Improved Algorithms for Fair Matroid Submodular Maximization (full version)

Sepideh Mahabadi
Microsoft Research
smahabadi@microsoft.com

Sherry Sarkar*
Carnegie Mellon University
sherrys@andrew.cmu.edu

Jakub Tarnawski
Microsoft Research
jakub.tarnawski@microsoft.com

Abstract

Submodular maximization subject to matroid constraints is a central problem with many applications in machine learning. As algorithms are increasingly used in decision-making over datapoints with sensitive attributes such as gender or race, it is becoming crucial to enforce fairness to avoid bias and discrimination. Recent work has addressed the challenge of developing efficient approximation algorithms for fair matroid submodular maximization. However, the best algorithms known so far are only guaranteed to satisfy a relaxed version of the fairness constraints that loses a factor 2, i.e., the problem may ask for ℓ elements with a given attribute, but the algorithm is only guaranteed to find $\lfloor \ell/2 \rfloor$. In particular, there is no provable guarantee when $\ell = 1$, which corresponds to a key special case of perfect matching constraints.

In this work, we achieve a new trade-off via an algorithm that gets arbitrarily close to full fairness. Namely, for any constant $\varepsilon > 0$, we give a constant-factor approximation to fair monotone matroid submodular maximization that in expectation loses only a factor $(1 - \varepsilon)$ in the lower-bound fairness constraint. Our empirical evaluation on a standard suite of real-world datasets – including clustering, recommendation, and coverage tasks – demonstrates the practical effectiveness of our methods.

1 Introduction

Machine learning is increasingly deployed in high-stakes decision-making, raising concerns about the propagation of bias and unfairness in automated systems. These challenges are especially acute in domains such as education, law enforcement, hiring, and credit [MMD16; Whi22; Eur22]. In response, a growing body of research has focused on developing algorithms that incorporate fairness constraints for core problems including clustering [CKLV17], data summarization [CKSDKV18], classification [ZVGG17], voting [CHV18], and ranking [CSV18].

This paper studies fairness in the context of monotone submodular maximization subject to matroid constraints. Submodular functions, which capture the principle of diminishing returns, are fundamental to a range of machine learning applications such as recommender systems [EG11], feature selection [DK11], active learning [GK11], and data summarization [LB11]. Matroids provide a

*Part of this work was done while an intern at Microsoft Research.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

general framework for modeling independence constraints, encompassing cardinality, partition, graph connectivity, and linear independence constraints.

While numerous fairness definitions have been proposed, we adopt a widely used group fairness model, which partitions the universe into *disjoint* groups and enforces *lower and upper* bounds on the representation of each sensitive group in the selected set. See Section 2.1 for a precise definition. This model generalizes several fairness notions, such as proportional representation [Mon95; BLS17], diversity constraints [CCRL13; Bid06], and statistical parity [DHPRZ12]. It has been used for both submodular maximization [CSV18; CHV18; EMNTT20; EFNTT23; WFM21; TY23; YT23; ETNV24] as well as a multitude of other optimization problems, such as clustering [CKLV17; KAM19; JNN20; HMV23], voting [CHV18], data summarization [CKSDKV18], matching [CKLV19] or ranking [CHV18].

In the absence of fairness constraints, monotone submodular maximization under a single matroid constraint is very well understood, as a tight $(1-1/e) \approx 0.63$-approximation is achievable [CCPV11; Fei98]. The intersection of two matroid constraints (which we refer to as "matroid intersection") admits an almost 0.5-approximation [LSV10]. The fair variant has been primarily explored under cardinality constraints [CHV18], where a tight $(1-1/e)$-approximation is also known. In the (single-pass) streaming setting, there is a 0.3178-approximation [FLNSZ22] for the non-fair matroid version; furthermore, since the intersection of cardinality constraint and fairness can be reduced to a single matroid constraint [EMNTT20], the same approximation factor can be obtained for it.

However, the intersection of a matroid constraint and a fairness constraint seems significantly more challenging, and is still poorly understood despite two recent works devoted to studying this problem in the streaming [EFNTT23] and the classic offline [ETNV24] settings; our focus is on the latter. Following [EFNTT23], we refer to the problem as Fair Matroid Monotone Submodular Maximization (**FMMSM**). To appreciate its difficulty, consider a key special case, Monotone Submodular Perfect Matching (**MSPM**), i.e., maximizing a monotone submodular function over the collection of all *perfect* matchings in a *bipartite* graph (V_G, E_G).[2] This collection of feasible sets is not downward-closed, which invalidates known algorithmic approaches.[3] The best known approximation factor for MSPM is a trivial $O(|V_G|)$-approximation; one can also apply the framework of [GHIM09] to obtain an $\widetilde{O}(\sqrt{|E_G|})$-approximation[4], which is superior for sparse graphs. In fact, this could possibly even be tight, as it almost matches a surprising negative result of [ETNV24] who showed a family of sparse graphs where the standard *multilinear relaxation* (commonly used in relax-and-round approaches for submodular optimization) has an integrality gap of $\Omega(\sqrt{|E_G|})$. The existence of a constant-factor approximation to MSPM was posed by [ETNV24] as an exciting open problem.

The algorithms given in [EFNTT23; ETNV24] for FMMSM circumvent the difficulty posed by the lower bound constraints by relaxing them. They obtain the following two results:

Theorem 1.1 (Two-pass algorithm of [EFNTT23]) *There is a polynomial-time algorithm for FMMSM that violates lower bound constraints by a factor 2 and obtains $\alpha/2$-approximation, where α is the approximation ratio of an algorithm for maximizing a monotone submodular function under a matroid intersection constraint.*

We can have α be almost $1/2$ [LSV10] and thus get an almost $1/4$-approximation. ([EFNTT23] work in the streaming setting and instead use the streaming algorithm for matroid intersection of [GJS21]; this results in a $1/11.66$-approximation in two passes.) Here, violating lower bound constraints by a

[2]To see why MSPM is a special case of FMMSM, set E_G as the universe, consider a partition matroid that encodes that every vertex on the left shall have degree at most 1 in the solution, and set fairness constraints so that every vertex on the right shall have degree at least 1 and at most 1.

[3]Of course, a proper subset of a perfect matching is not a perfect matching. But more importantly, the collection of all *subsets of perfect matchings* (which is downward-closed) does not belong to any of the families that are known to make approximate submodular maximization tractable. In particular, it is not a matroid, an intersection of a small number of matroids, or a so-called p-extendible set system or a p-system [CCPV11] for $p = O(1)$.

[4]The work [GHIM09] shows that we can in polynomial time obtain numbers c_e for $e \in E_G$ such that for any $S \subseteq E_G$, the function $\hat{f}(S) := \sqrt{\sum_{e\in S} c_e}$ is an $\widetilde{O}(\sqrt{|E_G|})$-approximation to $f(S)$. Maximizing $\hat{f}(S)$ amounts to maximizing $\sum_{e\in S} c_e$, which is the maximum-weight bipartite perfect matching problem, solvable in polynomial time.

factor 2 means that, if a color has a lower bound of ℓ, the solution is guaranteed to have at least $\lfloor \ell/2 \rfloor$ elements of that color. Note that in MSPM we have $\ell = 1$ and thus $\lfloor \ell/2 \rfloor = 0$.

Theorem 1.2 ([ETNV24]) *There is a polynomial-time algorithm for FMMSM that satisfies* lower and upper *bound constraints in expectation rather than exactly, and obtains a* $(1-1/e)$*-approximation in expectation.*

Theorem 1.2 also guarantees certain two-sided tail bounds on the violation of each fairness constraint which apply if ℓ is large enough. It is the only algorithm considered in this paper that violates the *upper* bounds. The algorithm proceeds by solving and rounding the multilinear relaxation.

If we consider a relaxed version of MSPM where instead of a *perfect* matching we want a *large* matching that also has high submodular function value, then a simple greedy algorithm will yield a $1/3$-approximation (Theorem 2.6) and construct a maximal matching, thus getting $1/2$ of the maximum possible size. The results in Theorems 1.1 and 1.2 give no improvement upon this.[5] While one can try to generalize this simple approach to FMMSM, it faces another issue that is salient in the context of fairness motivations: while at least half of the total lower bound mass will be satisfied, there could be “unlucky” colors (marginalized groups) that never get represented in the solution; this is precisely the reason why we seek fair algorithms in the first place.

1.1 Our contributions

In this work we provide an algorithm that satisfies the fairness constraints within a factor better than 2, while also giving guarantees for every individual group (rather than only in aggregate like the simple greedy strategy discussed above). To achieve the former objective, we trade off part of the objective value; to achieve the latter, we employ randomization.

Theorem 1.3 (informal version of Theorem 3.4) *For every* $\varepsilon \in (0, 1)$ *there is a polynomial-time algorithm for FMMSM whose output*

- *satisfies the matroid constraint,*
- *satisfies fairness upper bound constraints,*
- *for a group with fairness lower bound* ℓ*, has in expectation at least* $(1-\varepsilon)\ell$ *elements from that group,*
- *has expected size at least* $(1-\varepsilon)$ *times the maximum size of any feasible solution,*
- *satisfies Chernoff-style high-probability bounds on size, as well as total fairness violation,*
- *has expected submodular function value at least* $0.499 \cdot \varepsilon \cdot \mathrm{OPT}$.

Our bound on the submodular function value is actually shown with respect to a more powerful optimum, namely, an optimal set that satisfies the matroid and upper-bound constraints, but not necessarily the lower-bound constraints. If one wants to compare to this optimum, then the $O(\varepsilon)$ factor loss in value is unavoidable. To see this, consider MSPM in a graph $P_3 \times N$ consisting of a disjoint union of N paths of length 3, with a linear objective function assigning values $0, 1, 0$ to each path's edges. A perfect matching of size $2N$ has 0 value, and a maximal matching of size N has value N; one can interpolate between these smoothly.

We note that by instantiating $\varepsilon = 1/2$ we obtain an almost $1/4$-approximation while violating lower bounds by a factor 2, which is similar to the bounds of Theorem 1.1 ([EFNTT23]).

[5] The result of Theorem 1.2 ([ETNV24]) only satisfies the upper bounds in expectation, so to obtain a feasible solution for MSPM (i.e., a matching), one needs to delete some violating edges from the solution, which will damage the objective value and the solution cardinality. Therefore, as stated, the result gives no theoretical guarantees for MSPM. By opening up the algorithm and its proof, one can use the bounds for randomized swap rounding [CVZ10, Theorem II.3] to get some meaningful bound: $\mathbb{P}[|S \cap \delta(v)| \geq 1] \geq 1 - e^{-1/2} \approx 0.4$. This means that, for a bipartite graph with $n + n = 2n$ vertices, $0.4n$ right-side vertices will be guaranteed to have an incident edge in S; when we select one edge per right-side vertex, we can expect a matching of size $0.4n$, and one can perhaps hope for a similar bound on submodular value, i.e., $0.4(1 - 1/e)\,\mathrm{OPT} \approx 0.25\,\mathrm{OPT}$. But a simple greedy algorithm gets $0.5n$ and $\mathrm{OPT}/3$, respectively.

As a second contribution, we also employ our techniques to obtain a deterministic algorithm. There are several variants that we could formulate; we choose to show a general setting of matroid intersection, where the trade-off is between size and objective value. The relation to fairness is that an algorithm that finds a solution of maximum size that is an α-approximation to the objective value would imply an α-approximation algorithm for FMMSM (see [EFNTT23], Proposition C.6).

Theorem 1.4 (informal version of Theorem 3.6) *For every $\varepsilon \in (0, 1)$ there is a deterministic polynomial-time algorithm for the problem of maximizing a monotone submodular function subject to two matroid constraints whose output has size at least $(1 - \varepsilon)$ times the maximum size of any feasible solution minus one, and obtains a $(0.499 \cdot \varepsilon)$-approximation to the submodular function value.*

Experimental results. We show the effectiveness of our algorithm empirically against prior work and natural baselines on a suite of standard benchmarks. We measure the submodular objective value and total fairness violation. Our algorithms produce solutions whose value is competitive with the highest-value baseline, which completely ignores the lower bound constraints and accordingly has the highest fairness violations. In two out of three scenarios, our algorithms dominate prior work [EFNTT23]. Finally, a key strength of our approach is the flexibility given by ε, allowing users to tune the balance between utility and fairness.

Our techniques. Let us begin with the simple setting of perfect matchings (MSPM). Consider the symmetric difference of a high-value matching Y and a perfect matching P (where Y might be small and P might have no value). This decomposes into a collection of vertex-disjoint alternating cycles and augmenting paths.

One possible algorithm is to ignore the cycles, and choose some of the augmenting paths to apply to Y, so that its size grows to at least $(1 - \varepsilon)|P|$. We can do this by computing the marginal contribution of the elements that Y would lose in each path, and taking the least damaging paths; by submodularity, this loses at most a $(1 - \varepsilon)$ fraction of value in Y.

While this does ensure a large matching, some ε fraction of vertices can still be "unlucky" and end up unmatched. Deterministically this would be hard to avoid (short of solving MSPM/FMMSM completely, with no fairness violation); our next idea is to choose the paths randomly in the above solution. This will work for MSPM, as long as we take care to select a $(1-\varepsilon)$ fraction of the $|P|-|Y|$ many augmenting paths, even if we already have $|Y| \geq (1 - \varepsilon)|P|$. Then every vertex that was not matched in Y has a $(1 - \varepsilon)$ probability of being matched in the new solution.

However, there are two main challenges when trying to generalize the above approach to matroid and fairness constraints. Firstly, having fairness bounds with $\ell_c < u_c$ means that Y can have fewer elements than P in some colors but more elements in other colors, and can even have $|Y| = |P|$ while still violating many fairness lower bounds. This means that we need to find and apply not only augmenting paths, but also alternating paths that exchange an element of an oversaturated color for one of an undersaturated color, without increasing the solution size. We show that as long as the total fairness violation is large, there are many such disjoint paths, which implies that applying a random fraction of them still retains enough value.

The second, larger obstacle arises due to dealing with general matroids. We are able to use tools from matroid theory to show the existence of many disjoint alternating or augmenting paths in an appropriate matroid intersection exchange graph whose vertices correspond to elements of Y and P (which were edges in the case of MSPM). We need to carefully refine the paths via an asymmetric shortcutting process to ensure that applying them leaves the solution independent in the matroid while also not disrupting the counts of elements in the colors not being exchanged. Moreover, in general, multiple augmenting paths in matroids cannot be applied simultaneously. We deal with this using an iterative framework where we apply a single path, rebuild the exchange graph, and find a new large collection of disjoint paths; we then bound the loss in value after each step.

Paper organization. We discuss more related work in Section 1.2. In Section 2 we introduce all necessary notation, definitions, and useful facts. In Section 3 we describe our algorithms and prove their properties. Section 4 is devoted to the experimental evaluation. We conclude and discuss the limitations and broader impact of our work in Section 5.

1.2 Additional related work

The *non-monotone* Fair Matroid Submodular Maximization problem was explored by [YT23] under the cardinality constraint. They achieved a 0.2005-approximation for the special case where for all colors c, we have $\ell_c/|V_c| = a$ and $u_c/|V_c| = b$ for some constants $a, b \in [0, 1]$. Later, [ETNV24] recovered and further generalized their results. In particular for general matroids, they achieved a $(1-\beta)/(8+\varepsilon)$ approximation algorithm that guarantees the number of elements from each group c is between $\lfloor \beta\ell_c \rfloor$ and u_c for a trade-off parameter $\beta \in [0, 1/2]$.

In this work, we consider the setting where the color groups are disjoint. The more general case, where groups may overlap, was previously studied by [CHV18] for the special case of FMMSM with a cardinality constraint. They show that when elements can belong to three or more groups, simply checking the feasibility becomes NP-hard. However, by allowing for violations of the fairness constraints and in particular guaranteeing the fairness constraint in expectation, they gave a $(1 - 1/e - o(1))$-approximation algorithm for the problem.

An alternative notion of fairness in submodular maximization has been explored in [TWRTZ19; TY23; WLBW24], where the focus is on ensuring that each group – potentially not limited to subsets of the ground set V – receives at least a specified amount of value from the selected solution. In these formulations, the value is modeled using a monotone submodular function. This line of work can be cast as a multi-objective submodular maximization problem [KMGG08; CVZ10; Udw18].

2 Preliminaries

We denote the symmetric difference $(X \setminus Y) \cup (Y \setminus X)$ of two sets X and Y by $X \triangle Y$.

Submodular functions. We consider functions $f : 2^V \to \mathbb{R}_+$ defined on a ground set V. We say that f is *submodular* if $f(Y \cup \{e\}) - f(Y) \geq f(X \cup \{e\}) - f(X)$ for any two sets $Y \subseteq X \subseteq V$ and any element $e \in V \setminus X$. Moreover, f is *monotone* if $f(Y) \leq f(X)$ for any two sets $Y \subseteq X \subseteq V$. We assume that f is given as an oracle that computes $f(S)$ for given $S \subseteq V$; we consider the running time of this oracle to be $O(1)$.

The following fact is folklore. We provide a proof for completeness.

Fact 2.1 *Let f be a non-negative submodular function and $X_1, X_2, ..., X_k \subseteq X$ be disjoint subsets of X. Then*

$$\sum_{i=1}^{k} f(X \setminus X_i) \geq (k-1) f(X).$$

Proof. We use induction on k. The base case $k = 1$, i.e., that $f(X \setminus X_1) \geq 0$, follows because $f \geq 0$. For $k > 1$, we apply the inductive hypothesis to the set family $X_1 \cup X_2, X_3, X_4, ..., X_k$. We get

$$f(X \setminus (X_1 \cup X_2)) + \sum_{i=3}^{k} f(X \setminus X_i) \geq (k-2) f(X).$$

By submodularity,

$$f(X \setminus X_1) + f(X \setminus X_2) - f(X \setminus (X_1 \cup X_2)) \geq f(X).$$

Adding these two inequalities gives the statement. □

Matroids. A *matroid* is a set family $\mathcal{I} \subseteq 2^V$ with the properties:

- *Downward-closedness*: if $X \subseteq Y$ and $Y \in \mathcal{I}$, then $X \in \mathcal{I}$;
- *Augmentation*: if $X, Y \in \mathcal{I}$ and $|X| < |Y|$, then there exists $e \in Y$ with $X + e \in \mathcal{I}$.

We abbreviate $X \cup \{e\}$ as $X + e$ and $X \setminus \{e\}$ as $X - e$. We assume that the matroid is given as an oracle that, for a given $S \subseteq V$, answers whether $S \in \mathcal{I}$; we consider the running time of this oracle to be $O(1)$. We say that a set $S \subseteq V$ is *independent* if $S \in \mathcal{I}$.

Matroid exchange graph. Let $\mathcal{I}$ be a matroid on universe V and Y, Z be two independent sets.

Definition 2.2 *We define the exchange graph for Y and Z with respect to $\mathcal{I}$ as the bipartite graph*

$$(Y \setminus Z, Z \setminus Y, \{(y, z) : Y - y + z \in \mathcal{I}\}).$$

Lemma 2.3 ([Sch03], Corollary 39.12a) *If $|Y| = |Z|$, then the exchange graph for Y and Z with respect to $\mathcal{I}$ contains a perfect matching.*

Lemma 2.4 ([Sch03], Corollary 39.13) *Let Y be an independent set and let $Z \subseteq V$ be such that $|Z| = |Y|$. If the exchange graph for Y and Z with respect to $\mathcal{I}$ contains a unique perfect matching between $Y \setminus Z$ and $Z \setminus Y$, then Z is also an independent set.*

2.1 Fair Matroid Monotone Submodular Maximization (FMMSM)

The universe V is partitioned into C sets: $V = V_1 \cup V_2 \cup ... \cup V_C$, where V_c denotes elements of color c. Every element has exactly one color. The set of colors is denoted by $[C] = \{1, 2, ..., C\}$. For every color $c \in [C]$ we have *fairness bounds*: lower bound ℓ_c and upper bound u_c.

The set of upper bounds gives rise to a *partition matroid* that we will denote by $\mathcal{U}$. That is,

$$\mathcal{U} = \{S \subseteq V \mid |S \cap V_c| \leq u_c \ \ \forall c \in [C]\}.$$

It is well-known that such a collection of sets forms a matroid. We will call a set $S \in \mathcal{U}$ *upper-fair*.

If a set satisfies both the lower and the upper bounds, we say that it is *fair*. That is, we define the family of fair sets $\mathcal{C}$ as follows:

$$\mathcal{C} = \{S \subseteq V \mid \ell_c \leq |S \cap V_c| \leq u_c \ \ \forall c \in [C]\}.$$

The FMMSM problem asks to find a set $S \in \mathcal{I} \cap \mathcal{C}$ (i.e., fair and independent S) that maximizes $f(S)$. We use OPT for the optimal value, i.e., $\mathrm{OPT} = \max_{S \in \mathcal{I} \cap \mathcal{C}} f(S)$. We assume that there exists a fair and independent set, i.e., $\mathcal{I} \cap \mathcal{C} \neq \emptyset$. We say that an algorithm is an α-approximation if it outputs a set S with $f(S) \geq \alpha \cdot \mathrm{OPT}$.

For any set $S \subseteq V$ we define its *fairness violation* $\mathrm{fav}(S) := \sum_c \max\{|S \cap V_c| - u_c, \ell_c - |S \cap V_c|, 0\}$. Note that if S is upper-fair, then $\mathrm{fav}(S) = \sum_c \max\{\ell_c - |S \cap V_c|, 0\}$.

Lemma 2.5 ([EFNTT23], Appendix C) *There is an exact polynomial-time algorithm for FMMSM for the case when f is a linear function.*

Matroid intersection. Given two matroids and a monotone submodular function f defined on V, we can define the problem of maximizing a submodular function subject to a matroid intersection constraint similarly to FMMSM.

In particular, if we ignore the lower bounds completely, FMMSM turns into the above matroid intersection problem for matroids $\mathcal{I}$ and $\mathcal{U}$.

Theorem 2.6 ([CCPV11]) *The greedy algorithm gives a $1/3$-approximation to this problem.*

Theorem 2.7 ([LSV10]) *For any $\delta > 0$ there is a polynomial-time algorithm that gives a $(0.5 - \delta)$-approximation to this problem.*

3 Our algorithm

In this section we describe our algorithms: randomized (Theorem 3.4) and deterministic (Section 3.1, Theorem 3.6). See also the pseudocode provided in Algorithm 2. We first need to introduce some notions.

The proof of Theorem 3.4 will begin by constructing a maximum-cardinality independent and fair set P, which will stay unchanged throughout the execution. We also construct an independent and upper-fair set Y of high f-value. We will use P as a source of fairness and iteratively trade off Y's value for P's elements in colors that are undersaturated by Y.

Definition 3.1 *Given Y and P as above, we say that a color $c \in [C]$ is* undersaturated *if $|Y \cap V_c| < |P \cap V_c|$, and* oversaturated *if $|Y \cap V_c| > |P \cap V_c|$.*

The technical crux of the proof of Theorem 3.4 is Lemma 3.3, in which we show the existence of many disjoint structures, each of which can be used to advance our fairness objective. We will call them augmenting or alternating, as they indeed correspond to such paths in the appropriately defined matroid intersection exchange graph that we consider in the proof of Lemma 3.3.

Definition 3.2 *Let Y be an independent and upper-fair set, and let $X \subseteq V$. Define the result Y' of applying X to Y as the symmetric difference $Y' = Y \triangle X$. We say that X is* alternating *(with respect to Y) if Y' is independent ($Y' \in \mathcal{I}$) and there is exactly one undersaturated color $c' \in [C]$ and one oversaturated color $c'' \in [C]$ such that for all $c \in [C]$,*

$$|Y' \cap V_c| = |Y \cap V_c| + \begin{cases} 1 & \text{for } c = c', \\ -1 & \text{for } c = c'', \\ 0 & \text{for } c \neq c', c''. \end{cases}$$

We say that X is augmenting *if all the above conditions are satisfied, except that there is no color c''.*

In both cases, we say that X increases *c'.*

Note that we have $|Y'| = |Y|$ if X is alternating and $|Y'| = |Y| + 1$ if X is augmenting. Also, Y' is upper-fair, since the only color where it has more elements than Y is c', and we have $|Y' \cap V_{c'}| = |Y \cap V_{c'}| + 1 < |P \cap V_{c'}| + 1$ (and P is fair).

Lemma 3.3 *Let Y and P be two independent and upper-fair sets with $|Y| \leq |P|$. Denote*

$$k = \sum_{c \in [C]} \max(0, |P \cap V_c| - |Y \cap V_c|) .$$

Then we may find in polynomial time a collection $X_1, ..., X_k$ of disjoint subsets of $Y \cup P$, of which at least $|P| - |Y|$ many are augmenting and the rest are alternating. Moreover, for every undersaturated color c, exactly $|P \cap V_c| - |Y \cap V_c|$ many of the paths increase c.

Proof. To simplify notation, we assume without loss of generality that $Y \cap P = \emptyset$. Otherwise we could work with sets $Y \setminus P$ and $P \setminus Y$.

Consider the matroid intersection exchange graph for Y and P with respect to matroids $\mathcal{I}$ and $\mathcal{U}$. This is defined as the *directed* bipartite graph obtained by taking the union of the exchange graph for Y and P with respect to $\mathcal{I}$, whose edges we direct right-to-left (from P to Y), and of the exchange graph for Y and P with respect to $\mathcal{U}$, whose edges we direct left-to-right (from Y to P). That is, we have edges

$$\{y \to p : Y + p - y \in \mathcal{U}\} \quad \text{and} \quad \{y \leftarrow p : Y + p - y \in \mathcal{I}\} .$$

Inside this graph we will carefully construct a subgraph consisting of two matchings $M_\to$ (directed left-to-right) and $M_\leftarrow$ (directed right-to-left). The augmenting and alternating paths will be found in that subgraph.

To construct $M_\leftarrow$, we first define $T_P := \{p \in P : Y + p \in \mathcal{I}\}$. We have $|T_P| \geq |P| - |Y|$ (by repeated application of the matroid augmentation property). We will call the elements in T_P *P-sinks*. Let T'_P be an arbitrary subset of T_P of size exactly $|P| - |Y|$. We then invoke Lemma 2.3 on the exchange graph for Y and $P \setminus T'_P$ with respect to $\mathcal{I}$ (which is a subgraph of our matroid intersection exchange graph). It implies the existence of a perfect matching between Y and $P \setminus T'_P$; since one exists, we can find one in polynomial time. We obtain the matching $M_\leftarrow$ by removing the edges of that matching that are incident to $T_P \setminus T'_P$. Then, $M_\leftarrow$ matches every vertex of $P \setminus T_P$.

We construct the matching $M_\to$ manually by matching up as many elements of the same color between Y and P as possible. That is, for every color $c \in [C]$ we add $\min(|Y \cap V_c|, |P \cap V_c|)$ edges from $Y \cap V_c$ to $P \cap V_c$ to the matching $M_\to$.

We define the set S of *sources* as all vertices in P that did not get matched in $M_\to$. Note that they are only in undersaturated colors, and their number is exactly k. (In principle it is possible to have a source that is also a P-sink; this can happen if Y is not maximal in $\mathcal{I} \cap \mathcal{U}$.)

We also define the set T_Y of Y*-sinks* as all vertices in Y that did not get matched in $M_{\rightarrow}$. Note that they are only in oversaturated colors, and their number is exactly $k - (|P| - |Y|)$, as we have

$$\begin{aligned}
|P| - |Y| &= \sum_{c \in [C]} |P \cap V_c| - |Y \cap V_c| \\
&= \sum_{c\text{: undersaturated}} (|P \cap V_c| - |Y \cap V_c|) - \sum_{c\text{: oversaturated}} (|Y \cap V_c| - |P \cap V_c|) \\
&= k - |T_Y|.
\end{aligned}$$

To recap, we have k sources S (all in P), $k - (|P| - |Y|)$ Y-sinks T_Y, and at least $|P| - |Y|$ P-sinks T_P. Moreover, for every undersaturated color c, exactly $|P \cap V_c| - |Y \cap V_c|$ many of the sources are of color c.

Now we show how to construct k vertex-disjoint simple paths in $M_{\rightarrow} \cup M_{\leftarrow}$ that start at sources (S) and end at sinks ($T_Y \cup T_P$). For every path, we proceed as follows:

- start at at unused source (in P),
- whenever at a vertex of P, stop if that vertex is a sink (in T_P); otherwise it has an incident outgoing edge of $M_{\leftarrow}$; follow this edge,
- whenever at a vertex of Y, stop if that vertex is a sink (in T_Y); otherwise it has an incident outgoing edge of $M_{\rightarrow}$; follow this edge.

Since every path must terminate at a different sink, at least $k - (k - (|P| - |Y|)) = |P| - |Y|$ of the P-sinks will be used. Furthermore, a path cannot revisit a vertex, since the indegree of every vertex is at most 1 and sources have no incoming edges. This implies that all paths are simple and vertex-disjoint.

The k paths constructed above might not yet be augmenting/alternating paths in the matroid intersection exchange graph, as they may contain chords; in general, only chordless paths guarantee that applying them preserves independence. (Matroid intersection algorithms usually apply shortest paths, which are chordless.) We need to shortcut them; however, doing so naively could destroy the property that all left-to-right edges in the paths are between elements of the same color, which we require to satisfy the condition in Definition 3.2.

We carry out the shortcutting as follows. Let $X' = (p_1, y_1, p_2, y_2, ...)$ be one of the k paths. As long as there exists a chord of the form (p_i, y_j) with $j > i$ (i.e., the directed edge $y_j \leftarrow p_i$ exists in the matroid intersection exchange graph; equivalently, $Y + p_i - y_j \in \mathcal{I}$), replace the corresponding subpath with this chord (i.e., remove the vertices $y_i, p_{i+1}, ..., p_j$ from the sequence X'). Note that we do not use chords of the form (y_i, p_j) with $j > i$. Doing this to each of the k paths obtains our final collection $X_1, ..., X_k$.

We now verify that it satisfies the statement of the lemma. As the paths before shortcutting were vertex-disjoint, they remain so afterwards. We claim that the paths ending at P-sinks (recall that there are at least $|P| - |Y|$ many) yield augmenting sets, and the paths ending at Y-sinks yield alternating sets. Consider a path $X_i = (p_1, y_1, p_2, y_2, ...)$. Note that $Y' = Y \triangle X_i = Y \cup \{p_1, p_2, ...\} \setminus \{y_1, y_2, ...\}$. The color-count condition of Definition 3.2 follows easily from the property that every left-to-right edge $y_i \rightarrow p_{i+1}$ in X_i belongs to $M_{\rightarrow}$, so y_i and p_{i+1} are of the same color. Thus we can take c' to be the color of p_1. If the last element of X_i is in Y (a Y-sink), we take c'' to be its color.

It remains to show that $Y' = Y \triangle X_i \in \mathcal{I}$. This argument closely follows that of [Sch03], Theorem 41.2. Let us first consider the case where X_i ends at a Y-sink: $X_i = (p_1, y_1, p_2, y_2, ..., p_t, y_t)$. We want to apply Lemma 2.4 on the exchange graph for Y and Y' with respect to $\mathcal{I}$. This is a bipartite graph on $\{y_1, ..., y_t\}$ on the left side and $\{p_1, ..., p_t\}$ on the right side, and it is equal to the corresponding induced subgraph of edges going right-to-left in the matroid intersection exchange graph; we need to show that it contains a unique perfect matching. We proceed iteratively: p_1 cannot be connected to any y_j with $j > 1$, for otherwise we would have a shortcut. So any matching must have p_1 matched to y_1. Removing these two vertices, we consider the out-neighborhood of p_2. Again, p_2 has no shortcuts to later y_j, so its only out-neighbor (after the removal of y_1) is y_2. So, p_2 must be matched to y_2. We may continue inductively to construct the unique matching between $\{y_1, ..., y_t\}$ and $\{p_1, ..., p_t\}$.

The case where $X_i = (p_1, y_1, p_2, y_2, ..., p_t, y_t, p_{t+1})$ ends at a P-sink is similar, with one more step. The first case shows that $Z = Y \cup \{p_1, ..., p_t\} \setminus \{y_1, ..., y_t\}$ is independent. We need only show that $Z + p_{t+1}$ is independent. Note that $p_{t+1} \in T_P$, meaning that $Y + p_{t+1} \in \mathcal{I}$. By the matroid augmentation property, $Y + p_{t+1}$ must have an element which can be added to Z while preserving independence. The possible candidates are $(Y + p_{t+1}) \setminus Z = \{y_1, ..., y_t, p_{t+1}\}$. However, no y_j can be added: since $p_1, ..., p_t \notin T_P$, we know that $Y \cup \{p_1, ..., p_t\}$ has rank $|Y|$, and $Z + y_j \subseteq Y \cup \{p_1, ..., p_t\}$ would have rank $|Z| + 1 = |Y| + 1$ if $Z + y_j$ were independent. Therefore the only possible candidate is p_{t+1}, and so we have that $Y' = Z + p_{t+1}$ is independent. □

Now we are ready to state and prove our main result.

Theorem 3.4 *There is a randomized polynomial-time algorithm for FMMSM parametrized by $\varepsilon \in (0, 1)$ that outputs a set $S \in \mathcal{I} \cap \mathcal{U}$ (i.e., independent and upper-fair) such that*

- $\mathbb{E}[|S|] \geq (1 - \varepsilon)N$ *with a high-probability tail bound:*
  for $\delta > 0$, $\mathbb{P}[|S| < (1 - \delta)(1 - \varepsilon)N] \leq \exp(-\Omega_\delta(N))$
- $\mathbb{E}[f(S)] \geq 0.499 \cdot \varepsilon \cdot \mathrm{OPT}_{\mathrm{MatInt}}$
- *for every $c \in [C]$ we have $\mathbb{E}[|S \cap V_c|] \geq (1 - \varepsilon)\ell_c$*
- *with a high-probability tail bound on the total fairness violation:*
  for $\delta > 0$, $\mathbb{P}[\mathrm{fav}(S) > (1 + \delta)\varepsilon \sum_c \ell_c] \leq \exp\left(-\Omega_\delta\left(\sum_c \ell_c\right)\right)$

where N is the maximum size of a set in $\mathcal{I} \cap \mathcal{U}$, and $\mathrm{OPT}_{\mathrm{MatInt}}$ is the maximum f-value of a set in $\mathcal{I} \cap \mathcal{U}$ (clearly we have $\mathrm{OPT}_{\mathrm{MatInt}} \geq \mathrm{OPT}$ as $\mathcal{C} \subseteq \mathcal{U}$).

We stress that S is upper-fair with probability 1, not only in expectation. We also remark that one can show a similar tail bound for every individual ℓ_c, though the right-hand side $\exp(-\Omega_\delta(\ell_c))$ may not be meaningful unless ℓ_c is large. On the other hand, no such bound can be shown for the f-value, which in the worst case can be concentrated on a single element of the universe.

The guarantee $\mathbb{E}[f(S)] \geq 0.499 \cdot \varepsilon \cdot \mathrm{OPT}_{\mathrm{MatInt}}$ of the second bullet point comes from using the local search algorithm of Theorem 2.7 as a subroutine. We can instead use the simpler algorithm of Theorem 2.6 to get a slightly worse guarantee of $\mathbb{E}[f(S)] \geq \frac{1}{3} \cdot \varepsilon \cdot \mathrm{OPT}_{\mathrm{MatInt}}$; we do so in our experimental evaluation.

Proof of Theorem 3.4. As the first step, we compute a maximum-cardinality fair and independent set P, which may be done in polynomial time by Lemma 2.5. We can say that $|P| = N$, i.e., the maximum size of an independent and fair set is the same as the maximum size of an independent and upper-fair set. To see this, suppose that there was an independent and upper-fair set F with $|F| > |P|$; then we could apply Lemma 3.3 to P and F to obtain an augmenting set X, and $P \triangle X$ would be a larger independent and fair set, a contradiction.

As the second step, we compute a high-value independent and upper-fair set Y_0. Using the algorithm of Theorem 2.7 ([LSV10]) (with $\delta = 10^{-3}$) we get that

$$f(Y_0) \geq 0.499 \cdot \mathrm{OPT}_{\mathrm{MatInt}} \ . \tag{1}$$

We denote

$$k(Y) = \sum_{c \in [C]} \max(0, |P \cap V_c| - |Y \cap V_c|)$$

for any solution Y, and $k := k(Y_0)$ to shorten notation.

We will perform a number I of iterations which will be $(1 - \varepsilon)k$ in expectation. More precisely, let us set $I = \lceil (1 - \varepsilon)k \rceil$ with probability $(1 - \varepsilon)k - \lfloor (1 - \varepsilon)k \rfloor$, and $\lfloor (1 - \varepsilon)k \rfloor$ otherwise.[6]

We perform I iterations. In the i-th iteration, we apply Lemma 3.3 to Y_{i-1} (and P) to obtain a collection $X_i^1, ..., X_i^{k(Y_{i-1})}$ of augmenting or alternating sets. We choose one of them, $X_i \in$

[6]Ideally we would just set $I = (1 - \varepsilon)k$, but this number can be fractional, and using a fixed value of $\lfloor (1 - \varepsilon)k \rfloor$ or $\lceil (1 - \varepsilon)k \rceil$ would lead to losses in objective value, cardinality, or fairness. For example, if $\ell_c = 1$, then a bound such as $|S \cap V_c| \geq (1 - \varepsilon)\ell_c - 1$ would be meaningless.

$\{X_i^1, ..., X_i^{k(Y_{i-1})}\}$, uniformly at random, and apply it to obtain a new solution $Y_i = Y_{i-1} \triangle X_i$. Finally, we return $S := Y_I$.

All solutions $Y_0, ..., Y_I$ are independent and upper-fair; it remains to verify the guarantees of Theorem 3.4. We start by noting that

$$k(Y_i) = k - i\,. \tag{2}$$

To see this, note that during the algorithm's execution, no new color ever becomes undersaturated, as by Definition 3.2, Y_i can have fewer elements than Y_{i-1} in a color c'' only if c'' was oversaturated in Y_{i-1}. On the other hand, for exactly one undersaturated color c', Y_i has one more element in c' than Y_{i-1}. Thus we have $k(Y_i) = k(Y_{i-1}) - 1$ and (2) follows. (Colors c that are neither under- or oversaturated remain such forever.)

Fairness lower bounds. Building upon the previous paragraph, we consider a random process involving colored balls that will mirror what is happening in the algorithm. Let $U \subseteq [C]$ be the set of colors that are undersaturated in Y_0. At the beginning, for every $c \in U$, we create $|P \cap V_c| - |Y_0 \cap V_c|$ balls of color c. (So we start with k balls in total.) At every iteration i there is exactly one color c' (that is undersaturated in Y_{i-1}, so $c' \in U$) where $|Y_i \cap V_{c'}| = |Y_{i-1} \cap V_{c'}| + 1$; we then remove one random ball of color c'. Then, by Definition 3.2 (since all other colors in U retain their element count), we have that the number of balls of every color $c \in U$ is equal to $|P \cap V_c| - |Y_i \cap V_c|$ (and their total number is $k(Y_i) = k - i$).

Now we claim that in this process, at every iteration a uniformly random ball is removed. This is because, by Lemma 3.3, for every $c \in U$, exactly $|P \cap V_c| - |Y_{i-1} \cap V_c|$ of the $k(Y_{i-1})$ augmenting or alternating sets increase c, and we choose randomly among these sets.

It follows that at the end, the set of removed I balls is distributed uniformly among all subsets of this size. Consider a color c. If $c \notin U$, then c will not be undersaturated at the end, so $|S \cap V_c| \geq |P \cap V_c| \geq \ell_c$. Now fix $c \in U$ and denote by B_c the number of removed balls of color c. Conditioning on I, we have

$$\begin{aligned}
\mathbb{E}[|S \cap V_c|] &= \mathbb{E}[|Y_0 \cap V_c| + B_c] \\
&= |Y_0 \cap V_c| + \frac{I}{k}(|P \cap V_c| - |Y_0 \cap V_c|) \\
&\geq \frac{I}{k}|P \cap V_c| \\
&\geq \frac{I}{k}\ell_c
\end{aligned}$$

and thus $\mathbb{E}[|S \cap V_c|] = \mathbb{E}[\mathbb{E}[|S \cap V_c| \mid I]] \geq \frac{\mathbb{E}[I]}{k}\ell_c = (1-\varepsilon)\ell_c$ as required.

Cardinality. Our proof that $\mathbb{E}[|S|] \geq (1-\varepsilon)|P| = (1-\varepsilon)N$ is very similar to the proof above for a single color. We start with $|P| - |Y_0|$ red balls and $k - (|P| - |Y_0|)$ non-red balls (k in total). At every iteration i, if an augmenting set was chosen (so that $|Y_i| = |Y_{i-1}| + 1$), we remove a red ball, otherwise we remove a non-red ball. Suppose that at every iteration i there were exactly $|P| - |Y_{i-1}|$ augmenting sets among the $k - i + 1$ sets; then the set of balls removed at the end would be distributed uniformly among all subsets of I balls. Then, if B denotes the number of removed red balls, we would have $\mathbb{E}[B] = \frac{I}{k}(|P| - |Y_0|)$ (conditioned on I). Now, in fact at every iteration i there are *at least* $|P| - |Y_{i-1}|$ augmenting sets among the $k - i + 1$ sets; hence, the distribution of B dominates the above uniform-ball-subset distribution, which is formally known as $\mathrm{Hypergeometric}(k, |P| - |Y_0|, I)$. In particular, this implies that $\mathbb{E}[B] \geq \frac{I}{k}(|P| - |Y_0|)$. We conclude by saying that $\mathbb{E}[|S|] = |Y_0| + \mathbb{E}[B] \geq |Y_0| + \frac{\mathbb{E}[I]}{k}(|P| - |Y_0|) \geq (1-\varepsilon)|P|$.

Cardinality tail bound. Recall that $N = |P|$, and that for any $\delta > 0$ we want to prove that $\mathbb{P}[|S| < (1-\delta)(1-\varepsilon)N] \leq \exp(-\Omega_\delta(N))$. It is known [Hoe63] that the hypergeometric distribution satisfies the same Chernoff-type bounds as the binomial distribution (as it corresponds to a sum of samples that are negatively correlated, rather than independent). In particular (conditioning on I

throughout), we have

$$\mathbb{P}\left[B < \left(1-\frac{\delta}{2}\right)\mu\right] \le \exp\left(-\frac{1}{2}\left(\frac{\delta}{2}\right)^2\mu\right) \le \exp\left(-\Omega_\delta(\mu)\right)$$

$$\text{where } \mu = \mathbb{E}[\mathrm{Hypergeometric}(k, |P|-|Y_0|, I)] = \frac{I}{k}(|P|-|Y_0|).$$

If $|Y_0| \ge (1-\delta)(1-\varepsilon)N$, then $|S| = |Y_0| + B$ is large enough with probability 1, so we can assume otherwise, i.e., that $|Y_0| < (1-\delta)(1-\varepsilon)N \le (1-\delta)N$. Thus

$$k \ge |P| - |Y_0| \ge \delta N > \Omega(1), \tag{3}$$

so for N large enough we have $\frac{1}{k} < \frac{\delta}{2}(1-\varepsilon)$ and thus $\frac{I}{k} \ge \frac{\lfloor (1-\varepsilon)k \rfloor}{k} \ge \frac{(1-\varepsilon)k-1}{k} = 1-\varepsilon-\frac{1}{k} \ge \left(1-\frac{\delta}{2}\right)(1-\varepsilon)$. By this and (3), $\mu = \frac{I}{k}(|P|-|Y_0|) \ge \left(1-\frac{\delta}{2}\right)(1-\varepsilon)\delta N \ge \Omega_\delta(N)$, so that $\exp\left(-\Omega_\delta(\mu)\right) = \exp\left(-\Omega_\delta(N)\right)$. Finally, if the good event $B \ge \left(1-\frac{\delta}{2}\right)\mu$ happens, then

$$B \ge \left(1-\frac{\delta}{2}\right)\frac{I}{k}(|P|-|Y_0|) \ge \left(1-\frac{\delta}{2}\right)^2(1-\varepsilon)(|P|-|Y_0|) \ge (1-\delta)\,(1-\varepsilon)(|P|-|Y_0|)$$

and thus $|S| = |Y_0| + B \ge (1-\delta)(1-\varepsilon)N$.

Total fairness violation tail bound. Let us first remark that the algorithm increases some undersaturated color c' at every iteration, so one could think that $\mathrm{fav}(S)$ is small with probability 1. However, undersaturation is measured with respect to P, and we can have $|P \cap V_c| \gg \ell_c$ for some c. Increasing a color beyond ℓ_c elements does not make progress with respect to fairness violation. Nevertheless, we can prove a high-concentration bound in terms of ℓ_c. Recall that for any $\delta > 0$ we want to show that $\mathbb{P}[\mathrm{fav}(S) > (1+\delta)\varepsilon\sum_c \ell_c] \le \exp\left(-\Omega_\delta\left(\sum_c \ell_c\right)\right)$.

The proof will be similar as above, but now the balls, on top of having a color, can be *striped* or not. Namely, for each $c \in [C]$, we create $\max(0, |P \cap V_c| - |Y_0 \cap V_c|)$ balls of color c, of which $\max(0, \ell_c - |Y_0 \cap V_c|)$ many will be *striped*. (We have k balls in total, of which $\mathrm{fav}(Y_0)$ are striped.) Again, at each iteration, if c' is the color that the algorithm increases, we remove a random ball of color c'.

Let X be the number of striped balls removed by the end. We then have

$$\mathrm{fav}(S) \le \mathrm{fav}(Y_0) - X\,. \tag{4}$$

This is because whenever we increase some color c' that has fewer than $\ell_{c'}$ elements, the fairness violation of the solution decreases by 1, but X only accounts for this decrease if we happen to sample a *striped* c'-colored ball. Since there are only as many striped c'-colored balls as there are fairness violations of color c', at the end we will have removed at least as many of the violations as of the balls.

Now we proceed as for cardinality. We have

$$\mathbb{P}\left[X < \left(1-\frac{\delta\varepsilon}{2}\right)\mu\right] \le \exp\left(-\frac{1}{2}\left(\frac{\delta\varepsilon}{2}\right)^2\mu\right) \le \exp\left(-\Omega_\delta(\mu)\right)$$

$$\text{where } \mu = \mathbb{E}[\mathrm{Hypergeometric}(k, \mathrm{fav}(Y_0), I)] = \frac{I}{k}\,\mathrm{fav}(Y_0).$$

If $\mathrm{fav}(Y_0) \le (1+\delta)\varepsilon\sum_c \ell_c$ then we are done, so assume otherwise. Then

$$k \ge \mathrm{fav}(Y_0) > (1+\delta)\varepsilon\sum_c \ell_c > \Omega(1), \tag{5}$$

so for $\sum_c \ell_c$ large enough we have $\frac{1}{k} < \frac{\delta\varepsilon}{2}(1-\varepsilon)$ and thus $\frac{I}{k} \ge 1-\varepsilon-\frac{1}{k} \ge \left(1-\frac{\delta\varepsilon}{2}\right)(1-\varepsilon)$. By this and (5), $\mu = \frac{I}{k}\,\mathrm{fav}(Y_0) \ge \left(1-\frac{\delta\varepsilon}{2}\right)(1-\varepsilon)(1+\delta)\varepsilon\sum_c \ell_c \ge \Omega(\sum_c \ell_c)$, so that $\exp(-\Omega_\delta(\mu)) = \exp(-\Omega_\delta(\sum_c \ell_c))$. Finally, if the good event $X \ge \left(1-\frac{\delta\varepsilon}{2}\right)\mu$ happens, then

$$X \ge \left(1-\frac{\delta\varepsilon}{2}\right)\frac{I}{k}\,\mathrm{fav}(Y_0) \ge \left(1-\frac{\delta\varepsilon}{2}\right)^2(1-\varepsilon)\,\mathrm{fav}(Y_0) \ge (1-\delta\varepsilon-\varepsilon)\,\mathrm{fav}(Y_0)$$

and thus

$$\mathrm{fav}(S) \overset{(4)}{\le} \mathrm{fav}(Y_0) - X \le \mathrm{fav}(Y_0) - (1-\delta\varepsilon-\varepsilon)\,\mathrm{fav}(Y_0) = (1+\delta)\varepsilon\,\mathrm{fav}(Y_0) \le (1+\delta)\varepsilon\sum_c \ell_c\,.$$

Objective value. Intuitively, at every iteration i, we select randomly from among $k-i+1$ disjoint augmenting or alternating sets. Even if the newly added elements do not add any f-value, by submodularity we expect to lose only at most a $1/(k-i+1)$ fraction of the f-value of the current solution. After $I \approx (1-\varepsilon)k$ iterations we then end up with a telescoping product that simplifies to $\frac{\varepsilon k}{k} f(Y_0)$.

We now give a formal proof. We show by induction on i that

$$\mathbb{E}[f(Y_i)] \geq \frac{k-i}{k} f(Y_0)\,. \tag{6}$$

For $i \geq 1$, condition on Y_{i-1}. Then

$$\begin{aligned}
\mathbb{E}[f(Y_i)] &= \mathbb{E}[f(Y_{i-1} \triangle X_i)] \\
&= \frac{1}{k(Y_{i-1})} \sum_{j=1}^{k(Y_{i-1})} f(Y_{i-1} \triangle X_i^j) \\
&\geq \frac{1}{k-i+1} \sum_{j=1}^{k-i+1} f(Y_{i-1} \setminus (Y_{i-1} \cap X_i^j)) \\
&\geq \frac{k-i}{k-i+1} f(Y_{i-1})\,,
\end{aligned}$$

where the first inequality follows by monotonicity and the second inequality is by applying Fact 2.1 to the set family $Y_{i-1} \cap X_i^1, ..., Y_{i-1} \cap X_i^{k-i+1} \subseteq Y_{i-1}$. Now taking expectation over Y_{i-1},

$$\begin{aligned}
\mathbb{E}[f(Y_i)] &= \mathbb{E}[\mathbb{E}[f(Y_i) \mid Y_{i-1}]] \\
&\geq \mathbb{E}\left[\frac{k-i}{k-i+1} f(Y_{i-1})\right] \\
&\geq \frac{k-i}{k-1+1} \cdot \frac{k-(i-1)}{k} f(Y_0)
\end{aligned}$$

where we applied the inductive hypothesis. Having (1) and (6), we can write

$$\mathbb{E}[f(S)] = \mathbb{E}[\mathbb{E}[f(Y_I) \mid I]] \geq \mathbb{E}\left[\frac{k-I}{k} f(Y_0)\right] = \frac{k-(1-\varepsilon)k}{k} f(Y_0) \geq \varepsilon \cdot 0.499 \cdot \mathrm{OPT}_{\mathrm{MatInt}}\,.$$

□

The pseudocode for Lemma 3.3 can be found in Algorithm 1 and the pseudocode for Theorem 3.4 can be found in Algorithm 2.

3.1 Deterministic algorithm

Now we turn to our deterministic result, Theorem 3.6. We begin by showing a lemma that is an analogue of Lemma 3.3.

Lemma 3.5 *For any two matroids $\mathcal{I}_1$, $\mathcal{I}_2$, let $Y, P \in \mathcal{I}_1 \cap \mathcal{I}_2$ be two sets in their intersection, with $|Y| \leq |P|$. Then we may find in polynomial time a collection $X_1, ..., X_{|P|-|Y|}$ of disjoint subsets of $Y \cup P$ such that for each set X_i we have $Y \triangle X_i \in \mathcal{I}_1 \cap \mathcal{I}_2$ and $|Y \triangle X_i| = |Y| + 1$.*

Proof. We proceed similarly as in the proof of Lemma 3.3. We consider the matroid intersection exchange graph for Y and P with respect to $\mathcal{I}_1$ and $\mathcal{I}_2$, defined as in that proof. Define $T = \{p \in P : Y + p \in \mathcal{I}_1\}$ and $S = \{p \in P : Y + p \in \mathcal{I}_2\}$. We have $|T|, |S| \geq |P| - |Y|$. Let T', S' be arbitrary subsets of T, S respectively, both of size exactly $|P| - |Y|$. We invoke Lemma 2.3 on the exchange graph for Y and $P \setminus T'$, obtaining $M_{\leftarrow}$ as a perfect matching between Y and $P \setminus T'$. Similarly, we obtain $M_{\rightarrow}$ as a perfect matching between Y and $P \setminus S'$.

Now we can construct the $|P| - |Y|$ paths as in the proof of Lemma 3.3, starting from sources (set S') and proceeding in the only possible way until we reach a sink (vertex in T'). (Alternatively, we

Algorithm 1 Generate Augmenting Paths

Require: Matroid $\mathcal{I}$ on universe V and fairness constraints with colors $[C]$ and lower bound ℓ_c and upper bound u_c for each color $c \in [C]$.
Require: Sets Y and P, where Y and P are independent sets with respect to matroids $\mathcal{I}$ and $\mathcal{U}$, and $|Y| \leq |P|$.
Ensure: A set of augmenting and alternating paths between Y and P.

Build an initially empty bipartite graph with "left vertices" being elements of Y and "right vertices" being elements of P
for each $(y, p) \in Y \times P$ **do**
    Add directed edge $y \rightarrow p$ if $Y + p - y \in \mathcal{U}$
    Add directed edge $y \leftarrow p$ if $Y + p - y \in \mathcal{I}$
end for
Define P-sinks $T_P := \{p \in P : Y + p \in \mathcal{I}\}$ and let T'_P be an arbitrary subset of T_P of size exactly $|P| - |Y|$.
Find a matching between Y and $P \setminus T'_P$ and drop edges adjacent to $T_P \setminus T'_P$. Call this $M_{\leftarrow}$.
Start with $M_{\rightarrow} = \emptyset$.
for every color $c \in [C]$ **do**
    Add $\min(|Y \cap V_c|, |P \cap V_c|)$ edges from $Y \cap V_c$ to $P \cap V_c$ to the matching $M_{\rightarrow}$.
end for
Define the set S of *sources* as all vertices in P that did not get matched in $M_{\rightarrow}$.
Define the set T_Y of Y*-sinks* as all vertices in Y that did not get matched in $M_{\rightarrow}$.
Let $\mathcal{X}$ be an initially empty collection of paths.
while there is an unused source **do**
    Start at at unused source (in P).
    if at a vertex of P **then**
        if vertex is a sink (in T_P) **then**
            Stop path and add to collection $\mathcal{X}$.
        else
            Follow the outgoing edge of $M_{\leftarrow}$.
        end if
    else
        if vertex is a sink (in T_Y) **then**
            Stop path and add to collection $\mathcal{X}$.
        else
            Follow the outgoing edge of $M_{\rightarrow}$.
        end if
    end if
end while
for each path X in $\mathcal{X}$ **do**
    Shortcut X along right-to-left directed edges (see proof of Lemma 3.3 for details).
end for
return collection of disjoint augmenting / alternating sets $\mathcal{X}$.

Algorithm 2 Random Augmenting Paths

Require: Matroid $\mathcal{I}$ on universe V, a submodular function $f : 2^V \to \mathbb{R}_{\geq 0}$, fairness constraints with colors $[C]$ and lower bound ℓ_c and upper bound u_c for each color $c \in [C]$.
Ensure: Approximately fair and high value set S.

```
Find a fair independent set P via Lemma 2.5 or any matroid intersection algorithm.
Find a high f-value independent set Y_0 with respect to matroids I and U via either local search
    (to get guarantee Theorem 2.7) or Greedy (to get guarantee Theorem 2.6).
Let k(Y) be the number of paths returned by imputing the matchings Y and P into Algorithm 1.
Let I be ⌊(1 − ε)k(Y_0)⌋ with probability ⌈(1 − ε)k(Y_0)⌉ − (1 − ε)k(Y_0) and ⌈(1 − ε)k(Y_0)⌉
    otherwise.
for i = 1 to I do
    Compute feasible augmenting/alternating paths X_i^1, ..., X_i^{k(Y_{i−1})} between Y_{i−1} and P as
        outlined in Lemma 3.3 (see Algorithm 1).
    Choose one X_i ∈ {X_i^1, ..., X_i^{k(Y_{i−1})}} uniformly at random.
    Apply it to obtain Y_i = Y_{i−1} Δ X_i.
end for
return S := Y_I
```

can note that $M_{\leftarrow} \cup M_{\rightarrow}$ is a circulation for demands – i.e., outflow minus inflow – $+1$ on sources and -1 on sinks, and take the paths from its cycle-path decomposition.) All paths start and end in P.

Next, we shortcut the paths. Here, we replace subpaths with chords in both directions, rather than only in one direction as in the proof of Lemma 3.3. Moreover, if there is an internal vertex that is in S, we need to truncate the path so that it begins at that vertex; and similarly, if there is an internal vertex that is in T, we truncate the path so that it ends at that vertex. These operations only shrink the vertex sets of the paths, thus they preserve their vertex-disjointness.

We end once the path X_i is from S to T via $(Y \cup P) \setminus (S \cup T)$ and contains no chords. The same argument as in the proof of Lemma 3.3 then shows that X_i is an augmenting path, i.e., that $Y \triangle X_i \in \mathcal{I}_1 \cap \mathcal{I}_2$. □

Now we can state our deterministic algorithm for any two matroids $\mathcal{I}_1$ and $\mathcal{I}_2$.

Theorem 3.6 *There is a deterministic polynomial-time algorithm for the problem of maximizing a monotone submodular function subject to a matroid intersection constraint, parametrized by* $\varepsilon \in (0, 1)$*, that outputs a set* $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ *such that*

- $|S| > (1 - \varepsilon)N - 1$
- $f(S) \geq 0.499 \cdot \varepsilon \cdot \mathrm{OPT}_{\mathrm{MatInt}}$

where N *is the maximum size of a set in* $\mathcal{I}_1 \cap \mathcal{I}_2$*, and* $\mathrm{OPT}_{\mathrm{MatInt}}$ *is the maximum* f*-value of a set in* $\mathcal{I}_1 \cap \mathcal{I}_2$*.*

Proof. As in the algorithm of Theorem 3.4, we start by computing a maximum-cardinality set P in the matroid intersection, as well as a high-value set Y in the matroid intersection. We have $|P| = N$ and $f(Y_0) \geq 0.499 \cdot \mathrm{OPT}_{\mathrm{MatInt}}$.

We then perform $I := \lfloor (1 - \varepsilon)(|P| - |Y_0|) \rfloor$ iterations. In the i-th iteration, we apply Lemma 3.5 to Y_{i-1} (and P) to obtain a collection $X_i^1, ..., X_i^{|P|-|Y_{i-1}|}$ of sets. We choose the one of them, $X_i \in \{X_i^1, ..., X_i^{|P|-|Y_{i-1}|}\}$, that, when used to obtain a new solution $Y_i = Y_{i-1} \triangle X_i$, maximizes $f(Y_i)$. Finally, we return $S := Y_I$.

All solutions $Y_0, ..., Y_I$ are in the matroid intersection, and they grow in size by 1 per step. Thus we have $|S| = |Y_0| + \lfloor (1-\varepsilon)(|P|-|Y_0|) \rfloor > |Y_0| + (1-\varepsilon)(|P|-|Y_0|) - 1 \geq (1-\varepsilon)|P| - 1 = (1-\varepsilon)N - 1$.

The proof for the objective value guarantee is the same as in Theorem 3.4; at each step, since an average set preserves a $\frac{|P|-|Y_0|-i}{|P|-|Y_0|-i+1}$ fraction of the value, so does the best set. We conclude by saying that since $I \leq (1-\varepsilon)(|P|-|Y_0|)$,

$$f(S) = f(Y_I) \geq \frac{|P|-|Y_0|-(1-\varepsilon)(|P|-|Y_0|)}{|P|-|Y_0|} f(Y_0) \geq \varepsilon \cdot 0.499 \cdot \mathrm{OPT}_{\mathrm{MatInt}} \ .$$

□

4 Experimental evaluation

We evaluate the performance of our algorithms empirically against prior work and natural baselines closely following the experimental setup of prior work [EMNTT20; EFNTT23], on a suite of benchmarks that are standard in the field: graph coverage, clustering, and recommender systems, under different fairness and matroid constraint settings. Our metrics are the submodular objective value $f(S)$ and total fairness violation $\mathrm{fav}(S)$. All of the considered algorithms return sets that are independent and upper-fair, so the measured fairness violations are all with respect to the lower bounds. All three benchmarks use a partition matroid.

We compare the following algorithms:

- OUR(ε) – our algorithm of Theorem 3.4, for a range of settings of $\varepsilon \in \{0.2, 0.5, 0.8\}$. To compute a high-value solution Y, we run the natural greedy algorithm, which obtains a $1/3$-approximation (Theorem 2.6), as the local search algorithm of Theorem 2.7, while polynomial-time, is impractical. The large fair set P is obtained via augmenting paths, ignoring f.
- TWOPASS – the algorithm of [EFNTT23] (Theorem 1.1). Since it was originally developed for the streaming setting, to get a fair comparison we simplify away the parts (namely the first pass) whose purpose was ensuring low memory usage. The first step of the algorithm obtains a fair set via augmenting paths (ignoring f). This is then divided in two, and each half is extended to an independent and upper-fair solution using a matroid intersection subroutine. For this we employ the greedy algorithm (the original implementation of [EFNTT23] used a swapping algorithm to ensure low memory and linear runtime, but it obtains inferior values).
- LBMI (Lower Bound Matroid Intersection) – an algorithm that always returns a fair set, with no theoretical guarantee on the value but with reasonably good value in practice (similar in spirit to GREEDY-FAIR-STREAMING from [EFNTT23]). It starts by building a fair set via augmenting paths, ignoring f, and then extends to a maximal solution using the greedy algorithm.
- UBMI (Upper Bound Matroid Intersection) – an algorithm that ignores lower bound constraints and just solves the matroid intersection problem for $\mathcal{I}$ and $\mathcal{U}$ (similar in spirit to MATROID-INTERSECTION from [EFNTT23]). Also here we use the greedy algorithm.
- RANDOM – an algorithm that randomly shuffles the universe and then adds each element if this keeps the solution independent and upper-fair.

For a fair comparison of the main underlying ideas, we made sure that the compared algorithms, particularly OUR and TWOPASS, use the same subroutines for similar tasks; the implementations could likely benefit from heuristically taking f into account rather than ignoring f when building large fair sets, or from some local-search based postprocessing of the final solution. We do not compare to the algorithm of [ETNV24] (Theorem 1.2) as solving the multilinear extension makes it impractical.

We repeat the randomized algorithms 40 times. All experiments can be run on commodity hardware (CPU only, single-threaded; we do not report runtimes) and take several hours to finish. Our code is in the supplementary material.

We outline the experimental scenarios below. In each experiment we vary a solution size scaling factor r, which roughly corresponds to the rank of the matroid $\mathcal{I}$. We select fairness bounds ℓ_c, u_c to ensure feasibility, requiring that each color group V_c is proportionally represented in the solution set S – either matching its share in the dataset (coverage, movies) or ensuring similar group sizes (clustering). Results are reported in Figs. 1 and 2 and discussed in Section 4.4.

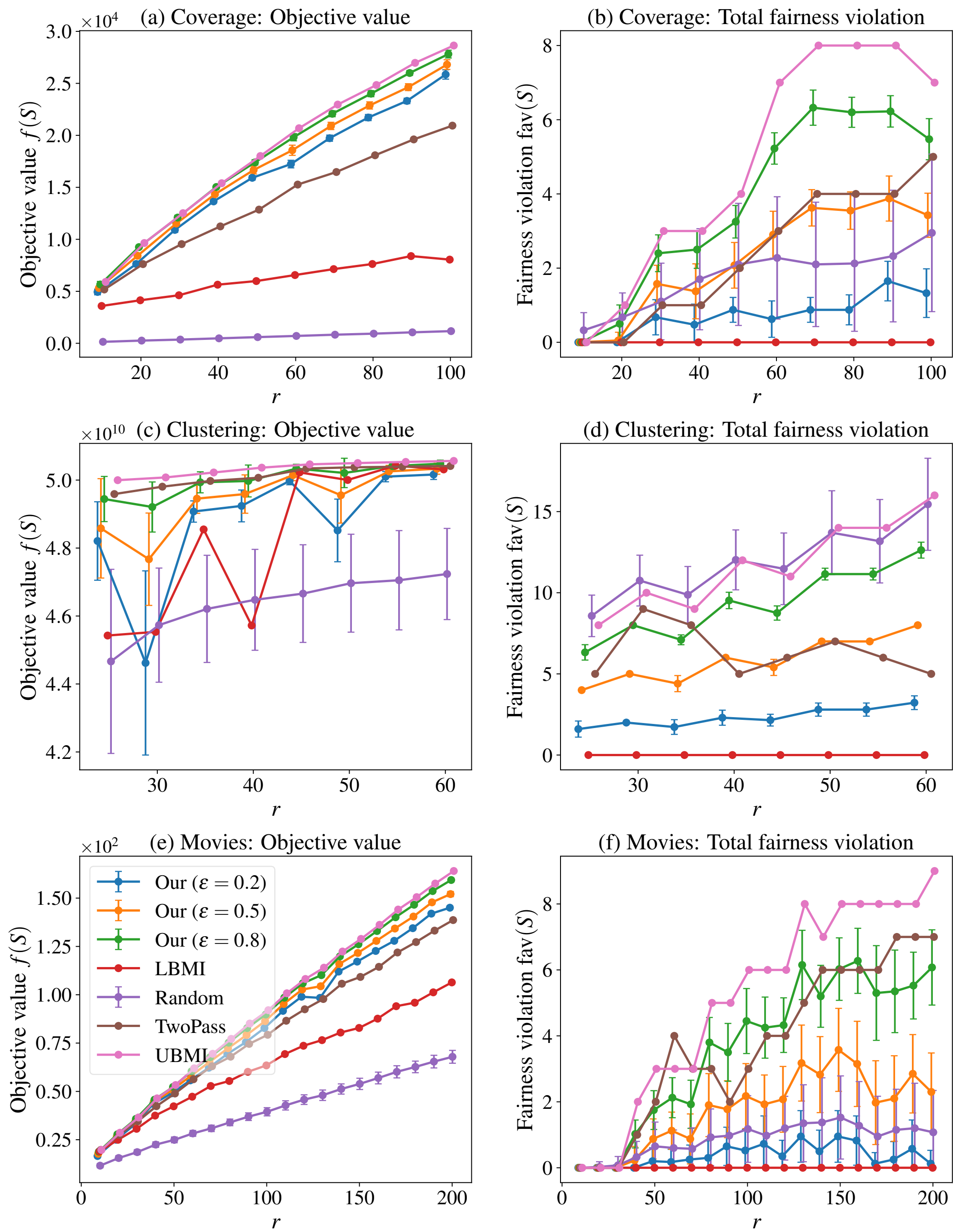

Figure 1: Our experimental results. Each row corresponds to one experiment; the left plot shows the objective value of each algorithm for a range of solution scale factors r, and the right plot shows fairness violations. For randomized algorithms we report averages, with error bars that correspond to sample standard deviation.

Computational complexity. We start with the complexity of the general randomized algorithm of Theorem 3.4. Firstly, the runtime of constructing P (a maximum-cardinality fair and independent set) via augmenting paths is $O(N^{1.5}|V|)$ (by [Sch03], Chapter 41.2 Notes). To construct Y (an upper-fair and independent set of high f-value), we expend $O(N|V|)$ time using the greedy algorithm.

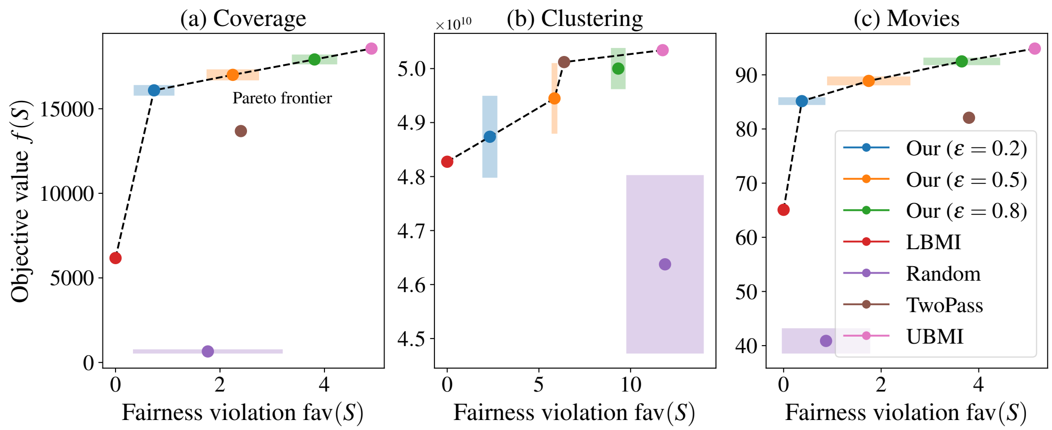

Figure 2: For each experiment and algorithm we take the average objective value and fairness violation over all r-values, and plot this as a single point. For randomized algorithms, the colored rectangles correspond to standard deviations. The dashed line corresponds to the Pareto frontier of the trade-off between objective value and fairness violation.

Next, at each of the I iterations, we must (1) recompute the exchange graph between Y_i and P, (2) find $M_{\leftarrow}$ and $M_{\rightarrow}$ as the subgraph of interest, (3) decompose $M_{\leftarrow} \cup M_{\rightarrow}$ into paths, and (4) shortcut these paths. Step (1) takes $O(N^2)$ time, since we query if a directed edge exists between y and p for all $y \in Y$ and $p \in P$. Finding perfect matchings in step (2) takes at most $O(N^3)$ time (in a practical implementation we could use the Hopcroft-Karp algorithm). Decomposing the resulting subgraph into paths takes at most $O(N)$ time. And lastly, shortcutting the paths again takes at most $O(N^2)$ time. Since I can be $\Theta(N)$, the total runtime is at most $O(N^4)$.

A more efficient implementation is possible if $\mathcal{I}$ is a partition matroid. The intersection of two partition matroids can be naturally interpreted as a bipartite multigraph (the colors, i.e., parts of $\mathcal{U}$ are one side, the parts of $\mathcal{I}$ are the other side, and an element corresponds to an edge between the two parts it belongs to). In this case, we may look at the following exchange graph: direct the edges of Y from left to right, and the edges of P from right to left. This directed graph may be decomposed into paths. These paths are *simultaneously feasible*, and so we do not need to recompute an exchange graph at every step (or shortcut). Since there are $O(N)$ edges, the runtime to decompose this directed graph is $O(N)$. Over the I iterations, we have a total runtime of at most $O(N^2)$.

4.1 Graph coverage

We use the Pokec social network [LK14]. Given a digraph $G = (V, E)$ of users and their friendships, we select a subset $S \subseteq V$ to maximize coverage, defined by $f(S) = \left|\bigcup_{v \in S} N(v)\right|$, where $N(v)$ is the neighborhood of v. User profiles include age, gender, height, and weight. We impose a partition matroid on body mass index (BMI). Profiles missing height or weight or with implausible data are removed, yielding a graph with 582,289 nodes and 5,834,695 edges. Users are partitioned into four BMI categories (underweight, normal, overweight, obese), with upper bounds $\lceil \frac{|V_i|}{|V|} r \rceil$ for each group V_i. We also enforce fairness by age, with 7 groups: $[1, 10]$, $[11, 17]$, $[18, 25]$, $[26, 35]$, $[36, 45]$, $[46+]$, no age. We set $\ell_c = \lfloor 0.9 \frac{|V_c|}{|V|} r \rfloor$ and $u_c = \lceil 1.5 \frac{|V_c|}{|V|} r \rceil$. We use r from 10 to 200.

4.2 Exemplar-based clustering

We use a dataset of 4521 phone calls from a Portuguese bank marketing campaign [MCR14]. The goal is to select a representative subset $S \subseteq V$ for service quality assessment. Each record $e \in V$ is represented as $x_e \in \mathbb{R}^7$ using 7 numeric features, including age and account balance. We impose a partition matroid on account balance, with 5 groups: $(-\infty, 0)$, $[0, 2000)$, $[2000, 4000)$, $[4000, 6000)$, $[6000, \infty)$. Each group V_i has upper bound $r/5$. Fairness is enforced by age, with 6 groups: $[0, 29]$, $[30, 39]$, $[40, 49]$, $[50, 59]$, $[60, 69]$, $[70+]$, and bounds

$\ell_c = 0.1r + 2$, $u_c = 0.4r$ for each c. We maximize the monotone submodular function [GK10]: $f(S) = \sum_{e' \in V} \left(d(e', 0) - \min_{e \in S \cup \{0\}} d(e', e)\right)$ where $d(e', e) = \|x_{e'} - x_e\|_2^2$ and x_0 is the origin. We use r from 30 to 60.

4.3 Recommender system

We simulate a movie recommendation system using the Movielens 1M dataset [HK16], with about one million ratings for 3900 movies by 6040 users. As in prior work [MBNTC17; NTMZMS18; EMNTT20; EFNTT23], we compute a low-rank completion of the user-movie matrix [TCSBHTBA01], yielding $w_u \in \mathbb{R}^{20}$ for each user u and $v_m \in \mathbb{R}^{20}$ for each movie m. The product $w_u^\top v_m$ estimates user u's rating for movie m. For user u, the monotone submodular utility for a set S of movies is $f(S) = \alpha \cdot \sum_{m' \in M} \max\left(\max_{m \in S}\left(v_m^\top v_{m'}\right), 0\right) + (1-\alpha) \cdot \sum_{m \in S} w_u^\top v_m$, with parameter $\alpha = 0.85$ balancing coverage and personalized user score. We enforce proportional representation of movies by release date using a partition matroid with 9 decade groups (1911–2000), with upper bounds $\lceil 1.2 \frac{|V_d|}{|V|} r \rceil$ for each decade V_d. Movies are also partitioned into 18 genres c (colors), with fairness bounds $\ell_c = \lfloor 0.8 \frac{|V_c|}{|V|} r \rfloor$ and $u_c = \lceil 1.4 \frac{|V_c|}{|V|} r \rceil$. We use r from 10 to 200.

4.4 Results and discussion

Our results are depicted in Figs. 1 and 2. Similarly as prior work, we observe that enforcing fairness does come at some cost in the utility value, and that the utility values of the algorithms are much better in practice than the theoretical bounds guarantee.

In all three experiments, our algorithms produce solutions whose value is relatively competitive with UBMI, which completely ignores the lower bound constraints and accordingly has the highest fairness violations. In two of the three scenarios (coverage and movies), all OUR algorithms produce a higher f-value than all the other baselines (RANDOM, LBMI, and TWOPASS); in particular, TWOPASS is dominated by both OUR(0.2) and OUR(0.5) with respect to both metrics. For clustering the situation is somewhat unclear, but TWOPASS generally does better. In terms of violation of the lower bound fairness constraints, our different settings of ε, as expected, provide a smooth tradeoff. The baseline that guarantees no fairness violations, LBMI, does relatively poorly in terms of f-value.

This tunability of ε is a key strength of our approach, allowing users to select an operating point that best matches their specific requirements for the balance between utility and fairness.

5 Conclusion, limitations, broader impact, and future work

In this work we gave an improved algorithm for FMMSM which, for any $\varepsilon > 0$, returns an approximate solution that satisfies an expected $(1 - \varepsilon)$ fraction of each fairness lower bound while satisfying the matroid constraint and the fairness upper bound constraints exactly; the returned solution is also large in size and enjoys high-concentration guarantees.

Recent studies have shown that automated algorithms used in decision-making can introduce bias and discrimination. We make progress towards mitigating such effects in problems that can be formulated as submodular maximization under a matroid constraint, which are relevant to a range of applications such as forming representative committees or curating content for news feeds. We show the strong performance of our algorithm empirically on several real-world tasks. As in prior work, we observe that there is indeed a balance between fairness and utility value; however, this "price of fairness" should not be interpreted as fairness leading to inferior outcomes, but rather as a trade-off between two valuable metrics. The parametric nature of our algorithm (the tunable ε parameter) provides a new tool to help in navigating this balance.

Our work leaves open the exciting question of the approximability of FMMSM (without violations of fairness constraints) and MSPM. Is there a constant-factor approximation algorithm for MSPM? Or is there a superconstant hardness of approximation for FMMSM? (As remarked in [ETNV24], the latter result would give a negative answer to a fundamental question posed by Vondrák [Von13].) We also do not consider non-monotone objective functions or the streaming setting in this work.

Finally, it is important to note that the fairness notion we employ, though standard and general, does not capture some notions of fairness considered in the literature (see e.g. [CR18; TWRTZ19]). No universal definition of fairness exists; the choice of which definition to apply is application-dependent and an active area of research.

References

[Bid06] Dan Biddle. *Adverse impact and test validation: A practitioner's guide to valid and defensible employment testing*. Gower Publishing, Ltd., 2006.

[BLS17] Markus Brill, Jean-François Laslier, and Piotr Skowron. "Multiwinner Approval Rules as Apportionment Methods". In: *AAAI*. AAAI Press, 2017, pp. 414–420.

[CCPV11] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. "Maximizing a Monotone Submodular Function Subject to a Matroid Constraint". In: *SIAM Journal on Computing* 40.6 (2011), pp. 1740–1766. DOI: 10.1137/080733991.

[CCRL13] Joanne McGrath Cohoon, James P. Cohoon, Seth Reichelson, and Selwyn Lawrence. "Effective Recruiting for Diversity". In: *FIE*. IEEE Computer Society, 2013, pp. 1123–1124.

[CHV18] L. Elisa Celis, Lingxiao Huang, and Nisheeth K. Vishnoi. "Multiwinner Voting with Fairness Constraints". In: *IJCAI*. ijcai.org, 2018, pp. 144–151.

[CKLV17] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. "Fair Clustering Through Fairlets". In: *NIPS*. 2017, pp. 5029–5037.

[CKLV19] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. "Matroids, Matchings, and Fairness". In: *AISTATS*. Vol. 89. Proceedings of Machine Learning Research. PMLR, 2019, pp. 2212–2220.

[CKSDKV18] L. Elisa Celis, Vijay Keswani, Damian Straszak, Amit Deshpande, Tarun Kathuria, and Nisheeth K. Vishnoi. "Fair and Diverse DPP-Based Data Summarization". In: *ICML*. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 715–724.

[CR18] Alexandra Chouldechova and Aaron Roth. "The frontiers of fairness in machine learning". In: *arXiv preprint arXiv:1810.08810* (2018).

[CSV18] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. "Ranking with Fairness Constraints". In: *ICALP*. Vol. 107. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, 28:1–28:15.

[CVZ10] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. "Dependent Randomized Rounding via Exchange Properties of Combinatorial Structures". In: *FOCS*. IEEE Computer Society, 2010, pp. 575–584.

[DHPRZ12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. "Fairness through awareness". In: *Proceedings of the 3rd innovations in theoretical computer science conference*. 2012, pp. 214–226.

[DK11] Abhimanyu Das and David Kempe. "Submodular meets Spectral: Greedy Algorithms for Subset Selection, Sparse Approximation and Dictionary Selection". In: *ICML*. Omnipress, 2011, pp. 1057–1064.

[EFNTT23] Marwa El Halabi, Federico Fusco, Ashkan Norouzi-Fard, Jakab Tardos, and Jakub Tarnawski. "Fairness in Streaming Submodular Maximization over a Matroid Constraint". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 9150–9171.

[EG11] Khalid El-Arini and Carlos Guestrin. "Beyond keyword search: discovering relevant scientific literature". In: *KDD*. ACM, 2011, pp. 439–447.

[EMNTT20] Marwa El Halabi, Slobodan Mitrovic, Ashkan Norouzi-Fard, Jakab Tardos, and Jakub Tarnawski. "Fairness in Streaming Submodular Maximization: Algorithms and Hardness". In: *NeurIPS*. 2020.

[ETNV24] Marwa El Halabi, Jakub Tarnawski, Ashkan Norouzi-Fard, and Thuy-Duong Vuong. "Fairness in Submodular Maximization over a Matroid Constraint". In: *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*. Ed. by Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li. Vol. 238. Proceedings of Machine Learning Research. PMLR, 2024, pp. 1027–1035. URL: https://proceedings.mlr.press/v238/el-halabi24a.html.

[Eur22] European Union FRA. *Bias in Algorithms —- Artificial Intelligence and Discrimination*. Luxembourg: European Union Agency for Fundamental Rights. Publications Office of the European Union, 2022.

[Fei98] Uriel Feige. "A Threshold of ln n for Approximating Set Cover". In: *J. ACM* 45.4 (1998), pp. 634–652.

[FLNSZ22] Moran Feldman, Paul Liu, Ashkan Norouzi-Fard, Ola Svensson, and Rico Zenklusen. "Streaming Submodular Maximization Under Matroid Constraints". In: *ICALP*. Vol. 229. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 59:1–59:20.

[GHIM09] Michel X. Goemans, Nicholas J. A. Harvey, Satoru Iwata, and Vahab S. Mirrokni. "Approximating submodular functions everywhere". In: *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, January 4-6, 2009*. Ed. by Claire Mathieu. SIAM, 2009, pp. 535–544. DOI: 10.1137/1.9781611973068.59. URL: https://doi.org/10.1137/1.9781611973068.59.

[GJS21] Paritosh Garg, Linus Jordan, and Ola Svensson. "Semi-streaming Algorithms for Submodular Matroid Intersection". In: *IPCO*. Vol. 12707. Lecture Notes in Computer Science. Springer, 2021, pp. 208–222.

[GK10] Ryan Gomes and Andreas Krause. "Budgeted Nonparametric Learning from Data Streams". In: *ICML*. Omnipress, 2010, pp. 391–398.

[GK11] Daniel Golovin and Andreas Krause. "Adaptive Submodularity: Theory and Applications in Active Learning and Stochastic Optimization". In: *J. Artif. Intell. Res.* 42 (2011), pp. 427–486.

[HK16] F Maxwell Harper and Joseph A Konstan. "The MovieLens datasets: History and context". In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5.4 (2016), p. 19.

[HMV23] Sedjro Salomon Hotegni, Sepideh Mahabadi, and Ali Vakilian. "Approximation Algorithms for Fair Range Clustering". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, 23–29 Jul 2023, pp. 13270–13284. URL: https://proceedings.mlr.press/v202/hotegni23a.html.

[Hoe63] Wassily Hoeffding. "Probability Inequalities for Sums of Bounded Random Variables". In: *Journal of the American Statistical Association* 58.301 (1963), pp. 13–30. DOI: 10.1080/01621459.1963.10500830. URL: https://doi.org/10.1080/01621459.1963.10500830.

[JNN20] Matthew Jones, Huy Nguyen, and Thy Nguyen. "Fair k-Centers via Maximum Matching". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 4940–4949. URL: https://proceedings.mlr.press/v119/jones20a.html.

[KAM19] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. "Fair k-Center Clustering for Data Summarization". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 3448–3457. URL: http://proceedings.mlr.press/v97/kleindessner19a.html.

[KMGG08] Andreas Krause, H Brendan McMahan, Carlos Guestrin, and Anupam Gupta. "Robust submodular observation selection". In: *Journal of Machine Learning Research* 9.Dec (2008), pp. 2761–2801.

[LB11] Hui Lin and Jeff A. Bilmes. "A Class of Submodular Functions for Document Summarization". In: *ACL*. The Association for Computer Linguistics, 2011, pp. 510–520.

[LK14] Jure Leskovec and Andrej Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection*. http://snap.stanford.edu/data. June 2014.

[LSV10] Jon Lee, Maxim Sviridenko, and Jan Vondrák. “Submodular maximization over multiple matroids via generalized exchange properties”. In: *Mathematics of Operations Research* 35.4 (2010), pp. 795–806.

[MBNTC17] Slobodan Mitrović, Ilija Bogunović, Ashkan Norouzi-Fard, Jakub Tarnawski, and Volkan Cevher. “Streaming Robust Submodular Maximization: A Partitioned Thresholding Approach”. In: *Advances in Neural Information Processing Systems*. 2017.

[MCR14] Sérgio Moro, Paulo Cortez, and Paulo Rita. “A data-driven approach to predict the success of bank telemarketing”. In: *Decis. Support Syst.* 62 (2014), pp. 22–31.

[MMD16] Cecilia Munoz, Smith Megan, and DJ Patil. *Big data: A report on algorithmic systems, opportunity, and civil rights*. Washington, DC: Executive Office of the President. The White House, 2016.

[Mon95] Burt L Monroe. “Fully proportional representation”. In: *American Political Science Review* 89.4 (1995), pp. 925–940.

[NTMZMS18] Ashkan Norouzi-Fard, Jakub Tarnawski, Slobodan Mitrović, Amir Zandieh, Aida Mousavifar, and Ola Svensson. “Beyond $1/2$-approximation for submodular maximization on massive data streams”. In: *ICML* (2018).

[Sch03] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Vol. B. Jan. 2003.

[TCSBHTBA01] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. “Missing value estimation methods for DNA microarrays”. In: *Bioinformatics* 17.6 (2001), pp. 520–525.

[TWRTZ19] Alan Tsang, Bryan Wilder, Eric Rice, Milind Tambe, and Yair Zick. “Group-Fairness in Influence Maximization”. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 5997–6005. DOI: 10.24963/ijcai.2019/831. URL: https://doi.org/10.24963/ijcai.2019/831.

[TY23] Shaojie Tang and Jing Yuan. “Beyond submodularity: a unified framework of randomized set selection with group fairness constraints”. In: *Journal of Combinatorial Optimization* 45.4 (2023), p. 102.

[Udw18] Rajan Udwani. “Multi-objective maximization of monotone submodular functions with cardinality constraint”. In: *Advances in Neural Information Processing Systems* 31 (2018).

[Von13] Jan Vondrák. *Submodular Functions and Their Applications*. Plenary talk at the ACM-SIAM Symposium on Discrete Algorithms (SODA) 2013. 2013. URL: https://theory.stanford.edu/~jvondrak/data/SODA-plenary-talk.pdf.

[WFM21] Yanhao Wang, Francesco Fabbri, and Michael Mathioudakis. “Fair and Representative Subset Selection from Data Streams”. In: *Proceedings of the Web Conference 2021*. WWW ’21. Ljubljana, Slovenia: Association for Computing Machinery, 2021, pp. 1340–1350. ISBN: 9781450383127. DOI: 10.1145/3442381.3449799. URL: https://doi.org/10.1145/3442381.3449799.

[Whi22] White House OSTP. *Blueprint for an AI Bill of Rights*. Washington, DC: White House Office of Science and Technology Policy, 2022.

[WLBW24] Yanhao Wang, Yuchen Li, Francesco Bonchi, and Ying Wang. “Balancing Utility and Fairness in Submodular Maximization”. In: *Proceedings of the 27th International Conference on Extending Database Technology, EDBT*. 2024, pp. 1–14.

[YT23] Jing Yuan and Shaojie Tang. “Group fairness in non-monotone submodular maximization”. In: *Journal of Combinatorial Optimization* 45.3 (2023), p. 88.

[ZVGG17] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. “Fairness Constraints: Mechanisms for Fair Classification”. In: *AISTATS*. Vol. 54. Proceedings of Machine Learning Research. PMLR, 2017, pp. 962–970.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, particularly in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, we provide the full source code in the supplementary material, zipped together with all relevant datasets for ease of execution. We will also open-source release the code together with the camera-ready version.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Yes, we provide the full source code in the supplementary material, zipped together with all relevant datasets for ease of execution. We will also open-source release the code together with the camera-ready version.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Yes, we report sample standard deviation error bars in the plots (and in the result files in the supplementary material). These correspond to the randomness used by the algorithm.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The experiments can be run on commodity hardware (CPU only, single-threaded) and take several hours to finish. We mention this in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, in Section 5.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We release no new assets (except code, which is discussed elsewhere).

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.