# Hausdorff Edit Distance

Jonathan James Perry[*] Benjamin Raichel[*]

## Abstract

The Hausdorff distance is a standard measure of similarity between two finite point sets. Here we introduce the Hausdorff Edit Distance problem, where given a parameter $k$, the goal is to make up to $k$ edits, either insertions or deletions, to the point sets so as to minimize their Hausdorff distance. When only deletions are allowed the problem is polynomial time solvable, though the problem is APX-Hard when insertions are allowed, due to its connection with $k$-center clustering. By reducing to repeated calls to any approximate $k$-center clustering algorithm, we show how to achieve similar approximation factors and running times for variants of the Hausdorff Edit Distance problem.

## 1 Introduction

**Hausdorff Distance.** Given two finite point sets $P$ and $Q$ in a metric space, the Hausdorff distance $d_{\mathcal{H}}(P,Q)$ is a natural and common way of measuring the similarity between $P$ and $Q$. Define the one sided Hausdorff distance, $d_h(P,Q)$, as the maximum distance of a point in $P$ from its nearest point in $Q$, that is $d_h(P,Q) = \max_{p\in P} \|p-Q\|$ (where $\|\cdot\|$ denotes the metric over the points). Then the standard (two sided) Hausdorff distance is $d_{\mathcal{H}}(P,Q) = \max\{d_h(P,Q), d_h(Q,P)\}$. Clearly the Hausdorff distance can be computed in quadratic time by looking at all pairwise distances, though using Voronoi diagrams in the plane it can be computed in $O(N \log N)$ time where $N = \max\{|P|,|Q|\}$. Using an approximate nearest neighbor data structure for points in $\mathbb{R}^d$ for any constant $d$, one can get $(1+\varepsilon)$-approximation in $O(N\log N + N/\varepsilon^d)$ time [17].

The Hausdorff distance is well known to be sensitive to outliers, as even a single outlier point can arbitrarily increase the Hausdorff distance. One way to address this is to use the *Partial Hausdorff Distance* introduced in [19]. Given parameters $k$ and $\ell$, the $k,\ell$ Partial Hausdorff Distance is the maximum of the $k$th ranked value in $\{\|p-Q\| \mid p \in P\}$ and the $\ell$th ranked value in $\{\|q-P\| \mid q \in Q\}$. Alternatively, one can consider the RMS Hausdorff distance, where rather than taking the maximum (or $k$th ranked) nearest neighbor distance, one instead takes the sum of the squares of all nearest neighbor distances, see [2, 4] and references therein. These and many other prior works also minimize the Hausdorff distance or other measures under translation (or other transformations), though a discussion of this topic is outside the scope of the current paper.

**Edit Distance.** In this paper we consider the case that the point sets $P$ and $Q$ may be faulty, and our goal is to make up to $k$ edits to $P$ or $Q$ so as to minimize their Hausdorff distance. Here an edit will either be an insertion or a deletion of a point from either $P$ or $Q$. When only deletions are allowed, this relates to the Partial Hausdorff Distance discussed above, as well as other outlier problems such as clustering with outliers discussed below. For geometric problems on unordered point sets, the more general case where insertions are allowed appears to be less well studied, and indeed may not make sense for many problems, such as center based clustering. However, for ordered point sets there are several relevant papers (as indeed such cases bear more resemblance to classic string edit distance).

The Fréchet distance is a standard measure of similarity between polygonal curves (i.e. ordered point sets). [13] defined and gave various results for the Fréchet edit distance, where the goal is to make up to $k$ insertions or deletions so as to minimize the Fréchet distance (prior works considered shortcutting, which relates to the deletion only case). In the geometric edit distance problem [1, 14, 15], given two ordered point sequences, the goal is to find a monotone matching of sequences which minimizes the sum of the distances of the matched points plus a penalty on the number of unmatched points, i.e. arguably further from our notion of edit than [13]. Finally, [12] considers the graph edit distance problem, which requires modifying edges and vertices and gets even further from our problem. In so doing, they define a heuristic approach which they call Hausdorff Edit Distance. This measure, however, differs dramatically from ours, in that it is a measure between graphs, and is a summed distance over all matched vertices and edges, and thus is only mentioned as it shares the same name.

**Clustering.** Insertions in the Hausdorff Edit Distance problem are closely related to the standard $k$-center clustering problem. Here one is given a set $P$ of $n$ points

[*] Department of Computer Science, University of Texas at Dallas, USA {jperry,benjamin.raichel}@utdallas.edu. Work on this paper was partially supported by NSF AF Award CCF-2311179.

from a metric space, and the goal is to select a set $C$ of $k$ *center* points from the metric space, so as to minimize the maximum distance of any point in $P$ to its nearest center in $C$. This problem is known to be APX-hard. Specifically, unless P=NP, for general metric spaces the problem cannot be approximated with any factor less than 2 [18], and even in the plane it remains hard to approximate with a factor of roughly 1.82 [11]. Conversely, the standard greedy algorithm of Gonzalez [16] yields a 2-approximation in any metric space. When $P \subset \mathbb{R}^d$ for constant $d$, [3] gave an $O(n \log k) + (k/\varepsilon)^{O(k^{1-1/d})}$ time $(1+\varepsilon)$-approximation.

When we separate the deletion and insertion budgets, Hausdorff Edit Distance then closely relates to $k$-center clustering with outliers, where given a number $0 \leq \ell \leq n$, you are allowed to choose some $\ell$ points from $P$ that are not required to be covered by the centers. [7] initiated the formal study of $k$-center clustering with outliers and gave a simple greedy 3-approximation for any metric on the point set $P$. This was subsequently improved to a 2-approximation in [6] using LP based methods. As pointed out in [10], these results assume the metric consists only of the point set $P$, as opposed to allowing $P$ to be a subset from some larger metric space from which the centers $C$ can be chosen (as we do in the current paper, which captures for example the Euclidean case). By the triangle inequality, however, these results would still imply constant factor approximations for the case where $P$ is a subset from some larger metric space. Many other variants, including bi-criteria approximations, streaming, and dynamic variants have been considered [5, 9, 10].

**Our Contributions.** This paper introduces the Hausdorff Edit Distance problem, where insertions and deletions are allowed. As remarked above, the deletion only case relates to the previously defined Partial Hausdorff Distance problem, though allowing insertions is new, and we believe is a valuable addition given the ubiquity of the Hausdorff distance. Our problem closely relates to $k$-center clustering, which we directly use to achieve the following results. Below, $P$ is a set of $n$ points, $Q$ is a set of $m$ points, $k$ and $\ell$ are integer parameters, and $N = \max\{m, n\}$.

- In Section 3, the Hausdorff Edit Distance problem is first shown to be APX-hard to compute. Conversely, our main result shows that given any $\alpha$-approximation to $k$-center with run time $T(n, k)$, then for any constant $\varepsilon > 0$, one can compute an $(\alpha + \varepsilon)$-approximation to the Hausdorff Edit Distance in $O\big((mn + k^2N)\log(N) + k \cdot T(N, k)\big)$ time.
- Using the standard greedy 2-approximation for $k$-center, our main result implies a $(2+\varepsilon)$-approximation for Hausdorff Edit Distance in $O\big((mn + k^2N)\log(N)\big)$ time. In Appendix A.1, we argue that for the Euclidean case where $P, Q \subset \mathbb{R}^d$ for constant $d$, this can be improved to either a $(2+\varepsilon)$-approximation in $O\big(k^2N\log(N)\big)$ time or a $(1+\varepsilon)$-approximation in $O\big(k^2N\log(N)\big) + (k/\varepsilon)^{O(k^{1-1/d})}$ time.
- In Section 3.1 we consider the case where we have separate insertion and deletion budgets, $k$ and $\ell$. This case is again APX-hard, and we show that given any $O(T(n, k, \ell))$ time $\alpha$-approximation to $k$-center clustering with $\ell$ outliers, then for any constant $\varepsilon > 0$, one can compute an $(\alpha + \varepsilon)$-approximation in $O\big((mn + k\ell \cdot T(\max\{m, n\}, k, \ell))\log(mn)\big)$ time.
- In Section 3.2 we consider the case where only deletions are allowed. This case is no longer APX-hard. We show that it can be solved exactly in $O(mn)$ time in general, exactly in $O(N \log N)$ time in the plane, and can be $(1+\varepsilon)$-approximated in $O(N \log N + N/\varepsilon^d)$ time for points in $\mathbb{R}^d$ for constant $d$. The deletion only variant is very similar to the previously defined Partial Hausdorff Distance, though unlike [19], we provide a formal algorithm and analysis, which both establishes the equivalence to our variant with split budgets, and allows for efficient computation in bounded dimensions.
- Finally, in Appendix A.4 we consider several natural variants of our problem, and remark how our techniques easily extend to these variants.

## 2 Preliminaries

Let $(X, \|\cdot\|)$ be a metric space with point set $X$ and metric $\|\cdot\|$, where for $p, q \in X$ we write $\|p-q\|$ to denote their distance.[1] Throughout, $(X, \|\cdot\|)$ will be any fixed ambient metric, and when we say we are given a finite point set $P$, it is inferred that $P \subseteq X$ and we are using the metric $\|\cdot\|$. We assume that $\|p-q\|$ takes constant time to compute (otherwise, our running times must be multiplied by the time it takes to compute $\|p-q\|$).

For finite point sets $P, Q$, let $\|P - Q\| = \min_{p\in P, q\in Q} \|p-q\|$, where for a single point $p$ we write $\|p-Q\| = \|\{p\} - Q\|$. For finite sets $P, Q$ we then define the one-sided Hausdorff Distance as

$$d_h(P, Q) = \max_{p\in P}\left(\min_{q\in Q} \|p-q\|\right) = \max_{p\in P} \|p-Q\|,$$

where if $P = \emptyset$ then $d_h(P, Q) = 0$, and if $P \neq \emptyset$ and $Q = \emptyset$ then $d_h(P, Q) = \infty$. The standard Hausdorff

[1] We use the standard Euclidean distance notation as our general metric distance notation, both for conceptual ease and to distinguish it from our Hausdorff $d_{\mathcal{H}}$ notation.

Distance is then the bi-directional extension

$$d_{\mathcal{H}}(P,Q) = \max\Big(d_h(P,Q), d_h(Q,P)\Big).$$

For finite point sets $P$ and $Q$, let their symmetric difference be denoted $P \oplus Q = \{x \mid (x \in P \land x \notin Q) \lor (x \in Q \land x \notin P)\}$. For a finite point set $P$, we refer to either an insertion of a point into $P$ or a deletion of a point from $P$ as an *edit*. Then for a finite set $P$ of $n$ points and any integer $0 \leq k \leq n$, let $\mathtt{E}(P,k)$ be the set of all sets produced by up to $k$ edits to $P$, that is $\mathtt{E}(P,k) = \{Q \mid |P \oplus Q| \leq k\}$.

The Hausdorff Edit Distance is then defined as

$$d^{\mathtt{E}}_{\mathcal{H}}(P,Q,k) = \min_{0\leq k'\leq k}\left(\min_{\substack{P'\in\mathtt{E}(P,k'),\\ Q'\in\mathtt{E}(Q,k-k')}}\Big(d_{\mathcal{H}}(P',Q')\Big)\right),$$

that is, the minimum possible Hausdorff Distance after up to $k$ edits are made between $P$ and $Q$. We write $\mathsf{real}(d^{\mathtt{E}}_{\mathcal{H}}(P,Q,k))$ to denote a pair $P' \in \mathtt{E}(P,k'), Q' \in \mathtt{E}(Q,k-k')$, for some $0 \leq k' \leq k$, realizing $d^{\mathtt{E}}_{\mathcal{H}}(P,Q,k)$.

Let $P' \in \mathtt{E}(P,k'), Q' \in \mathtt{E}(Q,k-k')$ for some $0 \leq k' \leq k$. For $\alpha \geq 1$, we refer to $P'$, $Q'$ as an *$\alpha$-approximation* to $d^{\mathtt{E}}_{\mathcal{H}}(P,Q,k)$ if $d_{\mathcal{H}}(P',Q') \leq \alpha \cdot d^{\mathtt{E}}_{\mathcal{H}}(P,Q,k)$. That is, limiting to $k$ edits is a hard constraint and the approximation is on the distance.

Let $\mathtt{I}(P,k) = \{Q \mid P \subseteq Q, |Q| - |P| \leq k\}$ and $\mathtt{D}(P,k) = \{Q \mid Q \subseteq P, |P| - |Q| \leq k\}$. Then one can analogously define the Hausdorff Insertion Distance $d^{\mathtt{I}}_{\mathcal{H}}(P,Q,k)$ or Hausdorff Deletion Distance $d^{\mathtt{D}}_{\mathcal{H}}(P,Q,k)$ by respectively replacing the $\mathtt{E}$ sets in the above definition with $\mathtt{I}$ or $\mathtt{D}$ sets.

We now define the standard $k$-center clustering objective. For a finite set $P$ and an integer $k \geq 0$ define

$$kcenter(P,k) = \min_{|C|\leq k}\max_{p\in P}\|p - C\|,$$

that is, we wish to cover the set $P$ with a set of centers $C$ so as to minimize the maximum distance of a point in $P$ to its nearest center in $C$. For $\alpha \geq 1$, we refer to a set $C'$, such that $|C'| \leq k$, as an $\alpha$-approximation to $kcenter(P,k)$ if $\max_{p\in P}\|p - C'\| \leq \alpha \cdot kcenter(P,k)$.

## 3 Hausdorff Edit Distance

In this section we first show that computing $d^{\mathtt{E}}_{\mathcal{H}}(P,Q,k)$ is APX-Hard, and then provide approximation algorithms for $d^{\mathtt{E}}_{\mathcal{H}}(P,Q,k)$ and its variants.[2] These results are achieved by making a connection between Hausdorff Edit Distance and $k$-Center Clustering, which becomes clear after first making the following observation:

**Lemma 1** *For any finite point sets $P, Q$ and integer $k \geq 0$, $d^{\mathtt{E}}_{\mathcal{H}}(P,Q,k) = d^{\mathtt{I}}_{\mathcal{H}}(P,Q,k)$.*

**Proof.** First, $d^{\mathtt{E}}_{\mathcal{H}}(P,Q,k) \leq d^{\mathtt{I}}_{\mathcal{H}}(P,Q,k)$ as they are minimization problems respectively over edits or insertions, where since an insertion is a type of edit, for any set $P'$ and integer $k' \geq 0$ we have $\mathtt{I}(P',k') \subseteq \mathtt{E}(P',k')$.

Now we argue $d^{\mathtt{E}}_{\mathcal{H}}(P,Q,k) \geq d^{\mathtt{I}}_{\mathcal{H}}(P,Q,k)$. So let $\{P',Q'\} = \mathsf{real}(d^{\mathtt{E}}_{\mathcal{H}}(P,Q,k))$ be the pair realizing the Hausdorff Edit Distance. If no deletions occurred, i.e. $P \subseteq P'$ and $Q \subseteq Q'$, then $P', Q'$ is a possibility for $d^{\mathtt{I}}_{\mathcal{H}}(P,Q,k)$ and so we are done. So suppose otherwise, and let $p$ be a point that was deleted from $P$, that is $p \in P \setminus P'$. Rather than deleting $p$ from $P$ we instead insert $p$ into $Q$, that is, let $\hat{P} = P' \cup \{p\}$ and $\hat{Q} = Q' \cup \{p\}$. We claim that $d_{\mathcal{H}}(\hat{P},\hat{Q}) \leq d_{\mathcal{H}}(P',Q')$. First, observe that as $p \in \hat{Q}$, $d_h(\hat{P},\hat{Q}) = \max_{x\in\hat{P}}\|x - \hat{Q}\| = \max_{x\in P'}\|x - \hat{Q}\| \leq \max_{x\in P'}\|x - Q'\| = d_h(P',Q')$. Similarly, as $p \in \hat{P}$, $d_h(\hat{Q},\hat{P}) = \max_{x\in\hat{Q}}\|x - \hat{P}\| = \max_{x\in Q'}\|x - \hat{P}\| \leq \max_{x\in Q'}\|x - P'\| = d_h(Q',P')$. Thus $d_{\mathcal{H}}(\hat{P},\hat{Q}) = \max\{d_h(\hat{P},\hat{Q}), d_h(\hat{Q},\hat{P})\} \leq \max\{d_h(P',Q'), d_h(Q',P')\} = d_{\mathcal{H}}(P',Q')$. This in turn implies $d^{\mathtt{E}}_{\mathcal{H}}(P,Q,k) \geq d^{\mathtt{I}}_{\mathcal{H}}(P,Q,k)$ as we then can replace all deletions from either set by corresponding insertions without increasing the cost. □

**Lemma 2** *For any finite point set $P$ and integer $k \geq 0$, $d^{\mathtt{E}}_{\mathcal{H}}(P,\emptyset,k) = kcenter(P,k)$.*

**Proof.** By Lemma 1, we have $d^{\mathtt{E}}_{\mathcal{H}}(P,\emptyset,k) = d^{\mathtt{I}}_{\mathcal{H}}(P,\emptyset,k)$, thus we will equivalently prove that $d^{\mathtt{I}}_{\mathcal{H}}(P,\emptyset,k) = kcenter(P,k)$. First, we show that $d^{\mathtt{I}}_{\mathcal{H}}(P,\emptyset,k) \geq kcenter(P,k)$.

For readability, we write "$\min\limits_{k',P',Q'}$" as shorthand for "$\min_{0\leq k'\leq k}\min_{P'\in\mathtt{I}(P,k'),Q'\in\mathtt{I}(\emptyset,k-k')}$" below.

$$\begin{aligned} d^{\mathtt{I}}_{\mathcal{H}}(P,\emptyset,k) &= \min_{k',P',Q'}\Big(d_{\mathcal{H}}(P',Q')\Big) \geq \min_{k',P',Q'}\Big(d_h(P',Q')\Big) \\ &= \min_{k',P',Q'}\left(\max_{p\in P'}\|p - Q'\|\right) \geq \min_{Q'\in\mathtt{I}(\emptyset,k)}\left(\max_{p\in P}\|p - Q'\|\right) \\ &= \min_{|Q'|\leq k}\left(\max_{p\in P}\|p - Q'\|\right) = kcenter(P,k) \end{aligned}$$

Now we argue $d^{\mathtt{I}}_{\mathcal{H}}(P,\emptyset,k) \leq kcenter(P,k)$. So let $C = \mathsf{real}(kcenter(P,k))$, and let $r = kcenter(P,k) = \max_{p\in P}\|p - C\|$. Thus $d_h(P,C) = \max_{p\in P}\|p - C\| = r$.

Suppose there exists some $c \in C$ such that $\|c - P\| > r$, then $\max_{p\in P}\|p - C \setminus c\| = \max_{p\in P}\|p - C\| = r$. Hence any such point can be deleted without affecting the quality of the $k$-center solution, and so we assume every point in $C$ has some point from $P$ within distance $r$, which implies $d_h(C,P) = \max_{c\in C}\|c - P\| \leq r$. Thus $d^{\mathtt{I}}_{\mathcal{H}}(P,\emptyset,k) \leq d_{\mathcal{H}}(P,C) = \max\{d_h(P,C), d_h(C,P)\} \leq r$. □

Computing $kcenter(P,k)$ is known to be an APX-Hard problem. Specifically, unless P=NP, in general metric spaces it is hard to approximate within less than

[2] For simplicity our results are presented in terms of computing the value $d^{\mathtt{E}}_{\mathcal{H}}(P,Q,k)$ (or related quantities), however, the proofs implicitly construct sets that achieve this value.

a factor of 2 [18], and even in the plane it is hard to approximate within a factor of roughly 1.8 [11]. Thus the above lemmas immediately imply the following.

**Theorem 3** *For any finite point sets $P, Q$ and integer $k \geq 0$, the problems of computing either $d_{\mathcal{H}}^{\mathtt{E}}(P, Q, k)$ or $d_{\mathcal{H}}^{\mathtt{I}}(P, Q, k)$ are APX-Hard. The problems remain APX-Hard even when $P, Q \subset \mathbb{R}^2$.*

We now provide an approximate decision algorithm for Hausdorff Edit Distance, capable of using any $\alpha$-approximation algorithm for $kcenter(P, k)$ as a subroutine, where $\alpha > 1$ is some constant. Let this subroutine be denoted $\mathbf{kcen}(P, k)$.

```
Algorithm 1: HausEdit($P, Q, k, r$)
Mark all $p \in P$ such that $\|p - Q\| \leq r$.
  Mark all $q \in Q$ such that $\|q - P\| \leq r$.
Create sets $P' \subseteq P$ and $Q' \subseteq Q$ by removing all
  marked points.
$\beta = \infty$
for $k' = 0$ to $k$ do
    $\beta = \min\{\beta, \max\{\mathbf{kcen}(P', k'), \mathbf{kcen}(Q', k-k')\}\}$
if $\beta \leq \alpha r$ then        // $\mathbf{kcen}(P, k)$ is an $\alpha$-approx
    return True
else
    return False
```

**Lemma 4** *Let $P$ be a set of $n$ points, $Q$ be a set of $m$ points, and let $k \geq 0$ be an integer. Let $\mathbf{kcen}(P, k)$ be an algorithm which returns an $\alpha$-approximation to $kcenter(P, k)$ in $O(T(n, k))$ time.[3] Then if $\mathbf{HausEdit}(P, Q, k, r)$ returns True then $r \geq d_{\mathcal{H}}^{\mathtt{E}}(P, Q, k)/\alpha$, and if $\mathbf{HausEdit}(P, Q, k, r)$ returns False then $r < d_{\mathcal{H}}^{\mathtt{E}}(P, Q, k)$. The running time of $\mathbf{HausEdit}(P, Q, k, r)$ is $O\big(mn + k \cdot T(\max\{m, n\}, k)\big)$.*

**Proof.** Recall by Lemma 1 that $d_{\mathcal{H}}^{\mathtt{E}}(P, Q, k) = d_{\mathcal{H}}^{\mathtt{I}}(P, Q, k)$ and thus it suffices to consider the insertion only case. Let $P'$ and $Q'$ be as defined in $\mathbf{HausEdit}(P, Q, k, r)$. First, observe that $d_{\mathcal{H}}^{\mathtt{I}}(P, Q, k) \leq r$ if and only if $d_{\mathcal{H}}^{\mathtt{I}}(P', Q', k) \leq r$. This follows as any point in $P \setminus P'$ is already within distance $r$ to a point in $Q$ (and similarly for any point in $Q \setminus Q'$), and thus does not require an insertion to $r$-cover it. Moreover, no point $p \in P \setminus P'$ can be used to $r$-cover a point $q \in Q'$ as then $\|p - q\| \leq r$ which would imply $q \in Q \setminus Q'$. Finally, any point added to $Q'$ in $\mathsf{real}(d_{\mathcal{H}}^{\mathtt{I}}(P', Q', k))$ to $r$-cover a point in $P'$ will then itself be $r$-covered by $P'$ and hence does not require points in $P \setminus P'$ to cover it.

Next observe that $d_{\mathcal{H}}^{\mathtt{I}}(P', Q', k) \leq r$ if and only if $\min_{0 \leq k' \leq k} \max\{d_{\mathcal{H}}^{\mathtt{I}}(P', \emptyset, k'), d_{\mathcal{H}}^{\mathtt{I}}(\emptyset, Q', k - k')\} \leq r$, as shown by the following series of implications, where $\mathcal{X}$ is the set of all pairs of point sets $(\mathcal{C}_{P'}, \mathcal{C}_{Q'})$ such that $|\mathcal{C}_{Q'}| + |\mathcal{C}_{P'}| \leq k$ and $||c - Q'|| \leq r$ for all $c \in \mathcal{C}_{P'}$ and $||c - P'|| \leq r$ for all $c \in \mathcal{C}_{Q'}$. (Note that $\mathcal{C}_{P'}$ will be the subset of points we are inserting into $P'$ to $r$-cover points in $Q'$, so it suffices to restrict to subsets such that every point in $\mathcal{C}_{P'}$ is within distance $r$ of a point in $Q'$.)

For readability, we write "$\exists$ s.t. " as shorthand for "$\exists(\mathcal{C}_{P'}, \mathcal{C}_{Q'}) \in \mathcal{X}$ *s.t.* " below.

$$\begin{aligned}
& d_{\mathcal{H}}^{\mathtt{I}}(P', Q', k) \leq r \\
\Leftrightarrow\ & \exists \text{ s.t. } d_{\mathcal{H}}(P' \cup \mathcal{C}_{P'}, Q' \cup \mathcal{C}_{Q'}) \leq r \\
\Leftrightarrow\ & \exists \text{ s.t. } \max\{d_h(P' \cup \mathcal{C}_{P'}, Q' \cup \mathcal{C}_{Q'}), \\
& \qquad d_h(Q' \cup \mathcal{C}_{Q'}, P' \cup \mathcal{C}_{P'})\} \leq r \\
\Leftrightarrow\ & \exists \text{ s.t. } \max\{d_h(P', Q' \cup \mathcal{C}_{Q'}), d_h(\mathcal{C}_{P'}, Q' \cup \mathcal{C}_{Q'}), \\
& \qquad d_h(Q', P' \cup \mathcal{C}_{P'}), d_h(\mathcal{C}_{Q'}, P' \cup \mathcal{C}_{P'})\} \leq r \\
\Leftrightarrow\ & \exists \text{ s.t. } \max\{d_h(P', \mathcal{C}_{Q'}), d_h(\mathcal{C}_{P'}, Q' \cup \mathcal{C}_{Q'}), \\
& \qquad d_h(Q', \mathcal{C}_{P'}), d_h(\mathcal{C}_{Q'}, P' \cup \mathcal{C}_{P'})\} \leq r \\
\Leftrightarrow\ & \exists \text{ s.t. } \max\{d_h(P', \mathcal{C}_{Q'}), d_h(Q', \mathcal{C}_{P'})\} \leq r \\
& \qquad \text{since } \mathcal{C}_{Q'} \text{ and } \mathcal{C}_{P'} \text{ must be } r\text{-covered} \\
\Leftrightarrow\ & \exists \text{ s.t. } \max\{d_{\mathcal{H}}(P', \mathcal{C}_{Q'}), d_{\mathcal{H}}(Q', \mathcal{C}_{P'})\} \leq r \\
& \qquad \text{since the other direction must be} \leq r. \\
\Leftrightarrow\ & \min_{0 \leq k' \leq k} \max\{d_{\mathcal{H}}^{\mathtt{I}}(P', \emptyset, k'), d_{\mathcal{H}}^{\mathtt{I}}(\emptyset, Q', k - k')\} \leq r
\end{aligned}$$

Recall that by Lemma 1 and Lemma 2, for any point set $S$ and integer $\ell$, $d_{\mathcal{H}}^{\mathtt{I}}(S, \emptyset, \ell) = d_{\mathcal{H}}^{\mathtt{E}}(S, \emptyset, \ell) = kcenter(S, \ell)$. Thus, putting everything together,

$$\begin{aligned}
& d_{\mathcal{H}}^{\mathtt{E}}(P, Q, k) \leq r \ \text{ if any only if} \\
& \min_{0 \leq k' \leq k} \max\{kcenter(P', k'), kcenter(Q', k - k')\} \leq r.
\end{aligned} \tag{3.1}$$

Now at the end of $\mathbf{HausEdit}(P, Q, k, r)$, $\beta = \min_{0 \leq k' \leq k} \max\{\mathbf{kcen}(P', k'), \mathbf{kcen}(Q', k - k')\}$. As $\mathbf{kcen}(S, \ell)$ is an $\alpha$-approximation to $kcenter(S, \ell)$ for any point set $S$ and integer $\ell \geq 0$, Eq. (3.1) then implies that if $d_{\mathcal{H}}^{\mathtt{E}}(P, Q, k) \leq r$ then $\beta \leq \alpha r$ and so the algorithm returns True. Conversely, if $d_{\mathcal{H}}^{\mathtt{E}}(P, Q, k) > \alpha r$ then again by Eq. (3.1) we have $\beta > \alpha r$ as $\mathbf{kcen}(S, \ell) \geq kcenter(S, \ell)$ for any set $S$ and integer $\ell \geq 0$, and thus the algorithm returns False.

As for the running time, the sets $P'$ and $Q'$ can be computed in $O(mn)$ time. The running times of $\mathbf{kcen}(P', k')$ and $\mathbf{kcen}(Q', k - k')$ are respectively bounded by $T(n, k)$ and $T(m, k)$, and there are $k + 1$ possible values for $k'$. Thus the overall running time is $O\big(mn + k \cdot T(\max\{m, n\}, k)\big)$. □

**Lemma 5** *Let $R$ be the set consisting of 0 together with all pairwise distances in the set $P \cup Q$. Then there exists value $\rho \in R$ such that $d_{\mathcal{H}}^{\mathtt{E}}(P, Q, k) \leq \rho \leq 2d_{\mathcal{H}}^{\mathtt{E}}(P, Q, k)$.*

**Proof.** Eq. (3.1) from the proof of Lemma 4 implies that if $d_{\mathcal{H}}^{\mathtt{E}}(P, Q, k) \neq 0$ then it either occurs at a value

[3] We assume $T(n, k)$ is an increasing function of $n$ and $k$.

$kcenter(P', k')$ or $kcenter(Q', k-k')$ for some $k'$ and sets $P' \subseteq P$ and $Q' \subseteq Q$, or at a value of $r$ where the $P'$ and $Q'$ sets change.

First, note that $P'$ and $Q'$ are determined by removing all points within distance $r$ of any point in $Q$ or $P$, respectively. Thus the set of distances between a point in $P$ and a point in $Q$ captures all values of $r$ where the sets $P'$ and $Q'$ might change.

For any point set $S$ and integer $\ell \geq 0$, the greedy algorithm of Gonzalez [16] achieves a 2-approximation to $kcenter(S, \ell)$, while only placing centers at points in $S$. (Recall $S$ may be a strict subset from a metric on a larger point set $X$.) Thus if $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k)$ is achieved at some $kcenter(P', k')$ or $kcenter(Q', k-k')$ value, then there is some value $\rho$ in the set of all pairwise distances in $P$ or all pairwise distances in $Q$ such that $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k) \leq \rho \leq 2d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k)$.

Combining the cases we have that there exists value $\rho \in R$ such that $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k) \leq \rho \leq 2d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k)$. □

**Lemma 6** *Let $P$ be a set of $n$ points, $Q$ be a set of $m$ points, and $k \geq 0$ be an integer. Let* $\mathbf{kcen}(P, k)$ *be an algorithm which for some constant $1 < \alpha \leq 2$ returns an $\alpha$-approximation to $kcenter(P, k)$ in $O(T(n, k))$ time. Then for any constant $\varepsilon > 0$, one can compute an $(\alpha + \varepsilon)$-approximation to $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k)$ in $O\big((mn + k^2 \max\{m, n\}) \log(mn) + k \cdot T(\max\{m, n\}, k)\big)$ time.*

**Proof.** Let $R$ be the set of values from Lemma 5. We sort $R$ and then binary search over it with $\mathbf{HausEdit}(P, Q, k, r)$, where for the $\mathbf{kcen}(P, k)$ subroutine we use the standard $O(nk)$ time 2-approximation algorithm for $kcenter(P, k)$ due to Gonzalez [16] (which may, for now, differ from the $\alpha$-approximate subroutine from the current lemma statement). This ultimately yields a pair of values $r' < r^*$ that are consecutive in the sorted order of $R$, such that $\mathbf{HausEdit}(P, Q, k, r')$ returned False and $\mathbf{HausEdit}(P, Q, k, r^*)$ returned True.[4] Lemma 4 then implies $r' < d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k) \leq 2r^*$.

By Lemma 5 there exists a value $\rho \in R$ such that $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k) \leq \rho \leq 2d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k)$. Thus by Lemma 4 for any $x \geq \rho$, $\mathbf{HausEdit}(P, Q, k, x)$ must return True. This implies $r' < \rho$ as $\mathbf{HausEdit}(P, Q, k, r')$ returned False, and as $\rho \in R$ and $r^*$ is the next value in the sorted order of $R$ after $r'$, this in turn implies $r^* \leq \rho$. Combining this with the above inequalities gives $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k) \leq 2r^* \leq 2\rho \leq 4d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k)$, or equivalently $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k) \in [r^*/2, 2r^*]$.

Consider the set of values $X = \{x_0, x_1, \ldots, x_z\}$, where $x_i = \frac{r^*}{2}(1+\varepsilon/2)^i$ and $z = \lceil \log_{1+\varepsilon/2}(4) \rceil = O(1)$ for any constant $\varepsilon > 0$. Note that for any $i$ we have $x_{i+1}/x_i = (1 + \varepsilon/2)$, and $z$ was chosen such that $x_z \geq 2r^*$. Now we binary search over $X$ for an adjacent pair $x_j, x_{j+1} \in X$ such that $\mathbf{HausEdit}(P, Q, k, x_j)$ returns False and $\mathbf{HausEdit}(P, Q, k, x_{j+1})$ returns True, except this time we use the $\alpha$-approximate subroutine for $\mathbf{kcen}(P, k)$ from the current lemma statement. (By the above $\mathbf{HausEdit}(P, Q, k, x_z)$ must return True, so not all in $X$ can return False, and if all return True then $\alpha x_0$ is the desired approximation.) Since $\mathbf{HausEdit}(P, Q, k, x_j)$ returns False, by Lemma 4, $x_j < d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k)$, and as $x_{j+1}/x_j = (1 + \varepsilon/2)$, this implies that $x_{j+1} < (1 + \varepsilon/2)d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k)$. Since $\mathbf{HausEdit}(P, Q, k, x_{j+1})$ returns True we have $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k) \leq \alpha x_{j+1}$. Thus $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k) \leq \alpha x_{j+1} \leq \alpha(1 + \varepsilon/2)d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k) \leq (\alpha + \varepsilon)d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k)$ since $\alpha \leq 2$. In other words, $\alpha x_{j+1}$ is the desired $(\alpha + \varepsilon)$-approximation to $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k)$.

As for the running time, the set $R$ can be computed and sorted in $O(mn \log(mn))$ time. Each of the $O(\log(mn))$ calls to $\mathbf{HausEdit}(P, Q, k, r)$ when using the $O(nk)$ time 2-approximation for $\mathbf{kcen}(P, k)$ take $O(mn + k^2 \max\{m, n\})$ time. The $O(1)$ calls to $\mathbf{HausEdit}(P, Q, k, x_i)$ when using the $O(T(n, k))$ time $\alpha$-approximation for $\mathbf{kcen}(P, k)$ take $O\big(mn + k \cdot T(\max\{m, n\}, k)\big)$ time.[5] Thus the total time is $O\big((mn + k^2 \max\{m, n\}) \log(mn) + k \cdot T(\max\{m, n\}, k)\big)$ as claimed. □

Using the standard $O(nk)$ time 2-approximation algorithm for $kcenter(P, k)$ due to Gonzalez [16] for $\mathbf{kcen}(P, k)$ in Lemma 6 directly gives the following.

**Theorem 7** *Given a set $P$ of $n$ points, a set $Q$ of $m$ points, and an integer $k \geq 0$, then for any chosen constant $\varepsilon > 0$, one can compute $(2 + \varepsilon)$-approximation in $O((mn + k^2 \max\{m, n\}) \log(mn))$ time.*

In Appendix A.1 we show that for points in low dimensional Euclidean space we can get improved running times and approximations by using grids and WSPDs.

**Theorem 8** *Given a set $P \subset \mathbb{R}^d$ of $n$ points, a set $Q \subset \mathbb{R}^d$ of $m$ points, and an integer $k \geq 0$, where $d$ is a constant and $N = \max\{m, n\}$, then for any chosen constant $\varepsilon > 0$, one can compute a:*

- *$(2 + \varepsilon)$-approximation in $O\big(k^2 N \log(N)\big)$ time.*
- *$(1 + \varepsilon)$-approximation in $O\big(k^2 N \log(N)\big) + (k/\varepsilon)^{O(k^{1-1/d})}$ time.*

### 3.1 Separate Budgets

Let $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k, \ell)$ be the Hausdorff Separate Budget Edit Distance, which differs from the Hausdorff Edit Distance by separating the budgets for insertion and

[4] Since $0 \in R$, Lemma 4 implies if it returned True for all $r \in R$ then $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k) = 0$. Since $\rho \in R$ (as described in Lemma 5), it cannot return False for all $r \in R$ as Lemma 4 then implies $\rho < d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k)$.

[5] For $0 < \varepsilon \leq 1$, we are searching over $z = \lceil \log_{1+\varepsilon/2}(4) \rceil = O(1/\log(1+\varepsilon)) = O(1/\varepsilon)$ values. Thus more precisely we make $O(\log(1/\varepsilon))$ calls, which is a constant when assuming $\varepsilon$ is constant.

deletion, which are noted as $k$ and $\ell$, respectively. Below is our main theorem from this section, where $kcenterOut(P, k, \ell)$ denotes the $k$-center with $\ell$ outliers objective. All other details are in Appendix A.2

**Theorem 9** *Let $P$ be a set of $n$ points, $Q$ be a set of $m$ points, and let $k, \ell \geq 0$ be integers. Let* $\mathbf{kcen}(P, k, \ell)$ *be an algorithm which returns an $\alpha$-approximation to $kcenterOut(P, k, \ell)$ in $O(T(n, k, \ell))$ time, for some constant $\alpha > 1$. Then for any constant $\varepsilon > 0$, one can compute an $(\alpha + \varepsilon)$-approximation to $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k, \ell)$ in $O\big((mn + k\ell \cdot T(\max\{m, n\}, k, \ell)) \log(mn)\big)$ time.*

### 3.2 Deletion Only

Let $d^{\mathtt{D}}_{\mathcal{H}}(P, Q, k)$ be the Hausdorff Deletion Distance, which minimizes $d_{\mathcal{H}}\big(\mathtt{D}(P, k'), \mathtt{D}(Q, k - k')\big)$ where $0 \leq k' \leq k$. This is very similar to the Partial Hausdorff Distance [19], as discussed in the introduction, though unlike [19], here we provide a formal algorithm and analysis. (Indeed, the lemma below provides the formal argument of equivalence of the problems for separated budgets, as discussed in Appendix A.4.)

**Algorithm 2: HausDel**$(P, Q, k)$

```
1 If $k \geq n + m$ then return 0. If $P = \emptyset$ or $Q = \emptyset$
    return $\infty$.
2 Compute the multisets $D^P = \{\|p - Q\| \mid p \in P\}$
    and $D^Q = \{\|q - P\| \mid q \in Q\}$.
3 Let $\mathfrak{D} = \langle \mathfrak{D}_1, \mathfrak{D}_2, \ldots, \mathfrak{D}_{n+m} \rangle$ denote the values
    in $D = D^P \uplus D^Q$ listed in decreasing order.
4 return $\mathfrak{D}_{k+1}$
```

In Algorithm 2 and throughout this section, $X \uplus Y$ denotes the mutiset union (i.e. additive union) of two multisets $X$ and $Y$. Thus $|X \uplus Y| = |X| + |Y|$.

**Lemma 10** *Let $P$ be a set of $n$ points, $Q$ be a set of $m$ points, and $k \geq 0$ an integer. Then* $\mathbf{HausDel}(P, Q, k)$ *computes $d^{\mathtt{D}}_{\mathcal{H}}(P, Q, k)$ in $O(nm)$ time.*

**Proof.** The first line of the algorithm handles some trivial cases, and in the remainder of the proof we assume $k < n + m$ and neither $P \neq \emptyset$ nor $Q \neq \emptyset$.

Let $\mathfrak{D}(X, Y) = \langle \mathfrak{D}(X, Y)_1, \ldots \mathfrak{D}(X, Y)_{|X|+|Y|} \rangle$ be the list of (bi-chromatic) nearest neighbor distances between any sets $X$ and $Y$, sorted in descending order. Thus $d_{\mathcal{H}}(X, Y) = \mathfrak{D}(X, Y)_1$, and note that in the algorithm $\mathbf{HausDel}(P, Q, k)$, we have $\mathfrak{D} = \mathfrak{D}(P, Q)$.

Consider any pair of sets $P' \in \mathtt{D}(P, k'), Q' \in \mathtt{D}(Q, k - k')$ for any value $0 \leq k' \leq k$. Observe that when we delete points from $P$ and $Q$, producing the sets $P'$ and $Q'$, the nearest neighbor distances of the remaining points cannot decrease. Thus the list $\mathfrak{D}(P', Q')$ is obtained from the list $\mathfrak{D}(P, Q)$ by deleting up to $k$ values from the list, and mapping each remaining value $\mathfrak{D}(P, Q)_i$ to a unique value $\mathfrak{D}(P', Q')_j$ such that $\mathfrak{D}(P, Q)_i \leq \mathfrak{D}(P', Q')_j$. Thus $\mathfrak{D}(P, Q)_{k+1} \leq \mathfrak{D}(P', Q')_1$, as clearly over the space of all such mappings of the list $\mathfrak{D}(P, Q)$, the resulting list with the minimum first value would arise from deleting the top $k$ values and not increasing the $(k + 1)$th value.

Let $z$ be the largest index such that $z \leq k$ and $\mathfrak{D}(P, Q)_z > \mathfrak{D}(P, Q)_{k+1}$ ($z = 0$ if no such index exists). Now let $\langle x_1, \ldots, x_{n+m} \rangle$ denote the ordered list of points from $P$ and $Q$ corresponding to the values $\langle \mathfrak{D}(P, Q)_1, \ldots, \mathfrak{D}(P, Q)_{n+m} \rangle$, and let $\hat{P} \subseteq P$ and $\hat{Q} \subseteq Q$ be the sets resulting from deleting $x_1, \ldots, x_z$ (no points are deleted if $z = 0$). We now argue that $\mathfrak{D}(P, Q)_{k+1} = \mathfrak{D}(\hat{P}, \hat{Q})_1$, which, since we already argued that $\mathfrak{D}(P, Q)_{k+1}$ was a lower bound on any possible deletion of $k$ points, implies $d^{\mathtt{D}}_{\mathcal{H}}(P, Q, k) = \mathfrak{D}(P, Q)_{k+1}$.

Observe that for any $i > z$, no point in $x_1, \ldots, x_z$ can be the nearest neighbor of $x_i$ from the other set (i.e. realizing $\mathfrak{D}(P, Q)_i$), as this would imply $x_z$ has a neighbor from the other set in distance less than $\mathfrak{D}(P, Q)_z$, a contradiction. This implies $\mathfrak{D}(\hat{P}, \hat{Q}) = \langle \mathfrak{D}(P, Q)_{z+1}, \ldots, \mathfrak{D}(P, Q)_{n+m} \rangle$. Thus we have $\mathfrak{D}(P, Q)_{k+1} = \mathfrak{D}(\hat{P}, \hat{Q})_1$ as claimed, since by the definition of $z$ we have that $\mathfrak{D}(P, Q)_{z+1} = \mathfrak{D}(P, Q)_{k+1}$.

As for the running time, the multiset $D = D^P \uplus D^Q$ can be computed in $O(nm)$ time by looking at all pairwise distances. The return value $\mathfrak{D}_{k+1}$ is the $k + 1$ largest value in the multiset $D$, which can be computed in $O(|D|) = O(n + m)$ time using the standard linear time median selection algorithm [8], thus the total time is dominated by the time to compute $D$. □

In the above algorithm it takes $O(nm)$ to compute the multiset of all bi-chromatic nearest neighbor distances, namely $D = D^P \uplus D^Q$. In the plane, these distances can be computed more efficiently using Voronoi diagrams. Namely, it takes $O(n \log n)$ time to compute the Voronoi diagram of $P$, which allows for $O(\log n)$ time point location queries. Thus, by querying all of $Q$, we get $D^Q$ in $O((n + m) \log n)$ time. Similarly, $D^P$ can be computed in $O((n + m) \log m)$ time. As returning the $k + 1$ largest value in $D$ takes $O(n + m)$ time, we have the following.

**Corollary 11** *Let $P \subset \mathbb{R}^2$ be a set of $n$ points, $Q \subset \mathbb{R}^2$ be a set of $m$ points, and $k \geq 0$ be an integer. Let $N = \max\{m, n\}$, then one can compute $d^{\mathtt{D}}_{\mathcal{H}}(P, Q, k)$ in $O\big(N \log(N)\big)$ time.*

In Appendix A.3 we argue that by using Approximate Nearest Neighbor data structures one can achieve a fast $(1 + \varepsilon)$-approximation for points in $\mathbb{R}^d$, for constant $d$.

**Corollary 12** *Let $P \subset \mathbb{R}^d$ be a set of $n$ points, $Q \subset \mathbb{R}^d$ be a set of $m$ points, and $k \geq 0$ be an integer, where $d$ is a constant. Let $N = \max\{m, n\}$, then one can compute* *a* $(1+\varepsilon)$*-approximation to* $d^{\mathtt{D}}_{\mathcal{H}}(P,Q,k)$ *in* $O\big(N\log(N)+N/\varepsilon^d\big)$ *time.*

Appendix A.4 discusses how our above results for the deletion only and general Hausdorff Edit Distance can be extended to other natural variants.

## References

[1] P. K. Agarwal, K. Fox, J. Pan, and R. Ying. Approximating dynamic time warping and edit distance for a pair of point sequences. In *Proceedings of the 32nd International Symposium on Computational Geometry*, pages 6:1–6:16, 2016.

[2] P. K. Agarwal, S. Har-Peled, M. Sharir, and Y. Wang. Hausdorff distance under translation for points and balls. *ACM Trans. Algorithms*, 6(4):71:1–71:26, 2010.

[3] P. K. Agarwal and C. M. Procopiuc. Exact and approximation algorithms for clustering. *Algorithmica*, 33(2):201–226, 2002.

[4] R. B. Avraham, M. Henze, R. Jaume, B. Keszegh, O. E. Raz, M. Sharir, and I. Tubis. Partial-matching RMS distance under translation: Combinatorics and algorithms. *Algorithmica*, 80(8):2400–2421, 2018.

[5] L. Biabani, A. Hennes, M. Monemizadeh, and M. Schmidt. Faster query times for fully dynamic k-center clustering with outliers. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 9226–9247. Curran Associates, Inc., 2023.

[6] D. Chakrabarty, P. Goyal, and R. Krishnaswamy. The non-uniform k-center problem. *ACM Trans. Algorithms*, 16(4), June 2020.

[7] M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '01, page 642–651, USA, 2001. Society for Industrial and Applied Mathematics.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Third Edition.* The MIT Press, 3rd edition, 2009.

[9] M. de Berg, M. Monemizadeh, and Y. Zhong. k-Center Clustering with Outliers in the Sliding-Window Model. In P. Mutzel, R. Pagh, and G. Herman, editors, *29th Annual European Symposium on Algorithms (ESA 2021)*, volume 204 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 13:1–13:13, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[10] H. Ding, H. Yu, and Z. Wang. Greedy Strategy Works for k-Center Clustering with Outliers and Coreset Construction. In M. A. Bender, O. Svensson, and G. Herman, editors, *27th Annual European Symposium on Algorithms (ESA 2019)*, volume 144 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 40:1–40:16, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[11] T. Feder and D. Greene. Optimal algorithms for approximate clustering. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, page 434–444, New York, NY, USA, 1988. Association for Computing Machinery.

[12] A. Fischer, C. Y. Suen, V. Frinken, K. Riesen, and H. Bunke. Approximation of graph edit distance based on hausdorff matching. *Pattern Recognition*, 48(2):331–343, 2015.

[13] E. Fox, A. Nayyeri, J. J. Perry, and B. Raichel. Fréchet Edit Distance. In W. Mulzer and J. M. Phillips, editors, *40th International Symposium on Computational Geometry (SoCG 2024)*, volume 293 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 58:1–58:15, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[14] K. Fox and X. Li. Approximating the geometric edit distance. *Algorithmica*, 84(9):2395–2413, 2022.

[15] O. Gold and M. Sharir. Dynamic time warping and geometric edit distance: Breaking the quadratic barrier. *ACM Transactions on Algorithms*, 14(4):50, 2018.

[16] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[17] S. Har-peled. *Geometric Approximation Algorithms.* American Mathematical Society, USA, 2011.

[18] W.-L. Hsu and G. L. Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979.

[19] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

## A Hausdorff Edit Distance Continued

### A.1 The Euclidean Case

Throughout, $P$ and $Q$ have been finite point sets in an arbitrary metric. If instead we restrict $P$ and $Q$ to be finite point sets in $\mathbb{R}^d$ with the standard Euclidean metric, then when $d$ is a constant, the above result can be improved both in terms of running time and approximation quality (though the latter requires bounding $k$). We now sketch how to achieve this.

For the running time, the $mn$ term in Theorem 7 arises due to two reasons. The first reason is on Line 1 in **HausEdit**$(P,Q,k,r)$. Here the algorithm marks a set $P_r$ of points which is defined as all $p \in P$ within distance $r$ of some point $q \in Q$ (and similarly define $Q_r$). Analogously define the set $P_{\alpha r}$ consisting of all $p \in P$ within distance $\alpha r$ of some point $q \in Q$. Observe that the argument of correctness for the algorithm is identical if instead of precisely marking $P_r$ we instead mark any subset $\hat{P} \subseteq P$ such that $P_r \subseteq \hat{P} \subseteq P_{\alpha r}$. When $d$ is a constant, finding such a set $\hat{P}$ can be done in $O(n+m)$ time using standard grid based techniques, see for example [17].[6] Thus using grids, the run time of **HausEdit**$(P, Q, k, r)$ becomes $O\big(m + n + k \cdot T(\max\{m, n\}, k)\big) = O\big(k \cdot T(\max\{m, n\}, k)\big)$.

The other reason for the $mn$ term in Theorem 7 is from the use of the $O(mn)$ sized set $R$ from Lemma 5, consisting of all pairwise distances in $P \cup Q$. However, if $P, Q \subset \mathbb{R}^d$ for any constant $d$, then we can use WSPD's (see for example [17]) to approximate the values from the set $R$ in Lemma 5. In particular, in $O((m+n)\log(m+n) + (m+n)/\varepsilon^d)$ time one can construct a set $R'$ of size $O((m+n)/\varepsilon^d)$ such that for any value $r \in R$ there exists a value $r' \in R'$ where $r \leq r' \leq (1+\varepsilon)r$. Thus in Lemma 6 rather than binary searching over $R$, we can binary search over $R'$. Recall that there the binary search over $R$ was used to construct a constant spread interval containing $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k)$. If instead we binary search over $R'$ this will produce an interval containing $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k)$ that is a $(1+\varepsilon)$ factor larger, and as all we require is that the interval has constant spread, we can in fact set $\varepsilon = 1$. The rest of the proof, which then searches over this constant spread interval, remains the same and the running time then becomes $O\big((m+n+k^2\max\{m,n\})\log(mn) + k \cdot T(\max\{m,n\},k)\big) = O\big(k^2\max\{m,n\}\log(mn) + k\cdot T(\max\{m,n\},k)\big)$.

Using the 2-approximation of Gonzalez for **kcen** we thus get a $O\big(k^2\max\{m,n\}\log(mn)\big)$ time $(2+\varepsilon)$-approximation. On the other hand, for a point set $P \subset \mathbb{R}^d$ for constant $d$, [3] gave an $O(n \log k) + (k/\varepsilon)^{O(k^{1-1/d})}$ time $(1+\varepsilon)$-approximation to *kcenter*$(P, k)$. Thus instead using this for **kcen** gives an $O(k^2\max\{m,n\}\log(mn)) + (k/\varepsilon)^{O(k^{1-1/d})}$ time $(1+\varepsilon)$-approximation. Thus in summary we have the following.

**Theorem 8 (Restated)** *Given a set $P \subset \mathbb{R}^d$ of $n$ points, a set $Q \subset \mathbb{R}^d$ of $m$ points, and an integer $k \geq 0$, where $d$ is a constant and $N = \max\{m, n\}$, then for any chosen constant $\varepsilon > 0$, one can compute a:*

- $(2+\varepsilon)$*-approximation in* $O\big(k^2 N \log(N)\big)$ *time.*
- $(1+\varepsilon)$*-approximation in* $O\big(k^2 N \log(N)\big) + (k/\varepsilon)^{O(k^{1-1/d})}$ *time.*

## A.2 Separate Budgets Continued

Here we give the full details from Section 3.1.

Let $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k, \ell)$ be the Hausdorff Separate Budget Edit Distance, which differs from the Hausdorff Edit Distance by separating the budgets for insertion and deletion, which are noted as $k$ and $\ell$, respectively. Despite Lemma 1 not holding for this problem, we will prove nearly equivalent results as those shown for Hausdorff Edit Distance.

First, observe that $d^{\mathtt{I}}_{\mathcal{H}}(P, Q, k) = d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k, 0)$. As Theorem 3 showed computing $d^{\mathtt{I}}_{\mathcal{H}}(P, Q, k)$ is APX-Hard, we immediately have the following corollary.

**Corollary 13** *For any point sets $P, Q$ and integers $k, \ell \geq 0$, the problem of computing $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k, \ell)$ is APX-Hard. The problem remains APX-Hard even when $P, Q \subset \mathbb{R}^2$.*

Recall that $\mathtt{D}(P, \ell)$ denotes the set of all subsets of $P$ resulting from deleting at most $\ell$ points from $P$. Then for the $k$-center clustering with outliers problem [6, 7] the objective is to compute *kcenterOut*$(P, k, \ell) = \min_{P^{\mathtt{D}} \in \mathtt{D}(P,\ell)}\big(\textit{kcenter}(P^{\mathtt{D}}, k)\big)$.

**Lemma 14** *For any finite point set $P$ and integers $k, \ell \geq 0$, $d^{\mathtt{E}}_{\mathcal{H}}(P, \emptyset, k, \ell) = $ kcenterOut$(P, k, \ell)$.*

**Proof.** $d^{\mathtt{E}}_{\mathcal{H}}(P, \emptyset, k, \ell) = \min_{P^{\mathtt{D}} \in \mathtt{D}(P,\ell)}\big(d^{\mathtt{I}}_{\mathcal{H}}(P^{\mathtt{D}}, \emptyset, k)\big)$ and by definition *kcenterOut*$(P, k, \ell) = \min\limits_{P^{\mathtt{D}} \in \mathtt{D}(P,\ell)}\big(\textit{kcenter}(P^{\mathtt{D}}, k)\big)$. The lemma statement then follows by Lemma 1 and Lemma 2, which when combined say that $d^{\mathtt{I}}_{\mathcal{H}}(P, \emptyset, k) = $ *kcenter*$(P, k)$. □

We are also able to construct an equivalent decision algorithm, **HausEdit**$(P, Q, k, \ell, r)$, which uses as a subroutine any $\alpha$-approximation algorithm for *kcenterOut*$(P, k, \ell)$, which we denote by **kcen**$(P, k, \ell)$.

**Algorithm 3: HausEdit**$(P, Q, k, \ell, r)$

```
Mark all $p \in P$ such that $\|p - Q\| \leq r$.
  Mark all $q \in Q$ such that $\|q - P\| \leq r$.
Create sets $P' \subseteq P$ and $Q' \subseteq Q$ by removing all
  marked points.
$\beta = \infty$
for $\ell' = 0$ to $\ell$ do
    for $k' = 0$ to $k$ do
        $\beta = \min\{\beta, \max\{$kcen$(P', k', \ell'),$
                    kcen$(Q', k - k', \ell - \ell')\}\}$
if $\beta \leq \alpha r$ then      // kcen$(P, k, \ell)$ is an $\alpha$-approx.
    return True
else
    return False
```

**Lemma 15** *Let $P$ and $Q$ be sets of $n$ and $m$ points. Let $k, \ell \geq 0$ be integers, and* **kcen**$(P, k, \ell)$ *an algorithm returning an $\alpha$-approximation to kcenterOut$(P, k, \ell)$ in $O(T(n, k, \ell))$ time.*[7] *Then if* **HausEdit**$(P, Q, k, \ell, r)$ *returns True then $r \geq d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k, \ell)/\alpha$, and if* **HausEdit**$(P, Q, k, \ell, r)$ *returns False then $r < d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k, \ell)$. The running time of* **HausEdit**$(P, Q, k, \ell, r)$ *is* $O\big(mn + k\ell \cdot T(\max\{m, n\}, k, \ell)\big)$.

**Proof.** The proof follows the same logic as that of Lemma 4, though with some changes required. For clarity, we reproduce the entire proof with the relevant changes.

Let $P', Q'$ be as defined in **HausEdit**$(P, Q, k, \ell, r)$. First, observe that $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k, \ell) \leq r$ if and only if $d^{\mathtt{E}}_{\mathcal{H}}(P', Q', k, \ell) \leq r$. This follows as any point in $P \setminus P'$ is already within distance $r$ to a point in $Q \setminus Q'$, and vice versa. Thus points in $P \setminus P'$ and $Q \setminus Q'$ are $r$-covered without

[6] As a rough sketch, consider the uniform grid of cell side length $(\alpha - 1)r/\sqrt{d}$ and diameter $(\alpha - 1)r$. Hash all points from $P$ into the cells of this grid. For every point $q \in Q$, there are $O((2(\alpha - 1)/\sqrt{d})^d) = O(1)$ cells intersecting a ball of radius $r$ around $q$, and any point in these cells is at most $r + (\alpha - 1)r = \alpha r$ away from $q$. Thus we simply mark all points from $P$ contained in any such cell around a point of $Q$.

[7] We assume $T(n, k, \ell)$ is an increasing function of $n$, $k$, and $\ell$.

the need for any insertions or deletions, and any deletions from $P'$ or $Q'$ will not change this fact. Moreover, points in $P \setminus P'$ do not $r$-cover any point in $Q'$ (nor do points in $Q \setminus Q'$ $r$-cover points in $P'$), and any point added to $Q'$ in $\mathsf{real}(d^{\mathtt{I}}_{\mathcal{H}}(P', Q', k, \ell))$ to $r$-cover a point in $P'$ that is not deleted, will then itself be $r$-covered and hence does not require points in $P \setminus P'$ to cover it.

Next we argue that $d^{\mathtt{E}}_{\mathcal{H}}(P', Q', k, \ell) \leq r$ if and only if $\min_{0 \leq \ell' \leq \ell, 0 \leq k' \leq k} \max\{\mathit{kcenterOut}(P', k', \ell'), \mathit{kcenterOut}(Q', k - k', \ell - \ell')\} \leq r$. Let $\mathcal{X}$ represent the set of tuples $(P^{\mathtt{D}}, Q^{\mathtt{D}}, \mathcal{C}_{P^{\mathtt{D}}}, \mathcal{C}_{Q^{\mathtt{D}}})$ such that $P^{\mathtt{D}} \in \mathtt{D}(P', \ell')$ and $Q^{\mathtt{D}} \in \mathtt{D}(Q', \ell - \ell')$ for some integer $0 \leq \ell' \leq \ell$, and such that $|\mathcal{C}_{P^{\mathtt{D}}}| + |\mathcal{C}_{Q^{\mathtt{D}}}| \leq k$ where $||c - Q^{\mathtt{D}}|| \leq r$ for all $c \in \mathcal{C}_{P^{\mathtt{D}}}$ and $||c - P^{\mathtt{D}}|| \leq r$ for all $c \in \mathcal{C}_{Q^{\mathtt{D}}}$. (Without loss of generality, we view all deletions as occurring before any insertion, thus first producing the sets $P^{\mathtt{D}}$ and $Q^{\mathtt{D}}$. Note the set $\mathcal{C}_{P^{\mathtt{D}}}$ will then be the subset of points we are inserting into $P^{\mathtt{D}}$ to $r$-cover points in $Q^{\mathtt{D}}$, so it suffices to restrict to subsets such that every point in $\mathcal{C}_{P^{\mathtt{D}}}$ is within distance $r$ of a point in $Q^{\mathtt{D}}$.) For readability, we write "$\exists$ s.t. " as shorthand for "$\exists (P^{\mathtt{D}}, Q^{\mathtt{D}}, \mathcal{C}_{P^{\mathtt{D}}}, \mathcal{C}_{Q^{\mathtt{D}}}) \in \mathcal{X}$ *s.t.* " below.

$$\begin{aligned}
& d^{\mathtt{E}}_{\mathcal{H}}(P', Q', k, \ell) \leq r \\
\Leftrightarrow\ & \exists \text{ s.t. }\ d_{\mathcal{H}}(P^{\mathtt{D}} \cup \mathcal{C}_{P^{\mathtt{D}}}, Q^{\mathtt{D}} \cup \mathcal{C}_{Q^{\mathtt{D}}}) \leq r \\
\Leftrightarrow\ & \exists \text{ s.t. }\ \max\{d_h(P^{\mathtt{D}} \cup \mathcal{C}_{P^{\mathtt{D}}}, Q^{\mathtt{D}} \cup \mathcal{C}_{Q^{\mathtt{D}}}), \\
& \qquad d_h(Q^{\mathtt{D}} \cup \mathcal{C}_{Q^{\mathtt{D}}}, P^{\mathtt{D}} \cup \mathcal{C}_{P^{\mathtt{D}}})\} \leq r \\
\Leftrightarrow\ & \exists \text{ s.t. }\ \max\{d_h(P^{\mathtt{D}}, Q^{\mathtt{D}} \cup \mathcal{C}_{Q^{\mathtt{D}}}), d_h(\mathcal{C}_{P^{\mathtt{D}}}, Q^{\mathtt{D}} \cup \mathcal{C}_{Q^{\mathtt{D}}}), \\
& \qquad d_h(Q^{\mathtt{D}}, P^{\mathtt{D}} \cup \mathcal{C}_{P^{\mathtt{D}}}), d_h(\mathcal{C}_{Q^{\mathtt{D}}}, P^{\mathtt{D}} \cup \mathcal{C}_{P^{\mathtt{D}}})\} \leq r \\
\Leftrightarrow\ & \exists \text{ s.t. }\ \max\{d_h(P^{\mathtt{D}}, \mathcal{C}_{Q^{\mathtt{D}}}), d_h(\mathcal{C}_{P^{\mathtt{D}}}, Q^{\mathtt{D}} \cup \mathcal{C}_{Q^{\mathtt{D}}}), \\
& \qquad d_h(Q^{\mathtt{D}}, \mathcal{C}_{P^{\mathtt{D}}}), d_h(\mathcal{C}_{Q^{\mathtt{D}}}, P^{\mathtt{D}} \cup \mathcal{C}_{P^{\mathtt{D}}})\} \leq r \\
\Leftrightarrow\ & \exists \text{ s.t. }\ \max\{d_h(P^{\mathtt{D}}, \mathcal{C}_{Q^{\mathtt{D}}}), d_h(Q^{\mathtt{D}}, \mathcal{C}_{P^{\mathtt{D}}})\} \leq r \\
& \qquad \text{since } \mathcal{C}_{Q^{\mathtt{D}}} \text{ and } \mathcal{C}_{P^{\mathtt{D}}} \text{ must be } r\text{-covered} \\
\Leftrightarrow\ & \exists \text{ s.t. }\ \max\{d_{\mathcal{H}}(P^{\mathtt{D}}, \mathcal{C}_{Q^{\mathtt{D}}}), d_{\mathcal{H}}(Q^{\mathtt{D}}, \mathcal{C}_{P^{\mathtt{D}}})\} \leq r \\
& \qquad \text{since the other direction must be} \leq r \\
\Leftrightarrow\ & \min\nolimits_{0 \leq \ell' \leq \ell, 0 \leq k' \leq k} \max\{d^{\mathtt{E}}_{\mathcal{H}}(P', \emptyset, k', \ell'), \\
& \qquad d^{\mathtt{E}}_{\mathcal{H}}(\emptyset, Q', k - k', \ell - \ell')\} \leq r
\end{aligned}$$

Recall that by Lemma 14, for any point set $S$ and integers $a$ and $b$, $d^{\mathtt{E}}_{\mathcal{H}}(S, \emptyset, a, b) = \mathit{kcenterOut}(S, a, b)$. Thus putting everything together we have that

$$\begin{aligned}
& d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k, \ell) \leq r \ \text{ if and only if} \qquad \text{(A.1)} \\
& \min_{\substack{0 \leq \ell' \leq \ell, \\ 0 \leq k' \leq k}} \max\big\{\mathit{kcenterOut}(P', k', \ell'), \\
& \qquad \mathit{kcenterOut}(Q', k - k', \ell - \ell')\big\} \leq r.
\end{aligned}$$

Now at the end of $\mathbf{HausEdit}(P, Q, k, r)$, $\beta = \min_{0 \leq \ell' \leq \ell, 0 \leq k' \leq k} \max\{\mathbf{kcen}(P', k', \ell'), \mathbf{kcen}(Q', k - k', \ell - \ell')\}$. As $\mathbf{kcen}(S, a, b)$ is an $\alpha$-approximation to $\mathit{kcenterOut}(S, a, b)$ for any point set $S$ and integers $a, b \geq 0$, Eq. (A.1) then implies that if $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k, \ell) \leq r$ then $\beta \leq \alpha r$ and so the algorithm returns True. Conversely, if $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k, \ell) > \alpha r$ then again by Eq. (A.1) we have $\beta > \alpha r$ as $\mathbf{kcen}(S, a, b) \geq \mathit{kcenterOut}(S, a, b)$ for any set $S$ and integers $a, b \geq 0$, and thus the algorithm returns False.

As for the running time, the sets $P'$ and $Q'$ can be computed in $O(mn)$ time. The running times of $\mathbf{kcen}(P', k', \ell')$ and $\mathbf{kcen}(Q', k - k', \ell - \ell')$ are respectively bounded by $T(n, k, \ell)$ and $T(m, k, \ell)$, and there are $\ell + 1$ possible values for $\ell'$ and $k + 1$ possible values for $k'$. Thus the overall running time is $O\big(mn + k\ell \cdot T(\max\{m, n\}, k, \ell)\big)$. □

It is easy to see that Lemma 5 applies to the separate budgets problem as well. Additionally, Lemma 6 also still applies as it is simply searching using the decision procedure. The analogous lemma for the separate budget case is given below as a theorem. Recall Lemma 6 first used the Gonzalez algorithm to get a constant spread interval, which was then searched over using an $\alpha$-approximation. Here we assume the same $\alpha$-approximation is used in both parts for simplicity, and as there is not a clear best algorithm to choose now. (For simplicity, Lemma 6 assumed $\alpha \leq 2$, though the proof extends to any constant $\alpha$.)

**Theorem 9 (Restated)** *Let $P$ be a set of $n$ points, $Q$ be a set of $m$ points, and let $k, \ell \geq 0$ be integers. Let $\mathbf{kcen}(P, k, \ell)$ be an algorithm which returns an $\alpha$-approximation to $\mathit{kcenterOut}(P, k, \ell)$ in $O(T(n, k, \ell))$ time, for some constant $\alpha > 1$. Then for any constant $\varepsilon > 0$, one can compute an $(\alpha + \varepsilon)$-approximation to $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k, \ell)$ in $O\big((mn + k\ell \cdot T(\max\{m, n\}, k, \ell)) \log(mn)\big)$ time.*

As discussed in the introduction, there are various known algorithms for $k$-center clustering with outliers, with different trade-offs. For example, if we use the simple greedy constant factor approximation of [7] in the above theorem, then we get a polynomial time constant factor approximation for $d^{\mathtt{E}}_{\mathcal{H}}(P, Q, k, \ell)$.

### A.3 Argument for Corollary 12

Given a set $P \subset \mathbb{R}^d$ of $n$ points, for a query point $q$, we refer to a point $x \in P$ as a $(1 + \varepsilon)$ approximate nearest neighbor (ANN) of $q$ if $\|x - q\| \leq (1 + \varepsilon) \min_{p \in P} \|p - q\|$. It is known that for any point set $P \subset \mathbb{R}^d$ for constant $d$, in $O(n \log n)$ time one can construct a data structure such that given any query point $q$, in $O(\log n + 1/\varepsilon^d)$ time it returns a $(1 + \varepsilon)$-ANN [17].

We can replace the multiset $D$ of exact bi-chromatic nearest neighbor distances from $\mathbf{HausDel}(P, Q, k)$ (Algorithm 2) with a multiset $D^\varepsilon$ of approximate bi-chromatic nearest neighbor distances as follows. For the point set $P$, in $O(n \log n)$ time we construct a $(1 + \varepsilon)$-ANN data structure, which supports $O(\log n + 1/\varepsilon^d)$ time nearest neighbor queries, and then query all points in $Q$. Then we build such a structure for $Q$ and query all points of $P$. Thus the multiset of all distances between queried points and their $(1 + \varepsilon)$-ANN is the desired multiset $D^\varepsilon$, which took $O(n \log n + m(\log n + 1/\varepsilon^d) + m \log m + n(\log m + 1/\varepsilon^d)) = O(N \log N + N/\varepsilon^d)$ time to compute, where $N = \max\{m, n\}$.

Recall that $\mathbf{HausDel}(P, Q, k)$ returned the $k + 1$ largest value in $D$, denoted $\mathfrak{D}_{k+1}$. Let $\mathfrak{D}^\varepsilon_{k+1}$ denote the $k + 1$ largest value in $D^\varepsilon$. Clearly $\mathfrak{D}_{k+1} \leq \mathfrak{D}^\varepsilon_{k+1}$ as our $(1 + \varepsilon)$-ANN distances are never smaller than the true nearest neighbor distances. Moreover, $\mathfrak{D}^\varepsilon_{k+1} \leq (1 + \varepsilon)\mathfrak{D}_{k+1}$, as clearly the largest $\mathfrak{D}^\varepsilon_{k+1}$ could possibly be is if all ANN distances were exactly a $(1 + \varepsilon)$ factor larger (the maximum allowed), in which case the value at any rank would be exactly a $(1 + \varepsilon)$

factor larger. Note, the actual point realizing $\mathfrak{D}^{\varepsilon}_{k+1}$ may not correspond to the one realizing $\mathfrak{D}_{k+1}$, though it does not matter.

## A.4 Additional Applications

Here we observe that the above results easily extend to other natural variants.

**Set-Partitioned Budgets.** In Section 3.1 we considered the version where we had separate insertion and deletion budgets $k$ and $\ell$. Now consider the version where these insertion and deletion budgets are further partitioned among $P$ and $Q$, i.e. as part of the input we are given $k_P$, $k_Q$, $\ell_P$, and $\ell_Q$. Algorithm 3 immediately extends to this case if we simply remove the for loops and directly compute $\max\{\mathbf{kcen}(P', k_P, \ell_P), \mathbf{kcen}(Q', k_Q, \ell_Q)\}$, as indeed the purpose of the loops was to guess the partition of $k$ and $\ell$ into $k_P$, $k_Q$, $\ell_P$, and $\ell_Q$. Thus we get an analogous version of Theorem 9, though without the $k\ell$ term in the running time.

For the problem where edits of either type are allowed though the edits are partitioned among $P$ and $Q$, i.e. we are given $k_P$ and $k_Q$, we can no longer assume there are no deletions, i.e. Lemma 1 does not hold. However, in this case we can simply modify the for loops in Algorithm 3 to guess the partition of $k_P$ and $k_Q$ into deletions and insertions, yielding an analogous version of Theorem 9, though where the $k\ell$ term in the running time is replaced with $k_P k_Q$.

Note that the version of the problem where edits are only allowed to one side, say $P$, is now a special case of the problem where $k_Q = 0$ (or $k_Q = 0$ and $\ell_Q = 0$). However, in this case, since we are considering the $r$ decision problem, we must delete all points in $P'$. Thus we can use $k$-center rather than $k$-center with outliers (i.e. modify Algorithm 1 not Algorithm 3 ), yielding analogous versions of Theorem 7 and Theorem 8, where the $k^2$ terms in the running times improve to $k$.

**Deletion Only Set-Partitioned Budgets.** We can also consider the variant of the deletion only problem where, rather than being given a single deletion budget $k$, we are given separate deletion budgets $k_P$ for $P$ and $k_Q$ for $Q$. This case is easily handled by modifying Algorithm 2. Specifically, there we had multisets $D^P$ and $D^Q$, and we returned the $k+1$ largest element in $D^P \uplus D^Q$. Instead, we simply return the maximum of the $k_P + 1$ largest element in $D^P$ and the $k_Q + 1$ largest element in $D^Q$. The running time and correctness of Lemma 10 and Corollary 11 still apply with only superficial changes in the arguments.

As discussed in the introduction, [19] defined the Partial Hausdorff Distance, where given parameters $k$ and $\ell$, we want the maximum of the $k$th smallest value in $\{\|p - Q\| \mid p \in P\}$ and the $\ell$th smallest value in $\{\|q - P\| \mid q \in Q\}$. This is precisely the deletion only Hausdorff distance for parameters $n - k$ and $m - \ell$, and in fact our proof of Lemma 10 formally argues the equivalence of finding the value of a given rank and the deletion only variant.

**Substitutions.** In this paper we allowed insertions and deletions for our edits. One could also consider substitutions of one point for another. However, this introduces many possible variants, as then one must decide whether they are shared or split edit budgets among $P$ and $Q$, and among the three possible operations.

Observe that a substitution can be viewed as a deletion plus an insertion,[8] which suggests that some of these variants may be amenable to approximations using $k$-center clustering with outliers as done above. However, there are cases where one may wish to perform an insertion (with no deletion) over a substitution. In particular, this can occur when any further substitution would require moving (i.e. deleting) a point from $P \setminus P'$ or $Q \setminus Q'$. Thus depending on how the budgets are shared or not the problem may become more challenging, and may be a good direction to pursue for future research.

[8] We assume the insertion and deletion both occur within $P$ or both within $Q$, though you could create even more variants, where the insertion and deletion can occur in different sets.