Online Algorithms for the Santa Claus Problem

Anonymous Author(s)
Affiliation
Address
email

Abstract

The Santa Claus problem is a fundamental problem in *fair division*: the goal is to partition a set of *heterogeneous* items among *heterogeneous* agents so as to maximize the minimum value of items received by any agent. In this paper, we study the online version of this problem where the items are not known in advance and have to be assigned to agents as they arrive over time. If the arrival order of items is arbitrary, then no good assignment exists in the worst case. However, we show that even for arbitrary items, if their arrival order is random, then for any $\varepsilon > 0$, we can obtain a competitive ratio of $1 - \varepsilon$ when the optimal assignment gives value at least $\Omega(\log n/\varepsilon^2)$ to every agent. We also show that this result is almost tight: namely, if the optimal solution has value at most $C \ln n/\varepsilon$ for some constant C, then there is no $(1-\varepsilon)$-competitive algorithm even with random arrival order.

1 Introduction

Fair allocation of resources is one of the central themes of algorithmic fairness and game theory. In fact, the theory of fair division has its roots in mathematics going back to as early as 1948 [56]. In the general setting, this problem comprises a set of items that must be divided among a set of agents in an egalitarian manner, where each agent has a (possibly non-uniform) valuation for each item. A natural objective to capture the goal of fair division is to maximize the minimum total value of items received by any agent. This gives rise to the famous "Santa Claus problem" that we describe below.

In the Santa Claus problem, originally described by Bansal and Sviridenko in 2006 [12] (although it was studied under different names or assumptions prior to this), the Santa Claus is said to have a set of m presents to be distributed equitably among n children. Each child $i \in [n]$ has some arbitrary non-negative value v_{ij} for present $j \in [m]$. Furthermore, Santa's goal is to distribute the presents in a way that makes the least satisfied child maximally satisfied. More formally, this means that the assignment seeks to maximize the minimum total value of the presents received by any child, where the total value of presents received by a child is the sum of her values for the presents that she received. The Santa Claus problem can be formalized as the following integer program:

$$\max \left\{ \min_{i \in [n]} \sum_{j=1}^{m} v_{ij} x_{ij} \mid \sum_{i=1}^{n} x_{ij} \leq 1 \; \forall j \in [m], \; x \in \{0,1\}^{mn} \right\}.$$

There is significant literature going back more than 50 years that studies variants of this problem in the offline setting (see related work). In many practical situations, however, the set of items to be allocated is not known in advance. For example, in online advertising, ad-space providers will receive monetary bids from competing agents for the display of their advertisements in real-time for an available space on a webpage [17, 15, 38, 59]. The provider must make then irrevocable decisions as to which advertiser's bid to accept for the display of their ad to ensure that revenue is maximized only using knowledge of prior allocations and the current bids for the available space

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

[10, 22]. Beyond advertising, online allocation procedures have been critical in the study of donation distribution, wireless charging networks, and especially organ donor matching (see [4] for a survey of these applications).

Motivated by these applications, we consider the *online* Santa Claus problem in this paper. In this setting, the items arrive in an online sequence and must be allocated to one of the agents immediately upon arrival. As in the offline problem, our goal is to maximize the minimum total value among all agents. To illustrate the problem, consider the simple example in Figure 1 on the right. At the time of the first arriving item, all agents can be matched to this item and therefore with probability $\frac{1}{3}$ the rightmost agent will receive it. However, as we continue forward with the input stream, we see in retrospect that the only nonzero max-min solution can be achieved if that first item was indeed allocated to the rightmost agent (as it is the only possible item they can receive).

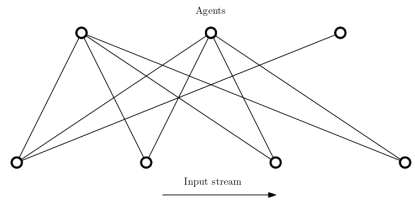

Figure 1: Example online problem instance. Fixed agents are presented in the top row and the arriving items are arranged from left to right in correspondence with their arrival order. Edges indicate a nonzero item valuation to a given agent.

Following standard terminology for online algorithms (see, e.g., [16]), we define the *competitive ratio* of an online algorithm as the minimum ratio between the value of the (maximization) objective in the algorithm's solution to that in the optimal (offline) solution in hindsight. We furthermore discuss the *additive regret* as the additive loss factor of our algorithm. More formally, we say our algorithm, ALG, has competitive ratio c and additive regret b if these are the largest and smallest values respectively that satisfy $\text{ALG} \geq c \cdot \text{OPT} - b$.

Prior work on the max-min objective in the online setting required various relaxations of the problem, such as allowing for some reordering in the allocation process [28], restricting the number of agents [39, 57, 58], or allowing migration of items after assignment [19]. This is because of two reasons. First, even in the offline setting, there remains a significant gap between the best upper and lower bounds on the approximation ratio of the Santa Claus problem, and bridging this gap is a major open problem. Second, as we will soon see, there is a simple construction for the online problem that shows the competitive ratio cannot be better than n. To bypass these bottlenecks, our first assumption in this paper is that the items arrive in *random order*. This is a standard assumption that has been used to simplify many related online problems [9, 24, 31, 32, 34, 41, 45]. But, even with this assumption, we show that obtaining a competitive solution is impossible in general *for small problem instances*. This motivates our second assumption: that the objective value of the optimal solution is sufficiently large. With these two assumptions, we give an algorithm that obtains a competitive ratio of $(1 - \varepsilon)$ for any $\varepsilon > 0$. We note that using standard techniques, the assumption about the optimal objective being sufficiently large can be replaced by a corresponding additive regret in the competitive ratio.

We now formally define the two online input stream models that we talk about in this paper: *adversarial* and *random order* input.

Definition 1.1 (Adversarial Input)**.** *An adversary selects the value vector $v \in [0, 1]^n$ of each arriving item for all the agents, as well as the order in which these vectors arrive.*

Definition 1.2 (Random-Order Input)**.** *An adversary selects the value vector $v \in [0, 1]^n$ of each arriving item for all the agents, but these vectors are randomly permuted to determine their arrival order.*

Note that in the literature, the independent and identically distributed (i.i.d.) input model is also often studied for related problems [34, 42, 49, 41, 2, 44, 55, 52]. In this model, the adversary picks a distribution over inputs that is unknown to the algorithm and arriving items are sampled i.i.d. from this distribution. The random order model is *stronger* than the i.i.d. model in the sense that any algorithmic result for the random order model automatically extends to the i.i.d. setting as well. This includes the algorithmic results that we obtain in this paper for the random order arrival model.

Table 1: Online Results for the Santa Claus Problem

Our Results

Input Model	*Algorithm*	*Competitive Ratio*
Adversarial	RANDOM	$\left(\frac{1-\varepsilon}{n}\right)$ OPT $- O\left(\frac{n \log n}{\varepsilon^2}\right)$
Random Order	GREEDYWR	$(1-\varepsilon)$OPT $- O\left(\frac{\log n}{\varepsilon}\right)$

1.1 Our Contributions

First, we give a simple construction to show that if the arrival order of the items is adversarial, then the best competitive ratio that can be achieved is only $1/n$. (In fact, we also match this competitive ratio using a simple algorithm in the supplementary material.)

Theorem 1.3 (Adversarial Input)**.** *In the adversarial setting, no algorithm can obtain a competitive ratio better than* $1/n$.

This motivates us to consider the *random order model*, where an adversary again selects the set of items but they are then presented in random permutation order. For this setting, we give an algorithm that obtains a *fractional* assignment that is nearly optimal:

Theorem 1.4 (Random Order: Algorithm)**.** *For any* $\varepsilon > 0$*, there is an online algorithm for the Santa Claus problem that has a competitive ratio of* $(1-\varepsilon)$ *in the random order input model for all input instances such that the value of the optimal offline solution satisfies* $\text{OPT} \geq \Omega\left(\log n/\varepsilon^2\right)$.

We will further show that, as a result of the size assumption of the input, we can further utilize a randomized rounding procedure to give an *integral* allocation that retains the near optimality.

Finally, we show that the lower bound on the value of OPT in the above theorem is *necessary*:

Theorem 1.5 (Random Order: Impossibility Result)**.** *For any* $\varepsilon \in (0,1)$*, there is no online algorithm for the Santa Claus problem in the random order input model that has a competitive ratio of* $(1-\varepsilon)$ *when* $\text{OPT} < C \cdot \frac{\ln n}{\varepsilon}$ *for some (absolute) constant* $C > 0$.

The reader will note that the lower bound on OPT is precise as a function of n, but there is a slight mismatch between the upper and lower bounds as a function of ε – bridging this gap is an interesting open question.

We summarize our results for the *online* Santa Claus problem in Table 1.1.

Outline of the Paper. The remainder of the paper is organized as follows: in Section 1.2 we discuss the literature on related problems and how the present work compares, in Section 3 we present a $(1-\varepsilon)$-competitive algorithm for the random order model and in Section 4 we compliment these results with novel impossibility results on the Santa Claus problem. We lastly provide concluding remarks in Section 5.

1.2 Related Work

The general case of the Santa Claus problem was initially explored (under a different name) in the field of algorithmic game theory for the fair allocation of goods [47]. By studying the assignment LP of [46] for the dual "makespan" problem, Bezakova and Dani [13] derived an additive approximation of $\max_{ij} v_{ij}$, i.e., the objective in the algorithm's solution is at least $\text{OPT} - \max_{ij} v_{ij}$ where OPT is the objective value of the optimal solution. They also extended the hardness result on the dual makespan minimization problem [46] to demonstrate that the Santa Claus problem is NP-hard and cannot be approximated to a factor better than 2. Later, Bansal and Sviridenko [12] demonstrated that the integrality gap of the configuration LP for this problem is $\Omega(\sqrt{n})$, while Asadpour and Saberi [7] complimented this result with a $O(\sqrt{m}\log^3 m)$ upper bound for the same LP relaxation. To date, the best algorithmic result for the Santa Claus is an $\tilde{O}(n^{\varepsilon})$-approximate algorithm, where $\varepsilon = \Omega(\log\log n/\log n)$, in quasi-polynomial time obtained by Chakrabarty, Chuzhoy, and Khanna [18].

For the special case of *restricted assignment*, i.e. $v_{ij} \in \{0, v_j\}$, Bansal and Sviridenko [12] provided an $\Omega\left(\log\log\log m/\log\log m\right)$-approximate algorithm that relies on rounding a configuration LP.

Later, Feige gave a non-constructive proof that this LP relaxation was within a constant factor of OPT [29]. Asadpour, Feige, and Saberi [6] made this constructive, obtaining a $1/4$-approximation via a rounding algorithm based on local search, but the algorithm is not known to converge in polynomial time. Further work has since improved this constant factor [40, 20, 23, 5], improved upon the running time [21], and extended the setting beyond additive valuations [11].

Online Assignment. The study of online assignment is expansive, but much of the classical work is for adversarial arrival order. Even in the random order setting, a broad range of problems have been considered in recent years including the secretary problem [9, 45], AdWords [24, 31, 34], online matching [32, 41], online packing [37, 30, 44], online scheduling [51, 48], etc. One example of max-min online assignment in the random order setting is the work of Gollapudi and Panigrahi [36] who considered revenue maximization with fairness objectives. Another related work is that of Molinaro [51] for the dual min-max objective, who builds on prior work leveraging the experts framework from online learning [37] to give algorithms that simultaneously perform well in the adversarial and random-order settings. A third line of work relevant to our paper is that of online packing problems in the random arrival order (e.g., [30, 2, 44, 55, 52]. In particular, the results of Agrawal *et al.* [2] have a similar flavor to ours: they obtain $(1-\varepsilon)$-competitiveness assuming a large enough optimal value for the online packing problem in the random order setting.

2 Preliminaries

In this section, we introduce the notation that we will use in the rest of the paper. Let $\mathbf{v}^1, ..., \mathbf{v}^m \in [0,1]^n$ denote the input sequence of items arriving in random order where v_i^t is the value of the t-th item to the i-th agent. We additionally denote by $\mathbf{x}^1, ..., \mathbf{x}^m \in [0,1]^n$ the fractional allocation of each item by the algorithm. We further let the corresponding allocations of a fixed (offline) optimal solution be denoted as $\overline{\mathbf{x}}^1, ..., \overline{\mathbf{x}}^m \in [0,1]^n$ and let OPT denote the max-min objective value of the optimal solution. For simplicity, we slightly abuse notation by letting $\mathbf{V}^t = (v_1^t x_1^t, \ldots, v_n^t x_n^t)$ and $\overline{\mathbf{V}}^t = (v_1^t \overline{x}_1^t, \ldots, v_n^t \overline{x}_n^t)$. Our algorithm thus seeks to maximize the coordinate-wise minimum of $\sum_{t=1}^m \mathbf{V}^t$.

3 Online Algorithm for the Santa Claus Problem in the Random Order Model

In this section, we present the approximately greedy algorithm, Algorithm 1, and analyze its competitive ratio in the random order model. Building on the work of Molinaro for online scheduling [51], we use a greedy algorithm for a smoothed version of our objective function and a restart procedure during the online allocation process to reduce the impact of correlations that arise in this input model.

A natural strategy for our problem is to allocate the arriving item to the least satisfied agent. However, one can show this strategy has too high an additive regret [37, 51] as the change in our solution value can vary quickly from one iteration to the next. Instead, we use a *smoothed* version of the greedy algorithm. The algorithm is designed as follows: we first define ϕ_ε to be a re-scaled variant of the LogSumExp function that serves as a smoothed minimum. For the first half of the input stream, we select an allocation for each arriving item that maximizes the increase in our smoothed objective function. This stage can be thought of as approximately greedy with respect to the *gradient* of ϕ_ε. After $\frac{m}{2}$ items have been allocated, we "restart" the allocation by maximizing the increase in our objective with respect to the $t > \frac{m}{2}$ allocations only. This restart procedure is essential for reducing the correlations that arise in sampling without replacement in the random order model since at each iteration of the allocation procedure, our decision depends on at most $\frac{m}{2} - 1$ items. The pseudocode of this procedure is presented in Algorithm 1.

3.1 Algorithm Analysis

In the analysis of the competitive ratio of Algorithm 1, we will leverage several key facts about the smoothed minimum function ϕ_ε. Allocating with respect to this function produces an almost greedy process on the function's gradient, allowing us to follow the analysis of [1, 24, 51] to get the desired guarantees for the random-order model. We now state our main result in Theorem 3.1.

Theorem 3.1. *For any $\varepsilon > 0$, Algorithm 1 guarantees in the random-order input model that the expected value of the allocation assigned to any agent is at least*

$$(1-\varepsilon) \cdot \text{OPT} - O(\log n/\varepsilon).$$

Note that this theorem immediately implies the following corollary since the additive regret term can be absorbed in the multiplicative error for sufficiently large OPT:

Corollary 3.2. *For any $\varepsilon > 0$, Algorithm 1 has a competitive ratio of $(1-\varepsilon)$ for $\text{OPT} \geq \Omega(\log n/\varepsilon^2)$.*

In order to prove this theorem, we first show some properties of the smoothed minimum function ϕ_ε utilized by Algorithm 1. We will then prove some technical lemmas that will help establish the theorem.

Lemma 3.3 effectively defines the additive error with respect to our true objective, the agent-wise minimum, and stability of the smooth function following each allocation decision. As was shown in prior work, allocating with respect to the order statistics or even the L_p^p norm produces either too high of a regret factor, or instability in the derivative value under small perturbations to the input value. As such, the smoothing and the following properties are critical to maintaining our regret and competitive ratio bounds.

Lemma 3.3. *For all $u \in \mathbb{R}^n$, $v \in [0,1]^n$, and $\varepsilon > 0$, the function $\phi_\varepsilon(x) = \frac{-1}{\varepsilon} \ln\left(\sum_{i=1}^n e^{-\varepsilon x_i}\right)$ satisfies the following:*

(a) $\min_i\{u_i\} - \frac{\ln n}{\varepsilon} \leq \phi_\varepsilon(u) < \min_i\{u_i\}$

(b) $\nabla\phi_\varepsilon(u+v) \in e^{\pm\varepsilon} \cdot \nabla\phi_\varepsilon(u)$

The proof of these properties are deferred to Appendix A due to space constraints.

Now utilizing these two properties we can prove the following important bound on the inner product of the smoothed minimum's gradient. This will be used throughout our analysis to bound the incremental change in the objective MAXMIN value after each allocation decision, and the summation of these changes can be seen as the accumulated "reward" at any given stage of the input stream. The proof is by direct integration of stability property (b) and is thus deffered to Appendix A.

Lemma 3.4. *For $u \in \mathbb{R}^n$ and $v, v' \in [0,1]^n$, if $\phi_\varepsilon(u+v) \geq \phi_\varepsilon(u+v')$ then $\langle \nabla\phi_\varepsilon(u), v\rangle \geq e^{-2\varepsilon}\langle \nabla\phi_\varepsilon(u), v'\rangle$.*

We now follow in the intuition of Agrawal and Devanur [1] to prove Theorem 3.1 by bounding the incremental increase in our "reward" both before and after the restart at $t = \frac{m}{2}$. By implementing this restart in the allocation procedure, we segment the stream into two portions that have identical probabilistic guarantees and reduce the correlation between input elements to allow for an optimal competitive ratio and low additive regret.

Proof of Theorem 3.1. We note again that in the approximately greedy procedure of Algorithm 1, we are essentially seleting items greedily according to the *gradient* of ϕ_ε to maximize the incremental changes. Due to the restart at $m/2$, we define

$$\nabla^t = \nabla\phi_\varepsilon\left(\sum_{\tau=1}^{t-1} \mathbf{V}^\tau\right) \text{ for } t \leq \frac{m}{2} \text{ and } \nabla^t = \nabla\phi_\varepsilon\left(\sum_{\tau=m/2+1}^{t-1} \mathbf{V}^\tau\right) \text{ for } t > \frac{m}{2}$$

ALGORITHM 1: SMOOTH GREEDY WITH RESTART

Input: $0 < \varepsilon < 1$, input stream $\mathcal{M}$ of m items
Define $\phi_\varepsilon(u) = \frac{-1}{\varepsilon} \ln\left(\sum_i e^{-\varepsilon u_i}\right)$;
for $t = 1$ *to* $m/2$ **do**
  Select $\mathbf{x}^t \in \Delta^n$ to maximize $\phi_\varepsilon\left(\sum_{\tau=1}^t \mathbf{V}^\tau\right)$;
end
for $t = m/2+1$ *to* m **do**
  Select $\mathbf{x}^t \in \Delta^n$ to maximize $\phi_\varepsilon\left(\sum_{\tau=\frac{m}{2}+1}^t \mathbf{V}^\tau\right)$;
end

We now proceed by deriving a bound on $\min_i\{\sum_{\tau=1}^{m} \mathbf{V}_i^\tau\}$ in terms of our smoothed-approximation function ϕ_ε, thus bounding the accumulated error by the algorithm when estimating our true objective. By the concavity of ϕ_ε and Lemma 3.3, we have that

$$\phi_\varepsilon\left(\sum_{\tau=1}^{t}\mathbf{V}^\tau\right) - \phi_\varepsilon\left(\sum_{\tau=1}^{t-1}\mathbf{V}^\tau\right) \geq \left\langle \nabla\phi_\varepsilon\left(\sum_{\tau=1}^{t}\mathbf{V}^\tau\right), \mathbf{V}^t\right\rangle \qquad \text{(Concavity)}$$

$$\geq e^{-\varepsilon}\left\langle \nabla\phi_\varepsilon\left(\sum_{\tau=1}^{t-1}\mathbf{V}^\tau\right), \mathbf{V}^t\right\rangle = e^{-\varepsilon}\langle \nabla^t, \mathbf{V}^t\rangle. \qquad \text{(Lemma 3.3)}$$

Without loss of generality we proceed by considering the first half of the input sequence ($t \leq \frac{m}{2}$). By summing this inequality over the input stream prior to the restart (from $t = 1$ to $\frac{m}{2}$), we have

$$\phi_\varepsilon\left(\sum_{\tau=1}^{m/2}\mathbf{V}^\tau\right) - \phi_\varepsilon(0) = \phi_\varepsilon\left(\sum_{\tau=1}^{m/2}\mathbf{V}^\tau\right) + \frac{\ln n}{\varepsilon} \geq e^{-\varepsilon}\sum_{t=1}^{m/2}\langle \nabla^t, \mathbf{V}^t\rangle \tag{1}$$

Now, taking into account the allocation before and after the restart at $\frac{m}{2}$ and invoking Eq. 1 for each half of the input stream with the concavity of ϕ_ε, we obtain

$$\sum_{t=1}^{m}\langle \nabla^t, \mathbf{V}^t\rangle \leq e^{\varepsilon}\left(\phi_\varepsilon\left(\sum_{\tau=1}^{m/2}\mathbf{V}^\tau\right) + \phi_\varepsilon\left(\sum_{\tau=m/2+1}^{m}\mathbf{V}^\tau\right) + \frac{2\ln n}{\varepsilon}\right) \qquad \text{(Ineq. 1)}$$

$$\leq e^{\varepsilon}\left(\phi_\varepsilon\left(\sum_{\tau=1}^{m}\mathbf{V}^\tau\right) + \frac{2\ln n}{\varepsilon}\right) \qquad \text{(Concavity)}$$

$$\leq e^{\varepsilon}\left(\min_i\{\sum_{\tau=1}^{m}\mathbf{V}_i^\tau\} + \frac{2\ln n}{\varepsilon}\right) \qquad \text{(Lemma 3.3)}$$

where the last inequality is a direct result of Lemma 3.3. Lastly, by rearranging terms we can bound the element-wise minimum as

$$\min_i\{\sum_{\tau=1}^{m}\mathbf{V}_i^\tau\} \geq e^{-\varepsilon}\left(\sum_{t=1}^{m}\langle \nabla^t, \mathbf{V}^t\rangle\right) - \frac{2\ln n}{\varepsilon}. \tag{2}$$

Thus, the output of our algorithm will approximate the *actual* objective function within a multiplicative factor of $e^{-\varepsilon} \approx 1 - \varepsilon$ and an additive regret on the order of $\ln n/\varepsilon$.

We now proceed to bound the gap between our algorithmic solution to that of the optimal *offline* solution which, combined with the above error, will give the final result. As such, the final step of our analysis will be to take the expectation of this inequality to get the final bounds in Theorem 3.1. Due to the restart and random order input model, the expected increase in maximal value over the first and second half is equivalent,

$$\mathbb{E}\left[\sum_{t=1}^{m/2}\langle \nabla^t, \mathbf{V}^t\rangle\right] = \mathbb{E}\left[\sum_{t=m/2+1}^{m}\langle \nabla^t, \mathbf{V}^t\rangle\right]$$

so without loss of generality we need only bound the first half's value and apply this bound to both portions. The benefit of this restart will yield a tolerable error as compared to the optimal offline solution in each half of the allocation procedure, rather than a continuously accumulating divergence between the two solutions.

Now, by the nature of greedy selection to maximize our objective function, we must have that in each iteration our algorithm's selection has value at least as good as that of the optimal offline solution $\phi_\varepsilon(\mathbf{V}^t) \geq \phi_\varepsilon(\overline{\mathbf{V}}^t)$ for each t. Combining this with Lemma 3.4, we must have

$$\langle \nabla^t, \mathbf{V}^t\rangle \geq e^{-2\varepsilon}\langle \nabla^t, \overline{\mathbf{V}}^t\rangle$$

Furthermore, since $\nabla\phi_\varepsilon \in \ell_1^+$ [33] and ∇^t is independent of $\mathbf{V}^t$, we prove the following purely probabilistic result that in Appendix A will be used to bound the above summations to derive the final result.

Lemma 3.5. *Consider a set of vectors $\{y^1, ..., y^m\} \in [0,1]^n$ and let $\{\mathbf{Y}^i\}_{i=1}^k$ be sampled without replacement. Let $\mathbf{Z} \in \ell_1^+$ be a random vector that depends only on $\{\mathbf{Y}^i\}_{i=1}^{k-1}$. Then for all $\varepsilon > 0$,*

$$\mathbb{E}\langle \mathbf{Y}^k, \mathbf{Z}\rangle \geq e^{-\varepsilon} \min_i \{\mathbb{E}\mathbf{Y}_i^k\} - \frac{\ln n}{\varepsilon\,(m-k+1)}.$$

Using this lower bounding result, we can now bound the sum of rewards as

$$\mathbb{E}\left[\langle \nabla^t, \overline{\mathbf{V}}^t\rangle\right] \geq e^{-\varepsilon} \min_i \{\mathbb{E}\left[\overline{\mathbf{V}}_i^t\right]\} - \frac{\ln n}{\varepsilon(m-t+1)}. \tag{3}$$

Adding this result over all $t \leq \frac{m}{2}$ in combination with $\min_i\{\mathbb{E}[\overline{\mathbf{V}}^t]_i\} = \frac{\text{OPT}}{m}$ we conclude that

$$\begin{aligned}
e^{2\varepsilon} \cdot \mathbb{E} \sum_{t \leq \frac{m}{2}} \langle \nabla^t, \mathbf{V}^t\rangle &\geq \mathbb{E} \sum_{t \leq \frac{m}{2}} \langle \nabla^t, \overline{\mathbf{V}}^t\rangle && \text{(Lemma 3.3)}\\
&\geq e^{-\varepsilon} \sum_{t \leq \frac{m}{2}} \min_i \{\mathbb{E}\left[\overline{\mathbf{V}}_i^t\right]\} - \sum_{t \leq \frac{m}{2}} \frac{\ln n}{\varepsilon(m-t+1)} && \text{(Ineq. 3)}\\
&= e^{-\varepsilon}\left(\frac{\text{OPT}}{2}\right) - \frac{\ln n}{\varepsilon}
\end{aligned}$$

Due to the restart at $t = \frac{m}{2}$, we can extend the sum to $t = m$ by simply multiplying the above RHS by two. Finally, invoking inequality (2), we see that

$$\mathbb{E}\left[\min_i \sum_{\tau=1}^m \mathbf{V}^\tau\right] \geq e^{-4\varepsilon}\text{OPT} - O\left(\frac{\log n}{\varepsilon}\right) \geq (1 - O(\varepsilon))\,\text{OPT} - O\left(\frac{\log n}{\varepsilon}\right).$$

by the Taylor approximation $e^{-x} \geq 1 - x$ and the assumption on the size of OPT. □

3.2 Online Rounding Algorithm

The previous algorithm produces an online fractional solution for the Santa Claus problem. We now show that using simple randomized rounding, we can convert this into an integer solution.

Theorem 3.6. *Fix any $\varepsilon > 0$. Given a $(1-\varepsilon)$competitive online fractional algorithm for the Santa Claus problem, there is an online (integral) algorithm whose competitive ratio is $(1 - O(\varepsilon))$, provided* $\text{OPT} \geq \Omega\left(\frac{\log n}{\varepsilon^2}\right)$.

Proof. The algorithm is simply randomized rounding. If an item j is allocated with fraction x_{ij} to agent i such that $\sum_{i=1}^n x_{ij} \leq 1$ by the fractional solution, then we assign item i to agent j with probability x_{ij}. Note that since $v_{ij} \in [0,1]$, the fractional value derived by an agent i from an item j, given by $v_{ij}x_{ij}$ is also in $[0,1]$. Thus, by Chernoff bounds, the probability that the total value of agent i in the rounded assignment is less than $(1-\varepsilon)\sum_{j=1}^m v_{ij}x_{ij}$ is at most

$$\exp\left(-\frac{\varepsilon^2}{3} \cdot \sum_{j=1}^m v_{ij}x_{ij}\right) \leq \exp\left(-\frac{\varepsilon^2}{3} \cdot (1-\varepsilon) \cdot \text{OPT}\right) \leq \frac{1}{n^2}, \text{ for OPT} \geq \Omega\left(\frac{\log n}{\varepsilon^2}\right).$$

Now, we have

$$(1-\varepsilon)\sum_{j=1}^m v_{ij}x_{ij} \geq (1-\varepsilon)^2 \cdot \text{OPT} > (1-2\varepsilon) \cdot \text{OPT}.$$

It follows that the probability that the total value of agent i in the rounded assignment is less than $(1-2\varepsilon) \cdot \text{OPT}$ is at most $\frac{1}{n^2}$. Using the union bound over the n agents, we get that the probability that the total value of *any* agent in the rounded assignment is less than $(1-2\varepsilon) \cdot \text{OPT}$ is at most $\frac{1}{n}$. Thus, the expected value of the objective is at least $(1-\frac{1}{n})(1-2\varepsilon) \cdot \text{OPT} > (1-3\varepsilon) \cdot \text{OPT}$ for large enough n. □

4 Random Order Lower Bound

We here present an impossibility result on the Santa Claus problem under the random order input models. Following from Theorem 1.3, it is clear that the online result for the Santa Claus problem is highly nontrivial and dependent upon the adversarial selections. For worst-case instances, we cannot hope to achieve a guarantee that is any greater than the most pessimistic possibility.

While in the random-order and IID input models the impact of such an adversary is diminished, we demonstrate that nearly optimal online results can only be achieved if the offline solution is large enough. This condition has been studied extensively for similar problems like bin-packing and covering which demonstrates a $\Omega(\log n/\varepsilon^2)$ lower bound [2], while other problems such as load balancing and makespan do not have a full consensus on the tightest bound. We here present the first result for the Santa Claus problem under the random-order assumption in Theorem 4.1.

Theorem 4.1. *For any $\gamma \in (0,1)$, if a randomized algorithm* ALG *for the online Santa Claus problem with random order input satisfies*

$$\text{ALG} \geq (1-\gamma) \cdot \text{OPT},$$

then $\text{OPT} \geq \frac{C \ln n}{\gamma}$ *for some (absolute) constant $C > 0$.*

We use the following construction from the proof of Theorem 1.3. There are n agents, of which $n-1$ are *private* agents and the remaining one is a *public* agent. Every private agent has k distinct private items for which their valuation is 1 each, and the valuation for every other agent 0. In addition, there are k public items, each of which has a valuation of 1 for every agent. The optimal solution is to assign the private items to the corresponding private agents, and the public items to the public agent. Thus, $\text{OPT} = k$.

Now, when the items are presented in uniform random order, consider the first ε fraction of presented items – call this the ε-prefix. Our main claim is that with constant probability, the following statements both hold:

(a) there are about ε fraction of public items in this ε-prefix

(b) there is at least one type of private item that is missing from this ε-prefix.

For property (a), we need the following concentration inequality from Devanur and Hayes [24]:

Lemma 4.2 (Lemma 3 in [24])**.** *Let $Y = (Y_1, \ldots, Y_m)$ be a vector of real numbers, and let $\varepsilon \in (0,1)$. Let S be a random subset of $[m]$ of size εm, and set $Y_S := \sum_{j \in S} Y_j$. Then, for every $\delta \in (0,1)$,*

$$\mathbf{Pr}\left[|Y_S - \mathbb{E}[Y_S]| \geq \frac{2}{3}\|Y\|_\infty \ln\left(\frac{2}{\delta}\right) + \|Y\|_2 \sqrt{2\varepsilon \ln\left(\frac{2}{\delta}\right)}\right] \leq \delta.$$

Property (a) concerning the fraction of public items in the ε-prefix follows almost immediately from the above lemma:

Lemma 4.3. *The probability that there are fewer than $\frac{5\varepsilon}{12}$ fraction of public items in the ε-prefix is at most $2\exp(-\varepsilon k/8)$.*

Proof. We invoke 4.2 with the following setting of variables. Let Y be a binary vector where $Y_i = 1$ for $i \in [k]$ and $Y_i = 0$ otherwise. (Y_i is the indicator for whether an item is a public item.) S represents the set of indices in the ε-prefix. Then, Y_S counts the number of public items in the ε-prefix in a random ordering of the items. Clearly, $\mathbb{E}[Y_S] = \varepsilon k$, $\|Y\|_\infty = 1$, and $\|Y\|_2 = \sqrt{k}$. Finally, set $\delta = 2e^{-\frac{\varepsilon k}{8}}$, or more so $\ln\left(\frac{2}{\delta}\right) = \frac{\varepsilon k}{8}$. Now, by 4.2, we have

$$\mathbf{Pr}\left[|Y_S - \varepsilon k| \geq \frac{2}{3} \cdot \frac{\varepsilon k}{8} + \sqrt{k} \cdot \sqrt{2\varepsilon \cdot \frac{\varepsilon k}{8}}\right] = \mathbf{Pr}\left[|Y_S - \varepsilon k| \geq \frac{7}{12} \cdot \varepsilon k\right] \leq 2e^{-\frac{\varepsilon k}{8}}.$$

and the lemma follows. □

Now, we show property (b) to demonstrate the non-existence of at least one type of private item in the ε-prefix.

Lemma 4.4. *The probability that all the $n-1$ types of private items appear in the ε-prefix is at most* $\exp\left(-(n-1)\cdot 4^{\frac{-\varepsilon k}{1-\varepsilon}}\right)$.

Proof. Fix a type of private item, say those of type-j, i.e. only agent j (for some $j \in [n-1]$) has unit value for this type of item while all other agents have 0 value. First, we bound the probability that no item of type-j appears in the ε-prefix. To do this, note that this probability can be written, using the chain rule for conditional probabilities, as the product (over i from 1 to εnk) of the probabilities of the ith item in the ε-prefix not being a type-j item under the condition that the first $i-1$ items were not type-j items either. Clearly, this probability, for any $i \le \varepsilon nk$, is at least $1-\frac{1}{(1-\varepsilon)n}$ since there are k items of type-j among at most $(1-\varepsilon)nk$ items overall after the conditioning. Thus, the probability that no item of type-j appears in the ε-prefix is at least

$$\left(1-\frac{1}{(1-\varepsilon)n}\right)^{\varepsilon nk} \ge 4^{\frac{-\varepsilon}{1-\varepsilon}\cdot k} \text{ by choosing } n \ge \frac{2}{1-\varepsilon}.$$

Denote $p = 4^{\frac{-\varepsilon}{1-\varepsilon}\cdot k}$. Consider the events that at least one item of type-j appears in the ε-prefix. These events are negatively correlated and therefore, the probability that at least one item of type-j appears in the ε-prefix for every $j \in [n-1]$ is at most

$$(1-p)^{n-1} \le \left(1-4^{\frac{-\varepsilon}{1-\varepsilon}\cdot k}\right)^{n-1} \le e^{-4^{\frac{-\varepsilon}{1-\varepsilon}k}(n-1)}.$$

□

Now, by setting $k = \frac{1-\varepsilon}{2\varepsilon}\cdot \lg(n-1)$ we obtain

$$(n-1)\cdot 4^{\frac{-\varepsilon}{1-\varepsilon}k} = (n-1)\cdot 2^{-\frac{2\varepsilon}{1-\varepsilon}\cdot\frac{1-\varepsilon}{2\varepsilon}\cdot\lg(n-1)} = (n-1)\cdot 2^{-\lg(n-1)} = 1,$$

and furthermore,

$$\frac{\varepsilon k}{8} = \frac{1-\varepsilon}{16}\cdot\lg(n-1).$$

Plugging these expressions into 4.3 and 4.4, we get that the probability of every private item type appearing in the ε-prefix is at most $\frac{1}{e}$ and the probability of fewer than $\frac{5\varepsilon}{12}$ fraction of public items appearing in the ε-prefix is at most $2e^{-\frac{1-\varepsilon}{16}\cdot\lg(n-1)}$. The latter probability can be further simplified to

$$2e^{-\frac{1-\varepsilon}{16}\cdot\lg(n-1)} = 2e^{-\frac{1-\varepsilon}{16}\cdot\frac{\ln(n-1)}{\ln 2}} = \left(\frac{2}{n-1}\right)^{\frac{1-\varepsilon}{16\ln 2}} < \frac{1}{e} \text{ for large enough } n.$$

Using the union bound over these two events, we conclude that with probability at least $1-\frac{2}{e}$, there is at least one missing private item type in the ε-prefix, and also at least $\frac{5\varepsilon}{12}$ fraction of private items appear in the ε-prefix. In this case, in the ε-prefix, the algorithm cannot distinguish between the private agent whose item type is missing and the public agent. Thus, in expectation (over the randomness of the algorithm), at least half the public items in the ε-prefix are assigned to the private agent whose item type is missing. As a consequence, at least $\frac{5\varepsilon}{24}$ fraction of the public items are not allocated to the public agent in the entire algorithm, which means that the total valuation of the public agent is at most $\left(1-\frac{5\varepsilon}{24}\right)k$. Thus, in order to guarantee

$$\text{ALG} \ge (1-\gamma)\cdot\text{OPT},$$

we need $\gamma \ge \frac{5\varepsilon}{24}$, i.e., $\frac{1}{\varepsilon} \ge \frac{5}{24\gamma}$. This implies by the set value of k that

$$\text{OPT} = k = \frac{1-\varepsilon}{2\varepsilon}\cdot\lg(n-1) = \frac{\frac{1}{\varepsilon}-1}{2}\cdot\lg(n-1) \ge \frac{\frac{5}{24\gamma}-1}{2}\cdot\lg(n-1) \ge C\cdot\frac{\ln n}{\gamma} \text{ for some constant } C.$$

This completes the proof of 4.1. □

5 Conclusion

In the present work, we have presented impossibility results for the online Santa Claus problem subject to an adversarial or random order input model, as well as a near optimal algorithm for the random order setting. These results effectively address the necessary assumptions on the problem to obtain optimal online solutions, and furthermore obtain these results via simplistic algorithms. Since the MaxMin objective function is the dual of MinMax used in the makespan and load balancing literature, we hope that the impossibility results presented here can be carried over to address the remaining open questions regarding their tightest additive terms in competitive ratio analysis.

References

[1] AGRAWAL, S., AND DEVANUR, N. R. Fast algorithms for online stochastic convex programming. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms* (2014), SIAM, pp. 1405–1424.

[2] AGRAWAL, S., WANG, Z., AND YE, Y. A dynamic near-optimal algorithm for online linear programming, 2014.

[3] ALAEI, S., HAJIAGHAYI, M., AND LIAGHAT, V. The online stochastic generalized assignment problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2013, pp. 11–25.

[4] ALEKSANDROV, M., AND WALSH, T. Online fair division: A survey. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2020), vol. 34, pp. 13557–13562.

[5] ANNAMALAI, C., KALAITZIS, C., AND SVENSSON, O. Combinatorial algorithm for restricted max-min fair allocation, 2016.

[6] ASADPOUR, A., FEIGE, U., AND SABERI, A. Santa claus meets hypergraph matchings. *ACM Trans. Algorithms 8*, 3 (jul 2012).

[7] ASADPOUR, A., AND SABERI, A. An approximation algorithm for max-min fair allocation of indivisible goods. *SIAM Journal on Computing 39*, 7 (2010), 2970–2989.

[8] AZAR, Y., AND EPSTEIN, L. On-line machine covering. *Journal of Scheduling 1*, 2 (1998), 67–77.

[9] BABAIOFF, M., IMMORLICA, N., KEMPE, D., AND KLEINBERG, R. A knapsack secretary problem with applications. In *Approximation, randomization, and combinatorial optimization. Algorithms and techniques*. Springer, 2007, pp. 16–28.

[10] BALSEIRO, S. R., AND GUR, Y. Learning in repeated auctions with budgets: Regret minimization and equilibrium. *Management Science 65*, 9 (2019), 3952–3968.

[11] BAMAS, E., GARG, P., AND ROHWEDDER, L. The submodular santa claus problem in the restricted assignment case, 2020.

[12] BANSAL, N., AND SVIRIDENKO, M. The santa claus problem. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 2006), STOC '06, Association for Computing Machinery, p. 31–40.

[13] BEZÁKOVÁ, I., AND DANI, V. Allocating indivisible goods. *ACM SIGecom Exchanges 5*, 3 (2005), 11–18.

[14] BLANCHARD, P., HIGHAM, D. J., AND HIGHAM, N. J. Accurately computing the log-sum-exp and softmax functions. *IMA Journal of Numerical Analysis 41*, 4 (2021), 2311–2330.

[15] BLUM, A., SANDHOLM, T., AND ZINKEVICH, M. Online algorithms for market clearing. *Journal of the ACM (JACM) 53*, 5 (2006), 845–879.

[16] BORODIN, A., AND EL-YANIV, R. *Online computation and competitive analysis*. cambridge university press, 2005.

[17] BUCHBINDER, N., JAIN, K., AND NAOR, J. S. Online primal-dual algorithms for maximizing ad-auctions revenue. In *European Symposium on Algorithms* (2007), Springer, pp. 253–264.

[18] CHAKRABARTY, D., CHUZHOY, J., AND KHANNA, S. On allocating goods to maximize fairness. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA* (2009), IEEE Computer Society, pp. 107–116.

[19] CHEN, X., AND QIN, S. On-line machine covering on two machines with local migration. *Computers & Mathematics with Applications 62*, 5 (2011), 2336–2341.

[20] CHENG, S.-W., AND MAO, Y. Restricted max-min fair allocation, 2018.

[21] CHENG, S.-W., AND MAO, Y. Restricted max-min allocation: Approximation and integrality gap, 2019.

[22] CONITZER, V., KROER, C., SODOMKA, E., AND STIER-MOSES, N. E. Multiplicative pacing equilibria in auction markets. *Operations Research* (2021).

[23] DAVIES, S., ROTHVOSS, T., AND ZHANG, Y. A tale of santa claus, hypergraphs and matroids, 2019.

[24] DEVANUR, N. R., AND HAYES, T. P. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce* (2009), pp. 71–78.

[25] DEVANUR, N. R., AND JAIN, K. Online matching with concave returns. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing* (2012), pp. 137–144.

[26] DEVANUR, N. R., JAIN, K., SIVAN, B., AND WILKENS, C. A. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *Proceedings of the 12th ACM conference on Electronic commerce* (2011), pp. 29–38.

[27] EPSTEIN, L. A survey on makespan minimization in semi-online environments. *Journal of Scheduling 21*, 3 (2018), 269–284.

[28] EPSTEIN, L., LEVIN, A., AND VAN STEE, R. Max-min online allocations with a reordering buffer. *SIAM Journal on Discrete Mathematics 25*, 3 (2011), 1230–1250.

[29] FEIGE, U. On allocations that maximize fairness. In *SODA* (2008), vol. 8, Citeseer, pp. 287–293.

[30] FELDMAN, J., HENZINGER, M., KORULA, N., MIRROKNI, V. S., AND STEIN, C. Online stochastic packing applied to display ad allocation. In *European Symposium on Algorithms* (2010), Springer, pp. 182–194.

[31] FELDMAN, J., KORULA, N., MIRROKNI, V., MUTHUKRISHNAN, S., AND PÁL, M. Online ad assignment with free disposal. In *International workshop on internet and network economics* (2009), Springer, pp. 374–385.

[32] FELDMAN, J., MEHTA, A., MIRROKNI, V., AND MUTHUKRISHNAN, S. Online stochastic matching: Beating 1-1/e. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science* (2009), IEEE, pp. 117–126.

[33] GAO, B., AND PAVEL, L. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805* (2017).

[34] GOEL, G., AND MEHTA, A. Online budgeted matching in random input models with applications to adwords. In *SODA* (2008), vol. 8, Citeseer, pp. 982–991.

[35] GOEL, G., AND MEHTA, A. Online budgeted matching in random input models with applications to adwords. In *SODA* (2008), vol. 8, Citeseer, pp. 982–991.

[36] GOLLAPUDI, S., AND PANIGRAHI, D. Fair allocation in online markets. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management* (2014), pp. 1179–1188.

[37] GUPTA, A., AND MOLINARO, M. How experts can solve lps online. In *European Symposium on Algorithms* (2014), Springer, pp. 517–529.

[38] HAJIAGHAYI, M. T., KLEINBERG, R., AND SANDHOLM, T. Automated online mechanism design and prophet inequalities. In *AAAI* (2007), vol. 7, pp. 58–65.

[39] HE, Y., AND JIANG, Y. Optimal semi-online preemptive algorithms for machine covering on two uniform machines. *Theoretical Computer Science 339*, 2-3 (2005), 293–314.

[40] JANSEN, K., AND ROHWEDDER, L. A note on the integrality gap of the configuration lp for restricted santa claus, 2018.

[41] KARANDE, C., MEHTA, A., AND TRIPATHI, P. Online bipartite matching with unknown distributions. In *Proceedings of the forty-third annual ACM symposium on Theory of computing* (2011), pp. 587–596.

[42] KARP, R. M., VAZIRANI, U. V., AND VAZIRANI, V. V. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing* (1990), pp. 352–358.

[43] KAWASE, Y., AND SUMITA, H. Online max-min fair allocation. *arXiv preprint arXiv:2111.07235* (2021).

[44] KESSELHEIM, T., TÖNNIS, A., RADKE, K., AND VÖCKING, B. Primal beats dual on online packing lps in the random-order model. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing* (2014), pp. 303–312.

[45] KLEINBERG, R. A multiple-choice secretary algorithm with applications to online auctions. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms* (2005), Citeseer, pp. 630–631.

[46] LENSTRA, J. K., SHMOYS, D. B., AND TARDOS, É. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical programming 46*, 1 (1990), 259–271.

[47] LIPTON, R. J., MARKAKIS, E., MOSSEL, E., AND SABERI, A. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce* (2004), pp. 125–131.

[48] LIU, F.-H., LIU, H.-H., AND WONG, P. W. Greedy is optimal for online restricted assignment and smart grid scheduling for unit size jobs. *Theory of Computing Systems 65*, 6 (2021), 1009–1032.

[49] MAHDIAN, M., AND YAN, Q. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In *Proceedings of the forty-third annual ACM symposium on Theory of computing* (2011), pp. 597–606.

[50] MEHTA, A., SABERI, A., VAZIRANI, U., AND VAZIRANI, V. Adwords and generalized online matching. *Journal of the ACM (JACM) 54*, 5 (2007), 22–es.

[51] MOLINARO, M. Online and random-order load balancing simultaneously. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms* (2017), SIAM, pp. 1638–1650.

[52] MOLINARO, M., AND RAVI, R. The geometry of online packing linear programs. *Mathematics of Operations Research 39*, 1 (2014), 46–59.

[53] NIELSEN, F., AND SUN, K. Guaranteed bounds on information-theoretic measures of univariate mixtures using piecewise log-sum-exp inequalities. *Entropy 18*, 12 (2016), 442.

[54] PANIGRAHI, D., DAS SARMA, A., AGGARWAL, G., AND TOMKINS, A. Online selection of diverse results. In *Proceedings of the fifth ACM international conference on Web search and data mining* (2012), pp. 263–272.

[55] SEIDEN, S. S. On the online bin packing problem. *Journal of the ACM (JACM) 49*, 5 (2002), 640–671.

[56] STEIHAUS, H. The problem of fair division. *Econometrica 16* (1948), 101–104.

[57] TAN, Z., AND CAO, S. Semi-online machine covering on two uniform machines with known total size. *Computing 78*, 4 (2006), 369–378.

[58] WU, Y., CHENG, T., AND JI, M. Optimal algorithms for semi-online machine covering on two hierarchical machines. *Theoretical Computer Science 531* (2014), 37–46.

[59] ZHOU, Y., CHAKRABARTY, D., AND LUKOSE, R. Budget constrained bidding in keyword auctions and online knapsack problems. In *International Workshop on Internet and Network Economics* (2008), Springer, pp. 566–576.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Our results here are purely theoretical.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Some proofs are deferred to appendix due to space constraints.
3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [N/A]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]