

Improved Streaming Algorithm for Fair k -Center Clustering

Longkun Guo^{1,2*}, Zeyu Lin¹, Chaoqi Jia³, Chao Chen³

¹ School of Mathematics and Statistics, Fuzhou University, Fuzhou 350116, China

² School of Integrated Circuits, Guangdong University of Technology, Guangzhou 510006, China

³ School of Accounting, Information Systems and Supply Chain, RMIT University, Melbourne, VIC 3000, Australia
longkun.guo@gmail.com, zeyu_lin@foxmail.com, {chaoqi.jia,chao.chen}@rmit.edu.au

Abstract

Many real-world applications call for incorporating fairness constraints into the k -center clustering problem, where the dataset is partitioned into m demographic groups, each with a specified upper bound on the number of centers to ensure fairness. Focusing on big data scenarios, this paper addresses the problem in a streaming setting, where data points arrive sequentially in a continuous stream. Leveraging a structure called the λ -independent center set, we propose a one-pass streaming algorithm that first computes a reserved set of points during the streaming process. In the post-streaming process, we then select centers from the reserved point set by analyzing three possible cases and transforming the most complex one into a specially constrained vertex-cover problem on an auxiliary graph. Our algorithm achieves an approximation ratio of $5 + \epsilon$ and memory complexity $O(k \log \alpha)$, where α is the aspect ratio and $\epsilon > 0$ is any small constant. Furthermore, we extend our approach to semi-structured data streams, where data points arrive in groups. In this setting, we present a $(3 + \epsilon)$ -approximation algorithm for $m = 2$, which can be readily adapted to solve the offline fair k -center problem, achieving an approximation ratio of 3 that matches the current state of the art. Lastly, we conduct extensive experiments to evaluate the performance of our approaches, demonstrating that they outperform existing baselines in both clustering cost and runtime efficiency.

Code — <https://github.com/ChaoqiJia/StreamFKC>

1 Introduction

Fair k -center clustering is a popular problem in various fields, including data summarization (Kleindessner, Awasthi, and Morgenstern 2019; Angelidakis et al. 2022) and machine learning (Chierichetti et al. 2017; Jones, Nguyen, and Nguyen 2020). The problem has been widely studied, with many definitions of fairness proposed and corresponding approximation algorithms discussed, such as group fairness (Chierichetti et al. 2017; Wu et al. 2024; Backurs et al. 2019), data summarization fairness (Kleindessner, Awasthi, and Morgenstern 2019; Ceccarello, Pietracaprina, and Pucci 2024; Wu et al. 2024; Lin, Guo, and Jia 2024), colorful fairness (Bandyapadhyay et al. 2019; Anegg, Vargas Koch, and Zenklusen 2022;

Jia, Sheth, and Svensson 2022) and so on. In this paper, we focus on the fair k -center problem for data summarization. We consider a dataset S of size n divided into m disjoint groups, denoted as $S = S_1 \cup \dots \cup S_m$. Our goal is to select k centers to minimize the maximum distance from each point to its nearest center, while the number of centers chosen from each group S_l is bounded by k_l . This formulation helps avoid bias toward some sensitive features by controlling the number of objects from each category in the output. For example, it can dictate the number of movies of each genre shown to a user in a recommendation system or limit the number of old messages included in a summary of a user’s feed (Mahabadi and Trajanovski 2024).

A traditional challenge in clustering is the need to process large-scale datasets across numerous applications. In such scenarios, storing the entire input in memory becomes impractical, giving rise to the streaming model. In this model, data points arrive sequentially in a stream, and only a limited portion can be retained in memory due to space constraints. The streaming algorithm must decide, upon the arrival of each data point, whether to store it or discard it. Notably, designing effective streaming algorithms is more challenging than developing offline algorithms, as decisions must be made based on partial information rather than having access to the entire dataset. Motivated by this challenge, we aim to propose approximation algorithms for the fair k -center problem under the streaming setting.

1.1 Related Work

Fairness is an important concept for many research fields such as data summarization (Kleindessner, Awasthi, and Morgenstern 2019), private security (Zhang, Yuan, and Pan 2024), and machine learning (Zhang et al. 2024, 2025). In this paper, we briefly review the literature on the k -center problem, focusing on studies about data summarization fairness and streaming model settings.

Data summarization fairness k -center. The fair k -center clustering was formally addressed in the context of data summarization by Kleindessner, Awasthi, and Morgenstern (2019). They proposed a 5-approximation algorithm for the case of two groups and a $(3 \times 2^{m-1} - 1)$ -approximation for the general case with m groups. Subsequently, Jones, Nguyen, and Nguyen (2020) reduced the approximation ra-

*Corresponding Author

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

tio to 3 for arbitrary m groups. Before these two relevant works, the matroid center problem, a generalization of the k -center problem that enforces a matroid constraint on the center set rather than a simple cardinality constraint, was studied by Chen et al. (2016). They developed a 3-approximation algorithm, although with considerably higher time complexity. More recently, Chen et al. (2024) introduced the fair k -supplier problem, which involves selecting k facilities from a dataset partitioned into m disjoint groups, subject to group-wise upper bounds on the number of facilities selected. They presented a 5-approximation algorithm for this problem by maximum matching techniques.

Computation of k -center for Big Data. For large-scale datasets in real-time streaming scenarios, streaming k -center clustering was studied by Matthew McCutchen and Khuller (2008), who provided a streaming $(4+\epsilon)$ -approximation algorithm with outliers, where up to z input points can be dropped and $O(kz/\epsilon)$ memory is used. Then, a deterministic one-pass streaming algorithm was developed for the same problem by Ceccarello, Pietracaprina, and Pucci (2019), achieving an improved approximation ratio $(3+\epsilon)$ with $O((k+z)(96/\epsilon)^d)$ memory in d dimensions. For the matroid center problem, a $(17+\epsilon)$ -approximation one-pass algorithm with a running time $O_\epsilon((nk+k^{3.5})+k^2\log(\Lambda))$ has been developed by Kale (2019), where k is the rank of the matroid, Λ is the aspect ratio of the metric, and ϵ terms are hidden by the O_ϵ notation. For the fair k -center problem, Chiplunkar, Kale, and Ramamoorthy (2020) provided a distributed algorithm achieving an approximation ratio of $(17+\epsilon)$ and with $O(kn/l+mk^2l)$ running time for l processors. They also developed a two-pass streaming algorithm with an approximation ratio 3 for the fair k -center problem. Recently, the fair range k -center was addressed by Nguyen, Nguyen, and Jones (2022) with an approximation ratio of 13 and Ceccarello, Pietracaprina, and Pucci (2024) employed a coresets-based method for the colorful fair k -center problem in doubling metrics, achieving a $(3+\epsilon)$ -approximation ratio for sequential, streaming and MapReduce/MPC algorithms. In addition, a one-pass streaming algorithm for representative fair k -center clustering by Lin, Guo, and Jia (2024) achieved the state-of-the-art approximation ratio of $(7+\epsilon)$. This paper improves the approximation ratio to $(5+\epsilon)$ under a streaming setting by utilizing λ -independent center sets and modifying the post-streaming algorithms.

1.2 Our Contributions

The main contributions can be summarized as follows:

- Devise a one-pass streaming algorithm for fair k -center, achieving an improved approximation ratio of $5+\epsilon$ while consuming $O(k\log\alpha)$ memory, where α is the aspect ratio and $\epsilon > 0$ is any small constant. This significantly improves the previous SOTA ratio $7+\epsilon$ due to Lin, Guo, and Jia (2024).
- For semi-structured data streams where data points of each group S_l are streamed as a batch, we propose a streaming algorithm achieving an approximation ratio of $3+\epsilon$ for $m=2$.

- Conduct extensive experiments on real-world datasets to demonstrate the practical performance gains of our algorithms in clustering accuracy and runtime efficiency.

Notably, we also provide an example demonstrating that our ratio analysis is tight for the algorithm. Moreover, the streaming algorithm with ratio $3+\epsilon$ can be adapted to solve the offline fair k -center, achieving a ratio of 3, which matches the state-of-the-art ratio 3 for the offline setting due to Jones, Nguyen, and Nguyen (2020).

2 Preliminary and Algorithmic Framework

Let S be a finite set of data points with size n distributed in a metric space, where $d : S \times S \rightarrow \mathbb{R}_{\geq 0}$ is the distance on S that satisfies *the triangle inequality*. For a given parameter $k \in \mathbb{N}$, the traditional k -center clustering problem is to select a set of k points $C \subseteq S$ to serve all points in S such that $\max_{s \in S} d(s, C)$ is minimized, where $d(s, C) = \min_{c \in C} d(s, c)$ is the distance between a point s and the center set C . Let the data set be divided into m disjoint groups $S = S_1 \cup \dots \cup S_m$. There exists a fairness constraint that the number of chosen centers from group S_l is bounded by k_l , where k_l is the upper bound for group l for all $l \in [m]$ and $[m]$ denotes $\{1, 2, \dots, m\}$ for brevity. Moreover, $\sum_{l=1}^m k_l = k$. Then, the fair k -center clustering problem is to find a center set C satisfying the formulation as follows:

$$\begin{aligned} \min_{C \subseteq S} \quad & \max_{s \in S} d(s, C) \\ \text{s.t.} \quad & |C \cap S_l| \leq k_l, \quad \forall l, \\ & |C| \leq k. \end{aligned}$$

2.1 Algorithmic Framework

Next, we introduce the general framework of our algorithms. Throughout this paper, we assume that the optimal radius of the fair k -center problem is known as r^* (i.e., $r^* = \max_{s \in S} d(s, C^*)$ for the optimum center set C^*). Although the exact value of r^* is actually unknown, we can employ the previous approach for finding suitable replacement of r^* in the offline setting due to Guo et al. (2025) by compromising the ratio with a multiplicative factor of $1+\epsilon$. In general, all of our algorithms mainly proceed in two stages.

- **Streaming stage.** Select a set of representative data points along the stream according to the optimum radius r^* , where the union of the selected point sets might have size larger than k ;
- **Post-streaming stage.** Compute the set of actual centers from the set of streamed and stored representative data points according to r^* .

2.2 The Streaming Stage

For the streaming stage, we employ the λ -independent center set as defined below:

Definition 1. (λ -independent center set) $\Gamma \subseteq S$ is a λ -independent center set of S if and only if it satisfies the following two conditions:

- 1) For any two points $p, q \in \Gamma$, the distance between them is larger than λ , i.e. $d(p, q) > \lambda$.
- 2) For any point $p \in S$, there exists a point $q \in \Gamma$, such that $d(p, q) \leq \lambda$.

We say a λ -independent center set $\Gamma \subseteq S$ is *minimal* iff removal of any point from Γ makes Γ no longer a λ -independent center set. Assume C^* is the center set of an optimal solution and recall that r^* is the optimal radius. Then we have:

Lemma 1. For a minimal λ -independent center set $\Gamma \subseteq S$, if $\lambda \geq 2r^*$, then $|\Gamma| \leq k$.

Proof. We only need to show the case for $\lambda = 2r^*$. Suppose $|\Gamma| > k$. Then, by the Pigeonhole principle, there must exist at least a pair of points $i, j \in \Gamma$, which are covered by the same center in the optimal solution, say c^* in C^* . Then, both $d(i, c^*) \leq r^*$ and $d(c^*, j) \leq r^*$ hold. By the triangle inequality, we then have $d(i, j) \leq d(i, c^*) + d(c^*, j) \leq r^* + r^* = 2r^*$. On the other hand, we have $d(i, j) > 2r^*$ according to Cond. (1) in Def. 1, resulting in a contradiction and hence completing the proof. \square

So for the streaming stage, we only need to construct a *minimal* λ -independent center set Γ_l for each $l \in \{1, 2\}$ regarding $\lambda = 2r^*$ along the stream:

Upon each arriving point i do
if $i \in S_l$ and $d(i, \Gamma_l) > \lambda$ both hold then
 Grow Γ_l (initially empty) by adding i .

3 Post-streaming Stage of Fair k -Center

Observing that the size of $\Gamma_1 \cup \Gamma_2$ obtained in the streaming stage may exceed k_l for $\lambda = 2r^*$, we devise a method to construct the actual center set C by selecting points from the union of Γ_l for $m = 2$ based on dealing with the cases whether $|\Gamma_l| \leq k_l$ holds or not. We extend the method to general m in the full version.

3.1 Overview of Post-streaming Stage

Our algorithm for this stage is grounded in analyzing the sizes of Γ_1 and Γ_2 . There are exactly three possible cases regarding the sizes of Γ_1 and Γ_2 :

- (1) $|\Gamma_1| \leq k_1, |\Gamma_2| \leq k_2$;
- (2) $|\Gamma_1| \leq k_1, |\Gamma_2| > k_2$ or $|\Gamma_1| > k_1, |\Gamma_2| \leq k_2$;
- (3) $|\Gamma_1| > k_1, |\Gamma_2| > k_2$.

For Case (1), we can directly use $C = \Gamma_1 \cup \Gamma_2$ as the desired center set, because: (1) $|C| = |\Gamma_1 \cup \Gamma_2| \leq k_1 + k_2 = k$; (2) for any point $s \in S$, $d(s, C) \leq 2r^*$ holds due to the definition of λ -independent center set and $\lambda = 2r^*$.

Then, we give a method to solve Case (2) and show that Case (3) can be reduced to Case (2).

3.2 Processing of Case (2)

Without loss of generality, we assume that $|\Gamma_l| > k_l$ and $|\Gamma_{3-l}| \leq k_{3-l}$ for $l \in \{1, 2\}$. Then, our algorithm simply proceeds as follows:

Construct $\Gamma'_l = \{i | i \in \Gamma_l, d(i, \Gamma_{3-l}) > 3r^*\}$ and return $C = \Gamma_{3-l} \cup \Gamma'_l$ as the desired center set.

The correctness of such C can be derived from the following theorem:

Theorem 1. For $C = \Gamma'_l \cup \Gamma_{3-l}$, we have: (1) $|\Gamma'_l| \leq k_l$, $|\Gamma_{3-l}| \leq k_{3-l}$, $|C| \leq k$; (2) for $\forall s \in S$, $d(s, C) \leq 5r^*$.

Proof. To prove $|\Gamma'_l| \leq k_l$ of Cond. (1), we first show that each $i \in \Gamma'_l$ cannot be covered by any point of S_{3-l} in the optimal clustering, i.e., $d(i, j) > r^*$ holds for any point $j \in S_{3-l}$. Suppose this is not true, i.e., for a point $i \in \Gamma'_l$, there exists a point j in S_{3-l} with $d(i, j) \leq r^*$. According to the definition of the $2r^*$ -independent center set, there must exist a point $p \in \Gamma_{3-l}$ with $d(j, p) \leq 2r^*$. So we get $d(i, p) \leq d(i, j) + d(j, p) \leq r^* + 2r^* = 3r^*$, where the second inequality is by the triangle inequality. This contradicts the definition of Γ'_l that $d(i, \Gamma_{3-l}) > 3r^*$ holds for each $i \in \Gamma'_l$. Then, $|\Gamma'_l| \leq k_l$ must hold, because otherwise there must exist at least two points $p, q \in \Gamma'_l$ belonging to an identical cluster in the optimal solution. That means $d(p, q) \leq 2r^*$, contradicting the fact $d(p, q) > 2r^*$ as p, q are two points of the $2r^*$ -independent center set Γ_l .

Next, we show $d(s, C) \leq 5r^*$ for any $s \in S$. Firstly, following the algorithm, there exists a point $p \in \Gamma_l$ for any $s \in S_l$ such that $d(s, p) \leq 2r^*$. Then if point $p \in \Gamma'_l$, $d(s, p) \leq 2r^*$ is true; otherwise, i.e. $p \notin \Gamma'_l$, then there exists a point $q \in \Gamma_{3-l}$ with $d(p, q) \leq 3r^*$, indicating $d(s, q) \leq d(s, p) + d(p, q) \leq 2r^* + 3r^* = 5r^*$. Secondly, for each point $s \in S_{3-l}$, there exists a point $p \in \Gamma_{3-l}$ such that $d(s, p) \leq 2r^*$. Moreover, Γ_{3-l} remains unchanged during the algorithm, so $d(s, p) \leq 2r^*$ remains true for $p \in \Gamma_{3-l}$. Hence, for each $s \in S = S_1 \cup S_2$, $d(s, C) \leq 5r^*$ holds. \square

3.3 Processing of Case (3)

We propose a more sophisticated algorithm to ensure the approximation ratio of 5 for Case (3). The key idea of our algorithm is to construct an auxiliary bipartite graph $G(\Gamma_1 \cup \Gamma_2, E)$, such that the center selection problem regarding $\Gamma_1 \cup \Gamma_2$ is transformed into a special constrained vertex cover problem in G . The construction of $E(G)$ simply proceeds as:

For every pair of points $p \in \Gamma_1$ and $q \in \Gamma_2$, add an edge $e(p, q)$ to $E(G)$ if and only if $d(p, q) \leq 3r^*$.

Then, the aim is to find a vertex cover in G with a size bounded by k that also satisfies the fairness constraints k_1 and k_2 . Notably, although finding a vertex cover is \mathcal{NP} -hard in general, we manage to devise a polynomial-time exact algorithm for the problem based on certain special properties of the constructed auxiliary graph.

The key idea of our algorithm is to repeatedly eliminate degree 0 and degree 1 points in G when any exist, by selecting the point covering such points from $V(G)$ as a new center in the set C . While G contains no degree-0 or degree-1 points, our algorithm arbitrarily chooses an edge in G , and selects one of its endpoints as a new center by adding it to C . The procedure repeats until the problem reduces to Case (2). The detailed algorithm is illustrated in Alg. 1.

Algorithm 1: Processing of Case (3)

Input: Γ_1 and Γ_2 with $|\Gamma_1| > k_1, |\Gamma_2| > k_2$.
Output: Center set C .
// Phase 1: Construction of G and C .

- 1 Set $C \leftarrow \emptyset$;
- 2 Construct the auxiliary graph $G(\Gamma_1 \cup \Gamma_2, E)$, where an edge $e(p, q)$ exists in E if and only if $p \in \Gamma_1, q \in \Gamma_2$ and $d(p, q) \leq 3r^*$;
- 3 **for** each point $i \in G$ with degree 0 **do**
- 4 **if** $d(i, C) > 2r^*$ **then**
- 5 Set $C \leftarrow C \cup \{i\}$;
- 6 Remove each point $i \in G$ with degree 0;
- 7 // Phase 2: Final computation of C .
- 8 **while** $|C| \leq k = k_1 + k_2$ and $V(G) \neq \emptyset$ **do**
- 9 **if** there exists no degree-1 point in G **then**
- 10 Arbitrarily select an edge $e(p, q)$ and remove it from G ;
- 11 Set $C \leftarrow C \cup \{p\}$ and $G \leftarrow G \setminus \{p, q\}$;
- 12 Set $\Gamma_l \leftarrow \Gamma_l \setminus \{p\}$ and $\Gamma_{3-l} \leftarrow \Gamma_{3-l} \setminus \{q\}$ for Γ_l containing p ;
- 13 **else**
- 14 Find $i \in G$ with largest $N_1(i)$, where $N_1(i)$ is the set of degree-1 neighbours of i in G ;
- 15 Set $C \leftarrow C \cup \{i\}$ and $G \leftarrow G \setminus \{i\} \setminus N_1(i)$;
- 16 Set $\Gamma_l \leftarrow \Gamma_l \setminus \{i\}$ and $\Gamma_{3-l} \leftarrow \Gamma_{3-l} \setminus N_1(i)$ for Γ_l containing i ;
- 17 **if** there exists l with $|C \cap S_l| + |\Gamma_l| \leq k_l$ **then**
- 18 Call the algorithm for Case (2) to find the center set C' on $\Gamma_l \cup \Gamma_{3-l}$ with $k'_l = k_l - |C \cap S_l|$.
- 19 Return $C \cup C'$.

Lemma 2. Each pair of points from the same Γ_l belongs to different clusters of the optimum solution when $\lambda = 2r^*$ before commencing of Alg. 1.

Proof. When $\lambda = 2r^*$, according to Def. 1, the distance between each pair of points in Γ_l is larger than $2r^*$. Suppose there exists a pair of points i, j in Γ_l such that they are in the same cluster of the optimum solution. That is, there exists a center c^* in the center set of the optimum solution such that $d(i, c^*) \leq r^*$ and $d(j, c^*) \leq r^*$ both hold. So $d(i, j) \leq d(i, c^*) + d(j, c^*) \leq r^* + r^* = 2r^*$, contradicting $d(i, j) > 2r^*$. \square

Lemma 3. When Phase 1 of Alg. 1 completes, $|C \cap S_l| \leq k_l$ holds for the current center set C .

Proof. Following Steps 3-6 in Alg. 1, we add $i \in G$ with degree 0 to C . That is, each point $i \in \Gamma_l$ added to C satisfies $d(i, \Gamma_{3-l}) > 3r^*$. Then for any point $j \in S_{3-l}$, we have $d(j, \Gamma_{3-l}) \leq 2r^*$, and hence $d(i, j) > r^*$ holds. That is, each point $i \in C \cap \Gamma_l$ cannot belong to a cluster centered at a point of S_{3-l} in the optimal solution. On the other hand, each pair of points $i, j \in C$ belongs to different optimal clusters because $d(i, j) > 2r^*$ (similar to the proof of Lem. 2). Therefore, we have $|C \cap S_l| \leq k_l$ for every l . \square

For brevity, Γ_l^j and C^j respectively denote Γ_l and C in the j th iteration of the while-loop (Phase 2) of Alg. 1.

Lemma 4. In the j th iteration of the while loop of Alg. 1, we have: (1) $|\Gamma_l^j| \leq k - |C^j|$; (2) $|C^j| \leq k$; (3) $|C^j \cap S_l| \leq k_l$.

Proof. For Cond. (1), we first show it is true when $j = 1$. Before the first while loop commences, $|\Gamma_l^1| \leq k - |C^1|$ is satisfied. Suppose otherwise, there would exist a contradiction as analyzed below. We will show the distance between every two points belonging to $C^1 \cup \Gamma_l^1$ is larger than $2r^*$, so every two points therein cannot belong to the same cluster in an optimal solution, and hence there are at least $|\Gamma_l^1| + |C^1| > k$ centers in an optimal solution, leading to a contradiction. This can be deduced following Steps 3-6 in Alg. 1, where points with a zero degree are added to C^1 for the first iteration. For each pair of points $i \in C^1$ and $j \in \Gamma_l^1$, the absence of edge $e(i, j)$ in G indicates $d(i, j) > 3r^*$ according to the construction of the auxiliary graph G . Moreover, for each two points $i, j \in \Gamma_l^1$, $d(i, j) > 2r^*$ holds because Γ_l is a $2r^*$ -independent center set for S_l ($l \in \{1, 2\}$), and i, j are points in initial Γ_l . Lastly, for each $i, j \in C^1$, $d(i, j) > 2r^*$ holds because of Step 4 of the algorithm. Therefore, $d(i, j) > 2r^*$ holds for each two points $i, j \in C^1 \cup \Gamma_l^1$.

We demonstrate that Cond. (1) holds for the $(j + 1)$ th iteration by induction. Assume that Cond. (1) is valid for the j th iteration concerning Γ^j . Then, we either remove two points or remove a point i along with its degree-1 neighbors. Consequently, after this removal, it follows that:

$$|\Gamma_l^{j+1}| \leq |\Gamma_l^j| - 1 \leq k - |C^j| - 1 \leq k - |C^{j+1}|,$$

where the first inequality is due to the removal of at least one point from both Γ_l^j and Γ_{3-l}^j in every iteration. The second inequality is derived from our inductive assumption. The third inequality arises because, on one hand, we add one point to C^j to form C^{j+1} , and on the other hand, according to the algorithm, the removal of points from G does not result in new points of degree 0 in the auxiliary graph G . Hence, Cond. (1) is also true for the $(j + 1)$ th iteration.

Cond. (2) can be immediately derived from Cond. (1) because $|\Gamma_l^j| \geq 0$ holds.

Moving on to Cond. (3), Lem. 3 establishes that in Steps 3-6 of Alg. 1, the inequality $|C^1 \cap S_l| \leq k_l$ is true. That is, Cond. (3) holds for $j = 1$. Assume that Cond. (3) holds for j th iteration. We need only to verify that this inequality holds in the $(j + 1)$ th iteration of Alg. 1. According to Steps 8-15, Alg. 1 involves adding a single point to C in each iteration. So if $|C^j \cap S_l| < k_l$ holds for any $l \in \{1, 2\}$, then $|C^{j+1} \cap S_l| \leq k_l$. Otherwise (i.e., $|C^j \cap S_l| = k_l$), we have $|\Gamma_{3-l}^j| + |C^j| \leq k$ by Cond. (1). This implies

$$|\Gamma_{3-l}^j| + |C^j \cap S_{3-l}| \leq k - |C^j \cap S_l| = k - k_l = k_{3-l}.$$

According to the *if* condition in Step 16 of Alg. 1, when the above inequality holds, Alg. 1 terminates and reduces to Case (2). This completes the proof. \square

Theorem 2. Alg. 1 terminates in time $O(k^2)$ and outputs a feasible solution C to fair k -center, for which $d(s, C) \leq 5r^*$ holds for any point $s \in S$.

Proof. For the runtime, Step 2 of Alg. 1 takes $O(k^2)$ time to construct G as G contains at most $O(k)$ vertices and $O(k^2)$

edges. Then, the while-loop (Steps 7-18) iterates for at most $O(k)$ times as it removes at least one point from G in each iteration, and each iteration takes $O(k)$ time. Therefore, the total runtime sums up to $O(k^2)$.

From Cond. (2) and (3) of Lem. 4, we immediately get $|C| \leq k$ and $|C \cap S_l| \leq k_l$, which indicates C is a feasible solution. It remains to bound the distance from any point $s \in S$ to C . For any $s \in S_l$, a point $i \in \Gamma_l$ must exist such that $d(s, i) \leq 2r^*$ holds according to the definition of Γ_l . If $i \in C$ holds, then we have $d(s, C) \leq 2r^*$. Otherwise, i.e. $i \notin C$, then there must exist an edge $e(i, j)$ with $j \in \Gamma_{3-l}$ in the auxiliary graph. According to Steps 8-15 in Alg. 1, at least one endpoint of $e(i, j)$, either i or j , must be added to C . Since $i \notin C$ by assumption, $j \in C$ holds. Moreover, the existence of edge $e(i, j)$ also means $d(i, j) \leq 3r^*$. So by triangle inequality, we have

$$d(s, C) \leq d(s, j) \leq d(s, i) + d(i, j) \leq 2r^* + 3r^* = 5r^*.$$

Therefore, regardless of whether $i \in C$ holds or not, $d(s, C) \leq 5r^*$ holds for any $s \in S_l$ for any $l \in \{1, 2\}$. \square

Combining Cases (1), (2) and (3) with the streaming stage, we eventually achieve a complete streaming algorithm for fair k -center provided that r^* is known. For the space complexity, the algorithm stores $m = 2$ independent center sets for each case where each independent center set has $O(k)$ points, so it consumes a space complexity of $O(k)$. However, as r^* is actually unknown, we need an $O(\log \alpha)$ multiplicative factor over the memory complexity, where α is the aspect ratio that is the ratio between the maximum and minimum distance between the points of S . Combining Thm. 2, we have:

Corollary 1. *When r^* is unknown, fair k -center admits a streaming algorithm with an approximation ratio $5 + \epsilon$ and a memory complexity $O(k \log \alpha)$.*

Notably, we can extend the algorithms respectively for Cases (2) and Cases (3) and obtain an algorithm for general m (See the extended algorithm in the full version).

4 Streaming Semi-structured Datasets

In this section, we consider scenarios in which the data points are streamed following the demographic group order, i.e. all points belonging to S_l arrive before S_{l+1} , $\forall l \in [m - 1]$. We devise an algorithm for $m = 2$, achieving a ratio of $3 + \epsilon$ that almost matches the state-of-the-art ratio of 3 for the offline setting. The algorithm can be extended to achieve a ratio 4 for general m as shown in the full version.

Our key idea of the improved algorithm for $m = 2$ is to obtain an extra point set Γ_{sub} in the streaming stage and use some of its points as centers for points in Γ_1 . More precisely, we first construct an independent set $\Gamma'_1 = \Gamma_1$ with $\lambda = 2r^*$ upon the data stream of S_1 . Then, upon the stream of S_2 , we select points using different designated approaches depending on whether $|\Gamma'_1| \leq k_1$ holds. As the constructed independent center sets are slightly different, we adopt notation Γ'_l instead of Γ_l . The detailed algorithm is as in Alg. 2.

Then we prove that with only the representative points of $\Gamma'_1 \cup \Gamma'_2$ and Γ_{sub} , our post-streaming algorithm can compute an approximation solution with ratio 3.

Algorithm 2: Streaming semi-structured data

Input: A stream of points $S = \bigcup_l S_l$ in which all points of S_1 arrive before the points of S_2 , and $\lambda = 2r^*$.
Output: The sets of representative points.

- 1 Set $\Gamma_{sub} = \emptyset$ and $\Gamma'_l = \emptyset$ for $l = 1, 2$;
- 2 **upon** each arriving point $i \in S_1$ **do**
- 3 **if** $d(i, \Gamma'_1) > \lambda$ **then**
- 4 Set $\Gamma'_1 \leftarrow \Gamma'_1 \cup \{i\}$; // Recall that $d(i, \emptyset) = \infty$.
- 5 **upon** each arriving point $i \in S_2$ **do**
- 6 **if** $|\Gamma'_1| \leq k_1$ **then**
- 7 **if** $d(i, \Gamma'_1) > 3\lambda/2$ and $d(i, \Gamma'_2) > \lambda$ **then**
- 8 Set $\Gamma'_2 \leftarrow \Gamma'_2 \cup \{i\}$;
- 9 **else**
- 10 **if** $d(i, \Gamma'_1 \cup \Gamma'_2) > \lambda$ **then**
- 11 Set $\Gamma'_2 \leftarrow \Gamma'_2 \cup \{i\}$;
- 12 **if** there exists a point $j \in \Gamma'_1$ that has no replacement in Γ_{sub} and $d(i, j) \leq \lambda/2$ **then**
- 13 Set $\Gamma_{sub} \leftarrow \Gamma_{sub} \cup \{i\}$;
- 14 **Return** Γ'_1, Γ'_2 , and Γ_{sub} .

Lemma 5. *For Γ'_1 and Γ'_2 produced by Alg. 2, we have: (1) $|\Gamma'_1| + |\Gamma'_2| \leq k$; (2) $|\Gamma'_2| \leq k_2$; (3) There exists a subset $\Gamma''_1 \subseteq \Gamma'_1$ and accordingly a subset $\Gamma'_{sub} = \{\sigma(c) | c \in \Gamma''_1\} \subseteq \Gamma_{sub}$, such that $|\Gamma'_1 \setminus \Gamma''_1 \cup \Gamma'_{sub} \cap S_1| \leq k_1$ holds, where $\sigma(c)$ is the stored replacement of c in Γ_{sub} with $d(c, \sigma(c)) \leq r^*$.*

Proof. For Cond. (1), according to the algorithm, we have $d(p, q) > 2r^*$ for any pair of two points $p, q \in \bigcup_l \Gamma'_l, l = 1, 2$. So every pair of points of $\bigcup_l \Gamma'_l$ must appear in two different clusters in the optimal solution similar to the proof of Lem 2. Thus, $|\bigcup_{l=1}^2 \Gamma'_l| \leq k$ holds.

For Cond. (2), if $|\Gamma'_1| > k_1$, we have $|\Gamma'_2| \leq k - |\Gamma'_1| < k - k_1 = k_2$. Otherwise (i.e., $|\Gamma'_1| \leq k_1$), as Γ'_1 covers all points of the clusters centered at points of S_1 in the optimal solution within a radius $3r^*$. Then, the remaining points to be covered in the algorithm appear in only the optimal clusters centered at points of S_2 , which are at most k_2 clusters. On the other hand, every pair of points in Γ'_2 must appear in different optimal clusters. So $|\Gamma'_2| \leq k_2$.

For Cond. (3), if $|\Gamma'_1| \leq k_1$ holds after streaming then the proof is done. So we need only to prove the case for $|\Gamma'_1| > k_1$. Clearly, there exists a subset of Γ'_1 that contains at least $|\Gamma'_1| - k_1$ points, say Γ''_1 , appearing in optimal clusters centered at points of S_2 . Clearly, there exists at least one point $j \in S_2$ for each point $i \in \Gamma''_1$ with $d(i, j) \leq r^*$. Moreover, according to Alg. 2, we add such point $\sigma(i) = j$ to Γ_{sub} as a replacement point of i , collectively composing the set Γ'_{sub} . That is, $\Gamma'_1 \setminus \Gamma''_1 \cup \Gamma'_{sub}$ feasibly covers all points of Γ'_1 with radius r^* , and contains at most k_1 points of Γ'_1 . \square

Following the above lemma, we can immediately obtain a 3-approximation solution by simply using $C = \Gamma'_2 \cup (\Gamma'_1 \setminus \Gamma''_1 \cup \Gamma'_{sub})$ as the desired center set.

5 Experimental Results

In this section, we conduct an empirical evaluation of our algorithms utilizing both simulated and real-world datasets, compared with three previous approximation algorithms that serve as baselines. All experiments are averaged over at least 20 iterations and are conducted on a Linux machine equipped with a 12th Gen Intel(R) Core(TM) i9-12900K CPU and 32 GB of RAM using Python 3.8. A detailed description of the experimental setup is provided in follows.

5.1 Experimental Setting

Datasets. We employ both simulated and real-world datasets to evaluate our approximation algorithm. We summarize the detailed datasets in Tab. 1.

Real-world datasets. We apply our algorithms to seven real-world datasets: Wholesale (Cardoso 2013), Student (Cortez 2008), Bank (Moro, Rita, and Cortez 2014), CreditCard (Yeh 2009), Adult (Becker and Kohavi 1996), SushiA and CelebA (Liu et al. 2015), following the most recent related works (Jones, Nguyen, and Nguyen 2020; Chen et al. 2019). Consistent with previous studies, we utilize meaningful numerical features for clustering and incorporate selected binary categorical attributes to construct datasets with fair constraints across all datasets.

Simulated datasets. We provide two datasets that serve as benchmarks: the simulated dataset provides an exact optimal solution, while the large-scale dataset incurs more file I/O overhead, making it ideal for measuring computational efficiency. First, the dataset (called *SimuA*) is used to assess the *empirical approximation ratio* of our algorithms and the baseline methods. Inspired by previous research (Kleindessner, Awasthi, and Morgenstern 2019), we use their method to construct a simulated dataset with a known optimal solution for the fair k -center problem. Secondly, we leverage the implementation from Chiplunkar, Kale, and Ramamoorthy (2020) to generate a *100 GB* dataset (called *SimuB*), which allows us to evaluate the runtime performance of different algorithms.

Constraint Settings. For the simulated dataset, we configured the parameters to compare both the average and maximum approximation ratios for these algorithms and to verify the approximation ratio of our algorithms. To ensure fair center selection, we aligned the number of required centers from each group to be proportional to the size of that group. Following the fairness principle of disparate impact as outlined by Feldman et al. (2015), we restricted the selection to k_l data points from the l th group to serve as centers. We then evaluated the clustering quality by varying the number of k clusters across these datasets.

Algorithms. In comparison with existing streaming fair k -center clustering algorithms, this experiment includes an extended algorithm from the offline 3-approximation algorithm by Jones, Nguyen, and Nguyen (2020) (denoted as *ExJones*), a two-pass streaming 3-approximation algorithm by Chiplunkar, Kale, and Ramamoorthy (2020) (denoted as two-pass *3-Approx*), and a one-pass streaming 7-approximation algorithm (Lin, Guo, and Jia 2024) (denoted

Dataset	#Records	#Dim.
Wholesale	440	6
Student	649	16
Bank	4,521	7
SushiA	5,000	10
CreditCard	30,000	19
Adult	32,561	6
CelebA	202,599	15,360
SimulatedA	4,000,000	1,000
SimulatedB	[2k,4k,6k,8k,10k]	2

Table 1. Datasets Summary.

as *7-Approx*). Moreover, we propose two algorithms in this paper: 1) a one-pass 5-approximation algorithm (denoted as *5-Approx*); 2) a one-pass 3-approximation algorithm with $m = 2$ (denoted as *3-Approx*) for semi-structured data streams in our experiments.

Metrics. To evaluate clustering quality, we use the *cost* metric defined in Sec. 2 and compare its average value across datasets. We also report runtime, measured in seconds.

5.2 Experimental Analysis

In Fig. 1, we first report the empirical approximation ratios of the algorithms on a simulated dataset with known optimal solutions. This allows us to compare the experimental objective values of our algorithms and the baselines against the optimal objective. We then evaluate the clustering costs of these algorithms on real-world datasets. Finally, we report the runtime(s) performance on large-scale simulated datasets, focusing on file I/O operations in our algorithms compared to other baselines.

Approximation Factor. We compare algorithm performance by evaluating the relative solution ratio against the optimal solution provided for the simulated dataset. The ratio of the evaluation result can be called the empirical approximation ratio, and the maximum value represents the worst-case cost in all the experimental results. We run the code of constructing the simulated dataset shared by Kleindessner, Awasthi, and Morgenstern (2019) for their algorithm setting $m = 2$, $k = 100$ and varying the size of the simulated datasets $|S|$ from 2,000 to 10,000. For each size of the dataset, we perform 10 runs on 10 kinds of fairness with 10 different constraint ratios.

We observe that the practical performance of all algorithms aligns well with their theoretical bounds. Notably, the 3-Approx (semi) demonstrates the best performance, even outperforming the two-pass 3-Approx algorithm (Chiplunkar, Kale, and Ramamoorthy 2020). When the dataset size $|S| = 2,000$, the maximal empirical approximation ratio of 7-Approx is greater than 5, validating the suitability of the dataset for testing these algorithms. Compared with the method of Jones et al., our algorithm achieves a lower empirical approximation ratio, demonstrating the advantage of our algorithm. Consequently, our algorithms exhibit superior performance on the datasets, conforming to the theoretical guarantees.

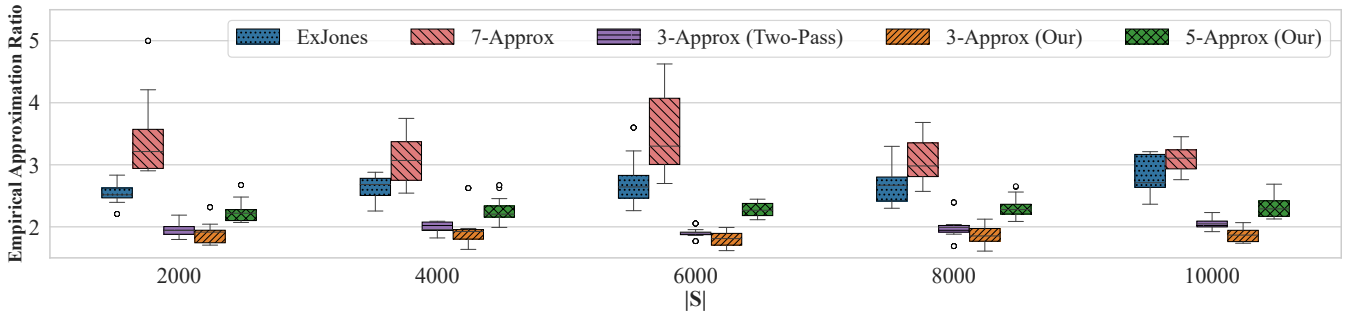


Figure 1. Empirical approximation ratio ($cost/r^*$) of our algorithms in comparison with other baselines.

Dataset	ExJones	7-Approx	5-Approx	3-Approx (Semi)	3-Approx (Two pass)
Wholesale	1.36	1.28	<u>1.04</u>	0.84	0.61
Student	<u>1.85</u>	1.85	1.90	1.79	1.73
CelebA	24.36	26.27	<u>21.34</u>	21.34	21.25
Bank	<u>0.49</u>	0.93	0.61	0.40	0.43
SushiA	1.42	1.66	<u>1.41</u>	1.51	1.48
Credit	<u>0.93</u>	1.89	1.03	0.94	0.81
Adult	0.65	1.08	<u>0.62</u>	0.52	0.56

Table 2. Cost comparison on the real-world datasets. (The underline values highlight the best results of the general one-pass algorithms. In addition, the bold indicates the best result when semi-structured data streaming is included alongside the one-pass streaming algorithms.)

Comparison with Baselines. We report the clustering costs achieved by our algorithms and three baselines across seven real-world datasets in Tab. 2. The parameter k is set proportionally to the dataset size: datasets above the dividing line use 1% of the total number of records, while those below use 1‰, which depends on the variation in dataset scales. This same proportion is also applied to the center constraints across different groups. All datasets are normalized prior to running the algorithms, and for CelebA, the first 1,000 samples are used in the experiments.

In Tab. 2, we observe that 3-Approx (Semi) outperforms most of the other one-pass algorithms, while the two-pass 3-approximation algorithm maintains better performance than the one-pass streaming algorithms on the seven datasets. We attribute this to the fact that our algorithm is a modification of the original center selection method based on streaming. By combining streaming techniques with the structure of an *independent center set*, it effectively identifies more meaningful center points. As a result, it achieves lower clustering costs than the other baselines under the one-pass streaming setting (shown as the underline values). We also observe that the results reported by (two-pass) 3-Approx consistently outperform all the one-pass streaming algorithms. This can be explained by the fact that the two-pass 3-approximation algorithm has the advantage of refining the center set in the second pass, enabling it to identify more accurate center points.

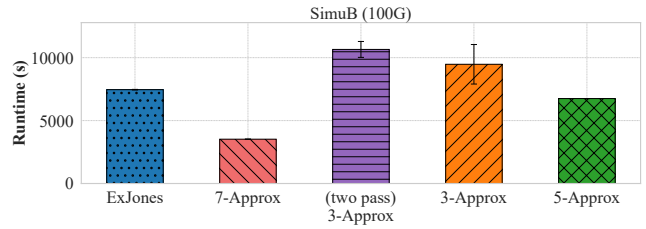


Figure 2. Runtime on the 100G simulated dataset.

Streaming Runtime. Using the method from Chiplunkar, Kale, and Ramamoorthy (2020), we generate a 100 GB dataset containing 4,000,000 items and 1,000 features. In Fig. 2, we evaluate the runtime performance of our algorithms and baseline methods on this dataset. Our results show that the 5-Approx algorithm runs faster than the algorithms of Chiplunkar, Kale, and Ramamoorthy and Jones, Nguyen, and Nguyen (2020), which aligns with theoretical expectations, as all of our algorithms are one-pass streaming methods. Among these methods, 7-Approx (Lin, Guo, and Jia 2024) runs the fastest, because it selects fewer points during the streaming process and omits post-processing steps, thereby reducing computational overhead.

6 Conclusion

In this paper, we first devise a one-pass streaming algorithm with an approximation ratio of $5 + \epsilon$ and a memory complexity of $O(k \log \alpha)$, where α is the aspect ratio. Motivated by the broad applications of semi-structured data streams, we then present an algorithm with a ratio of $(3 + \epsilon)$ and the same memory complexity for the problem on semi-structured streams with $m = 2$. This $(3 + \epsilon)$ -approximation can be directly used to solve the offline problem and achieve a slightly improved ratio of 3, matching the state-of-the-art ratio for offline fair k -center. Lastly, we conduct extensive experiments to demonstrate the performance gains of our algorithms compared to baselines, including state-of-the-art methods. In the future, we will strive to design both offline and streaming algorithms with better approximation ratios for fair k -center, k -median, and k -means in doubling metrics and Euclidean spaces.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No. 12271098) and Key Project of the Natural Science Foundation of Fujian Province (No. 2025J02011).

References

- Anegg, G.; Vargas Koch, L.; and Zenklusen, R. 2022. Techniques for Generalized Colorful k -Center Problems. In *Proceedings of the 30th Annual European Symposium on Algorithms (ESA-22)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
- Angelidakis, H.; Kurpisz, A.; Sering, L.; and Zenklusen, R. 2022. Fair and Fast k -Center Clustering for Data Summarization. In *Proceedings of the 39th International Conference on Machine Learning (ICML-22)*, 669–702. PMLR.
- Backurs, A.; Indyk, P.; Onak, K.; Schieber, B.; Vakilian, A.; and Wagner, T. 2019. Scalable fair clustering. In *Proceedings of the 36th International Conference on Machine Learning (ICML-19)*, 405–413. PMLR.
- Bandyapadhyay, S.; Inamdar, T.; Pai, S.; and Varadarajan, K. 2019. A Constant Approximation for Colorful k -Center. *Leibniz International Proceedings in Informatics, LIPIcs*, 144.
- Becker, B.; and Kohavi, R. 1996. Adult. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5XW20>.
- Cardoso, M. 2013. Wholesale customers. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5030X>.
- Ceccarelo, M.; Pietracaprina, A.; and Pucci, G. 2019. Solving k -center Clustering (with Outliers) in MapReduce and Streaming, almost as Accurately as Sequentially. *Proceedings of the VLDB Endowment*, 12(7): 766–778.
- Ceccarelo, M.; Pietracaprina, A.; and Pucci, G. 2024. Fast and accurate fair k -center clustering in doubling metrics. In *Proceedings of the 33rd ACM Web Conference (WWW-24)*, 756–767.
- Chen, D. Z.; Li, J.; Liang, H.; and Wang, H. 2016. Matroid and knapsack center problems. *Algorithmica*, 75: 27–52.
- Chen, X.; Fain, B.; Lyu, L.; and Munagala, K. 2019. Proportionally fair clustering. In *Proceedings of the 36th International Conference on Machine Learning (ICML-19)*, 1032–1041. PMLR.
- Chen, X.; Ji, S.; Wu, C.; Xu, Y.; and Yang, Y. 2024. An approximation algorithm for diversity-aware fair k -supplier problem. *Theoretical Computer Science*, 983: 114305.
- Chierichetti, F.; Kumar, R.; Lattanzi, S.; and Vassilvitskii, S. 2017. Fair clustering through fairlets. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS-17)*, 5036–5044.
- Chiplunkar, A.; Kale, S.; and Ramamoorthy, S. N. 2020. How to solve fair k -center in massive data models. In *Proceedings of the 37th International Conference on Machine Learning (ICML-20)*, 1877–1886.
- Cortez, P. 2008. Student Performance. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5TG7T>.
- Feldman, M.; Friedler, S. A.; Moeller, J.; Scheidegger, C.; and Venkatasubramanian, S. 2015. Certifying and removing disparate impact. In *proceedings of the 21st ACM SIGKDD international conference on knowledge discovery and data mining (KDD-15)*, 259–268.
- Guo, L.; Jia, C.; Liao, K.; Lu, Z.; and Xue, M. 2025. Near-Optimal Algorithms for Instance-Level Constrained k -Center Clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 36(10): 18844–18858.
- Jia, X.; Sheth, K.; and Svensson, O. 2022. Fair colorful k -center clustering. *Mathematical Programming*, 192(1-2): 339–360.
- Jones, M.; Nguyen, H.; and Nguyen, T. 2020. Fair k -centers via maximum matching. In *Proceedings of the 37th International Conference on Machine Learning (ICML-20)*, 4940–4949. PMLR.
- Kale, S. 2019. Small Space Stream Summary for Matroid Center. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*, 145: 20.
- Kleindessner, M.; Awasthi, P.; and Morgenstern, J. 2019. Fair k -center clustering for data summarization. In *Proceedings of the 36th International Conference on Machine Learning (ICML-19)*, 3448–3457. PMLR.
- Lin, Z.; Guo, L.; and Jia, C. 2024. Streaming Fair k -Center Clustering over Massive Dataset with Performance Guarantee. In *Proceedings of the 28th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-24)*, 105–117. Springer.
- Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of the 15th International Conference on Computer Vision (ICCV-15)*.
- Mahabadi, S.; and Trajanovski, S. 2024. Core-sets for Fair and Diverse Data Summarization. *Advances in Neural Information Processing Systems*, 36.
- Matthew McCutchen, R.; and Khuller, S. 2008. Streaming Algorithms for k -Center Clustering with Outliers and with Anonymity. *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, 165–178.
- Moro, S.; Rita, P.; and Cortez, P. 2014. Bank Marketing. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5K306>.
- Nguyen, H. L.; Nguyen, T.; and Jones, M. 2022. Fair Range k -center. *arXiv preprint arXiv:2207.11337*.
- Wu, X.; Feng, Q.; Huang, Z.; Xu, J.; and Wang, J. 2024. New Algorithms for Distributed Fair k -Center Clustering: Almost Accurate as Sequential Algorithms. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS-24)*, 1938–1946.
- Yeh, I.-C. 2009. Default of Credit Card Clients. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C55S3H>.
- Zhang, H.; Wu, B.; Yang, X.; Yuan, X.; Liu, X.; and Yi, X. 2025. Dynamic Graph Unlearning: A General and Efficient Post-Processing Method via Gradient Transformation. In *Proceedings of the ACM on Web Conference 2025*, 931–944.

Zhang, H.; Wu, B.; Yuan, X.; Pan, S.; Tong, H.; and Pei, J. 2024. Trustworthy Graph Neural Networks: Aspects, Methods, and Trends. *Proc. IEEE*, 112(2): 97–139.

Zhang, H.; Yuan, X.; and Pan, S. 2024. Unraveling Privacy Risks of Individual Fairness in Graph Neural Networks. In *Proceedings of the 40th IEEE International Conference on Data Engineering (ICDE-24)*, 1712–1725. IEEE.