - (1-K_r(\tau_s)) \end{aligned}$$

where the third equality follows from Observation 5.3.

Combining the two cases, we have

$$\begin{aligned} \mathbb{E}[A(t)] \ge \mathbb{E}[K_r(\tau_s) \cdot \mathbf{1}[\tau_s \ge t]] & \quad (4) \\ - \mathbb{E}[(1-K_r(\tau_s)) \cdot \mathbf{1}[\tau_s < t]] & \quad (5) \\ + \mathbb{E}[(1-M)^+ \cdot \mathbf{1}[\tau_s < t]]. & \quad (6) \end{aligned}$$

We will now bound each term individually.

Lemma 5.4. *Term (4) satisfies*

$$2\mathbb{E}[K_r(\tau_s) \cdot \mathbf{1}[\tau_s \ge t]] = \Pr[\tau_s \ge t, \tau_r \ge t].$$

Proof. By symmetry,

$$\begin{aligned} 2\mathbb{E}[K_r(\tau_s) \cdot \mathbf{1}[\tau_s \ge t]] &= \mathbb{E}[K_r(\tau_s) \cdot \mathbf{1}[\tau_s \ge t]] + \mathbb{E}[K_s(\tau_r) \cdot \mathbf{1}[\tau_r \ge t]] \\ &= \Pr[\tau_r > \tau_s, \tau_s \ge t] + \Pr[\tau_s > \tau_r, \tau_r \ge t] \\ &= \Pr[\tau_s \ge t, \tau_r \ge t] \end{aligned}$$

where the last equality uses the fact that $\tau_s = \tau_r$ can never occur because $\boldsymbol{s}$ and $\boldsymbol{r}$ have different densities. □

The following lemma is proved similarly.

Lemma 5.5. *Term (5) satisfies*

$$2\mathbb{E}[(1-K_r(\tau_s)) \cdot \mathbf{1}[\tau_s < t]] = \Pr[\tau_s < t, \tau_r < t].$$

Finally, we have the following.

Lemma 5.6. *If t does not coincide with any density, then term (6) satisfies*

$$2\mathbb{E}[(1-M)^+ \cdot \mathbf{1}[\tau_s < t]] \ge \Pr[\tau_s < t],$$

where $M = \sum_{i=1}^{n} x(r_i) \cdot \mathbf{1}[\tau_s < q(r_i) \le t]$.

Proof. Let the *truncated instances* $\overline{\boldsymbol{s}}$ and $\overline{\boldsymbol{r}}$ be obtained by removing all items with density above t from $\boldsymbol{s}$ and $\boldsymbol{r}$, respectively. Let $\overline{K_s}, \overline{K_r}$ be the optimal knapsacks of the truncated instances, and let $\overline{\tau_s}$ and $\overline{\tau_r}$ be the random quantiles from these knapsacks. Whenever several random quantiles are used below, draw them independently conditional on the sample-real orientations.

Let Z denote the vector of independent orientation coins that determine $\boldsymbol{s}$ and $\boldsymbol{r}$. Conditional on $\tau_s < t$ and Z, the quantile τ_s is drawn from a top-density prefix of $\overline{K_s}$ and hence stochastically dominates $\overline{\tau_s}$. Since $\overline{\tau_r}$ is drawn independently of these two quantiles conditional on Z, for every Z,

$$\begin{aligned}\Pr[\overline{\tau_r} < \tau_s,\ \tau_s < t \mid Z] &= \Pr[\tau_s < t \mid Z] \Pr[\overline{\tau_r} < \tau_s \mid \tau_s < t, Z] \\ &\geq \underbrace{\Pr[\tau_s < t \mid Z]}_{f(Z)} \underbrace{\Pr[\overline{\tau_r} < \overline{\tau_s} \mid Z]}_{g(Z)}.\end{aligned}$$

We next apply the FKG inequality only to buyers whose two densities straddle t. Let S be their orientation coins, with 1 meaning that the lower-density item is the sample. Let L be the coins of the "low" buyers whose two densities are below t, and let H be the coins of the "high" buyers whose two densities are above t. Then $f(Z) = f(S, H)$ and $g(Z) = g(S, L)$. Indeed, because the sample knapsack is filled to capacity after adding the dummy item, $\Pr[\tau_s < t \mid Z]$ is exactly the residual capacity left after packing the sample items of density above t. The orientation of a low buyer can change which below-t item is in the sample, but cannot change this residual capacity, so f is independent of L. On the other hand, both possible items of a high buyer are removed from the sample and real instances by truncation, regardless of their orientation. Thus, g is independent of H.

Define $F(S) := \mathbb{E}_H[f(S, H)]$ and $G(S) := \mathbb{E}_L[g(S, L)]$. Since L and H are independent conditional on S, we have $\mathbb{E}[f(Z)g(Z) \mid S] = F(S)G(S)$, so $\mathbb{E}[f(Z)g(Z)] = \mathbb{E}[F(S)G(S)]$.

Both F and G are non-decreasing. Indeed, changing a coordinate of S from 0 to 1 removes an above-t item from the sample, so f increases. After truncation, the same change adds the other item to the sample and removes it from the real instance, so $\overline{\tau_s}$ increases and $\overline{\tau_r}$ decreases in stochastic order; hence, g also increases. The FKG inequality therefore gives

$$\begin{aligned}\mathbb{E}[f(Z)g(Z)] &= \mathbb{E}[F(S)G(S)] \\ &\geq \mathbb{E}[F(S)]\mathbb{E}[G(S)] = \Pr[\tau_s < t] \Pr[\overline{\tau_r} < \overline{\tau_s}] \\ &= \frac{1}{2} \Pr[\tau_s < t],\end{aligned}$$

where the last equality follows from symmetry and the absence of tied densities in $\overline{\boldsymbol{s}}$ and $\overline{\boldsymbol{r}}$, which imply that $\Pr[\overline{\tau_r} < \overline{\tau_s}] = 1/2$.

Finally, $\min\{M, 1\} = \overline{K_r}(\tau_s)$, and thus

$$\mathbb{E}[(1 - M)^+ \mathbf{1}[\tau_s < t]] = \Pr[\overline{\tau_r} < \tau_s,\ \tau_s < t] \geq \mathbb{E}[f(Z)g(Z)] \geq \frac{1}{2} \Pr[\tau_s < t]. \qquad \square$$

We are ready to prove the final theorem.

Proof of Theorem 5.2. Combining Lemmas 5.4 to 5.6, we have

$$\begin{aligned}2\mathbb{E}[A(t)] &\geq 2\left(\mathbb{E}[K_r(\tau_s) \cdot \mathbf{1}[\tau_s \geq t]] - \mathbb{E}[(1 - K_r(\tau_s)) \cdot \mathbf{1}[\tau_s < t]] + \mathbb{E}[(1 - M)^+ \cdot \mathbf{1}[\tau_s < t]]\right) \\ &\geq \Pr[\tau_s \geq t, \tau_r \geq t] - \Pr[\tau_s < t, \tau_r < t] + \Pr[\tau_s < t] \\ &= \Pr[\tau_r \geq t] \geq \Pr[\tau_r > t] = \mathbb{E}[K_r(t)],\end{aligned}$$

for almost all $t \geq 0$, and therefore the theorem follows by integrating the bound above. $\square$

5.3 Single-Sample Prophet Inequality for a Concave Divisible Item

Finally, we consider the single-sample prophet inequality for a concave divisible item. The setting is similar to the fractional-knapsack setting, with the generalization that every valuation function $v : [0,1] \to \mathbb{R}_{\geq 0}$ is concave. Indeed, a fractional-knapsack item (q, x) can be represented by the concave valuation function $f(x') := q \min\{x, x'\}$.

A natural starting point would be to generalize Algorithm 8 as follows: from the samples $\boldsymbol{s}$, build a function $f_s : [0,1] \to \mathbb{R}_{\geq 0}$ where

$$f_s(t) := \max_{\substack{x_1,\ldots,x_n \geq 0 \\ \sum_{i=1}^n x_i = t}} \sum_{i=1}^n s_i(x_i).$$

This function is concave in t. Then, take $\tau_s = f_s'(u)$, where u is drawn uniformly from $[0,1]$. Finally, for every arriving real valuation r_i, accept the maximum feasible x_i such that $r_i'(x_i) > \tau_s$. However, we show that this approach is insufficient to obtain a competitive ratio of $1/2$ for this problem.

Lemma 5.7. *The natural generalization of Algorithm 8 has a competitive ratio of at most* $7/16 = 0.4375 < 1/2$ *for concave divisible items.*

We defer the proof of this lemma to Appendix E. We next provide a simple algorithm that achieves a competitive ratio of $\frac{1-1/e}{2} \approx 0.3161$. This leaves a gap between the achieved ratio and the information-theoretic upper bound of $1/2$. Intuitively, for some large $M > 0$, the algorithm breaks each valuation function into marginal gains of the form $v(k/M) - v((k-1)/M)$ and then runs M instances of the single-sample indivisible-item prophet inequality, one for each "slot" of the marginal gains. However, since the marginal gains of each valuation function are correlated and may affect the aggregate performance of the M instances, the algorithm decouples this correlation by randomly permuting the marginal gains among the M slots. This idea is formally described in Algorithm 9.

Algorithm 9 Single-Sample Prophet Inequality for a Concave Divisible Item

Input: Sample valuations $\boldsymbol{s}$, arriving buyers with valuations $\boldsymbol{r}$, and a large integer parameter $M \in \mathbb{Z}_{>0}$.

Output: Feasible allocations $x_1, \ldots, x_n$.

Initialize M instances of Algorithm 7 as $\mathsf{SSPI}_1, \ldots, \mathsf{SSPI}_M$.
for buyer $i \in [n]$ **do**
  Let $\boldsymbol{S}^{(i)}$ be a length-M vector with $S_k^{(i)} \leftarrow s_i(k/M) - s_i((k-1)/M)$ for $k \in [M]$.
  Independently and uniformly randomly permute $\boldsymbol{S}^{(i)}$.
  Send $(S_k^{(i)}, 1)$ to SSPI_k for all $k \in [M]$ as the sample for buyer i.
end for
for buyer $i \in [n]$ **do**
  Let $\boldsymbol{R}^{(i)}$ be a length-M vector with $R_k^{(i)} \leftarrow r_i(k/M) - r_i((k-1)/M)$ for $k \in [M]$.
  Independently and uniformly randomly permute $\boldsymbol{R}^{(i)}$.
  Send $(R_k^{(i)}, 1)$ to SSPI_k for all $k \in [M]$ as the real value of buyer i, and receive allocation $x_k^{(i)} \in \{0,1\}$.
  Allocate $x_i \leftarrow \frac{1}{M} \sum_{k=1}^M x_k^{(i)}$ to buyer i.
end for

Theorem 5.8. *As $M \to \infty$, the competitive ratio of Algorithm 9 approaches $\frac{1-1/e}{2} \approx 0.3161$ for the concave divisible-item case.*

Proof. Fix $\boldsymbol{r}$ and write $\Delta_{i,k} = r_i(k/M) - r_i((k-1)/M)$ for the k-th marginal piece of buyer i. By concavity, $\Delta_{i,1} \geq \cdots \geq \Delta_{i,M} \geq 0$. If buyer i wins $m_i := \sum_{k=1}^{M} x_k^{(i)}$ of the M SSPI instances, then the values won by these instances are some m_i entries of this sequence, whereas the algorithm obtains $r_i(m_i/M) = \sum_{j=1}^{m_i} \Delta_{i,j}$. Thus, the algorithm's actual reward is at least the sum of the rewards obtained by the M SSPI instances. Moreover, each instance allocates at most one piece, so $\sum_{i=1}^{n} m_i \leq M$, and the resulting allocation is feasible.

For any fixed instance k, the sample and real marginals presented by a buyer are exchangeable: each is a uniformly random marginal from the corresponding valuation, and the fair coin that swaps the two underlying valuations makes the joint distribution invariant under swapping the sample and real marginals. Hence, this is a mixture of valid Game of Googol instances, so Algorithm 7 obtains at least half of $\mathbb{E}\left[\max_i R_k^{(i)}\right]$ by Theorem 5.1. It remains only to compare the sum of these per-instance maxima with the divisible optimum.

Let $N_i(w) = |\{j : \Delta_{i,j} \geq w\}|$, and let $\mathsf{Opt}_M(\boldsymbol{r})$ be the optimum restricted to allocations in multiples of $1/M$. Equivalently, $\mathsf{Opt}_M(\boldsymbol{r})$ is the sum of the largest M numbers among all the $\Delta_{i,j}$: since the marginals of each buyer are decreasing, taking a marginal makes all earlier marginals at least as attractive. In a fixed instance, buyer i contributes a marginal of value at least w with probability $N_i(w)/M$, independently across buyers; intuitively, the only loss is that several valuable marginals may land in the same instance. Therefore, using $1 - z \leq e^{-z}$ and $1 - e^{-z} \geq (1 - 1/e)\min\{1, z\}$,

$$\begin{aligned}
\mathbb{E}\left[\sum_{k=1}^{M} \max_i R_k^{(i)} \,\middle|\, \boldsymbol{r}\right] &= M \int_0^\infty \left(1 - \prod_{i=1}^{n}\left(1 - \frac{N_i(w)}{M}\right)\right) dw \\
&\geq (1 - 1/e) \int_0^\infty \min\left\{M, \sum_{i=1}^{n} N_i(w)\right\} dw \\
&= (1 - 1/e)\mathsf{Opt}_M(\boldsymbol{r}).
\end{aligned}$$

Combining the two losses and then averaging over $\boldsymbol{r}$ gives $\mathsf{Alg}_M \geq \frac{1-1/e}{2} \cdot \mathbb{E}[\mathsf{Opt}_M(\boldsymbol{r})]$. Finally, as $M \to \infty$, $\mathsf{Opt}_M(\boldsymbol{r}) \to \mathsf{Opt}(\boldsymbol{r})$, proving the asymptotic ratio. □

For completeness, the following lemma makes the convergence $\mathsf{Opt}_M(\boldsymbol{r}) \to \mathsf{Opt}(\boldsymbol{r})$ quantitative. Its proof is deferred to Appendix E.

Lemma 5.9. *For any fixed integer $M \geq 1$, we have*

$$\mathsf{Opt}_M(\boldsymbol{r}) \geq \frac{M}{M+n} \cdot \mathsf{Opt}(\boldsymbol{r}).$$

Therefore, for fixed n, as $M \to \infty$, we have $\mathsf{Opt}_M(\boldsymbol{r}) \to \mathsf{Opt}(\boldsymbol{r})$.

6 Applications

Finally, we apply our main theorem with specific initializations of the random-order free-disposal algorithm and the single-sample single-item prophet inequality for each setting. In particular, we obtain the following guarantees for the indivisible-item, divisible-item, and multi-unit settings.

Theorem 6.1. *There is a* $\frac{1}{6\sqrt{3}} \approx \frac{1}{10.40}$*-competitive* SSCombPI *algorithm for the indivisible-item case under XOS valuations. Furthermore, this algorithm runs in polynomial time and uses a polynomial number of demand and supporting-price queries.*

Proof of Theorem 6.1. Apply Algorithm 3 with Algorithm 4 as the FDAlg component, Algorithm 7 as the SSPI component, and $q = 1 - \frac{1}{\sqrt{3}}$. Then, by Theorems 3.6, 4.1 and 5.1, we obtain the competitive guarantee. Furthermore, since Algorithm 4 runs in polynomial time and uses polynomially many demand and supporting-price queries while Algorithm 7 runs in polynomial time, we obtain the runtime guarantee. □

Theorem 6.2. *There is a* $\frac{1}{6\sqrt{3}} \approx \frac{1}{10.40}$*-competitive* SSCombPI *algorithm for the divisible-item case under continuous capped-XOS valuations. Furthermore, if the valuation functions are concave, then for any* $\epsilon > 0$*, the algorithm achieves competitive ratio* $\frac{1}{6\sqrt{3}} - \epsilon$ *and runs in* $\text{poly}(n, m, \log(1/\epsilon))$ *time using* $\text{poly}(n, m, \log(1/\epsilon))$ *value, first-order (supergradient), and supporting-price queries.*

Proof of Theorem 6.2. Apply Algorithm 3 with Algorithm 5 as the FDAlg component, Algorithm 8 as the SSPI component, and $q = 1 - \frac{1}{\sqrt{3}}$. Then, by Theorems 3.6, 4.6 and 5.2, we obtain the competitive guarantee. Furthermore, as Algorithm 8 runs in polynomial time unconditionally, while Algorithm 5 runs in $\text{poly}(n, m, \log(1/\epsilon))$ time and uses $\text{poly}(n, m, \log(1/\epsilon))$ value, first-order (supergradient), and supporting-price queries (up to an ϵ loss in smooth competitiveness) when the valuation functions are concave (Theorem 4.6), we obtain the runtime guarantee. □

Multi-unit items. In this setting, suppose that there are ℓ identical units of each iteme, each buyer can receive at most one unit of each item, and buyer i has an XOS valuation $v_i(S)$ for the set of items received.

One can "reduce" this problem to the divisible-item case by constructing a capped-XOS valuation for each buyer that caps the demand of each item to at most a $1/\ell$ fraction, and then running the divisible-item algorithm on the constructed instance. However, we need to make sure that the divisible-item algorithm returns allocation vectors that are in $\{0, 1/\ell\}^m$ to map it back to discrete subsets of units. To this end, we show that the same two algorithms used for divisible items also give a valid allocation in this setting. The proof is deferred to Appendix F.

Theorem 6.3. *For every integer* $\ell \geq 1$*, there is a* $\frac{1}{6\sqrt{3}} \approx \frac{1}{10.40}$*-competitive* SSCombPI *algorithm for multi-unit items with* ℓ *units of each item under XOS valuations. It uses Algorithm 3 with exactly the same components as in the divisible-item case: Algorithm 5 as* FDAlg *and Algorithm 8 as* SSPI*. Furthermore, the algorithm runs in polynomial time using polynomially many demand and supporting-price queries.*

7 Related Work

Sample-based prophet inequalities. Prophet inequalities with limited distributional information were initiated by Azar, Kleinberg, and Weinberg [AKW14], who showed that a single sample suffices for constant-factor guarantees in a number of combinatorial selection problems and gave a general reduction from certain order-oblivious secretary algorithms to single-sample prophet inequalities. Rubinstein, Wang, and Weinberg [RWW20] subsequently showed that, for the classical single-choice problem, a particularly simple single-sample threshold rule achieves the optimal 1/2 guarantee obtainable even with complete knowledge of the distributions. A constant number of samples also suffices to get arbitrarily close to the optimal full-information guarantee for

the i.i.d., random-order, and free-order variants [RWW20, GHTZ21, CC23]. Meanwhile, Correa, Cristi, Epstein, and Soto [CCES20] studied the closely related two-sided Game of Googol, making explicit the sample–real symmetry that also underlies our work.

A subsequent line of work has developed direct single-sample algorithms for richer feasibility constraints. Caramanis et al. [CDF+22] introduced a greedy-ordered selection framework that yields improved guarantees for several matroid and matching settings and gives a single-sample algorithm for budget-additive combinatorial auctions. Closely related work of Kaplan, Naori, and Raz [KNR22] studies online weighted matching with a sample and develops connections between random-order algorithms, adversarial-order-with-a-sample algorithms, and single-sample prophet inequalities. More recently, single-sample guarantees have been studied for uniform matroids, multi-unit selection, and various knapsack constraints [PS25, NW26, NRS26]. Our work differs from these results in that the objects arriving online are themselves multidimensional valuation functions: rather than selecting or rejecting a scalar-valued candidate, the algorithm must decide how to allocate several resources to each buyer.

Combinatorial prophet inequalities. There is an extensive literature on prophet inequalities for combinatorial allocation when the underlying distributions are known. Feldman, Gravin, and Lucier [FGL14] showed that static item prices obtain a 1/2 fraction of the optimal expected welfare for XOS valuations. Dütting, Feldman, Kesselheim, and Lucier [DFKL20] developed a general balanced-pricing framework that unifies and extends a number of such prophet-inequality and posted-price results. These works exploit additive prices or supporting prices to summarize the contribution of individual resources to a combinatorial allocation. Supporting prices play a related but different role in our reduction: they allow a provisional multidimensional allocation to be decomposed into single-resource competitions while certifying the value of any allocation obtained by subsequently discarding resources. Other related work has studied random-order arrivals [EHKS18] and broader valuation classes like subadditive valuations [DKL20, CC23].

When the distributions are unknown, obtaining comparable guarantees from very few samples is substantially more difficult. Prior single-sample results for multidimensional allocation were known for special valuation classes such as unit-demand and budget-additive valuations [KNR22, CDF+22]. Most directly related to the present work, Dütting et al. [DKL+24] gave the first constant-factor single-sample prophet inequality for general XOS valuations, obtaining a 1/576 guarantee in the pointwise, or Game of Googol, model. Their algorithm uses buyer-wise greedy with reallocations together with a secretary-style "compete against yourself" argument. The same work also shows that with polynomially many samples one can obtain nearly the 1/2 full-information guarantee through learned posted prices.

Free disposal and online welfare maximization. The buyer-wise greedy algorithm we use for XOS valuations originates in work of Dobzinski, Nisan, and Schapira [DNS05] on welfare maximization for complement-free valuations and also forms a central ingredient of [DKL+24]. Our analysis observes that its free-disposal value can be represented as a monotone submodular function subject to a partition-matroid constraint. This allows us to invoke the random-order greedy analysis of Buchbinder, Feldman, Filmus, and Garg [BFFG20], including the stronger prefix-type guarantee that leads to our notion of smooth competitiveness. In the opposite direction, our unconditional 1/2 impossibility result for adversarial-order XOS free disposal builds on the online submodular welfare maximization impossibility of Udwani and Wajc [UW26], which holds even for randomized algorithms with unlimited computation. This work was built upon the 1/2 computational-hardness of Kapralov, Post, and Vondrák [KPV13] on the same problem. Thus, random order enters our

work only inside the auxiliary free-disposal problem; the actual buyers in the prophet-inequality instance may still arrive adversarially.

Prophet inequalities with divisible resources. Prophet inequalities for divisible resources have received comparatively less attention. Oliveira, Singh, and Singla [OSS26] recently studied online welfare maximization with multidimensional concave valuations over divisible resources and obtained a tight 1/2 prophet inequality under distributional knowledge using Aumann–Shapley supporting prices. Their use of supporting prices is closely related to the continuous supporting-price representation in our work. On the sample-based side, Nuti, Reiffenhäuser, and Sayutina [NRS26] obtain an optimal 1/2 single-sample guarantee for fractional knapsack in the standard independent-draw model. We prove the same guarantee in the stronger Game of Googol model and use fractional knapsack as the single-resource primitive in our reduction. This yields single-sample combinatorial prophet inequalities for continuous capped-XOS valuations over divisible resources, the continuous analogue of the XOS setting considered for indivisible items.

A Extension to Subadditive Valuations

In this section, we provide a straightforward extension to subadditive valuations in the indivisible-item setting. Recall that a valuation $v : 2^{[m]} \to \mathbb{R}_{\geq 0}$ is subadditive if

$$v(S \cup T) \leq v(S) + v(T) \qquad \text{for all } S, T \subseteq [m].$$

Such a valuation need not have the exact supporting prices required by Definition 1. We overcome this obstruction by replacing v with its XOS approximation, obtained via the following classical result.

Theorem A.1 ([Dob07, BR11])**.** *Let $v : 2^{[m]} \to \mathbb{R}_{\geq 0}$ be normalized, monotone, and subadditive. There is a normalized, monotone XOS valuation $\widehat{v}$ such that, for every $S \subseteq [m]$,*

$$\frac{v(S)}{H_m} \leq \widehat{v}(S) \leq v(S),$$

where $H_m = \sum_{i=1}^{m} 1/i$ is the m-th harmonic number.

Given this, it is straightforward to show that any α-competitive prophet inequality for XOS buyers can be turned into an (α/H_m)-competitive prophet inequality for subadditive buyers.

Theorem A.2. *If there is an α-competitive single-sample combinatorial prophet inequality for XOS buyers over m indivisible items, then there is an $\alpha/H_m = \Omega(\alpha/\log m)$-competitive single-sample combinatorial prophet inequality for subadditive buyers over the same items.*

Proof. For a subadditive valuation v, fix a deterministic choice of its approximating XOS valuation $\widehat{v}$ from Theorem A.1. Let $\mathsf{Alg}_{\mathrm{XOS}}$ be an α-competitive single-sample combinatorial prophet inequality for XOS buyers. Given the subadditive samples $\boldsymbol{s}$, run $\mathsf{Alg}_{\mathrm{XOS}}$ on the transformed samples $\widehat{\boldsymbol{s}} := (\widehat{s}_1, \ldots, \widehat{s}_n)$. When buyer i arrives with real valuation r_i, present $\widehat{r}_i$ to $\mathsf{Alg}_{\mathrm{XOS}}$ and allocate to buyer i the bundle S_i selected by $\mathsf{Alg}_{\mathrm{XOS}}$.

The resulting allocation is feasible because it is exactly the allocation selected by $\mathsf{Alg}_{\mathrm{XOS}}$. Moreover, since $\widehat{r}_i(S) \leq r_i(S)$ for every bundle S, its welfare under the original valuations satisfies the following pointwise inequality:

$$\sum_{i=1}^{n} r_i(S_i) \geq \sum_{i=1}^{n} \widehat{r}_i(S_i).$$

Fix a realized real profile $\boldsymbol{r}$, and let $(S_1^*, \ldots, S_n^*)$ be an optimal allocation for $\boldsymbol{r}$. The lower bound in Theorem A.1 gives

$$\mathsf{Opt}(\widehat{\boldsymbol{r}}) \geq \sum_{i=1}^{n} \widehat{r}_i(S_i^*) \geq \frac{1}{H_m} \sum_{i=1}^{n} r_i(S_i^*) = \frac{1}{H_m} \mathsf{Opt}(\boldsymbol{r}).$$

Therefore, taking expectations over the sample-real assignment and all internal randomness, and then using the α-competitiveness of $\mathsf{Alg}_{\mathrm{XOS}}$, we obtain

$$\mathbb{E}\left[\sum_{i=1}^{n} r_i(S_i)\right] \geq \mathbb{E}\left[\sum_{i=1}^{n} \widehat{r}_i(S_i)\right] \geq \alpha \mathbb{E}\left[\mathsf{Opt}(\widehat{\boldsymbol{r}})\right] \geq \frac{\alpha}{H_m} \mathbb{E}\left[\mathsf{Opt}(\boldsymbol{r})\right]. \qquad \square$$

B Deferred Proofs from Section 2

Proof of Proposition 2.1. First, suppose that v has a supporting-price map. For each $\boldsymbol{x} \in [0,1]^m$, define the following capped-additive clause for every $\boldsymbol{x}' \in [0,1]^m$:

$$h_{\boldsymbol{x}}(\boldsymbol{x}') = \sum_{i=1}^{m} q_i(\boldsymbol{x}) \min\{x_i, x_i'\}.$$

Consider the family $\{h_{\boldsymbol{x}}\}_{\boldsymbol{x}\in[0,1]^m}$. For any $\boldsymbol{x}, \boldsymbol{x}' \in [0,1]^m$, write $\boldsymbol{x} \wedge \boldsymbol{x}'$ for the coordinatewise minimum of $\boldsymbol{x}$ and $\boldsymbol{x}'$. We then have

$$h_{\boldsymbol{x}'}(\boldsymbol{x}) = \langle \boldsymbol{q}(\boldsymbol{x}'), \boldsymbol{x}' \wedge \boldsymbol{x} \rangle \le v(\boldsymbol{x}' \wedge \boldsymbol{x}) \le v(\boldsymbol{x})$$

where the first inequality uses the supporting-price property at $\boldsymbol{x}'$, and the second uses monotonicity. On the other hand, $h_{\boldsymbol{x}}(\boldsymbol{x}) = \langle \boldsymbol{q}(\boldsymbol{x}), \boldsymbol{x} \rangle = v(\boldsymbol{x})$. Therefore, $v(\boldsymbol{x}) = \max_{\boldsymbol{x}'\in[0,1]^m} h_{\boldsymbol{x}'}(\boldsymbol{x})$, so v is a capped-XOS valuation function with the capped-additive family $\{h_{\boldsymbol{x}'}\}_{\boldsymbol{x}'\in[0,1]^m}$.

Conversely, suppose that v is capped-XOS, and fix $\boldsymbol{x} \in [0,1]^m$. Choose a capped-additive clause $h(\boldsymbol{y}) = \sum_{i=1}^m a_i \min\{y_i, c_i\}$ such that $h(\boldsymbol{x}) = v(\boldsymbol{x})$. Define

$$q_i = a_i \cdot \frac{\min\{x_i, c_i\}}{x_i}$$

for all $i \in [m]$, with the convention that $q_i = 0$ if $x_i = 0$. It is immediate that $\langle \boldsymbol{q}, \boldsymbol{x} \rangle = \sum a_i \min\{x_i, c_i\} = h(\boldsymbol{x}) = v(\boldsymbol{x})$. For any $\mathbf{0} \le \boldsymbol{x}' \le \boldsymbol{x}$, we then have

$$\langle \boldsymbol{q}, \boldsymbol{x}' \rangle = \sum_{i=1}^{m} a_i \cdot \frac{\min\{x_i, c_i\}}{x_i} \cdot x_i' \le \sum_{i=1}^{m} a_i \cdot \frac{\min\{x_i', c_i\}}{x_i'} \cdot x_i' = \sum_{i=1}^{m} a_i \cdot \min\{x_i', c_i\} = h(\boldsymbol{x}') \le v(\boldsymbol{x}')$$

where the first inequality holds because, for a fixed c_i, the function $\min\{t, c_i\}/t$ is non-increasing in t. Therefore, $\boldsymbol{q}$ is a supporting-price vector at $\boldsymbol{x}$. □

C Deferred Proofs from Section 3

Proof of Lemma 3.2. We first prove the result when PAlloc is deterministic. This is without loss of generality, as we can subsequently take an expectation over the internal randomness of PAlloc.

First, fix the coordination sample vector $\boldsymbol{s}^c$. Then, for each buyer i, $(s_i^1, \dots, s_i^k, r_i)$ forms a random permutation of $k+1$ fixed valuation functions. This implies that the collection of pairs $(\boldsymbol{q}_i^t, \boldsymbol{x}_i^t)$ for all samples $t \in [k]$, together with $(\widehat{\boldsymbol{q}}_i, \widehat{\boldsymbol{x}}_i)$, forms a random permutation of $k+1$ fixed pairs, since the pairs are generated deterministically from $\mathsf{PAlloc}_i(\boldsymbol{s}^c, s_i^t)$ and $\mathsf{PAlloc}_i(\boldsymbol{s}^c, r_i)$ by the local property of PAlloc. Then, by the guarantee of k-SPI, for every item $j \in [m]$, we have

$$\mathbb{E}\left[\sum_{i=1}^{n} x_{i,j}' \widehat{q}_{i,j} \mid \boldsymbol{s}^c\right] \ge \beta \cdot \mathbb{E}_{\boldsymbol{s}^1,\dots,\boldsymbol{s}^k,\boldsymbol{r}}\left[\mathsf{Opt}(\{(\widehat{q}_{i,j}, \widehat{x}_{i,j})\}_{i\in[n]}) \mid \boldsymbol{s}^c\right].$$

Summing this over all $j \in [m]$, we have

$$\mathbb{E}\left[\sum_{i=1}^{n} \langle \boldsymbol{x}_i', \widehat{\boldsymbol{q}}_i \rangle \mid \boldsymbol{s}^c\right] \ge \beta \cdot \mathbb{E}_{\boldsymbol{s}^1,\dots,\boldsymbol{s}^k,\boldsymbol{r}}\left[\sum_{j=1}^{m} \mathsf{Opt}(\{(\widehat{q}_{i,j}, \widehat{x}_{i,j})\}_{i\in[n]}) \mid \boldsymbol{s}^c\right]$$

$$= \beta \cdot \mathbb{E}_{\boldsymbol{s}^1,\ldots,\boldsymbol{s}^k,\boldsymbol{r}} \left[\mathsf{FD}(\widehat{\boldsymbol{x}}_1, \ldots, \widehat{\boldsymbol{x}}_n) \mid \boldsymbol{s}^c \right].$$

Now, since $\boldsymbol{x}'_i \le \widehat{\boldsymbol{x}}_i$ while $\widehat{\boldsymbol{q}}_i = \boldsymbol{q}_{r_i}(\widehat{\boldsymbol{x}}_i)$, the supporting-price property implies that the LHS is at most $\mathbb{E}[\sum_{i=1}^n r_i(\boldsymbol{x}'_i) \mid \boldsymbol{s}^c]$, which is the expected reward of the algorithm given $\boldsymbol{s}^c$. Finally, taking an expectation over $\boldsymbol{s}^c$, we have

$$\mathbb{E}\left[\sum_{i=1}^n r_i(\boldsymbol{x}'_i)\right] \ge \beta \cdot \mathbb{E}\left[\mathsf{FD}(\widehat{\boldsymbol{x}}_1, \ldots, \widehat{\boldsymbol{x}}_n)\right] \ge \alpha\beta \cdot \mathbb{E}[\mathsf{Opt}(\boldsymbol{r})]$$

where we have used the fact that PAlloc is α-competitive under the free-disposal value. □

Proof of Lemma 3.5. Fix the random order π and the internal randomness of FDAlg. By relabeling the buyers, assume that π is the identity permutation. As in the proof of Lemma 3.3, for a partial allocation $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ and every item $j \in [m]$ and value $w > 0$, write

$$\mathsf{FD}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) = \sum_{j=1}^m \int_0^\infty \underbrace{\min\left\{1, \sum_{i=1}^k x_{i,j} \cdot \mathbf{1}[q_{r_i,j}(\boldsymbol{x}_i) \ge w]\right\}}_{y_{j,w}(\boldsymbol{x}_{1..k})} dw.$$

It suffices to prove that

$$\mathbb{E}[y_{j,w}(\boldsymbol{x}_{1..n})] \ge (1-q)\mathbb{E}[y_{j,w}(\boldsymbol{z}_{1..n})]$$

for every j and w. We prove the stronger inequality

$$\mathbb{E}[y_{j,w}(\boldsymbol{x}_{1..n})] \ge (1-q)\mathbb{E}[y_{j,w}(\boldsymbol{x}_{1..n} \cup \boldsymbol{z}_{1..n})],$$

where $y_{j,w}(\boldsymbol{x}_{1..n} \cup \boldsymbol{z}_{1..n})$ denotes the merged free-disposal contribution, as in the proof of Lemma 3.3.

Fix a partial realization of the role coins $H_1, \ldots, H_{k-1}$ and the samples $s'_1, \ldots, s'_{k-1}$ for some $k \in [n]$. Let $\mathcal{H}_{k-1} = (H_1, \ldots, H_{k-1}, s'_1, \ldots, s'_{k-1})$ be this partial realization. This fixes the state $\Sigma^{(k-1)}$ of FDAlg before buyer k, as well as all preceding allocations. Let $f(a) := \min\{1, a\}$, and set

$$\begin{aligned}
A &= \sum_{i=1}^{k-1} x_{i,j} \cdot \mathbf{1}[q_{r_i,j}(\boldsymbol{x}_i) \ge w], \\
B &= \sum_{i=1}^{k-1} \left(x_{i,j} \cdot \mathbf{1}[q_{r_i,j}(\boldsymbol{x}_i) \ge w] + z_{i,j} \cdot \mathbf{1}[q_{s_i^c,j}(\boldsymbol{z}_i) \ge w]\right), \\
c &= u_{k,j} \cdot \mathbf{1}[q_{v_{k,1},j}(\boldsymbol{u}_k) \ge w] \quad \text{where } \boldsymbol{u}_k = \mathsf{FDAlg}(\Sigma^{(k-1)}, v_{k,1}), \\
d &= u'_{k,j} \cdot \mathbf{1}[q_{v_{k,2},j}(\boldsymbol{u}'_k) \ge w] \quad \text{where } \boldsymbol{u}'_k = \mathsf{FDAlg}(\Sigma^{(k-1)}, v_{k,2}).
\end{aligned}$$

Then $0 \le A \le B$ and $c, d \ge 0$. If $H_k = 0$, then buyer k is competing, so $\boldsymbol{x}_k$ is obtained from r'_k, which is equally likely to be either $v_{k,1}$ or $v_{k,2}$, while $\boldsymbol{z}_k$ is produced from the zero valuation. If $H_k = 1$, then buyer k is coordinating, so $\boldsymbol{z}_k$ is obtained from s'_k, which is equally likely to be either $v_{k,1}$ or $v_{k,2}$, while $\boldsymbol{x}_k$ is produced from the zero valuation. Therefore,

$$\mathbb{E}[y_{j,w}(\boldsymbol{x}_{1..k}) - y_{j,w}(\boldsymbol{x}_{1..k-1}) \mid \mathcal{H}_{k-1}] = \frac{1-q}{2}\left(f(A+c) - f(A) + f(A+d) - f(A)\right),$$

whereas

$$\begin{aligned}
&\mathbb{E}[y_{j,w}(\boldsymbol{x}_{1..k} \cup \boldsymbol{z}_{1..k}) - y_{j,w}(\boldsymbol{x}_{1..k-1} \cup \boldsymbol{z}_{1..k-1}) \mid \mathcal{H}_{k-1}] \\
&\quad = \frac{1}{2}\left(f(B+c) - f(B) + f(B+d) - f(B)\right).
\end{aligned}$$

By the concavity of f and the fact that $A \le B$, each marginal increment at A is at least the corresponding marginal increment at B. Hence the first expected increment is at least $1-q$ times the second.

Taking expectations, summing this inequality over k from 1 to n, and telescoping proves the stronger inequality for the fixed random order and internal randomness. Averaging over π and the internal randomness of FDAlg, then summing over j and integrating over w, gives

$$\mathbb{E}[\mathsf{FD}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)] \ge (1-q)\mathbb{E}[\mathsf{FD}(\boldsymbol{z}_1, \dots, \boldsymbol{z}_n)],$$

as desired. □

D Deferred Proofs from Section 4

Lemma D.1. *Consider the partition-matroid instance in Theorem 4.3. Independently retain each part with probability q, and run greedy on the retained parts in a uniformly random order. If A is the resulting set and O is an optimal solution to the original instance, then*

$$\mathbb{E}[f(A)] \ge \left(q - \frac{q^2}{2}\right) f(O).$$

Furthermore, if each greedy step selects an element within additive $\delta \ge 0$ of $\max_{u \in P_i} f(u \mid A^-)$, then $\mathbb{E}[f(A)] \ge \left(q - \frac{q^2}{2}\right)\big(f(O) - \ell\delta\big)$, and when $q = 1$, $\mathbb{E}[f(A)] \ge 0.5096 f(O) - O(\ell\delta)$, where ℓ is the number of parts.

Proof. Give every part P_i an independent arrival time T_i drawn uniformly from $[0,1]$, and let $A(t)$ be the greedy solution after processing all parts with arrival time at most t. For every fixed q, the set of parts with $T_i \le q$ retains each part independently with probability q, and their order is uniformly random. Thus, $A(q)$ has the same distribution as the set A in the statement.

By monotonicity, we may assume that O contains exactly one element o_i from each part P_i. Let $O(t) = \{o_i : T_i > t\}$ be the optimal elements whose parts have not arrived by time t. We first claim that, for every realization of the arrival times and every t,

$$f(A(t)) + f(A(t) \cup O(t)) \ge f(O).$$

Indeed, consider the arrival of a part P_i, and write A^- and A^+ for the greedy sets immediately before and after this arrival, respectively. Let O^- be the set of unarrived optimal elements immediately before the arrival and let $O^+ = O^- \setminus \{o_i\}$. If u_i is the element chosen by greedy, then

$$\begin{aligned} f(A^+) - f(A^-) &= f(u_i \mid A^-) \ge f(o_i \mid A^-) \ge f(o_i \mid A^- \cup O^+) \\ &= f(A^- \cup O^-) - f(A^- \cup O^+) \\ &\ge f(A^- \cup O^-) - f(A^+ \cup O^+), \end{aligned}$$

where the first inequality is the greedy choice, the second is submodularity, and the last is monotonicity. Hence, $f(A(t)) + f(A(t) \cup O(t))$ cannot decrease at an arrival. Its initial value is $f(\emptyset) + f(O) \ge f(O)$, proving the claim.

Let $h(t) = \mathbb{E}[f(A(t))]$. Condition on the process through time $t < 1$. Every unarrived part has instantaneous arrival rate $1/(1-t)$. If P_i arrives next, greedy gains at least $f(o_i \mid A(t))$. Consequently, for almost every $t < 1$,

$$h'(t) \ge \frac{1}{1-t}\mathbb{E}\left[\sum_{i : T_i > t} f(o_i \mid A(t))\right]$$

$$\geq \frac{1}{1-t}\mathbb{E}\left[f(O(t) \mid A(t))\right]$$
$$\geq \frac{f(O) - 2h(t)}{1-t}.$$

The second inequality follows from submodularity, and the last follows from the claim because $f(O(t) \mid A(t)) = f(A(t) \cup O(t)) - f(A(t))$.

Multiplying this differential inequality by $(1-t)^{-2}$ gives

$$\frac{d}{dt}\left(\frac{h(t)}{(1-t)^2}\right) \geq \frac{f(O)}{(1-t)^3}.$$

Since $h(0) = f(\emptyset) \geq 0$, integration from 0 to $q < 1$ yields

$$h(q) \geq (1-q)^2 h(0) + \frac{1-(1-q)^2}{2} f(O) \geq \left(q - \frac{q^2}{2}\right) f(O).$$

The case $q = 1$ follows by continuity.

If each step is δ-approximate, then $f(u_i \mid A^-) \geq f(o_i \mid A^-) - \delta$. If $k(t)$ parts have arrived by time t, summing this over those arrivals gives $f(A(t)) + f(A(t) \cup O(t)) \geq f(O) - k(t)\delta$, while summing over the $\ell - k(t)$ unarrived parts gives $\sum_{i:T_i > t}(f(o_i \mid A(t)) - \delta) \geq f(O(t) \mid A(t)) - (\ell - k(t))\delta$. Combining the two bounds yields $h'(t) \geq \frac{f(O) - 2h(t) - \ell\delta}{1-t}$, which integrates to $h(q) \geq (q - q^2/2)\big(f(O) - \ell\delta\big)$. In the refined analysis of [BFFG20] for $q = 1$, the greedy optimality condition is likewise invoked once per part at its arrival time to lower-bound the step gain (and to bound the auxiliary benchmark at the threshold time), so an additive δ slack per part similarly subtracts $O(\ell\delta)$ from the final guarantee. □

Proof of Theorem 4.2. The impossibility follows from the unconditional lower bound for online submodular welfare maximization (OSWM) of Udwani and Wajc [UW26]. In OSWM, there are n buyers and m arriving items, and each buyer j has a submodular valuation f_j over subsets of the items. The algorithm must immediately and irrevocably assign each arriving item to at most one buyer, seeking to maximize $\sum_{j=1}^{n} f_j(T_j)$, where the T_j are the disjoint sets of assigned items. Under adversarial arrivals, the algorithm may query valuations only on subsets of items that have already arrived.

We use their lower bound for the following subclass of submodular valuations.

Definition 11 (Weighted coverage valuation functions)**.** A valuation function $f : 2^{[m]} \to \mathbb{R}_{\geq 0}$ is a *weighted coverage valuation* if there is a finite universe U, nonnegative weights $w(t)$ for $t \in U$, and a set family $\{A_j\}_{j \in [m]}$ with $A_j \subseteq U$ such that $f(T) = \sum_{t \in \bigcup_{j \in T} A_j} w(t)$ for all $T \subseteq [m]$.

Theorem D.2 ([UW26])**.** *For every constant $\epsilon > 0$, no randomized online algorithm for welfare maximization with weighted coverage valuations under adversarial item arrivals is $(1/2 + \epsilon)$-competitive, even against an oblivious adversary and regardless of computation time.*

Let the instance of OSWM used in the theorem above be defined by the items $1, \ldots, m$ that arrive in this order, the buyers $1, \ldots, n$, and the set families $\mathcal{A}_j = \{A_{j,i}\}_{i \in [m]}$ for all $j \in [n]$. For all $j \in [n]$, let $U_j = \bigcup_{i=1}^{m} A_{j,i}$ be the universe containing the set family $\mathcal{A}_j$. Without loss of generality, assume that $U_1, \ldots, U_n$ are disjoint, since the buyers' valuations are independent of one another. Finally, let the additive weight function w be defined over the universe $U = \bigcup_{j=1}^{n} U_j$, i.e. $w(B) = \sum_{t \in B} w(t)$ for all $B \subseteq U$, so that we have $f_j(T) = w(\bigcup_{i \in T} A_{j,i})$ for all buyers $j \in [n]$.

We now embed such an instance into an online free-disposal allocation (OFDA) instance as follows.

- **The offline items** are the elements of $U = \bigcup_{j \in [n]} U_j$.
- **The online buyers** correspond to the online items in OSWM. Specifically, there are m buyers, arriving in the order $1, \ldots, m$.
- When buyer $i \in [m]$ arrives, their **valuation function** v_i over any subset $S \subseteq U$ is

$$v_i(S) = \max_{j \in [n]} w(S \cap A_{j,i}).$$

  Note that this is obviously an XOS valuation: the j-th additive clause has value $w(t)$ for items $t \in A_{j,i}$, and 0 otherwise.

We will now show that

- Any (online) allocation for OFDA can be mapped (online) to a feasible (online) allocation for OSWM with at least as much welfare, and
- Any (online) feasible allocation for OSWM can be mapped (online) to an (online) allocation for OFDA with the same free-disposal value.

These two conditions are sufficient to establish our hardness result using Theorem D.2. First, the two conditions imply that the two instances have equal hindsight optima. Then, if there were an online algorithm that was $(1/2 + \epsilon)$-competitive for OFDA, there would be an online algorithm that was $(1/2 + \epsilon)$-competitive for OSWM, but this is impossible.

To prove the claims above, the intuition is that, in OFDA, buyer i's choice of a subset S corresponds to choosing the additive clause $j \in [n]$ that supports $v_i(S)$; this, in turn, corresponds to allocating item i to buyer j in OSWM.

OFDA to OSWM. For a partial allocation $S_1, \ldots, S_k$ through buyer $k \in [m]$ in OFDA, let $j_i \in [n]$ be the additive clause that supports $v_i(S_i)$. Correspondingly, in OSWM, we partially assign items $1, \ldots, k$ to buyers $j_1, \ldots, j_k$, respectively, and let $T_1, \ldots, T_n$ denote the sets of items allocated to each of the n buyers so far. Note that this map can be implemented online. Since, in OFDA, the additive clause j_i supports $v_i(S_i)$,

$$\begin{aligned}
\mathsf{FD}(S_1, \ldots, S_k) &= \sum_{t \in U} w(t) \cdot \max_{i \in [k]} \mathbf{1}[t \in S_i \cap A_{j_i,i}] \\
&\leq \sum_{t \in U} w(t) \cdot \max_{i \in [k]} \mathbf{1}[t \in A_{j_i,i}] \\
&= \sum_{j=1}^{n} \sum_{t \in U_j} w(t) \cdot \max_{i \in [k]} \mathbf{1}[t \in A_{j_i,i}] \\
&= \sum_{j=1}^{n} w\left(U_j \cap \bigcup_{i \in [k] : j_i = j} A_{j,i} \right) \\
&= \sum_{j=1}^{n} w\left(\bigcup_{i \in T_j} A_{j,i} \right) = \sum_{j=1}^{n} f_j(T_j)
\end{aligned}$$

where the second equality holds because $U_1, \ldots, U_n$ are disjoint universes whose union is U, while the third and fourth equalities hold because, by definition and disjointness, U_j intersects only $A_{j,i}$.

OSWM to OFDA. In the derivation above, the only inequality occurs because $S_i \cap A_{j_i,i} \subseteq A_{j_i,i}$. Therefore, for a partial feasible assignment of items $1, \dots, k$ to buyers $j_1, \dots, j_k$ in OSWM, if we allocate the set $S_i = A_{j_i,i}$ to buyer $i \in [k]$ in OFDA, then equality holds throughout the derivation. Noting that this map can be implemented online, we have proven our claim. □

E Deferred Proofs from Section 5

Proof of Lemma 5.7. For $a \geq b > 0$, define the concave valuation

$$\phi_{a,b}(x) := a \min\left\{x, \frac{1}{2}\right\} + b\left(x - \frac{1}{2}\right)^+.$$

Thus, its first and second half-units have marginal values a and b, respectively. Fix $M > 10$ and $0 < \epsilon < 1/8$. There are three buyers, numbered $1, 2, 3$, whose low and high valuations are

buyer	L_i	H_i
1	$\phi_{4\epsilon,3\epsilon}$	$\phi_{2,1}$
2	$\phi_{8\epsilon,2\epsilon}$	$\phi_{M,4}$
3	$\phi_{6\epsilon,5\epsilon}$	$\phi_{M-1,\epsilon}$

For every buyer i, the two possibilities in the table form her fixed Game of Googol pair. An independent fair coin assigns one possibility to s_i and the other to r_i. The buyers arrive in the fixed order $1, 2, 3$. Write $z_i = \texttt{0}$ when $(s_i, r_i) = (L_i, H_i)$ and $z_i = \texttt{1}$ when $(s_i, r_i) = (H_i, L_i)$.

We use the strict threshold comparison in the natural algorithm, consistent with Algorithm 8. No tie-breaking is required in this instance, as the twelve marginal values occurring across the six possibilities are all distinct:

$$M > M - 1 > 4 > 2 > 1 > 8\epsilon > 6\epsilon > 5\epsilon > 4\epsilon > 3\epsilon > 2\epsilon > \epsilon.$$

It is convenient to double all rewards so that a selected half-unit of marginal value a contributes a. The sample optimum consists of the two largest sampled marginals, and the threshold equals each of these two values with probability $1/2$. Similarly, the prophet's doubled reward is the sum of the two largest real marginals. Enumerating the eight equally likely orientation vectors gives the following table; the two displayed thresholds in each row are equiprobable.

$\boldsymbol{z}$	τ_s	$\mathbb{E}[2\mathsf{Alg}(\boldsymbol{s}, \boldsymbol{r}) \mid \boldsymbol{z}]$	$2\mathsf{Opt}(\boldsymbol{r})$
000	$(8\epsilon, 6\epsilon)$	3	$2M - 1$
001	$(M - 1, 8\epsilon)$	$\frac{M+3}{2}$	$M + 4$
010	$(M, 4)$	$\frac{M-1}{2}$	$M + 1$
011	$(M, M - 1)$	0	3
100	$(2, 1)$	$M + 4$	$2M - 1$
101	$(M - 1, 2)$	$M + 2$	$M + 4$
110	$(M, 4)$	$\frac{M-1}{2}$	$M - 1 + 8\epsilon$
111	$(M, M - 1)$	0	14ϵ

For example, in orientation 000, all samples are low and all real valuations are high. Buyer 1 therefore consumes the entire item at either threshold and obtains doubled reward $2 + 1 = 3$, whereas the prophet takes the first marginals of buyers 2 and 3 for a doubled reward of $2M - 1$. In orientation 001, the threshold is $M - 1$ or 8ϵ: the algorithm respectively gets the first marginal M

of buyer 2 or both marginals of buyer 1, for a conditional expected doubled reward of $(M+3)/2$. The remaining rows follow in the same way.

Averaging the last two columns gives

$$2\mathsf{Alg} = \frac{7M+19}{16} \qquad \text{and} \qquad 2\mathsf{Opt} = \frac{8M+9+22\epsilon}{8}.$$

Consequently, the competitive ratio on this instance is

$$\frac{\mathsf{Alg}}{\mathsf{Opt}} = \frac{7M+19}{2(8M+9+22\epsilon)}.$$

Finally, keeping any fixed $\epsilon \in (0, 1/8)$ and taking $M \to \infty$ makes the ratio converge to

$$\frac{7}{16} = 0.4375 < \frac{1}{2}.$$

Thus, the competitive ratio of the natural generalization in the single-sample setting is at most $7/16$. □

Proof of Lemma 5.9. Fix $\boldsymbol{r}$. Let $\boldsymbol{x}^*$ be an allocation attaining Opt. We round each coordinate up to the nearest multiple of $1/M$ by setting $\widehat{x}_i := \frac{\lceil Mx_i^* \rceil}{M}$. Let $k_i := \lceil Mx_i^* \rceil$. Monotonicity gives

$$\widehat{\mathsf{Opt}} := \sum_{i=1}^{n} r_i(\widehat{x}_i) \geq \sum_{i=1}^{n} r_i(x_i^*) = \mathsf{Opt},$$

while feasibility of $\boldsymbol{x}^*$ implies

$$K := \sum_{i=1}^{n} k_i \leq M \sum_{i=1}^{n} x_i^* + n \leq M + n.$$

Therefore, $\widehat{\mathsf{Opt}}$ is at most the optimal value obtained by including $K \leq M+n$ marginal gains. Recall that Opt_M is exactly the optimal value obtained by including exactly M marginal gains. One way to obtain Opt_M is to include the largest M marginal gains in $\widehat{\mathsf{Opt}}$, which have value at least $\frac{M}{K} \cdot \widehat{\mathsf{Opt}}$. Therefore,

$$\mathsf{Opt}_M \geq \frac{M}{K} \cdot \widehat{\mathsf{Opt}} \geq \frac{M}{M+n} \cdot \widehat{\mathsf{Opt}} \geq \frac{M}{M+n} \cdot \mathsf{Opt}.$$ □

F Deferred Proofs from Section 6

Proof of Theorem 6.3. We first extend each buyer's XOS valuation $v_i(S) = \max_{t \in L_i} \sum_{j \in S} a_{i,t,j}$ to a capped-XOS valuation by

$$\widetilde{v}_i(\boldsymbol{x}) := \max_{t \in L_i} \sum_{j=1}^{m} a_{i,t,j} \min\{\ell x_j, 1\}.$$

Then $\widetilde{v}_i$ is capped-XOS, is constant in each coordinate beyond $1/\ell$, and satisfies $\widetilde{v}_i(\mathbf{1}_S/\ell) = v_i(S)$ for all $S \subseteq [m]$. By monotonicity and the supporting-price property, any supporting-price vector $\boldsymbol{q}_i(T)$ for v_i (zeroed outside T) is a valid additive clause for v_i that is tight at T; including all such vectors in L_i ensures that $\widetilde{\boldsymbol{q}}_i(\mathbf{1}_S/\ell) := \ell \boldsymbol{q}_i(S)$ is a valid supporting-price vector for $\widetilde{v}_i$ at $\mathbf{1}_S/\ell$.

We next show by induction on i that Algorithm 5 can always choose a provisional allocation $\boldsymbol{x}_i \in \{0, 1/\ell\}^m$. By Claim 4.9, any maximizer of $\Phi_i(\boldsymbol{x}) := \widetilde{v}_i(\boldsymbol{x}) + \mathsf{FD}_{\mathbf{1}-\boldsymbol{x}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1})$ is a valid greedy choice for buyer i, and truncating coordinates of $\boldsymbol{x}$ to $1/\ell$ preserves $\widetilde{v}_i(\boldsymbol{x})$ without decreasing $\mathsf{FD}_{\mathbf{1}-\boldsymbol{x}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1})$. By the induction hypothesis, item j's current free-disposal knapsack consists of ℓ blocks of weight $1/\ell$ (padding unused capacity with density-0 blocks). If b_j is the minimum density among these ℓ blocks, then reducing item j's capacity to $1 - x_j$ for $x_j \in [0, 1/\ell]$ removes mass x_j entirely from this least valuable block, so on $[0, 1/\ell]^m$,

$$\Phi_i(\boldsymbol{x}) = \mathsf{FD}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}) + \widetilde{v}_i(\boldsymbol{x}) - \sum_{j=1}^{m} b_j x_j.$$

Since $\widetilde{v}_i$ is a maximum of linear functions on $[0, 1/\ell]^m$, Φ_i has a maximizer at a vertex $\mathbf{1}_S/\ell$, where $\widetilde{v}_i(\mathbf{1}_S/\ell) - \sum_{j=1}^{m} b_j(\mathbf{1}_S)_j/\ell = v_i(S) - \sum_{j \in S}(b_j/\ell)$. Finding a maximizing vertex $\boldsymbol{x}_i = \mathbf{1}_{S_i}/\ell$ is therefore a single demand query for v_i at prices b_j/ℓ, and computing $\widetilde{\boldsymbol{q}}_i(\boldsymbol{x}_i) = \ell \boldsymbol{q}_i(S_i)$ takes one supporting-price query.

Consequently, every sample and real demand passed to Algorithm 8 lies in $\{0, 1/\ell\}$. Since its residual capacity R starts at 1 and decreases by $\min\{R, x(r_i)\}$, R remains a multiple of $1/\ell$ and every accepted allocation belongs to $\{0, 1/\ell\}$, yielding a feasible multi-unit allocation of equal welfare. Finally, applying Theorems 3.6, 4.6 and 5.2 to $\widetilde{\boldsymbol{v}}$ with $q = 1 - 1/\sqrt{3}$ gives a $\frac{1}{6\sqrt{3}}$-competitive ratio against $\mathsf{Opt}(\widetilde{\boldsymbol{v}}) \geq \mathsf{Opt}(\boldsymbol{v})$, and the polynomial runtime follows from the query implementation above. □

References

[AKW14] Pablo D 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*, pages 1358–1377. SIAM, 2014.

[AS74] Robert J Aumann and Lloyd S Shapley. *Values of Non-Atomic Games*. Princeton University Press, 1974.

[BBK20] Yatao Bian, Joachim M Buhmann, and Andreas Krause. Continuous submodular function maximization. *arXiv preprint arXiv:2006.13474*, 2020.

[BFFG20] Niv Buchbinder, Moran Feldman, Yuval Filmus, and Mohit Garg. Online submodular maximization: beating 1/2 made simple. *Mathematical Programming*, 183(1):149–169, 2020.

[BLKB17] An Bian, Kfir Levy, Andreas Krause, and Joachim M Buhmann. Continuous DR-submodular maximization: Structure and algorithms. *Advances in Neural Information Processing Systems*, 30, 2017.

[BR11] Kshipra Bhawalkar and Tim Roughgarden. Welfare guarantees for combinatorial auctions with item bidding. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 700–709. SIAM, 2011. Full version available at https://timroughgarden.org/papers/subadd.pdf.

[CC23] José Correa and Andrés Cristi. A constant factor prophet inequality for online combinatorial auctions. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 686–697, 2023.

[CCES20] José Correa, Andrés Cristi, Boris Epstein, and José A Soto. The two-sided game of googol and sample-based prophet inequalities. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2066–2081. SIAM, 2020.

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

[CHN14] Moses Charikar, Monika Henzinger, and Huy L. Nguyen. Online bipartite matching with decomposable weights. In *Algorithms – ESA 2014*, volume 8737 of *Lecture Notes in Computer Science*, pages 260–271. Springer, 2014.

[DFKL20] Paul Dütting, Michal Feldman, Thomas Kesselheim, and Brendan Lucier. Prophet inequalities made easy: Stochastic optimization by pricing nonstochastic inputs. *SIAM Journal on Computing*, 49(3):540–582, 2020.

[DKL20] Paul Dütting, Thomas Kesselheim, and Brendan Lucier. An $O(\log \log m)$ prophet inequality for subadditive combinatorial auctions. *ACM SIGecom Exchanges*, 18(2):32–37, 2020.

[DKL+24] Paul Dütting, Thomas Kesselheim, Brendan Lucier, Rebecca Reiffenhäuser, and Sahil Singla. Online combinatorial allocations and auctions with few samples. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1231–1250. IEEE, 2024.

[DNS05] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, pages 610–618. ACM, 2005.

[Dob07] Shahar Dobzinski. Two randomized mechanisms for combinatorial auctions. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 89–103. Springer, 2007.

[EHKS18] Soheil Ehsani, MohammadTaghi Hajiaghayi, Thomas Kesselheim, and Sahil Singla. Prophet secretary for combinatorial auctions and matroids. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 700–714. SIAM, 2018.

[Fei09] Uriel Feige. On maximizing welfare when utility functions are subadditive. *SIAM Journal on Computing*, 39(1):122–142, 2009.

[FGL14] Michal Feldman, Nick Gravin, and Brendan Lucier. Combinatorial auctions via posted prices. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 123–135. SIAM, 2014.

[FKM+09] Jon Feldman, Nitish Korula, Vahab S. Mirrokni, S. Muthukrishnan, and Martin Pál. Online ad assignment with free disposal. In *Internet and Network Economics*, volume 5929 of *Lecture Notes in Computer Science*, pages 374–385. Springer, 2009.

[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.

[GLS88] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[KNR22] Haim Kaplan, David Naori, and Danny Raz. Online weighted matching with a sample. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1247–1272. SIAM, 2022.

[KPV13] Michael Kapralov, Ian Post, and Jan Vondrák. Online submodular welfare maximization: Greedy is optimal. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1216–1225. SIAM, 2013.

[NRS26] Pranav Nuti, Rebecca Reiffenhäuser, and Alice Sayutina. Single-sample prophet inequalities for knapsack problems, 2026. Preprint. Accessible via https://pranavnuti.com/Single-Sample%20Prophet%20Inequalities%20for%20Knapsack%20Problems.pdf.

[NW26] Pranav Nuti and Peter Westbrook. Static pricing for single sample multi-unit prophet inequalities. In *2026 SIAM Symposium on Simplicity in Algorithms (SOSA)*, pages 127–141. SIAM, 2026.

[OSS26] Thiago Oliveira, Mohit Singh, and Sahil Singla. Philosopher and prophet inequalities for divisible items. *arXiv preprint arXiv:2607.11742*, 2026.

[PS25] Kanstantsin Pashkovich and Alice Sayutina. Single sample prophet inequality for uniform matroids of rank 2. *Operations Research Letters*, 60:107257, 2025.

[RWW20] Aviad Rubinstein, Jack Z. Wang, and S. Matthew Weinberg. Optimal Single-Choice Prophet Inequalities from Samples. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 60:1–60:10, 2020.

[UW26] Rajan Udwani and David Wajc. Online submodular welfare maximization: Computation is not the bottleneck. In *2026 IEEE 67th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2026.