Online EFX Allocations with Predictions*

Themistoklis Melissourgos
University of Essex
Colchester, United Kingdom
themistoklis.melissourgos@essex.ac.uk

Nicos Protopapas
Archimedes Unit, Athena Research Center
Athens, Greece
n.protopapas@athenarc.gr

ABSTRACT

Background: Online fair division arises in settings where goods arrive sequentially and must be allocated immediately, without knowledge of future arrivals. Ensuring strong fairness guarantees in such environments is a fundamental challenge.
Objectives and Research Questions: We focus on (approximate) envy-freeness up to any good (EFX) and ask if it can be guaranteed in online allocation problems, and under what conditions such guarantees are possible or impossible. Furthermore, we ask how the availability and accuracy of predictions on agents' valuations can help achieve an EFX allocation.
Methods: We adopt the framework of algorithms with predictions, assuming access to possibly inaccurate predicted valuations, and measure the prediction error using the total variation distance.
Results: First, we show several impossibility results for computing online EFX allocations, either for general valuations or when several agents are involved. Hence we focus on the case of two identical and normalized agents where a simple algorithm yields a $\frac{\sqrt{5}-1}{2}$ approximation. Using predictions we show lower bounds on the needed accuracy to achieve a-EFX approximations for given a, as well as an algorithm that returns an a-EFX approximation for a given prediction accuracy.
Conclusions: We study the use of predictions for the approximation of EFX in online settings, and despite very strong impossibility results, we show how the approximation can gracefully be improved with access to erroneous predictions.

KEYWORDS

Online fair division, Envy-freeness, Algorithms with predictions, EFX

ACM Reference Format:
Themistoklis Melissourgos and Nicos Protopapas. 2026. Online EFX Allocations with Predictions. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 18 pages.

1 INTRODUCTION

Envy-freeness (EF) is one of the most natural and well-studied fair division notions. In the typical setting, there is a common set of resources that needs to be allocated among a given set of agents, each having a personal valuation function for the resources. In the case of divisible resources, the agents can be allocated an arbitrary portion of each resource, and it is known that EF allocations always exist even when restrictions are imposed on the solution [35]. On the other hand, when only *indivisible resources* are available, each of them has to be allocated intact to some agent, and it is easy to show that even approximate EF allocations are not guaranteed to exist for any positive approximation factor: consider the simple case where a single good has to be allocated between two agents who both value it positively.

This non-existence has sparked a fruitful line of work that, over the last 15 years, has proposed several envy-freeness relaxations as close to exact envy-freeness as possible, aiming to prove they always exist. Arguably, the most prominent such relaxation is the notion of *envy-freeness up to any good (EFX)*, which allows agents to envy others as long as the envy is eliminated with the virtual removal of a good with the lowest marginal value from the envied agent's bundle. A more relaxed notion is that of *envy-freeness up to one good (EF1)*, where envy can be eliminated by the virtual removal of a good with the highest marginal value from the envied agent's bundle. The EF1 notion was formally defined by Budish [13] and Lipton et al. [28] under a different name, and it was proven in the same works that it always exists for increasingly broader classes of valuation functions. The stronger notion of EFX was defined independently by Caragiannis et al. [15] and Gourvès et al. [22] and besides specific special cases, most notably for up to three agents [16], a proof of existence is elusive. In particular, even though it is widely conjectured that EFX allocations always exist for any number of agents in the broad class of monotone valuation functions, this has not been refuted so far, and it remains open whether EFX allocations exist even for four agents with additive valuations. Further results are discussed in the excellent survey of Amanatidis et al. [4].

To amend the efforts toward a proof of EFX existence, the notion of *approximate envy-freeness up to any good (a-EFX)* was proposed in [33]. An allocation is a-EFX for some given $a \in [0, 1]$ if, for every agent, her allocated bundle's valuation is no smaller than a times that of any other agent's bundle when a good of minimum marginal value is removed from the latter bundle. It is known from [7] that $(\varphi - 1)$-EFX allocations always exist for any number n of agents with additive valuations. This approximation factor was recently improved to $2/3$ for the case $n \leq 7$ in [5].

Beyond the classical offline setting, many allocation problems arise in dynamic environments where goods appear sequentially and must be allocated immediately and irrevocably. In such settings, it is natural to assess fairness once the allocation process is completed, based on the final distribution of goods. If all values were known in advance, such problems could be reduced to the offline case; but without this information, computing an approximate fair allocation becomes significantly more challenging – even impossible. Indeed, recently Neoh et al. [31] have shown that no meaningful approximation is possible even for the weaker notion

*This work has been accepted as an Extended Abstract in AAMAS 2026.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus.

of EF1 and under mild assumptions, when no information is known about the future arrivals.

A typical example of the *online* setting would be the following. A foodbank collects surplus or near-expiry food *packets* daily from supermarkets and charities, and distributes them to *endpoints* like community kitchens, schools, or shelters. Each packet contains a mix of products from a source, but its exact contents are unknown to the foodbank and the endpoints prior to its arrival. Daily, multiple packets arrive asynchronously at the foodbank, but for ease of presentation, let us consider a single packet arriving each day. The packets are also perishable, so they must be inspected, categorized, and *assigned to an endpoint immediately*, i.e., the same day. The common practice currently among foodbanks is that allocation decisions consider mostly the urgency, relevance, proximity, and capacity of endpoints. This fast and complex decision-making can cause significant envy between endpoints, especially as foodbanks typically do not share data, requiring endpoints to trust the process blindly [1].

To address this, we propose a transparent online fair division model where all packet contents and assignments are made visible to all endpoints.[1] The central question now becomes: *Can fairness be ensured in such a transparent system?* We aim to design algorithms that, over a specified time period, produce a-EFX allocations for the highest possible value of $a \in [0, 1]$. In our model, endpoints are agents with personal valuations. At the start of the month, the foodbank gives all endpoints an estimate of the total contents expected over 30 days. As each packet arrives and is inspected, each endpoint reports its *relative value* as a fraction of the expected month's total, and the foodbank allocates it based on these values.

This simplified, transparent setup removes much of the aforementioned decision complexity, as the only input required from endpoints now is the current relative valuation. However, a key challenge remains: even if the total set of products is known, their distribution across packets is not. This uncertainty, however, prevents any algorithm from guaranteeing a-EFX allocations for any positive a. To help the algorithm of the foodbank achieve approximate EFX allocations, we allow *predictions* in the model, following the recently introduced paradigm of *algorithmic design with predictions* [29]. At the start of the month, the foodbank provides estimated contents for each future packet (e.g., derived via machine learning). Endpoints respond with their estimated relative value for the packets, i.e., a vector of predicted values. Then, as each packet arrives, the foodbank reveals its true contents to the endpoints and asks for their true relative value (which might differ from the respective prediction). Finally, the foodbank runs an a-EFX algorithm which prescribes who will receive the current packet. The foodbank has a contract with its endpoints, guaranteeing that on day 30, the allocation of packets that have been arriving since day 1 will be an a-EFX allocation for some promised value of $a \in [0, 1]$.

We focus on additive valuation functions, one for each agent $i \in N$, over a finite set of goods (one for each time-step). The metric we use to capture the distance between the prediction vector p_i and the vector of true values v_i is the *total variation distance (TV distance).* In order to have a meaningful, uniform magnitude of distance, we consider normalized valuations, where the empty set has value 0 and the set of all goods has value 1 to all agents. Therefore, p_i and v_i can be thought of as probability distributions, and their TV distance (also called *error*) d_i captures the total mispredicted value as a fraction of the entire value that will arrive over time. The quantity $\eta_i = 1 - d_i$ is the prediction accuracy of agent i, and we consider also the worst guarantees over agents, that is, $D := \max_{i \in N} d_i$, and $\eta := 1 - D$. The goods arrive over time, and their true value appears to the agents only upon their arrival. An algorithm takes as input the predictions $(p_i)_{i \in N}$ and the accuracy levels $(\eta_i)_{i \in N}$ and at each time-step $t = 1, \ldots, T$, the true valuation $v_i(g_t)$ of agent i for good g_t is revealed to the algorithm; then the algorithm has to irrevocably allocate the good to an agent. The objective is, for a given level of accuracy η, to design an algorithm that computes an a-EFX allocation with the maximum possible $a \in [0, 1]$. In most results, we assume valuations are normalized, so the algorithm knows the total value of the goods to expect; the algorithm does not know the exact number of arriving goods in advance (the horizon), however.

Normalizing agents' true valuations provides a consistent scale for comparing outcomes and simplifies the analysis of efficiency and fairness. The same assumption has been used in related allocation models [12, 14, 20, 24]. In our setting, normalization is not only analytically convenient but also crucial for obtaining meaningful guarantees: without it, even very weak notions of efficiency or fairness become unattainable. Conceptually, it can be seen as a mild restriction on an otherwise powerful adversary—one that remains adaptive in revealing items but commits to a fixed total value—thus preserving the online nature of the problem while ensuring the results are interpretable.

1.1 Our results

We pose the question: *How does the quality (approximation factor a) of a-EFX allocations depend on the prediction accuracy of the agents?* We consider additive valuations and provide bounds on the prediction accuracy (or equivalently, on the allowed prediction error) as a function of $a \in [0, 1]$.

In Section 3, we explore the limitations of algorithms that do not have per-item predictions (but know the value of the whole set of goods for each agent, i.e., their valuations are normalized). Our results show that for two agents with identical valuations, one can achieve $(\varphi - 1)$-EFX using a simple threshold-based algorithm, while no a-EFX algorithm without predictions exists for any $a \in (\varphi - 1, 1]$. When the valuations are not restricted to be identical, the latter impossibility result holds for any $a \in (0, 1]$. Therefore, to achieve any improvement on a we turn our focus to algorithms with predictions in Section 4.

In Section 4.1 we show that an algorithm with both good consistency and robustness guarantees is impossible, even with two identical and normalized agents. Our main results start in Section 4.2. We start this by exploring the limitations of the extreme case, where the algorithms rely entirely on predictions, disregarding the true values. This will give a first benchmark to compare against. We show that for $n \geq 2$ agents with accuracy at least $1 - \frac{\tilde{a}-a}{(2n-2+\tilde{a})(1+a)}$ for some given $a, \tilde{a} \in [0, 1]$ with $a \leq \tilde{a}$, if an algorithm can compute an $\tilde{a}$-EFX allocation on the predicted values, then a slight modification of it can compute an a-EFX allocation on the true values. We

[1] This can be done anonymously to respect privacy.

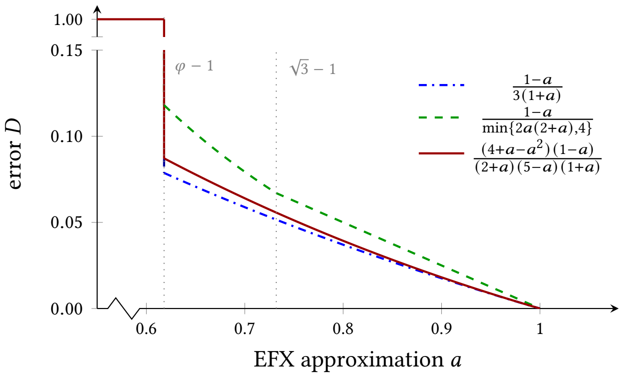

Figure 1: Prediction error $D = 1 - \eta$ as a function of $a \in [0, 1]$ for two agents with identical valuations. Dash-dot plot: Theorem 4.2. Dashed plot: Theorem 4.6. Solid plot: Theorem 3.1 and Theorem 4.8.

show that when $\tilde{a} = 1$, this accuracy is also necessary (the bound is tight) among algorithms oblivious to the true values.

Next, in Section 4.3, we study the limitations of algorithms that use both predictions and true valuations, and show lower bounds on the level of accuracy as a function of the desired a. In particular, we show that for two agents, accuracy of $1 - \frac{1-a}{\min\{6a,4\}}$ is needed by any a-EFX algorithm for $a \in \left(\frac{1}{2}, 1\right]$. This bound slightly improves when the agents have identical valuations, where the necessary accuracy of an a-EFX algorithm becomes $1 - \frac{1-a}{\min\{2a(2+a),4\}}$ for any $a \in (\varphi - 1, 1]$. We show similarly strong bounds for $n \geq 3$ agents with identical valuations, even for $a \in (0, 1]$. All our impossibility results are proven using k-value predictions (i.e., vectors with only $k \in \mathbb{N}$ distinct values) for small k, making them particularly strong.

Finally, we attempt to bridge the gap between the lower bound $\left(1 - \frac{1-a}{\min\{2a(2+a),4\}}\right)$ and the upper bound $\left(1 - \frac{1-a}{3(1+a)}\right)$ on the accuracy for two agents with identical valuations. We manage to do so with Algorithm 2 which guarantees an a-EFX allocation for any $a \in (\varphi - 1, 1]$ while using predictions of accuracy $1 - \frac{(4+a-a^2)(1-a)}{(2+a)(5-a)(1+a)}$. Furthermore, we use this algorithm to derive an improved upper bound of $1 - \frac{2}{5} \cdot \frac{1-a}{1+a}$ when predictions are 2-value functions, and complement it with a lower bound of $1 - \frac{1-a}{2}$.

Fig. 1 illustrates our most important results, namely the bounds derived on the maximum prediction error D as a function of $a \in [0, 1]$ for the case of two agents with additive, identical valuations. The red solid plot visualizes the maximum prediction error under which Algorithm 2 (our main positive result, Theorem 4.8) guarantees an a-EFX allocation for $a \in (\varphi - 1, 1]$ (for $a \in [0, \varphi - 1]$ this error is 1 due to Theorem 3.1, i.e., there is a $(\varphi - 1)$-EFX algorithm regardless of the error). The blue dash-dot plot shows the (tight) bound on the error of algorithms which simply ignore the input and fully rely on the predictions (Theorem 4.2); any meaningful algorithm should handle at least this error. Finally, the green dashed plot shows an upper bound (impossibility result) on the error that any algorithm with predictions can handle for the given accuracy, due to Theorem 4.6.

1.2 Further related work

Early work on online fair division was initiated by Aleksandrov et al. [2], who analyzed randomized mechanisms under ex-ante and ex-post envy-freeness; see also the survey by Aleksandrov and Walsh [3]. Benade et al. [11] studied online allocation with a focus on minimizing envy over time while ensuring Pareto efficiency, showing that approximate guarantees are possible only under stochastic assumptions.

A complementary line of work relaxes the information constraints of online models by allowing algorithms to peek into future arrivals [6, 23, 31], including the framework of *temporal fair division* [18, 19], where fairness must hold at every prefix—an especially restrictive requirement for EFX [19].

Other directions study alternative notions such as maximin share, EF1, or Nash welfare [34, 36, 38], or consider settings where agents, rather than goods, arrive online [25, 26]. Related work also examines the online allocation of divisible resources [8, 9, 20].

A middle ground between full information for the future and no information at all is provided by the recently proposed framework of algorithmic design with predictive advice [29, 30]. Under this paradigm, the input to the algorithm also comes with some possibly unreliable predictions (e.g., about future inputs when considering online algorithms). An updating list of relevant papers is hosted in [27]. The idea has been used to augment fair division, though in markedly different settings. The work of [8] uses prediction in an online fair division problem, where the goal is to allocate divisible resources in order to maximize the Nash welfare. Banerjee et al. [9] use predictions on an online public goods setting, where goods appear online and a fraction of some budget is allocated to them, whereas the work of [10] explores the allocation of perishable goods with the help of predictions. Most relevant to our work, Choo et al. [17] applied this idea to indivisible goods, using predictions to improve guarantees for proportionality up to one good (PROP1), another well-known fairness notion, using the maximum item value as the prediction.

2 PRELIMINARIES

For a positive integer k, we denote by $[k]$ the set $\{1, 2, \ldots, k\}$. Given a non-empty bundle A and a valuation function f, we denote ${}_{\mathrm{x}}A^f := A \setminus \{g\}$, where $g \in \arg\max_{g' \in A} f(A \setminus \{g'\})$, while if $A = \emptyset$, then ${}_{\mathrm{x}}A^f := \emptyset$. Similarly, we denote ${}_{1}A^f := A \setminus \{g\}$, where $g \in \arg\min_{g' \in A} f(A \setminus \{g'\})$, and if $A = \emptyset$, then ${}_{1}A^f := \emptyset$. For notation simplicity, instead of $f({}_{\mathrm{x}}A^f)$ we write $f({}_{\mathrm{x}}A)$, and we omit the superscript when we have identical valuation functions, i.e., instead of ${}_{\mathrm{x}}A^f$ we write ${}_{\mathrm{x}}A$. The same notational simplification also applies to ${}_{1}A^f$. We also simplify the notation when we refer to the valuation of a single good, and write $f(g)$ instead of $f(\{g\})$. Function f will be called *k-value* if its codomain has cardinality at most k. In some of our results, we will use the golden ratio which we denote by $\varphi := \frac{1+\sqrt{5}}{2} \approx 1.618$.

The model. We consider a set $N = [n]$ of $n \geq 2$ agents, and a set $M = \{g_1, g_2, \ldots, g_T\}$ of $T \geq 1$ goods. As it is common in fair division of indivisible goods, we will consider *monotone* valuation functions. A valuation function $f : 2^M \to \mathbb{R}_{\geq 0}$ is monotone if for any two sets $A, B \subseteq M$, we have $f(A \cup B) \geq f(A)$. We will especially focus on a natural subclass of monotone valuation functions, namely that of *additive* valuations. A valuation function $f : M \to \mathbb{R}_{\geq 0}$ is called additive if for any $A \subseteq M$, it holds that $f(A) = \sum_{g \in A} f(g)$. Importantly, we consider *normalized* valuation functions, that is, $f(\emptyset) = 0$ and $f(M) = 1$. The normalization condition is crucial for our setting with predictions, a fact that will become apparent shortly.

The setting is online, and involves discrete time-steps with a finite (true) horizon $T \geq 1$. Each agent i first receives a *prediction*, that is, a vector $p_i = (p_i(g_1), p_i(g_2), \dots, p_i(g_{T'}))$, where $T' \geq 1$ is the predicted horizon, and $p_i : M \to \mathbb{R}_{\geq 0}$ is an additive, normalized valuation function, i.e., $p_i(g_t) \geq 0$ for all $t \in [T']$, and $\sum_{t \in [T']} p_i(g_t) = 1$. Then, at each time $t = 1, 2, \dots, T$, a single good g_t arrives and has to be allocated *irrevocably* to some agent. Each agent $i \in [n]$ at time t evaluates good g_t according to her additive, normalized *true* valuation function $v_i : M \to \mathbb{R}_{\geq 0}$, in other words, a vector $v_i = (v_i(g_1), v_i(g_2), \dots, v_i(g_T))$, where $v_i(g_t) \geq 0$ for all $t \in [T]$, and $\sum_{t \in [T]} v_i(g_t) = 1$. In the special case of *identical valuations* every agent has the same prediction $p = (p(g_t))_{t \in [T']}$, and the same true valuation $v = (v(g_t))_{t \in [T]}$.

$$\mathrm{TV}\,(p_i, v_i) = \|p_i - v_i\|_{\mathrm{TV}} := \frac{1}{2} \sum_{t \in [T_{\max}]} |p_i(g_t) - v_i(g_t)|,$$

where $T_{\max} := \max\{T, T'\}$, and if $T' < T$, then $p_i(g_t) = 0$ for $t \in \{T'+1, T'+2, \dots, T\}$, while if $T' > T$, then $v_i(g_t) = 0$ for $t \in \{T+1, T+2, \dots, T'\}$.[2] For ease of presentation, sometimes we refer to this distance as *error*, and we denote it by $d_i := \mathrm{TV}\,(p_i, v_i)$, and $D := \max_{i \in [n]} d_i$. Each agent i has *accuracy* $\eta_i := 1 - d_i$, which measures the fraction of the total value that is guaranteed to be predicted correctly by the agent. Therefore, if the accuracy is 100%, then the error is 0%, and the agent is capable of perfect predictions. In the other extreme where the accuracy is 0%, the error is 100%, and essentially this is equivalent to having no access to predictions.[3]

Envy-freeness and relaxations. An *allocation* $A = (A_1, A_2, \dots, A_n)$ of a set M of goods to n agents is a partition of M, where *bundle* A_i is allocated to agent $i \in [n]$. We say that agent i *envies* agent j under her valuation f_i if $f_i(A_i) < f_i(A_j)$, she *envies up to any good (EFX-envies)* agent j if $f_i(A_i) < f_i({}_{\mathrm{x}}A_j)$, and she *envies up to one good (EF1-envies)* agent j if $f_i(A_i) < f_i({}_{1}A_j)$. An allocation is *envy-free (EF)* if there is no envious agent, it is *envy-free up to any good (EFX)* if there is no EFX-envious agent, and it is *envy-free up to one good (EF1)* if there is no EF1-envious agent. The quantities $\max\{f_i(A_j) - f_i(A_i), 0\}$, $\max\{f_i({}_{\mathrm{x}}A_j) - f_i(A_i), 0\}$, and $\max\{f_i({}_{1}A_j) - f_i(A_i), 0\}$, are the *envy*, *EFX-envy*, and *EF1-envy* of agent i towards agent j, respectively.

Apart from the exact versions of these fairness notions, we will study the approximate version of EFX and EF1. Under allocation A and valuation f_i, agent i *a-envies up to any good (a-EFX-envies)* agent j if $f_i(A_i) < a \cdot f_i({}_{\mathrm{x}}A_j)$ for some $a \in [0, 1]$. Similarly, agent i *a-envies up to one good (a-EF1-envies)* agent j if $f_i(A_i) < a \cdot f_i({}_{1}A_j)$ for some $a \in [0, 1]$. An allocation is *a-approximately envy-free up to any good (a-EFX)* or *a-approximately envy-free up to one good (a-EF1)* if there is no a-EFX-envious agent or a-EF1-envious agent, respectively. It is straightforward that a 1-EFX or 1-EF1 allocation is an (exact) EFX or EF1 allocation, respectively, and that any allocation is 0-EFX and 0-EF1. Throughout our results, it will be clear whether f_i refers to the prediction p_i or the true valuation v_i.

For some time-step t, we will also denote by $A^t := (A_1^t, A_2^t, \dots, A_n^t)$ the allocation (and the respective bundles to the agents) right after good g_t gets allocated to some agent.

Due to space restrictions, all missing proofs are deferred to the appendix.

3 ALGORITHMS WITHOUT PREDICTIONS

As a warm-up, we demonstrate what qualities of solutions are achievable without the use of predictions, and what qualities are impossible without predictions.

We first consider the easier of the two relaxed envy-freeness notions, EF1, for general, identical valuations, without the use of predictions. It turns out that Theorem 3.7 of [19], originally stated for the temporal EF1 setting, can be applied in our setting, when we have $n \geq 2$ agents with monotone, identical valuations, without predictions, and even without normalization. This result shows that, for any true horizon T, the algorithm that allocates each arriving good to the agent with the lowest-valued bundle, provides at any given time $t \in [T]$ an exact EF1 allocation.

Recently, [31] provided an algorithm that computes an exact EF1 allocation for 2 agents with normalized valuations that are not constrained to be identical, and they also showed that for $n \geq 3$ agents, no algorithm without predictions can achieve an a-EFX for any $a \in (0, 1]$. In fact, concurrent with the drafting of our paper, the aforementioned manuscript appeared, and partially overlaps with our results at Theorems 3.1 to 3.3. Since these results are useful for our exposition, we present the statements for completeness and defer our independently established proofs to the appendix, for the sake of completeness.

Switching to the stronger relaxation of envy-freeness, namely EFX, even for two agents with additive, identical valuations satisfying the normalization condition, no algorithm can achieve an a-EFX for $a \in (\varphi - 1, 1]$ without access to predictions, yet, there is a simple greedy algorithm that matches this bound.

Theorem 3.1. *Suppose we have 2 agents with additive, identical, normalized valuations, and without predictions. Consider the following algorithm: allocate each arriving good to agent 1, as long as the good will not make her exceed value $\varphi - 1$, otherwise allocate it to agent 2. This algorithm guarantees an allocation which is $(\varphi - 1)$-EFX. This bound is tight.*

By increasing the number of agents to three or more, the approximation of EFX becomes impossible without the use of predictions, even when their valuations are identical.

Theorem 3.2. *Suppose we have $n \geq 3$ agents with additive, identical, normalized valuations, and without predictions. For any given*

[2] Notice that if $T' < T$, then the agent sees goods that were not predicted to exist. In that case, she might have extra TV distance from the last $T - T'$ time-steps, calculated by adding $T - T'$ "dummy" time-steps in the prediction and setting those prediction values to 0. Similarly, if $T' > T$, then the agent expects goods that never appear. In that case, to calculate the TV distance, she adds $T' - T$ "dummy" time-steps and sets their true values to 0.

[3] In this paper, we show several impossibility results on k-value valuations for specific $k \in \mathbb{N}$ and notice that, by definition, the result holds for m-value valuations for any $m \geq k$. Also, whenever $T' = T = r$ in our impossibility results, this can be extended to arbitrary $T, T' \geq r$ by introducing "dummy" predicted or true goods with predicted/true value 0 without affecting the analysis.

Algorithm 1 (LPT) Compute an (offline) exact EFX allocation for $n \geq 2$ agents with additive, identical valuations

Require: A valuation function f over set $M = \{g_1, g_2, \dots, g_T\}$.
Ensure: An EFX allocation A for $n \geq 2$ agents.

```
(A_1, A_2, ..., A_n) ← (∅, ∅, ..., ∅)
Rename goods of M according to non-ascending order of value
   f, creating M' := {g'_1, g'_2, ..., g'_T}
for t ∈ [T] do
    Find an agent i* with f(A_{i*}) = min_{i∈[n]} {f(A_i)}
    A_{i*} ← A_{i*} ∪ {g'_t}
return A
```

$a \in (0, 1]$, there is no algorithm that guarantees an a-EFX allocation, even if the time horizon is known.

When we alleviate the "identical valuations" restriction, the approximation of EFX without predictions becomes impossible, even for two agents.

Theorem 3.3. *Suppose we have two agents with additive, normalized valuations, without predictions. For any given $a \in (0, 1]$, there is no algorithm that guarantees an a-EFX allocation, even if the time horizon is known.*

4 ALGORITHMS WITH PREDICTIONS

As we have seen so far, there are sharp dichotomies on the approximation factor of EF1 and EFX allocations when no predictions are involved. In particular, for identical valuations, 1-EF1 exists for $n \geq 2$ agents; $(\varphi - 1)$-EFX exists for $n = 2$ and no a-EFX exists for $a \in (\varphi - 1, 1]$, while for $n \geq 3$ no a-EFX exists for $a \in (0, 1]$. When the valuation functions are not restricted to be identical, the impossibility results become significantly stronger: 1-EF1 exists for $n = 2$, but when $n \geq 3$, no a-EF1 exists for any $a \in (0, 1]$; for $n \geq 2$, no a-EFX exists for any $a \in (0, 1]$. This means that to improve the approximation quality of EF1 and EFX allocations, predictions have to be involved.

4.1 Consistency and Robustness

The usual goal when designing online algorithms with predictions is to guarantee an approximation ratio $c > 0$ under perfect predictions, while at the same time guaranteeing an approximation ratio $r > 0$ under arbitrarily bad predictions. Such an algorithm is called *c-consistent*, and *r-robust*. Clearly $c \geq r$.

Here, we focus on the approximation factor of EFX. Unfortunately, for online EFX allocations, even in the restrictive setting of identical valuations for two agents, it is not possible to design algorithms to preserve both desiderata. As we show in the following theorem, there is a tradeoff between c and r: even if we compromise for c-consistency for c slightly higher than $\varphi - 1$, the approximation we get even without prediction through Theorem 3.1, we cannot yield r-robustness for any $r > \frac{\varphi-1}{2}$. This observation is somewhat discouraging: even a minimal use of predictions to improve the bound recovers only half of the worst-case guarantee available in the no-prediction setting. Moreover, if we require $c = 1$ (an exact EFX allocation), we have to settle for $r \to 0$. In fact, for any $c > a \in [\varphi - 1, 1)$, the robustness cannot be greater than $\frac{1-a}{2a}$.

We will show this with an adversarial argument, following standard techniques from the learning augmented algorithms literature (see e.g., [37]). Notice that if we require an algorithm with consistency $\varphi - 1$, we can use the algorithm of Theorem 3.1 (whose robustness is also $\varphi - 1$). In what follows, we consider algorithms with consistency greater than $\varphi - 1$.

Theorem 4.1. *Suppose we have two agents with additive, normalized valuations, with predictions. For any $a \in [\varphi - 1, 1)$, any algorithm that guarantees a c-EFX under perfect prediction for any $c > a$ cannot guarantee an allocation better than $\frac{1-a}{2a}$-EFX, even when $T' = T = 4$.*

Proof. For the sake of contradiction, fix a value $a \in [\varphi - 1, 1)$, and consider an arbitrary algorithm that guarantees a c-EFX, where $c > a$ when predictions are perfect. Let the adversary provide the prediction $p = \left(0, \frac{1-a}{1+a}, \frac{a}{1+a}, \frac{a}{1+a}\right)$, and notice that $\frac{1-a}{1+a} < \frac{a}{1+a}$. Then, the adversary presents true values $v(g_1) = 0$, $v(g_2) = \frac{1-a}{1+a}$. Observe that for the first two steps, any deterministic algorithm has only two options: either to allocate the items together to a single agent, or to allocate them to two different agents.

If the algorithm allocates g_1, g_2 to the same agent, w.l.o.g., agent 1, then the adversary presents $v(g_3) = v(g_4) = \frac{a}{1+a}$ (perfect prediction). Let us call the resulting allocation A, for which there are three options: (i) if g_3 and g_4 are allocated together to agent 1, then $\frac{v(A_2)}{v({}_{\mathrm{x}}A_1)} = 0 \leq \varphi - 1$, a contradiction; (ii) if g_3 and g_4 are both allocated to agent 2 then $\frac{v(A_1)}{v({}_{\mathrm{x}}A_2)} = \frac{1-a}{a} \leq a < c$, a contradiction. Therefore, the only option is (iii) where w.l.o.g., g_3 goes to agent 1 and g_4 goes to agent 2, and then indeed $v({}_{\mathrm{x}}A_2) = 0$ so agent 1 does not c-EFX-envy agent 2, and $\frac{v(A_2)}{v({}_{\mathrm{x}}A_1)} = a < c$, a contradiction. So the algorithm cannot allocate g_1, g_2 to the same agent.

If the algorithm allocates g_1, g_2 separately, w.l.o.g., g_1 goes to agent 1 and g_2 goes to agent 2, then the adversary presents $v(g_3) = \frac{2a}{1+a}$, $v(g_4) = 0$ (imperfect prediction). For the resulting allocation A there are four options: if g_3 and g_4 are both allocated to agent 1, then $\frac{v(A_2)}{v({}_{\mathrm{x}}A_1)} = \frac{1-a}{2a}$; (ii) if g_3 and g_4 are allocated together to agent 2, then $\frac{v(A_1)}{v({}_{\mathrm{x}}A_2)} = 0 \leq \frac{1-a}{2a}$; (iii) if g_3 goes to agent 1 and g_4 goes to agent 2, then $\frac{v(A_2)}{v({}_{\mathrm{x}}A_1)} = \frac{1-a}{2a}$; and (iv) if g_3 goes to agent 2 and g_4 goes to agent 1, then $\frac{v(A_1)}{v({}_{\mathrm{x}}A_2)} = 0 \leq \frac{1-a}{2a}$. Therefore, no matter how the last two goods are allocated, the resulting allocation cannot be better than $\frac{1-a}{2a}$-EFX. □

For $n = 2$ agents with identical valuations, Theorem 3.1 presented a $(\varphi - 1)$-consistent, $(\varphi - 1)$-robust algorithm albeit oblivious to predictions. As Theorem 4.1 shows, requiring any consistency $c > \varphi - 1$, slashes the robustness to at most $\frac{\varphi-1}{2}$, while if $n \geq 3$ or the valuations are not necessarily identical, Theorems 3.2 and 3.3, respectively, show that the robustness is 0 regardless of the consistency level. In light of these impossibilities, from this point on, we turn to *algorithms that are aware of their accuracy.* We pose the question: *Given a prediction and its accuracy guarantee in the input, what is the best $a \in [0, 1]$ that an a-EFX algorithm can ensure?*

4.2 Full reliance on predictions

We start by exploring the capabilities of algorithms that rely entirely on predictions. The following theorem shows that, when the accuracy in the prediction is good enough, we can get any $\tilde{a}$-EFX algorithm (in the offline setting) as a black box, and with some careful manipulation, we end up with a slightly weaker a-EFX guarantee for the true values.

Theorem 4.2. *Suppose we have $n \geq 2$ agents with additive, normalized valuations. The agents have accuracy $\eta \geq 1 - \frac{\tilde{a}-a}{(2n-2+\tilde{a})(1+a)}$ for some given $a, \tilde{a} \in [0,1]$ with $a \leq \tilde{a}$. Given an $\tilde{a}$-EFX allocation A^o according to $(p_i(g_t))_{i\in[n],t\in[T']}$, we can compute in polynomial time an allocation B which is a-EFX according to the true valuations $(v_i(g_t))_{i\in[n],t\in[T]}$.*

The proof of Theorem 4.2 reveals that the algorithm used to produce an a-EFX allocation (according to the true valuations) is oblivious to the true valuations themselves. In particular, first it takes as input an $\tilde{a}$-EFX allocation based on the predicted values, then shifts bundles around such that there is an agent who is not envied by any other, and then takes that prescribed allocation and follows it blindly on the true values (where, if unpredicted goods appear, they get allocated to the unenvied agent). We can actually show that when we can compute an exact EFX allocation with respect to the predicted values, the accuracy bound of Theorem 4.2 is tight among algorithms that rely entirely on predictions. In other words, if it is possible to compute a 1-EFX allocation for the predictions, for the algorithms that follow any 1-EFX derived by the predictions and ignore the true values, the level of accuracy stated in the aforementioned result for $\tilde{a} = 1$, i.e. $1 - \frac{1-a}{(2n-1)(1+a)}$, is necessary to achieve an a-EFX allocation. We show that this is true even for identical valuations.

Proposition 4.3. *Suppose we have $n \geq 2$ agents with additive, identical, normalized valuations. The agents have accuracy $\eta < 1 - \frac{1-a}{(2n-1)(1+a)}$ for some given $a \in [0,1]$, that is, the error between the provided prediction $(p(g_t))_{t\in[T']}$ and the true values $(v(g_t))_{t\in[T]}$ is $D = 1-\eta > \frac{1-a}{(2n-1)(1+a)}$. Then, any algorithm that computes an exact EFX allocation on the predictions and follows it by ignoring the true values, is not able to compute an a-EFX allocation according to the true valuation, even when $T' = T$.*

Proof. Consider the following instance with identical valuations, for which we can compute an exact EFX allocation on the predictions (e.g., by using Algorithm 1). Let $T' = 2n-1$, and where $p(g_t) = \frac{1}{2n-1}$ for all $t \in [2n-1]$. Then, the only 1-EFX allocation (regardless of the algorithm used to produce it) is one where all agents receive 2 goods, except for one, w.l.o.g. agent 1, who receives 1 good. Consider an adversary producing $T = 2n-1$ true values $v(g_1) = \frac{1}{2n-1} - D < \frac{2a}{(2n-1)(1+a)}$, $v(g_2) = \frac{1}{2n-1} + D > \frac{2}{(2n-1)(1+a)}$, $v(g_t) = \frac{1}{2n-1}$ for $t \in \{3,4,\ldots,2n-1\}$. The algorithm ignores the true values and assigns the goods according to the aforementioned 1-EFX allocation based only on the predictions. Now $v(A_1) = v(g_1) < \frac{1}{2n-1} - \frac{1-a}{(2n-1)(1+a)} = \frac{2a}{(2n-1)(1+a)}$, and $v({}_{\text{x}}A_2) = v(g_2) > \frac{1}{2n-1} + \frac{1-a}{(2n-1)(1+a)} = \frac{2}{(2n-1)(1+a)}$, therefore this is an a'-EFX allocation for $a' \leq \frac{v(A_1)}{v({}_{\text{x}}A_2)} < a$. So the algorithm fails to produce an a-EFX allocation. □

Before proceeding to show an example of Theorem 4.2, we mention a result – considered to be folklore by now – for efficiently computing exact EFX allocations for $n \geq 2$ agents with additive, identical valuations. Algorithm 1 is essentially the well-known *longest-processing-time-first (LPT)* algorithm, from the literature of job-scheduling over identical machines (see e.g., [32]). There, the problem is to schedule jobs (here a job is represented by a good $g_t, t \in [T]$), each coming with a known processing time (here, $f(g_t)$), into machines (here, bundles A_i of agents $i \in [n]$) and the goal is to minimize the maximum processing time of a machine (here, $f(A_i)$) over all machines. It is easy to show that in the context of fair division of indivisible goods under an additive valuation, it provides an exact EFX allocation.[4] The LPT algorithm is a central component of Algorithm 2.

To showcase the contribution of predictions to the quality of EFX allocations, let us see the following instantiation of Theorem 4.2. We have the simple case of two agents with additive, identical valuations. As Theorem 3.1 shows, an algorithm without access to predictions cannot guarantee an a-EFX for any $a \geq 0.619$. In contrast, when we are allowed prediction of accuracy 94.5%, Algorithm 1 can guarantee a 0.718-EFX allocation in polynomial time.

Example 4.4. For two agents with additive, identical, normalized valuations, LPT (Algorithm 1) guarantees an exact EFX (that is, a 1-EFX) on the predicted values in polynomial time. If, additionally, the agents have prediction accuracy $\eta \geq 1 - \frac{1-(\varphi-0.9)}{(2\cdot 2-2+1)(1+\varphi-0.9)} \simeq 0.945$, then LPT's allocation is $(\varphi - 0.9) \simeq 0.718$-EFX on the true values.

4.3 Use of predictions and true values

The limitations derived so far, show that, unless both predictions and the observed (true) values are taken into consideration by an algorithm, the accuracy required to output an a-EFX allocation for a given $a \in [0,1]$, is relatively high, even for identical valuations (see Proposition 4.3). To bridge the gap between the lower and the upper bound on the accuracy, we turn our focus on algorithms that use the information of the predictions to decide where to allocate each arriving good.

4.3.1 Impossibility results. We first show lower bounds on the necessary level of accuracy, which essentially tell us that no algorithms to compute a-EFX exist when their accuracy is below a threshold. Recall from Theorem 3.3 that, without predictions, no algorithm can guarantee an a-EFX allocation for any $a \in (0,1]$, even for two agents. The following lower bound shows that it remains impossible to compute an a-EFX for any $a \in \left(\frac{1}{2}, 1\right]$ even for algorithms enhanced with predictions of accuracy less than a particular threshold. For the case of $a \in \left(0, \frac{1}{2}\right)$ we know from Theorem 3.3 that predictions (i.e., positive accuracies) are necessary to get an allocation better than 0-EFX, but it is an open question whether there is a positive lower bound on the accuracy.

Theorem 4.5. *Suppose we have 2 agents with additive, normalized valuations, with a provided prediction of accuracy $\eta < 1 - \frac{1-a}{\min\{6a,4\}}$ for some given $a \in \left(\frac{1}{2}, 1\right]$. Then, there is no algorithm that guarantees an a-EFX allocation, even when $T' = T = 4$, and the predictions and the true valuations are 4-value functions.*

When the valuation functions are identical, the accuracy lower bounds become slightly better. Recall that Theorem 3.1 presents

[4] We note that a generalized version of this algorithm has been used in [21] to find EFX allocations for a superclass of additive valuations. For the sake of completeness, we present a self-sustained proof for additive valuations in the Proposition C.7.

a simple greedy algorithm that achieves $(\varphi - 1)$-EFX allocations without predictions, which is best possible in the absence of predictions. In the following theorem, we show that even if an algorithm has access to predictions of up to a certain accuracy level, it still cannot guarantee an a-EFX for any $a \in (\varphi - 1, 1]$.

Theorem 4.6. *Suppose we have 2 agents with additive, identical, normalized valuations, with a provided prediction of accuracy* $\eta < 1 - \frac{1-a}{\min\{2a(2+a),4\}}$ *for some given* $a \in (\varphi - 1, 1]$. *Then, there is no algorithm that guarantees an a-EFX allocation, even when* $T' = T = 4$, *and the predictions and the true valuations are 4-value functions.*

Proof. For the sake of contradiction, suppose there is an algorithm that for some $a \in (\varphi-1, 1]$ and accuracy $\eta < 1-\frac{1-a}{\min\{2a(2+a),4\}}$, guarantees an a-EFX allocation.

If $a \in (\varphi - 1, \sqrt{3} - 1]$, and therefore, $2a(2 + a) \leq 4$, the adversary fixes a rational $r \in \left(0, D - \frac{1-a}{2a(2+a)}\right]$, sets $\lambda = \max\left\{\frac{a^2+a-1}{2a(2+a)} - r, 0\right\}$, and an $\varepsilon \in \left(\frac{1-a}{2a(2+a)} + r \cdot \frac{1-a}{1+a}, \frac{1-a}{2a(2+a)} + r\right)$. It then gives the following prediction to the algorithm: $p(g_1) = 2\lambda$, $p(g_2) = 2\varepsilon$, $p(g_3) = \frac{1}{2} - 2\varepsilon - \lambda$, and $p(g_4) = \frac{1}{2} - \lambda$, and observe that $\varepsilon \geq \lambda$. The adversary reveals the true values $v(g_1) = 2\lambda$, $v(g_2) = 2\varepsilon$. There are two cases:

(i) If g_1 and g_2 get allocated to the same agent, w.l.o.g. agent 1, then the adversary reveals the true values $v(g_3) = v(g_4) = \frac{1}{2} - \varepsilon - \lambda$. If g_3 and g_4 get allocated to agent 1, this is a 0-EFX allocation, so the algorithm fails to provide an a-EFX allocation for the target a.

If g_3 and g_4 get allocated to agent 2, then agent 2 does not EFX-envy agent 1, since $1 - 2(\varepsilon + \lambda) \geq 2\varepsilon$ for the domains of ε, a and this is an a'-EFX allocation with $a' = \frac{2(\varepsilon+\lambda)}{1/2-\varepsilon-\lambda} < a$, where the inequality comes from the fact that $\varepsilon < \frac{1-a}{2a(2+a)} + r$. If g_3 is allocated to agent 1 and g_4 is allocated to agent 2, then agent 1 does not envy agent 2, and notice that g_1 is the smallest-valued good that agent 1 has, since $2\lambda \leq 2\varepsilon \leq \frac{1}{2} - \varepsilon - \lambda$ for the given domains of ε and a. This is an a'-EFX allocation with $a' = \frac{1/2-\varepsilon-\lambda}{1/2+\varepsilon-\lambda} < a$, where the inequality comes from the fact that $\varepsilon > \frac{1-a}{2a(2+a)} + r \cdot \frac{1-a}{1+a}$. If g_3, g_4 are allocated to agent 2 and 1, respectively, a symmetric argument applies.

(ii) If w.l.o.g. g_1 is allocated to agent 1 and g_2 is allocated to agent 2, then the adversary reveals the true values $v(g_3) = \frac{1}{2} - 3\varepsilon - \lambda$ and $v(g_4) = \frac{1}{2} + \varepsilon - \lambda$. Notice that g_1 is still the smallest-valued good among all, since $2\lambda \leq 2\varepsilon \leq \frac{1}{2} - 3\varepsilon - \lambda \leq \frac{1}{2} + \varepsilon - \lambda$ for the given domains of ε and a. If both g_3 and g_4 get allocated to agent 1, then he does not EFX-envy agent 2, and this is an a'-EFX allocation for $a' = \frac{2\varepsilon}{1-2(\varepsilon+\lambda)} \leq \frac{2(\varepsilon+\lambda)}{1/2-\varepsilon-\lambda} < a$, where the last inequality comes from one of the bounds in case (i). If both g_3 and g_4 get allocated to agent 2, then he does not EFX-envy agent 1, and this is an a'-EFX allocation for $a' = \frac{2\lambda}{1-2(\varepsilon+\lambda)} \leq \frac{2\varepsilon}{1-2(\varepsilon+\lambda)} < a$, where the last inequality comes from the previous bound. If g_3 gets allocated to agent 2 and g_4 gets allocated to agent 1, then agent 1 does not envy agent 2, and this is an a'-EFX for $a' = \frac{1/2-\varepsilon-\lambda}{1/2+\varepsilon-\lambda} < a$, where the inequality comes from one of the bounds in case (i). If g_3 goes to agent 1 and g_4 goes to agent 2, then agent 2 does not envy agent 1, and this is an a'-EFX for $a' = \frac{1/2-3\varepsilon+\lambda}{1/2+\varepsilon-\lambda} \leq \frac{1/2-\varepsilon-\lambda}{1/2+\varepsilon-\lambda} < a$, where the last inequality comes from the previous bound.

From the above cases we can see that the algorithm fails to output an a-EFX allocation. Now observe that the error between the prediction and the true values in the two cases is equal to $\varepsilon < \frac{1-a}{2a(2+a)} + r \leq D$, which contradicts our assumption that the algorithm can provide an a-EFX allocation for error (at most) D.

If $a \in (\sqrt{3} - 1, 1]$, or equivalently, $2a(2 + a) > 4$, the adversary fixes a rational $\varepsilon \in \left(\frac{1-a}{4}, \min\left\{\frac{a}{4(2+a)}, D\right\}\right)$ and gives the following prediction to the algorithm: $p(g_1) = 2\varepsilon$, $p(g_2) = 2\varepsilon$, $p(g_3) = \frac{1}{2} - 3\varepsilon$, and $p(g_4) = \frac{1}{2} - \varepsilon$. Then, we follow the same case analysis as above, where we substitute λ with ε. Therefore, the algorithm fails to output an a-EFX allocation. Notice that, again, the error of this instance equals ε, which can take any value in $\left(\frac{1-a}{4}, \min\left\{\frac{a}{4(2+a)}, D\right\}\right)$, a non-empty interval for $a \in (\sqrt{3} - 1, 1]$. Therefore, the adversary can choose some ε arbitrarily close to the lower limit of that interval, making the algorithm fail to provide an a-EFX for error less than D, a contradiction. □

When we have $n \geq 3$ agents, this bound becomes significantly worse, especially for large values of a. Furthermore, we get non-zero lower bounds for small values of a, contrary to the case of two agents. We show this in the following theorem.

Theorem 4.7. *Suppose we have* $n \geq 3$ *agents with additive, identical, normalized valuations, with a provided prediction of accuracy* $\eta < 1 - \min\left\{\frac{1}{2(n-1+2a)}, \frac{1-a^2}{4+(2n-3)a}\right\}$ *for some given* $a \in (0, 1]$. *Then, there is no algorithm that guarantees an a-EFX allocation, even when* $T' = T = 2n - 1$, *and the predictions and the true valuations are 3-value functions.*

One can observe from Theorem 4.7 that when $n \geq 3$, for any fixed $a \in (0, 1)$, the lower bound on the accuracy is a function $1 - \Theta(1/n)$. The upper bound on the accuracy given in Theorem 4.2 for identical valuations becomes $1 - \frac{1-a}{(2n-1)(1+a)}$, which is also a $1-\Theta(1/n)$ function. This means that as the number of agents grows, the necessary accuracy level tends to 100%, and is asymptotically matched by the accuracy that suffices for a simple algorithm relying entirely on predictions. This fact at first glance might seem to make the case that for large n, finding better algorithms than those that blindly use predictions is not interesting. However, for fixed a and n, there is still a non-negligible gap between the lower and upper bounds of accuracy, which is worth investigating. In the following section, we try to bridge the largest bound-gap for identical valuations, which is for the case of two agents.

4.3.2 Positive results.

We focus on two agents with additive, identical valuations, and present Algorithm 2 which, for the same approximation factor a, requires less accuracy than that of Theorem 4.2 (or equivalently, for the same accuracy, it provides an approximate EFX allocation with better approximation factor than that of the aforementioned result). This a-EFX algorithm has a known accuracy that is a function $\eta(a)$ of $a \in [0, 1]$. It starts by finding an exact EFX allocation A using LPT (Algorithm 1) on the predicted values, and uses this as a guideline for the final allocation. According to the allocation's characteristics (its form), it prescribes appropriate thresholds for true values of goods that arrive over time, and allocates them to the right agent so that the final allocation is a-EFX.

Theorem 4.8. *Suppose we have 2 agents with additive, identical, normalized valuations, with a provided prediction of accuracy* $\eta \geq 1 - \frac{(4+a-a^2)(1-a)}{(2+a)(5-a)(1+a)}$ *for some given* $a \in (\varphi - 1, 1]$. *Then, Algorithm 2*

Algorithm 2 Two agents with additive, identical valuations: Computing an a-EFX for $a \in (\varphi - 1, 1]$ when prediction accuracy is at least $1 - \frac{(4+a-a^2)(1-a)}{(2+a)(5-a)(1+a)}$

Require: A prediction vector $(p(g_t))_{t \in [T']}$.
Ensure: An a-EFX allocation B.

1: $(A_1, A_2) \leftarrow (\emptyset, \emptyset)$, $\quad (B_1, B_2) \leftarrow (\emptyset, \emptyset)$
2: $d_{\max} \leftarrow \frac{(4+a-a^2)(1-a)}{(2+a)(5-a)(1+a)}$
3: $z \leftarrow$ highest value in p, $\quad y \leftarrow$ 2^{nd} highest value in p
4: $g^z \in \{g \mid p(g) = z\}$, $g^y \in \{g \mid p(g) = y\}$, $g^x \in \{g \mid p(g) < y\}$
5: **if** $T' \leq 3$ **then**
6: $\quad (B_1, B_2) \leftarrow$ Output of Algorithm 3 in Section C.3
7: $\quad$ **go to** line 28
8: $(A_1, A_2) \leftarrow$ Output of Algorithm 1 with input p $\quad$ //A is exact EFX allocation based on p
9: **if** $p(A_1) \geq \frac{4+a-a^2}{(2+a)(5-a)}$ **or** $a = 1$ **then**
10: $\quad (B_1, B_2) \leftarrow (A_1, A_2)$ $\quad$ //Proposition C.5 applies, or $a = 1$
11: $\quad$ **for** $t \in \{T'+1, T'+2, \ldots, T\}$ **do**
12: $\quad\quad B_1 \leftarrow B_1 \cup \{g_t\}$ $\quad$ //All goods g_t with $t > T'$ (if any) get allocated to agent 1
13: **else**
14: $\quad$ **for** $t \in [T]$ **do**
15: $\quad\quad$ **if** Form 1: $A_2 = \{g_1^z, g_2^z\}$, $A_1 = \{g_3^z, g_1^y, g_2^y, \ldots, g_\ell^y, g_1^x, g_2^x, \ldots, g_k^x\}$ for $\ell + k \geq 1$ **then**
16: $\quad\quad\quad$ **if** $g_t = g^z$ with $v(g_t) \leq z + \frac{d_{\max}}{2}$ **and** $|B_2| \leq 1$ **then**
17: $\quad\quad\quad\quad B_2 \leftarrow B_2 \cup \{g_t\}$ **else** $B_1 \leftarrow B_1 \cup \{g_t\}$
18: $\quad\quad$ **if** Form 2 or Form 4: $A_2 = \{g_1^y, g^*\}$, where $g^* \in \{g_2^y, g^x\}$, $A_1 = \{g^z, g_1^x, g_2^x, \ldots, g_k^x\}$, for $k \geq 1$ **then**
19: $\quad\quad\quad$ **if** $g_t \in \{g_1^y, g^*, g^z\}$ with ($v(g_t) \leq y + \frac{d_{\max}}{2}$ **and** $|B_2| \leq 1$) **or** $|\{g_1^y, g^*, g^z\} \cap B_1| \geq 1$ **then**
20: $\quad\quad\quad\quad B_2 \leftarrow B_2 \cup \{g_t\}$ **else** $B_1 \leftarrow B_1 \cup \{g_t\}$
21: $\quad\quad$ **if** Form 3: $A_2 = \{g_1^z, g_1^y\}$, $A_1 = \{g_2^z, g_1^x, g_2^x, \ldots, g_k^x\}$ for $k \geq 1$ **then**
22: $\quad\quad\quad$ **if** in p, good g^y does not arrive after both g^z goods **then**
23: $\quad\quad\quad\quad$ **if** $g_t \in \{g_1^y, g_1^z, g_2^z\}$ with $v(g_t) \leq z + \frac{d_{\max}}{2}$ **and** $|B_2| \leq 1$ **then**
24: $\quad\quad\quad\quad\quad B_2 \leftarrow B_2 \cup \{g_t\}$ **else** $B_1 \leftarrow B_1 \cup \{g_t\}$
25: $\quad\quad\quad$ **else**
26: $\quad\quad\quad\quad$ **if** $g_t \in \{g_1^y, g_1^z, g_2^z\}$ with ($v(g_t) \leq z + \frac{(1-a)^2}{(2+a)(5-a)}$ **and** $|B_2| \leq 1$) **or** $|\{g_1^y, g_1^z, g_2^z\} \cap B_1| \geq 1$ **then**
27: $\quad\quad\quad\quad\quad B_2 \leftarrow B_2 \cup \{g_t\}$ **else** $B_1 \leftarrow B_1 \cup \{g_t\}$
28: **return** (B_1, B_2)

outputs an a-EFX allocation, and performs a constant number of basic operations per time-step.

Proof Sketch. Algorithm 2 first uses Algorithm 1 to find an exact EFX allocation on the prediction, called A. Then, A is classified according to its *form*, that is, some important properties of the two bundles A_1, A_2. Finally, a form-specific routine is used, which, at each time-step decides the recipient of the good in constant time, resulting in an a-EFX allocation B.

Let $d_{\max} := \frac{(4+a-a^2)(1-a)}{(2+a)(5-a)(1+a)}$ denote the maximum error between the prediction and the true valuation for an a-EFX allocation. W.l.o.g., let us rename agent 1, agent 2 such that it holds $p(A_1) \leq p(A_2)$. The algorithm first checks if the prediction has horizon at most 3, which is an easy case handled by Algorithm 3 (see Section C.3). If that is not the case, it proceeds to check if $a = 1$, which would mean that the algorithm knows that the prediction is perfect ($d_{\max} = 0$), and so it assigns all predicted goods according to A, and any other ones that might arrive with value 0 to agent 1. The most challenging part of the algorithm is when $a < 1$. We show that whenever $p(A_1) \geq \frac{1+a}{1-a} \cdot d_{\max} = \frac{4+a-a^2}{(2+a)(5-a)}$, for some $a \in (\varphi - 1, 1)$, then A is an a-EFX allocation according to the true values. What remains to show is how the algorithm produces an a-EFX allocation B when $p(A_1) \in \left[\frac{1}{3}, \frac{4+a-a^2}{(2+a)(5-a)}\right)$. In this latter case, we show that $|A_1| \leq 2$ and we specifically focus on $|A_2| = 2$ – the other cases are trivially or easily resolved. At this point, we split the inputs into various forms based on the top three values in the prediction (e.g., in Form 1, A_1 consists of two items with the highest value in the prediction), and through a careful case analysis, we establish that the allocation returned by the algorithm, given the accuracy guarantee, is a-EFX for each form. ◻

Example 4.9. As we saw in Theorem 4.4, for prediction accuracy 0.945, LPT can guarantee a 0.718-EFX allocation on the true values. According to Theorem 4.8, for the same accuracy, Algorithm 2 guarantees a 0.734-EFX allocation on the true values. Conversely, accuracy 0.941 is enough in order for the algorithm to provide a 0.718-EFX allocation.

4.3.3 *Results on 2-value predictions.*

One can notice that, even though Algorithm 2 works for two agents with any additive, identical valuation, in essence, it only considers the three highest levels of value inside the prediction vector p. This means that, even if we were restricted to have a 3-value predictor, the proof of Theorem 4.8 would not give any better bounds. However, in the case where we have a 2-value predictor, the same proof yields improved results. It is important to note that, even though the prediction is a 2-value function, the true valuation is not restricted at all.

Corollary 4.10 (Corollary of Theorem 4.8). *Suppose we have 2 agents with additive, identical, normalized valuations, with a provided 2-value prediction of accuracy $\eta \geq 1 - \frac{2}{5} \cdot \frac{1-a}{1+a}$ for some given $a \in (\varphi - 1, 1]$. Then, Algorithm 2 outputs an a-EFX allocation, and performs a constant number of basic operations per time-step.*

It is natural that the accuracy upper and lower bounds improve in this case. However, notice that the latter improves disproportionately, indicating that for 2-value predictions, most likely there is an algorithm with better guarantees than Algorithm 2.

Theorem 4.11. *Suppose we have 2 agents with additive, identical, normalized valuations, with a provided 2-value prediction of accuracy $\eta < 1 - \frac{1-a}{2}$ for some given $a \in \left(\sqrt{3} - 1, 1\right]$. Then, there is no algorithm that guarantees an a-EFX allocation, even when $T' = T = 4$.*

ACKNOWLEDGMENTS

This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

REFERENCES

[1] Renzo Akkerman, Marjolein Buisman, Frans Cruijssen, Sander de Leeuw, and Rene Haijema. 2023. Dealing with donations: Supply chain management challenges for food banks. *Int. J. Prod. Econ.* 262 (2023), 108926.

[2] Martin Aleksandrov, Haris Aziz, Serge Gaspers, and Toby Walsh. 2015. Online Fair Division: Analysing a Food Bank Problem. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI '15*. 2540–2546.

[3] Martin Aleksandrov and Toby Walsh. 2020. Online Fair Division: A Survey. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*. 13557–13562.

[4] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A Voudouris, and Xiaowei Wu. 2023. Fair division of indivisible goods: Recent progress and open questions. *Artif. Intell.* 322 (2023), 103965.

[5] Georgios Amanatidis, Aris Filos-Ratsikas, and Alkmini Sgouritsa. 2024. Pushing the Frontier on Approximate EFX Allocations. In *Proceedings of the 25th ACM Conference on Economics and Computation, EC '24*. 1268–1286.

[6] Georgios Amanatidis, Alexandros Lolos, Evangelos Markakis, and Victor Turmel. 2025. Online Fair Division for Personalized 2-Value Instances. In *Algorithmic Game Theory - 18th International Symposium, SAGT 2025, Proceedings.* to appear.

[7] Georgios Amanatidis, Evangelos Markakis, and Apostolos Ntokos. 2020. Multiple birds with one stone: Beating 1/2 for EFX and GMMS via envy cycle elimination. *Theoret. Comput. Sci.* 841 (2020), 94–109.

[8] Siddhartha Banerjee, Vasilis Gkatzelis, Artur Gorokh, and Billy Jin. 2022. Online Nash social welfare maximization with predictions. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 1–19.

[9] Siddhartha Banerjee, Vasilis Gkatzelis, Safwan Hossain, Billy Jin, Evi Micha, and Nisarg Shah. 2023. Proportionally Fair Online Allocation of Public Goods with Predictions. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI '23*. 20–28.

[10] Siddhartha Banerjee, Chamsi Hssaine, and Sean R. Sinclair. 2023. Online Fair Allocation of Perishable Resources. *SIGMETRICS Perform. Eval. Rev.* 51, 1 (2023), 55–56.

[11] Gerdus Benade, Aleksandr M Kazachkov, Ariel D Procaccia, Alexandros Psomas, and David Zeng. 2024. Fair and efficient online allocations. *Operations Research* 72, 4 (2024), 1438–1452.

[12] Umang Bhaskar, Neeldhara Misra, Aditi Sethia, and Rohit Vaish. 2023. The price of equity with binary valuations and few agent types. In *International Symposium on Algorithmic Game Theory*. Springer, 271–289.

[13] Eric Budish. 2011. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *J. of Polit. Econ.* 119, 6 (2011), 1061–1103.

[14] Ioannis Caragiannis, Christos Kaklamanis, Panagiotis Kanellopoulos, and Maria Kyropoulou. 2012. The efficiency of fair division. *Theor. Comput. Syst.* 50, 4 (2012), 589–610.

[15] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. 2019. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Trans. Econ. Comput.* 7, 3 (2019).

[16] Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn. 2024. EFX Exists for Three Agents. *J. ACM* 71, 1 (2024).

[17] Davin Choo, Winston Fu, Derek Khu, Tzeh Yuan Neoh, Tze-Yang Poon, and Nicholas Teh. 2025. Approximate Proportionality in Online Fair Division. arXiv:2508.03253

[18] Benjamin Cookson, Soroush Ebadian, and Nisarg Shah. 2025. Temporal fair division. *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI '25* 39, 13 (2025), 13727–13734.

[19] Edith Elkind, Alexander Lam, Mohamad Latifian, Tzeh Yuan Neoh, and Nicholas Teh. 2025. Temporal Fair Division of Indivisible Items. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '25*. 676–685.

[20] Vasilis Gkatzelis, Alexandros Psomas, and Xizhi Tan. 2021. Fair and Efficient Online Allocations with Normalized Valuations. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 6 (2021), 5440–5447.

[21] Paul W. Goldberg, Kasper Høgh, and Alexandros Hollender. 2023. The Frontier of Intractability for EFX with Two Agents. In *Algorithmic Game Theory - 16th International Symposium, SAGT 2023, Proceedings*. 290–307. https://doi.org/10.1007/978-3-031-43254-5_17

[22] Laurent Gourvès, Jérôme Monnot, and Lydia Tlilane. 2014. Near Fairness in Matroids. In *Proceedings of the Twenty-First European Conference on Artificial Intelligence, ECAI '14*. 393–398.

[23] Jiafan He, Ariel D. Procaccia, Alexandros Psomas, and David Zeng. 2019. Achieving a Fairer Future by Changing the Past. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI '19*. 343–349.

[24] Hadi Hosseini and Aditi Sethia. 2025. Equitable allocations of mixtures of goods and chores. *arXiv preprint arXiv:2501.06799* (2025).

[25] Ian Kash, Ariel D Procaccia, and Nisarg Shah. 2014. No agent left behind: Dynamic fair division of multiple resources. *J. Artif. Intell. Res.* 51 (2014), 579–603.

[26] Pooja Kulkarni, Ruta Mehta, and Parnian Shahkar. 2025. Online Fair Division: Towards Ex-Post Constant MMS Guarantees. In *Proceedings of the 26th ACM Conference on Economics and Computation, EC '25*. 638–638.

[27] Alexander Lindermayr and Nicole Megow. 2024. ALPS: Algorithms with Predictions. https://algorithms-with-predictions.github.io/.

[28] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. 2004. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce, EC '04*. 125–131.

[29] Thodoris Lykouris and Sergei Vassilvitskii. 2021. Competitive Caching with Machine Learned Advice. *J. ACM* 68, 4 (2021), 24:1–24:25.

[30] Michael Mitzenmacher and Sergei Vassilvitskii. 2022. Algorithms with predictions. *Commun. ACM* 65, 7 (2022), 33–35. https://doi.org/10.1145/3528087

[31] Tzeh Yuan Neoh, Jannik Peters, and Nicholas Teh. 2025. Online Fair Division with Additional Information. *arXiv preprint arXiv:2505.24503* (2025).

[32] Michael L. Pinedo. 2022. *Scheduling: Theory, Algorithms, and Systems* (6th ed.). Springer.

[33] Benjamin Plaut and Tim Roughgarden. 2020. Almost envy-freeness with general valuations. *SIAM J. Discrete Math.* 34, 2 (2020), 1039–1068.

[34] Jiaxin Song, Biaoshuai Tao, Wenqian Wang, and Yuhao Zhang. 2025. Online MMS Allocation for Chores. arXiv:2507.14039 [cs.GT] https://arxiv.org/abs/2507.14039

[35] Walter Stromquist. 1980. How to Cut a Cake Fairly. *Amer. Math. Monthly* 87, 8 (1980), 640–644. http://www.jstor.org/stable/2320951

[36] Yuanyuan Wang and Tianze Wei. 2026. Online fair allocations with binary valuations and beyond. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 17267–17275.

[37] Alexander Wei and Fred Zhang. 2020. Optimal robustness-consistency tradeoffs for learning-augmented online algorithms. *Advances in Neural Information Processing Systems* 33 (2020), 8042–8053.

[38] Shengwei Zhou, Rufan Bai, and Xiaowei Wu. 2023. Multi-agent online scheduling: MMS allocations for indivisible items. In *International Conference on Machine Learning*. 42506–42516.

A MISSING PROOFS FROM SECTION 3

A.1 Proof of Theorem 3.1

Proof. (Positive result) When the algorithm terminates, in the resulting allocation A, agent 1 will have value $x := v(A_1) \leq \varphi - 1$, by the algorithm's definition. Agent 2 will have value $y := v(A_2) = 1 - x \geq 2 - \varphi$. If $x \geq y$, then this a-EFX allocation has $a \geq \frac{y}{x} \geq \frac{2-\varphi}{\varphi-1} = \varphi - 1$. If $x < y$, then all goods of agent 2 have value strictly greater than $\varphi - 1 - x$, otherwise, at least one of them would have been given to agent 1 (since together with the new good agent 1 would have value at most $x + (\varphi - 1 - x) = \varphi - 1$). If agent 2 has a single good, then agent 1 does not EFX-envy agent 2, and agent 2 does not envy agent 1, since $y > x$, so this is an exact EFX. If agent 2 has at least two goods, then each has value greater than $\varphi - 1 - x$ as argued earlier, so $1 - x = y > 2(\varphi - 1 - x)$ which implies that $x > 2\varphi - 3$. Then, in this a-EFX allocation, $y > x$ so agent 2 does not envy agent 1, and also $v({}_{\mathrm{x}}A_2) < y - (\varphi - 1 - x) = 1 - x - (\varphi - 1 - x) = 2 - \varphi$. So, we have $a \geq \frac{2\varphi-3}{2-\varphi} = \varphi - 1$.

(Tightness) For the sake of contradiction, suppose there is an algorithm that guarantees an a-EFX for some $a \in (\varphi - 1, 1]$. Consider a rational value $\lambda \in [0, a - \varphi + 1)$. Let the adversary be giving to the algorithm goods of value $\varepsilon := \frac{\lambda}{4}$. Let the number of rounds be $T := \left\lceil \frac{2\varphi-3}{\varepsilon} \right\rceil + 3$, known by the algorithm. W.l.o.g., the first good is allocated to agent 1. Now consider the time $t \geq 2$ when a good is allocated to agent 2. There are two cases:

(i) If after that allocation, $v(A_1^t) =: x \leq 2\varphi - 3$, then the adversary gives a good of value $1 - x - \varepsilon$ to the algorithm (and notice that $v(A_2^t) = \varepsilon$). If the latter good is allocated to agent 1, then this is an a'-EFX allocation with $a' \leq \frac{\varepsilon}{1-2\varepsilon} = \frac{\lambda}{4-2\lambda} < a$, by definition of λ. This is a contradiction. If the latter good is allocated to agent 2, then this is an a'-EFX allocation with $a' \leq \frac{x}{1-x-\varepsilon} \leq \frac{2\varphi-3}{4-2\varphi-\varepsilon} < \frac{2\varphi-3}{4-2\varphi-(2-\varphi)} = \frac{2\varphi-3}{2-\varphi} = \varphi - 1 < a$, where the third inequality comes from the fact that $\varepsilon < \lambda \leq 2 - \varphi$. So in this case the algorithm cannot provide an a-EFX allocation for $a > \varphi - 1$.

(ii) Otherwise, there exists a time t' where $v(A_2^{t'}) = 0$ and $v(A_1^{t'}) =: x \in (2\varphi - 3, 2\varphi - 3 + \varepsilon]$. Then, the adversary gives consecutively two goods, each of value $\frac{1-x}{2}$. If both are allocated to agent 1, then this is obviously a 0-EFX, a contradiction. If both are allocated to agent 2, then agent 2 does not envy agent 1, but agent 1 has EFX-envy towards agent 2, and this a'-EFX allocation has $a' \leq \frac{x}{(1-x)/2} \leq \frac{2(2\varphi-3+\varepsilon)}{4-2\varphi-\varepsilon} \leq \frac{2(2\varphi-3)+\lambda/2}{4-2\varphi-\lambda/4} < \varphi - 1 + \lambda < a$, where the second to last inequality comes from the fact that $\lambda < a - \varphi + 1 \leq 2 - \varphi$. So this is not an a-EFX allocation. Finally, if one good is allocated to agent 1 and the other to agent 2, then agent 1 does not envy agent 2, but agent 2 has EFX-envy towards agent 1, and this a'-EFX allocation has $a' \leq \frac{(1-x)/2}{x+(1-x)/2-\varepsilon} = \frac{1-x}{1+x-2\varepsilon} \leq \frac{4-2\varphi}{2\varphi-2-2\varepsilon} = \frac{4-2\varphi}{2\varphi-2-\lambda/2} < \varphi - 1 + \lambda < a$, where again the second to last inequality comes from the fact that $\lambda < 2 - \varphi$. So the algorithm in this case cannot produce an a-EFX allocation. Therefore, for any fixed $a \in (\varphi - 1, 1]$, an a-EFX allocation is impossible to guarantee.

From the case analysis above, we see that the maximum number of rounds required is for case (ii), and it is $\left\lceil \frac{2\varphi-3+\varepsilon}{\varepsilon} \right\rceil + 2$. For all the cases that the algorithm might fall into, the adversary provides as many extra goods as needed to complete the known $T = \left\lceil \frac{2\varphi-3+\varepsilon}{\varepsilon} \right\rceil + 2$ rounds, where all these goods' values are set to 0 and do not affect the bounds on the approximation ratios. □

A.2 Proof of Theorem 3.2

Proof. Consider an algorithm that knows the true horizon $T = n + 1$ and does not use predictions. Let the first two goods that the adversary gives to the algorithm have value $\varepsilon := \frac{a}{3(n-1)}$. There are two cases:

(i) If the algorithm allocates the two goods to the same agent, say i, then the adversary provides a third good of value $1 - 2\varepsilon$, and another $n - 2$ goods of value 0.[5] Regardless of where the latter $n - 1$ goods are allocated, there is an agent with value 0 who a-EFX-envies agent i for any $a > 0$, so in this case, the algorithm does not guarantee an a-EFX allocation.

(ii) If the algorithm allocates the first two goods to two agents, say i, j, then the adversary provides $n - 1$ goods of value $\frac{1-2\varepsilon}{n-1}$. Then, by the pigeonhole principle, at least one of the agents will have at least two goods (one of which has value $\frac{1-2\varepsilon}{n-1}$), and at least one of them will have value at most ε. Therefore, one of the latter agents is a'-EFX free for $a' \leq \frac{\varepsilon}{(1-2\varepsilon)/(n-1)} < \frac{\varepsilon}{1/(2(n-1))} = 2\varepsilon(n-1) < a$, where the second inequality comes from the fact that $1 - 2\varepsilon = 1 - \frac{2a}{3(n-1)} > 1/2$ for any $a \leq 1$ and $n \geq 3$. Therefore, in this case too, the algorithm cannot give an a-EFX allocation. □

A.3 Proof of Theorem 3.3

Proof. For the sake of contradiction, suppose there is an algorithm that can provide an a-EFX allocation for some $a > 0$. The algorithm also knows that $T = \left\lfloor \frac{4}{a} \right\rfloor + 2$. We will denote the value of a good g as (ℓ, r), where $\ell := v_1(g)$ and $r := v_2(g)$. We fix an $\varepsilon := \frac{a}{4}$. Let the first good that the adversary gives to the algorithm be of value $(\varepsilon, 0)$. There are two cases:

(i) If the algorithm allocates the first good to agent 2, then the adversary keeps giving to the algorithm goods of value $(\varepsilon, 0)$ until the algorithm allocates a good to agent 1 (which might never happen, which means that after $t = T$, agent 1 has no goods, and this is a 0-EFX allocation). Let time $t \geq 2$ be the round in which either a good is allocated to agent 1 for the first time or when $t\varepsilon \in (1 - \varepsilon, 1]$. Observe that if the latter condition is the case, then no more goods of value $(\varepsilon, 0)$ can be produced by the adversary, otherwise, according to agent 1, the total value for all the goods produced would exceed the value of 1, thus defy the normalization. We know that $v_1(A_1^t) \leq \varepsilon$ and $v_1(A_2^t) \geq (t-1)\varepsilon$. Then, the adversary produces a good of value $(1 - t\varepsilon, 1)$. If the algorithm allocates the latter good to agent 2, then agent 2 does not EFX-envy agent 1, but agent 1 has EFX-envy towards agent 2, and the a'-EFX allocation has $a' \leq \frac{v_1(A_1^{t+1})}{v_1(A_2^{t+1})-\varepsilon} \leq \frac{\varepsilon}{(1-t\varepsilon)+(t-1)\varepsilon-\varepsilon} = \frac{\varepsilon}{1-2\varepsilon} \leq \frac{a}{2} < a$, where the penultimate inequality comes from the fact that $1 - 2\varepsilon = 1 - \frac{a}{2} \geq \frac{1}{2}$ for any $a \leq 1$. If the algorithm allocates the good to agent 1, then agent 2 has EFX-envy towards agent 1, and this can only be a 0-EFX allocation since $v_2(A_2^{t+1}) = 0$ and $v_2({}_{\mathrm{x}}A_1^{t+1}) = 1$. So in this case, the algorithm cannot provide an a-EFX allocation.

(ii) If the algorithm allocates the first good to agent 1, then the adversary keeps giving to the algorithm goods of value $(0, \varepsilon)$ until the algorithm allocates a good to agent 2 (which might never

[5]The latter $n - 2$ goods of value 0 are given so that the adversary fulfils the promise that $T = n + 1$, and do not affect the bounds on the EFX approximation ratio.

happen). Similarly to case (i), let time $t \geq 2$ be the round in which either a good is allocated to agent 2 for the first time or when $(t-1)\varepsilon \in (1-\varepsilon, 1]$. Observe again that in the latter condition, for a reason symmetric to that of case (i), no more goods of value $(0, \varepsilon)$ can be produced by the adversary. We know that $v_2(A_2^t) \leq \varepsilon$ and $v_2(A_1^t) \geq (t-2)\varepsilon$. Then, the adversary produces a good of value $(1-\varepsilon, 1-(t-1)\varepsilon)$. If the algorithm allocates the latter good to agent 1, then agent 1 does not EFX-envy agent 2, but agent 2 has EFX-envy towards agent 1, and the a'-EFX allocation has $a' \leq \frac{v_2(A_2^{t+1})}{v_2(A_1^{t+1})-0} \leq \frac{\varepsilon}{(1-(t-1)\varepsilon)+(t-2)\varepsilon} = \frac{\varepsilon}{1-\varepsilon} \leq \frac{a}{3} < a$, where the penultimate inequality comes from the fact that $1-\varepsilon = 1-\frac{a}{4} \geq \frac{3}{4}$ for any $a \leq 1$. If the algorithm allocates the good to agent 2, then agent 1 has EFX-envy towards agent 2, and the a'-EFX allocation has $a' \leq \frac{\varepsilon}{1-\varepsilon} < a$ as argued earlier. So in this case too, the algorithm cannot provide an a-EFX allocation.

From the case analysis above, we see that the maximum number of goods produced is $1 + t + 1 = \left\lfloor \frac{1}{\varepsilon} \right\rfloor + 2 = \left\lfloor \frac{4}{a} \right\rfloor + 2$, which is needed for case (ii). If the algorithm falls in any of the two cases and the aforementioned number of rounds has not been reached, the adversary provides the algorithm with an extra amount of goods until that number of rounds is achieved (in order to fulfill its promise that $T = \left\lfloor \frac{4}{a} \right\rfloor + 2$). All the extra goods have value $(0,0)$ and do not affect the bounds on the approximation ratios. □

B MISSING PROOFS FROM SECTION 4.2

B.1 Proof of Theorem 4.2

Proof. We will first prove the following stronger statement. Suppose we have $n \geq 2$ agents with additive, normalized valuations. Each agent $i \in [n]$ has accuracy $\eta_i \geq 1 - \frac{\tilde{a}_i - a_i}{(2n-2+\tilde{a}_i)(1+a_i)}$ for some given $a_i, \tilde{a}_i \in [0,1]$ with $a_i \leq \tilde{a}_i$. Given an allocation A^o in which every agent i has no $\tilde{a}_i$-EFX-envy towards any other agent according to $(p_i(g_t))_{t\in[T']}$, we can compute in polynomial time an allocation B in which every agent i has no a_i-EFX-envy towards any other agent according to her true valuation $(v_i(g_t))_{t\in[T]}$.

Let the allocation $A^o = (A_1^o, A_2^o \dots, A_n^o)$ be the output of the statement's algorithm when its input is the prediction vector $(p_{i,t})_{i\in[n],t\in[T']}$. Therefore in A^o each agent $i \in [n]$ is not $\tilde{a}_i$-EFX-envious towards any other agent according to the prediction values. We will first turn A^o into an allocation A in which there is at least one agent $k \in [n]$ that is not envied by any other agent (if this is not already the case). We define the "envy-graph" of A^o; this is a directed graph whose nodes are the agents, and an edge (i,j) exists if i envies j. If the graph has a source, then this source is our required k and we are done. If the graph has no source, it contains a cycle $i_1, i_2, \dots, i_r, i_1$ of size r for some $r \in \{2, 3 \dots, n\}$. We can eliminate this cycle by rotating the bundles, i.e., $A_{i_s}^o \leftarrow A_{i_{s+1}}^o$ for all $s \in [r]$, where $i_{r+1} := i_1$. Notice that by rotating the bundles, we have eliminated $r \geq 2$ edges of the graph and did not create any edges, since the bundles of A^o remained the same even though they changed owners. By eliminating cycles repeatedly until there are no more left, we derive an allocation A whose envy-graph is acyclic and therefore has a source, that is, the required agent k who is not envied by any other agent. Finally, since the edges of the initial envy-graph were at most $n(n-1)$ and in each cycle elimination we strictly reduced the number of edges by 2, allocation A will be found after at most $n(n-1)/2$ eliminations. Creating the envy-graph of A^o and each of its updates takes time polynomial in n and T, so computing A can be done in polynomial time.

Now let us focus on an arbitrary agent i under allocation A. We denote $x_{i*} := p_i(A_i)$, and $x_{i,j} := p_i({}_{\mathrm{x}}A_j)$ for $j \neq i$. Since agent i is not $\tilde{a}_i$-EFX-envious towards any other agent, we have

$$x_{i*} \geq \tilde{a}_i \cdot x_{ij}, \quad \text{for all } j \neq i. \tag{1}$$

We also know that for any $i \neq j$, $x_{ij} \geq \frac{p_i(A_j)}{2}$: If $A_j = \emptyset$ then this is obviously true, and if $A_j \neq \emptyset$ then, by setting $g \in \arg\min_{g\in A_j} p_i(A_j)$, we have $p_i(g) \leq \min_{g'\in A_j\setminus\{g\}}\{p_i(g')\} \leq x_{ij}$, so $p_i(A_j) = p_i(g) + x_{ij} \leq 2 \cdot x_{ij}$ for any $j \neq i$. Finally, by the normalization assumption, we have $1 = \sum_{j\in[n]} p_i(A_j) \leq x_{i*} + (n-1)\cdot\frac{2}{\tilde{a}_i}x_{i*}$, or equivalently, $x_{i*} \geq \frac{\tilde{a}_i}{2n-2+\tilde{a}_i}$.

Now we modify A to get allocation B in the following way, depending on how the true time horizon T compares to the predicted horizon T'.

$$B_i = \begin{cases} A_i \setminus \{g_{T+1}, \dots, g_{T'}\}, & \text{if } T \leq T', \text{ for all } i \in [n] \\ A_i, & \text{if } T > T', \text{ for all } i \neq k \\ A_i \cup \{g_{T'+1}, \dots, g_T\}, & \text{if } T > T', \text{ for } i = k. \end{cases}$$

In other words, if the number of goods that arrive is at most that of the prediction, then the goods that arrive are allocated according to the prediction. If more goods arrive than those predicted, then they are placed in the bundle of agent k who was not envied by anyone in allocation A.

Consider the true values' distribution $(v_{i,t})_{i\in[n],t\in[T]}$, and similarly to the previous notation, let us denote $y_{i*} := v_i(B_i)$, and $y_{i,j} := v_i({}_{\mathrm{x}}B_j)$ for $j \neq i$. For some fixed $a_i \in [0,1]$ with $a_i \leq \tilde{a}_i$, recall that $(v_{i,t})_{i\in[n],t\in[T]}$ has TV distance $d_i \leq \frac{\tilde{a}_i - a_i}{(2n-2+\tilde{a}_i)(1+a_i)}$ from $(p_{i,t})_{i\in[n],t\in[T']}$. Notice that $v_i(B_i) \geq x_{i*} - d_i$, so $y_{i*} \geq x_{i*} - d_i$. Also, for every $j \in [n] \setminus \{k\}$ we have $y_{ij} \leq x_{ij} + d_i \leq \frac{x_{i*}}{\tilde{a}_i} + d_i$, where the last inequality comes from Eq. (1). Finally, since agent k is not envied by any other agent in allocation A, we have $x_{i*} \geq p_i(A_k)$. We then have $y_{ik} \leq p_i(A_k) + d_i \leq x_{i*} + d_i \leq \frac{x_{i*}}{\tilde{a}_i} + d_i$, where the last inequality is due to the fact that $\tilde{a}_i \leq 1$.

Then in B, agent i is not a_i'-EFX-envious towards any other agent for

$$\begin{aligned} a_i' = \frac{y_{i*}}{\max_{i\in[n]}\{y_{ij}\}} &\geq \frac{x_{i*} - d_i}{x_{i*}/\tilde{a}_i + d_i} \\ &\geq \frac{\tilde{a}_i - (2n-2+\tilde{a}_i)d_i}{1 + (2n-2+\tilde{a}_i)d_i} \geq \frac{\tilde{a}_i - (\tilde{a}_i - a_i)/(1+a_i)}{1 + (\tilde{a}_i - a_i)/(1+a_i)} = a_i. \end{aligned}$$

We have proven that given an allocation A^o in which every agent i has no $\tilde{a}_i$-EFX-envy towards any other agent according to $(p_i(g_t))_{t\in[T']}$, we can compute in polynomial time an allocation B in which every agent i has no a_i-EFX-envy towards any other agent according to her true valuation $(v_i(g_t))_{t\in[T]}$. The theorem's statement now follows immediately. □

C MISSING PROOFS FROM SECTION 4.3

C.1 Proof of Theorem 4.5

Proof. Consider an arbitrary $a \in \left(\frac{1}{2}, 1\right]$ and an a-EFX algorithm with prediction accuracy $\eta < 1 - \frac{1-a}{\min\{6a,4\}}$. The adversary fixes a rational number $\lambda \in \begin{cases} \left[0, \frac{2a-1}{6a}\right), & \text{if } a \in \left(\frac{1}{2}, \frac{2}{3}\right] \\ \left(\frac{1-a}{4}, \frac{a}{8}\right), & \text{if } a \in \left(\frac{2}{3}, 1\right] \end{cases}$, and sets $\varepsilon =$

$\begin{cases} \frac{1}{6} - \lambda, & \text{if } a \in \left(\frac{1}{2}, \frac{2}{3}\right] \\ \lambda, & \text{if } a \in \left(\frac{2}{3}, 1\right] \end{cases}$. Notice that for both $a \in \left(\frac{1}{2}, \frac{2}{3}\right]$ and $a \in \left(\frac{2}{3}, 1\right]$, we have

$$\varepsilon \in \left(\left(\frac{1}{2} - \lambda\right) \cdot \frac{1-a}{1+a}, \quad \frac{a}{2(2-a)} - \lambda \cdot \frac{2+a}{2-a} \right) \tag{2}$$

due to the domain of λ. The adversary also provides the following prediction to the algorithm for the two agents: $p_1(g_1) = 2\varepsilon, p_1(g_2) = 2\lambda, p_1(g_3) = \frac{1}{2} - 2\varepsilon - \lambda, p_1(g_4) = \frac{1}{2} - \lambda$, and $p_2(g_1) = 2\lambda, p_2(g_2) = 2\varepsilon, p_2(g_3) = \frac{1}{2} - 2\varepsilon - \lambda, p_2(g_4) = \frac{1}{2} - \lambda$. The adversary then reveals the true values $v_1(g_1) = 2\varepsilon, v_1(g_2) = 2\lambda$, and $v_2(g_1) = 2\lambda, v_2(g_2) = 2\varepsilon$. There are four cases:

(i) If g_1 and g_2 get allocated to agent 1, then the adversary reveals true values $v_1(g_3) = \frac{1}{2} - 3\varepsilon - \lambda, v_1(g_4) = \frac{1}{2} + \varepsilon - \lambda$, and $v_2(g_3) = \frac{1}{2} - \varepsilon - \lambda$, $v_2(g_4) = \frac{1}{2} - \varepsilon - \lambda$. Then, if both g_3, g_4 get allocated to agent 1, then agent 2 is EFX-envious towards agent 1, and this is a 0-EFX allocation. If both goods get allocated to agent 2, then agent 1 is EFX-envious to agent 2 and this is an a'-EFX allocation with $a' = \frac{2\varepsilon + 2\lambda}{1/2 + \varepsilon - \lambda} < a$, where the inequality comes from the upper bound of ε in Eq. (2). If g_3 gets allocated to agent 1 and g_4 to agent 2, then this is an a'-EFX allocation with $a' \leq \frac{1/2 - \varepsilon - \lambda}{1/2 + \varepsilon - \lambda} < a$, where the inequality holds for all the domain of a due to the lower bound of ε in Eq. (2).

(ii) If g_1 and g_2 get allocated to agent 2, then the adversary reveals true values $v_1(g_3) = \frac{1}{2} - \varepsilon - \lambda, v_1(g_4) = \frac{1}{2} - \varepsilon - \lambda$, and $v_2(g_3) = \frac{1}{2} - 3\varepsilon - \lambda$, $v_2(g_4) = \frac{1}{2} + \varepsilon - \lambda$. The analysis is symmetric to that of case (i) and shows that all allocations induce an a'-EFX allocation for $a' < a$.

(iii) If g_1 gets allocated to agent 1 and g_2 gets allocated to agent 2, then the adversary reveals true values $v_1(g_3) = \frac{1}{2} - 3\varepsilon - \lambda$, $v_1(g_4) = \frac{1}{2} + \varepsilon - \lambda$, and $v_2(g_3) = \frac{1}{2} - 3\varepsilon - \lambda$, $v_2(g_4) = \frac{1}{2} + \varepsilon - \lambda$. If both g_3 and g_4 get allocated to the same agent, w.l.o.g. agent 2, then agent 1 is EFX-envious towards agent 2, and $v_1(\{g_2, g_3, g_4\}) - \min\{v_1(g_2), v_1(g_3), v_1(g_4)\} \leq v_1(\{g_2, g_4\}) = \frac{1}{2} + \varepsilon + \lambda$. So this is an a'-EFX allocation with $a' \leq \frac{2\varepsilon}{1/2 + \varepsilon + \lambda} < a$, which holds due to the fact that $\varepsilon < \frac{a}{2(2-a)} - \lambda \cdot \frac{2+a}{2-a} \leq \left(\frac{1}{2} + \lambda\right) \frac{a}{2-a}$ for $\lambda \geq 0$, where the first inequality comes from Eq. (2). If the two last goods get allocated to distinct agents, w.l.o.g., g_3 gets allocated to agent 1 and g_4 to agent 2, then agent 1 is EFX-envious towards agent 2, and $v_1(\{g_2, g_4\}) - \min\{v_1(g_2), v_1(g_4)\} \leq v_1(g_4) = \frac{1}{2} + \varepsilon - \lambda$. This is an a'-EFX allocation with $a' \leq \frac{1/2 - \varepsilon - \lambda}{1/2 + \varepsilon - \lambda} < a$, where the inequality holds for all the domain of a due to the lower bound of ε in Eq. (2).

(iv) If g_1 gets allocated to agent 2 and g_2 gets allocated to agent 1, then the adversary reveals the same true values as in case (iii). Here we simply use the fact that $\varepsilon \geq \lambda$: this is obvious when $a \in \left(\frac{2}{3}, 1\right]$, and for the case where $a \in \left(\frac{1}{2}, \frac{2}{3}\right]$, we have $\lambda \leq \frac{2a-1}{6a} \leq \frac{1}{12}$, which implies, $\varepsilon = \frac{1}{6} - \lambda \geq \frac{1}{12} \geq \lambda$. Therefore, agents 1 and 2 now have bundles that are worth to them at most as much as they were worth in case (iii), and so if in case (iii) there was no a-EFX allocation, the same holds here.

Finally, notice that the error between the prediction and the true values in all the above cases is ε: if $a \in \left(\frac{1}{2}, \frac{2}{3}\right]$, then $\varepsilon = \frac{1}{6} - \lambda$, and λ can take any value strictly smaller than $\frac{2a-1}{6a}$, so ε can take any value strictly greater than $\frac{1-a}{6a}$; if $a \in \left(\frac{2}{3}, 1\right]$, then $\varepsilon = \lambda$, and λ can take any value strictly greater than $\frac{1-a}{4}$. □

C.2 Proof of Theorem 4.7

Before presenting the proof of Theorem 4.7, we need to present two lemmas. Specifically Lemma C.1 and Lemma C.2, where the former dominates for small values of a, and the latter for large values of a. Theorem 4.7 combines them into a single statement.

Lemma C.1. *Suppose we have $n \geq 3$ agents with additive, identical, normalized valuations, with a provided prediction of accuracy $\eta < 1 - \frac{1}{2(n-1+2a)}$ for some given $a \in (0, 1]$, that is, the error between the prediction and the true valuation is $1 - \eta > \frac{1}{2(n-1+2a)}$. Then, there is no algorithm that guarantees an a-EFX allocation, even when $T' = T = n+1$, and the predictions and the true valuations are 3-value functions.*

Proof. For the sake of contradiction, suppose there is an algorithm that can provide an a-EFX allocation for some $a \in (0, 1]$. The algorithm knows that $T = n + 1$ and has accuracy $\eta < 1 - \frac{1}{2(n-1+2a)}$. The adversary fixes a rational value $\varepsilon \in \left(0, \frac{a}{n-1+2a}\right)$, and gives to the algorithm the prediction: $p(g_1) = p(g_2) = \varepsilon$, $p(g_t) = \frac{2n-3}{(n-1)(n-2)} \left(\frac{1}{2} - \varepsilon\right)$ for $t \in \{3, 4, \ldots, n\}$, and $p(g_{n+1}) = \frac{1}{n-1} \left(\frac{1}{2} - \varepsilon\right)$. The adversary reveals the true values of the first two goods to the algorithm, namely $v(g_1) = v(g_2) = \varepsilon$. There are two cases:

(i) If the algorithm allocates the two goods to the same agent, say i, then the adversary provides $n-2$ goods of true value $p(g_t) = \frac{1-2\varepsilon}{n-2}$ for $t \in \{3, 4, \ldots, n\}$, and a final good of true value 0.[6] Since the goods of positive value are n and two of them are allocated to the same agent, regardless of where the goods $g_3, \ldots, g_{n+1}$ are allocated, there is an agent with value 0. Recall now that agent i has two goods, each of value $\varepsilon > 0$, so this is a 0-EFX allocation, a contradiction to the assumption that $a > 0$.

(ii) If the algorithm allocates the first two goods to two agents, say i, j, then the adversary provides $n - 1$ goods of true value $v(g_t) = \frac{1-2\varepsilon}{n-1}$ for $t \in \{3, 4, \ldots, n + 1\}$. Notice that $\frac{1-2\varepsilon}{n-1} > \varepsilon$ for any $a \leq 1$ by definition of ε, therefore, any of $g_3, \ldots, g_{n+1}$ is more valuable than g_1 and g_2. By the pigeonhole principle, at least one of the agents will have at least two goods (one of which has value $\frac{1-2\varepsilon}{n-1}$), and at least one of them will have value at most ε. Notice also that, w.l.o.g., agent i will be allocated (at least) one of $\{g_1, g_2\}$ and one of $\{g_3, \ldots, g_{n+1}\}$. So, one of the agents is EFX-envious towards i and the allocation is a'-EFX for $a' \leq \frac{\varepsilon}{(1-2\varepsilon)/(n-1)} < a$, where the last inequality comes from the upper bound of ε. Therefore, in this case too, the algorithm cannot provide an a-EFX allocation, a contradiction.

Finally, the error between the prediction and the two scenarios of true values of the adversary is the following. For case (i) it is

$$\frac{1}{2} \cdot \left(0 + 0 + (n-2) \cdot \left[\frac{1-2\varepsilon}{n-2} - \frac{(2n-3)(1/2 - \varepsilon)}{(n-1)(n-2)}\right]\right) + \\ + \frac{1}{2} \cdot \left[\frac{1/2 - \varepsilon}{n-1} - 0\right] = \frac{1}{n-1}\left(\frac{1}{2} - \varepsilon\right),$$

[6] The latter good of value 0 is given so that the adversary fulfills the promise that $T = n + 1$, and does not affect the bounds on the EFX approximation ratio.

and for case (ii) it is

$$\frac{1}{2}\cdot\left(0+0+(n-2)\cdot\left[\frac{(2n-3)(1/2-\varepsilon)}{(n-1)(n-2)}-\frac{1-2\varepsilon}{n-1}\right]\right)$$
$$+\frac{1}{2}\cdot\left[\frac{1-2\varepsilon}{n-1}-\frac{1/2-\varepsilon}{n-1}\right]=\frac{1}{n-1}\left(\frac{1}{2}-\varepsilon\right),$$

for any $\varepsilon \in \left(0, \frac{a}{n-1+2a}\right)$ of the adversary's choice. □

Lemma C.2. *Suppose we have $n \geq 3$ agents with additive, identical, normalized valuations, with a provided prediction of accuracy $\eta < 1 - \frac{1-a^2}{4+(2n-3)a}$ for some given $a \in (0, 1]$, that is, the error between the prediction and the true valuation is $1 - \eta > \frac{1-a^2}{4+(2n-3)a}$. Then, there is no algorithm that guarantees an a-EFX allocation, even when $T' = T = 2n - 1$, and the predictions and the true valuations are 3-value functions.*

Proof. For the sake of contradiction, suppose there is an algorithm that can provide an a-EFX allocation for some $a \in (0, 1]$. The algorithm knows that $T = 2n-1$ and has accuracy $\eta < 1 - \frac{1-a^2}{4+(2n-3)a}$. The adversary fixes rational values $k \in \left(0, \frac{a}{4+(2n-3)a}\right)$ and $\varepsilon \in \left(\frac{1-a^2}{4+(2n-3)a}, \frac{1}{4+(2n-3)a}\right]$, and gives to the algorithm the prediction: $p(g_1) = p(g_2) = \cdots = p(g_{2n-3}) = k$, $p(g_{2n-2}) = \frac{1-(2n-3)k}{2} - \varepsilon$, and $p(g_{2n-1}) = \frac{1-(2n-3)k}{2} + \varepsilon$. The adversary reveals the true values of the first $2n-3$ goods to the algorithm, namely $v(g_1) = v(g_2) = \cdots = v(g_{2n-3}) = k$. For ease of presentation, a good g_t for $t \in [2n-3]$ will be called a "k-good". After time-step $t = 2n-3$ there are three cases:

(i) Exactly one agent, w.l.o.g. agent n, does not have any k-good. Then, there is at least another agent, w.l.o.g. agent $n-1$, that has at most 1 k-good (and therefore, exactly 1 k-good); otherwise, at least $n-1$ agents have at least two k-goods, so we would have at least $2(n-1) = 2n-2 > 2n-3$ k-goods, a contradiction. Also, there is at least another agent, w.l.o.g. agent $n-2$, that has at most 2 k-goods; otherwise, we would have at least $n-2$ agents with at least 3 k-goods, and agent $n-1$ with exactly 1 k-good, therefore $3(n-2)+1 = 3n-5 > 2n-3$ k-goods in total, a contradiction. Consequently, the adversary reveals the true values $v(g_{2n-2}) = v(g_{2n-1}) = \frac{1-(2n-3)k}{2}$.

If neither g_{2n-2} nor g_{2n-1} gets allocated to agent n, then this is a 0-EFX allocation, so the algorithm fails to produce an a-EFX allocation for the required value of a. If one of these goods gets allocated to agent n and the other gets allocated to an agent with exactly one k-good, w.l.o.g. agent $n-1$, then one of the agents with at most 2 k-goods, such as agent $n-2$, is EFX-envious towards agent $n-1$, and due to that, this is an a'-EFX allocation with

$$a' \leq \frac{2k}{(1-(2n-3)k)/2} = \frac{4}{1/k-(2n-3)}$$
$$< \frac{4}{4/a+(2n-3)-(2n-3)} = a, \tag{3}$$

where the last inequality comes from the fact that $k < \frac{a}{4+(2n-3)a}$. If one of the goods gets allocated to agent n and the other one gets allocated to an agent with more than one k-goods, then agent $n-1$ EFX-envies that agent and this is an a'-EFX allocation with $a' \leq \frac{k}{(1-(2n-3)k)/2} \leq \frac{2k}{(1-(2n-3)k)/2} < a$, where the last inequality comes from Eq. (3). If both get allocated to agent n, then agent $n-1$ is EFX-envious towards her, and due to that, this is an a'-EFX allocation with

$$a' \leq \frac{k}{(1-(2n-3)k)/2} \leq \frac{2k}{(1-(2n-3)k)/2} < a,$$

where again, the last inequality comes from Eq. (3). So, the algorithm fails to produce an a-EFX allocation.

(ii) There are two agents, w.l.o.g. agents $n-1$ and n, that have no k-good. If there are at least three agents with no k-good then no matter where g_{2n-2}, g_{2n-1} get allocated, this will be a 0-EFX allocation; because at least one agent will have no goods, and there will be at least one agent with 2 k-goods (which have positive value), since we have more than $n-3$ goods. So the remaining case is when only agents $n-1$ and n have no k-good. Then, there is at least one agent, w.l.o.g. agent $n-2$, with at least 3 k-goods; otherwise, $n-2$ agents would have at most 2 k-goods, so the total number of such goods would be at most $2(n-2) = 2n-4 < 2n-3$, a contradiction. Now the adversary reveals the true values $v(g_{2n-2}) = \frac{1-(2n-3)k}{2} - 2\varepsilon$ and $v(g_{2n-1}) = \frac{1-(2n-3)k}{2} + 2\varepsilon$. If after the algorithm allocates g_{2n-2}, g_{2n-1}, some agent among agents $n-1$ and n has not received any good, then this is a 0-EFX allocation, since another agent exists with at least 2 k-goods. If each of agents $n-1$ and n receives one of the last two goods, w.l.o.g. let agent $n-1$ receive g_{2n-2}, and agent n receive g_{2n-1}, then the latter EFX-envies agent $n-2$ who has 3 k-goods. This will be an a'-EFX allocation with $a' \leq \frac{(1-(2n-3)k)/2-2\varepsilon}{2k} < \frac{4ak/2}{2k} = a$, where the last inequality comes from the fact that $\varepsilon > \frac{1-a^2}{4+(2n-3)a} > \frac{1-(2n-3+4a)k}{4}$. Therefore, the algorithm fails to output an a-EFX allocation.

(iii) Every agent has at least one k-good. Then, there are at least three agents, w.l.o.g. agents $n-2$, $n-1$, and n, that have exactly one k-good; otherwise, at least $n-2$ agents would have at least 2 such goods each, and 2 agents would have exactly 1 k-good, so the total number of k-goods would be at least $2(n-2)+2 = 2n-2 > 2n-3$, a contradiction. The adversary reveals the true values $v(g_{2n-2}) = \frac{1-(2n-3)k}{2} - 2\varepsilon$ and $v(g_{2n-1}) = \frac{1-(2n-3)k}{2} + 2\varepsilon$. No matter where these last two goods get allocated, the agent that will receive g_{2n-1} will also have at least one k-good, and at least one agent, w.l.o.g. agent n, will only have a single k-good. Therefore, the latter agent will EFX-envy the former and this will be an a'-EFX allocation for $a' \leq \frac{k}{(1-(2n-3)k)/2+2\varepsilon} \leq \frac{2k}{(1-(2n-3)k)/2} < a$, where the last inequality is due to Eq. (3). So, the algorithm cannot provide an a-EFX allocation.

Finally, notice that the error between the prediction and the true valuation in this instance equals ε, which can take any value in $\left(\frac{1-a^2}{4+(2n-3)a}, \frac{1}{4+(2n-3)a}\right]$. □

We are ready now to present the proof of Theorem 4.7

Proof. If $\frac{1}{2(n-1+2a)} \leq \frac{1-a^2}{4+(2n-3)a}$, then the adversary provides the algorithm with the instance specified in the proof of Lemma C.1, and produces $n-2$ dummy goods $g_{n+2}, g_{n+3}, \ldots, g_{2n-1}$ of value 0 to fulfill the promise that $T = 2n-1$.[7] If $\frac{1}{2(n-1+2a)} > \frac{1-a^2}{4+(2n-3)a}$, then the adversary provides the algorithm with the instance specified in the proof of Lemma C.2. □

[7]Notice that these extra goods do not affect the analysis since they can be allocated to an agent that is not envied.

C.3 Proof of Theorem 4.8

Proof. Algorithm 2 first uses LPT (Algorithm 1) to find an exact EFX allocation, called A. Then, A is classified according to its *form*, that is, some important properties of the two bundles A_1, A_2. Finally, a form-specific routine is used, which, at each time-step decides the recipient of the good in constant time, resulting in an a-EFX allocation B.

Let us denote the maximum error between the prediction and the true valuation by $d_{\max} := \frac{(4+a-a^2)(1-a)}{(2+a)(5-a)(1+a)}$, for $a \in (\varphi - 1, 1]$. In the trivial case where $a = 1$, $d_{\max} = 0$, the true values are identical to the predicted values (with the exception of additional goods that might arrive with true value 0 which do not affect the approximation factor the solution), therefore Algorithm 2 outputs an 1-EFX allocation A. In what follows, we will be considering $a \in (\varphi - 1, 1)$. W.l.o.g., let agent 1 have at most as much value as agent 2 in A, i.e., $p(A_1) \leq p(A_2)$, and let us denote by g an arbitrary good in $\arg\min_{g' \in A_2} p(g')$. First, observe that, since A has been produced by LPT, $p(A_1) \geq p(A_2 \setminus g)$. Therefore, $p(A_1) \geq 1/3$, otherwise, $p(A_1) < 1/3$ implies that $p(A_2/g) \leq p(A_1) < 1/3$, and so, $p(g) \leq p(A_2 \backslash g) < 1/3$. This means that $p(A_1)+p(A_2\backslash g)+p(g) < 1$, which contradicts the normalization condition.

The following is a preliminary result for the case where the prediction consists of at most three goods, and will be useful in our main algorithm (Algorithm 2).

Lemma C.3. *Suppose we have 2 agents with additive, identical, normalized valuations, with a provided predicted horizon $T' \leq 3$ and accuracy $\eta \geq 1 - \frac{1-a}{1+a}$ for some given $a \in [0,1]$, that is, the error between the prediction and the true valuation is $1 - \eta \leq \frac{1-a}{1+a}$. Then, Algorithm 3 guarantees an a-EFX allocation. Furthermore, if the algorithm knows that the predicted horizon is the true one (i.e., $T = T'$), then it guarantees an exact EFX even for prediction accuracy 0 (or equivalently, error 1).*

Proof. Suppose we have an instance and algorithm as described in the first part of the statement, i.e., a predicted horizon $T' \leq 3$ and accuracy $\eta \geq 1 - \frac{1-a}{1+a}$ for some given $a \in [0,1]$. Interestingly, the algorithm does not require a prediction vector. Essentially, it does the following. It starts by placing the first good (g_1) to agent 1. If there is no other round, then this is an exact EFX allocation; if there is another round, $t = 2$, the good g_2 arrives and the algorithm now assumes that a third one (g_3) will have the entire remaining value of $1 - v(g_1) - v(g_2)$. It then calculates which one has the greatest true value by comparing $v(g_1), v(g_2)$ and $1 - v(g_1) - v(g_2)$.

- If $\max\{v(g_1), v(g_2), 1 - v(g_1) - v(g_2)\} \neq 1 - v(g_1) - v(g_2)$, then $\max\{v(g_1), v(g_2), 1 - v(g_1) - v(g_2)\}$ equals either $v(g_1)$ or $v(g_2)$. Then g_2 is allocated to B_2 so that the highest-valued good remains alone in its bundle. Then, g_3 is allocated to the agent with current lowest value (breaking ties arbitrarily). If $v(g_2) = 1 - v(g_1)$, then even if the true horizon is $T \geq 3$, $v(g_t) = 0$ for all $t \in \{3, 4, \ldots, T\}$, and all goods g_t will be placed to the agent with lowest value between $v(g_1), v(g_2)$, which will be an exact EFX allocation.
- If $\max\{v(g_1), v(g_2), 1 - v(g_1) - v(g_2)\} = 1 - v(g_1) - v(g_2)$, then this means that $1 - v(g_1) - v(g_2) > 0$, or equivalently, $v(g_2) < 1 - v(g_1)$ and, as mentioned earlier, the algorithm assumes that g_3 has value $1 - v(g_1) - v(g_2)$. Then it allocates g_2 to B_1 (which now joins g_1), and at $t = 3$ it allocates g_3 to B_2. If at $t = 3$ we have $v(g_3) = 1 - v(g_1) - v(g_2)$, then any goods g_t for $t \in \{4, 5, \ldots, T\}$ that will potentially arrive have $v(g_t) = 0$ and will be placed to the least valued bundle, creating an exact EFX allocation.

Up to this point, we have proven the second part of the statement. However, notice that it might be the case that at $t = 3$, we have $v(g_3) < 1 - v(g_1) - v(g_2)$, or equivalently, $v(g_1) + v(g_2) + v(g_3) < 1$, which means that among the aforementioned goods g_t, for $t \in \{4, 5, \ldots, T\}$, there are some with positive value. The algorithm places all goods g_t to agent $i \in [2]$ who has the least valued bundle after $t = 3$, that is, B_i^3. This means that $v(B_i^{T'}) \leq v(B_{3-i}^{T'})$ holds. There are two cases:

(i) $v(B_i^T) \leq v(B_{3-i}^T)$. Therefore, agent $3-i$ does not envy agent i. Also, $v(B_i^T) = v(B_i^{T'}) + 1 - v(g_1) - v(g_2) - v(g_3) \geq v({}_{\mathrm{x}}B_{3-i}^{T'}) = v({}_{\mathrm{x}}B_{3-i}^T)$, where the first equality comes from the fact that all goods g_t, for $t \in \{4, 5, \ldots, T\}$, will be allocated to $B_i^{T'}$, and their cumulative value is $1 - v(g_1) - v(g_2) - v(g_3)$; the first inequality comes from the fact that g_1 and g_2 were placed by the algorithm such that if $v(g_3) = 1 - v(g_1) - v(g_2)$ then $v(B_i^{T'}) \geq v({}_{\mathrm{x}}B_{3-i}^{T'})$, but as discussed above, $v(g_3)$ can have at most some discrepancy $1 - v(g_1) - v(g_2) - v(g_3)$ which remains to come from goods $g_t, t \in \{4, 5, \ldots, T\}$; however, notice that in case g_3 is placed at $B_i^{T'}$, the aforementioned goods g_t will join that bundle, restoring the remaining value, i.e., $v(B_i^{T'}) + 1 - v(g_1) - v(g_2) - v(g_3) \geq v({}_{\mathrm{x}}B_{3-i}^{T'})$; finally, the last equality comes from the fact that B_{3-i} received no more goods after $t = T'$. Therefore, this is an exact EFX allocation.

(ii) $v(B_i^T) > v(B_{3-i}^T)$. Therefore, agent i does not envy agent $3-i$. As mentioned in the previous case, g_3 might have some discrepancy $1 - v(g_1) - v(g_2) - v(g_3)$ in its value. By the statement's assumption, this discrepancy is upper bounded by $\frac{1-a}{1+a}$, therefore we have $v(B_i^{T'}) + v(B_{3-i}^{T'}) = v(g_1) + v(g_2) + v(g_3) \geq 1 - \frac{1-a}{1+a}$. Since $v(B_i^{T'}) \leq v(B_{3-i}^{T'})$, we have $v(B_{3-i}^{T'}) \geq \frac{1}{2} - \frac{1}{2} \cdot \frac{1-a}{1+a}$. Therefore, this is an a'-EFX allocation with $a' = \frac{v(B_{3-i}^T)}{v({}_{\mathrm{x}}B_i^T)} \geq \frac{v(B_{3-i}^{T'})}{v(B_i^{T'})+(1-a)/(1+a)} \geq \frac{1/2-(1-a)/(2(1+a))}{1/2+(1-a)/(2(1+a))} = a$, where the first inequality comes from the fact that $v({}_{\mathrm{x}}B_i^T) \leq v(B_i^T)$, and $v(B_i^T)$ is at most $v(B_i^{T'})$ plus extra value $1-v(g_1)-v(g_2)-v(g_3) \leq \frac{1-a}{1+a}$ from the goods $g_t, t \in \{4, 5, \ldots, T\}$ received.

Finally, notice that the above arguments did not use any predicted values for the goods, only the prediction that the total value to arrive will be contained within the first three goods. □

Corollary C.4. *Suppose we have 2 agents with additive, identical, normalized valuations, with a provided prediction such that $\sum_{t=4}^{T'} p(g_t) \leq \frac{1-a}{1+a} - D$ and accuracy $\eta \geq 1 - D$ for some given $a \in [0,1]$ and $D \in \left[0, \frac{1-a}{1+a}\right]$, that is, the error between the prediction and the true valuation is $1 - \eta \leq D$. Then, Algorithm 3 guarantees an a-EFX allocation.*

Proof. From the proof of Lemma C.3, we can see that Algorithm 3 does not need any predicted values to allocate the first three goods, and what prevents the approximation of the EFX allocation from being 1 is the discrepancy that (virtual) values $p(g_4) = p(g_5) = \cdots = p(g_T) = 0$ have with respect to the true

Algorithm 3 Two agents with additive, identical valuations, and predicted horizon $T' \leq 3$: Computing an a-EFX for $a \in [0,1]$ when prediction accuracy is at least $1 - \frac{1-a}{1+a}$

Require: A value $T' \in [3]$ of predicted horizon.
Ensure: An a-EFX allocation B.

//time-step $t = 1$
$(B_1, B_2) \leftarrow (\{g_1\}, \emptyset)$ //g_1 is allocated to an agent, w.l.o.g., agent 1
//time-step $t = 2$
if $\min\{T', T\} \geq 2$ **then**
  if $\max\{v(g_1), v(g_2), 1 - v(g_1) - v(g_2)\} \neq 1 - v(g_1) - v(g_2)$ **then**
    $B_2 \leftarrow B_2 \cup \{g_2\}$
  else
    $B_1 \leftarrow B_1 \cup \{g_2\}$
//time-step $t = 3$
if $\min\{T', T\} \geq 3$ **then**
  if $v(B_1) \geq v(B_2)$ **then**
    $B_2 \leftarrow B_2 \cup \{g_3\}$
  else
    $B_1 \leftarrow B_1 \cup \{g_3\}$
//time-step $t = T'+1, T'+2, \ldots, T$, in case $T > T'$
for $t \in \{T'+1, T'+2, \ldots, T\}$ **do**
  if $v(B_1^{T'}) \leq v(B_2^{T'})$ **then**
    $B_1 \leftarrow B_1 \cup \{g_t\}$
  else
    $B_2 \leftarrow B_2 \cup \{g_t\}$

ones $v(g_4), v(g_5), \ldots, v(g_T)$. In particular, if the error of the instance ($\sum_{t=4}^{T} v(g_t)$) is at most $\frac{1-a}{1+a}$, then the algorithm is guaranteed to provide an a-EFX allocation. Given that the prediction accuracy is at least $1 - D$, we know that $\sum_{t=4}^{T} v(g_t) \leq \sum_{t=4}^{T'} p(g_t) + D$, therefore the aforementioned error can be guaranteed if $\sum_{t=4}^{T'} p(g_t) \leq \frac{1-a}{1+a} - D$. ◻

Next, we show the following.

Proposition C.5. *If $p(A_1) \geq \frac{1+a}{1-a} \cdot d_{\max} = \frac{4+a-a^2}{(2+a)(5-a)}$, for some $a \in (\varphi - 1, 1)$, then A is an a-EFX allocation according to the true values.*

Proof. By definition of the naming of the agents, $p(A_1) \leq 1/2$, $p(A_2) \geq 1/2$. Suppose that $p(A_1) \geq \frac{1+a}{1-a} \cdot d_{\max}$. Then $p(A_2) \leq 1 - \frac{1+a}{1-a} \cdot d_{\max} = \frac{2(3+a)}{(2+a)(5-a)}$. There are two cases:

(i) If $v(A_1) \geq v(A_2)$, then agent 1 does not envy agent 2, and $v(A_1) \leq p(A_1) + d_{\max} \leq 1/2 + d_{\max}$, while $v(A_2) \geq p(A_2) - d_{\max} \geq 1/2 - d_{\max}$. So this is an a'-EFX allocation for $a' \geq \frac{v(A_2)}{v(A_1)} \geq \frac{1/2 - d_{\max}}{1/2 + d_{\max}} \geq a$, by definition of $d_{\max}$.

(ii) If $v(A_1) < v(A_2)$, then agent 2 does not envy agent 1, and $v(A_1) \geq p(A_1) - d_{\max}$, while $v({}_{\mathrm{x}}A_2) \leq p({}_{\mathrm{x}}A_2) + d_{\max} \leq p(A_1) + d_{\max}$. Therefore, this is an a'-EFX allocation for $a' = \frac{v(A_1)}{v({}_{\mathrm{x}}A_2)} \geq \frac{p(A_1) - d_{\max}}{p(A_1) + d_{\max}} \geq \frac{d_{\max}(1+a)/(1-a) - d_{\max}}{d_{\max}(1+a)/(1-a) + d_{\max}} = \frac{2a}{2} = a$, where the last inequality comes from the lower bound of $p(A_1)$.

Furthermore, if the predicted horizon is shorter than the true one, the extra goods g_t, $t \in \{T'+1, T'+2, \ldots, T\}$ can be allocated to agent 1, in other words, $A_1 \leftarrow A_1 \cup \{g_{T'+1}, \ldots, g_T\}$ and the final allocation remains a-EFX. Indeed, case (i) goes through; in case (ii), since agent 1's is the envious agent, the only complication would be if agent 2, received a small (even 0-valued) good g_t, $t \in \{T'+1, T'+2, \ldots, T\}$, which would make $v({}_{\mathrm{x}}A_2)$ larger than before. However, notice that agent 2 does not receive any such good, so the analysis of this case holds too.

Notice that the proof goes through for any $d_{\max} \leq \frac{1}{2} \cdot \frac{1-a}{1+a}$. ◻

So in case $p(A_1) \geq \frac{1+a}{1-a} \cdot d_{\max} = \frac{4+a-a^2}{(2+a)(5-a)}$, allocation A itself is an a-EFX. What remains to show is how the algorithm produces an a-EFX allocation B when $p(A_1) \in \left[\frac{1}{3}, \frac{4+a-a^2}{(2+a)(5-a)}\right)$.

Proposition C.6. *If $p(A_1) < \frac{1+a}{1-a} \cdot d_{\max} = \frac{4+a-a^2}{(2+a)(5-a)}$ for some $a \in (\varphi - 1, 1)$, then $|A_2| \leq 2$.*

Proof. Let $g \in \arg\min_{g' \in A_2} p(g')$. We have $p(A_2 \setminus g) \leq p(A_1) < \frac{4+a-a^2}{(2+a)(5-a)} < \frac{4+3\sqrt{5}}{29}$, for the given domain of a. This implies that $p(g) = 1 - p(A_1) - p(A_2 \setminus g) \geq 1 - 2 \cdot p(A_1)$. For the sake of contradiction, suppose $|A_2| \geq 3$. Then $1 - p(A_1) = p(A_2) \geq 3 \cdot p(g) \geq 3 \cdot (1 - 2 \cdot p(A_1))$ implies that $p(A_1) \geq \frac{2}{5} > \frac{4+3\sqrt{5}}{29}$, a contradiction.

Notice that the proof goes through for any $d_{\max} \leq \frac{2}{5} \cdot \frac{1-a}{1+a}$. ◻

It is immediate that if $|A_2| = 0$, then $p(A_1) = 0$ (by definition of the naming of the two agents), which violates the normalization condition. If $|A_2| = 1$, then agent 1 does not EFX-envy agent 2, and so, A remains an a'-EFX allocation for the true values with $a' \geq \frac{p(A_2) - d_{\max}}{p(A_1) + d_{\max}} \geq \frac{1/2 - d_{\max}}{1/2 + d_{\max}} \geq a$, for all $a \in (\varphi - 1, 1)$.

The remaining of this proof shows how Algorithm 2 produces an a-EFX allocation B given the LPT allocation A, where $p(A_1) \in \left[\frac{1}{3}, \frac{4+a-a^2}{(2+a)(5-a)}\right)$, $|A_2| = 2$, and $|A_1| \geq 1$ (since otherwise $p(A_1) = 0 < \frac{1}{3}$, a contradiction). For ease of presentation, we will categorize the values of a given prediction p in three groups: G^z, containing all goods with highest value $z \in [0,1]$ in p, G^y, containing all goods with the second-highest value $y \in [0, z)$, and G^x, containing the rest, with values strictly smaller than y. We will also slightly abuse the notation and say that a good is of type g^z, g^y, and g^x, if it belongs to G^z, G^y, and G^x, respectively. Having satisfied the constraint that A is the outcome of the LPT algorithm (Algorithm 1) and that $|A_2| = 2$, we can have the following forms of A_2:

Form 1: $A_2 = \{g_1^z, g_2^z\}$. Then, A_1 contains at least one g^z good, otherwise LPT would not have given both g_1^z, g_2^z to A_2. If it contains two g^z goods, then $p(A_1) \geq p(A_2) \geq \frac{1}{2} > \frac{1+a}{1-a} \cdot d_{\max}$, which is a contradiction. Therefore, $A_1 = \{g_3^z, g_1^y, g_2^y, \ldots, g_\ell^y, g_1^x, g_2^x, \ldots, g_k^x\}$ for some $\ell, k \geq 0$. There are two cases. Case (i): $v(B_1) \geq v(B_2)$; then agent 1 does not envy agent 2, and B is an a'-EFX allocation for $a' \geq \frac{v(B_2)}{v(B_1)} \geq \frac{p(A_2) - d_{\max}}{p(A_1) + d_{\max}} \geq \frac{1/2 - d_{\max}}{1/2 + d_{\max}} \geq a$. Case (ii): $v(B_1) < v(B_2)$; then, agent 2 does not envy agent 1, and we have $v(B_1) \geq p(A_1) - d_{\max} = 1 - 2z - d_{\max}$, and $v({}_{\mathrm{x}}B_2) \leq z + \frac{d_{\max}}{2}$ (since both goods in B_2 have value at most $z + \frac{d_{\max}}{2}$). Therefore, B is an a'-EFX allocation with $a' \geq \frac{1 - 2z - d_{\max}}{z + d_{\max}/2} \geq \frac{1/3 - d_{\max}}{1/3 + d_{\max}/2} \geq a$, for all $a \in (\varphi - 1, 1)$, where the second to last inequality comes from the fact that $z \leq 1/3$ (otherwise, $p(A_1) + p(A_2) > 1$, a contradiction).

Notice that the above inequalities hold even if $d_{\max} = \frac{2}{3} \cdot \frac{1-a}{2+a}$, which is larger than $\frac{(4+a-a^2)(1-a)}{(2+a)(5-a)(1+a)}$ for all $a \in (\varphi - 1, 1)$.

Form 2: $A_2 = \{g_1^y, g_2^y\}$. Then, A_1 has to contain at least one g^z good (g^y is only defined if a g^z exists). Also, A_1 contains at most one g^z good, otherwise LPT would have given one g^z good to A_2. Finally, A_1 cannot contain both a g^z and a g^y good, otherwise $p(A_1) > p(A_2)$, a contradiction. So, it must be $A_1 = \{g^z, g_1^x, g_2^x, \dots, g_k^x\}$ for some $k \ge 0$. There are two cases. Case (i): $v(B_1) \ge v(B_2)$; then, agent 1 does not envy agent 2, and B is an a'-EFX allocation for $a' \ge \frac{v(B_2)}{v(B_1)} \ge \frac{p(A_2)-d_{\max}}{p(A_1)+d_{\max}} \ge \frac{1/2-d_{\max}}{1/2+d_{\max}} \ge a$. Case (ii): $v(B_1) < v(B_2)$; then, agent 2 does not envy agent 1, and among the goods g_1^y, g_2^y, g^z at most one's true value can exceed its predicted value by more than $\frac{d_{\max}}{2}$ (otherwise the error between the prediction and the true valuation is greater than $d_{\max}$, a contradiction). Also, according to the algorithm, B_1 can only have one of these goods.

There are two subcases.

(I) B_1 contains a g^y good, w.l.o.g., g_1^y. If $v(g_1^y) \le y + \frac{d_{\max}}{2}$, then this means that both goods of B_2, namely g_2^y, g^z, have true value at most $y + \frac{d_{\max}}{2}$; otherwise, one of them would have already been allocated to B_1, and g_1^y would be allocated to B_2. Then, $v(B_1) \ge v(g_1^y) \ge y - d_{\max}$, and $v({}_{\mathrm{x}}B_2) \le y + \frac{d_{\max}}{2}$. So, B is an a'-EFX allocation with $a' \ge \frac{y-d_{\max}}{y+d_{\max}/2} \ge a$, where the last inequality comes from the fact that $d_{\max} \le \frac{1-a}{3+2a}$ for all $a \in (\varphi - 1, 1)$, and $y > \frac{1}{2} - \frac{1+a}{2(1-a)} d_{\max}$ since $1 - 2y = p(A_1) < \frac{1+a}{1-a} d_{\max}$. If $v(g_1^y) > y + \frac{d_{\max}}{2}$, then each of the other two goods' true values must be $v(g_2^y) \le y + \frac{d_{\max}}{2}$ and $v(g^z) \le z + \frac{d_{\max}}{2}$ (otherwise, the error between the prediction and the true valuation is greater than $d_{\max}$, a contradiction). Notice also that $z \le p(A_1) = 1 - 2y$. Therefore, $v(B_1) \ge v(g_1^y) > y + \frac{d_{\max}}{2}$, $v({}_{\mathrm{x}}B_2) \le z + \frac{d_{\max}}{2} \le 1 - 2y + \frac{d_{\max}}{2}$, and this is an a'-EFX allocation with $a' \ge \frac{y+d_{\max}/2}{1-2y+d_{\max}/2} \ge a$, where the last inequality holds due to the aforementioned lower bound of y and the fact that $d_{\max} \le \frac{1-a}{a(5+a)}$ for all $a \in (\varphi - 1, 1)$.

(II) B_1 contains the g^z good. If $v(g^z) \le y + \frac{d_{\max}}{2}$, then both g_1^y, g_2^y have true value at most $y + \frac{d_{\max}}{2}$ (otherwise, the algorithm would have allocated one of them to B_1, and g^z would be placed at B_2). Then, $v(B_1) \ge v(g^z) \ge z - d_{\max} \ge y - d_{\max}$, $v({}_{\mathrm{x}}B_2) \le y + \frac{d_{\max}}{2}$, and so, this is an a'-EFX allocation with $a' \ge \frac{y-d_{\max}}{y+d_{\max}/2} \ge a$, where the last inequality was shown to hold in the previous subcase. If $v(g^z) > y + \frac{d_{\max}}{2}$, then $v(B_1) \ge v(g^z) - d_{\max} \ge y - \frac{d_{\max}}{2}$, and $v({}_{\mathrm{x}}B_2) \le y + d_{\max}$. Therefore, this is an a'-EFX allocation with $a' \ge \frac{y-d_{\max}/2}{y+d_{\max}} \ge \frac{y-d_{\max}}{y+d_{\max}/2} \ge a$.

Finally, notice that the above inequalities would hold even if $d_{\max} = \frac{1-a}{3+2a}$, which is larger than $\frac{(4+a-a^2)(1-a)}{(2+a)(5-a)(1+a)}$ for all $a \in (\varphi - 1, 1)$.

Form 3: $A_2 = \{g_1^z, g^y\}$. Then A_1 contains at most one g^z good, otherwise $p(A_1) > p(A_2)$, a contradiction. If it does not contain any g^z good, then $A_1 = \{g_1^y, g_2^y, \dots, g_\ell^y, g_1^x, g_2^x, \dots, g_k^x\}$ for some $\ell, k \ge 0$; then it must be $\ell \ge 2$, otherwise LPT would not give g^y to A_2. Also, according to the LPT algorithm, $\ell y \ge z$. Furthermore, we know that $\ell y \le p(A_1) < \frac{4+a-a^2}{(2+a)(5-a)}$, and $z + y = p(A_2) > 1 - \frac{4+a-a^2}{(2+a)(5-a)}$. All these, imply that $y \ge 1 - 2 \cdot \frac{4+a-a^2}{(2+a)(5-a)} = \frac{2+a+a^2}{(2+a)(5-a)}$. Therefore, $\ell \le \frac{p(A_1)}{y} < \frac{4+a-a^2}{2+a+a^2} < 2$, for all $a \in (\varphi - 1, 1)$, a contradiction. So A_1 contains a g^z good. Now notice that it cannot contain both a g^z and a g^y good, otherwise $p(A_1) \ge p(A_2) \ge \frac{1}{2} > \frac{4+a-a^2}{(2+a)(5-a)}$, a contradiction. Therefore, it must be $A_1 = \{g_2^z, g_1^x, g_2^x, \dots, g_k^x\}$ for some $k \ge 0$.

There are two cases, and notice that according to Algorithm 2, all g^x goods are given to agent 1. Case (i): $v(B_1) \ge v(B_2)$; then, agent 1 does not envy agent 2, and B is an a'-EFX allocation for $a' \ge \frac{v(B_2)}{v(B_1)} \ge \frac{p(A_2)-d_{\max}}{p(A_1)+d_{\max}} \ge \frac{1/2-d_{\max}}{1/2+d_{\max}} \ge a$. Case (ii): $v(B_1) < v(B_2)$; which splits into three subcases.

(I) The three largest goods of the instance come in the order g^y, g^z, g^z. Then, if $v(g^y) \le z + d_{\max}/2$, it is given to agent 2, and afterwards, one of the g^z goods with true value at most $z + d_{\max}/2$ is given to her too (at most one of g^y, g^z, g^z can have value greater than $z + d_{\max}/2$, otherwise the total error between predictions and true values exceeds $d_{\max}$), while the other g^z good is given to agent 1. Therefore, in this case we get $v(B_1) \ge z + \sum_{j \in [k]} p(g_j^x) - d_{\max}$, and $v({}_{\mathrm{x}}B_2) \le z + d_{\max}/2$. If $v(g^y) > z + d_{\max}/2$, it is given to agent 1, and goods g_1^z, g_2^z are given to agent 2 (and as noted above, both must have true value at most $z + d_{\max}/2$). Therefore in this case we have $v(B_1) > (z + d_{\max}/2) + \sum_{j \in [k]} p(g_j^x) - d_{\max} = z + \sum_{j \in [k]} p(g_j^x) - d_{\max}/2$, and $v(B_2) \le z + d_{\max}/2$. So, B is an a'-EFX allocation with $a' \ge \frac{z+\sum_{j \in [k]} p(g_j^x)-d_{\max}}{z+d_{\max}/2}$. If $z \ge 1/3$, then $a' \ge \frac{1-d_{\max}/z}{1+d_{\max}/(2z)} \ge \frac{1/3-d_{\max}}{1/3+d_{\max}/2} \ge a$, while if $z < 1/3$, note that $z + \sum_{j \in [k]} p(g_j^x) \ge y + \sum_{j \in [k]} p(g_j^x) = 1 - 2z$, so $a' \ge \frac{1-2z-d_{\max}}{z+d_{\max}/2} \ge \frac{1/3-d_{\max}}{1/3+d_{\max}/2} \ge a$.

(II) The three largest goods of the instance come in the order g^z, g^y, g^z. If for the first g^z good we have $v(g^z) \le z + d_{\max}/2$, it goes to agent 2. Afterwards, if $v(g^y) \le z + d_{\max}/2$, it will go to agent 2 and the remaining g^z will go to agent 1, and if $v(g^y) > z + d_{\max}/2$, the opposite allocation of these two goods will happen. If for the first g^z good we have $v(g^z) > z + d_{\max}/2$, it goes to agent 1 and the remaining g^y, g^z goods go to agent 2, while both have true value at most $z + d_{\max}/2$ as argued earlier. So from all the above cases, we will have $v(B_1) \ge z + \sum_{j \in [k]} p(g_j^x) - d_{\max}$ and $v({}_{\mathrm{x}}B_2) \le z + d_{\max}/2$. Therefore, B is an a'-EFX allocation with $a' \ge \frac{z+\sum_{j \in [k]} p(g_j^x)-d_{\max}}{z+d_{\max}/2} \ge a$ for all $a \in (\varphi - 1, 1)$, where the last inequality comes from the same analysis as that of the final step in Subcase (I).

(III) The three largest goods of the instance come in the order g^z, g^z, g^y. If for the first g^z good we have $v(g^z) \le z + \frac{(1-a)^2}{(2+a)(5-a)}$, it goes to agent 2. Then, if the second g^z good has true value $v(g^z) \le z + \frac{(1-a)^2}{(2+a)(5-a)}$, it goes to agent 2, and the remaining g^y good goes to agent 1, in which case we have $v(B_1) = 1 - v(B_2) \ge 1 - 2\left(z + \frac{(1-a)^2}{(2+a)(5-a)}\right) = 1 - 2z -$

$\frac{2(1-a)^2}{(2+a)(5-a)}$, and $v({}_{\mathrm{x}}B_2) \le z + \frac{(1-a)^2}{(2+a)(5-a)}$. So, in this case, B is an a'-EFX allocation with $a' \ge \frac{1-2z-2(1-a)^2/((2+a)(5-a))}{z+(1-a)^2/((2+a)(5-a))} \ge a$ for all $a \in (\varphi - 1, 1)$, where the last inequality comes from the fact that $z \le p(A_1) < \frac{4+a-a^2}{(2+a)(5-a)}$. If the second g^z good has true value $v(g^z) > z + \frac{(1-a)^2}{(2+a)(5-a)}$ then it goes to agent 1, and the remaining g^y good goes to agent 2, which implies $v(B_1) \ge z + \frac{(1-a)^2}{(2+a)(5-a)} - d_{\max} = z - \frac{(3+a)(1-a)}{(2+a)(5-a)(1+a)}$, and $v({}_{\mathrm{x}}B_2) \le z + \left(d_{\max} - \frac{(1-a)^2}{(2+a)(5-a)}\right) = z + \frac{(3+a)(1-a)}{(2+a)(5-a)(1+a)}$. If for the first g^z good we have $v(g^z) > z + \frac{(1-a)^2}{(2+a)(5-a)}$, then it goes to agent 1, and the remaining goods g^z, g^y go to agent 2. This implies that $v(B_1) \ge z + \frac{(1-a)^2}{(2+a)(5-a)} - d_{\max} = z - \frac{(3+a)(1-a)}{(2+a)(5-a)(1+a)}$, and $v({}_{\mathrm{x}}B_2) \le z + \left(d_{\max} - \frac{(1-a)^2}{(2+a)(5-a)}\right) = z + \frac{(3+a)(1-a)}{(2+a)(5-a)(1+a)}$. Finally, from the normalization condition we have $\sum_{j\in[k]} p(g_j^x) = 1 - 2z - y \ge 1 - 3z$, and furthermore, $\sum_{j\in[k]} p(g_j^x) + z = p(A_1) < \frac{4+a-a^2}{(2+a)(5-a)}$, which imply that $1 - 3z < \frac{4+a-a^2}{(2+a)(5-a)} - z$, or equivalently, $z > \frac{3+a}{(2+a)(5-a)}$. So, B is an a'-EFX allocation with $a' \ge \frac{z-(3+a)(1-a)/((2+a)(5-a)(1+a))}{z+(3+a)(1-a)/((2+a)(5-a)(1+a))} > \frac{1-(1-a)/(1+a)}{1+(1-a)/(1+a)} = a$, for all $a \in (\varphi - 1, 1)$.

Form 4: A_2 contains a g^x good. There are the following cases.

(i) $A_2 = \{g^y, g^x\}$; then, A_1 must contain a g^z good for the same reason as above. If it contains at least two g^z goods or a g^z and a g^y good, then $p(A_1) > p(A_2)$, a contradiction. Therefore, it must be $A_1 = \{g_2^z, g_1^x, g_2^x, \dots, g_k^x\}$ for some $k \ge 0$. The analysis of this case is omitted since it is similar to that of Form 2, with the only modification being that instead of a g^y good in A_2 we have a g^x good, where $p(g^x) \ge p(g_j^x)$ for all $j \in [k]$.

(ii) $A_2 = \{g^z, g^x\}$; then A_1 must contain a g^y good for the same reason as above. If A_1 also contains a g^z good, then $p(A_1) > p(A_2)$, a contradiction. Therefore, it must be $A_1 = \{g_1^y, g_2^y, \dots, g_\ell^y, g_1^x, g_2^x, \dots, g_k^x\}$ for some $\ell \ge 1, k \ge 0$. If $\ell + k = 1$, then $k = 0$ and the prediction has only three goods, and since $\frac{(4+a-a^2)(1-a)}{(2+a)(5-a)(1+a)} \le \frac{1-a}{1+a}$ for all $a \in (\varphi - 1, 1)$, Algorithm 3 guarantees an a-EFX allocation due to Lemma C.3. If $\ell + k \ge 2$, we know that A is an exact EFX allocation, so $p(A_1) \ge p({}_{\mathrm{x}}A_2)$, which can be used to show that $p(g^x) = p(A_2) - p({}_{\mathrm{x}}A_2) \ge p(A_2) - p(A_1) > \left(1 - \frac{1+a}{1-a} \cdot d_{\max}\right) - \frac{1+a}{1-a} \cdot d_{\max} = 1 - 2 \cdot \frac{1+a}{1-a} \cdot d_{\max}$. Let w.l.o.g., $p(g_1^x) \ge p(g_2^x) \ge \cdots \ge p(g_k^x)$. There are two subcases.

(I) $p(g^x) > p(g_1^x)$; then $\ell \ge 2$, otherwise LPT would have allocated g^x to A_1, since $y < z$. So in this case, $p(A_1) \ge 2y \ge 2p(g^x)$, where the last inequality is by definition of g^x.

(II) There is a $j \in [k]$ such that $p(g^x) \le p(g_j^x)$. So, we have $p(A_1) \ge y + p(j) \ge 2p(g^x)$.

In both subcases we have $p(A_1) \ge 2p(g^x)$. Furthermore, notice that $2\left(1 - 2 \cdot \frac{1+a}{1-a} \cdot d_{\max}\right) \ge \frac{1+a}{1-a} \cdot d_{\max}$, since $d_{\max} \le \frac{2}{5} \cdot \frac{1-a}{1+a}$ for all $a \in (\varphi - 1, 1)$. Therefore, we have $p(A_1) \ge 2p(g^x) \ge 2\left(1 - 2 \cdot \frac{1+a}{1-a} \cdot d_{\max}\right) \ge \frac{1+a}{1-a} \cdot d_{\max} > p(A_1)$, a contradiction.

(iii) $A_2 = \{g_1^x, g_2^x\}$; then, A_1 must contain a g^z and a g^y good (by definition, there cannot be a g^x without a g^y and a g^z good), therefore, $p(A_1) > p(A_2)$, a contradiction.

Finally, notice that Algorithm 2 first reads the prediction, finds an exact EFX allocation A using the LPT algorithm as a subroutine, which takes $O(T' \log T')$ time as shown below.

Proposition C.7. *For $n \ge 2$ agents with the same additive valuation function $f : [T] \to \mathbb{R}_{\ge 0}$ given explicitly in the input, the LPT algorithm (Algorithm 1) outputs an exact EFX allocation in time $O(T \log(nT))$.*

Proof. Notice that the algorithm works in the offline setting, since it needs to have as input the values $f(g_1), f(g_2), \dots, f(g_T)$. By induction on t, we will prove that after creating the set M' of the goods according to a non-ascending order of value, after each round $t \in [T]$, the partial allocation is an exact EFX allocation. In round $t = 1$, an arbitrary agent has been allocated the most valuable good, and this is obviously an exact EFX allocation. Now suppose that we have an exact EFX allocation at time $t^* - 1$ for some $t^* \in \{2, 3, \dots, T-1\}$; we will show that the algorithm gives an exact EFX allocation at time t^*. Let the for-loop of the algorithm be at round t^*. Note that agent i^* (who receives a good in this round) cannot increase her EFX-envy towards other agents. So it suffices to show that the rest of the agents do not EFX-envy i^*. At line 4, no agent envied i^*. Also, by definition of M', we have $f(g'_{t^*}) \le f(g'_t)$ for all $t \in [t^*]$, so at line 5 agent i^* gets allocated the good of minimum value among the ones already allocated to the agents. Therefore, for every agent $i \in [n]$ we have $f(A_i) \ge f({}_{\mathrm{x}}A_{i^*})$, so no one EFX-envies i^*. Notice that 2 takes $O(T \log T)$ time to sort T values, and line 4 takes $O(\log n)$ time (e.g., by using a min-heap), resulting in the for-loop having time $O(T \log n)$. □

Then, Algorithm 2 categorizes A to a particular Form in time $O(T')$ by finding the smallest valued bundle to determine A_1, and checking in which bundle each good is. Then, as each good arrives online, it decides its allocation by doing a constant number of basic operations, that is, updating the total value of each bundle and comparing the true value of the good with its predicted one. □

C.4 Proof of Corollary 4.10

Proof. Observe that when the prediction is a 2-value function, the proof of Theorem 4.8 goes through for $d_{\max} = \frac{2}{5} \cdot \frac{1-a}{1+a}$. That is because Proposition C.5 holds for any $d_{\max} \le \frac{1}{2} \cdot \frac{1-a}{1+a}$, Proposition C.6 holds for any $d_{\max} \le \frac{2}{5} \cdot \frac{1-a}{1+a}$, and, since there are no g^x goods, A can have either Form 1, Form 2, or Form 3, with $k = 0$. The latter two forms involve only three predicted goods, so Lemma C.3 applies. Finally, Form 1 goes through for $d_{\max} \le \frac{2}{3} \cdot \frac{1-a}{2+a}$. It turns out that the minimum of all the above bounds is $\frac{2}{5} \cdot \frac{1-a}{1+a}$. □

C.5 Proof of Theorem 4.11

Proof. For the sake of contradiction, suppose there is an algorithm that for some $a \in (\sqrt{3} - 1, 1]$ and accuracy $\eta < 1 - \frac{1-a}{2}$, guarantees an a-EFX allocation. The adversary fixes a rational $\varepsilon \in \left(\frac{1-a}{2}, \frac{a}{2(2+a)}\right)$, and notice that this interval is non-empty only for $a \in (\sqrt{3} - 1, 1]$. Let the number of rounds be $T := 4$, which is known by the algorithm (since it is given the prediction itself).

The adversary gives to the algorithm the prediction: $p(g_1) = p(g_2) = \varepsilon$, $p(g_3) = p(g_4) = \frac{1}{2} - \varepsilon$. The adversary reveals the true values $v(g_1) = v(g_2) = \varepsilon$. There are two cases:

(i) g_1, g_2 get allocated to the same agent, w.l.o.g. agent 1. Then, the adversary reveals the true values $v(g_3) = v(g_4) = \frac{1}{2} - \varepsilon$. if both g_3, g_4 get allocated to agent 1, agent 2 is 0-EFX envious towards him, therefore this is not an a-EFX allocation as requested. If both g_3, g_4 get allocated to agent 2, then this is an a'-EFX allocation with $a' = \frac{2\varepsilon}{1/2-\varepsilon} < a$, where the inequality comes from the fact that $\varepsilon < \frac{a}{2(2+a)}$. If, w.l.o.g., g_3 goes to agent 1 and g_4 goes to agent 2, then this is an a'-EFX allocation with $a' = \frac{1/2-\varepsilon}{1/2} < a$, where the inequality is due to the fact that $\varepsilon > \frac{1-a}{2}$.

(ii) W.l.o.g., g_1 gets allocated to agent 1, and g_2 gets allocated to agent 2. Then the adversary reveals the true values $v(g_3) = \frac{1}{2} - 2\varepsilon$, $v(g_4) = \frac{1}{2}$. If g_3, g_4 get allocated to the same agent, then this is an a'-EFX allocation with $a' = \frac{\varepsilon}{1-2\varepsilon} \leq \frac{2\varepsilon}{1/2-\varepsilon} < a$, where the last inequality comes from the fact that $\varepsilon < \frac{a}{2(2+a)}$. If, w.l.o.g., g_3 goes to agent 1 and g_4 goes to agent 2, then this is an a'-EFX allocation with $a' = \frac{1/2-\varepsilon}{1/2} < a$, where the inequality is due to the fact that $\varepsilon > \frac{1-a}{2}$.

Therefore, it is not possible for the algorithm to output an a-EFX allocation. Also, notice that the TV distance between the prediction and the adversary is ε which can take any value in $\left(\frac{1-a}{2}, \frac{a}{2(2+a)}\right)$. $\square$