Proportionally Fair Online Allocation of Public Goods Using Machine-Learned Predictions

Siddhartha Banerjee
Cornell University
sbanerjee@cornell.edu

Vasilis Gkatzelis
Drexel University
gkatz@drexel.edu

Safwan Hossain
University of Toronto
safwan.hossain@mail.utoronto.ca

Billy Jin
Cornell University
bzj3@cornell.edu

Evi Micha
University of Toronto
emicha@cs.toronto.edu

Nisarg Shah
University of Toronto
nisarg@cs.toronto.edu

Abstract

We design online algorithms for fair allocation of public goods to a set of N agents over a sequence of T rounds and focus on improving their performance using machine-learned predictions. In the basic model, a public good arrives in each round, the algorithm learns every agent's value for the good, and must irrevocably decide the amount of investment in the good without exceeding a total budget of B across all rounds. The algorithm can utilize (potentially inaccurate) machine-learned predictions of each agent's total value for all the goods to arrive. We measure the performance of the algorithm using a *proportional fairness* objective, which informally demands that every group of agents be rewarded in proportion to its size and the cohesiveness of its preferences.

In the special case of binary agent preferences and a unit budget, we show that $O(\log N)$ proportional fairness can be achieved without using any predictions, and that this is optimal even if perfectly accurate predictions were available. However, for general preferences and budget no algorithm can achieve better than $\Theta(T/B)$ proportional fairness without predictions. We show that algorithms with (reasonably accurate) predictions can do much better, achieving $\Theta(\log(T/B))$ proportional fairness. We also extend this result to a general model in which a batch of L public goods arrive in each round and achieve $O(\log(\min(N, L) \cdot T/B))$ proportional fairness. Our exact bounds are parametrized as a function of the error in the predictions and the performance degrades gracefully with increasing errors.

1 Introduction

In classic online algorithms, the input is presented in stages and the algorithm needs to make irrevocable decisions at each stage without knowing the input from future stages. Its performance, called the competitive ratio, is measured by comparing the worst-case ratio of the achieved solution quality to the optimal solution quality in hindsight [1]. The uncertainty regarding the future often forces such algorithms to make overly cautious decisions, resulting in unappealing competitive ratios.

An emerging line of research asks whether one can utilize the impressive progress in machine learning (ML) to augment online algorithms with machine-learned predictions regarding the future [2]. Ideally, one would hope for bicriteria guarantees, ensuring that the competitive ratio is improved when the predictions are reasonably accurate but remaining robust even when they are not. More generally, one can hope to express the competitive ratio of the online algorithm directly in terms of the error in the predictions. This powerful paradigm has already received significant attention, including from the machine learning community, for problems such as caching [3–5], the secretary problem [6–8], scheduling [9], the ski rental problem [10–12], set cover [13], and other problems [14–22].

However, all of this work is limited to *single-agent* decision-making problems. Recently, Banerjee et al. [23] applied this paradigm to design online algorithms for a *multi-agent* resource allocation problem, in which a set of *private* goods (which can only be allocated to and enjoyed by a single agent) need to be divided amongst a group of agents in a *fair* manner. This problem has direct applications to settings like inheritance division and divorce settlement, and has garnered interest from various research communities, dating back to Steinhaus [24]. Using the *Nash welfare* from bargaining theory [25] as their notion of fairness, Banerjee et al. [23] show that predictions about agents' total value for all goods can be utilized to achieve significantly improved approximations.

The solutions proposed by Banerjee et al. [23], however, do not capture resource allocation settings involving *public* goods, i.e., goods whose benefit can be enjoyed by multiple agents (e.g., a highway or a park). In many important problems, like participatory budgeting, committee selection, or shared memory allocation, some scarce resources need to be dedicated to make each public good available, and an algorithm needs to decide which goods to invest in, aiming to make the agents happy. Fairness in these settings is often captured by notions like the Nash welfare and the *core* [26]. Yet, despite the significance of its applications, only a few papers have successfully studied the fair allocation of public goods [27–33] (relative to the extensive literature on private goods; see [34]), even fewer have provided *online* algorithms for this problem [35], and none utilize machine-learned predictions.

We address this gap by designing online algorithms for fair allocation to public goods using predictions about how agents value the goods. The key research questions we address are:

> *How can we allocate public goods in an online yet fair manner? To what extent can predictions about agent preferences help improve the fairness guarantees?*

1.1 Our Results

We study the design of online algorithms for fair allocation to public goods arriving over a sequence of T rounds based on the preferences of N agents. In the basic model, in each round t, a new public good (which we refer to as good t) arrives and the algorithm learns the value $v_{i,t}$ of each agent i for this good. Using this information, the algorithm needs to make an irrevocable decision regarding an amount $x_t \in [0, 1]$ to invest in this good, before the end of this round. This value x_t can be represented as the amount of a scarce resource (e.g., money) devoted toward making this good available, or the probability of implementing the good. Each agent i then receives value $v_{i,t} \cdot x_t$ from this investment, and the total value derived by an agent is the sum of the values gained across rounds.

Clearly, the algorithm would like to increase the x_t's as much as possible, but the algorithm is limited by a total budget constraint: $\sum_t x_t \leqslant B$, where B is given. Informally, the budget constraint forces the algorithm to invest more in goods that are highly valued by many agents, which is challenging in an online setting, since the agents' values for future goods are unknown. To deal with this uncertainty, the algorithm has access to *machine-learned predictions* regarding the *total value* of each agent i: this prediction $\widetilde{V}_i$ provides an estimate on the total value $V_i = \sum_t v_{i,t}$ of agent i for all goods to arrive.

We focus on a quantitative fairness objective, called *proportional fairness* (Definition 1), which is stronger than previously considered objectives of the Nash welfare and the core (see Section 2.1). Lower objective values indicate better fairness, with 1 indicating perfect proportional fairness.

In Section 3, as a warm-up, we consider the setting where agent values are binary (i.e., $v_{i,t} \in \{0, 1\}$ for all agents i and goods t) and the budget is $B = 1$. Binary values correspond to *approval voting* – where agents either like a good or not; the unit budget forces $\sum_t x_t \leqslant 1$, meaning that x_t can also be viewed as the fraction of an available resource invested in good t. For this special case, we show that it is already possible to achieve $O(\log N)$ proportional fairness without using any predictions and this is optimal even if the algorithm had access to *perfect predictions* ($\widetilde{V}_i = V_i$ for each agent i).

In Section 4, we consider general values and budget, and show that this is no longer true: without access to predictions, no algorithm can achieve $o(T/B)$ proportional fairness even with $N = 1$ agent. Our main positive result shows that, by using the predictions, we can achieve an exponential improvement to $O(\log(T/B))$ proportional fairness for N agents, as long as the predictions are reasonably accurate (constant multiplicative error). We also show this to be optimal given predictions.

Finally, in Section 5 we extend our model even further by allowing a batch of L public goods to arrive in each round t and achieve $O(\log(\min(N, L) \cdot T/B))$ proportional fairness. In fact, we show that this model strictly generalizes not only our initial public-goods model, but also the private-goods model of Banerjee et al. [23], and our positive result in this very broad model (almost) implies theirs.

1.2 Related Work

Allocation of private goods. The majority of the prior work on online fair division has focused on *private* goods, for which achieving even basic notions of fairness comes at the cost of extreme inefficiency in the absence of any predictions regarding agents' values for future goods [36, 37]. Banerjee et al. [23] show that total value predictions can be leveraged to achieve improved fairness guarantees with respect to the Nash social welfare objective. Note that assuming access to predictions of agents' total values for all goods is related to work which assumes that the values of each agent are normalized to add up to 1 [38, 39] or that they are drawn randomly from a normalized distribution [40]. Our work reinforces the findings of Banerjee et al. [23] that predictions help significantly improve fairness guarantees, but in our more general model with public goods.

Allocation to public goods. Much of the literature on public goods focuses on the offline setting – agents have approval preferences, and we have a fixed budget (i.e., the offline version of the setting in Section 3). This offline version has been studied under various names, such as probabilistic voting [41], fair mixing [42], fair sharing [43], portioning [44], and randomized single-winner voting [45]. Ebadian et al. [45] show that with access to only *ranked* preferences, $\Theta(\log T)$ is the best possible proportional fairness in the offline setting. Interestingly, we show that incomplete information resulting from online arrivals leads to the same $\Theta(\log T)$ proportional fairness.

Multi-winner voting extends this by selecting a subset of k candidates, which is the offline version of our model in Section 4 with $B = k$. Multi-winner voting can be extended to fair public decision-making [30] and participatory budgeting [27, 46], which are further generalized by the public good model of Fain et al. [31]. These generalizations allow more complex feasibility restrictions on the allocations than our budget constraint, but work in the offline setting. To the best of our knowledge, the only work to consider *online* allocation to public goods is that of Freeman et al. [35], who consider optimizing the Nash welfare in a model similar to ours. However, they do not provide any approximation guarantees; instead, they study natural online algorithms from an axiomatic viewpoint.

Primal-dual analysis. Finally, we remark that our main positive result is derived using a primal-dual-style analysis, which is common in the online resource allocation literature (see the survey of Buchbinder and Naor [47] for an excellent overview). Almost all of this work deals with additive objective functions. Two notable exceptions to this are the work of [48], who show how to extend these approaches to non-linear functions of additive rewards, as well as Azar et al. [49], who consider a variant of the proportional allocation objective, but require additional boundedness conditions on the valuations. Bamas et al. [13] show how to adapt primal-dual analyses to incorporate predictions for a number of single-agent problems.

2 Model

We study an online resource allocation problem with N agents and T rounds. Our algorithms observe the number of agents N, but they may not know the number of rounds T, in advance. We study both *horizon-aware* algorithms which know T and *horizon-independent* algorithms which do not. For simplicity we use $[k]$ to denote the set $\{1, \ldots, k\}$ for a given $k \in \mathbb{N}$.

Online arrivals. In the basic version of the model, in each round $t \in [T]$, a public good, which we refer to as good t, "arrives". Upon its arrival, we learn the value $v_{i,t} \geqslant 0$ of every agent $i \in [N]$ for it. In Section 5, we extend the model to allow a batch of L public goods arriving in each round.

Online allocations. When good t arrives, the online algorithm must irrevocably decide the allocation $x_t \in [0, 1]$ to good t, before the next round.[1] We use $\mathbf{x} = (x_t)_{t \in [T]}$ to denote the final allocation computed by the online algorithm. In the absence of any further constraints, the decision would be simple: allocate as much as possible to every good by setting $x_t = 1$ for each $t \in [T]$. However,

[1] In this work we focus on *deterministic* algorithms, but this is w.l.o.g. since we consider fractional allocations.

the extent to which the algorithm can allocate these goods is limited by an overall budget constraint: $\sum_{t=1}^{T} x_t \leqslant B$, where $B \geqslant 0$ is a fixed budget known to the online algorithm in advance.

Linear agent utilities. Choosing to allocate x_t to good t simultaneously yields utility $v_{i,t} \cdot x_t$ to every agent $i \in [N]$. Moreover, we assume that agent utilities are additive across goods, i.e., the final utility of agent i is given by $u_i(\mathbf{x}) = \sum_{t=1}^{T} v_{i,t} \cdot x_t$. This class of linear agent utilities is widely studied and it admits several natural interpretations, depending on the application of interest. In applications like budget division, each public good t is a project (e.g., an infrastructure project), and x_t is the amount of a resource (e.g., time or money) invested in the project. In applications such as participatory budgeting or voting, each public good t is an alternative or a candidate, and x_t is the (marginal) probability of it being selected (indeed, one can compute a lottery over subsets of goods of size at most B under which the marginal probability of selecting each good t is precisely x_t).

When working with fractions x/y with $x, y \geqslant 0$, we adopt the convention that $x/y = 1$ when both $x = y = 0$, while $x/y = +\infty$ if $y = 0$ and $x > 0$. We use $H_k = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{k}$ to denote the k-th harmonic number.

2.1 Proportional Fairness

We want the allocation $\mathbf{x}$ computed by our online algorithm to be *fair*. In this work, we use the notion of proportional fairness, which is a quantitative fairness notion that was first proposed in the context of rate control in communication networks [50].

Definition 1 (Proportional Fairness)**.** For $\alpha \geqslant 1$, allocation $\mathbf{x}$ is called α-proportionally fair if for every other allocation $\mathbf{w}$ we have $\frac{1}{N} \sum_{i=1}^{N} \frac{u_i(\mathbf{w})}{u_i(\mathbf{x})} \leqslant \alpha$. If $\mathbf{x}$ is 1-proportionally fair, we simply refer to it as proportionally fair[2]. We say that an online algorithm is α-proportionally fair if it always produces an α-proportionally fair allocation.

It is known that in the offline setting, where all agent values are known up front, a 1-proportionally fair allocation $\mathbf{x}$ always exists, and this is the lowest possible value of proportional fairness [31]. It is also known that proportional fairness is a strong guarantee that implies several other guarantees sought in the literature. In Appendix A, we show two examples: the core and Nash social welfare.

2.2 Set-Aside Greedy Algorithms

In order to compute (approximately) proportionally fair allocations, we consider a family of online algorithms, called *Set-Aside Greedy Algorithms*. Recent work [23, 39] has demonstrated how such algorithms can be used to get strong performance guarantees for online allocation of *private goods*; we show that with non-trivial modifications, they can also achieve compelling fairness guarantees for allocating public goods.

At a high level, an algorithm in this family divides the overall budget B into two equal portions.

1. The first half, called the *set-aside budget*, is used to allocate $y_t \in [0, 1]$ to each good t in such a manner that $\sum_{t=1}^{T} y_t \leqslant B/2$ and this portion of the allocation guarantees each agent i a certain minimum utility of Δ_i (i.e., $\sum_{t=1}^{T} v_{i,t} \cdot y_t \geqslant \Delta_i$). For example, if $y_t = B/(2T)$ for each $t \in [T]$, then we can use $\Delta_i = \frac{B}{2T} \cdot \sum_{t=1}^{T} v_{i,t}$. This ensures that in the proportional fairness definition (Definition 1), the ratio $\frac{u_i(\mathbf{w})}{u_i(\mathbf{x})}$ is not excessively large for any agent i.
2. The second half, called the *greedy budget*, is used to allocate $z_t \in [0, 1 - y_t]$ to each good t in such a manner that $\sum_{t=1}^{T} z_t \leqslant B/2$. This portion of the budget is used in a adaptive greedy-like fashion toward online optimization of the desired objective.

We refer to y_t and z_t as *semi-allocations* to good t, and the final allocation to good t is determined by combining these two semi-allocations, i.e., $x_t = y_t + z_t$. An important quantity in both our algorithm design and its analysis is the promised utility to an agent, defined next.

Definition 2 (Promised Utility)**.** The semi-allocations $y_1, \ldots, y_T$ guarantee that by the end of the algorithm each agent i will receive a utility of $\sum_{t=1}^{T} v_{i,t} \cdot y_t \geqslant \Delta_i$ from the set-aside portion of

[2]The 1-proportional fair criterion is more commonly (but equivalently) written as $\frac{1}{N} \sum_{i=1}^{N} \frac{u_i(\mathbf{w}) - u_i(\mathbf{x})}{u_i(\mathbf{x})} \leqslant 0$.

the budget. By round t the algorithm has already set semi-allocations $z_1, \ldots, z_{t-1}$ using the greedy portion of the budget, and needs to now decide z_t. At this stage, as a function of z_t, the algorithm can guarantee that each agent i will eventually receive utility at least $\widetilde{u}_{i,t}(z_t) = \Delta_i + \sum_{\tau=1}^{t} v_{i,\tau} \cdot z_\tau$, even if they do not benefit from any more of the greedy budget. We refer to this as the *promised utility*.

3 Warm Up: Binary Utilities and Unit Budget

Before presenting our main results, we first build some intuition regarding our online setting and the proportional fairness objective by considering the interesting special case where agents have binary utilities for goods (i.e., $v_{i,t} \in \{0, 1\}$ for each i, t) and the total budget is $B = 1$. In this setting, which is motivated by approval voting, we say that agent i "likes" good t if $v_{i,t} = 1$, and does not like good t otherwise. Note that with $B = 1$, the budget constraint is $\sum_{t=1}^{T} x_t \leqslant 1$, which means x_t can be interpreted as the *fraction* of an available resource (e.g., time or money) that is dedicated to good t.

First, in the trivial case with a single agent ($N = 1$), we can simply set $x_t = 1$ when the first good t liked by the agent arrives,[3] which easily yields (exact) proportional fairness.

It is tempting to extend this idea to the case of $N > 1$ agents. However, we find that even in this restricted scenario with binary utilities and unit budget, no online algorithm achieves $o(\log N)$-proportional fairness, or even the weaker guarantee of $o(\log N)$-approximation of the Nash welfare. In fact, this remains true even if the algorithm is horizon-aware (i.e., knows T in advance) and knows precisely how many goods each agent will like in total. Intuitively, this is because we show that no online algorithm can sufficiently distinguish between instances where many agents like the same goods and those where agents like mostly disjoint goods. The proof appears in the appendix.

Theorem 1. *With binary agent utilities ($v_{i,t} \in \{0, 1\}, \forall i, t$) and unit budget ($B = 1$), every online algorithm is $\Omega(\log N)$-proportionally fair (in fact, achieves $\Omega(\log N)$-approximation of the Nash welfare), even if the algorithm is horizon-aware and knows in advance the total number of goods each agent will like.*

Next, we provide a set-aside greedy algorithm that achieves $O(\log N)$ proportional fairness (and therefore, $O(\log N)$-NSW optimality), thus establishing $\Theta(\log N)$ as the best possible approximation in this restricted scenario. We remark that this restricted case of binary utilities and unit budget already poses an interesting challenge by preventing constant approximation, but $O(\log N)$ approximation is still quite reasonable as it does not depend on the horizon T (which can typically be very large) and in practice the number of agents N is reasonably small. We also remark that we achieve the $O(\log N)$ upper bound using a horizon-independent algorithm, while the lower bound of Theorem 1 holds even when the algorithm is horizon-aware.

The algorithm, which, due to space limitations, is deferred to Appendix B (Algorithm 2), works as follows: it uses the set-aside portion of the budget to set $y_t = 1/(2N)$ whenever good t is the first liked good of at least one agent (note that $\sum_t y_t \leqslant 1/2$). This ensures that each agent i gets utility at least $\Delta_i = 1/(2N)$.[4] Based on this, the algorithm uses the following expression of promised utility to agent i in round t: $\widetilde{u}_{i,t}(z_t) = \frac{1}{2N} + \sum_{\tau=1}^{t} v_{i,\tau} \cdot z_\tau$.

The algorithm chooses z_t in a greedy manner (i.e., smallest possible) such that, for each agent i, the ratio of her value $v_{i,t}$ for good t to her promised utility $\widetilde{u}_{i,t}(z_t)$ is at most a target quantity. We defer the proof of this result to the appendix because we will present this technique in much more generality in Section 5.1 (the only adjustment required in the proof of the next result is the slightly different expression of $\Delta_i = 1/(2N)$ specific to this case of binary utilities and unit budget).

Theorem 2. *Algorithm 2 with $\alpha \geqslant 2\ln(2N)$ realizes an α-proportional fair allocation.*

4 General Utilities and Budget

Having built some intuition about the online setting and the proportional fairness objective by considering the restricted special case of the problem wherein all values v_{it} are in $\{0, 1\}$ and the

[3] If the agent does not like any good, (exact) proportional fairness is trivially achieved.

[4] Agents who do not like any good can only improve the approximation.

budget is $B = 1$, we now turn to the more general model described in Section 2. Recall that this model generalizes the setting in Section 3 in two ways:

1. Agent values $v_{i,t}$ can now be any (non-negative) real number.
2. The budget constraint is $\sum_{t=1}^{T} x_t \leqslant B$, for an arbitrary $B \geqslant 0$, so the per-round constraint of $x_t \leqslant 1$, for each $t \in [T]$, is no longer redundant.

4.1 The Case for Predictions

In this general case, we prove that the problem becomes significantly more difficult: without access to any machine-learned predictions, every online algorithm is $\Omega(T/B)$-proportionally fair (in fact, achieves $\Omega(T/B)$-approximation of the Nash welfare), in stark contrast to the $O(\log N)$-proportional fairness that we were able to achieve in the previous section. The proof is in the appendix.

Proposition 1. *Under general agent values and budget B, every online algorithm is $\Omega(T/B)$-proportionally fair (in fact, achieves $\Omega(T/B)$-approximation of the Nash welfare).*

The hardness instance used above is specifically engineered to exploit the fact that the algorithm has no information about the future. In most practical settings, it is reasonable to assume that the algorithm has access to some information about the input. This could come from historical data, stochastic assumptions, or simply from properties of the problem at hand (e.g. if $v_{i,t}$ represents the monetary value that agent i has for good t, then we may have bounds on how large $v_{i,t}$ can be.)

Motivated by this, we now turn to the growing literature on prediction-augmented algorithms and allow the algorithm access to additional side-information about agents' valuations. Clearly, if the entire valuation matrix $(v_{i,t})_{i\in[N],t\in[T]}$ is available beforehand, then the problem is trivial; the challenge lies in understanding what *minimal* additional information (or 'prediction') can lead to sharp improvements in performance, and how robust these improvements are to errors in these predictions. To this end, we now adapt an idea introduced by Banerjee et al. [23] for online allocation with private goods, and assume that the algorithm has side information about each agent's total value for all the goods.

Definition 3 (Total Value Predictions)**.** For any agent i, we define her total value to be $V_i = \sum_{t=1}^{T} v_{i,t}$. Moreover, for $c_i, d_i \geqslant 1$, $\widetilde{V}_i$ is said to be a $[c_i, d_i]$-prediction of V_i if $\widetilde{V}_i \in \left[\frac{1}{d_i} V_i,\ c_i V_i\right]$.

In other words, c_i and d_i denote the multiplicative factors by which the prediction $\widetilde{V}_i$ may overestimate and underestimate, respectively, the value of V_i. When $c_i = d_i = 1$, we call them *perfect predictions*.

In the next section, we assume that we have access to $\widetilde{V}_i$ for each agent i. The purpose of the c_i and d_i is to parameterize the *robustness* of our algorithm, i.e., the degradation in its performance as the predictions get worse. Our algorithm does need to know (an upper bound on) the d_i's for tuning one of its parameters; it does not however need to know the c_i's (these are only used in the analysis).

4.2 Achieving Proportional Fairness with Predictions

Using the above notion of predictions, we design Algorithm 3, given in Appendix C, a variant of our earlier Set-Aside Greedy algorithm that has a dramatically better proportional fairness guarantee compared to the hardness result of $\Omega(T/B)$ in Proposition 1. Given perfect predictions, our algorithm achieves a proportional fairness of $O(\log(T/B))$. Moreover, Algorithm 3 turns out to be remarkably robust to prediction errors; in particular, all our asymptotic guarantees remain unchanged as long as all the $c_i = O(1)$ and $d_i = O(\log(T/B))$.

As before, the algorithm splits the budget into two parts, and the total allocation to the good in round t is obtained by adding the contributions (semi-allocations) from each part, i.e., $x_t = y_t + z_t$. The semi-allocation from the first (*set-aside*) part is $y_t = B/(2T)$ for each $t \in [T]$. This portion of the allocation guarantees each agent i utility at least $\Delta_i = \frac{B}{2T} \cdot V_i$. Now in each round $t \in [T]$, the algorithm uses the second the part of the budget to compute a *greedy* semi-allocation z_t. This is done by choosing z_t to optimize a function of the agents' *predicted promised utilities*.

Definition 4 (Predicted Promised Utility)**.** Given a prediction $\widetilde{V}_i$ of the total value of agent i, the predicted promised utility of agent i in round $t \in [T]$ is defined as

$$\widetilde{u}_{i,t}(z_t) = \frac{B}{2T} \cdot \widetilde{V}_i + \sum_{\tau=1}^{t} v_{i,\tau} \cdot z_\tau. \tag{1}$$

We omit $z_1, \ldots, z_{t-1}$ from the argument of $\widetilde{u}_{i,t}$ since they are fixed prior to round t.

Note that this quantity can be computed by the algorithm, as a function of z_t it wants to choose, since it has knowledge of $\widetilde{V}_i$ (prediction) and semi-allocations $\{z_\tau\}_{\tau<t}$ from the previous rounds. We use these predicted promised utilities in Algorithm 3 to achieve the following guarantee.

Theorem 3. *For any $\alpha \geqslant 4\ln\left(\frac{2T}{B}\right) + \frac{4}{N}\sum_i \ln(d_i)$, Algorithm 3 produces a feasible allocation $\mathbf{x}$, which satisfies $\max_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N} \frac{u_i(\mathbf{w})}{c_i \cdot u_i(\mathbf{x})} \leqslant \alpha$, where the maximum is over all feasible allocations $\mathbf{w}$.*

This result is almost subsumed by our positive result (Theorem 5) for the more general model in the next section, in which we allow a batch of L public goods to arrive in each round; setting $L = 1$ recovers precisely the bounds derived in this section. The only difference is that in the algorithm for the batched model (Algorithm 1), the greedy semi-allocation step is a convex optimization problem, which can be solved up to an ϵ error; for the single-good-per-round case, we are able to replace this with a combinatorial step in Algorithm 3 that can be performed exactly in polynomial time. This requires us to show that the greedy semi-allocations computed by this combinatorial step satisfy the same properties (specifically, those in Proposition 5) that a solution to the convex optimization problem would have satisfied. We present this result and its proof in the appendix as Proposition 4.

Let us consider the implications of Theorem 3. The expression in the statement of Theorem 3 is not exactly the proportional fairness objective, since the term for each agent i is scaled with a (potentially different) factor c_i. Applying the arguments in Section 2.1, we can turn this into an approximation of proportional fairness, the core, and the Nash social welfare. The proof is in the appendix.

Corollary 1. *For $\alpha \geqslant 4\ln\left(\frac{2T}{B}\right) + \frac{4}{N}\sum_i \ln(d_i)$, Algorithm 3 is*

1. *$(\alpha \cdot \max_{i\in[N]} c_i)$-proportionally fair, and hence in the $(\alpha \cdot \max_{i\in[N]} c_i)$-core, and*
2. *achieves $(\alpha \cdot (\prod_{i\in[N]} c_i)^{\frac{1}{N}})$-approximation of the Nash welfare.*

4.3 Hardness with Predictions

Theorem 3 shows that in online allocation of public goods with general values and budget, having access to reasonable predictions of each agent's total value can lead to a dramatic improvement in the proportional fairness guarantee from $\Omega(T/B)$ to $O(\log(T/B)))$. Given the size of the side information (which lies in $\mathbb{R}^N$, since we need one prediction per agent) relative to the ambient size of the input (which lies in $\mathbb{R}^{NT}$, with one valuation per agent per round), this is a surprising improvement in performance.

One may wonder whether these predictions are so strong that one can do even better. The following result shows, however, that even with a single agent, and perfect knowledge of her total value V_1, any online algorithm is $\Omega(\log(T/B))$-proportionally fair. Thus Algorithm 3 is essentially optimal for our setting. The proof is in the appendix.

Theorem 4. *For $N = 1$ agent, every online algorithm is $\Omega(\log(T/B))$-proportionally fair (in fact, achieves an $\Omega(\log(T/B))$-approximation for the Nash welfare), even with perfect knowledge of horizon T and the total value of the agent $\widetilde{V}_1 = V_1 = \sum_{t=1}^{T} v_{i,t}$.*

5 Extension to Batched Arrivals

In this section, we present our most general setting, which we refer to as the *batched public goods* model. This not only generalizes the model in Section 2 with a single good per round, but also the setting of Banerjee et al. [23] with private goods.

In each round $t \in [T]$, a batch of L public goods arrive (as opposed to a single public good). Upon their arrival, the algorithm learns the value $v_{i,\ell,t} \geqslant 0$ of each agent $i \in [N]$ for each good $\ell \in [L]$

in the batch. It then must irrevocably decide the allocation $x_{\ell,t} \in [0,1]$ to each good ℓ in the batch, before the next round. We use $\mathbf{x}_t = (x_{\ell,t})_{\ell \in [L]}$ to denote the allocation in round t, and $\mathbf{x} = (\mathbf{x}_t)_{t \in [T]}$ to denote the final allocation. We also incorporate two types of constraints on the allocation $\mathbf{x}$:

1. (Per-round constraint) $\sum_{\ell=1}^{L} x_{\ell,t} \leqslant 1$ for all $t \in [T]$.
2. (Overall constraint) $\sum_{t=1}^{T} \sum_{\ell=1}^{L} x_{\ell,t} \leqslant B$ for $B \geqslant 0$ known to the algorithm in advance.

The allocation $x_{\ell,t}$ to good ℓ in round t yields utility $v_{i,\ell,t} \cdot x_{\ell,t}$ to every agent i. We assume that agent utilities are additive, i.e., the final utility of agent i is given by $u_i(\mathbf{x}) = \sum_{t=1}^{T} \sum_{\ell=1}^{L} v_{i,\ell,t} \cdot x_{\ell,t}$. Our aim, as before, is is to realize an α-proportionally fair allocation (Definition 1 does not require any modifications except using these new utility functions) for the smallest possible α.

Note that the overall constraint is the same as in the model in Section 2 with a single good per round. However, in the batched model, the per-round constraint can place additional restriction on how much budget can be spent in any single round. Note also that choosing the per-round bound to be 1 is without loss of generality; in particular, since agent utilities are linear, having a per-round constraint of b can be reduced to our setting by scaling each allocation, as well as the total budget, by a factor of b. Finally, the per-round constraint becomes vacuous if $B \leqslant 1$, and the overall constraint becomes vacuous if $B \geqslant T$; therefore, we assume, without loss of generality, that $1 \leqslant B \leqslant T$.

This model captures the following special cases. Before we dive into the algorithm and analysis, we briefly mention some special cases of interest which this model generalizes.

1. *Single public good.* When $L = 1$, we trivially recover the setting of Section 4, where there is one public good in each round.
2. *Single private good.* When $L = N$, $B = T$, and $v_{i,\ell,t} = 0$ if $i \neq j$, we recover the setting studied by Banerjee et al. [23]. In their setting, there is a single private good arriving in each round, which the algorithm needs to split among the N agents. When cast in our model, $v_{i,i,t}$ is the value that agent i has for the good in round t, and $x_{i,t}$ is the fraction of good t that agent i is allocated.They only study per-round constraints, so one of our contributions is a generalization of their result to the budgeted setting.
3. *Batched private goods.* When $L = L' \cdot N$, $B = T$, and $v_{i,\ell,t} = 0$ if $i \not\equiv j \pmod{N}$, we capture a setting where there are L' private goods arriving in each round, and the algorithm can (fractionally) allocate at most 1 good in total among the agents in each round.

5.1 The Set-Aside Greedy Algorithm for Batched Public Goods

We present an algorithm, Algorithm 1, for the batched public goods model which generalizes our guarantees from Section 4 and, partly, from the work of Banerjee et al. [23] as well. First, we need to extend predicted promised utilities to the batched model.

Definition 5 (Predicted Promised Utility)**.** The predicted promised utility of agent i in round t is

$$\widetilde{u}_{i,t}(\mathbf{z}_1, \ldots, \mathbf{z}_t) = \frac{B}{2\min\{N, L\}T} \cdot \widetilde{V}_i + \sum_{\tau=1}^{t} \sum_{\ell=1}^{L} v_{i,\ell,\tau} z_{\ell,\tau} \tag{2}$$

For clarity, we will often omit the dependence on $\mathbf{z}_1, \ldots, \mathbf{z}_{t-1}$ and just write $\widetilde{u}_{i,t}(\mathbf{z}_t)$.

At a high level, Algorithm 1 takes in as input a target approximation factor α (which we set later) and proceeds as follows. In round t, let F_t be the set of 'favorite goods', which satisfies that for each agent i, there exists some good $\ell \in F_t$ such that ℓ is among the most preferred goods of i in round t. Note that this guarantees that $|F_t| \leqslant \min\{N, L\}$. The algorithm allocates $x_{\ell,t} = y_{\ell,t} + z_{\ell,t}$, where the set-aside allocation is $y_{\ell,t} = \frac{B}{2|F_t|T} \cdot \mathbb{1}\{\ell \in F_t\}$ for each $\ell \in [L]$ and the greedy allocation $\mathbf{z}_t$ is computed by an optimization problem. Note that the objective function maximized is concave and the constraints are linear. Hence, this is a convex optimization problem, which can be solved up to any $\epsilon > 0$ accuracy in time that is polynomial in the input size and $1/\epsilon$. For simplicity, we present our proof assuming that the optimization problem is solved exactly, but it is very easy to see that solving it up to a constant error does not change our proportional fairness guarantee asymptotically.

The purpose of the $\mathbf{y}_t$ part of the allocation is to guarantee each agent a minimum share of their total utility V_i: Note that since $|F_t| \leqslant \min\{N, L\}$, every agent i is guaranteed to receive at least

ALGORITHM 1: Set-Aside Greedy Algorithm for Batched Public Goods

Input: Target proportional-fairness level α; total value predictions $(\widetilde{V}_i)_{i\in[N]}$

Define $q = \frac{\alpha}{2B}$.
for all $t = 1$ to T **do**
  Given values $\{v_{i,\ell,t}\}_{i\in[N],\ell\in[L]}$, find 'favorite goods' set F_t as follows: Initialize $F_t \leftarrow \emptyset$.
  for all $i = 1$ to N **do**
    Let $\ell_i \leftarrow \arg\max_\ell\{v_{i,\ell,t}\}$ (breaking ties arbitrarily), and update $F_t \leftarrow F_t \cup \{\ell_i\}$.
  end for
  Set-aside semi-allocation: set $y_{\ell,t} = \frac{B}{2|F_t|T} \cdot \mathbb{1}\{\ell \in F_t\}$.
  Greedy semi-allocation: Let $(\mathbf{z}_t, \lambda_t)$ be an optimal solution to the following optimization problem: Maximize $\frac{1}{N}\sum_{i=1}^N \ln(\widetilde{u}_{i,t}(\mathbf{z})) + \lambda q$ subject to $\sum_{\ell=1}^L z_\ell + \lambda = 1 - \frac{B}{2T}$ and $\mathbf{z}, \lambda \geqslant 0$.
  Allocate $x_{\ell,t} = y_{\ell,t} + z_{\ell,t}$.
end for

$\max_\ell\{v_{i,\ell,t}\} \cdot \frac{B}{2\min\{N,L\}T}$ utility from the $\mathbf{y}_t$ set-aside semi-allocation in each round t. Summing over all T rounds, each agent is guaranteed to receive at least $V_i \cdot \frac{B}{2\min\{N,L\}T}$ from the set-aside semi-allocation. Algorithm 1 achieves the following guarantee; the proof is in the appendix.

Theorem 5. *In the batched public goods model, for any* $\alpha \geqslant 4\ln\left(\frac{2\min\{N,L\}T}{B}\right) + \frac{4}{N}\sum_i \ln(d_i)$*, Algorithm 1 produces a feasible allocation* $\mathbf{x}$*, which satisfies* $\max_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^N \frac{u_i(\mathbf{w})}{c_i \cdot u_i(\mathbf{x})} \leqslant \alpha$*, where the maximum is over all feasible allocations* $\mathbf{w}$*.*

Using exactly the same proof as that of Corollary 1, we obtain the following guarantees for Algorithm 1 with respect to proportional fairness, the core, and the Nash social welfare.

Corollary 2. *For* $\alpha \geqslant 4\ln\left(\frac{2\min\{N,L\}T}{B}\right) + \frac{4}{N}\sum_i \ln(d_i)$*, Algorithm 1 is*

1. $(\alpha \cdot \max_i c_i)$*-proportionally fair, and hence in the* $(\alpha \cdot \max_i c_i)$*-core.*
2. *achieves* $(\alpha \cdot (\prod_{i\in[N]} c_i)^{\frac{1}{N}})$*-approximation of the Nash social welfare.*

Recall that the private goods setting of Banerjee et al. [23] is a special case of our setting in which $L = N$, $B = T$, and the valuation matrix of the N agents for the N public goods in each round is a diagonal matrix. For this special case, our Nash welfare approximation is $O((\prod_{i\in[N]} c_i)^{1/N} \cdot (\ln N + (1/N) \cdot \sum_i \ln d_i))$. The Nash welfare approximation obtained by Banerjee et al. [23] is almost the same, except that $\ln N$ is replaced by $\min\{\ln N, \ln T\}$. Thus, for $T = \Omega(N)$, our result strictly generalizes theirs and they prove this bound to be almost tight. For $T = o(N)$, they derive a better approximation that depends on $\ln T$ instead of $\ln N$, (and it is unknown if this is tight). It would be interesting to see if our result can also be improved in this case.

6 Discussion and Future Directions

The Geometry of Online Fair Allocation. While we focus on the proportional fairness objective, a natural question is whether our results can be extended to other non-linear objective functions, such as the class of generalized p-mean welfare measures [39, 51]. Similarly, while our results give insight into the interaction between per-round budget constraints and overall budget constraints, one can ask if these ideas can be extended to more complex constraints such as general packing constraints [31].

Alternate Information Structures. An interesting direction for future research is to develop a better understanding regarding whether what types of predictions would be the most appropriate for online fair division. For example, what if the algorithm is also provided with a prediction regarding the total value of each good across all agents, but not specifying which of the agents will like it and by how much? Could this additional information allow us to overcome the logarithmic lower bounds and maybe even achieve a constant approximation? Would it help to know more detailed patterns about agent valuations — for example, if they take discrete values, or are periodic, or have low variance?

Alternate Arrival Models. A related question to the one above is in regards to the process for generating agent valuations. In our model, the values $V = (v_{i,t})_{i\in[N],t\in[T]}$ are allowed to arrive in an adversarial order. What if we consider slightly more optimistic models such as the random-order model or the stochastic model [1]? It is possible that predictions are not needed when studying these arrival models since they inherently already give the algorithm some information about the input.

Incentives. Finally, one of the most important open questions is how to convert online algorithms to online mechanisms, whereby agents report their private $v_{i,t}$ values in each round. In such a setting, can we design mechanisms which are incentive compatible? There has been recent work on similar questions for static fair allocation [52–55], as well as online welfare maximization [56–59]; extending these ideas to online fair allocation is an important open problem.

References

[1] Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis*. cambridge university press, 2005.

[2] Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. In Tim Roughgarden, editor, *Beyond the Worst-Case Analysis of Algorithms*, page 646–662. Cambridge University Press, 2021.

[3] Thodoris Lykouris and Sergei Vassilvtiskii. Competitive caching with machine learned advice. In *International Conference on Machine Learning*, pages 3296–3305. PMLR, 2018.

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

[5] Zhihao Jiang, Debmalya Panigrahi, and Kevin Sun. Online algorithms for weighted paging with predictions. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[6] Paul Dütting, Silvio Lattanzi, Renato Paes Leme, and Sergei Vassilvitskii. Secretaries with advice. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 409–429, 2021.

[7] Antonios Antoniadis, Christian Coester, Marek Elias, Adam Polak, and Bertrand Simon. Online metric algorithms with untrusted predictions. In *International Conference on Machine Learning*, pages 345–355. PMLR, 2020.

[8] Antonios Antoniadis, Themis Gouleakis, Pieter Kleer, and Pavel Kolev. Secretary and online matching problems with machine learned advice. *Advances in Neural Information Processing Systems*, 33:7933–7944, 2020.

[9] Silvio Lattanzi, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. Online scheduling via learned weights. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1859–1877. SIAM, 2020.

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

[11] Shufan Wang, Jian Li, and Shiqiang Wang. Online algorithms for multi-shop ski rental with machine learned advice. *Advances in Neural Information Processing Systems*, 33:8150–8160, 2020.

[12] Shom Banerjee. Improving online rent-or-buy algorithms with sequential decision making and ml predictions. *Advances in Neural Information Processing Systems*, 33:21072–21080, 2020.

[13] Étienne Bamas, Andreas Maggiori, and Ola Svensson. The primal-dual method for learning augmented algorithms. *Advances in Neural Information Processing Systems*, 33:20083–20094, 2020.

[14] Matteo Almanza, Flavio Chierichetti, Silvio Lattanzi, Alessandro Panconesi, and Giuseppe Re. Online facility location with multiple advice. *Advances in Neural Information Processing Systems*, 34, 2021.

[15] Antonios Antoniadis, Christian Coester, Marek Elias, Adam Polak, and Bertrand Simon. Learning-augmented dynamic power management with multiple states via new ski rental bounds. *Advances in Neural Information Processing Systems*, 34, 2021.

[16] Michael Dinitz, Sungjin Im, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. Faster matchings via learned duals. *Advances in Neural Information Processing Systems*, 34, 2021.

[17] Chenkai Yu, Guanya Shi, Soon-Jo Chung, Yisong Yue, and Adam Wierman. The power of predictions in online control. *Advances in Neural Information Processing Systems*, 33: 1994–2004, 2020.

[18] Bo Sun, Russell Lee, Mohammad Hajiesmaili, Adam Wierman, and Danny Tsang. Pareto-optimal learning-augmented algorithms for online conversion problems. *Advances in Neural Information Processing Systems*, 34, 2021.

[19] Sungjin Im, Ravi Kumar, Mahshid Montazer Qaem, and Manish Purohit. Online knapsack with frequency predictions. *Advances in Neural Information Processing Systems*, 34, 2021.

[20] Étienne Bamas, Andreas Maggiori, Lars Rohwedder, and Ola Svensson. Learning augmented energy minimization via speed scaling. *Advances in Neural Information Processing Systems*, 33:15350–15359, 2020.

[21] Keerti Anand, Rong Ge, Amit Kumar, and Debmalya Panigrahi. A regression approach to learning-augmented online algorithms. *Advances in Neural Information Processing Systems*, 34, 2021.

[22] Yingying Li, Xin Chen, and Na Li. Online optimal control with linear dynamics and predictions: Algorithms and regret analysis. *Advances in Neural Information Processing Systems*, 32, 2019.

[23] Siddhartha Banerjee, Vasilis Gkatzelis, Artur Gorokh, and Billy Jin. Online nash social welfare maximization with predictions. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022*. SIAM, 2022.

[24] Hugo Steinhaus. The problem of fair division. *Econometrica*, 16:101–104, 1948.

[25] J. Nash. The bargaining problem. *Econometrica*, 18(2):155–162, April 1950.

[26] Duncan K Foley. Lindahl's solution and the core of an economy with public goods. *Econometrica: Journal of the Econometric Society*, pages 66–72, 1970.

[27] Brandon Fain, Ashish Goel, and Kamesh Munagala. The core of the participatory budgeting problem. In *International Conference on Web and Internet Economics*, pages 384–399. Springer, 2016.

[28] Mayuresh Kunjir, Brandon Fain, Kamesh Munagala, and Shivnath Babu. ROBUS: fair cache allocation for data-parallel workloads. In Semih Salihoglu, Wenchao Zhou, Rada Chirkova, Jun Yang, and Dan Suciu, editors, *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017*, pages 219–234. ACM, 2017.

[29] Eric J. Friedman, Vasilis Gkatzelis, Christos-Alexandros Psomas, and Scott Shenker. Fair and efficient memory sharing: Confronting free riders. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 1965–1972. AAAI Press, 2019.

[30] Vincent Conitzer, Rupert Freeman, and Nisarg Shah. Fair public decision making. In Constantinos Daskalakis, Moshe Babaioff, and Hervé Moulin, editors, *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, pages 629–646. ACM, 2017.

[31] Brandon Fain, Kamesh Munagala, and Nisarg Shah. Fair allocation of indivisible public goods. In Éva Tardos, Edith Elkind, and Rakesh Vohra, editors, *Proceedings of the 2018 ACM Conference on Economics and Computation, Ithaca, NY, USA, June 18-22, 2018*, pages 575–592. ACM, 2018.

[32] Dominik Peters, Grzegorz Pierczynski, and Piotr Skowron. Proportional participatory budgeting with cardinal utilities. *CoRR*, abs/2008.13276, 2020.

[33] Kamesh Munagala, Yiheng Shen, Kangning Wang, and Zhiyi Wang. Approximate core for committee selection via multilinear extension and market clearing. *CoRR*, abs/2110.12499, 2021.

[34] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors. *Fair Allocation*, page 259–260. Cambridge University Press, 2016.

[35] Rupert Freeman, Seyed Majid Zahedi, and Vincent Conitzer. Fair and efficient social choice in dynamic settings. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4580–4587. ijcai.org, 2017.

[36] Gerdus Benade, Aleksandr M. Kazachkov, Ariel D. Procaccia, and Christos-Alexandros Psomas. How to make envy vanish over time. In Éva Tardos, Edith Elkind, and Rakesh Vohra, editors, *Proceedings of the 2018 ACM Conference on Economics and Computation, Ithaca, NY, USA, June 18-22, 2018*, pages 593–610. ACM, 2018.

[37] David Zeng and Alexandros Psomas. Fairness-efficiency tradeoffs in dynamic fair division. In Péter Biró, Jason Hartline, Michael Ostrovsky, and Ariel D. Procaccia, editors, *EC '20: The 21st ACM Conference on Economics and Computation, Virtual Event, Hungary, July 13-17, 2020*, pages 911–912. ACM, 2020.

[38] Vasilis Gkatzelis, Alexandros Psomas, and Xizhi Tan. Fair and efficient online allocations with normalized valuations. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*. AAAI Press, 2021.

[39] Siddharth Barman, Arindam Khan, and Arnab Maiti. Universal and tight online algorithms for generalized-mean welfare. In *The Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*. AAAI Press, 2022.

[40] Anna Bogomolnaia, Hervé Moulin, and Fedor Sandomirskiy. A simple online fair division problem. *CoRR*, abs/1903.10361, 2019. URL http://arxiv.org/abs/1903.10361.

[41] Anna Bogomolnaia, Hervé Moulin, and Richard Stong. Collective choice under dichotomous preferences. *Journal of Economic Theory*, 122(2):165–184, 2005.

[42] Haris Aziz, Anna Bogomolnaia, and Hervé Moulin. Fair mixing: the case of dichotomous preferences. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*, pages 753–781, 2019.

[43] Conal Duddy. Fair sharing under dichotomous preferences. *Mathematical Social Sciences*, 73: 1–5, 2015.

[44] Florian Brandl, Felix Brandt, Dominik Peters, and Christian Stricker. Distribution rules under dichotomous preferences: Two out of three ain't bad. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, pages 158–179, 2021.

[45] Soroush Ebadian, Anson Kahng, Dominik Peters, and Nisarg Shah. Optimized distortion and proportional fairness in voting. In *Proceedings of the 23rd ACM Conference on Economics and Computation (EC)*, 2022. Forthcoming.

[46] Haris Aziz and Nisarg Shah. Participatory budgeting: Models and approaches. In *Pathways Between Social Science and Computational Social Science*, pages 215–236. Springer, 2021.

[47] Niv Buchbinder and Joseph Naor. *The design of competitive online algorithms via a primal-dual approach*. Now Publishers Inc, 2009.

[48] Nikhil R Devanur and Kamal Jain. Online matching with concave returns. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 137–144, 2012.

[49] Yossi Azar, Niv Buchbinder, and Kamal Jain. How to allocate goods in an online market? *Algorithmica*, 74(2):589–601, 2016.

[50] Frank P Kelly, Aman K Maulloo, and David Kim Hong Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research society*, 49(3):237–252, 1998.

[51] Soroush Ebadian, Rupert Freeman, and Nisarg Shah. Efficient resource allocation with secretive agents. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, 2022. Forthcoming.

[52] Simina Brânzei, Vasilis Gkatzelis, and Ruta Mehta. Nash social welfare approximation for strategic agents. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 611–628. ACM, 2017.

[53] Georgios Amanatidis, Georgios Birmpas, George Christodoulou, and Evangelos Markakis. Truthful allocation mechanisms without payments: Characterization and implications on fairness. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 545–562, 2017.

[54] Daniel Halpern, Ariel D Procaccia, Alexandros Psomas, and Nisarg Shah. Fair division with binary valuations: One rule to rule them all. In *International Conference on Web and Internet Economics*, pages 370–383. Springer, 2020.

[55] Haris Aziz, Bo Li, and Xiaowei Wu. Strategyproof and approximately maxmin fair share allocation of chores. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 60–66, 2019.

[56] Mingyu Guo, Vincent Conitzer, and Daniel M Reeves. Competitive repeated allocation without payments. In *WINE 2009*, pages 244–255. Springer, 2009.

[57] Santiago R Balseiro, Vahab Mirrokni, Renato Paes Leme, and Song Zuo. Dynamic double auctions: Towards first best. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 157–172. SIAM, 2019.

[58] Artur Gorokh, Siddhartha Banerjee, and Krishnamurthy Iyer. From monetary to non-monetary mechanism design via artificial currencies. In *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, pages 563–564, 2017.

[59] Artur Gorokh, Siddhartha Banerjee, and Krishnamurthy Iyer. The remarkable robustness of the repeated fisher market. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 562–562, 2021.

Appendix

A Implications of Proportional Fairness

We remark that proportional fairness implies two other prominent fairness notions.

Proportional fairness implies the core. For $\alpha \geqslant 1$, allocation $\mathbf{x}$ is said to be in the α-core if there is no subset of agents S and allocation $\mathbf{w}$ such that $\frac{|S|}{N} \cdot u_i(\mathbf{w}) \geqslant \alpha \cdot u_i(\mathbf{x})$ for all $i \in S$ and at least one of these inequalities is strict. We say that an online algorithm is α-core if it always produces an allocation in the α-core. The following is a well-known relation between proportional fairness and the core.

Proposition 2. *For $\alpha \geqslant 1$, every α-proportionally fair allocation is in the α-core.*

Proof. If an α-proportionally fair allocation $\mathbf{x}$ is not in the α-core, then by definition, there exists a subset of agents S and an allocation $\mathbf{w}$ such that $\frac{|S|}{N} u_i(\mathbf{w}) \geqslant \alpha \cdot u_i(\mathbf{x})$ for all $i \in S$ and at least one of these inequalities is strict. Then, $\frac{u_i(\mathbf{w})}{u_i(\mathbf{x})} \geqslant \alpha \cdot \frac{N}{|S|}$ for all $i \in S$ and at least one of these inequalities is strict, which implies

$$\sum_{i \in S} \frac{u_i(\mathbf{w})}{u_i(\mathbf{x})} > \alpha \cdot N \Rightarrow \frac{1}{N} \sum_{i=1}^{N} \frac{u_i(\mathbf{w})}{u_i(\mathbf{x})} \geqslant \frac{1}{N} \sum_{i \in S} \frac{u_i(\mathbf{w})}{u_i(\mathbf{x})} > \alpha,$$

contradicting the fact that $\mathbf{x}$ is α-proportionally fair. □

Proportional fairness implies optimal Nash welfare. A common objective function studied in multi-agent systems is the Nash social welfare, which aggregates individual agent utilities into a collective measure by taking the geometric mean. That is, the Nash social welfare of allocation $\mathbf{x}$ is given by $\mathsf{NSW}(\mathbf{x}) = \left(\prod_{i=1}^{N} u_i(\mathbf{x})\right)^{1/N}$. For $\alpha \geqslant 1$, we say that allocation $\mathbf{x}$ achieves an α-approximation of the Nash welfare if $\frac{\mathsf{NSW}(\mathbf{w})}{\mathsf{NSW}(\mathbf{x})} \leqslant \alpha$ for all allocations $\mathbf{w}$. We say that an online algorithm achieves an α-approximation of the Nash welfare if it always produces an allocation that achieves an α-approximation of the Nash welfare. It is also well-known that α-proportional fairness implies an α-approximation of the Nash welfare (in particular, a proportionally fair allocation has the maximum possible Nash welfare).

Proposition 3. *For $\alpha \geqslant 1$, if allocation $\mathbf{x}$ is α-proportionally fair, then $\mathbf{x}$ achieves an α-approximation of the Nash welfare.*

Proof. Take any allocation $\mathbf{w}$. The result follows by observing that

$$\frac{\mathsf{NSW}(\mathbf{w})}{\mathsf{NSW}(\mathbf{x})} = \left(\prod_{i=1}^{N} \frac{u_i(\mathbf{w})}{u_i(\mathbf{x})}\right)^{1/N} \leqslant \frac{1}{N} \sum_{i=1}^{N} \frac{u_i(\mathbf{w})}{u_i(\mathbf{x})} \leqslant \alpha,$$

where the second transition is the AM-GM inequality. □

We remark that the upper bounds derived in this work hold for the stronger notion of proportional fairness, while the lower bounds hold even for the weaker notion of Nash welfare approximation.

B Algorithm for Binary Values and Unit Budget

ALGORITHM 2: Set-Aside Greedy Algorithm for Binary Values and Unit Budget

Input: Target proportional-fairness level α

for all $t = 1$ to T **do**

  (Set-aside semi-allocation) If there exists $i \in [N]$ with $v_{i,t} = 1$ and $v_{i,\tau} = 0$ for each $\tau \in [t-1]$, then set $y_t = 1/(2N)$, else set $y_t = 0$.

  (Greedy semi-allocation) Compute $z_t = \min\left\{z_t : \frac{1}{N}\sum_{i=1}^{N} \frac{v_{it}}{\tilde{u}_{it}(z_t)} \leqslant \alpha,\ z_t \geqslant 0\right\}$.

  Allocate $x_t = y_t + z_t$ to good t.

end for

C Algorithm for General Values and Budget in the Non-batched Model

ALGORITHM 3: Set-Aside Greedy algorithm for the general setting

Input: Target threshold α; total value predictions $(\widetilde{V}_i)_{i\in[N]}$

for all $t = 1$ to T **do**

  Set-aside semi-allocation: set $y_t = \frac{B}{2T}$.

  Greedy semi-allocation: $z_t = \min\{z_t^*, 1 - y_t\}$, where $z_t^* = \min\left\{z \geqslant 0 : \frac{1}{N}\sum_{i=1}^{N} \frac{v_{i,t}}{\tilde{u}_{i,t}(z)} \leqslant \frac{\alpha}{2B}\right\}$.

  Allocate $x_t = y_t + z_t$.

end for

D Proof of Theorem 1

Before we prove the theorem, we need the following technical lemma. Recall that for $k \in \mathbb{N}$, H_k is the k-th harmonic number.

Lemma 1. *For every $S \in \mathbb{N}$, $W \geqslant 1$, and $\mathbf{y} = (y_\ell)_{\ell\in[S]} \in [0,1]^{[S]}$ with $\sum_{\ell=1}^{S} y_\ell \leqslant W$, we have*

$$\max_{\ell\in[S]} \frac{\ell+1}{W + \sum_{j=1}^{\ell} j\cdot y_j} \geqslant \frac{H_S}{2W}.$$

Proof. Suppose for contradiction that there exist $S \in \mathbb{N}$, $W \geqslant 1$, and $\mathbf{y} = (y_\ell)_{\ell\in[S]} \in [0,1]^{[S]}$ such that $\sum_{\ell=1}^{S} y_\ell \leqslant W$ and, for every $\ell \in [S]$,

$$\frac{\ell+1}{W + \sum_{j=1}^{\ell} j\cdot y_j} < \frac{H_S}{2W} \Rightarrow \textstyle\sum_{j=1}^{\ell} j\cdot y_j > \dfrac{2W\cdot(\ell+1)}{H_S} - W.$$

Dividing the above equation by $\ell\cdot(\ell+1)$ and summing over $\ell \in [S]$, we have that

$$\sum_{\ell=1}^{S}\sum_{j=1}^{\ell} \frac{j\cdot y_j}{\ell\cdot(\ell+1)} > \sum_{\ell=1}^{S}\left(\frac{2W}{H_S\cdot\ell} - \frac{W}{\ell\cdot(\ell+1)}\right). \tag{3}$$

Let us analyze the LHS in Equation (3) by exchanging the order of summations. We have

$$\begin{aligned}\text{LHS} &= \sum_{j=1}^{S} j\cdot y_j\cdot\left(\sum_{\ell=j}^{S}\frac{1}{\ell\cdot(\ell+1)}\right) = \sum_{j=1}^{S} j\cdot y_j\cdot\left(\sum_{\ell=j}^{S}\frac{1}{\ell} - \frac{1}{\ell+1}\right)\\ &= \sum_{j=1}^{S} j\cdot y_j\cdot\left(\frac{1}{j} - \frac{1}{S+1}\right) \leqslant \sum_{j=1}^{S} y_j \leqslant W,\end{aligned}$$

where the third transition holds due to the telescopic sum.

t / $\mathbf{v}_t$	1	2	3	4	$5-8$	9	10	11	12	13 17	14 18	15 19	16 20	21 25 29 33	22 26 30 34	23 27 31 35	24 28 32 36
$v_{1,t}$	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0
$v_{2,t}$	1	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0
$v_{3,t}$	0	1	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0
$v_{4,t}$	0	0	1	1	0	0	0	0	1	0	0	0	1	0	0	0	1
	S_2				S_0	S_1				$2\times S_1$				$4\times S_1$			
	L_1					L'_2								L'_3			

t / $\mathbf{v}_t$	1	2	3	4	$5-8$	9	10	11	12	13 17	14 18	15 19	16 20	$21-24$	$25-36$
$v_{1,t}$	1	0	0	1	0	1	0	1	1	0	0	0	0	1	0
$v_{2,t}$	1	1	0	0	0	1	1	0	1	0	0	0	0	1	0
$v_{3,t}$	0	1	1	0	0	1	1	1	0	0	0	0	0	1	0
$v_{4,t}$	0	0	1	1	0	0	1	1	1	0	0	0	0	1	0
	S_2				S_0	S_3				$2\times S_1$				S_4	$3\times S_0$
	L_1					L_2								L_3	

Figure 1: Instances I_1 (top) and I_3 (bottom) when $N = 4$.

Now, let us analyze the RHS in Equation (3) using the same telescopic sum.

$$\text{RHS} = \frac{2W}{H_S} \sum_{\ell=1}^{S} \frac{1}{\ell} - W \cdot \sum_{\ell=1}^{S} \left(\frac{1}{\ell} - \frac{1}{\ell+1} \right) = \frac{2W}{H_S} \cdot H_S - W \cdot \left(1 - \frac{1}{S+1} \right) \geqslant W.$$

Hence, we proved that in Equation (3), the LHS is at most W and the RHS is at least W. But the equation proves the LHS to be strictly greater than the RHS, which is the desired contradiction. $\square$

Now, we are ready to prove Theorem 1.

Proof of Theorem 1. We prove that the Nash social welfare approximation of every online algorithm is at least $H_N/2 = \Omega(\log N)$. Suppose for contradiction that there is an online algorithm with a smaller approximation ratio.

Set $T = N \cdot (N^2 + N - 2)/2$. We use $\mathbf{v}_t = (v_{i,t})_{i \in [N]}$ to denote the vector of values of all the agents for good $t \in [T]$. For $r \in [N] \cup \{0\}$, we denote with S_r a sequence of N goods that arrive consecutively for which the agents have the following valuations. For the first good in the sequence, $v_{i,1} = 1$ for $i \leqslant r$ and $v_{i,1} = 0$ for $i \geqslant r + 1$. For $t \in \{2, \ldots, N\}$, $\mathbf{v}_t$ is obtained by cyclically permuting $\mathbf{v}_{t-1}$ to the right. For example, if $N = 3$, then S_2 consists of 3 goods that arrive consecutively with $\mathbf{v}_1 = (1, 1, 0)$, $\mathbf{v}_2 = (0, 1, 1)$, and $\mathbf{v}_3 = (1, 0, 1)$. Notice that S_r consists of N goods such that each good is liked by r agents and each agent likes r goods. Also, notice that S_0 consists of N goods for which all agents have zero value.

Next, we use S_r to construct two building blocks of our adversarial instances.

- For $r \in [N-1]$, let L_r be a sequence of consecutive goods constructed by concatenating S_{r+1} followed by r copies of S_0.
- For $r \in [N-1]$, let L'_r be a sequence of consecutive goods constructed by concatenating $r + 1$ copies of S_1.

Notice that for each $r \in [N-1]$, L_r and L'_r both consist of $(r + 1) \cdot N$ goods of which each agent likes exactly $r + 1$ goods.

Finally, for each $k \in [N-1]$, define instance $I_k = (L_1, L_2, \ldots, L_k, L'_{k+1}, \ldots, L'_{N-1})$, i.e., instance I_k is constructed by concatenating $L_1, L_2, \ldots, L_k, L'_{k+1}, \ldots, L'_{N-1}$ in that order. Figure 1 shows instances I_1 and I_3 with $N = 4$. Notice that for each $k \in [N-1]$, instance I_k consists of a total of T goods of which each agent likes $(N^2 + N - 2)/2$ goods. We let the algorithm know in advance that it will see instance I_k for some $k \in [N-1]$, and prove that it still cannot achieve proportional fairness better than $H_N/2$.

With slight abuse of notation, for $r \in [N-1]$, let L_r also denote the set of goods that appear in the sequence L_r. Let $\mathbf{x}$ denote the allocation produced by the online algorithm on instance I_{N-1}, i.e., when the algorithm observes $L_1, L_2, \ldots, L_{N-1}$. For $k \in [N-1]$, let $x_{L_k} = \sum_{t \in L_k} x_t$ denote the total allocation to goods in L_k.

Now, fix $k \in [N-1]$ and let $\mathbf{x}^k$ denote the allocation produced by the algorithm on instance I_k with values $v_{i,t}$. Because the algorithm cannot distinguish between I_k and I_{N-1} for the first k blocks (namely, $L_1, \ldots, L_k$), even under instance I_k, the algorithm must assign a total allocation of x_{L_ℓ} to block L_ℓ for each $\ell \leqslant k$. Hence, we have

$$\begin{aligned}
\mathsf{NSW}(\mathbf{x}^k) &= \left(\prod_{i=1}^{N}\left(\sum_{\ell=1}^{k}\sum_{t\in L_\ell} v_{i,t}\cdot x_t^k + \sum_{\ell=k+1}^{N-1}\sum_{t\in L'_\ell} v_{i,t}\cdot x_t^k\right)\right)^{1/N} \\
&\leqslant \frac{1}{N}\cdot\left(\sum_{i=1}^{N}\left(\sum_{\ell=1}^{k}\sum_{t\in L_\ell} v_{i,t}\cdot x_t^k + \sum_{\ell=k+1}^{N-1}\sum_{t\in L'_\ell} v_{i,t}\cdot x_t^k\right)\right) \\
&= \frac{1}{N}\cdot\left(\sum_{\ell=1}^{k}\sum_{t\in L_\ell}\sum_{i=1}^{N} v_{i,t}\cdot x_t^k + \sum_{\ell=k+1}^{N-1}\sum_{t\in L'_\ell}\sum_{i=1}^{N} v_{i,t}\cdot x_t^k\right) \\
&\leqslant \frac{1}{N}\cdot\left(\sum_{\ell=1}^{k}\sum_{t\in L_\ell}(\ell+1)\cdot x_t^k + \sum_{\ell=k+1}^{N-1}\sum_{j\in L'_\ell} x_t^k\right) \\
&\leqslant \frac{1}{N}\cdot\left(\sum_{\ell=1}^{k}(\ell+1)\cdot x_{L_\ell} + 1 - \sum_{\ell=1}^{k} x_{L_\ell}\right)
\end{aligned}$$

where the second transition follows from the AM-GM inequality and the fourth transition follows because, for each $\ell \in [N-1]$, each good $t \in L_\ell$ is liked by at most $\ell + 1$ agents and each good $t \in L'_\ell$ is liked by a single agent.

Consider an alternative allocation $\mathbf{y}^k$ which allocates $1/N$ to each of the first N goods of L_k (i.e., goods of its S_{k+1} portion). This provides each agent utility equal to $(k+1)/N$, thus achieving Nash social welfare equal to $(k+1)/N$.

Hence, the algorithm's approximation ratio α_k on instance I_k satisfies

$$\alpha_k \geqslant \frac{k+1}{\sum_{\ell=1}^{k}(\ell+1)\cdot x_{L_\ell} + 1 - \sum_{\ell=1}^{k} x_{L_\ell}} \geqslant \frac{k+1}{1 + \sum_{\ell=1}^{k}(\ell+1)\cdot x_{L_\ell}}.$$

Hence, the worst-case approximation ratio is at least $\max_{k\in[N-1]} \alpha_k$. Applying Lemma 1 with $S = N$ and $W = 1$, we get that this is at least $H_N/2$, as desired.[5] □

E Proof of Theorem 2

Proof. Let $\mathbf{x}$ denote Algorithm 2's final allocation, and consider the following LP and its dual:

$$(P) \qquad \max_{\mathbf{w}\in\mathbb{R}_+^T} \frac{1}{N}\sum_{i=1}^{N}\frac{u_i(\mathbf{w})}{u_i(\mathbf{x})} \qquad\qquad (D) \qquad \min_{p\in\mathbb{R}_+} p$$

$$\text{s.t. } \sum_{t=1}^{T} w_t \leqslant 1 \qquad\qquad \text{s.t. } p \geqslant \frac{1}{N}\sum_{i=1}^{N}\frac{v_{i,t}}{u_i(\mathbf{x})} \quad \forall t \in [T]$$

[5]We can set $y_1 = 0$ and $y_\ell = x_{L_{\ell-1}}$ for all $\ell \in \{2, \ldots, N\}$ when applying the lemma.

By construction, γ is feasible to the dual LP, because

$$\gamma \overset{(a)}{\geqslant} \frac{1}{N}\sum_{i=1}^{N}\frac{v_{it}}{\widetilde{u}_{it}(z_t)} \overset{(b)}{\geqslant} \max_i \frac{1}{N}\sum_{i=1}^{N}\frac{v_{it}}{u_i(\mathbf{x})},$$

where (a) is by definition of the algorithm, and (b) is because promised utilities are a lower bound on the true final utility. Thus by weak duality,

$$\max_{\mathbf{w}\in\mathbb{R}_+^T} \frac{1}{N}\sum_{i=1}^{N}\frac{u_i(\mathbf{w})}{u_i(\mathbf{x})} \leqslant \gamma = 2\ln(2N).$$

To complete the proof, it remains to show that $\mathbf{x}$ is a feasible allocation. Since $\sum_{t=1}^T y_t \leqslant \frac{1}{2}$, it suffices to show that $\sum_{t=1}^T z_t \leqslant \frac{1}{2}$. Now we have

$$\begin{aligned}
\gamma\sum_{t=1}^{T} z_t &= \sum_{t=1}^{T} z_t \cdot \frac{1}{N}\sum_{i=1}^{N}\frac{v_{it}}{\widetilde{u}_{it}(z_t)} && \Big(\text{since } z_t > 0 \text{ implies } \gamma = \frac{1}{N}\sum_{i=1}^{N}\frac{v_{it}}{\widetilde{u}_{it}(z_t)}\Big)\\
&= \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\frac{z_t v_{it}}{\widetilde{u}_{it}(z_t)} = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\left(1-\frac{\widetilde{u}_{it}(0)}{\widetilde{u}_{it}(z_t)}\right)\\
&\leqslant \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\left[\ln(\widetilde{u}_{it}(z_t)) - \ln(\widetilde{u}_{it}(0))\right] && (\text{since } 1-x \leqslant -\ln(x)\,\forall\, x\in\mathbb{R})\\
&= \frac{1}{N}\sum_{i=1}^{N}\left[\ln(\widetilde{u}_{iT}(z_T)) - \ln(\widetilde{u}_{i1}(0))\right] = \frac{1}{N}\sum_{i=1}^{N}\left[\ln(\widetilde{u}_{iT}(z_T)) - \ln\left(\frac{1}{2N}\right)\right]\\
&= \frac{1}{N}\sum_{i=1}^{N}\ln\left(\widetilde{u}_i(\mathbf{x})\right) + \ln(2N) && (\text{since predicted utility lower bounds final utility})\\
&\leqslant \ln(2N). && (\text{since final utilities are at most 1})
\end{aligned}$$

Hence, in order for $\sum_{t=1}^T z_t \leqslant \frac{1}{2}$, it suffices to have $\gamma = 2\ln(2N)$. This is why we defined γ to be equal to this quantity in the algorithm. □

F Proof of Proposition 1

Proof. The hardness instances we consider will have $N = 1$ agent. In this case, the proportional fairness of an online algorithm on a given instance is equal to the maximum possible utility the agent could have obtained in hindsight divided by the utility she obtained under the algorithm; this is also equal to the approximation ratio for the Nash welfare. For clarity, we will omit the subscript i in the notation in this proof, so that v_t is the value that the agent has for good t, and $u(\mathbf{x})$ is the utility of the agent under allocation $\mathbf{x}$. With $N = 1$, a problem instance is defined by a sequence of T values $(v_1, v_2, \dots, v_T)$.

Consider the following family of T instances: for $t \in [T]$, instance $I_t := (1, M, \dots, M^{t-1}, 0, \dots, 0)$, where M is a sufficiently large number. The algorithm knows in advance that it will see instance I_t for some $t \in [T]$, and prove that it still cannot achieve proportional fairness better than $T/2B$.

Let $\mathbf{x} = (x_1, \dots, x_T)$ be the allocation produced by the algorithm on instance $I_T = (1, M, \dots, M^{T-1})$. For any $t \in [T]$, since the algorithm cannot distinguish between I_t and I_T up to round t, the allocation up to round t under instance I_t must also be $(x_1, \dots, x_t)$. Further, without loss of generality, we can assume that the algorithm allocates 0 in any round where the value is 0. Therefore, for each $t \in [T]$, the allocation produced by the algorithm on instance I_t is $(x_1, \dots, x_t, 0, \dots, 0)$.

We claim that in order to be $\frac{T}{2B}$-proportionally fair, the algorithm needs to set $x_t \geqslant \frac{B}{T}$ for all t. To see this, suppose to the contrary that $x_t < \frac{B}{T}$ for some t. Then on instance I_t, the agent's utility under the algorithm is

$$u(\mathbf{x}) = \sum_{\tau=1}^{t} x_\tau M^\tau < \sum_{\tau=1}^{t-1} M^\tau + \frac{B}{T}\cdot M^t < \frac{2B}{T}M^t,$$

where the last inequality holds when M is large enough. On the other hand, the hindsight-optimal allocation on I_t is to simply set $x_t^* = 1$, which gets utility M^t. Therefore, if $x_t < \frac{B}{T}$ then the algorithm cannot be $\frac{T}{2B}$-proportionally fair on I_t.

We have shown that $x_t \geqslant \frac{B}{T}$ for all t is necessary for the algorithm to be $\frac{T}{2B}$-proportionally fair. However, since the overall budget is B, the only way this can happen is if $x_t = \frac{B}{T}$ for all t. The same calculation as above shows that under this allocation, the algorithm is at most $(T/2B)$-proportionally fair, establishing the desired $\Omega(T/B)$ lower bound. □

G Proof of Theorem 3

As indicated in Section 4.2, we only need to prove that the greedy semi-allocations combinatorially computed in Algorithm 3 still satisfy the properties in Proposition 5. The result below states these properties for $L = 1$. Note that the first property in Proposition 5 becomes vacuous for $L = 1$.

Proposition 4 (Some Properties of the Algorithm)**.** *Consider the dual certificate:*

$$q = \frac{\alpha}{2B} \geqslant \frac{2}{B}\left[\ln\left(\frac{2T}{B}\right) + \frac{1}{N}\sum_i \ln(d_i)\right]$$

$$p_t = \max\left\{0, \frac{1}{N}\sum_{i=1}^{N} \frac{v_{i,t}}{\widetilde{u}_{i,t}(\mathbf{z}_t)} - q\right\}.$$

Then, the allocation $\mathbf{x} = \mathbf{y} + \mathbf{z}$ returned by Algorithm 3 satisfies the following:

1. $z_t < 1 - \frac{B}{2T} \implies p_t = 0$.
2. $z_t > 0 \implies p_t + q = \sum_{i=1}^{N} \frac{v_{i,t}}{\widetilde{u}_{i,t}(z_t)}$.

Proof. (1) follows because if $z_t < 1 - \frac{B}{2T}$, then that must mean $z_t = z_t^*$. Hence, $\frac{1}{N}\sum_{i=1}^{N} \frac{v_{i,t}}{\widetilde{u}_{i,t}(z_t)} \leqslant \frac{\alpha}{2B} = q$, which implies $p_t = 0$.

(2) follows because if $z_t > 0$, then that must mean $z_t^* > 0$. Let $\Phi(z) = \frac{1}{N}\sum_{i=1}^{N} \frac{v_{i,t}}{\widetilde{u}_{i,t}(z)}$. Since Φ is a continuous function in z, this implies $\Phi(z_t^*) = \frac{\alpha}{2B} = q$. Because $z_t = \min\{z_t^*, 1 - y_t\} \leqslant z_t^*$ and Φ is decreasing, we have $\Phi(z_t) \geqslant \Phi(z_t^*) = q$. Thus $p_t = \max\{0, \Phi(z_t) - q\} = \Phi(z_t) - q$. □

H Proof of Corollary 1

Proof. The proof is a straightforward generalization of Propositions 2 and 3, but for completeness we include it here.

(1) Theorem 3 implies that

$$\frac{1}{\max_i c_i} \cdot \max_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N} \frac{u_i(\mathbf{w})}{u_i(\mathbf{x})} \leqslant \max_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N} \frac{u_i(\mathbf{w})}{c_i u_i(\mathbf{x})} \leqslant \alpha,$$

which directly implies the desired proportional fairness guarantee and, by Proposition 2, the desired core guarantee. While the same guarantee carries over to Nash welfare approximation, repeating the proof of Proposition 3 actually provides a better approximation.

(2) Let $\mathbf{x}^*$ be the allocation maximizing the Nash social welfare in hindsight. We have

$$\max_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N} \frac{u_i(\mathbf{w})}{c_i u_i(\mathbf{x})} \geqslant \frac{1}{N}\sum_{i=1}^{N} \frac{u_i(\mathbf{x}^*)}{c_i u_i(\mathbf{x})} \overset{(a)}{\geqslant} \left(\prod_{i=1}^{N} \frac{u_i(\mathbf{x}^*)}{c_i u_i(\mathbf{x})}\right)^{\frac{1}{N}} \overset{(b)}{=} \left(\prod_{i=1}^{N} \frac{1}{c_i}\right)^{\frac{1}{N}} \frac{\mathsf{NSW}(\mathbf{x}^*)}{\mathsf{NSW}(\mathbf{x})}$$

where (a) is by the AM-GM inequality, and (b) is by the definition of the Nash welfare. This, together with Theorem 3, yields the second part of the corollary. □

I Proof of Theorem 4

Proof. As in the proof of Proposition 1, approximations to proportional fairness and Nash welfare are equivalent with $N = 1$ agent, both coinciding with the ratio of the agent's maximum utility in hindsight to the agent's utility under the algorithm.

For the sake of contradiction, suppose that that there exists an online algorithm whose approximation ratio is $o(\log(T/B))$. Let $T = B \cdot (T'(T'+1) - 2)/2$ for some T'. For $r \in [T'-1]$, we denote with S_r the sequence of $B \cdot (r+1)$ goods that arrive consecutively for which the agent has utility equal to $(r+1)/T'$ for the first B goods and utility equal to 0 for the remaining goods and with S'_r the sequence of $B \cdot (r+1)$ goods such that the agent has utility equal to $1/T'$ for each of the goods. Now, for each $k \in [T'-1]$, let $I_k = (S_1, S_2, \ldots, S_k, S'_{k+1}, \ldots, S'_{T'-1})$, i.e. I_k be the instance that is constructed by concatenating $S_1, S_2, \ldots S_k, S'_{k+1}, \ldots, S_{T'-1}$ in that order. Notice that for $r \in [T'-1]$, S_r and S'_r have the same number of goods, equal to $B(r+1)$, and for each of them the agent has the same accumulated utility, equal to $B(r+1)/T'$. Hence, each instance I_k consists of T goods and the agent has the same accumulated utility for all these instances. We assume that the algorithms is aware that it will see instance I_k for some $k \in [T'-1]$.

With slight abuse of notation, for $r \in [T'-1]$, we let S_r to also denote the set of goods that appear in the sequence S_r. We denote with $\mathbf{x}$ the allocation produced by the online algorithm on instance $I_{T'-1}$. For $r \in [T'-1]$, let $x_{S_r} = \sum_{t \in S_r} x_t$ denote the total allocation to goods in S_r. Now, for $k \in [T'-1]$, let $\mathbf{x}^k$ denote the allocation produced by the algorithm on instance I_k. As the algorithm cannot distinguish between I_k and $I_{T'-1}$ for the first k blocks (i.e. $S_1, \ldots, S_k$), under instance I_k, the algorithm must assign a total allocation of x_{S_ℓ} to block S_ℓ for each $\ell \leqslant k$.

Moreover, under instance I_k, the optimal algorithm allocates 1 to each of the first B goods of S_k. Hence, we have that under instance I_k, if α_k is the approximation ratio of the algorithm, then

$$\begin{aligned}\alpha_k &\geqslant \frac{B \cdot (k+1)/N}{(2 \cdot x_{S_1} + 3 \cdot x_{S_2} + \ldots + (k+1)x_{S_k} + (1 - x_{S_1} - x_{S_2} - \ldots - x_{S_k}))/N} \\ &= B \cdot \frac{(k+1)}{x_{S_1} + 2 \cdot x_{S_2} + \ldots + k x_{S_k} + 1}.\end{aligned}$$

Hence, if α is the worst-case approximation ratio over all the instances I_k, we have

$$\alpha \geqslant B \cdot \max_{k \in [T'-1]} \frac{(k+1)}{x_{S_1} + 2 \cdot x_{S_2} + \ldots + k x_{S_k} + 1}$$

and by applying Lemma 1, with $S = T'$ and $W = B$, we get that $\alpha \geqslant B \cdot H_{T'}/(2B)$. Thus, we have that $\alpha = \Omega(\log(T')) = \Omega(\log(T/B))$. □

J Proof of Theorem 5

Let $\mathbf{x}$ denote the final allocation produced by Algorithm 1. Central to our analysis is the following linear program (and its dual), which essentially tries to *maximize* (i.e., find the worst case) the proportional-fairness level achieved by Algorithm 1:

$$(P) \quad \max_{\mathbf{w} \in \mathbb{R}^{LT}} \frac{1}{N} \sum_{i=1}^{N} \frac{u_i(\mathbf{w})}{c_i u_i(\mathbf{x})}$$

$$\text{s.t.} \sum_{t=1}^{T} \sum_{\ell=1}^{L} w_{\ell,t} \leqslant B$$

$$\sum_{\ell=1}^{L} w_{\ell,t} \leqslant 1 \quad \forall t \in [T]$$

$$w_{\ell,t} \geqslant 0 \quad \forall \ell \in [L],\, t \in [T]$$

$$(D) \quad \min_{p \in \mathbb{R}^T,\, q \in \mathbb{R}} \sum_{t=1}^{T} p_t + Bq$$

$$\text{s.t. } p_t + q \geqslant \frac{1}{N} \sum_{i=1}^{N} \frac{v_{i,\ell,t}}{c_i u_i(\mathbf{x})} \quad \forall \ell \in [L],\, t \in [T]$$

$$p_t, q \geqslant 0$$

Note that the dual constraints can be written more compactly as

$$p_t + q \geqslant \frac{1}{N} \max_{\ell} \sum_{i=1}^{N} \frac{v_{i,\ell,t}}{c_i u_i(\mathbf{x})} \quad \forall t \in [T].$$

From now on we will work with the dual constraints in the form above.

Algorithm 1 can now be viewed as choosing $\mathbf{x}$ in a manner such that it forces the LP (P) to have as small a value as possible. One way to achieve this is to consider a 'target' dual solution $q = \frac{\alpha}{2B}$ and $p_t = \max\{0, \frac{1}{N}\max_\ell \sum_{i=1}^{N} \frac{v_{i,\ell,t}}{u_i(\mathbf{x})} - q\}$ (for some appropriate choice of α). To realize this, Algorithm 1 wants to set the vector of allocations $\mathbf{x}_t$ for each round t, as well as the corresponding dual variable p_t, in a manner such that the primal and dual solutions are consistent (i.e., all constraints are feasible, and they obey complementary slackness).

The challenge in generating allocation $\mathbf{x}$ and dual certificate (p_t, q) in an *online* fashion is that at time t, the dual constraint depends on $u_i(\mathbf{x})$, the final utilities under the *entire* allocation made by the algorithm. However, these quantities are not known to the algorithm at time t, because it cannot look into the future. To work around this, we use the predicted promised utilities (Definition 5) as proxy in place of $u_i(\mathbf{x})$ when making the decision in round t.

The next lemma explains the sense in which the predicted promised utilities serve as a proxy; in particular, we show that under Algorithm 1, the predicted promised utility $\widetilde{u}_{i,t}(\mathbf{z}_t)$ is a lower bound on the true final utility of agent i under $\mathbf{x}$ up to a multiplicative factor of c_i, for each agent i and in each round t.

Lemma 2. *For every agent $i \in [N]$ and time $t \in [T]$, we have $\widetilde{u}_{i,t}(\mathbf{z}_t) \leqslant c_i \cdot u_i(\mathbf{x})$.*

Proof. We have

$$\begin{aligned} u_i(\mathbf{x}) &= \sum_{\tau=1}^{T}\sum_{\ell=1}^{L} v_{i,\ell,\tau}(y_{\ell,\tau} + z_{\ell,\tau}) \\ &= \sum_{\tau=1}^{T}\sum_{\ell=1}^{L} v_{i,\ell,\tau} y_{\ell,\tau} + \sum_{\tau=1}^{T}\sum_{\ell=1}^{L} v_{i,\ell,\tau} z_{\ell,\tau} \end{aligned}$$

We next bound the contribution due to the set-aside allocations $\mathbf{y}$. At any time $t \in [T]$, agent i receives at least $\max_\ell\{v_{i,\ell,t}\}\, y_{\ell,t} = \max_\ell\{v_{i,\ell,t}\}\, \frac{B}{2|F_t|T}$ utility from the set-aside semi-allocation $y_{\ell,t}$. Since $|F_t| \leqslant \min\{N, L\}$, we get

$$\sum_{\tau=1}^{T}\sum_{\ell=1}^{L} v_{i,\ell,\tau} y_{\ell,\tau} \geqslant \sum_{\tau=1}^{T} \max_\ell\{v_{i,\ell,t}\} \cdot \frac{B}{2\min\{N,L\}T} \geqslant \frac{\widetilde{V}_i}{c_i} \cdot \frac{B}{2\min\{N,L\}T}.$$

Substituting this back in the above expression, and using the fact that $c_i \geqslant 1$ for all i, we have

$$\begin{aligned} u_i(\mathbf{x}) &\geqslant \frac{\widetilde{V}_i}{c_i} \cdot \frac{B}{2\min\{N,L\}T} + \sum_{\tau=1}^{t}\sum_{\ell=1}^{L} v_{i,\ell,\tau} z_{\ell,\tau} \\ &\geqslant \frac{1}{c_i}\left(\frac{B}{2\min\{N,L\}T}\,\widetilde{V}_i + \sum_{\tau=1}^{t}\sum_{\ell=1}^{L} v_{i,\ell,\tau} z_{\ell,\tau}\right) = \frac{1}{c_i}\,\widetilde{u}_{i,t}(\mathbf{z}_t). \end{aligned}$$

□

Using this, we now formally define our new dual certificates, and list some key properties of the allocation and certificate which we need for our performance guarantee. Define

$$\Phi_\ell(\mathbf{z}) = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{v_{i,\ell,t}}{\widetilde{u}_{i,t}(\mathbf{z})} = \sum_{i=1}^{N} \frac{v_{i,\ell,t}}{\gamma_i + \sum_{\ell=1}^{L} v_{i,\ell,t} z_{\ell,t}},$$

where $\gamma_i = \frac{B\widetilde{V}_i}{2\min\{N,L\}T} + \sum_{\tau=1}^{t-1}\sum_{\ell=1}^{L} v_{i,\ell,\tau} z_{\ell,\tau}$; for the last transition, see the definition of predicted promised utilities (Equation (2)).

Proposition 5 (Some Properties of the Algorithm)**.** *Consider the dual certificate:*

$$q = \frac{\alpha}{2B} \geqslant \frac{2}{B}\left[\ln\left(\frac{2\min\{N,L\}T}{B}\right) + \frac{1}{N}\sum_i \ln(d_i)\right]$$

$$p_t = \max\left\{0, \max_\ell \Phi_\ell(\mathbf{z}_t) - q\right\}$$

Then, the allocation $\mathbf{x} = \mathbf{y} + \mathbf{z}$ returned by Algorithm 1 satisfies the following for each $t \in [T]$:

1. *For each $\ell \in [L]$, $z_{\ell,t} > 0 \implies \Phi_\ell(\mathbf{z}_t) = \max_{\ell'} \Phi_{\ell'}(\mathbf{z}_t)$.*
2. $\sum_\ell z_{\ell,t} < 1 - \frac{B}{2T} \implies p_t = 0$.
3. $\sum_\ell z_{\ell,t} > 0 \implies p_t + q = \max_\ell \Phi_\ell(\mathbf{z}_t)$.

It is easy to argue the existence of a greedy semi-allocation $\mathbf{z}_t$ satisfying the three properties above using Kakutani's fixed point theorem. However, in Algorithm 1, we are able to constructively find such an allocation by maximizing a concave objective function subject to a budget-like constraint. To show that an optimal solution to this optimization problem satisfies the three properties in Proposition 5, we need the following technical lemma.

Proposition 6. *Let $f_1, \ldots, f_n : \mathbb{R}^n \to \mathbb{R}$, such that $f_i = \frac{\partial F}{\partial x_i}$ for some concave differentiable function $F : \mathbb{R}^n \to \mathbb{R}$. Then, the solution $\mathbf{x}^*$ to the optimization problem $\max\{F(\mathbf{x}) : \mathbf{x} \in \Delta^n\}$ satisfies, for all $i \in [n]$,*

$$x_i^* > 0 \implies f_i(\mathbf{x}^*) = \max_{j \in [n]} f_j(\mathbf{x}^*).$$

Proof. Written out in full form, the optimization problem is

$$\begin{aligned} \max \quad & F(\mathbf{x}) \\ \text{s.t.} \quad & x_1 + \cdots + x_n = 1 \\ & x_i \geqslant 0 \quad \forall\, i \in [n] \end{aligned}$$

Let λ be the Lagrangian multiplier for the equality constraint and $(\theta_i : i \in [n])$ be the multipliers for the non-negativity constraints. The Lagrangian is given by $\mathcal{L}(\mathbf{x}, \lambda, \boldsymbol{\theta}) = F(\mathbf{x}) + \lambda(\sum_{i\in[n]} x_i - 1) + \sum_{i\in[n]} \theta_i x_i$. The KKT conditions require that there exist $\lambda, \boldsymbol{\theta}$ satisfying the following constraints:

- (Primal feasibility) $\mathbf{x}^* \in \Delta^n$,
- (Dual feasibility) $\theta_i^* \geqslant 0$ for all $i \in [n]$,
- (Complementary slackness) $x_i^* \theta_i^* = 0$ for all $i \in [n]$,
- (Stationarity) $\frac{\partial F(\mathbf{x}^*)}{\partial x_i} - \lambda^* + \theta_i^* = 0$ for all $i \in [n]$.

Since $\frac{\partial F}{\partial x_i} = f_i$, the stationarity condition can be written as $f_i(\mathbf{x}^*) = \lambda^* - \theta_i^*$. This implies $f_j(\mathbf{x}^*) \leqslant \lambda^*$ for all j since $\boldsymbol{\theta}^* \geqslant 0$. On the other hand, consider some i with $x_i^* > 0$. By complementary slackness, $\theta_i^* = 0$, and hence $f_i(\mathbf{x}^*) = \lambda^*$. Thus

$$x_i^* > 0 \implies f_i(\mathbf{x}^*) = \lambda^* = \max_j f_j(\mathbf{x}^*),$$

which completes the proof. □

We are now ready to prove Proposition 5.

Proof of Proposition 5. Let $F(\mathbf{z}, \lambda)$ be the objective of the optimization problem for computing the greedy semi-allocation in Algorithm 1:

$$F(\mathbf{z}, \lambda) = \frac{1}{N} \sum_{i=1}^{N} \ln\left(\widetilde{u}_{i,t}(\mathbf{z})\right) + \lambda q.$$

Define $f_1, \ldots, f_{L+1} : \mathbb{R}^{L+1} \to \mathbb{R}$ as follows:

- $f_i(\mathbf{z}, \lambda) = \Phi_i(\mathbf{z})$ for $i = 1, \ldots, L$,
- $f_{L+1}(\mathbf{z}, \lambda) = q$.

It can be checked that F is concave and differentiable, $\frac{\partial F}{\partial z_i} = f_i$ for all $i \in [L]$, and that $\frac{\partial F}{\partial \lambda} = f_{L+1}$. Thus, applying Proposition 6,[6] the solution $(\mathbf{z}_t, \lambda_t)$ to the optimization problem satisfies

- $\lambda_t > 0$ implies $q \geqslant \max_j\{\Phi_j(\mathbf{z}_t)\}$, and
- For all $\ell \in [L]$, $z_{\ell,t} > 0$ implies $\Phi_\ell(\mathbf{z}_t) = \max\{q, \max_j \Phi_j(\mathbf{z}_t)\}$.

We now check that $\mathbf{z}_t$ satisfies the conditions in Proposition 6.

1. $\sum_{\ell=1}^{L} z_{\ell,t} \leqslant 1 - \frac{B}{2T}$: This follows directly from the constraints of the optimization problem.
2. $z_{\ell,t} > 0 \implies \Phi_\ell(\mathbf{z}_t) = \max_j\{\Phi_j(\mathbf{z}_t)\}$: This holds because $z_{\ell,t} > 0$ implies $\Phi_\ell(\mathbf{z}_t) = \max\{q, \max_j \Phi_j(\mathbf{z}_t)\}$. But since $\Phi_\ell(\mathbf{z}_t)$ is a term in $\max_j \Phi_j(\mathbf{z}_t)$, this means $\Phi_\ell(\mathbf{z}_t) = \max_j \Phi_j(\mathbf{z}_t)$.
3. $\sum_\ell z_{\ell,t} < 1 - \frac{B}{2T} \implies \max_j \Phi_j(\mathbf{z}_t) \leqslant q$: If $\sum_\ell z_{\ell,t} < 1 - \frac{B}{2T}$ then $\lambda_t > 0$, which implies $q \geqslant \max_j \Phi_j(\mathbf{z}_t)$.
4. $\sum_\ell z_{\ell,t} > 0 \implies q \leqslant \max_j \Phi_j(\mathbf{z}_t)$: If $\sum_\ell z_{\ell,t} > 0$ then $z_{\ell,t} > 0$ for some $\ell \in [L]$, which implies $\Phi_\ell(\mathbf{z}_t) = \max\{q, \max_j \Phi_j(\mathbf{z}_t)\}$. Thus $q \leqslant \max_j \Phi_j(\mathbf{z}_t)$.

This completes the proof. □

We can now prove our main performance guarantee for Algorithm 1, which we restate below.

Theorem 5. *In the batched public goods model, for any* $\alpha \geqslant 4 \ln\left(\frac{2\min\{N,L\}T}{B}\right) + \frac{4}{N}\sum_i \ln(d_i)$, *Algorithm 1 produces a feasible allocation* $\mathbf{x}$, *which satisfies* $\max_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N} \frac{u_i(\mathbf{w})}{c_i \cdot u_i(\mathbf{x})} \leqslant \alpha$, *where the maximum is over all feasible allocations* $\mathbf{w}$.

Proof. To prove the theorem, it suffices to check the following 3 statements:

1. Algorithm 1 returns a feasible allocation $\mathbf{x}$.
2. The dual certificate (p_t, q) in Proposition 5 is feasible to the dual LP (D).
3. $\sum_t p_t + Bq \leqslant \alpha$.

Given these properties, the guarantee follows via weak LP duality for programs (P) and (D).

For Claim (2), note that by Lemma 2, predicted utilities are a lower bound on $c_i u_i(\mathbf{x})$ for all $i \in [N], t \in [T]$. Now by definition of our dual certificate, we have for all $i \in [N], t \in [T]$

$$p_t + q \geqslant \max_\ell \sum_{i=1}^{N} \frac{v_{i,\ell,t}}{\widetilde{u}_{i,t}(\mathbf{z}_t)} \geqslant \max_\ell \sum_{i=1}^{N} \frac{v_{i,\ell,t}}{c_i u_{i,t}(\mathbf{x})}$$

Claims (1) and (3) are implied by the following key invariant:

Key Invariant. $\sum_t (p_t + q) \sum_\ell z_{\ell,t} \leqslant \frac{\alpha}{4}$.

We first show how this key invariant implies Claims (2) and (3).

(1) Clearly $\mathbf{x}$ satisfies the per-round constraints by definition of the algorithm, so we just need to show that it satisfies the overall budget constraint. Since the set-aside semi-allocation

[6] From the proof of Proposition 6, it can be seen that it still holds even if the constraint set is a scaling of the unit simplex.

is chosen such that $\sum_{\ell,t} y_{\ell,t} = \frac{B}{2}$, we only need to check that $\sum_{\ell,t} z_{\ell,t} \leqslant \frac{B}{2}$. Now, since $p_t \geqslant 0 \, \forall \, t$, the key invariant implies

$$\sum_t \sum_\ell z_{\ell,t} \leqslant \frac{1}{q} \cdot \frac{\alpha}{4} = \frac{B}{2},$$

by the definition of $q = \frac{\alpha}{2B}$, and our choice of α.

(3) By (the contrapositive of) item 2 of Proposition 5, we have that if $p_t > 0$ then $\sum_\ell z_{\ell,t} = 1 - \frac{B}{2T}$. Using this and the key invariant, we have

$$\sum_t p_t \left(1 - \frac{B}{2T}\right) = \sum_t p_t \sum_\ell z_{\ell,t} \leqslant \frac{\alpha}{4}.$$

Moreover, since $B \leqslant T$, this implies $\sum_t p_t \leqslant \frac{\alpha}{2}$. Now, by our choice of $q = \frac{\alpha}{2B}$, we have

$$\sum_t p_t + Bq \leqslant \frac{\alpha}{2} + \frac{B\alpha}{2B} = \alpha.$$

Finally we turn to the **proof of the key invariant.** We have

$$\begin{aligned}
\sum_t (p_t + q) \sum_\ell z_{\ell,t} &= \sum_t \left(\frac{1}{N} \max_{\ell'} \sum_i \frac{v_{i,\ell',t}}{\widetilde{u}_{i,t}(\mathbf{z}_t)}\right) \sum_\ell z_{\ell,t} && \text{(by item 3 of Proposition 5)} \\
&= \frac{1}{N} \sum_t \sum_\ell z_{\ell,t} \cdot \max_{\ell'} \sum_i \frac{v_{i,\ell',t}}{\widetilde{u}_{i,t}(\mathbf{z}_t)} \\
&= \frac{1}{N} \sum_t \sum_\ell z_{\ell,t} \cdot \sum_i \frac{v_{i,\ell,t}}{\widetilde{u}_{i,t}(\mathbf{z}_t)} && \text{(by item 1 of Proposition 5)} \\
&= \frac{1}{N} \sum_t \sum_i \frac{\sum_\ell z_{\ell,t} v_{i,\ell,t}}{\widetilde{u}_{i,t}(\mathbf{z}_t)} = \frac{1}{N} \sum_t \sum_i \frac{\widetilde{u}_{i,t}(\mathbf{z}_t) - \widetilde{u}_{i,t}(0)}{\widetilde{u}_{i,t}(\mathbf{z}_t)}
\end{aligned}$$

Where the last equality follows from our choice of the predicted utility (Definition 5). Now, using the fact that $1 - x \leqslant -\ln(x) \, \forall \, x \in \mathbb{R}$, we have

$$\begin{aligned}
\sum_t (p_t + q) \sum_\ell z_{\ell,t} &\leqslant \frac{1}{N} \sum_t \sum_i \left[\ln\left(\widetilde{u}_{i,t}(\mathbf{z}_t)\right) - \ln\left(\widetilde{u}_{i,t}(0)\right)\right] \\
&= \frac{1}{N} \sum_i \left[\ln\left(\widetilde{u}_{i,t}(\mathbf{z}_T)\right) - \ln\left(\widetilde{u}_{i1}(0)\right)\right] && \text{(by telescoping)} \\
&= \frac{1}{N} \sum_i \left[\ln\left(\frac{B}{2\min\{N,L\}T} \widetilde{V}_i + \sum_t \sum_\ell v_{i,\ell,t} z_{\ell,t}\right) - \ln\left(\frac{B}{2\min\{N,L\}T} \widetilde{V}_i\right)\right] \\
&= \ln\left(\frac{2\min\{N,L\}T}{B}\right) + \frac{1}{N} \sum_i \ln\left(\frac{B}{2\min\{N,L\}T} + \frac{1}{\widetilde{V}_i} \sum_t \sum_\ell v_{i,\ell,t} z_{\ell,t}\right) \\
&\leqslant \ln\left(\frac{2\min\{N,L\}T}{B}\right) + \frac{1}{N} \sum_i \ln\left(\frac{B}{2T} + \frac{1}{\widetilde{V}_i} \sum_t \sum_\ell v_{i,\ell,t} z_{\ell,t}\right)
\end{aligned}$$

Now, observe that

$$\begin{aligned}
\sum_t \sum_\ell v_{i,\ell,t} z_{\ell,t} &\leqslant \sum_t \max_{\ell'}\{v_{i,\ell',t}\} \sum_\ell z_{\ell,t} \\
&\leqslant \sum_t \max_{\ell'}\{v_{i,\ell',t}\} \left(1 - \frac{B}{2T}\right) = V_i \left(1 - \frac{B}{2T}\right),
\end{aligned}$$

where the second inequality is because $\sum_\ell z_{\ell,t} \leqslant 1 - \frac{B}{2T}$, by the stopping condition. Thus

$$\frac{B}{2T} + \frac{1}{\widetilde{V}_i} \sum_t \sum_\ell v_{i,\ell,t} z_{\ell,t} \leqslant \frac{B}{2T} + \left(1 - \frac{B}{2T}\right) \frac{V_i}{\widetilde{V}_i} \leqslant \frac{B}{2T} + \left(1 - \frac{B}{2T}\right) d_i \leqslant d_i.$$

Therefore, we conclude that

$$\sum_t (p_t + q) \sum_\ell z_{\ell,t} \leqslant \ln\left(\frac{2\min\{N, L\}T}{B}\right) + \frac{1}{N} \sum_i \ln(d_i) \leqslant \frac{\alpha}{4},$$

which concludes the proof of the key claim. $\square$