A Generic Framework for Fair Consensus Clustering in Streams

Diptarka Chakraborty
National University of Singapore
Singapore, Singapore
diptarka@nus.edu.sg

Kushagra Chatterjee
National University of Singapore
Singapore, Singapore
kushagra.chatterjee@u.nus.edu

Debarati Das
Pennsylvania State University
Pennsylvania, USA
debaratix710@gmail.com

Tien-Long Nguyen
Pennsylvania State University
Pennsylvania, USA
tfn5179@psu.edu

ABSTRACT

Consensus clustering seeks to combine multiple clusterings of the same dataset, potentially derived by considering various non-sensitive attributes by different agents in a multi-agent environment, into a single partitioning that best reflects the overall structure of the underlying dataset. Recent work by Chakraborty et al. [COLT'25] introduced a *fair* variant under proportionate fairness and obtained a constant-factor approximation by naively selecting the best *closest fair* input clustering; however, their offline approach requires storing all input clusterings, which is prohibitively expensive for most large-scale applications.

In this paper, we initiate the study of fair consensus clustering in the streaming model, where input clusterings arrive sequentially and memory is limited. We design the first constant-factor algorithm that processes the stream while storing only a logarithmic number of inputs. En route, we introduce a new generic algorithmic framework that integrates *closest fair clustering* with *cluster fitting*, yielding improved approximation guarantees not only in the streaming setting but also when revisited offline. Furthermore, the framework is fairness-agnostic: it applies to any fairness definition for which an approximately close fair clustering can be computed efficiently. Finally, we extend our methods to the more general *k-median consensus clustering* problem.

KEYWORDS

Consensus Clustering, Fairness, Approximation Algorithms, Streaming Algorithms

ACM Reference Format:
Diptarka Chakraborty, Kushagra Chatterjee, Debarati Das, and Tien-Long Nguyen. 2026. A Generic Framework for Fair Consensus Clustering in Streams. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/TFHZ1924

This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/TFHZ1924

1 INTRODUCTION

Clustering – the task of partitioning data points into groups based on their mutual similarity or dissimilarity – lies at the core of unsupervised learning and is pervasive across machine learning and data analysis applications. In many settings, each data point represents an individual endowed with protected attributes, which can be encoded by assigning a specific color to each point. While traditional clustering algorithms typically succeed in optimizing application-specific objectives, they often fall short in ensuring fairness in their outputs, risking the introduction or perpetuation of biases against marginalized groups delineated by sensitive attributes such as gender, ethnicity, and race [5, 25]. These biases often stem not from the algorithms themselves, but rather from the historical marginalization inherent in the data used for training. Consequently, mitigating such biases to achieve fair outcomes has emerged as a central topic to guarantee demographic parity [17] and equal opportunity [22], not only in clustering but also in any machine learning driven process.

A seminal step toward ensuring fairness in several unsupervised learning tasks is the notion of fair clustering introduced by [16], where points are colored and each partition/cluster must maintain a certain "balance" among various colored points. Intuitively, this balance promotes fair representation and helps address disparate impact. Motivated by this notion of fairness, recent work [12] has initiated the study of the fair variant of the *consensus clustering* problem. Consensus clustering (also referred to as *median partition*), a fundamental problem in machine learning and data analysis, seeks to aggregate a set of input clusterings – often derived from distinct, non-sensitive attributes – into a single clustering that best captures the collective structure of the data by minimizing the *median objective*, i.e., the sum of distances to the inputs. Here, the distance between two clusterings is measured by counting the pairs of points co-clustered in one but not the other. This problem is particularly relevant in applications such as gene integration in bioinformatics [19, 20], data mining [33], and community detection [27]. However, the problem is NP-hard [26, 32] and even APX-hard – precluding any $(1+\varepsilon)$-approximation for arbitrary $\varepsilon > 0$ – even with only three input clusterings [6]. The best-known approximation factor to date is 11/7 [1], although several heuristics have been proposed that are known to work well in practice (e.g., [21, 28, 35]).

The standard consensus clustering procedures do not explicitly ensure fairness, whether it is *proportional fairness* or *statistical fairness*, in the aggregated clustering. For instance, consider community detection in social networks: valid partitions might arise from

non-sensitive attributes, such as age groups or food preferences; yet, the final consensus partition should be fair, with each community reflecting an appropriate representation across protected attributes, like race or gender. The known standard consensus algorithms, whether being approximation algorithms or heuristics, fail to achieve such fairness. Addressing this gap, recent work [12] has initiated the incorporation of fairness constraints into consensus clustering and provided constant-factor approximation algorithms for two colors, which was later extended to multiple colors in [11].

However, modern large-scale applications, ranging from federated learning systems to continuous data collection on online platforms, rarely permit full access to the entire input at once. This motivates the study of space-efficient algorithms in the streaming model, where the input arrives sequentially and the algorithm processes points on the fly while retaining only a small subset or a compact *sketch* of the data. At the end of the stream, all downstream computation is performed using this sketch. This naturally raises the question: can we maintain fairness efficiently in the streaming setting, achieving sublinear, ideally logarithmic, space per reported output, while preserving strong approximation guarantees? Although multiple passes over the stream can sometimes be permitted, single-pass algorithms are preferred because additional passes are often cost-prohibitive in real-world applications (see [2, 29]).

Further, in the insertion-only streaming model, which is the most commonly studied setting for clustering (e.g., [7, 15, 30, 31]), points from an underlying metric space arrive one by one, and deletions are disallowed. In this paper, we initiate the study of fair consensus clustering in the streaming model. While [12] provides constant-factor algorithms for fair consensus, their approach effectively computes a closest fair clustering for each input clustering, evaluates the median objective for each of these candidates, and returns the best – necessitating explicit storage of all input clusterings and their closest fair clusterings and thus precluding sublinear-space streaming computation. It raises a natural question: can we obtain comparable approximation guarantees in a streaming model using only sublinear space? We answer this in the affirmative: with a single pass over the input, we obtain constant-factor approximations for fair consensus clustering not only for the (1-)median objective, as in [12], but also for the more general *k-median* objective. Informally, k-median (fair) consensus clustering seeks k (fair) representative partitions, generalizing the standard (fair) consensus clustering framework. As a natural extension, it has been applied to model selection and clustering ensemble summarization, image segmentation ensembles, bioinformatics, and community detection [4, 10, 27, 34]. We elaborate more on related works in the full version.[1]

1.1 Our Contribution

Fair Consensus Clustering. In this problem, we are given a collection of input clusterings $\mathcal{I} = \{C_1, C_2, \ldots, C_m\}$ defined on a common ground set V (where $|V| = n$), arriving sequentially in a stream, together with a fairness constraint $\mathcal{P}$. The objective is to compute a *fair consensus clustering* C that minimizes the total distance between C and each input clustering in $\mathcal{I}$, while satisfying the fairness constraint $\mathcal{P}$ and using small space. Formally, we establish the following result:

- Suppose there exists a γ-approximation algorithm for the closest fair clustering problem under constraint $\mathcal{P}$, for some parameter $\gamma > 1$. Then there exists a randomized polynomial time streaming algorithm that, given the input clusterings $\mathcal{I}$, computes a fair clustering $\mathcal{F}$ in $O(n \log(mn))$ space, achieving a $(\gamma + 1.995)$-approximation with probability at least $1 - 1/poly(m)$.

It is worth noting that since an intended output of the fair consensus clustering problem is a clustering on V, it is of size $\Omega(n)$. Furthermore, it follows from a simple counting argument that the number of fair clusterings, even in the equi-proportionate case, is $n^{\Omega(n)}$, and thus a standard information-theoretic (encoding-decoding based) argument shows that the space requirement must be $\Omega(n \log n)$ bits. So the space usage of our streaming algorithm is nearly optimal.

***k*-Median Fair Consensus Clustering.** Next, we consider a more general setting where, instead of finding a single representative clustering, the goal is to identify a collection of k representative clusterings $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$ that together minimize the total distance between each input clustering and its nearest representative, while ensuring that each representative $\mathcal{S}_i$ satisfies the fairness constraint $\mathcal{P}$. Formally, we show the following:

- Suppose there exists a γ-approximation algorithm for the closest fair clustering problem under constraint $\mathcal{P}$, for some $\gamma > 1$. Then there exists a randomized polynomial time streaming algorithm that, given the input clusterings $\mathcal{I}$, computes a k-median fair consensus clustering $\mathcal{S}$ in $O(k^2 n \text{polylog}(mn))$ space, achieving a $(1.0151\gamma + 1.99951)$-approximation with probability at least $1 - 1/poly(m)$.

To the best of our knowledge, this is the first work to study *fair consensus clustering* in the streaming model, and also the first to address its more general variant, *k-median fair consensus clustering*. In fact, for the offline version of the *k-median fair consensus clustering* problem, we obtain an improved approximation guarantee of $(\gamma + 1.92)$.

Building on the recent results of [12], which study the closest fair clustering problem under two-color fairness constraints, as a corollary, we obtain the following approximation guarantees for the k-median fair consensus clustering problem in the streaming model: a 3.01461-approximation when the population is equi-proportionate; a 19.25621-approximation when the population ratio (of two colored groups) is $p : 1$ for any positive integer $p > 1$; and a 35.49781-approximation for the general case where the population ratio is $p : q$ where $p, q > 1$ are positive integers.

We further remark that our proposed algorithms are *robust to the specific definition of the fairness constraint*. This is because the fair component is used in a black-box manner: as long as there exists a good exact or approximation algorithm for the underlying closest fair clustering problem under a given fairness constraint, our streaming framework remains applicable. Consequently, the approach generalizes naturally to various settings, such as those involving an arbitrary number of colors (even with potentially overlapping colored groups), heterogeneous population ratios, or alternative notions like statistical fairness. In fact, our framework works for any constraint, not necessarily for fairness constraints, provided we have an algorithm to find an (approximately) closest constrained clustering (a nearest neighbor in the constrained set).

[1] The full version of the paper is available at https://arxiv.org/abs/2602.11500

Framework Overview. To design our algorithms, we introduce a *new two-phase framework* for fair consensus clustering, adapting the frameworks of [13, 14]. In the first phase, we construct a candidate set of consensus clusterings by combining two ideas: *cluster fitting* and *closest fair clustering*, and then select the one minimizing the overall objective. The primary challenge is adapting this process to the streaming setting, where it is not possible to store all the input clusterings simultaneously. To overcome this, we use the power of uniform sampling that retains only a logarithmic number of input clusterings while still guaranteeing a good approximation to the true consensus. Moreover, we show that using an additional uniformly sampled subset of the input, again of only logarithmic size, is sufficient to reliably identify the best candidate consensus clusterings. This results in a sublinear-space algorithm that stores only a logarithmic number of inputs while maintaining strong approximation guarantees.

Further to ensure fairness, our approach treats any available closest fair clustering algorithm as a *black box*, integrating it seamlessly into our consensus framework. This modularity ensures that our overall approximation inherits the fairness-approximation factor of the underlying subroutine. Consequently, whenever the closest fair clustering subroutine admits an exact or improved approximation (e.g., in the equi-proportionate two-color case due to [12]), our overall algorithm immediately yields a more efficient and tighter approximation.

We further extend this framework to the *k-median fair consensus clustering* problem, where the goal is to output k fair representative clusterings rather than one. Here, we develop a more carefully structured sampling method to ensure sufficient representation across all k clusters, along with fairness enforcement for each representative. Furthermore, to identify the best k-representative clusterings, we employ a more sophisticated technique, *monotone faraway sampling* [9] which, once again, requires only a logarithmic number of input samples. The combination of these sampling and fairness components yields the first streaming algorithms providing constant-factor approximations for both 1-median and k-median fair consensus clustering.

2 PRELIMINARIES

In this section, we define key terms and concepts that are essential for understanding the proofs and algorithms presented.

Definition 2.1 (*Fair Clustering*)**.** Given a set of points V and a fairness constraint $\mathcal{P}$, we call a clustering $\mathcal{F}$ of V a *Fair Clustering* if every cluster $F \in \mathcal{F}$ satisfies the constraint $\mathcal{P}$.

For example, Chierichetti et al. [16] considered the case where the set V is partitioned into two disjoint groups (or colors), red and blue, denoted by R and B, respectively. In this setting, the fairness constraint $\mathcal{P}$ requires that each cluster $F \in \mathcal{F}$ preserves the global color ratio, i.e.,

$$\frac{|F \cap R|}{|F \cap B|} = \frac{|R|}{|B|}.$$

Equivalently, every cluster must contain the same proportion of red and blue points as in the entire dataset.

For two clustering C and C' of V we define $\mathrm{dist}(C, C')$ as the distance between two clustering C and C'. The distance is measured by the number of pairs (u,v) that are together in C but separated by C' and the number of pairs (u,v) that are separated by C but together in C'. More specifically,

$$\mathrm{dist}(C,C') = \Big|\big\{\{u,v\} \mid u,v \in V, [u \sim_C v \wedge u \nsim_{C'} v] \vee [u \nsim_C v \wedge u \sim_{C'} v]\big\}\Big|$$

where $u \sim_C v$ denotes whether both u and v belong to the same cluster in C or not.

Definition 2.2 (Closest *Fair Clustering*)**.** Given an arbitrary clustering $\mathcal{D}$, a clustering $\mathcal{N}^*$ is called a closest *Fair Clustering* to $\mathcal{D}$ if for all *Fair Clustering* $\mathcal{N}$ we have $\mathrm{dist}(\mathcal{D},\mathcal{N}) \geq \mathrm{dist}(\mathcal{D},\mathcal{N}^*)$.

We denote a closest *Fair Clustering* by $\mathcal{N}^*$.

γ-close Fair Clustering. We call a *Fair Clustering* $\mathcal{N}$ a γ-close *Fair Clustering* to a clustering $\mathcal{D}$ if

$$\mathrm{dist}(\mathcal{D},\mathcal{N}) \leq \gamma \mathrm{dist}(\mathcal{D},\mathcal{N}^*).$$

Definition 2.3 (**1**-median Consensus Clustering problem)**.** Given a set of clusterings $\mathcal{I} = \{C_1, C_2, \ldots, C_m\}$, the goal is find a clustering C such that it minimizes the total distance between each input clustering and C. Formally, we seek to minimize the following objective

$$\mathrm{Obj}(\mathcal{I}, C) = \sum_{C_i \in \mathcal{I}} \mathrm{dist}(C_i, C)$$

The above objective is called the *median* objective. In addition to the objective, when we want the clustering to be fair, it is called the 1- *median fair consensus clustering problem.*

Definition 2.4 (***k***-median Consensus Clustering problem)**.** Given a collection of input clusterings $\mathcal{I} = \{C_1, \ldots, C_m\}$, the objective is to construct a set of k representative clusterings

$$\mathrm{Z} = \{\mathcal{Z}_1, \ldots, \mathcal{Z}_k\}$$

that minimizes the total distance between each input clustering and its nearest representative. Formally, we seek to minimize

$$\mathrm{Obj}(\mathcal{I}, \mathrm{Z}) = \sum_{C_i \in \mathcal{I}} \min_{\mathcal{Z}_j \in \mathrm{Z}} \mathrm{dist}(C_i, \mathcal{Z}_j).$$

In addition to the objective, when we want the clustering to be fair, we call it the *k-median fair consensus clustering problem.*

(β, k)-approximate Fair Consensus Clustering. : A set of representatives $\mathrm{Z} = \{\mathcal{Z}_1, \ldots, \mathcal{Z}_k\}$ such that each $\mathcal{Z}_i$ is fair is called a (β,k)-approximate Fair Consensus Clustering if the following is true

$$\mathrm{Obj}(\mathcal{I}, \mathrm{Z}) \leq \beta \mathrm{Obj}(\mathcal{I}, \mathrm{Z}^*)$$

where $\mathrm{Z}^* = \{\mathcal{Z}_1^*, \ldots, \mathcal{Z}_k^*\}$ be the set of optimal representatives.

Let us now define our streaming models.

Streaming Model. In this paper, we consider an *insertion-only* streaming model. The input is a sequence of triples (p,j,b), where:

- $p = (u,v)$ is an unordered pair of vertices in V,
- $j \in [m]$ identifies a clustering $C_j \in \mathcal{I}$, and
- $b \in \{0,1\}$ indicates whether u and v lie in the same cluster in the clustering C_j ($b = 0$) or in different clusters ($b = 1$).

Here, a triple $((u, v), j, b)$ refers an information about the clustering C_j. For each clustering C_j, we have n^2 such tuples, hence the length of the stream is mn^2.

In the insertion-only streaming framework commonly used in clustering, elements or points of a metric space are revealed sequentially. In consensus clustering, the metric space is the set of clusterings, so each clustering is treated as a single point in that space. Accordingly, following standard practice, we assume the stream delivers each clustering as a contiguous unit. We formalize this assumption by introducing a *contiguity property* in our streaming model.

Contiguity Property: For every $j \in [m]$, all triples referring to clustering C_j appear consecutively in the stream. Formally, if

$$(p_\alpha, j_\alpha, b_\alpha), (p_\beta, j_\beta, b_\beta), (p_\gamma, j_\gamma, b_\gamma)$$

are three triples with $\alpha < \beta < \gamma$ and $j_\alpha = j_\gamma = i$, then $j_\beta = i$ as well. Equivalently, the stream can be viewed as a concatenation of m *blocks*, one for each clustering:

$$\underbrace{(p_{1,1}, j_1, b_{1,1}), \ldots, (p_{1,t_1}, j_1, b_{1,t_1})}_{B_1}$$
$$\underbrace{(p_{2,1}, j_2, b_{2,1}), \ldots, (p_{2,t_2}, j_2, b_{2,t_2})}_{B_2}$$
$$\vdots$$
$$\underbrace{(p_{m,1}, j_m, b_{m,1}), \ldots, (p_{m,t_m}, j_m, b_{m,t_m})}_{B_m}$$

where block B_i contains all $O(n^2)$ triples associated with C_{j_i} where $j_i \in [m]$.

We will simply call any streaming model that satisfies the contiguity property an *insertion-only stream*. On the other hand, a model that does not satisfy this property – where triples can arrive in a fully arbitrary order – will be referred to as a *generalized insertion-only stream*.

In this paper, we provide an algorithm for the k-median fair consensus clustering in the insertion-only streaming model. Furthermore, for a special case of $k = 1$, i.e., the standard fair consensus clustering problem, our algorithm works even in the generalized insertion-only streaming model.

3 FAIR CONSENSUS CLUSTERING: A NEW ALGORITHMIC FRAMEWORK

In this section, we provide a generic framework for solving the 1-median fair consensus clustering problem. Let V be a set of n colored points, and let $\mathcal{I} = \{C_1, C_2, \ldots, C_m\}$ be an input set of m clusterings over V. The goal of a fair consensus clustering problem is to find a fair clustering $\mathcal{F}$ that minimizes the total distance between $\mathcal{F}$ and all clusterings in $\mathcal{I}$, that is, $\text{Obj}(\mathcal{I}, \mathcal{F}) = \sum_{i=1}^{m} \text{dist}(C_i, \mathcal{F})$.

Theorem 1. *Suppose that there is an γ-approximation closest fair clustering with running time $t_1(n)$, then there is a $(\gamma + 1.92)$-approximation algorithm for fair consensus clustering that runs in time $O(m^4n^2 + m^3 t_1(n))$.*

Theorem 1 is implied from our following key technical result, which is a framework leveraging on algorithms for closest fair clustering and fair correlation clustering.

Theorem 2. *Fix parameters $\gamma > 0$, $\eta > 0$, $\alpha > 0$, $0 < \beta < 1$, and $c > 1$. Suppose there exists an algorithm that, given a clustering over n points, computes a γ–close fair clustering in time $t_1(n)$, and suppose there exists a η–approximation algorithm for fair correlation clustering on a graph with n vertices that runs in time $t_2(n)$.*

Then, for any collection of m clusterings over n points, there exists an algorithm running in time

$$O\big(m^4n^2 + mt_1(n) + m^3 t_2(n)\big)$$

that outputs an $(r,1)$–approximate fair consensus clustering, where

$$r = \max\Big\{1+3(\eta+1)\alpha,\ \gamma+2-(\gamma+1)\beta,\ \gamma+2-\frac{2\alpha}{c},\ \gamma+2+(\gamma+1)\frac{\beta}{c-1}-2\Big(1-\frac{1}{c}\Big)\alpha\Big\}.$$

We remind the reader that in a fair correlation clustering problem, we are given as input a complete graph $G = (V, E^+(G) \cup E^-(G))$, where any two vertices are connected by an edge that is either in $E^+(G)$ (labeled "+") or in $E^-(G)$ (labeled "−"). The goal is to find a fair clustering $\mathcal{F}$ minimizing the correlation cost, $\text{cost}(\mathcal{F})$, that is, the number of pairs of vertices connected by a "+" edge but placed in different clusters, and the number of pairs of vertices connected by a "−" edge but placed in the same cluster.

Description of the algorithm. We next present the algorithm that, given a closest fair clustering algorithm for any definition of fairness, obtains a strictly better approximation guarantee for the fair consensus clustering problem. At a high level, our algorithm (Algorithm 2) first forms a candidate set consisting of fair clusterings, and selects from this set a clustering with minimum objective value, among the candidates. We establish this candidate set by iterating through each input clustering, and add to this set a fair clustering that is close to this input. A γ-close fair clustering to each input clustering $C_i \in \mathcal{C}$ can be found by employing a γ-approximation algorithm for closest fair clustering.

Additionally, we also include in this set a fair clustering constructed as follows: for every triple of input clustering $\{C_i, C_j, C_k\}$, we produce a fair clustering that maximizes the number of point pairs whose clustering relation (whether they are in the same cluster or in different clusters) agrees with the majority of the three clustering $\{C_i, C_j, C_k\}$.

Let us elaborate. A fair clustering produced from such a triple is obtained by applying the procedure ClusterFitting (Algorithm 1). This procedure takes as input a set of three clusterings $T = \{C_i, C_j, C_k\}$, and constructs a graph G with vertex set V and edge set $E(G) = E^+(G) \cup E^-(G)$, where

$$E^+(G) = \{(a,b) \mid a,b \in V, a \text{ and } b \text{ are together in at least two clusterings}\}, \text{ and}$$
$$E^-(G) = \{(a,b) \mid a,b \in V, a \text{ and } b \text{ are separated in at least two clusterings}\}.$$

Procedure ClusterFitting (T) computes a fair clustering $\widetilde{T}$ by running a fair correlation clustering algorithm on G.

Algorithm 1: ClusterFitting(T)

Input : Clusterings $T = \{C_i, C_j, C_k\}$
Output: A fair clustering $\widetilde{T}$

$V(G) \leftarrow V$
$E^+(G) \leftarrow$ $\{(a,b) | a,b \in V$, a and b are together in at least 2 clusterings$\}$
$E^-(G) \leftarrow \{(a,b) | a,b \in V$, a and b are separated in at least 2 clusterings$\}$
$\widetilde{T} \leftarrow$ a η-approximation fair correlation clustering of $G = (V(G), E^+(G) \cup E^-(G))$.
return $\widetilde{T}$

In summary, Algorithm 2 computes a candidate set $\widetilde{\mathcal{F}}$, defined as

$$\widetilde{\mathcal{F}} = \{\mathcal{F}_i | \mathcal{F}_i \text{ is a } \gamma\text{-close fair clustering to } C_i \in \mathcal{I}\} \cup \{\widetilde{T} | \widetilde{T} = \text{ClusterFitting}(T), \text{ for each } T \subseteq \mathcal{I} \text{ such that } |T| = 3\}. \quad (1)$$

The output of Algorithm 2 is a clustering in this candidate set minimizing the objective value $\text{Obj}(\mathcal{I})$.

Algorithm 2: A generic framework for fair consensus clustering

Input : A list of clusterings $\mathcal{I} = C_1, C_2, \ldots, C_m$ over V
Output: A fair clustering $\mathcal{F}$

Initialize an empty set $\widetilde{\mathcal{F}}$;
for *each* $C_i \in \mathcal{I}$ **do**
  $\mathcal{F}_i \leftarrow$ a γ-close fair clustering to C_i ;
  $\widetilde{\mathcal{F}} \leftarrow \widetilde{\mathcal{F}} \cup \{\mathcal{F}_i\}$;
for *each triple* $T = \{C_i, C_j, C_k\}$ *of* $\mathcal{I}$ **do**
  $\widetilde{T} \leftarrow$ ClusterFitting(T);
  $\widetilde{\mathcal{F}} \leftarrow \widetilde{\mathcal{F}} \cup \{\widetilde{T}\}$
return $\text{argmin}_{\mathcal{F} \in \widetilde{\mathcal{F}}} \text{Obj}(\mathcal{I}, \mathcal{F})$

We provide the analysis in the full version.

4 STREAMING 1–MEDIAN FAIR CONSENSUS

In this section, we present an algorithm for the 1-median fair consensus clustering problem in the streaming setting. Recall that, in 1-median fair consensus clustering, we are given a set of input clusterings $\mathcal{I} = \{C_1, C_2, \ldots, C_m\}$ defined over a common ground set V (with $|V| = n$). The goal is to output a fair clustering $\mathcal{F}$ that minimizes the objective $\sum_{C_i \in \mathcal{I}} \text{dist}(C_i, \mathcal{F})$, where $\text{dist}(\cdot,\cdot)$ denotes the distance between a candidate clustering and an input clustering.

Theorem 3. *Suppose that here is an γ-approximation closest fair clustering with running time $t_1(n)$, then there is a $(\gamma + 1.995)$-approximation algorithm for fair consensus clustering in the generalized insertion-only streaming model, that uses $O(n \log m)$ space, and has a query time of $O(n^2 \log^4 m + t_1(n) \log^3 m)$.*

The above theorem follows from fixing parameters in the following lemma; the setting of parameters is provided in the full version.

Lemma 4.1. *Suppose there is an algorithm with running time $t_1(n)$ that, given a clustering over n points, computes a γ–close fair clustering, and let there be a η–approximation fair correlation clustering algorithm with running time $t_2(n)$ on a graph with n vertices. Let $\alpha > 0$, $1 > \beta > 0$, $c > 1$, $s > 1$ and $g > s$ be fixed parameters such that $sc - s - 2c > 0$. Then, there exists an algorithm that, for any set of m clusterings over n points and $\varepsilon \in (0,1)$, with probability at least $1 - \frac{1}{m}$, outputs a $\left(\frac{1+\varepsilon}{1-\varepsilon} r, 1\right)$-approximate fair consensus clustering, where*

$$r = \max\Bigg\{(\gamma + 2 - (\gamma+1)\beta), \left(\gamma + 2 + (\gamma+1)\frac{\beta(g-s)+s}{g(s-1)} - \frac{2\alpha}{c}\right), \left(\gamma + 2 + (\gamma+1)\frac{\beta(g(2c+s) - sc) + sc}{g(cs - s - 2c)} - 2\left(1 - \frac{1}{s} - \frac{1}{c}\right)\alpha\right), (1 + 3(\eta+1)\alpha)\Bigg\}.$$

Moreover, the algorithm runs in time

$$O(n^2 \log^4 m + t_1(n) \log m + t_2(n) \log^3 m) \quad (2)$$

and uses $O(n \log m)$ space.

Algorithm Description. We work in the generalized insertion-only stream described in Section 2. Let us now describe our streaming algorithm for the (1-median) fair consensus clustering, `st-1Med`. The pseudocode and complexity analysis are presented in the full version.

The algorithm maintains two memory structures in parallel:

(1) **Sampled Store 1 (M_1):** Prior to the arrival of the stream, we independently sample $4g \log m$ indices $j_1, \ldots, j_{4g \log m}$ from $[m]$. For these sampled indices, all their corresponding triples that appear in the stream are stored in M_1.
(2) **Sampled Store 2 (M_2):** Similarly, before the stream starts, we sample $64\varepsilon^{-2} \log m$ indices $k_1, \ldots, k_t$ from $[m]$, where $t = 64\varepsilon^{-2} \log m$ and $0 < \varepsilon < 1$. For these sampled indices, all their triples are stored in M_2 as they appear in the stream.

After the stream ends.

(1) We use union-find, to construct the clusterings $C_{j_1}, \ldots, C_{j_{\log m}}$ from its triples stored in M_1. Then we apply a subroutine `find-candidates` on these $\lceil \log m \rceil$ clusterings to get a candidate set of fair clusterings $\widetilde{\mathcal{F}}$. The subroutine `find-candidates` assumes access to a γ-close fair clustering algorithm.
(2) We use union-find, to construct the clusterings $C_{k_1}, \ldots, C_{k_t}$ from its triples stored in M_2. Let, $\mathcal{W} = \{C_{k_1}, \ldots, C_{k_t}\}$.
We use $\mathcal{W}$ to find the best fair clustering $\mathcal{F} \in \widetilde{\mathcal{F}}$ with respect to $\mathcal{W}$, more specifically we find
$$\mathcal{F} = \text{argmin}_{\mathcal{F}' \in \widetilde{\mathcal{F}}} \sum_{C_i \in \mathcal{W}} \text{dist}(C_i, \mathcal{F}')$$
(3) Return $\mathcal{F}$.

Analyzing the approximation factor. If there is a constant g such that at least $\frac{m}{g}$ clusterings C_i satisfy $|I_i| \le (1-\beta)\text{AVG}$, then with high probability, our algorithm will sample at least one such clustering. Let $\mathcal{F}$ be a γ-close fair clustering to such sampled clustering, we can show that $\text{Obj}(\mathcal{I}, \mathcal{F}) \le (\gamma + 2 - (\gamma+1)\beta)\text{OPT}$. Hence, in the remaining part of the analysis, we assume that the number of

clusterings C_i with $|I_i| \le (1-\beta)\text{AVG}$ is less than $\frac{m}{g}$. More specifically, we have the following ordering

$$|I_1| \le |I_2| \le \ldots |I_t| \le (1-\beta)\text{AVG} < |I_{t+1}| \le \cdots \le |I_m|, \tag{3}$$

where $t < \frac{m}{g} \le t+1$.

Lemma 4.2. *Suppose that* (3) *holds. For any $t < h \le \frac{m}{s}$, we have*

$$|I_h| \le \left(1 + \frac{\beta(g-s)+s}{g(s-1)}\right)\text{AVG}.$$

Proof. Representing $\text{OPT} = m\text{AVG}$ as the sum off all I_i's, we get that

$$\begin{aligned} m\text{AVG} &= \sum_{i=1}^{m} |I_i| \ge \sum_{i=t+1}^{h-1} |I_i| + \sum_{i=h}^{m} |I_i| \\ &> (h-t-1)(1-\beta)\text{AVG} + (m-h+1)|I_h|. \end{aligned}$$

This implies

$$\begin{aligned} |I_h| &< \frac{m-(1-\beta)(h-t-1)}{m-h+1}\text{AVG} \\ &= \left(1 + \frac{\beta(h-1)}{m-h+1} + \frac{t(1-\beta)}{m-h+1}\right)\text{AVG}. \end{aligned}$$

As $h \le \frac{m}{s}$, we have $\frac{\beta(h-1)}{m-h+1} = \frac{\beta}{(\frac{m}{h-1})-1} \le \frac{\beta}{s-1}$. Moreover, using $t < \frac{m}{g}$, we obtain

$$\frac{t}{m-h+1} < \frac{m/g}{m-m/s+1} \le \frac{s}{g(s-1)}.$$

Since $0 < \beta < 1$, we have $\frac{t(1-\beta)}{m-h+1} < \frac{s(1-\beta)}{g(s-1)}$. Combining these two bounds, we get

$$\begin{aligned} |I_h| &< \left(1 + \frac{\beta}{s-1} + \frac{s(1-\beta)}{g(s-1)}\right)\text{AVG} \\ &= \left(1 + \frac{\beta(g-s)+s}{g(s-1)}\right)\text{AVG}. \end{aligned}$$

□

Lemma 4.3. *Suppose that* (3) *holds. If there is a clustering C_h with $t < h \le \frac{m}{s}$ satisfying $|S_h| \ge \frac{m}{c}$, then*

$$\text{Obj}(\mathcal{I}, \mathcal{F}_h) \le \left(\gamma + 2 + (\gamma+1)\frac{\beta(g-s)+s}{g(s-1)} - \frac{2\alpha}{c}\right)\text{OPT}.$$

Proof. As (3) holds, applying Lemma 4.2 with $h \le \frac{m}{s}$, we obtain $|I_h| \le (1 + \frac{\beta(g-s)+s}{g(s-1)})\text{AVG}$. Moreover, we have

$$\begin{aligned} \text{Obj}(\mathcal{I}, \mathcal{F}_h) &\le \sum_{i=1}^{m} (|I_i| + (\gamma+1)|I_h| - 2|I_i \cap I_h|) \\ &\le \text{OPT} + (\gamma+1)\left(1 + \frac{\beta(g-s)+s}{g(s-1)}\right)\text{OPT} - |S_h|2\alpha\text{AVG} \\ &\le \left(\gamma + 2 + (\gamma+1)\frac{\beta(g-s)+s}{g(s-1)} - \frac{2\alpha}{c}\right)\text{OPT}. \end{aligned}$$

□

We define

$$\begin{aligned} \mathcal{N}_h &:= \{C_j \mid h < j \le h + \frac{m}{c} + \frac{m}{s} : |I_j \cap I_h| \le \alpha\text{AVG}\}, \\ \mathcal{F}_{h,k} &:= \{C_j \mid |I_j \cap I_h| \le \alpha\text{AVG} \text{ and } |I_j \cap I_k| \le \alpha\text{AVG}\}. \end{aligned}$$

Lemma 4.4. *Suppose that* (3) *holds. If for some $h \le \frac{m}{s}$ with $|S_h| < \frac{m}{c}$, there exists a clustering $C_k \in \mathcal{N}_h$ such that $|\mathcal{F}_{h,k}| < \frac{m}{s}$, then*

$$\text{Obj}(\mathcal{I}, \mathcal{F}_k) \le \left(\gamma + 2 + (\gamma+1)\frac{\beta(g(2c+s)-sc)+sc}{g(cs-s-2c)} - 2\left(1 - \frac{1}{s} - \frac{1}{c}\right)\alpha\right)\text{OPT}.$$

Proof. With $C_k \in \mathcal{N}_h$, by definition of $\mathcal{N}_h$, it follows that $k \le h + \frac{m}{c} + \frac{m}{s} \le \frac{m}{sc/(2c+s)}$, where the last inequality is obtained by using the assumption $h \le \frac{m}{s}$. Applying Lemma 4.2 with $k \le \frac{m}{sc/(2c+s)}$, we get $|I_k| \le \left(1 + \frac{\beta(g(2c+s)-sc)+sc}{g(sc-s-2c)}\right)\text{AVG}$.

We now turn to give a lower bound for $|S_k|$. Note that by definition of $\mathcal{F}_{h,k}$, for every clustering $C_j \in \mathcal{I}$, if $C_j \notin \mathcal{F}_{h,k}$, then at least one of $|I_j \cap I_h| > \alpha\text{OPT}$ or $|I_j \cap I_k| > \alpha\text{OPT}$ must hold. In other words, it must be the case that $C_j \in S_h$ or $C_j \in S_k$ (or both) holds. Hence, $|S_h| + |S_k| \ge |\mathcal{I}| - |\mathcal{F}_{h,k}|$. Using the assumptions $|S_h| < \frac{m}{c}$ and $|\mathcal{F}_{h,k}| < \frac{m}{s}$, we obtain $|S_k| \ge m - \frac{m}{c} - \frac{m}{s}$.

Finally, using $|I_k| \le \left(1 + \frac{\beta(g(2c+s)-sc)+sc}{g(sc-s-2c)}\right)$ AVG and $|S_k| \ge m - \frac{m}{c} - \frac{m}{s}$, we have

$$\begin{aligned} &\text{Obj}(\mathcal{I}, \mathcal{F}_k) \\ &\le \sum_{i=1}^{m} (|I_i| + (1+\gamma)|I_k| - 2|I_i \cap I_k|) \\ &\le \text{OPT} + (\gamma+1)\left(1 + \frac{\beta(g(2c+s)-sc)+sc}{g(cs-s-2c)}\right)\text{OPT} - |S_k|2\alpha\text{AVG} \\ &\le \left(\gamma + 2 + (\gamma+1)\frac{\beta(g(2c+s)-sc)+sc}{g(cs-s-2c)} - 2\left(1 - \frac{1}{s} - \frac{1}{c}\right)\alpha\right)\text{OPT}. \end{aligned}$$

□

Lemma 4.5. *Suppose that* `st-1Med` *sampled two clusterings C_h and C_k such that $k \in \mathcal{N}_h$ and $|\mathcal{F}_{h,k}| \ge \frac{m}{s}$. Then, with probability at least $1 - m^{-4}$,* `st-1Med` *also sampled a clustering C_ℓ such that, for $\widetilde{T} = \text{ClusterFitting}(\{C_h, C_k, C_\ell\})$, the following holds:*

$$\text{Obj}\left(\mathcal{I}, \widetilde{T}\right) \le (1 + 3(\eta+1)\alpha)\text{OPT}.$$

Proof Sketch. Since $|\mathcal{F}_{h,k}| \ge \frac{m}{s}$, the probability that at least one clustering C_ℓ in $\mathcal{F}_{h,k}$ is sampled is at least

$$1 - \left(1 - \frac{4g\log m}{m}\right)^{\frac{m}{s}} \ge 1 - e^{-4g\log m/s} = 1 - \frac{1}{m^{4g/s}} \ge 1 - m^{-4}.$$

By definition of $\mathcal{N}_h$ and $\mathcal{F}_{h,k}$, for all $r, s \in \{h, k, \ell\}$, we have $|I_r \cap I_s| \le \alpha\text{AVG}$. Using this, we can show that $\text{Obj}\left(\mathcal{I}, \widetilde{T}\right) \le (1 + 3(\eta+1)\alpha)\text{OPT}$. A detailed proof is provided in the full version. □

Proof of Lemma 4.1. If the number of clusterings C_i with $|I_i| \le (1-\beta)\text{AVG}$ is at least $\frac{m}{g}$, then with probability at least $1 - m^{-4}$, our algorithm samples at least one such clustering. Let $\mathcal{F}$ be a γ-close fair clustering to such sampled clustering, we have $\text{Obj}(\mathcal{I}, \mathcal{F}) \le (\gamma + 2 - (\gamma+1)\beta)\text{OPT}$.

It remains to consider the case when the number of such clusterings C_i is less than $\frac{m}{g}$. With probability at least $1 - m^{-4(1-s/g)}$, our algorithm samples at least one clustering C_h with $\frac{m}{g} < h \le \frac{m}{s}$. If $|S_h| \ge \frac{m}{c}$, then by Lemma 4.3, let $\mathcal{F}_h$ be a γ-close fair clustering to C_h, we have

$$\text{Obj}(\mathcal{I}, \mathcal{F}_h) \le \left(\gamma + 2 + (\gamma+1)\frac{\beta(g-s)+s}{g(s-1)} - \frac{2\alpha}{c}\right)\text{OPT}.$$

If $|S_h| < \frac{m}{c}$, then $|\mathcal{N}_h| \geq \frac{m}{s}$. It follows that with probability at least $1 - m^{-4g/s}$, we sample a clustering $C_k \in \mathcal{N}_h$. If $\left|\mathcal{F}_{h,k}\right| < \frac{m}{s}$, then by Lemma 4.4, let $\mathcal{F}_k$ be a γ-close fair clustering to C_k, we have

$$\text{Obj}(\mathcal{I},\mathcal{F}_k) \leq \left(\gamma+2+(\gamma+1)\frac{\beta(g(2c+s)-sc)+sc)}{g(cs-s-2c)} - 2\left(1-\frac{1}{s}-\frac{1}{c}\right)\alpha\right)\text{OPT}.$$

If $\left|\mathcal{F}_{h,k}\right| \geq \frac{m}{s}$, then by Lemma 4.5, with probability at least $1 - m^{-4}$, there exists a sampled clustering C_ℓ such that, for $\widetilde{T} = \text{ClusterFitting}(\{C_h, C_k, C_\ell\})$, we have

$$\text{Obj}\left(\mathcal{I},\widetilde{T}\right) \leq (1+3(\eta+1)\alpha)\text{OPT}.$$

Combining all the cases, with probability at least $1-m^{-3}$, there is a fair clustering $\mathcal{F} \in \widetilde{\mathcal{F}}$ such that $\text{Obj}(\mathcal{I},\mathcal{F}) \leq r\text{OPT}$, where

$$r = \max\Bigg\{(\gamma+2-(\gamma+1)\beta),\ \left(\gamma+2+(\gamma+1)\frac{\beta(g-s)+s}{g(s-1)}-\frac{2\alpha}{c}\right),\ \left(\gamma+2+(\gamma+1)\frac{\beta(g(2c+s)-sc)+sc)}{g(cs-s-2c)}-2\left(1-\frac{1}{s}-\frac{1}{c}\right)\alpha\right),\ (1+3(\eta+1)\alpha)\Bigg\}.$$

As our algorithm uses a set $\mathcal{W}$ of $\Omega(\varepsilon^{-2}\log m)$ input clusterings sampled uniformly at random as a evaluation set, according to a result developed by Indyk [23], [24, Theorem 31], with probability at least $1-\frac{1}{m}$, our algorithm outputs a clustering $\mathcal{F}'$ such that $\text{Obj}(\mathcal{I},\mathcal{F}') \leq (1+\varepsilon)\text{Obj}(\mathcal{I},\mathcal{F})$. Therefore, with probability at least $1-\frac{1}{m}$, our algorithm outputs a fair clustering $\mathcal{F}'$ with $\text{Obj}(\mathcal{I},\mathcal{F}') \leq (1+\varepsilon)r\text{OPT}$. □

5 k–MEDIAN FAIR CONSENSUS CLUSTERING

In this section, we formally introduce and study the *k–median Consensus Clustering* problem. Given a collection of input clusterings $\mathcal{I} = \{C_1,...,C_n\}$, the objective is to construct a set of k representative clusterings $\mathsf{Z} = \{\mathcal{Z}_1, \dots, \mathcal{Z}_k\}$ that minimizes the total distance between each input clustering and its nearest representative. Formally, we seek to minimize

$$\text{Obj}(\mathcal{I},\mathsf{Z}) = \sum_{C_i \in \mathcal{I}} \min_{\mathcal{Z}_j \in \mathsf{Z}} \text{dist}(C_i,\mathcal{Z}_j).$$

In Section 3, given a γ-close fair clustering algorithm, we generated a $(\gamma+1.92)$ approximation to the fair consensus clustering problem. The high-level idea of the algorithm was to generate a set of candidate fair clusterings $\widetilde{\mathcal{F}}$, which contained the union of all closest fair clusterings to each input clustering $C_i \in \mathcal{I}$ and the set of clusterings we obtain through our cluster-fitting algorithm (Algorithm 1). We proved that one of these candidate clusterings would give us $(\gamma + 1.92)$ approximation. So, we find the best clustering $C \in \widetilde{\mathcal{F}}$ that is $\text{argmin}_{\mathcal{F}\in\widetilde{\mathcal{F}}}\text{Obj}(\mathcal{I},\mathcal{F})$.

Similar to this algorithm, we provide an algorithm that achieves a $(\gamma+1.92)$ approximation to the k-median fair consensus clustering problem by considering k-sized tuples from the candidate list. Consequently, we have the following theorem, the proof of which is deferred to the full version. We improve the running time in the next section when we describe the streaming variant.

Theorem 4. *Suppose we have a $t_1(n)$ time γ-close fair clustering algorithm and $t_2(n)$ time algorithm to find a η-approximate fair correlation clustering, then there exists an algorithm that, given a set of clusterings $\mathcal{I} = \{C_1,...,C_m\}$ finds a set of representatives $\mathsf{Z} = \{\mathcal{Z}_1,...,\mathcal{Z}_k\}$ in*

$$O(mt_1(n)+m^3(n^2+t_2(n))+m^{3k+1}n^2)$$

time such that

$$\text{Obj}(\mathcal{I},\mathsf{Z}) \leq (\gamma+1.92)\text{Obj}(\mathcal{I},\mathcal{Z}^*)$$

where $\mathsf{Z}^* = \{\mathcal{Z}_1^*,...,\mathcal{Z}_k^*\}$ is the optimal set of representatives.

6 STREAMING k–MEDIAN FAIR CONSENSUS

In this section, we present an algorithm for the k–median fair consensus clustering problem in the insertion-only streaming model.

Theorem 5. *Suppose that there is an γ-approximation closest fair clustering with running time $t_1(n)$, then there is a $(1.0151\gamma+1.99951)$-approximation algorithm for k–median fair consensus clustering in the insertion-only streaming model, which uses $O(k^2 n\text{polylog}(mn))$ space, has an update time of $O((km)^{O(1)}n^2\log n)$, and a query time of $O((k\log(mn))^{O(k)}n^2+k^3(n+t_1(n))\log^{12}m\log^3 n)$.*

We achieve this result by developing the following technical theorem.

Theorem 6. *Suppose there is an algorithm with running time $t_1(n)$ that, given a clustering over n points, computes a γ–close fair clustering, and let there be a η–approximation fair correlation clustering algorithm with running time $t_2(n)$ on a graph with n vertices. Let $\alpha > 0$, $1 > \beta > 0$, and $\varepsilon, \varepsilon_1 > 0$ be fixed parameters such that $\alpha/2 > 2\beta(1+\beta)/(1-\beta)$. Then, for any set of m clusterings over n points, there exists a streaming algorithm that outputs $((1+\varepsilon)(r+\varepsilon_1),k)$-approximate fair consensus clustering, where*

$$r = \max\Big\{2+\gamma-(\beta-\lambda)(1+\gamma),\ 2+\lambda+\gamma(1+\beta+\lambda)-\frac{\alpha\beta}{2(\beta+1)}+\frac{2\beta^2}{1-\beta}+o_m(1),\ (1+3(\gamma+1)(\rho+1)\alpha)\Big\}.$$

The algorithm uses a space complexity of $O(k^2 n\ \text{polylog}(mn))$, has update time $O((km)^{O(1)}n^2\ \log\ n)$, and query time $O((k\log(mn))^{O(k)}n^2+k^3t_2(n)\log^2 m\log^3 n)$.

Prior to presenting the algorithm, we introduce a few tools that will be used in the algorithm and its analysis.

Definition 6.1 ((k,ε)-coreset). Consider $\mathcal{I}$ a set of points over a metric space $\mathcal{M}$ equipped with a distance function dist(.,.), and an implicit set $\mathcal{X} \subseteq \mathcal{M}$ (of potential median), a weighted subset $P \subseteq \mathcal{I}$ (with a weight function $w : P \to \mathbb{R}$) is a (k,ε)-coreset of $\mathcal{I}$ with respect to $\mathcal{X}$ for the k-median if for any set $Y \subseteq \mathcal{X}$ of size k

$$(1-\varepsilon)\sum_{C_i\in\mathcal{I}}\text{dist}(C_i,Y) \leq \sum_{\mathcal{P}\in P} w(\mathcal{P})\text{dist}(Y,\mathcal{P}) \leq (1+\varepsilon)\sum_{C_i\in\mathcal{I}}\text{dist}(C_i,Y) \quad (4)$$

We use the following coreset constructions.

Theorem 7 ([3, 8, 18]). *There is an algorithm that, given a set $\mathcal{I}$ of m points of an arbitrary metric space $\mathcal{M}$ and an implicit set $\mathcal{X} \subseteq \mathcal{M}$ (without loss of generality assume $\mathcal{I} \subseteq \mathcal{X}$), outputs a ε-coreset of $\mathcal{I}$ with respect to $\mathcal{X}$ for the k–median problem, of size $O(\varepsilon^{-2}\log|\mathcal{X}|)$. The algorithm succeeds with probability at least $1-\frac{1}{m^2}$ and runs in time $O(\varepsilon^{-2}m\log|\mathcal{X}|)$.*

By combining Theorem 7 with the framework provided in [7], we obtain the following streaming coreset construction.

Lemma 6.2. *There is a (randomized) streaming algorithm that, given a set $\mathcal{I}$ of m points of an arbitrary metric space $\mathcal{M}$, arriving in an insertion-only streaming model, and an implicit set $\mathcal{X} \subseteq \mathcal{M}$ (without loss of generality assume $\mathcal{I} \subseteq \mathcal{X}$), maintains a ε-coreset of $\mathcal{I}$ with respect to $\mathcal{X}$ for the k−median problem, by storing at most $O(\varepsilon^{-2}k \log |\mathcal{M}| \log m)$ points of $\mathcal{I}$. The algorithm has worst-case update time of $(\varepsilon^{-1}k\log m)^{O(1)}$.*

Another sampling technique we use is the *monotone faraway sampling* introduced by [9].

Lemma 6.3 (Monotone Faraway Sampling [9])**.** *There is a (randomized) streaming algorithm that, given a set $\mathcal{I}$ of m points of an arbitrary metric space $\mathcal{M}$, arriving in an insertion-only streaming model, and parameters $\kappa, \mu \in (0,1)$, samples a subset $F \subseteq \mathcal{I}$ of size $O(k^2(\mu\kappa)^{-1} \log k \log(1 + k\kappa m))$ such that the following holds: Let $C^* = \{C_1^*,\ldots,C_k^*\}$ be an arbitrary optimum k−median of $\mathcal{I}$, and let $M_1,\ldots,M_k$ be the clustering of $\mathcal{I}$ induced from C^*. Then for each $i \in [k]$, there exists a $C_i' \in F$ satisfying*

$$\sum_{C \in M_i} \mathrm{dist}(C', C_i') \le 2\left(1 + \frac{1}{1-\kappa}\right) \sum_{C \in M_i} \mathrm{dist}(C, C_i^*) + \mu \frac{\mathrm{OPT}}{k}.$$

The algorithm requires both space and update time of $O(k^2(\mu\kappa)^{-1}\log k\log(1+k\kappa m))$.

Algorithms in Lemma 6.2 and Lemma 6.3 succeeds with probability at least $\frac{9}{10}$. Note that in our consensus clustering problem, the underlying metric space $\mathcal{M}$ consists of the set of all clusterings over V, and each input clustering (in $\mathcal{I}$) is a point in the metric space $\mathcal{M}$.

Description of the Algorithm. Our algorithm consists of four components. The first three components process simultaneously the input clusterings as they arrive in the stream to establish a candidate set and construct a coreset. By using such sets, the last component simulates the offline algorithm presented in Section 5 to output k fair clusterings as k-median.

•**Step 1.** We simultaneously do the following steps as the clusterings from the input set $\mathcal{I}$ arrives in the stream.

Step 1A (Sampling Algorithm) We choose constants λ, $\tau > 0$, the value of which to be fixed later. For each $\ell \in \{\frac{1}{2}, \frac{1}{2}(1+\tau), \frac{1}{2}(1+\tau)^2, \ldots, n^2\}$ and $p \in \{1, \frac{1}{1+\tau}, \frac{1}{(1+\tau)^2}, \ldots, \frac{1}{m}\}$, we construct a set $\mathcal{S}_{\ell,p} \subset \mathcal{I}$ as follows.

- **Step a** For each arrived C_i, discard it with probability $1-p$.
- **Step b** If $\mathrm{dist}(C_i, C_j) \ge \lambda\ell$, for all $C_j \in \mathcal{S}_{\ell,p}$, add C_i to $\mathcal{S}_{\ell,p}$.
- **Step c** If $|\mathcal{S}_{\ell,p}| \ge k\log^3 m$, set $\mathcal{S}_{\ell,p} = \emptyset$.

Set $\mathcal{S} = \cup_{\ell,p}\mathcal{S}_{\ell,p}$.

Step 1B (Monotone Faraway Sampling) As the clusterings from the input set $\mathcal{I}$ arrives in the stream, we run the algorithm from Lemma 6.3 with parameters $\kappa = 1/3$ and a constant $\mu > 0$. Let F be the output of this algorithm. Then, for any arbitrary optimal k−median clusterings $C^* = \{C_1^*,\ldots,C_k^*\}$ of $\mathcal{I}$, and the corresponding super-clusters $M_1, \ldots, M_k$, for each $i \in [k]$, there exists a $C_i' \in F$ satisfying

$$\sum_{C \in M_i} \mathrm{dist}(C, C_i') \le 5 \sum_{C \in M_i} \mathrm{dist}(C, C_i^*) + \mu \frac{\mathrm{OPT}}{k}.$$

Step 1C (Coreset Construction) We consider the candidate set $\widetilde{\mathcal{F}}$ as in (1). For each $C_i \in \mathcal{I}$, $\widetilde{\mathcal{F}}$ contains $\mathcal{F}_i$, a γ-close fair clustering to C_i. For every triple $T = (C_i, C_j, C_k)$, $\widetilde{\mathcal{F}}$ contains $\widetilde{T} = \mathrm{ClusterFitting}(\{T\})$.

As the clusterings from the input set $\mathcal{I}$ arrives in the stream, we run the algorithm from Lemma 6.2 with a constant $\varepsilon > 0$ to build a (k,ε)-coreset (P,w) with respect to the implicit set $\widetilde{\mathcal{F}}$.

• **Step 2 (Simulating the Offline Algorithm)** We use a candidate set $\mathcal{R}$ by iterating through each clustering $C_i \in \mathcal{S} \cup F$, including $\mathcal{F}_i$ in $\mathcal{R}$, where $\mathcal{F}_i$ is a γ-close fair clustering to C_i. Moreover, for every triple $T = \{C_i, C_j, C_k\} \subseteq \mathcal{S}$, we include $\widetilde{T} = \mathrm{ClusterFitting}(T)$ in $\mathcal{R}$. Finally, for each k-tuple $Y = (\mathcal{Y}_1,\ldots,\mathcal{Y}_k) \in \mathcal{R}^k$, we compute the objective $\mathrm{Obj}(P,Y)$ and return the k-tuple minimizing this value. We defer the entire analysis to the full version.

7 CONCLUSION AND FUTURE WORK

In this paper, we initiate the study of fair consensus clustering under the k-median objective in the streaming model and present the first constant-factor algorithm that operates in sublinear space. Relative to prior work on fair consensus clustering [12], our contribution advances the state of the art by handling streaming data with sublinear memory and by delivering the first constant-factor guarantee for the more general k-median objective. We also introduce a new generic algorithmic framework that works irrespective of the specific fairness notion and the number of colored groups (whether disjoint or not) – in fact, it works with any constraint – provided the corresponding closest fair (constrained) clustering problem can be efficiently approximated.

Improving space usage, particularly for the k-median variant, and tightening approximation guarantees are among a few interesting open directions. An exciting avenue is to narrow the gap between the approximation factor achieved by our framework and the best attainable for the closest fair clustering problem. However, existing hardness results for fair consensus clustering, together with exact algorithms for closest fair clustering in certain special cases [12], preclude any fully generic fair consensus algorithm from matching the approximation factor of the closest fair clustering exactly. Another intriguing direction of research is to explore learning-augmented approaches, leveraging machine-learned predictors further to improve the approximation quality in fair consensus clustering, at least empirically.

We would like to mention that in this paper we consider two variants of the streaming model, one in which all the pairs for a particular input clustering arrive together (in a stream), which we refer to as an insertion-only stream, and then a general model where pairs may appear in an arbitrary order, which we refer to as a generalized insertion-only stream. Our 1-median algorithm works for this generalized model, while extension to the k-median only works for the insertion-only model. Thus, designing a streaming algorithm for the k-median variant in the generalized insertion-only model would be an interesting open direction. It would also be intriguing to study other possible streaming models in the context of consensus clustering, and we leave this as a potential future direction.

ACKNOWLEDGMENTS

Diptarka Chakraborty was supported in part by an MoE AcRF Tier 1 grant (T1 251RES2303) and a Google South & South-East Asia Research Award. Kushagra Chatterjee was supported by an MoE AcRF Tier 1 grant (T1 251RES2303). Debarati Das was supported in part by NSF grant 2337832.

REFERENCES

[1] Nir Ailon, Moses Charikar, and Alantha Newman. 2008. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)* 55, 5 (2008), 1–27.

[2] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. 2002. Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 1–16.

[3] Olivier Bachem, Mario Lucic, and Silvio Lattanzi. 2018. One-shot coresets: The case of k-clustering. In *International conference on artificial intelligence and statistics*. PMLR, 784–792.

[4] Pierre Bellec, Pedro Rosa-Neto, Oliver C Lyttelton, Habib Benali, and Alan C Evans. 2010. Multi-level bootstrap analysis of stable clusters in resting-state fMRI. *Neuroimage* 51, 3 (2010), 1126–1139.

[5] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in Neural Information Processing Systems (NeurIPS)* 29 (2016).

[6] Paola Bonizzoni, Gianluca Della Vedova, Riccardo Dondi, and Tao Jiang. 2008. On the Approximation of Correlation Clustering and Consensus Clustering. *J. Comput. Syst. Sci.* 74, 5 (2008), 671–696.

[7] Vladimir Braverman, Dan Feldman, Harry Lang, and Daniela Rus. 2019. Streaming coreset constructions for m-estimators. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques* (2019).

[8] Vladimir Braverman, Shaofeng H-C Jiang, Robert Krauthgamer, and Xuan Wu. 2021. Coresets for clustering in excluded-minor graphs and beyond. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2679–2696.

[9] Vladimir Braverman, Harry Lang, Keith Levin, and Yevgeniy Rudoy. 2021. Metric k-median clustering in insertion-only streams. *Discrete Applied Mathematics* 304 (2021), 164–180.

[10] Rich Caruana, Mohamed Elhawary, Nam Nguyen, and Casey Smith. 2006. Meta clustering. In *Sixth International Conference on Data Mining (ICDM'06)*. IEEE, 107–118.

[11] Diptarka Chakraborty, Kushagra Chatterjee, Debarati Das, and Tien Long Nguyen. 2026. Generalizing Fair Clustering to Multiple Groups: Algorithms and Applications. In *The Fortieth AAAI Conference on Artificial Intelligence (AAAI-26), Singapore*.

[12] Diptarka Chakraborty, Kushagra Chatterjee, Debarati Das, Tien Long Nguyen, and Romina Nobahari. 2025. Towards Fair Representation: Clustering and Consensus. In *The Thirty Eighth Annual Conference on Learning Theory, 30-4 July 2025, Lyon, France (Proceedings of Machine Learning Research, Vol. 291)*, Nika Haghtalab and Ankur Moitra (Eds.). PMLR, 838–853.

[13] Diptarka Chakraborty, Debarati Das, and Robert Krauthgamer. 2023. Clustering permutations: New techniques with streaming applications. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 31–1.

[14] Diptarka Chakraborty, Himika Das, Sanjana Dey, and Alvin Hong Yao Yan. 2025. Improved Rank Aggregation Under Fairness Constraint. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, James Kwok (Ed.). International Joint Conferences on Artificial Intelligence Organization, 330–338. Main Track.

[15] Moses Charikar, Liadan O'Callaghan, and Rina Panigrahy. 2003. Better streaming algorithms for clustering problems. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*. 30–39.

[16] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. 2017. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems (NeurIPS)*. 5029–5037.

[17] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Innovations in Theoretical Computer Science*. 214–226.

[18] Dan Feldman and Michael Langberg. 2011. A unified framework for approximating and clustering data. In *Proceedings of the forty-third annual ACM symposium on Theory of Computing*. 569–578.

[19] Vladimir Filkov and Steven Skiena. 2004. Heterogeneous data integration with the consensus clustering formalism. In *International Workshop on Data Integration in the Life Sciences*. Springer, 110–123.

[20] Vladimir Filkov and Steven Skiena. 2004. Integrating microarray data by consensus clustering. *International Journal on Artificial Intelligence Tools* 13, 04 (2004), 863–880.

[21] Andrey Goder and Vladimir Filkov. 2008. Consensus clustering algorithms: Comparison and refinement. In *2008 Proceedings of the Tenth Workshop on Algorithm Engineering and Experiments (ALENEX)*. SIAM, 109–117.

[22] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*. 3315–3323.

[23] Piotr Indyk. 1999. Sublinear time algorithms for metric space problems. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*. 428–434.

[24] Piotr Indyk. 2001. *High-dimensional computational geometry*. stanford university.

[25] Matthew Kay, Cynthia Matuszek, and Sean A Munson. 2015. Unequal representation and gender stereotypes in image search results for occupations. In *ACM conference on human factors in computing systems*. 3819–3828.

[26] Mirko Křivánek and Jaroslav Morávek. 1986. NP-hard problems in hierarchical-tree clustering. *Acta informatica* 23 (1986), 311–323.

[27] Andrea Lancichinetti and Santo Fortunato. 2012. Consensus clustering in complex networks. *Scientific reports* 2, 1 (2012), 336.

[28] Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. 2003. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning* 52 (2003), 91–118.

[29] Shanmugavelayutham Muthukrishnan et al. 2005. Data streams: Algorithms and applications. *Foundations and Trends® in Theoretical Computer Science* 1, 2 (2005), 117–236.

[30] Guy Rosman, Mikhail Volkov, Danny Feldman, John W Fisher, and Daniela Rus. 2014. Coresets for k-segmentation of streaming data. *Advances in neural information processing systems* 27 (2014).

[31] Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. 2019. Fair coresets and streaming algorithms for fair k-means. In *International Workshop on Approximation and Online Algorithms*. Springer, 232–251.

[32] Chaitanya Swamy. 2004. Correlation Clustering: maximizing agreements via semidefinite programming.. In *SODA*, Vol. 4. Citeseer, 526–527.

[33] Alexander Topchy, Anil K Jain, and William Punch. 2003. Combining multiple weak clusterings. In *Third IEEE international conference on data mining*. IEEE, 331–338.

[34] Alexander Topchy, Anil K Jain, and William Punch. 2005. Clustering ensembles: Models of consensus and weak partitions. *IEEE transactions on pattern analysis and machine intelligence* 27, 12 (2005), 1866–1881.

[35] Junjie Wu, Hongfu Liu, Hui Xiong, Jie Cao, and Jian Chen. 2014. K-means-based consensus clustering: A unified view. *IEEE transactions on knowledge and data engineering* 27, 1 (2014), 155–169.