Tight Bounds for Approximate Near Neighbor Searching for Time Series under the Fréchet Distance

Karl Bringmann[*] Anne Driemel[†] André Nusser[‡] Ioannis Psarros[§]

Abstract

We study the c-approximate near neighbor problem under the continuous Fréchet distance: Given a set of n polygonal curves with m vertices, a radius $\delta > 0$, and a parameter $k \leq m$, we want to preprocess the curves into a data structure that, given a query curve q with k vertices, either returns an input curve with Fréchet distance at most $c \cdot \delta$ to q, or returns that there exists no input curve with Fréchet distance at most δ to q. We focus on the case where the input and the queries are one-dimensional polygonal curves—also called time series—and we give a comprehensive analysis for this case. We obtain new upper bounds that provide different tradeoffs between approximation factor, preprocessing time, and query time.

Our data structures improve upon the state of the art in several ways. We show that for any $0 < \varepsilon \leq 1$ an approximation factor of $(1 + \varepsilon)$ can be achieved within the same asymptotic time bounds as the previously best result for $(2 + \varepsilon)$. Moreover, we show that an approximation factor of $(2+\varepsilon)$ can be obtained by using preprocessing time and space $O(nm)$, which is linear in the input size, and query time in $O(\frac{1}{\varepsilon})^{k+2}$, where the previously best result used preprocessing time in $n \cdot O(\frac{m}{\varepsilon k})^k$ and query time in $O(1)^k$. We complement our upper bounds with matching conditional lower bounds based on the Orthogonal Vectors Hypothesis. Interestingly, some of our lower bounds already hold for any super-constant value of k. This is achieved by proving hardness of a one-sided sparse version of the Orthogonal Vectors problem as an intermediate problem, which we believe to be of independent interest.

[*]Saarland University and Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany. This work is part of the project TIPEA that has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No. 850979).

[†]Hausdorff Center for Mathematics, University of Bonn, Germany

[‡]Saarbrücken Graduate School of Computer Science and Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

[§]Athena Research Center, Athens, Greece. Part of this work was done while the author was a postdoctoral researcher at the Hausdorff Center for Mathematics and the University of Bonn, Germany. The author was (partially) supported by the European Union's Horizon 2020 Research and Innovation programme, under the grant agreement No. 957345: "MORE".

1 Introduction

The Fréchet distance and its variants provide a versatile class of distance measures for parametrized curves as they occur in application areas such as trajectories of moving objects (e.g., vehicles, animals, or robots), outlines of shapes, signatures, gestures, and other types of time series from sensor data [SBL20, SLZ$^+$20]. This distance measure is very similar to the Hausdorff distance, which is defined for sets, except that it takes the ordering of points along the curve into account. At the same time, by assuming the equivalence class of all reparametrizations of a curve, it is robust to local irregularities in the parametrization of the curves (e.g., errors due to local time delays or irregular measurements). An intuitive definition of the distance measure is given as follows. Imagine traversing the two curves at independent and varying speeds from the beginning to the end, and consider the maximum (Euclidean) distance that the two positions can maintain throughout the traversal without backtracking along the curves. Minimizing over all possible traversals yields the Fréchet distance of the two curves.

Due to the popularity of the distance measure for trajectory analysis and data analysis applications, many heuristics and algorithm engineering solutions have been proposed to speed up the distance computation and similarity retrieval [dBGM17, BB17, DV17, BDvDM17, BKN19, GHPS20]. A fundamental task in this area is near neighbor searching: Preprocess n curves into a data structure, such that we can query this data structure with a curve and retrieve an input curve that has small distance to the query curve. This problem has been studied intensively [dBIG13, dBMO17, Ind02, AD18, EP20, FFK20, DS17, Mir20, DP21] and, for the discrete version of the Fréchet distance, these efforts lead to a simple and likely optimal data structure [FFK20]. However, for the more classic continuous version of the Fréchet distance, the computational complexity of near neighbor searching is still largely open, and seems very challenging to resolve.

Therefore, in this paper we focus on the special case of one-dimensional curves, which we also refer to as time series. We aim to resolve approximate near neighbor searching for this special case of the continuous Fréchet distance. We obtain strong lower bounds based on the Orthogonal Vectors Hypothesis in the regime of small approximation factors. More specifically, we differentiate a range of lower bounds for different approximation factors and preprocessing/query time. We show that our bounds are tight by devising data structures that asymptotically match the lower bounds in all cases considered. The new data structures improve upon the state of the art in several ways. For the same preprocessing and query time, we can improve the approximation factor from $(2+\varepsilon)$ to $(1+\varepsilon)$. For the same approximation factor $(2+\varepsilon)$, we get a better time complexity—in some cases we can even achieve linear preprocessing time and space.

1.1 Problem Definition

Let us first formally define the distance measure considered in this work.

Definition 1 (Fréchet distance)**.** *Given two curves $P, Q : [0,1] \mapsto \mathbb{R}^d$, their Fréchet distance is*

$$\mathrm{d_F}(P,Q) := \min_{f,g\in\mathcal{T}} \max_{t\in[0,1]} \|P(f(t)) - Q(g(t))\|_2,$$

where $\mathcal{T}$ is the set of all monotone and surjective functions from $[0,1]$ to $[0,1]$. For functions f and g that realize the minimum above, we define $\phi : [0,1] \to [0,1]^2$ with $\phi(t) = (f(t), g(t)), t \in [0,1]$ and we refer to ϕ as a realizing traversal of the two curves.

The central problem of this work is then defined as follows.

Definition 2 (c-Approximate Near Neighbors problem (c-ANN)). *The input consists of a set $\mathcal{P}$ of n curves in $\mathbb{R}^d$, each of complexity m, and a number $2 \leq k \leq m$. Given a distance threshold $\delta > 0$ and an approximation factor $c > 1$, preprocess $\mathcal{P}$ into a data structure such that for any query curve Q of complexity k, the data structure reports as follows:*

- *if $\exists P \in \mathcal{P}$ such that $\mathrm{d_F}(P, Q) \leq \delta$, then it returns $P' \in \mathcal{P}$ such that $\mathrm{d_F}(P', Q) \leq c\delta$,*
- *if $\forall P \in \mathcal{P}$, $\mathrm{d_F}(P, Q) \geq c\delta$ then it returns "no",*
- *otherwise, it either returns a curve $P \in \mathcal{P}$ such that $\mathrm{d_F}(P, Q) \leq c\delta$, or "no".*

The assumption that all input curves have the same number of vertices m and that the queries have k vertices is mostly to simplify presentation; all our data structures are easily generalized to allow input curves of complexity *at most* m and query curves of complexity *at most* k. Note, however, that we assume the input has size in $\Omega(nm)$ and that $2 \leq k \leq m$. The case $k = 1$ is a boundary case that is easier to solve; we ignore it throughout this paper.

1.2 State of the Art

We start by reviewing the state of the art for the *discrete* variant of the Fréchet distance. In the discrete Fréchet distance, the continuous traversal ϕ is replaced by a discrete traversal of the two point sequences, we refer to [FFK20] for the exact definition. The currently best known data structure for $(1 + \varepsilon)$-ANN under the discrete Fréchet distance is by Filtser et al. [FFK20]. Their data structure uses space in $n \cdot \mathcal{O}(1/\varepsilon)^{kd} + \mathcal{O}(nm)$ and query time in $\mathcal{O}(kd)$, where k denotes the complexity of the query (measured in the number of vertices), m denotes the complexity of an input curve and n denotes the number of input curves. It is an interesting question whether the same bounds can be obtained for the continuous Fréchet distance. At first glance, the discrete and continuous variants of the Fréchet distance seem very similar, but there is an important difference: while the metric space of bounded complexity curves under the discrete Fréchet distance has bounded doubling dimension, this does not hold in the continuous case, even when restricted to polygonal curves of constant complexity [DKS16]. (A metric space has doubling dimension at most d if any ball of any radius r can be covered by 2^d balls of radius $\frac{r}{2}$.) This immediately shows that the technique employed by Filtser et al., which effectively applies a doubling oracle to the metric balls centered at input curves (more specifically, simplifications thereof), does not directly extend to the continuous Fréchet distance, since such a doubling oracle cannot exist in this case.

So the *discrete* Fréchet distance has a simple ANN that seems optimal, but there is indication that for the *continuous* Fréchet distance resolving the time complexity of ANN is more challenging.

Note that it is possible to reduce the ANN problem for the continuous Fréchet distance to the ANN problem for the discrete Fréchet distance by subsampling along the continuous curves. However, it seems that this approach introduces an (otherwise avoidable) dependency on the arclength. In 2018, Driemel and Afshani [AD18] described data structures based on multi-level partition trees (using semi-algebraic range searching techniques) which can also be used for exact near neighbor searching under the continuous Fréchet distance. For n curves of complexity m in $\mathbb{R}^2$, their data structure uses space bounded by $n \cdot (\log\log n)^{\mathcal{O}(m^2)}$ and the query time is bounded by $\sqrt{n} \cdot (\log n)^{\mathcal{O}(m^2)}$. (If the input is restricted to curves in $\mathbb{R}$, these bounds can be slightly improved.) Recently, Driemel and Psarros [DP21] obtained bounds for the continuous Fréchet distance that are similar to the bounds of Filtser et al., albeit at the expense of a higher approximation factor and only for curves in $\mathbb{R}$. They present a $(5 + \varepsilon)$-ANN data structure which uses space in $n \cdot \mathcal{O}\left(\frac{1}{\varepsilon}\right)^k + \mathcal{O}(nm)$ and has query time in $\mathcal{O}(k)$, and a $(2 + \varepsilon)$-ANN data structure, which uses space in $n \cdot \mathcal{O}\left(\frac{m}{k\varepsilon}\right)^k + \mathcal{O}(nm)$ and has query time in $\mathcal{O}\left(k \cdot 2^k\right)$. Even more efficient data structures can be obtained at the expense of an even larger approximation factor, see the work of Driemel, Silvestri, and Psarros [DS17] and [DP21] which uses locality-sensitive hashing. In these results neither the space nor the query time is exponential in the complexity of the curves (neither input nor query), but the approximation factor is linear in the query complexity k.

(Unconditional) lower bounds

Given these results, one may ask whether the cited bounds are optimal for the respective approximation factor that they guarantee. We review some efforts in answering this question and discuss the limitations of the current techniques. Driemel and Psarros [DP21, DP20] approach this question using a technique by Miltersen [Mil94] for proving cell-probe lower bounds. Their results indicate that any data structure answering a query for a near neighbor under the continuous Fréchet distance by using only a constant number of probes to memory cells cannot have a space usage that is independent of the arclength of the input curves (assuming a query radius of 1). In addition, their bounds indicate that, in some cases, space exponential in the complexity of the query k is necessary. However, these bounds hold only for data structures that use a constant number of probes to memory cells for answering a query, while we would also be interested in data structures that use higher query time, such as $\mathcal{O}(k)$ or $\mathcal{O}(\log n)$. A different lower bound technique was used by Driemel and Afshani [AD18]. They show a lower bound in the pointer model of computation on the space-time tradeoff for *range reporting* under the Fréchet distance. In this problem, all curves contained inside the query radius need to be output by the query. The resulting lower bound matches the above cited upper bounds even up to the asymptotic number of factors of $\log(n)$. The proof uses a construction of input curves in $\mathbb{R}^2$ and a set of queries, such that the intersection of any two query results has small volume while the queries themselves have large volume. The main drawback of this technique is that, being a volume argument, it inherently uses the fact that all curves inside the query need to be returned and therefore it cannot easily be applied in the near neighbor setting.

Conditional lower bounds

The recent rise of fine-grained complexity has also lead to a renewed interest in conditional lower bounds for nearest neighbor data structure problems, see, e.g. [AW15, ARW17, Rub18, CGL+19, CW19]. These lower bounds are for the offline version of the data structure problem, by considering the total time needed for preprocessing and performing a number of queries. They are obtained in a similar way as NP-hardness, specifically via reductions from some fine-grained hypothesis such as the *Strong Exponential Time Hypothesis (SETH)* [IP01] or the *Orthogonal Vectors Hypothesis (OVH)* [Wil05]. In the Orthogonal Vectors problem we are given two sets of vectors $A, B \subseteq \{0,1\}^d$ of size n and ask whether there exist two vectors $a \in A, b \in B$ such that $\langle a, b\rangle = 0$. The hypothesis postulates that for any constant $\varepsilon > 0$ there exists a constant $c > 0$ such that there is no algorithm solving the Orthogonal Vectors problem in time $\mathcal{O}(n^{2-\varepsilon})$ in dimension $d = c\log n$. It should be noted that OVH is at least as believable as SETH, because SETH implies OVH [Wil05]. As an example, based on the OV-hardness of bichromatic Euclidean closest pair [AW15] and reducing via a variant of OV with unbalanced size $|A| \ll |B|$ [AW14], one can show that for any $\varepsilon, \beta > 0$ there is no data structure for Euclidean nearest neighbors on n points in $\mathbb{R}^d$ with preprocessing time $\mathcal{O}(n^\beta)$ and query time $\mathcal{O}(n^{1-\varepsilon})$, in some dimension $d = c\log n$. This rules out any sublinear query time for any data structure with polynomial preprocessing time, unless OVH fails.

For computing the Fréchet distance of two polygonal curves there is a tight conditional lower bound [Bri14], also for the one-dimensional case [BM16, BOS19]. However, thus far, there seems to be no comprehensive study of conditional lower bounds for the corresponding data structure

Table 1: Known upper bounds and our results. For the discrete case we only cite the best known result. The space complexity is implicitly bounded by the preprocessing time in each case. Our preprocessing time is randomized; the bounds can be derandomized at the cost of a factor $\log n$ in preprocessing and query time (by using search trees instead of perfect hashing).

Fréchet distance	Approximation	Preprocessing Time	Query Time	Reference
discrete, dD	$(1+\varepsilon)$-ANN	$nm \cdot \big(\mathcal{O}(\frac{1}{\varepsilon})^{dk} + \mathcal{O}(d \log m)\big)$	$\mathcal{O}(dk)$	[FFK20]
continuous, 1D	$(2+\varepsilon)$-ANN	$n \cdot \mathcal{O}(\frac{m}{k\varepsilon})^k$	$\mathcal{O}(1)^k$	[DP21]
	$(5+\varepsilon)$-ANN	$n \cdot \mathcal{O}(\frac{1}{\varepsilon})^k + \mathcal{O}(nm)$	$\mathcal{O}(k)$	[DP21]
continuous, 1D	$(1+\varepsilon)$-ANN	$n \cdot \mathcal{O}(\frac{m}{k\varepsilon})^k$	$\mathcal{O}(1)^k$	Theorem 27
	$(2+\varepsilon)$-ANN	$n \cdot \mathcal{O}(\frac{m}{k\varepsilon})^k$	$\mathcal{O}(k)$	Theorem 29
	$(2+\varepsilon)$-ANN	$n \cdot \mathcal{O}(\frac{1}{\varepsilon})^k + \mathcal{O}(nm)$	$\mathcal{O}(1)^k$	Theorem 31
	$(2+\varepsilon)$-ANN	$\mathcal{O}(nm)$	$\mathcal{O}(\frac{1}{\varepsilon})^{k+2}$	Theorem 33
	$(3+\varepsilon)$-ANN	$n \cdot \mathcal{O}(\frac{1}{\varepsilon})^k + \mathcal{O}(nm)$	$\mathcal{O}(k)$	Theorem 35

Table 2: Our conditional lower bounds. Each row gives an approximation ratio and a setting of k and m where any poly(n) preprocessing time and $\mathcal{O}(n^{1-\varepsilon'})$ query time cannot be achieved simultaneously. The constants $\varepsilon, \varepsilon', c$ are quantified as $\forall \varepsilon, \varepsilon' > 0\colon \exists c > 0$. By $f(n) \ll g(n)$ we mean $f(n) = o(g(n))$. We refer to the respective theorems in Section 8 for the exact statements.

Fréchet dist.	Approx.	Preproc.	Query	Parameter Setting	Reference
continuous, 1D	$2-\varepsilon$	poly(n)	$\mathcal{O}(n^{1-\varepsilon'})$	$1 \ll k \ll \log n$ and $m = k \cdot n^{c/k}$	Thm. 51
	$3-\varepsilon$	poly(n)	$\mathcal{O}(n^{1-\varepsilon'})$	$m = k = c \log n$	Thm. 52
continuous, 2D	$3-\varepsilon$	poly(n)	$\mathcal{O}(n^{1-\varepsilon'})$	$1 \ll k \ll \log n$ and $m = k \cdot n^{c/k}$	Thm. 53

problem. We want to close this gap and show tight bounds for the case of one-dimensional curves. These are similar in spirit to the Euclidean nearest neighbor lower bounds discussed above.

1.3 Our Results

For the discrete Fréchet distance the ANN problem is by now well understood, but the continuous Fréchet distance remains very challenging. Therefore, in this paper we focus on the important special case of one-dimensional curves, which arise in various domains such as finance and signal processing, where they are typically called "time series". We give several new data structure bounds for the problem of approximate near neighbor searching for one-dimensional curves under the continuous Fréchet distance. Table 1 provides an overview of our upper bounds, compared to known results. In the second part of our paper, we show that most of these upper bounds are tight under the Orthogonal Vectors Hypothesis, when viewed as offline problems where the input and the set of queries are given in advance. To obtain these lower bounds, we introduce a novel OV-hard variant of Orthogonal Vectors in which one set contains sparse vectors, i.e., vectors that only contain few 1s; this problem may be of independent interest. Table 2 gives an overview of our lower bound results. To argue that most of our upper bounds are tight, we consider the following general scenario:

> *Suppose we have an α-ANN for some fixed constant α, we run its preprocessing on a data set of n curves, and then we run n queries.*

In particular, consider this scenario for the following three ranges of α.

- $1 < \alpha < 2$: Using our $(1+\varepsilon)$-ANN, this scenario takes total time $n \cdot \mathcal{O}(\frac{m}{k\varepsilon})^k$, which simplifies to $n \cdot \mathcal{O}(\frac{m}{k})^k$ since $\varepsilon = \alpha - 1$ is fixed. Assuming OVH, our first lower bound shows that this running time cannot be improved to $n \cdot f(k) \cdot (\frac{m}{k})^{o(k)}$ for any function f, for the following reason. Pick $k = k(n)$ sufficiently small such that $f(k) = n^{o(1)}$. Pick $m = k \cdot n^{c/k}$, so that $(\frac{m}{k})^{o(k)} = (\frac{k \cdot n^{c/k}}{k})^{o(k)} = n^{o(1)}$. Then the total running time would be $n \cdot f(k) \cdot (\frac{m}{k})^{o(k)} = n^{1+o(1)}$, which contradicts that either the preprocessing time is superpolynomial or the query time near-linear, as stated in Theorem 51. This shows that the factor $(\frac{m}{k})^{\Theta(k)}$ in our running time is necessary. Our second lower bound shows that the running time cannot be improved to $n \cdot (\frac{m}{k})^{f(k)} \cdot 2^{o(k)}$ for any function f, as for $m = k = c \log n$ the total time would become $n \cdot (\frac{m}{k})^{f(k)} \cdot 2^{o(k)} = n \cdot 1^{f(k)} \cdot n^{o(1)} = n^{1+o(1)}$, which contradicts that either the preprocessing time is superpolynomial or the query time near-linear, as stated in Theorem 52. This shows that the factor $\mathcal{O}(1)^k$ in our query time is necessary. In this sense, the running time of our $(1+\varepsilon)$-ANN is tight.
- $2 < \alpha < 3$: By using our second or third $(2+\varepsilon)$-ANN (Theorem 31 or 33) we solve this scenario in total time $\mathcal{O}(nm) + n \cdot \mathcal{O}(\frac{1}{\varepsilon})^{k+2}$, which simplifies to $\mathcal{O}(nm) + n \cdot \mathcal{O}(1)^k$ since $\varepsilon = \alpha - 2$ is fixed. Assuming OVH, our second lower bound shows that this cannot be improved to time $n \cdot (\frac{m}{k})^{f(k)} \cdot 2^{o(k)}$ for any function f, as for $m = k = c \log n$ we would obtain a total time of $n \cdot (\frac{m}{k})^{f(k)} \cdot 2^{o(k)} = n \cdot 1^{f(k)} \cdot n^{o(1)} = n^{1+o(1)}$, which contradicts that either the preprocessing time is superpolynomial or the query time near-linear, as stated in Theorem 52. This shows that the factor $\mathcal{O}(1)^k$ in our running time is necessary. In this sense, the running time of our $(2+\varepsilon)$-ANNs from Theorems 31 and 33 are tight. (Our $(2+\varepsilon)$-ANN from Theorem 29 is not tight in this sense, but it realizes a different tradeoff between preprocessing and query time.)
- $\alpha > 3$: In this range, our ANNs still require exponential time in terms of k, but we cannot hope for a tight lower bound using the current techniques. This is due to a fundamental limitation of proving inapproximability factor > 3 for a metric problem, cf. e.g. [Rub18, Open Question 3]. For this reason, we have no tight lower bounds in this range.

1.4 Technical Overview

The high-level view of our data structures employs a well-known technique: exhaustively enumerate a strategic subset of the query space with a set of "candidate" query curves during preprocessing, and store the answers to these candidate queries in a dictionary. During query time, we apply a simple transformation to the query curve (such as rounding vertices to a scaled integer grid) and look up the answer in the dictionary. Filtser et al. [FFK20] used this technique for the discrete Fréchet distance and Driemel and Psarros [DP21] showed that it can also be applied for the continuous Fréchet distance of one-dimensional curves. A particular challenge that appears in the continuous case is that the doubling dimension can be unbounded, even if the complexity of the curves is small. Intuitively, what can happen is that the query contains some small noise that appears in the middle of a long edge. The continuous Fréchet distance—being robust to this noise—may match these short edges to the interior of a long edge on the near neighbor input curve. However, we cannot afford to generate all possible noisy query curves of this type, since this would introduce a dependency on the arclength in our time and space bounds. Driemel and Psarros overcome this challenge with the use of signatures, which allow to "guess" the approximate shape of a query curve within some approximation factor. The idea is that the signature acts as a "low-pass" filter that eliminates the noisy short edges. However, this is a delicate process as the signature may eliminate too many edges on one of the curves (either on the near neighbor or on the query curve) leading to the near neighbor being missed during query time. In addition, the process may introduce false-positives, hence the high approximation factor of $(5 + \varepsilon)$ in the result of [DP21].

We see our contributions as three-fold:

1. Our first contribution is to improve the approximation factors of Driemel and Psarros [DP21] while staying within the same time bounds, cf. Table 1 for a comparison.

   (a) For Theorem 35, we use almost the same algorithm as Driemel and Psarros, but combine this with a more careful analysis based on new observations on the Fréchet distance of approximately monotone curves. As a result, we can achieve a $(3 + \varepsilon)$-approximation within the same time bounds as the previous $(5 + \varepsilon)$-ANN.

   (b) In Theorem 27 we even achieve an approximation factor of $(1 + \varepsilon)$ within the same time bounds as the previous $(2 + \varepsilon)$-ANN. To achieve this result, we introduce the concept of straightenings in Section 3. Straightenings share some properties of signatures, but they provide a more refined approximation, leading to fewer false positives. They allow us to "guess" the shape of a query curve up to approximation factor $(1 + \varepsilon)$.

   We derive useful properties of both signatures and straightenings. Central to our analysis is the concept of δ-visiting orders, which we introduce in Section 3 and analyze in Section 7.

2. Our second contribution is a range of data structures for the $(2 + \varepsilon)$-ANN which together provide a tradeoff between preprocessing time and query time (see Theorems 29, 31, and 33). In each case, the preprocessing time implicitly bounds the number of candidates that are generated and therefore the size of the dictionary used by the data structure. Thus, these data structures also achieve a tradeoff between space and query time. An important observation that leads to this result is that the enumeration of candidates can be "dualized" and then be shifted from the preprocessing time to the query time. In the extreme case, this allows us to design a data structure that has linear preprocessing time and space, by performing most of the candidate generation during query time, see Theorem 33 for the exact result.

3. Given the diverse range of upper bounds, it is natural to ask if these bounds can be improved. Our third main contribution is to show that most of our upper bounds are tight under the Orthogonal Vectors Hypothesis. All known OV-based hardness results for the Fréchet distance encode each of the dimensions using at least one vertex, thus transforming d-dimensional vectors into curves of length $k = \Omega(d)$. Since OVH postulates a lower bound in dimension $d = c \log n$, it is thus natural to prove OV-based lower bounds for curves of length $k = c \log n$. Our lower bound in Theorem 52 handles this setting, cf. Table 2.

   However, for some of our lower bounds we require $k = o(\log n)$, as this is necessary to rule out time $(m/k)^{o(k)}$. Surprisingly, we overcome the barrier of using at least one vertex per dimension. Specifically, we prove OV-based lower bounds for any $1 \ll k \ll \log n$, see Theorem 51. For this, we use two crucial observations: (i) it is possible to only encode the 1s of one vector set, while the 0s do not require any additional vertices on the curve, and (ii) we can show hardness of a variant of OV where one set contains only sparse vectors, i.e., vectors with a very small number of 1s. See Theorem 51 for the hardness result we obtain in this case. Interestingly, a similar construction is also possible for $(3-\varepsilon)$-ANN for two-dimensional curves, see Theorem 53.

Organization

In Section 2 we define the notation and state some known facts and observations. In Section 3 we define key concepts, and we present their properties and our main technical lemmas. Our data structures are described and analyzed in Sections 4, 5, and 6. In Section 7 we prove our main technical lemmas. In Section 8 we present our conditional lower bounds.

2 Preliminaries

For any positive integer n, we define $[n] \coloneqq \{1, \dots, n\}$. For any two points $p, q \in \mathbb{R}^d$, $\overline{pq}$ denotes the directed line segment connecting p with q in the direction from p to q. Any sequence of points $p_1, \dots, p_m \in \mathbb{R}^d$ defines a polygonal curve formed by the ordered line segments $\overline{p_i p_{i+1}}$. We call the points p_i the *vertices* of the curve and the line segments $\overline{p_i p_{i+1}}$ the *edges*. The resulting curve can be viewed as a continuous function $P\colon [0,1] \mapsto \mathbb{R}^d$. For $d = 1$, we may refer to the curve as a *one-dimensional curve* or as a *time series*. We define the *complexity* of a polygonal curve P as the number of its vertices and we denoted it by $|P|$. We say a polygonal curve is *degenerate* if there are three consecutive vertices p, q, r, such that q lies on the line segment $\overline{pr}$. In this case, we call q a degenerate vertex of this curve. Given a sequence of points $p_1, \dots, p_m$, we can define a non-degenerate curve by omitting degenerate vertices. We denote the resulting curve by $\langle p_1, \dots, p_m \rangle$. Note that for one-dimensional curves, the vertices of the resulting non-degenerate curve are the extrema of the function. For any two $0 \leq t_a < t_b \leq 1$ and any curve P, we denote by $P[t_a, t_b]$ the subcurve of P starting at $P(t_a)$ and ending at $P(t_b)$. For any two curves P, Q, with vertices $p_1, \dots, p_a$ and $p_b, \dots, p_m$, respectively, $P \circ Q$ denotes the polygonal curve $\langle p_1, \dots, p_a, p_b, \dots p_m \rangle$, that is the *concatenation* of P and Q. For n polygonal curves $P_1, \dots, P_n$, we denote by $\bigcirc_{i=1}^{n} P_i$ the concatenation $P_1 \circ P_2 \circ \dots \circ P_n$. Given a polygonal curve $P = \langle p_1, \dots, p_m \rangle$ and a point x in $\mathbb{R}^d$, we define the translated curve as $P + x \coloneqq \langle p_1 + x, \dots, p_m + x \rangle$. For a point $x \in \mathbb{R}^d$ and a polygonal curve P, we use the notation $x \in P$ to indicate that there exists a $t \in [0,1]$ such that $P(t) = x$. Let $\mathcal{G}_\varepsilon \coloneqq \{i \cdot \varepsilon \mid i \in \mathbb{Z}\}$ be the regular grid with side-length $\varepsilon > 0$.

We will use the following known observations (see also [BBW08] and [DHP13]).

Observation 3. *For any two line segments $X = \overline{ab}$, $Y = \overline{cd}$ it holds that $\mathrm{d_F}(X, Y) = \max\{\|a - c\|, \|b - d\|\}$.*

Observation 4. *Let two polygonal curves $Q : [0,1] \mapsto \mathbb{R}^d$ and $P : [0,1] \mapsto \mathbb{R}^d$ be the concatenations of two subcurves each, $Q = Q_1 \circ Q_2$ and $P = P_1 \circ P_2$. Then it holds that $\mathrm{d_F}(P, Q) \leq \max\{\mathrm{d_F}(Q_1, P_1), \mathrm{d_F}(Q_2, P_2)\}$.*

Observation 5. *Let Q be a line segment and let P be a curve with $\mathrm{d_F}(P, Q) \leq \delta$. Let P' be a curve that is formed from a subsequence of the vertex sequence of P including the first and last vertex of P. Then, $\mathrm{d_F}(P', Q) \leq \delta$.*

We also make use of an algorithm by Alt and Godau [AG95] for deciding whether the Fréchet distance between two polygonal curves exceeds a given threshold.

Theorem 6 ([AG95])**.** *There is an algorithm which, given polygonal curves P, Q and a threshold parameter $\delta > 0$, decides in $\mathcal{O}(|P| \cdot |Q|)$ time whether $\mathrm{d_F}(P, Q) \leq \delta$.*

Our data structures can be implemented to work on the Word-RAM and under certain assumptions on the Real-RAM, as discussed next. Central to our approach is the use of a dictionary, which we define as follows.

Definition 7 (Dictionary)**.** *A* dictionary *is a data structure which stores a set of (key, value) pairs and when presented with a key, either returns the corresponding value, or returns that the key is not stored in the dictionary.*

In the Word-RAM model, such a dictionary can be implemented using perfect hashing. For storing a set of n (key,value) pairs, where the keys come from a universe U^k, perfect hashing provides us with a dictionary using $\mathcal{O}(n)$ space and $\mathcal{O}(k)$ query time which can be constructed in $\mathcal{O}(n)$ expected time [FKS84]. During look-up, we compute the hash function in $\mathcal{O}(k)$ time, we access the corresponding bucket in the hashtable in $\mathcal{O}(1)$ time and check if the key stored there is equal to the query in $\mathcal{O}(k)$ time. This gives an efficient randomized implementation of dictionaries. Alternatively, we can use balanced binary search trees and pay an additional $\log n$ factor in preprocessing and query time of the dictionary. This deterministic algorithm also works in the Real-RAM model, if we assume that the floor function can be computed in constant time—a model which is often used in the literature [HP11]. In the Word-RAM model, we use the standard assumption that the word size is logarithmic in the size of the input, and we ensure that all numbers (vertices of the time series, results of intermediate computations, etc.) are restricted to be of the form a/b where a is an integer in $[-(nm)^{\mathcal{O}(1)}, (nm)^{\mathcal{O}(1)}]$ and $b = (nm)^{\mathcal{O}(1)}$.

3 Simplifications, signatures, and straightenings

In this section we state the main definitions and lemmas. To allow for an easier understanding of our results, we then already describe our algorithms and prove correctness using these lemmas. In Section 7 we then give the proofs of the lemmas presented in the current section.

3.1 Definitions

Let us start with two basic definitions.

Definition 8. *We say a curve $P : [0, 1] \to \mathbb{R}$ is* δ-monotone *if one of the following statements holds:*
(i) $\forall\ t < t' \in [0, 1] : P(t') \geq P(t) - \delta$,
(ii) $\forall\ t < t' \in [0, 1] : P(t') \leq P(t) + \delta$.
More specifically, we say the curve is δ-monotone increasing *in case (i) and* δ-monotone decreasing *in case (ii). Note that a curve can be both δ-monotone increasing and decreasing at the same time. In addition, we may say P is δ-monotone with respect to a directed edge $\overline{ab}$, if $a \leq b$ in case (i) and if $b \leq a$ in case (ii).*

Definition 9. *The δ-range of a point $p \in \mathbb{R}$ is the interval $B(p, \delta) = [p - \delta, p + \delta]$. The δ-range of a curve P is the interval $B(P, \delta) = \bigcup_{x \in P} B(x, \delta)$.*

We now define the notion of *simplification* that we use in this work.

Definition 10 (δ-simplification)**.** *Given a curve $P: [0,1] \mapsto \mathbb{R}^d$, a δ-simplification is a curve $P': [0,1] \mapsto \mathbb{R}^d$ that is given as $P' = \langle P(t_1), \dots, P(t_\ell) \rangle$ for a sequence of values $0 = t_1 < \dots < t_\ell = 1$, such that each $P(t_i)$ is a vertex of P, P' is non-degenerate, and*

$$\mathrm{d_F}(\overline{P(t_i)P(t_{i+1})}, P[t_i, t_{i+1}]) \leq \delta, \text{ for all } 1 \leq i < \ell. \tag{1}$$

We also refer to (1) as the *locality property.* Furthermore, note that if P' is a δ-simplification of P, then $\mathrm{d_F}(P, P') \leq \delta$ and the complexity of P' is at most the complexity of P. Note that the vertices of a δ-simplification P' give us a natural partition of P. Furthermore, we want to highlight that our definition of a simplification is one out of many definitions that are used in literature. In particular, in other work curves which are degenerate or non vertex-restricted are also called simplifications. Now we define some properties that a simplification can or must have.

Observation 11 (direction-preserving property)**.** *For any δ-simplification $P' = \langle P(t_1), \dots, P(t_\ell) \rangle$ of a curve $P: [0,1] \mapsto \mathbb{R}$ and any index i, the subcurve $P[t_i, t_{i+1}]$ is 2δ-monotone with respect to $\overline{P(t_i)P(t_{i+1})}$.*

Definition 12 (vertex-range-preserving property)**.** *Let $P' = \langle P(t_1), \dots, P(t_\ell) \rangle$ be a δ-simplification of a curve $P: [0,1] \mapsto \mathbb{R}$. We say P' is range-preserving on the vertex $P(t_i)$ if the following holds:*

(i) if $P(t_i)$ is a local maximum on P', then $P(t) \leq P(t_i)$ for all t in $[t_{i-1}, t_{i+1}]$, and

(ii) if $P(t_i)$ is a local minimum on P', then $P(t) \geq P(t_i)$ for all t in $[t_{i-1}, t_{i+1}]$.

We say P' is vertex-range-preserving, if it is vertex-range-preserving on all interior vertices.

Definition 13 (edge-range-preserving property)**.** *Let $P' = \langle P(t_1), \dots, P(t_\ell) \rangle$ be a δ-simplification of $P: [0,1] \mapsto \mathbb{R}$. We say that P' is edge-range-preserving on edge $\overline{P(t_i)P(t_{i+1})}$ if for any $t \in [t_i, t_{i+1}]$ it holds that $P(t) \in \overline{P(t_i)P(t_{i+1})}$. We say P' is edge-range-preserving if this condition holds for all edges of P'.*

Note that the vertex-range-preserving property is implied by the edge-range-preserving property, but not the other way around. However, the vertex-range preserving property implies the edge-range-preserving property on all edges except the first and the last edge.

Definition 14 (δ-edge-length property)**.** *We say that a one-dimensional curve $P = \langle p_1, \dots, p_m \rangle$ has the δ-edge-length property if*

- *$|p_1 - p_2| > \delta$ and $|p_{m-1} - p_m| > \delta$, and*
- *$|p_i - p_{i+1}| > 2\delta$ for all $i \in \{2, \dots, m-2\}$.*

Finally, we can define two of the main concepts that we use in our algorithms: δ-signatures and δ-straightenings. These two definitions help us to preprocess the input set of one-dimensional curves and the query curve in ways such that an efficient retrieval is possible.

Definition 15 (δ-signature)**.** *A δ-simplification P' of a one-dimensional curve P is a δ-signature if it has the δ-edge length property and is vertex-range-preserving.*

Definition 16 (δ-straightening)**.** *A δ-simplification P' of a one-dimensional curve P is a δ-straightening if it is edge-range-preserving.*

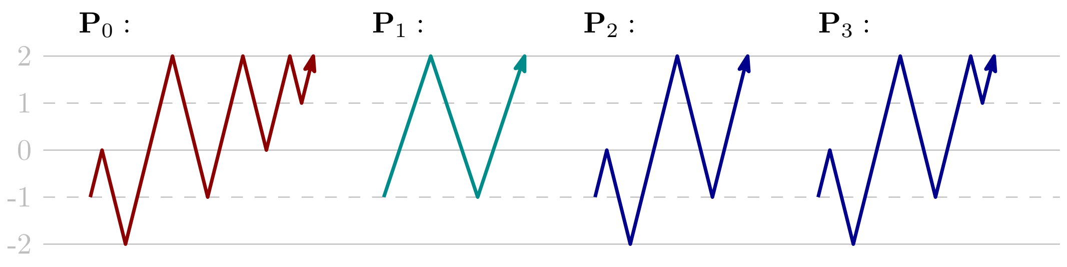

Figure 1: P_1 is a 1-signature of P_0, whereas P_2 and P_3 are 1-straightenings of P_0.

The above definition of a δ-signature is equivalent to the definition given in [DKS16]. For any $\delta > 0$ and any curve $P : [0,1] \mapsto \mathbb{R}$ of complexity m, a δ-signature of P can be computed in $\mathcal{O}(m)$ time [DKS16]. The δ-signature of a curve is unique under certain general-position assumptions, however we do not explicitly use this property in our proofs. Note that δ-straightenings are *not* unique. In fact, there can be many different δ-straightenings of the same curve, e.g., P itself is a δ-straightening of P for any $\delta > 0$. We give an example of a signature and different straightenings of the same curve in Figure 1.

We introduce the notion of visiting orders, which we will use to prove correctness of our data structures.

Definition 17. *Let $P : [0,1] \to \mathbb{R}$ and $Q : [0,1] \to \mathbb{R}$ be curves. Let $u_1, \ldots, u_\ell$ denote the ordered vertices of Q and let $v_1, \ldots, v_m$ denote the ordered vertices of P. A (partial) δ-**visiting order** of Q on P is a sequence of indices $i_1 \leq \cdots \leq i_\ell$, such that $|u_j - v_{i_j}| \leq \delta$ for each vertex u_j of Q.*

In particular, if we know that there exists a δ-visiting order of Q on P, then we can approximately "guess" Q from the vertex sequence of P, by enumerating all possible visiting orders of the vertices of P and for any fixed visiting order, enumerating all eligible grid sequences within the δ-ranges of these vertices.

Driemel, Krivosija and Sohler proved the following lemma (rephrased using δ-visiting orders).

Lemma 18 (Lemma 3.2 [DKS16])**.** *Let $P : [0,1] \to \mathbb{R}$ and $Q : [0,1] \to \mathbb{R}$ be curves and let P' be a δ-signature of P. If $\mathrm{d_F}(P,Q) \leq \delta$, then there exists a δ-visiting order of P' on Q.*

3.2 Main lemmas

In this section we present the main lemmas for signatures and straightenings that we will use in Sections 4 to 6. Their proofs are deferred to Section 7.

Most of our lemmas improve the basic triangle inequality $\mathrm{d_F}(P,Q) \leq \mathrm{d_F}(P,X) + \mathrm{d_F}(X,Q)$ in some situations involving signatures and straightenings.

Lemma 19. *Let $P : [0,1] \mapsto \mathbb{R}$ and $Q : [0,1] \mapsto \mathbb{R}$ be two curves and let Q' be any δ-straightening of Q. If $\mathrm{d_F}(P,Q') \leq \delta$ then $\mathrm{d_F}(P,Q) \leq \delta$.*

We would like to show the equivalent statement of Lemma 19 for signatures. However, as the example in Figure 2 shows, this is not possible. Instead, we show a slightly weaker bound in the following lemma.

Lemma 20. *Let $\delta = \delta' + \delta''$ for $\delta, \delta', \delta'' \geq 0$ and let $P : [0,1] \mapsto \mathbb{R}$ and $Q : [0,1] \mapsto \mathbb{R}$ be two curves. Let Q' be any δ'-signature of Q. If $\mathrm{d_F}(Q',P) \leq \delta$, $|Q(0) - P(0)| \leq \delta''$, and $|Q(1) - P(1)| \leq \delta''$, then $\mathrm{d_F}(P,Q) \leq \delta$.*

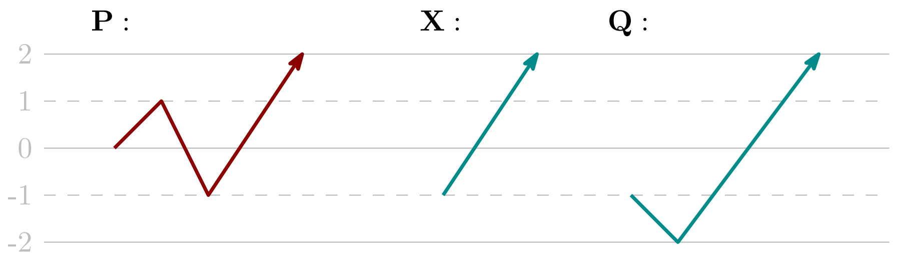

Figure 2: This example shows that an equivalent statement of Lemma 19 for signatures is not true. The curve $X = \langle -1, 2 \rangle$ is a 1-signature of $Q = \langle -1, -2, 2 \rangle$ and the curve $P = \langle 0, 1, -1, 2 \rangle$ has Fréchet distance 1 to X, but the Fréchet distance of P to Q is 2.

Note that Lemma 19 is much stronger than what we would get by merely applying the triangle inequality on the Fréchet distances on the curves P, Q and Q'. Lemma 20, although weaker, is still stronger than the bound we would get from the triangle inequality. To illustrate this we include the following corollary. Note that merely using triangle inequality would yield $\mathrm{d_F}(P, Q) \leq 6\delta$, instead of $\mathrm{d_F}(P, Q) \leq 3\delta$.

Corollary 21. *For one-dimensional curves P, Q let P' be a δ-signature of P, and let Q' be the 2δ-signature of Q. If $\mathrm{d_F}(P', Q') \leq 3\delta$ and $|P'(0) - Q'(0)| \leq \delta, |P'(1) - Q'(1)| \leq \delta$, then $\mathrm{d_F}(P, Q) \leq 3\delta$.*

Proof. Follows from applying of Lemma 20 twice. We first apply the lemma to P', Q' and P and obtain $\mathrm{d_F}(P, Q') \leq 3\delta$. In the second step, we apply the lemma to P, Q' and Q and obtain $\mathrm{d_F}(P, Q) \leq 3\delta$. □

The following lemma is used to show correctness for our $(1 + \varepsilon)$ and $(2 + \varepsilon)$-ANN.

Lemma 22. *Let $P : [0, 1] \mapsto \mathbb{R}$ and $Q : [0, 1] \mapsto \mathbb{R}$ be curves such that $\mathrm{d_F}(Q, P) \leq \delta$, there exists a δ-straightening Q' of Q which satisfies the following properties:*
(i) there exists a 11δ-visiting order of Q' on P, and
(ii) $\mathrm{d_F}(Q', P) \leq \delta$.

We use the following lemma to show correctness for our $(3 + \varepsilon)$-ANN. One part of the lemma statement, the existence of a 2δ-visiting orders, was already used in [DP20]. However, the resulting approximation factor of the ANN obtained there was $(5 + \varepsilon)$. In order to show correctness of our $(3 + \varepsilon)$-ANN, it is necessary to prove the bound of 3δ on the resulting Fréchet distance of the two signature curves. Note that the triangle inequality implies a bound of 4δ—which would not be sufficient for us.

Lemma 23. *For one-dimensional curves P, Q let P' be a δ-signature of P, and let Q' be a 2δ-signature of Q. If $\mathrm{d_F}(P, Q) \leq \delta$ then $\mathrm{d_F}(P', Q') \leq 3\delta$ and there exists a 2δ-visiting order of Q' on P'.*

4 $(1 + \varepsilon)$-Approximation

In this section, we show that there exists a $(1 + \varepsilon)$-ANN data structure for one-dimensional curves under the Fréchet distance, with space in $n \cdot \mathcal{O}(\frac{m}{k\varepsilon})^k$, expected preprocessing time in $nm \cdot \mathcal{O}(\frac{m}{k\varepsilon})^k$ and query time in $\mathcal{O}(k \cdot 2^k)$. We describe the data structure in Section 4.1 and we analyze its performance in Section 4.2.

4.1 The data structure

Data structure

We are given as input a set of one-dimensional curves $\mathcal{P}$, as sequences of vertices, the distance threshold $\delta > 0$, the approximation error $\varepsilon > 0$, and the complexity of the supported queries k. To discretize the query space, we use the grid $\mathcal{G}_{\varepsilon\delta/2}$ (recall that $\mathcal{G}_\varepsilon := \{i \cdot \varepsilon \mid i \in \mathbb{Z}\}$ is the regular grid with side-length ε). Let $\mathcal{H}$ be a dictionary which is initially empty. For each input one-dimensional curve $P \in \mathcal{P}$ we compute a set $\mathcal{C}' := \mathcal{C}'(P)$ which contains all curves Q such that: i) Q has complexity at most k, ii) all vertices of Q belong to $\mathcal{G}_{\varepsilon\delta/2}$, and iii) there is an $((11+\varepsilon/2)\delta)$-visiting order of Q on P. Formally,

$$\mathcal{C}' = \{\langle u_1, \ldots, u_\ell \rangle \mid \ell \leq k \text{ and } \exists(i_1, \ldots, i_\ell)(i_1 \leq \cdots \leq i_\ell \text{ and } (\forall j \in [\ell])(u_j \in B(p_{i_j}, (11+\varepsilon/2)\delta) \cap \mathcal{G}_{\varepsilon\delta/2}))\}.$$

Next, we filter $\mathcal{C}'$ to obtain the set $\mathcal{C}(P) = \{Q \in \mathcal{C}' \mid \mathrm{d_F}(Q, P) \leq (1+\varepsilon/2)\delta\}$. We store $\mathcal{C}(P)$ in $\mathcal{H}$ as follows: for each $Q \in \mathcal{C}(P)$, if Q is not already stored in $\mathcal{H}$, then we insert Q into $\mathcal{H}$, associated with a pointer to P.

The complete pseudocode for the preprocessing algorithm can be found in Algorithms 1 and 2. To achieve approximation factor $(1+\varepsilon)$, we run `preprocess`$(P, \delta, \varepsilon/2, k)$.

Query algorithm

Let Q be the query curve with vertices $q_1, \ldots, q_k$ and let $\varepsilon > 0$ be the approximation error. The query algorithm first enumerates all curves Q' such that

$$Q' \in \{\langle q_1, S, q_k \rangle \mid S \text{ is a subsequence of } q_2, \ldots, q_{k-1}\}.$$

For each such Q' we test whether it is a δ-straightening of Q. To this end, we first test if each shortcut taken in Q' is within distance δ from the corresponding subcurve of Q. Then we check for each shortcut if the corresponding subcurve of Q stays within range by testing all vertices of the subcurve one by one. If Q' is a δ-straightening of Q, then we snap the vertices of Q' to $\mathcal{G}_{\varepsilon\delta/2}$, to obtain a new curve Q'' and we probe $\mathcal{H}$: if Q'' is stored in $\mathcal{H}$, then we return its associated input curve $P \in \mathcal{P}$. If Q'' is not stored in $\mathcal{H}$, then we return "no".

The complete pseudocode for the query algorithm can be found in Algorithm 3. To achieve approximation factor $(1+\varepsilon)$, we run `query`$(Q, \delta, \varepsilon/2)$.

Algorithm 1 A call to `generate_orders`(m, k) returns all $(i_1, \ldots, i_\ell) \in [m]^\ell$, where $\ell \in [k]$ and such that $1 = i_1 \leq \cdots \leq i_\ell = m$. We assume $k \geq 2$.

```
1: procedure generate_orders($m \in \mathbb{N}$, $k \in \mathbb{N}$)
2:     $\mathcal{I}_2 \leftarrow \{(1, m)\}$
3:     for each $\ell = 3, \ldots, k$ do
4:         $\mathcal{I}_\ell \leftarrow \emptyset$
5:         for each $(i_1, \ldots, i_{\ell-1}) \in \mathcal{I}_{\ell-1}$ do                ▷ $i_{\ell-1} = m$
6:             for each $j = i_{\ell-2}, \ldots, m$ do
7:                 $\mathcal{I}_\ell \leftarrow \mathcal{I}_\ell \cup \{(i_1, \ldots, i_{\ell-2}, j, m)\}$
8:     return $\bigcup_{2 \leq \ell \leq k} \mathcal{I}_\ell$
```

Algorithm 2 Preprocessing algorithm. We call `preprocess` to build the data structure.

```
1: procedure preprocess(input set $\mathcal{P}$, $\delta > 0$, $\varepsilon > 0$, $k \in \mathbb{N}$)
2:     Initialize empty dictionary $\mathcal{H}$
3:     for each $P \in \mathcal{P}$ do
4:         $\mathcal{C}(P) \leftarrow$ generate_keys$(P, \delta, \varepsilon, k)$
5:         for each $Q \in \mathcal{C}(P)$ do
6:             if $Q$ not in $\mathcal{H}$ then
7:                 insert key $Q$ in $\mathcal{H}$, associated with a pointer to $P$
8: procedure generate_keys(curve $P$, $\delta > 0$, $\varepsilon > 0$, $k \in \mathbb{N}$)
9:     $\mathcal{C}' \leftarrow$generate_candidates$(P, \delta, (11 + \varepsilon), \varepsilon, k)$
10:    $\mathcal{C} \leftarrow \emptyset$
11:    for each $Q \in \mathcal{C}'$ do
12:        if $\mathrm{d_F}(P, Q) \leq (1 + \varepsilon)\delta$ then
13:            $\mathcal{C} \leftarrow \mathcal{C} \cup \{Q\}$
14:    return $\mathcal{C}$
15: procedure generate_candidates(curve $P$ with vertices $p_1, \ldots, p_m$, $\delta > 0$, $r > 0$, $\varepsilon > 0$, $k \in \mathbb{N}$)
16:    $\mathcal{S} \leftarrow \emptyset$, $\mathcal{C}' \leftarrow \emptyset$
17:    $\mathcal{I} \leftarrow$generate_orders$(m, k)$
18:    for each $(i_1, \ldots, i_\ell) \in \mathcal{I}$ do
19:        $\mathcal{S} \leftarrow \mathcal{S} \cup \prod_{j=1}^{\ell} B(p_{i_j}, r\delta) \cap \mathcal{G}_{\varepsilon\delta}$
20:    for each $\sigma \in \mathcal{S}$ do
21:        $\mathcal{C}' \leftarrow \mathcal{C}' \cup \{\langle\sigma\rangle\}$
22:    return $\mathcal{C}'$
```

Algorithm 3 Query algorithm

```
1: procedure query(curve $Q$ with vertices $q_1, \ldots, q_k$, $\delta > 0$, $\varepsilon > 0$)
2:     $\mathcal{I} \leftarrow$generate_orders$(k, k)$
3:     for each $(i_1, \ldots, i_\ell) \in \mathcal{I}$ do
4:         $flag \leftarrow 1$
5:         for $j = 1, \ldots, \ell - 1$ do
6:             if $\mathrm{d_F}(\overline{q_{i_j} q_{i_{j+1}}}, \langle q_{i_j}, \ldots, q_{i_{j+1}} \rangle) > \delta$ then        ▷ test $\delta$-simplification property
7:                 $flag \leftarrow 0$
8:             for each $t = i_j, \ldots, i_{j+1}$ do        ▷ test edge-range-preserving property
9:                 if $q_t \notin \overline{q_{i_j} q_{i_{j+1}}}$ then
10:                    $flag \leftarrow 0$
11:        if $flag = 1$ then
12:            $Q' \leftarrow \langle q_{i_1}, \ldots, q_{i_\ell} \rangle$        ▷ a $\delta$-straightening of $Q$
13:            $Q'' \leftarrow \left\langle \left\lfloor \frac{q_{i_1}}{\varepsilon\delta} \right\rfloor \cdot (\varepsilon\delta), \ldots, \left\lfloor \frac{q_{i_\ell}}{\varepsilon\delta} \right\rfloor \cdot (\varepsilon\delta) \right\rangle$        ▷ snap $Q'$ to $\mathcal{G}_{\varepsilon\delta}$
14:            if $Q''$ in $\mathcal{H}$ then
15:                return input curve $P$ associated with $Q''$ in $\mathcal{H}$
16:    return “no”
```

4.2 Analysis

In this section, we analyze the performance of our data structure.

Lemma 24. *For any curve P with vertices $p_1, \dots, p_m$, $\delta > 0$, $\varepsilon > 0$, $r \geq \varepsilon$, $k \in \mathbb{N}$, the procedure* ***generate_candidates***$(P, \delta, r, \varepsilon, k)$ *has running time in*

$$\binom{m+k-2}{k-2} \cdot \mathcal{O}\left(\frac{r}{\varepsilon}\right)^k.$$

Proof. The set $\mathcal{I}$ contains all sequences of indices $(i_1, \dots, i_\ell) \in [m]^\ell$ such that $\ell \leq k$, and $1 = i_1 \leq \cdots \leq i_\ell = m$. Let $\mathcal{I}_\ell$ be the subset of $\mathcal{I}$ containing the sequences of length ℓ as denoted in `generate_orders`. We first claim that `generate_orders`(m, k) runs in time $\mathcal{O}(|\mathcal{I}| \cdot k)$. To see that, consider any sequence of indices $s \in \mathcal{I}$. During the execution of `generate_orders`, s is added to the sets of indices (Line 7) only once. This step costs $\mathcal{O}(k)$, therefore the running time of `generate_orders`(m, k) is in $\mathcal{O}(|\mathcal{I}| \cdot k)$. Now, let $\mathcal{S}'$ be a multiset which contains all sequences (including duplicates) which are generated and inserted to $\mathcal{S}$ in all executions of Line 19 of `generate_candidates`. The running time of `generate_candidates`$(P, \delta, r, \varepsilon, k)$ is upper bounded by $\mathcal{O}(|\mathcal{S}'| \cdot k)$, because $|\mathcal{S}'| \geq |\mathcal{I}|$ and computing $\mathcal{C}'$ costs $\mathcal{O}(|\mathcal{S}'| \cdot k)$ time. We proceed by showing an upper bound on $|\mathcal{S}'|$.

Any sequence $(x_1, \dots, x_\ell) \in \mathcal{G}^\ell_{\varepsilon\delta}$, which is included in $\mathcal{S}'$, may appear in the computation taking place in Line 19 multiple times: once for each sequence of indices $(i_1, \dots, i_\ell) \in \mathcal{I}$ such that for each $j \in [\ell]$, $x_j \in B(p_{i_j}, r\delta)$. Notice that $|\mathcal{I}_\ell|$ is equal to the number of combinations of $\ell - 2$ objects taken (with repetition) from a set of size m, i.e. $|\mathcal{I}_\ell| = \binom{m+\ell-3}{\ell-2}$. Hence, by the Hockey-stick identity,

$$|\mathcal{I}| = \sum_{\ell=2}^{k} |\mathcal{I}_\ell| = \sum_{\ell=2}^{k} \binom{m+\ell-3}{\ell-2} = \sum_{\ell=0}^{k-2} \binom{m+\ell-1}{\ell} = \binom{m+k-2}{k-2}. \tag{2}$$

Using (2), we can bound $|\mathcal{S}'|$ as follows:

$$\begin{aligned} |\mathcal{S}'| &\leq \sum_{\ell=2}^{k} \sum_{(i_1,\dots i_\ell) \in \mathcal{I}_\ell} \left| \prod_{j=1}^{\ell} B(p_{i_j}, r\delta) \cap \mathcal{G}_{\varepsilon\delta} \right| \\ &\leq \sum_{\ell=2}^{k} |\mathcal{I}_\ell| \cdot \mathcal{O}\left(\frac{r}{\varepsilon}\right)^\ell \leq |\mathcal{I}| \cdot \mathcal{O}\left(\frac{r}{\varepsilon}\right)^k \leq \binom{m+k-2}{k-2} \cdot \mathcal{O}\left(\frac{r}{\varepsilon}\right)^k. \end{aligned}$$

Hence, the running time is $\mathcal{O}(|\mathcal{S}'| \cdot k) = \binom{m+k-2}{k-2} \cdot \mathcal{O}\left(\frac{r}{\varepsilon}\right)^k$. ◻

Lemma 25. *If* ***query***$(Q, \delta, \varepsilon/2)$ *returns an input curve $P \in \mathcal{P}$, then $\mathrm{d_F}(Q, P) \leq (1+\varepsilon)\delta$. If* ***query***$(Q, \delta, \varepsilon/2)$ *returns "no" then there is no $P \in \mathcal{P}$ such that $\mathrm{d_F}(Q, P) \leq \delta$.*

Proof. When `query`$(Q, \delta, \varepsilon/2)$ returns an input curve $P \in \mathcal{P}$, it must be that there exists a δ-straightening Q' of Q such that P is associated with Q'' in $\mathcal{H}$. This implies that $\mathrm{d_F}(Q'', P) \leq (1 + \varepsilon/2)\delta$. By the triangle inequality,

$$\mathrm{d_F}(Q', P) \leq \mathrm{d_F}(Q'', Q') + \mathrm{d_F}(Q'', P) \leq (1+\varepsilon)\delta.$$

Since Q' is a δ-straightening of Q, we have that $\mathrm{d_F}(Q', Q) \leq \delta$. Hence, by Lemma 19 applied on P, Q, Q' for distance threshold $(1+\varepsilon)\delta$, we obtain $\mathrm{d_F}(Q, P) \leq (1+\varepsilon)\delta$.

If `query`$(Q, \delta, \varepsilon/2)$ returns "no" then there is no δ-straightening Q' of Q such that Q'' is associated with an input curve in $\mathcal{H}$. Suppose, for the sake of contradiction, that there exists a curve $P \in \mathcal{P}$ such that $\mathrm{d_F}(Q, P) \leq \delta$. By Lemma 22, there exists a δ-straightening Q' of Q such that i) there exists an 11δ-visiting order of Q' on P and ii) $\mathrm{d_F}(Q', P) \leq \delta$. Let Q'' be the curve obtained by snapping vertices of Q' to the grid $\mathcal{G}_{\varepsilon\delta/2}$. By the triangle inequality, there exists a $((11 + \varepsilon/2)\delta)$-visiting order of Q'' on P and

$$\mathrm{d_F}(Q'', P) \leq \mathrm{d_F}(Q'', Q') + \mathrm{d_F}(Q', P) \leq (1 + \varepsilon/2)\delta.$$

Hence, $Q'' \in \mathcal{C}(P)$ and Q'' is associated with some input curve P' in $\mathcal{H}$. This leads to contradiction and we conclude that if `query`$(Q, \delta, \varepsilon/2)$ returns "no" then there is no curve $P \in \mathcal{P}$ such that $\mathrm{d_F}(P, Q) \leq \delta$. □

Lemma 26. *For any query curve Q of complexity k, $\delta > 0$, $\varepsilon > 0$, **query**(Q, δ, ε) runs in time $\mathcal{O}(k \cdot 2^k)$.*

Proof. Let $q_1, \ldots, q_k$ be the vertices of Q. We enumerate all sequences starting with q_1, followed by any possible subsequence of $q_2, \ldots, q_{k-1}$ and ending with q_k. There are at most 2^{k-2} such sequences, and for each one of them we test whether it defines a δ-straightening of Q. This is done in two steps: we first test if each shortcut is within distance δ from the corresponding subcurve, and then we decide if the edge-range-preserving property is satisfied. Computing the Fréchet distance between a shortcut and the original subcurve costs linear time in the complexity of the subcurve by Theorem 6. Hence, we can decide in $\mathcal{O}(k)$ time if the sequence in question defines a δ-simplification of Q. To decide if the edge-range-preserving property is satisfied, we check for each shortcut if the corresponding subcurve stays within range by testing all of its vertices one by one. Therefore, this step also costs $\mathcal{O}(k)$ time. Since we employ perfect hashing, each probe to $\mathcal{H}$ costs $\mathcal{O}(k)$ time. We can also check in $\mathcal{O}(k)$ time if the answer returned by $\mathcal{H}$ is the one we are searching for. Hence, the overall query time is in $\mathcal{O}(k \cdot 2^k)$.

□

Theorem 27. *Let $\varepsilon \in (0, 1]$. There is a data structure for the $(1 + \varepsilon)$-ANN problem, which stores n one-dimensional curves of complexity m and supports query curves of complexity k, uses space in $n \cdot \mathcal{O}\left(\frac{m}{k\varepsilon}\right)^k$, needs $\mathcal{O}(nm) \cdot \mathcal{O}\left(\frac{m}{k\varepsilon}\right)^k$ expected preprocessing time and answers a query in $\mathcal{O}(k \cdot 2^k)$ time.*

Proof. The data structure is described in Section 4.1. Correctness follows from Lemma 25. The bound on the query time follows from Lemma 26. It remains to analyze the running time of `preprocess`$(\mathcal{P}, \delta, \varepsilon/2, k)$ and the space complexity of the data structure.

By Lemma 24, for any $P \in \mathcal{P}$, the running time needed to compute $\mathcal{C}'$ is upper bounded by $\binom{m+k-2}{k-2} \cdot \mathcal{O}\left(\frac{1}{\varepsilon}\right)^k = \mathcal{O}\left(\frac{m}{k\varepsilon}\right)^k$. Hence, for each $P \in \mathcal{P}$, $|\mathcal{C}(P)| = \mathcal{O}\left(\frac{m}{k\varepsilon}\right)^k$. Therefore, the space required for each input curve $P \in \mathcal{P}$ is upper bounded by $\mathcal{O}(|\mathcal{C}(P)| \cdot k)$. Computing $\mathcal{C}(P)$ costs $\mathcal{O}(|\mathcal{C}'| \cdot mk) = \mathcal{O}\left(\frac{m}{k\varepsilon}\right)^k \cdot \mathcal{O}(m)$ time, because we need to decide for each curve $Q \in \mathcal{C}'$, whether its Fréchet distance from P is at most $(1+\varepsilon/2)\delta$, which can be done in $\mathcal{O}(|Q| \cdot |P|)$ time using Theorem 6. Assuming perfect hashing for $\mathcal{H}$, the overall expected preprocessing time is in $\mathcal{O}(nm) \cdot \mathcal{O}\left(\frac{m}{k\varepsilon}\right)^k$ and the space usage is in $\mathcal{O}(n) \cdot \mathcal{O}\left(\frac{m}{k\varepsilon}\right)^k$. □

5 $(2 + \varepsilon)$-Approximation

In this section we present three $(2 + \varepsilon)$-ANN data structures with different tradeoffs between preprocessing and query time.

5.1 Fast query algorithm

In this section, we propose a data structure for the $(2+\varepsilon)$-ANN problem, with query time in $\mathcal{O}(k)$. The space complexity and the preprocessing time are the same as in the $(1+\varepsilon)$-ANN data structure of Theorem 27.

Data structure

We are given as input a set of one-dimensional curves $\mathcal{P}$, as sequences of vertices, the distance threshold $\delta > 0$, the approximation error $\varepsilon > 0$ and the complexity of the supported queries k. The data structure is exactly the same as in Section 4. To build it, we call `preprocess`$(P, \delta, \varepsilon/2, k)$, as defined in Algorithm 2, in Section 4.1. Let $\mathcal{H}$ be the resulting dictionary, constructed by `preprocess`$(P, \delta, \varepsilon/2, k)$.

Query algorithm

Let Q be the query curve with vertices $q_1, \ldots, q_k$ and let $\varepsilon > 0$ be the approximation error. The query algorithm first computes a δ-signature Q' of Q, and then it snaps the vertices of Q' to the grid $\mathcal{G}_{\varepsilon\delta/2}$, to obtain a curve Q''. If Q'' is stored in $\mathcal{H}$, then we return its associated input curve $P \in \mathcal{P}$, otherwise we return "no". The query algorithm is implemented in `query2`, which can be found in Algorithm 4. To achieve approximation factor $2+\varepsilon$, we run `query2`$(Q, \delta, \varepsilon/2)$.

Algorithm 4 Query algorithm

```
1: procedure query2(curve $Q$ with vertices $q_1, \ldots, q_k$, $\delta > 0$, $\varepsilon > 0$)
2:     $Q' \leftarrow \delta$-signature of $Q$
3:     $q'_1, \ldots, q'_\ell \leftarrow$ vertices of $Q'$
4:     $Q'' \leftarrow \left\langle \left\lfloor \frac{q'_1}{\varepsilon\delta} \right\rfloor \cdot (\varepsilon\delta), \ldots, \left\lfloor \frac{q'_\ell}{\varepsilon\delta} \right\rfloor \cdot (\varepsilon\delta) \right\rangle$        ▷ snap $Q'$ to $\mathcal{G}_{\varepsilon\delta}$
5:     if $Q''$ in $\mathcal{H}$ then
6:         return input curve $P$ associated with $Q''$ in $\mathcal{H}$
7:     return "no"
```

Lemma 28. *If `query2`$(Q, \delta, \varepsilon/2)$ returns an input curve $P \in \mathcal{P}$, then $\mathrm{d_F}(Q, P) \leq (2+\varepsilon)\delta$. If `query2`$(Q, \delta, \varepsilon/2)$ returns "no" then there is no $P \in \mathcal{P}$ such that $\mathrm{d_F}(Q, P) \leq \delta$.*

Proof. If `query2`$(Q, \delta, \varepsilon/2)$ returns an input curve $P \in \mathcal{P}$, then it must be that Q'' is stored in $\mathcal{H}$, and P is its associated input curve. By the construction of $\mathcal{H}$, it must be that $\mathrm{d_F}(P, Q'') \leq (1+\varepsilon/2)\delta$. By the definition of signatures we know that $\mathrm{d_F}(Q, Q') \leq \delta$, and by the triangle inequality we obtain

$$\mathrm{d_F}(Q, Q'') \leq \mathrm{d_F}(Q, Q') + \mathrm{d_F}(Q'', Q') \leq (1+\varepsilon/2)\delta.$$

Hence, by the triangle inequality we obtain

$$\mathrm{d_F}(P, Q) \leq \mathrm{d_F}(P, Q'') + \mathrm{d_F}(Q, Q'') \leq (2+\varepsilon)\delta.$$

Now suppose that `query2`$(Q, \delta, \varepsilon/2)$ returns "no". This means that Q'' is not stored in $\mathcal{H}$. Suppose that there exists a $P \in \mathcal{P}$ such that $\mathrm{d_F}(P, Q) \leq \delta$. Then by Lemma 18 there exists a δ-visiting order of Q' on P. Therefore, by the triangle inequality, there exists a $((1+\varepsilon/2)\delta)$-visiting order of Q'' on P, which implies that $Q'' \in \mathcal{C}(P)$, and hence Q'' is stored in $\mathcal{H}$. This leads to a contradiction, since we have assumed that Q'' is not stored in $\mathcal{H}$. Hence, if `query2`$(Q, \delta, \varepsilon/2)$ returns "no" then there is no $P \in \mathcal{P}$ such that $\mathrm{d_F}(P, Q) \leq \delta$. □

Theorem 29. *Let $\varepsilon \in (0,1]$. There is a data structure for the $(2+\varepsilon)$-ANN problem, which stores n one-dimensional curves of complexity m and supports query curves of complexity k, uses space in $n \cdot \mathcal{O}\left(\frac{m}{k\varepsilon}\right)^k$, needs $\mathcal{O}(nm) \cdot \mathcal{O}\left(\frac{m}{k\varepsilon}\right)^k$ expected preprocessing time and answers a query in $\mathcal{O}(k)$ time.*

Proof. Correctness of the data structure follows from Lemma 28. The space complexity and the preprocessing time are analyzed in the proof of Theorem 27. It remains to show that `query2`$(Q, \delta, \varepsilon/2)$ runs in $\mathcal{O}(k)$ time.

To compute a δ-signature of Q, we use the algorithm of Driemel, Krivosija and Sohler [DKS16], which runs in $\mathcal{O}(k)$ time. Since we employ perfect hashing and we assume that the floor function can be computed in constant time, each probe to $\mathcal{H}$ costs $\mathcal{O}(k)$ time, and we can also check at the same time if the answer returned by $\mathcal{H}$ is the one we are searching for. We conclude that `query2`$(Q, \delta, \varepsilon/2)$ runs in $\mathcal{O}(k)$ time. □

5.2 Improved preprocessing time

In this section, we show that there exists a data structure for the $(2+\varepsilon)$-ANN problem, with space complexity and preprocessing time in $n \cdot \mathcal{O}(1/\varepsilon)^k + \mathcal{O}(nm)$. The query time is in $\mathcal{O}(k \cdot 2^k)$. This avoids the factor $(m/k)^k$ of our previous data structures.

Data structure

We are given as input a set of one-dimensional curves $\mathcal{P}$, as sequences of vertices, the distance threshold $\delta > 0$, the approximation error $\varepsilon > 0$, and the complexity of the supported queries k. To build the data structure, we use a modified version of the preprocessing algorithm in Section 4. For each input curve $P \in \mathcal{P}$, we compute a δ-signature P' of P. If the complexity of P' is at most $k+2$ then we compute a set $\mathcal{C}' := \mathcal{C}'(P')$ which contains all curves Q such that: i) Q has complexity at most k, ii) all vertices of Q belong to $\mathcal{G}_{\varepsilon\delta/2}$, and iii) there is a $((16+\varepsilon/4)\delta)$-visiting order of Q on P'. This step is similar to the one in the preprocessing algorithm in Section 4, although here we consider signatures of the input curves instead of the original curves.

The filtering process is also slightly different. We filter $\mathcal{C}'$ to obtain a set $\mathcal{C}(P)$ which contains only those curves of $\mathcal{C}'$ with: i) Fréchet distance at most $(2+\varepsilon/4)\delta$ from P, ii) their first point within distance $(1+\varepsilon/4)\delta$ from $P(0)$, and iii) their last point within distance $(1+\varepsilon/4)\delta$ from $P(1)$. Let $\mathcal{H}$ be a dictionary which is initially empty. For each $P \in \mathcal{P}$, we store $\mathcal{C}(P)$ in $\mathcal{H}$ as follows: for each $Q \in \mathcal{C}(P)$, if Q is not already stored in $\mathcal{H}$, then we insert Q into $\mathcal{H}$, associated with a pointer to P. The preprocessing algorithm is implemented in `preprocess2`, which can be found in Algorithm 5. We also make use of the subroutine `generate_candidates` described in Algorithm 2, in Section 4.1. To achieve approximation factor $(2+\varepsilon)$, we run `preprocess2`$(\mathcal{P}, \delta, 22, 2, \varepsilon/4, k)$.

Query algorithm

Let Q be the query curve with vertices $q_1, \ldots, q_k$ and let $\varepsilon > 0$ be the approximation error. The query algorithm is the same as in the data structure of Section 4, but we run it with different input parameters. In particular, we run `query`$(Q, 2\delta, \varepsilon/4)$ (see Algorithm 3) on the dictionary $\mathcal{H}$ which is constructed by `preprocess2`$(\mathcal{P}, \delta, 22, 2, \varepsilon/4, k)$.

Lemma 30. *If `query`$(Q, 2\delta, \varepsilon/4)$ returns an input curve $P \in \mathcal{P}$, then $\mathrm{d_F}(Q,P) \leq (2+\varepsilon)\delta$. If `query`$(Q, 2\delta, \varepsilon/4)$ returns "no" then there is no $P \in \mathcal{P}$ such that $\mathrm{d_F}(Q,P) \leq \delta$.*

Proof. When `query`$(Q, 2\delta, \varepsilon/4)$ returns an input curve $P \in \mathcal{P}$, it must be that there is a δ-straightening Q' of Q such that P is associated with Q'' in $\mathcal{H}$, where Q'' denotes the curve

Algorithm 5 Preprocessing algorithm. We call `preprocess2` to build the data structure.

```
1: procedure preprocess2(input set $\mathcal{P}$, $\delta > 0$, $r > 0$, $t > 0$, $\varepsilon > 0$, $k \in \mathbb{N}$)
2:     Initialize empty dictionary $\mathcal{H}$
3:     for each $P \in \mathcal{P}$ do
4:         $P' \leftarrow \delta$-signature of $P$
5:         if $|P'| \leq k + 2$ then
6:             $\mathcal{C}(P) \leftarrow$ generate_keys2($P', \delta, r, t, \varepsilon, k$)
7:             for each $Q \in \mathcal{C}(P)$ do
8:                 if $Q$ not in $\mathcal{H}$ then
9:                     insert key $Q$ in $\mathcal{H}$, associated with a pointer to $P$
10: procedure generate_keys2(curve $P$, $\delta > 0$, $r > 0$, $t > 0$, $\varepsilon > 0$, $k$)
11:     $\mathcal{C}' \leftarrow$generate_candidates($P, \delta, r + \varepsilon, \varepsilon, k$)
12:     $\mathcal{C} \leftarrow \emptyset$
13:     for each $Q \in \mathcal{C}'$ do
14:         if $\mathrm{d_F}(P, Q) \leq (t+\varepsilon)\delta$ and $|P(0) - Q(0)| \leq (1+\varepsilon)\delta$ and $|P(1) - Q(1)| \leq (1+\varepsilon)\delta$ then
15:             $\mathcal{C} \leftarrow \mathcal{C} \cup \{Q\}$
16:     return $\mathcal{C}$
```

produced by snapping vertices of Q' to $\mathcal{G}_{\varepsilon\delta/4}$. This implies that $Q'' \in \mathcal{C}(P)$, and therefore $\mathrm{d_F}(P', Q'') \leq (2 + \varepsilon/4)\delta$, $|P'(0) - Q''(0)| \leq (1 + \varepsilon/4)\delta$, $|P'(1) - Q''(1)| \leq (1 + \varepsilon/4)\delta$, where P' is the δ-signature of P computed by `preprocess`. By the triangle inequality,

$$\mathrm{d_F}(P', Q') \leq \mathrm{d_F}(P', Q'') + \mathrm{d_F}(Q', Q'') \leq (2 + \varepsilon/2)\delta.$$

Similarly, by the triangle inequality, $|P'(0) - Q'(0)| \leq (1 + \varepsilon/2)\delta$, $|P'(1) - Q'(1)| \leq (1 + \varepsilon/2)\delta$. Lemma 20 implies that $\mathrm{d_F}(P, Q') \leq (2+\varepsilon)\delta$, because P' is a δ-signature of P, $\mathrm{d_F}(P', Q') \leq (2+\varepsilon/2)\delta$, $|P(0) - Q'(0)| \leq (1+\varepsilon/2)\delta$ and $|P(1) - Q'(1)| \leq (1+\varepsilon/2)\delta$. Then, by Lemma 19, we conclude that $\mathrm{d_F}(P, Q) \leq (2 + \varepsilon)\delta$.

If `query`$(Q, 2\delta, \varepsilon/4)$ returns "no", then there is no input curve $P \in \mathcal{P}$ such that $|P'| \leq k + 2$, where P' is the δ-signature computed by `preprocess2` and such that there exists a δ-straightening Q' of Q with $Q' \in \mathcal{C}(P)$. Suppose for the sake of contradiction that there is an input curve $P \in \mathcal{P}$ such that $\mathrm{d_F}(Q, P) \leq \delta$. Then by the triangle inequality and the fact that $\mathrm{d_F}(P, P') \leq \delta$, we obtain $\mathrm{d_F}(Q, P') \leq 2\delta$. In addition, by Lemma 18 there is a δ-visiting order of P' on Q. Since P' satisfies the δ-edge-length property, any two consecutive interior vertices lie at distance at least 2δ to each other. Thus, no two consecutive interior vertices can belong to the same δ-range. Hence, $|P'| \leq |Q| + 2 \leq k + 2$. By Lemma 22, there exists a 2δ-straightening Q' of Q which satisfies

i) there exists a 22δ-visiting order of Q' on P',
ii) $\mathrm{d_F}(Q', P') \leq 2\delta$.

By the definition of signatures, we have $P(0) = P'(0)$ and $P(1) = P'(1)$, and since $\mathrm{d_F}(P, Q) \leq \delta$, we have $|P'(0) - Q(0)| \leq \delta$ and $|P'(1) - Q(1)| \leq \delta$. By the definition of straightenings, we have $Q'(0) = Q(0)$ and $Q'(1) = Q(1)$ and therefore $|P'(0) - Q'(0)| \leq \delta$ and $|P'(1) - Q'(1)| \leq \delta$. Hence, by the triangle inequality there exists a $((22+\varepsilon/4)\delta)$-visiting order of Q'' on P', $\mathrm{d_F}(Q'', P') \leq (2+\varepsilon/4)\delta$, $|P'(0) - Q''(0)| \leq (1 + \varepsilon/4)\delta$ and $|P'(1) - Q''(1)| \leq (1 + \varepsilon/4)\delta$. This implies that $Q'' \in \mathcal{C}(P)$ which leads to a contradiction. ☐

Theorem 31. *Let $\varepsilon \in (0, 1]$. There is a data structure for the $(2 + \varepsilon)$-ANN problem, which stores n one-dimensional curves of complexity m and supports query curves of complexity k, uses space* *in $n \cdot \mathcal{O}\left(\frac{1}{\varepsilon}\right)^k + \mathcal{O}(nm)$, needs $n \cdot \mathcal{O}\left(\frac{1}{\varepsilon}\right)^k + \mathcal{O}(nm)$ expected preprocessing time and answers a query in $\mathcal{O}(k \cdot 2^k)$ time.*

Proof. Correctness follows from Lemma 30. The bound on the query time follows from Lemma 26. It remains to bound the space complexity and the preprocessing time of the data structure.

Computing one δ-signature for each $P \in \mathcal{P}$ takes linear time $\mathcal{O}(mn)$ in total, using the algorithm of Driemel, Krivosija and Sohler [DKS16]. Let P' be the δ-signature of some curve $P \in \mathcal{P}$ as computed during preprocessing. If $|P'| > k$ we ignore P. By Lemma 24, for any $P' \in \mathcal{P}'$, the running time needed to compute $\mathcal{C}'$, is upper bounded by $\binom{|P'|+k-2}{k-2} \cdot \mathcal{O}\left(\frac{1}{\varepsilon}\right)^k = \mathcal{O}\left(\frac{1}{\varepsilon}\right)^k$. The space required for P' is upper bounded by $\mathcal{O}(|\mathcal{C}(P')| \cdot k + m) = \mathcal{O}(|\mathcal{C}'| \cdot k + m) = \mathcal{O}\left(\frac{1}{\varepsilon}\right)^k + \mathcal{O}(m)$. Computing $\mathcal{C}(P')$ costs $\mathcal{O}(|\mathcal{C}'| \cdot k) = \mathcal{O}\left(\frac{1}{\varepsilon}\right)^k$ time, since we take a decision on the Fréchet distance between each curve in $\mathcal{C}'$, and P', by making use of Theorem 6 . Assuming perfect hashing for $\mathcal{H}$, the overall expected preprocessing time is in $\mathcal{O}(n) \cdot \mathcal{O}\left(\frac{1}{\varepsilon}\right)^k$ and the space usage is in $\mathcal{O}(n) \cdot \mathcal{O}\left(\frac{1}{\varepsilon}\right)^k$. □

5.3 Linear preprocessing time

In this section we present a data structure for the $(2+\varepsilon)$-ANN problem with linear space and preprocessing time $\mathcal{O}(nm)$ and with query time in $\mathcal{O}(1/\varepsilon)^k$.

Data structure

We are given as input a set of one-dimensional curves $\mathcal{P}$, as sequences of vertices, a distance threshold $\delta > 0$, the approximation error $\varepsilon > 0$ and the complexity of the supported queries k. For each input curve $P \in \mathcal{P}$, we compute a δ-signature P' of P. If $|P'| > k+2$ then we ignore P, otherwise we snap it to $\mathcal{G}_{\varepsilon\delta/2}$ to obtain a curve P''. Let $\mathcal{H}$ be a dictionary which is initially empty. For each $P \in \mathcal{P}$, we store P'' in $\mathcal{H}$ as follows: if P'' is not already stored in $\mathcal{H}$, then we insert P'' into $\mathcal{H}$, associated with a pointer to P. To achieve approximation factor $2+\varepsilon$, we run `preprocess3`$(\mathcal{P}, \delta, \varepsilon/2, k)$, as defined in Algorithm 6.

Query algorithm

Let Q be a query curve of complexity k. We compute a set $\mathcal{C}' := \mathcal{C}'(Q)$ which contains all curves P such that: i) P has complexity at most k, ii) all vertices of P belong to $\mathcal{G}_{\varepsilon\delta/2}$, and iii) there is a $((1+\varepsilon/2)\delta)$-visiting order of P on Q. We filter $\mathcal{C}'$ to obtain a set $\mathcal{C}(Q)$ which contains only those curves of $\mathcal{C}'$ with: i) Fréchet distance at most $(2+\varepsilon/2)\delta$ from Q, ii) their first point within distance $(1+\varepsilon/2)\delta$ from $Q(0)$, and iii) their last point within distance $(1+\varepsilon/2)\delta$ from $Q(1)$. We probe $\mathcal{H}$ for each key $P \in \mathcal{C}(Q)$: if we find a $P \in \mathcal{C}(Q)$ stored in $\mathcal{H}$ then we return the associated input curve. If there is no $P \in \mathcal{C}(Q)$ stored in $\mathcal{H}$ then we return "no". To achieve the desired approximation, we run `query3`$(Q, \delta, \varepsilon/2)$, as defined in Algorithm 7.

Lemma 32. *If `query3`$(Q, \delta, \varepsilon/2)$ returns an input curve $P \in \mathcal{P}$, then $\mathrm{d_F}(Q, P) \leq (2+\varepsilon)\delta$. If `query3`$(Q, \delta, \varepsilon/2)$ returns "no" then there is no $P \in \mathcal{P}$ such that $\mathrm{d_F}(Q, P) \leq \delta$.*

Proof. If `query3`$(Q, \delta, \varepsilon/2)$ returns an input curve, then it must be that there is a curve $P'' \in \mathcal{C}(Q)$ which is stored in $\mathcal{H}$, associated with a pointer to P. Since P'' is stored in $\mathcal{H}$, there is a curve $P \in \mathcal{P}$ with a δ-signature P' such that $\mathrm{d_F}(P', P'') \leq \varepsilon\delta/2$. Moreover, since $P'' \in \mathcal{C}(Q)$, we have that $\mathrm{d_F}(Q, P'') \leq (2+\varepsilon/2)\delta$, $|Q(0) - P''(0)| \leq (1+\varepsilon/2)\delta$, $|Q(1) - P''(1)| \leq (1+\varepsilon/2)\delta$. By the triangle inequality we obtain, $\mathrm{d_F}(Q, P') \leq (2+\varepsilon)\delta$, $|Q(0) - P'(0)| \leq (1+\varepsilon)\delta$, $|Q(1) - P'(1)| \leq (1+\varepsilon)\delta$. By

Algorithm 6 Preprocessing algorithm

```
1: procedure preprocess3(input set $\mathcal{P}$, $\delta > 0$, $\varepsilon > 0$, $k$)
2:     Initialize empty dictionary $\mathcal{H}$
3:     for each $P \in \mathcal{P}$ do
4:         $P' \leftarrow \delta$-signature of $P$
5:         if $|P'| \leq k + 2$ then
6:             $p_1, \dots, p_\ell \leftarrow$ vertices of $P'$
7:             $P'' \leftarrow \left\langle \left\lfloor \frac{p_1}{\varepsilon\delta} \right\rfloor \cdot (\varepsilon\delta), \dots, \left\lfloor \frac{p_\ell}{\varepsilon\delta} \right\rfloor \cdot (\varepsilon\delta) \right\rangle$
8:             if $P''$ not in $\mathcal{H}$ then
9:                 insert key $P''$ in $\mathcal{H}$, associated with a pointer to $P$
```

Algorithm 7 Query algorithm

```
1: procedure query3(curve $Q$ with vertices $q_1, \dots, q_k$, $\delta > 0$, $\varepsilon > 0$)
2:     $\mathcal{C}(Q) \leftarrow$ generate_keys2($Q, \delta, 1, 2, \varepsilon, k + 2$)
3:     for each $P'' \in \mathcal{C}(Q)$ do
4:         if $P''$ in $\mathcal{H}$ then
5:             return input curve $P$ associated with $P''$ in $\mathcal{H}$
6:     return "no"
```

Lemma 20, since $d_F(Q, P') \leq (2 + \varepsilon)\delta$, $|Q(0) - P'(0)| \leq (1 + \varepsilon)\delta$, $|Q(1) - P'(1)| \leq (1 + \varepsilon)\delta$ and P' is a δ-signature of P, we conclude $d_F(Q, P) \leq (2 + \varepsilon)\delta$.

If `query3`$(Q, \delta, \varepsilon/2)$ returns "no", then it must be that there is no curve $P'' \in \mathcal{C}(Q)$ which is stored in $\mathcal{H}$. Suppose for the sake of contradiction that there is an input curve $P \in \mathcal{P}$ such that $d_F(Q, P) \leq \delta$. Let P' be the δ-signature of P, as computed during preprocessing. By Lemma 18, there is a δ-visiting order of P' on Q and therefore $|P'| \leq k + 2$. Let P'' be the curve produced by snapping the vertices of P' to the grid $\mathcal{G}_{\varepsilon\delta/2}$. By the triangle inequality there is a $((1+\varepsilon/2)\delta)$-visiting order of P'' on Q. Therefore, P'' must be included in $\mathcal{C}(Q)$, which leads to contradiction. □

Theorem 33. *Let $\varepsilon \in (0, 1]$. There is a data structure for the $(2 + \varepsilon)$-ANN problem, which stores n one-dimensional curves of complexity m and supports query curves of complexity k, uses space in $\mathcal{O}(nm)$, needs $\mathcal{O}(nm)$ expected preprocessing time and answers a query in $\mathcal{O}(1/\varepsilon)^{k+2}$ time.*

Proof. Correctness follows from Lemma 32. It remains to bound the space complexity, the preprocessing time and the query time.

Using the algorithm of Driemel, Krivosija and Sohler [DKS16], we can compute a signature in linear time. Since we assume that the floor function can be computed in $\mathcal{O}(1)$, and that $\mathcal{H}$ is implemented using perfect hashing, `preprocess3`$(\mathcal{P}, 1, \varepsilon/2, k)$ has running time $\mathcal{O}(nm)$. Therefore, the space usage is also in $\mathcal{O}(nm)$.

To bound the query time, we bound the running time of `generate_keys2`$(Q, \delta, 1, 2, \varepsilon/2, k + 2)$, because the last part of `query3` is an enumeration over all curves returned by `generate_keys2` and probing $\mathcal{H}$ for each one of them. To bound the running time of `generate_keys2`$(Q, \delta, 1, 2, \varepsilon/2, k+2)$, it suffices to bound the running time of `generate_candidates`$(Q, \delta, (1 + \varepsilon/2), \varepsilon/2, k + 2)$. By Lemma 24, this running time is upper bounded by $\binom{2k}{k-2} \cdot \mathcal{O}\left(\frac{1}{\varepsilon}\right)^{k+2} = \mathcal{O}\left(\frac{1}{\varepsilon}\right)^{k+2}$. Recall that we employ perfect hashing and we assume that the floor function can be computed in constant time. Hence each probe to $\mathcal{H}$ costs $\mathcal{O}(k)$ time, and we can also check in $\mathcal{O}(k)$ if $\mathcal{H}$ returns the correct answer. We conclude that `query2`$(Q, \delta, \varepsilon/2)$ runs in time $\mathcal{O}\left(\frac{1}{\varepsilon}\right)^{k+2}$. □

6 $(3+\varepsilon)$-Approximation

In this section, we present a data structure for the $(3+\varepsilon)$-ANN problem with preprocessing time and space complexity in $n \cdot \mathcal{O}(1/\varepsilon)^k + \mathcal{O}(nm)$ and query time in $\mathcal{O}(k)$.

Data structure

We are given as input a set of one-dimensional curves $\mathcal{P}$, as sequences of vertices, a distance threshold $\delta > 0$, the approximation error $\varepsilon > 0$ and the complexity of the supported queries k. To build the data structure, we use the preprocessing algorithm of the data structure in Section 5.2. Let $\mathcal{H}$ be the dictionary, constructed by `preprocess2`$(\mathcal{P}, \delta, 2, 3, \varepsilon/2, k)$.

Query algorithm

Let Q be a query curve. We run the query algorithm of the data structure in Section 5.1. In particular, we run `query2`$(Q, 2\delta, \varepsilon/2)$ on $\mathcal{H}$.

Lemma 34. *If* `query2`$(Q, 2\delta, \varepsilon/2)$ *returns an input curve* $P \in \mathcal{P}$*, then* $\mathrm{d_F}(Q,P) \leq (3+\varepsilon)\delta$*. If* `query2`$(Q, 2\delta, \varepsilon/2)$ *returns "no" then there is no* $P \in \mathcal{P}$ *such that* $\mathrm{d_F}(Q,P) \leq \delta$*.*

Proof. Let Q' be the 2δ-signature of Q and let Q'' be the curve obtained by snapping vertices of Q' to $\mathcal{G}_{\varepsilon\delta/2}$, as computed in `query2`.

If `query2`$(Q, 2\delta, \varepsilon/2)$ returns an input curve $P \in \mathcal{P}$, then it must be that $Q'' \in \mathcal{C}(P)$, where $\mathcal{C}(P)$ is the result of `generate_keys2`$(P', \delta, 2, 3, \varepsilon/2, k)$ and P' is a δ-signature of P, as computed by `preprocess2`. By the construction of $\mathcal{C}(P)$, it must be that $\mathrm{d_F}(P', Q'') \leq (3+\varepsilon/2)\delta$, $|P'(0) - Q''(0)| \leq (1+\varepsilon/2)\delta$ and $|P'(1) - Q''(1)| \leq (1+\varepsilon/2)\delta$. Hence, by the triangle inequality $\mathrm{d_F}(Q', P') \leq (3+\varepsilon)\delta$, $|P'(0) - Q'(0)| \leq (1+\varepsilon)\delta$ and $|P'(1) - Q'(1)| \leq (1+\varepsilon)\delta$. We now apply Lemma 20 twice. We first apply it on P', Q', Q. Since $\mathrm{d_F}(P', Q') \leq (3+\varepsilon)\delta$, $|P'(0) - Q'(0)| \leq (1+\varepsilon)\delta$, $|P'(1) - Q'(1)| \leq (1+\varepsilon)\delta$ and Q' is a 2δ-signature of Q, we obtain $\mathrm{d_F}(P', Q) \leq (3+\varepsilon)\delta$. Then, we apply it on P', P, Q. Since $\mathrm{d_F}(P', Q) \leq (3+\varepsilon)\delta$, $|P'(0) - Q(0)| = |P'(0) - Q'(0)| \leq (1+\varepsilon)\delta \leq (2+\varepsilon)\delta$, $|P'(1) - Q(1)| = |P'(1) - Q'(1)| \leq (1+\varepsilon)\delta \leq (2+\varepsilon)\delta$, and P' is a δ-signature of P, we obtain $\mathrm{d_F}(P, Q) \leq (3+\varepsilon)\delta$.

If `query2`$(Q, 2\delta, \varepsilon/2)$ returns "no" then Q'' is not stored in $\mathcal{H}$ as a key. For the sake of contradiction, we assume that there exists an input curve $P \in \mathcal{P}$ such that $\mathrm{d_F}(P, Q) \leq \delta$. Then by definition, $|P'(0) - Q'(0)| \leq \delta$ and $|P'(1) - Q'(1)| \leq \delta$. In addition, by Lemma 23, $\mathrm{d_F}(P', Q') \leq 3\delta$ and there is a 2δ-visiting order of Q' on P', By the triangle inequality we obtain $\mathrm{d_F}(P', Q'') \leq (3+\varepsilon/2)\delta$, $|P'(0) - Q''(0)| \leq (1+\varepsilon/2)\delta$, $|P'(1) - Q''(1)| \leq (1+\varepsilon/2)\delta$, and that there is a $((2+\varepsilon/2)\delta)$-visiting order of Q'' on P'. Hence, by the construction of $\mathcal{C}(P)$, it must be that $Q'' \in \mathcal{C}(P)$ which implies that Q'' is stored as a key in $\mathcal{H}$. This is a contradiction. $\square$

Theorem 35. *Let* $\varepsilon \in (0,1]$*. There is a data structure for the* $(3+\varepsilon)$*-ANN problem, which stores* n *one-dimensional curves of complexity* m *and supports query curves of complexity* k*, uses space in* $n \cdot \mathcal{O}(1/\varepsilon)^k + \mathcal{O}(nm)$*, needs* $n \cdot \mathcal{O}(1/\varepsilon)^k + \mathcal{O}(nm)$ *expected preprocessing time and answers a query in* $\mathcal{O}(k)$ *time. where* k *is the complexity of the query curve.*

Proof. Correctness follows from Lemma 34. The bounds on the preprocessing time and space complexity follow from Theorem 31. The bound on the query time follows from Theorem 29. $\square$

7 Proofs of main lemmas

In this section we give full proofs of the lemmas stated in Section 3. We start by proving a fundamental observation and lemma on the Fréchet distance of approximately monotone one-dimensional curves.

Observation 36. *Let Q be a directed line segment and let $P: [0,1] \mapsto \mathbb{R}$ be a curve. It holds that $\mathrm{d_F}(P,Q) \le \delta$ if and only if the following conditions are satisfied:*
(i) P is 2δ-monotone with respect to Q, and
(ii) $|P(0)-Q(0)| \le \delta$, $|P(1)-Q(1)| \le \delta$, and
(iii) $P \subseteq B(Q,\delta)$.

Proof. We assume that $Q(0) \le Q(1)$ as the other case is symmetric. Now, assume first that $\mathrm{d_F}(P,Q) \le \delta$, then (ii) holds because start and end points are matched in any traversal and (iii) holds as the Hausdorff distance is a lower bound for the Fréchet distance. Finally, (i) holds as otherwise there exist two indices $s,t \in [0,1]$ with $s < t$ and $P(t) < P(s) - 2\delta$. As Q is increasing, no traversal can match $P(s)$ and $P(t)$ in distance at most δ.

Second, assume that (i), (ii), and (iii) hold. Then $\mathrm{d_F}(P,Q) \le \delta$ is implied by Lemma 37, below, but to provide some intuition we give a simpler proof here. The following traversal with position s on P and position t on Q stays within distance δ. We start in $P(0), Q(0)$, then we continue on P until $P(s) = Q(0)+\delta$. Then we always choose t such that $Q(t) = \min_{s' \ge s} P(s') + \delta$ while traversing P, i.e., continuously increasing s. When we reach the end of Q, we can traverse P until the end while staying in $Q(1)$. It is easy to check that properties (i), (ii), and (iii) ensure distance δ during the described traversal. $\square$

The following lemma statement is similar to the above observation with the important difference that the line segment Q is replaced by a 2δ-monotone curve. The proof works by constructing a traversal greedily and showing correctness of the greedy algorithm.

Lemma 37. *Let P and Q be 2δ-monotone curves with*
(i) P is 2δ-monotone with respect to $\overline{Q(0)Q(1)}$, and
(ii) $|P(0)-Q(0)| \le \delta$, $|P(1)-Q(1)| \le \delta$, and
(iii) $P \subseteq B(Q,\delta)$, and
(iv) $Q \subseteq \overline{Q(0)Q(1)}$.
It holds that $\mathrm{d_F}(P,Q) \le \delta$.

Proof. We assume that $Q(0) \le Q(1)$ as the other case is symmetric. If Q is not 2δ-monotone increasing, then it also cannot be 2δ-monotone decreasing: if there are two points $s,t \in [0,1]$ with $s < t$ such that $Q(t) < Q(s) - 2\delta$, then, as $Q(t) \ge Q(0)$ by condition (iv), we have that $Q(s) > Q(t) + 2\delta \ge Q(0) + 2\delta$ and thus Q is not 2δ-monotone decreasing. However, as Q is 2δ-monotone, it has to be 2δ-monotone *increasing*. Due to condition (i), P is also 2δ-monotone *increasing*. We give a traversal of P,Q with distance at most δ — denoting the position during the traversal with $(s,t) \in [0,1]^2$ — that tries to maintain two invariants:

(1) P and Q are in a position $(s,t) \in [0,1]^2$ such that $P(s) = Q(t) + \delta$.

(2) The suffix of Q is strictly greater than the current value $Q(t)$, i.e., $\forall t' > t : Q(t') > Q(t)$.

In general, both invariants may be violated at the very beginning of the traversal, that is, for $s = t = 0$. Let us first describe how we traverse from the beginning of P,Q to a position $(s,t) \in [0,1]^2$ such that these invariants are fulfilled. We first traverse P until it first reaches $Q(0)+\delta$, while in Q

we stay in $Q(0)$. Note that by condition (ii), we cannot have $P(0) > Q(0) + \delta$. Furthermore, this traversal is feasible as the traversed prefix of P is in the range $[Q(0)-\delta, Q(0)+\delta]$, by condition (iii), and thus within distance δ to $Q(0)$. If we reach $P(1)$ before reaching $Q(0) + \delta$, then we know that $Q \subseteq [P(1)-\delta, P(1)+\delta]$ and we can thus traverse complete Q and $\mathrm{d_F}(P, Q) \leq \delta$. If we did not reach $P(1)$, we now traverse Q until its last point with value $Q(0)$, which is possible as the traversed prefix of Q lies in $[Q(0), Q(0) + 2\delta]$, due to condition (iv) and as Q is 2δ-monotone increasing, and the position on P is currently $Q(0) + \delta$.

From now on, we traverse P and Q with the same speed in image space, unless one of the two invariants would be violated by continuing the traversal. If both invariants would be violated at the same time, we break ties by restoring Invariant (1) before Invariant (2). Now, let s be the position on P and t the position on Q when an invariant would be violated. When Invariant (1) would be violated, we continue traversing P while staying in $Q(t)$ on Q until the next time we reach a position s' on P with value $P(s') = P(s)$. Note that we might not reach such a position s' because we reached the end of P. However, if we did not reach the end of P, the invariant is restored. This traversal keeps the two positions at distance δ as $P(s) = Q(t) + \delta$ and as P is 2δ-monotone increasing. In case Invariant (2) would be violated, we continue traversing Q until we reach the largest position $t' > t$ such that $Q(t') = Q(t)$. Note that afterwards, both invariants hold (as we restore Invariant (1) before Invariant (2)), and, in particular, we cannot reach the end of Q due to the existence of $Q(t')$ which we reach at the end of restoring Invariant (2). This traversal also keeps the two positions at distance δ as initially $Q(t) = P(s) - \delta$ and Q is 2δ-monotone increasing and there is no position t'' on Q with $Q(t'') < Q(t)$, i.e., all the points before reaching position t' on Q have to be in the range $[Q(t), Q(t) + 2\delta]$.

In all of the above cases we are guaranteed to make progress in our traversal. Furthermore, we will reach the end of P before or at the same time as we reach the end of Q because, first, while restoring invariants we can only reach the end of P but not of Q as argued above and, second, if we reach the end of Q while both invariants would continue to hold, we also have to reach the end of P at the same time as otherwise we would violate condition (iii) of the lemma. When we reach the end of P, we know that $P(1) \in [Q(1) - \delta, Q(1) + \delta]$ due to condition (ii), and the remaining Q is in $[P(1) - \delta, Q(1)]$. Thus, the remaining Q is in $[P(1) - \delta, P(1) + \delta]$ and consequently Q can be traversed until the end.

It follows from the traversal constructed thereby that $\mathrm{d_F}(P, Q) \leq \delta$. □

7.1 Proofs of lemmas for straightenings

Next, we want to prove Lemma 19 from Section 3. We first prove a simpler statement, which can be thought of as a special case where the straightening consists of only one edge.

Lemma 38. *Let $X = \overline{ab} \subset \mathbb{R}$ be a line segment and let $Q : [0,1] \mapsto \mathbb{R}$ be a curve such that: $Q(0) = X(0)$, $Q(1) = X(1)$, for all $t \in [0,1] : Q(t) \in \overline{ab}$ and $\mathrm{d_F}(Q, X) \leq \delta$. For any curve $P : [0,1] \mapsto \mathbb{R}$ with $\mathrm{d_F}(P, X) \leq \delta$, it holds that $\mathrm{d_F}(P, Q) \leq \delta$.*

Proof. To show the lemma statement, we want to apply Lemma 37 to P and Q. For this, we need to show that the conditions on Q and P from the lemma statement are met. By Observation 36 applied to Q and the line segment X, it follows that Q must be 2δ-monotone with respect to X, and by our assumptions, Q is range-preserving (condition (iv)). By Observation 36 applied to P and X, it also follows that P is 2δ-monotone, and conditions (ii), (iii) and (i) are satisfied. Therefore, Lemma 37 can be applied to P and Q and the claim is implied. □

Lemma 19. *Let $P : [0,1] \mapsto \mathbb{R}$ and $Q : [0,1] \mapsto \mathbb{R}$ be two curves and let Q' be any δ-straightening of Q. If $\mathrm{d_F}(P, Q') \leq \delta$ then $\mathrm{d_F}(P, Q) \leq \delta$.*

Proof. Let $q_1, \dots, q_\ell$ be the parameters corresponding to the vertices of Q' in Q, i.e., the vertices of Q' are $Q(q_1), \dots, Q(q_\ell)$. Let $\phi : [0,1] \to [0,1]^2$ be a δ-traversal between P and Q'. Let $0 = t_1 \leq \dots \leq t_\ell = 1$ be a partition of the parameter space of P such that for any $1 \leq i \leq \ell - 1$, the edge $\overline{Q(q_i)Q(q_{i+1})}$ is mapped to $P[t_i, t_{i+1}]$ under ϕ. As such, we have

$$\mathrm{d_F}(P[t_i, t_{i+1}], \overline{Q(q_i)Q(q_{i+1})}) \leq \delta$$

By the locality property of δ-simplifications, we also have that

$$\mathrm{d_F}(Q[q_i, q_{i+1}], \overline{Q(q_i)Q(q_{i+1})}) \leq \delta$$

Now, Lemma 38 implies that

$$\mathrm{d_F}(P[t_i, t_{i+1}], Q[q_i, q_{i+1}]) \leq \delta.$$

Finally, we apply Observation 4 on $P = \bigcirc_{i=1}^{\ell} P[t_i, t_{i+1}]$ and $Q = \bigcirc_{i=1}^{\ell} Q[q_i, q_{i+1}]$, and we obtain

$$\mathrm{d_F}(P, Q) \leq \max_{i \in [\ell]} \mathrm{d_F}\left(P[t_i, t_{i+1}], Q[q_i, q_{i+1}]\right) \leq \delta.$$

□

7.2 Proofs of lemmas for signatures

Next, we want to prove Lemma 20 from Section 3. We first prove an auxiliary statement for signature edges in Lemma 39. In particular, we need to take care of the first and last edge of the signature. For the other edges we can use Lemma 38. Technically, we will also need the symmetric statement of this lemma for $a > b$; this follows by mirroring at the origin. The proof of this lemma turns out be technically involved. For the proof of Lemma 20 we can then use the same approach as for Lemma 19 above.

Lemma 39. *Let $\delta = \delta' + \delta''$ for $\delta, \delta', \delta'' \geq 0$. Let $X = \overline{ab} \subset \mathbb{R}$ be a line segment with $a \leq b$ and let $Q : [0,1] \mapsto \mathbb{R}$ be a curve such that: $Q(0) = X(0)$, $Q(1) = X(1)$ and $\mathrm{d_F}(Q, X) \leq \delta'$. Let $P : [0,1] \mapsto \mathbb{R}$ be a curve with $\mathrm{d_F}(P, X) \leq \delta$.*

If either

(i) $|Q(0) - P(0)| \leq \delta''$ and $|Q(1) - P(1)| \leq \delta''$, or
(ii) $|Q(0) - P(0)| \leq \delta''$ and $\max_{t \in [0,1]}(Q(t)) \leq Q(1)$, or
(iii) $\min_{t \in [0,1]}(Q(t)) \geq Q(0)$ and $|Q(1) - P(1)| \leq \delta''$,
then it holds that $\mathrm{d_F}(P, Q) \leq \delta$.

Proof. Let $t_{\min} = \arg\min\{Q(t)\}$ and $t_{\max} = \arg\max\{Q(t)\}$. In case the minimum (resp. maximum) is not unique, we choose any of them. By Observation 36, we have that $\forall t \in [0,1]$ $Q(t) \in [Q(0) - \delta', Q(1) + \delta']$ and by assumption of case (i) $|P(0) - Q(0)| \leq \delta''$ and $|P(1) - Q(1)| \leq \delta''$. Therefore, by triangle inequality, we have in case (i), that

$$|P(0) - Q(t_{\min})| \leq \delta \quad \text{and} \quad |P(1) - Q(t_{\max})| \leq \delta$$

It is easy to see that this holds in the cases (ii) and (iii), as well, since $|P(0) - Q(0)| \leq \delta$ and $|P(1) - Q(1)| \leq \delta$ holds in any case as we assume $\mathrm{d_F}(P, X) \leq \delta$.

Now, define

$$t_1 = \min\{t \in [0,1] \mid P(t) \geq Q(t_{\min}) + \delta\}$$
$$t_2 = \max\{t \in [0,1] \mid P(t) \leq Q(t_{\max}) - \delta\}$$

If such a t_1 does not exist, then we set $t_1 = 1$. If t_2 does not exist, then we set $t_2 = 0$.

Note that by construction and Observation 36 we have

$$\mathrm{d_F}(P[0,t_1], Q(0)) \leq \delta \quad \text{and} \quad \mathrm{d_F}(P[t_2,1], Q(1)) \leq \delta \tag{3}$$

Indeed, (3) holds true since $Q(t_{\min}) \leq Q(0) \leq Q(t_{\min}) + \delta'$ and, likewise, $Q(t_{\max}) \geq Q(1) \geq Q(t_{\max}) - \delta'$, and, moreover, the image of the subcurve $P[0,t_1]$ is contained in the interval $[Q(0) - \delta, Q(t_{\min})+\delta]$ and the image of the subcurve $P[t_2, 1]$ is contained in the interval $[Q(t_{\max})-\delta, Q(1)+\delta]$.

In addition, we have

$$\mathrm{d_F}(P(t_1), Q[0,t_{\min}]) \leq \delta \quad \text{and} \quad \mathrm{d_F}(P(t_2), Q[t_{\max},1]) \leq \delta \tag{4}$$

Indeed, (4) holds true, since $\delta' \leq \delta$ and by Observation 36, Q is $2\delta'$-monotone increasing, and therefore the image of the subcurve $Q[0,t_{\min}]$ is contained in the interval $[Q(t_{\min}), Q(t_{\min}) + 2\delta']$ which by construction is equal to $[P(t_1) - \delta', P(t_1) + \delta']$ and the image of the subcurve $Q[t_{\max}, 1]$ is contained in the interval $[Q(t_{\max}) - 2\delta', Q(t_{\max})]$, which by construction is equal to $[P(t_2) - \delta', P(t_2) + \delta']$.

Now, assume that $t_1 \leq t_2$ and $t_{\min} \leq t_{\max}$. In this case, the subcurves $P[t_1,t_2]$ and $Q[t_{\min}, t_{\max}]$ are well-defined. By construction, $|P(t_1) - Q(t_{\min}) \mid\leq \delta$, $|P(t_2) - Q(t_{\max}) \mid\leq \delta$ and $Q[t_{\min}, t_{\max}] \subseteq \overline{Q(t_{\min})Q(t_{\max})}$. By Observation 36, P and Q are both 2δ-monotone with respect to X, and, by definition, $X = \overline{Q(0)Q(1)}$. Moreover, by the definition of t_1, t_2, we have $P[t_1,t_2] \subseteq B(Q[t_{\min}, t_{\max}], \delta)$. Therefore all conditions of Lemma 37 are satisfied, which implies that

$$\mathrm{d_F}(P[t_1,t_2], Q[t_{\min}, t_{\max}]) \leq \delta \tag{5}$$

In summary, we have by (3),(4), and (5) that

$$\max\begin{pmatrix} \mathrm{d_F}(P[0,t_1], Q(0)) \\ \mathrm{d_F}(P(t_1), Q[0,t_{\min}]) \\ \mathrm{d_F}(P[t_1,t_2], Q[t_{\min}, t_{\max}]) \\ \mathrm{d_F}(P(t_2), Q[t_{\max},1]) \\ \mathrm{d_F}(P[t_2,1], Q(1)) \end{pmatrix} \leq \delta$$

Now, by Observation 4 we can concatenate these subcurves and $\mathrm{d_F}(P,Q) \leq \delta$ is implied.

If the assumption $t_1 \leq t_2$ fails, then, in fact, a simpler decomposition works. Indeed, if $t_1 > t_2$, then it holds by (3) and (4) that

$$\max\begin{pmatrix} \mathrm{d_F}(P[0,t_1], Q(0)) \\ \mathrm{d_F}(P(t_1), Q) \\ \mathrm{d_F}(P[t_1,1], Q(1)) \end{pmatrix} \leq \delta$$

Therefore, also in this case, $\mathrm{d_F}(P,Q) \leq \delta$ holds true.

Finally, we need to consider the case that the assumption $t_{\min} \leq t_{\max}$ fails. We may assume that $t_1 \leq t_2$, as we covered the case $t_1 > t_2$ above. We will consider the different cases from the lemma statement separately. First, note that if $t_{\min} > t_{\max}$, then $|Q(t_{\max}) - Q(t_{\min})| \leq 2\delta$, since Q is 2δ-monotone, and therefore, Q is contained in the interval $[P(t_1) - \delta, P(t_1) + \delta]$. By a similar argument, Q is contained in the interval $[P(t_2) - \delta, P(t_2) + \delta]$.

Now, assume case (ii) from the lemma statement. In this case, we have by the above and by Lemma 37

$$\max\begin{pmatrix} \mathrm{d_F}(P[0,t_1], Q(0)) \\ \mathrm{d_F}(P(t_1), Q[0,t_{\min}]) \\ \mathrm{d_F}(P[t_1,1], Q[t_{\min},1])) \end{pmatrix} \leq \delta$$

Assume case (iii) from the lemma statement. In this case, we have symmetrically

$$\max \begin{pmatrix} \mathrm{d_F}(P[0,t_2], Q[0,t_{\max}]) \\ \mathrm{d_F}(P(t_2), Q[t_{\max},1]) \\ \mathrm{d_F}(P[t_2,1], Q(1)) \end{pmatrix} \leq \delta$$

Now, for case (i), we claim that there exist $0 \leq q_1 \leq q_2 \leq 1$, such that

$$\max \begin{pmatrix} \mathrm{d_F}(P[0,t_1], Q(0)) \\ \mathrm{d_F}(P(t_1), Q[0,q_1]) \\ \mathrm{d_F}(P[t_1,t_2], Q[q_1,q_2]) \\ \mathrm{d_F}(P(t_2), Q[q_2,1]) \\ \mathrm{d_F}(P[t_2,1], Q(1)) \end{pmatrix} \leq \delta$$

Indeed, from what we derived, $\mathrm{d_F}(P(t_1), Q[0,q_1]) \leq \delta$ and $\mathrm{d_F}(P(t_2), Q[q_2,1]) \leq \delta$ holds for any choice of $q_1, q_2 \in [0,1]$. The first and last line hold by (3). It remains to show that we can choose q_1, q_2 so that $\mathrm{d_F}(P[t_1,t_2], Q[q_1,q_2]) \leq \delta$ holds. Since $\mathrm{d_F}(P,X) \leq \delta$, there must be a subsegment $X[x_1,x_2]$ of X, such that $\mathrm{d_F}(P[t_1,t_2], X[x_1,x_2]) \leq \delta$. Recall that $\overline{Q(0)Q(1)} = X$ and by the intermediate value theorem we can define suitable q_1, q_2 as follows

$$q_1 = \max\{q \in [0,1] \mid Q(q) = X(x_1)\}$$
$$q_2 = \min\{q \in [q_1,1] \mid Q(q) = X(x_2)\}$$

Now, we can apply Lemma 37 and conclude that $\mathrm{d_F}(P[t_1,t_2], Q[q_1,q_2]) \leq \delta$. Therefore, also in case (i), we have $\mathrm{d_F}(P,Q) \leq \delta$. □

Now we are ready to prove Lemma 20.

Lemma 20. *Let $\delta = \delta' + \delta''$ for $\delta, \delta', \delta'' \geq 0$ and let $P : [0,1] \mapsto \mathbb{R}$ and $Q : [0,1] \mapsto \mathbb{R}$ be two curves. Let Q' be any δ'-signature of Q. If $\mathrm{d_F}(Q',P) \leq \delta$, $|Q(0) - P(0)| \leq \delta''$, and $|Q(1) - P(1)| \leq \delta''$, then $\mathrm{d_F}(P,Q) \leq \delta$.*

Proof. This follows by a modification of the proof of Lemma 19. Although the two proofs are very similar, the differences are subtle. Therefore, we give the full proof for the sake of completeness. Let $q_1, \ldots, q_\ell$ be the parameters corresponding to the vertices of Q' in Q, i.e., the vertices of Q' are $Q(q_1), \ldots, Q(q_\ell)$. Let $\phi : [0,1] \to [0,1]^2$ be a δ-traversal between P and Q'. Let $0 = t_1 \leq \cdots \leq t_\ell = 1$ be a partition of the parameter space of P such that for any $1 \leq i \leq \ell - 1$, the edge $\overline{Q(q_i)Q(q_{i+1})}$ is mapped to $P[t_i, t_{i+1}]$ under ϕ. As such, we have

$$\mathrm{d_F}(P[t_i,t_{i+1}], \overline{Q(q_i)Q(q_{i+1})}) \leq \delta \tag{6}$$

By the definition of δ-simplifications, we also have that

$$\mathrm{d_F}(Q[q_i,q_{i+1}], \overline{Q(q_i)Q(q_{i+1})}) \leq \delta' \leq \delta \tag{7}$$

Now, if the edge $\overline{Q(q_i)Q(q_{i+1})}$ of Q' is range-preserving, then Lemma 38 implies that

$$\mathrm{d_F}(P[t_i,t_{i+1}], Q[q_i,q_{i+1}]) \leq \delta. \tag{8}$$

Otherwise, it must be (by the definition of signatures) that either $i = 1$ or $i + 1 = \ell$ or both (the edge is the first or last edge of the signature Q' or Q' consists of just one edge). In any of those cases, Lemma 39 implies $\mathrm{d_F}(P[t_i,t_{i+1}], Q[q_i,q_{i+1}]) \leq \delta$.

Finally, we apply Observation 4 on $P = \bigcirc_{i=1}^{\ell} P[t_i,t_{i+1}]$ and $Q = \bigcirc_{i=1}^{\ell} Q[q_i,q_{i+1}]$, and we obtain

$$\mathrm{d_F}(P,Q) \leq \max_{i \in [\ell]} \mathrm{d_F}\left(P[t_i,t_{i+1}], Q[q_i,q_{i+1}]\right) \leq \delta.$$

□

7.3 Proofs of lemmas for visiting orders

In order to prove the existence of δ'-visiting orders for some $\delta' \in O(\delta)$ as claimed in Lemma 22, we introduce the concept of a visiting sequence. A visiting sequence is not necessarily monotonically increasing, while visiting orders according to Definition 17 are. Nonetheless, this definition of visiting sequence will turn out to be useful. It is important that a δ-visiting sequence is derived from a monotone traversal. We will show (Lemma 42 and 43) that any non-monotonic visiting sequence can be turned into a monotonic one at the expense of a constant factor in the radius of the visiting sequence.

Definition 40. *Let $P : [0,1] \to \mathbb{R}$ and $Q : [0,1] \to \mathbb{R}$ be curves, let $\delta > 0$, and let $\phi : [0,1] \to [0,1]^2$ be a monotone traversal. We say a vertex w of Q δ-**visits** a vertex v of P **under** ϕ if the following holds:*
(i) $|w - v| \leq \delta$ and
(ii) at least one of the following holds:
(a) ϕ associates w with v, or
(b) ϕ associates w with the interior of an edge of P that is incident to v, or
(c) ϕ associates v with the interior of an edge of Q that is incident to w.
Note that the induced relation on the vertices is symmetric for any fixed δ and ϕ.

Definition 41. *Let $P : [0,1] \to \mathbb{R}$ and $Q : [0,1] \to \mathbb{R}$ be curves and let $\phi : [0,1] \to [0,1]^2$ be a monotone traversal. Let S be a subsequence of the vertices of Q of length ℓ. Let $u_1, \ldots, u_\ell$ denote the ordered vertices of S and let $v_1, \ldots, v_m$ denote the ordered vertices of P. A δ-**visiting sequence** of S on P **under** ϕ is a sequence of indices $i_1, \ldots, i_\ell$, such that each u_j of S δ-visits the vertex v_{i_j} of P under ϕ.*

Lemma 42. *Let $P : [0,1] \mapsto \mathbb{R}$ and $Q : [0,1] \mapsto \mathbb{R}$ be curves such that $\mathrm{d_F}(Q, P) \leq \delta$ and let ϕ be a monotone traversal realizing this distance. Let v_i, v_j be two vertices of Q with $i < j$ in the ordering along Q. Assume v_i δ-visits a vertex w_a of P under ϕ and v_j δ-visits a vertex w_b of P under ϕ such that $a > b$ in the ordering along P. Then, it must be that v_i 3δ-visits w_b under ϕ and that v_j 3δ-visits w_a under ϕ.*

Proof. As $a > b$, however, in ϕ a point on an adjacent edge of w_a is matched earlier than a point on an adjacent edge of w_b, we conclude due to the monotonicity of ϕ that $\overline{w_b w_a}$ is an edge in P. Let $P(t)$ and $P(t')$ be the points that v_i and v_j are mapped to on $\overline{w_b w_a}$ under ϕ, respectively. By the monotonicity of ϕ we have $t \leq t'$. See Figure 3 for an illustration.

Assume that $w_a < w_b$, as the case $w_a > w_b$ is symmetric. Since $P(t)$ and $P(t')$ are both on the edge $\overline{w_b w_a}$, the fact that $t \leq t'$ implies that $P(t') \leq P(t)$. Using the facts that $|v_i - w_a| \leq \delta$ and $|v_j - w_b| \leq \delta$, we obtain

$$v_i - \delta \leq w_a \leq P(t') \leq P(t) \leq w_b \leq v_j + \delta.$$

At the same time we have

$$v_j - \delta \leq P(t') \leq P(t) \leq v_i + \delta.$$

It follows that $|v_i - v_j| \leq 2\delta$.

Thus, the claim that v_i is contained in the 3δ-range of w_b is then implied by triangle inequality, as well as the symmetric claim that v_j is contained in the 3δ-range of w_a. As v_i and v_j are both matched to the edge $\overline{w_b w_a}$, we also have that v_i and v_j visit the 3δ-ranges of w_b and w_a, respectively. □

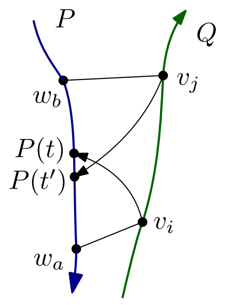

Figure 3: Illustration to the proof of Lemma 42. Assuming $w_a < w_b$ as in the proof, v_i visits w_a and v_j visits w_b, but $i < j$ and $a > b$, so the visiting sequence is not monotone.

Lemma 43. *Let $P : [0,1] \to \mathbb{R}$ and $Q : [0,1] \to \mathbb{R}$ be curves and let $\phi : [0,1] \to [0,1]^2$ be a monotone traversal that maps them within distance δ. Let S be a subsequence of the vertices of Q. Any δ-visiting sequence of S on P under ϕ implies a 3δ-visiting order of S on P.*

Proof. Let $u_1, \ldots, u_\ell$ denote the vertices of S and let $i_1, \ldots, i_\ell$ denote the visiting sequence. We generate a monotonically increasing sequence as follows. For every u_j, we set i_j to the minimum of the suffix sequence $i_j, \ldots, i_\ell$. If i_j was already a minimum, then nothing changes. Otherwise, let i_k be an index, where this minimum was attained. By Lemma 42 the vertex u_j is contained in the 3δ-range of the vertex v_{i_k}. After applying this to all elements of the sequence, starting with $j = 1$ and ending with $j = \ell$, the sequence $i_1, \ldots, i_\ell$ is monotonically increasing. □

The next two lemmas are used in the proof of Lemma 22.

Lemma 44. *Let $P : [0,1] \mapsto \mathbb{R}$ and $Q : [0,1] \mapsto \mathbb{R}$ be curves such that $\mathrm{d_F}(Q,P) \leq \delta$ and let ϕ be a monotone traversal realizing this distance. If none of the inner vertices of P and Q δ-visit each other under ϕ, then P and Q are 2δ-monotone.*

Proof. We prove the lemma by induction. We reconstruct the matching ϕ and use "matched" as shorthand for "matched under ϕ". Recall that we denote the ordered vertices of P and Q by $p_1, p_2, \ldots$ and $q_1, q_2, \ldots$, respectively. Note that if either P or Q consist of a single vertex or single segment, then the claim immediately follows from Observation 36. Otherwise, either p_2 is matched to a point on $\overline{q_1 q_2}$ or q_2 is matched to a point on $\overline{p_1 p_2}$ and p_2, q_2 are inner vertices. As the lemma statement is symmetric with respect to P and Q, we assume without loss of generality that p_2 is matched to $\overline{q_1 q_2}$. As p_2 and q_2 are inner vertices, they cannot δ-visit each other, and thus either $p_2 < q_2 - \delta$ or $p_2 > q_2 + \delta$. By mirroring the curves P and Q at the origin, these two cases are symmetric, and we thus assume $p_2 < q_2 - \delta$ without loss of generality. As p_2 is matched to $\overline{q_1 q_2}$, it follows that $q_1 < q_2$. Thus, $\overline{q_1 q_2}$ is increasing and $\overline{p_1 p_2}$ has to be 2δ-monotone increasing as otherwise the matching would have distance larger than δ. Now, for the inductive step, assume that $\langle p_1, \ldots, p_i \rangle$ and $\langle q_1, \ldots, q_j \rangle$ are 2δ-monotone increasing curves, p_i, q_j are inner vertices, and p_i is matched to a point on $\overline{q_{j-1} q_j}$ with $p_i < q_j - \delta$. Note that this again implies $q_{j-1} < q_j$.

Let us now prove the inductive step. If p_{i+1} is an inner vertex, then either (i) p_{i+1} is also matched to a point on $\overline{q_{j-1} q_j}$ or (ii) q_j is matched to a point on $\overline{p_i p_{i+1}}$.

In case (i), p_{i+1} extends a subcurve $\langle p_{i'}, \ldots, p_i \rangle$ with $i' \geq 1$ that is completely matched to a part of the increasing segment $\overline{q_{j-1} q_j}$. The subcurve $\langle p_{i'}, \ldots, p_i \rangle$ has to be 2δ-monotone increasing according to Observation 36. Either $p_{i'}$ is the start of P (i.e, $i' = 1$) and thus $\langle p_1, \ldots, p_{i+1} \rangle$ is 2δ-monotone increasing, or $p_{i'-1}$ has to be matched to a part of Q before q_{j-1} and thus q_{j-1} is an inner vertex. As q_{j-1} was already matched, it follows that either $p_{i'-1}$ is the start of P (i.e.,

$i' - 1 = 1$) and $p_{i'-1} \leq q_{j-1} + \delta$, or $p_{i'-1}$ is an inner vertex and $p_{i'-1} < q_{j-1} - \delta$ as they do not δ-visit each other. In both cases $\langle p_1, \ldots, p_{i'-1} \rangle$ is contained in $[-\infty, q_{j-1} + \delta)$; for the first case this holds as $\langle p_1, \ldots, p_{i'-1} \rangle$ is 2δ-monotone increasing by induction. Consequently, the concatenation of $\langle p_1, \ldots, p_{i'-1} \rangle$ and $\langle p_{i'}, \ldots, p_{i+1} \rangle$ is also 2δ-monotone increasing.

Now consider case (ii), i.e., q_j is matched to a point on $\overline{p_i p_{i+1}}$. In this case $\overline{p_i p_{i+1}}$ is increasing as $p_i < q_j$ and $q_j < p_{i+1}$, which is the case because q_j is matched to $\overline{p_i p_{i+1}}$ and $p_i < q_j - \delta$. Therefore, also in this case it holds that $\langle p_1, \ldots, p_{i+1} \rangle$ is 2δ-monotone increasing. Note that after exchanging P and Q, we again fulfill the inductive hypothesis. In particular, since q_j is matched to $\overline{p_i p_{i+1}}$ but p_{i+1} and q_j do not δ-visit each other as both are inner vertices, we must have $q_j < p_{i+1} - \delta$.

Now consider the case that p_{i+1} is not an inner vertex, i.e., it is the last vertex of P. In this case, part of $\overline{p_i p_{i+1}}$ has to be matched to q_j as no previous part of P was matched to q_j. This implies that $\overline{p_i p_{i+1}}$ again is increasing as $p_i < q_j - \delta$ and $p_{i+1} \geq q_j - \delta$. Hence $\langle p_1 \ldots p_{i+1} \rangle$ is 2δ-monotone increasing. As the remainder of Q, starting from q_j, has to be matched to part of $\overline{p_i p_{i+1}}$ and therefore this part is 2δ-monotone increasing by Observation 36, and $\langle q_1, \ldots, q_{j-1} \rangle$ is 2δ-monotone by induction and also contained in $[-\infty, p_i - \delta)$ as p_i is matched to the increasing $\overline{q_{j-1} q_j}$, it follows that the whole curve Q is 2δ-monotone increasing. ☐

Lemma 45. *Let $P : [0,1] \mapsto \mathbb{R}$ and $Q : [0,1] \mapsto \mathbb{R}$ be curves such that $\mathrm{d_F}(Q, P) \leq \delta$ and let ϕ be a monotone traversal realizing this distance. Further assume that for all $t \in [0,1]$ we have $Q(t) \in \overline{Q(0)Q(1)}$. If none of the inner vertices of Q δ-visit an inner vertex of P under ϕ, then the line segment $Q' = \overline{Q(0)Q(1)}$ is a range-preserving δ-simplification of Q with $\mathrm{d_F}(Q', P) \leq \delta$.*

Proof. By Lemma 44, Q and P must be 2δ-monotone. Moreover, Q' is range-preserving by assumption. Therefore, Q' is a range-preserving δ-simplification of Q. It remains to show the bound on the Fréchet distance of P and Q'. To this end, we want to invoke Observation 36. Indeed, it must be that

$$\forall t \in [0,1] : P(t) \in \bigcup_{s \in [0,1]} B(Q(s), \delta),$$

since $\mathrm{d_F}(P, Q) \leq \delta$ and since Q' is range-preserving. Therefore, the conditions of Observation 36 are satisfied and the bound is implied. ☐

We are now ready to prove Lemma 22 from Section 3.

Lemma 22. *Let $P : [0,1] \mapsto \mathbb{R}$ and $Q : [0,1] \mapsto \mathbb{R}$ be curves such that $\mathrm{d_F}(Q, P) \leq \delta$, there exists a δ-straightening Q' of Q which satisfies the following properties:*
(i) there exists a 11δ-visiting order of Q' on P, and
(ii) $\mathrm{d_F}(Q', P) \leq \delta$.

Proof. Let ϕ be a monotone traversal that realizes the Fréchet distance between P and Q. We will construct a δ-straightening Q' together with a $\mathcal{O}(\delta)$-visiting order of Q' on P. To this end, consider the subset of vertices of Q that each δ-visit *some* vertex of P under ϕ (Definition 40). Denote this subset by S. Lemma 43 implies that there exists a 3δ-visiting order of S on P. We denote this visiting order by the function $\kappa : S \to [m]$ that assigns every vertex of S the index of a vertex of P (where m denotes the number of vertices of P).

It is quite possible that S is not a δ-simplification of Q with the desired properties. In a second phase of the construction we will therefore add more vertices of Q to S. Consider any maximal subcurve $Q[s, s']$ of Q, such that none of the inner vertices of $Q[s, s']$ δ-visit a vertex of P under ϕ. It must be that $Q(s)$ corresponds to some vertex w of S and $Q(s')$ corresponds to some vertex w' of S. Moreover, w' comes directly after w along Q among the vertices included in S. Assume that $Q[s, s']$ has at least one inner vertex. We distinguish two cases:

(C1) $B(v_{\kappa(w)}, 3\delta) \cap B(v_{\kappa(w')}, 3\delta) \neq \emptyset$,

(C2) otherwise

In the first case (C1), we will add all inner vertices $Q[s, s']$ to S and assign them the index $\kappa(w)$ in the constructed visiting order κ. In the second case (C2), we will only add a specific subset of vertices, which we define as follows. Define α and β as follows:

$$\alpha = \max\{t \mid t \in [s, s'] \text{ and } Q(t) \in B(v_{\kappa(w)}, 3\delta)\}$$

$$\beta = \min\{t \mid t \in [\alpha, s'] \text{ and } Q(t) \in B(v_{\kappa(w')}, 3\delta)\}$$

Since the 3δ-ranges of $v_{\kappa(w)}$ and $v_{\kappa(w')}$ are disjoint, α and β are well-defined and it follows by definition that $s \leq \alpha \leq \beta \leq s'$. Therefore, the subcurves $Q[s, \alpha]$, $Q[\alpha, \beta]$, and $Q[\beta, s']$ are well-defined. Now, we proceed as follows, we add the inner vertices of $Q[s, \alpha]$ to S and assign them the index $\kappa(w)$ in the constructed visiting order κ. Secondly, we add the inner vertices of $Q[\beta, s]$ to S and assign them the index $\kappa(w')$ in the constructed visiting order κ.

We apply this to all such maximal subcurves $Q[s, s']$ (note that these are pairwise disjoint), thereby constructing the sequence S along with the visiting order κ. Let $u_1, \ldots, u_\ell$ be the sequence of vertices of the resulting S in their order along Q. Denote with Q' the curve that results from linearly interpolating $u_1, \ldots, u_\ell$. Note that it is different from Q only in the sections where we omitted the vertices of the subcurve $Q[\alpha, \beta]$ in case (C2). We claim that Q' is an edge-range-preserving δ-simplification of Q. To see this, consider a subcurve $Q[s, s']$, assume we are in case (C2). By construction, the subcurve $Q[\alpha, \beta]$ is range-preserving (for all $x \in [\alpha, \beta]$ we have $Q(x) \in \overline{Q(\alpha)Q(\beta)}$). Let $P[t, t']$ be a subcurve of P mapped to $Q[\alpha, \beta]$ under ϕ. Now, Lemma 45 applied to the subcurves $P[t, t']$ and $Q[\alpha, \beta]$ implies that $\overline{Q(\alpha)Q(\beta)}$ is an edge-range-preserving δ-simplification of $Q[\alpha, \beta]$ with $\mathrm{d_F}(\overline{Q(\alpha)Q(\beta)}, P[t, t']) \leq \delta$. Therefore, by Observation 4, when removing all vertices of Q in the parameter range (α, β) for each such maximal subcurve $Q[s, s']$, we obtain a δ-straightening Q' of Q with $\mathrm{d_F}(Q', P) \leq \delta$.

Finally, we argue that the constructed visiting order $\kappa(u_1), \ldots, \kappa(u_\ell)$ is an 11δ-visiting order of Q' on P. Clearly it is monotonically increasing by construction. Also, it is clear that any vertex added in the first phase is contained in the 3δ-range of its assigned vertex of P. It remains to argue for any vertex added to S in the second phase, that it is contained in the 11δ-range of its assigned vertex in P. Consider a subcurve $Q[s, s']$ from above and assume we are in case (C1). We have that $Q(s) \in B(v_{\kappa(w)}, 3\delta)$ and $Q(s') \in B(v_{\kappa(w')}, 3\delta)$. By the case distinction, these two ranges are not disjoint. Therefore, the subcurve starts and ends in the 9δ-range of the assigned vertex $v_{\kappa(w)}$. Moreover, by Lemma 44, $Q[s, s']$ has to be 2δ-monotone. This implies that the entire subcurve lies in the 11δ-range of $v_{\kappa(w)}$ and this is also the vertex that we assigned to all of its inner vertices. A similar argument can be applied in case (C2). By the way we chose α, we have that $Q(\alpha)$ is contained in the 3δ-range of $v_{\kappa(w)}$, which is also the vertex assigned to the entire subcurve. Since also the subcurve $Q[s, \alpha]$ is 2δ-monotone, all remaining vertices in the range $[s, \alpha]$ are contained in the 5δ-range of the same vertex. A symmetric argument can be applied to show that all remaining vertices in the range $[\beta, s']$ are contained in the 5δ-range of their assigned vertex. □

Finally, we also prove Lemma 23 from Section 3.

Lemma 23. *For one-dimensional curves P, Q let P' be a δ-signature of P, and let Q' be a 2δ-signature of Q. If $\mathrm{d_F}(P, Q) \leq \delta$ then $\mathrm{d_F}(P', Q') \leq 3\delta$ and there exists a 2δ-visiting order of Q' on P'.*

Proof. By the triangle inequality we have that $\mathrm{d_F}(P', Q) \leq \mathrm{d_F}(P', P) + \mathrm{d_F}(P, Q) \leq 2\delta$. Now Lemma 18 applied to P' and the 2δ-signature of Q implies that there exists a 2δ-visiting order of Q' on P'.

It remains to argue that $\mathrm{d_F}(P', Q') \leq 3\delta$. Let $\phi : [0,1] \to [0,1]^2$ be a δ-traversal of P and Q. Consider an edge X of Q' and let $Q[\alpha, \beta]$ be the subcurve of Q that corresponds to X. Let $P[\alpha', \beta']$ be a subcurve of P that is mapped to $Q[\alpha, \beta]$ under ϕ. By the triangle inequality

$$\mathrm{d_F}(P[\alpha', \beta'], X) \leq \mathrm{d_F}(P[\alpha', \beta'], Q[\alpha, \beta]) + \mathrm{d_F}(Q[\alpha, \beta]), X) \leq 3\delta$$

Assume that P' is range-preserving for now (we will treat the general case below) and let $P'[\alpha'', \beta'']$ be the corresponding subcurve of P' starting at $P(\alpha')$, ending at $P(\beta')$, and with inner vertices being the δ-signature vertices of P in the parametrization interval $[\alpha', \beta']$. Note that $P'[\alpha'', \beta'']$ is well-defined since P' is a range-preserving as assumed above. By Observation 5 it follows that $\mathrm{d_F}(P'[\alpha'', \beta''], X) \leq 3\delta$. To show the claim for the case of range-preserving P', we now want to use Observation 4 to concatenate the corresponding subcurves of P' and Q' and obtain that $\mathrm{d_F}(P', Q') \leq 3\delta$. For this, we can choose the values of α' and β' in the above argument such that we obtain a decomposition of P into subcurves. Concretely, let $X_1, \ldots, X_s$ be the edges of Q' in their order along Q', with $X_i = \overline{Q(\alpha_i)Q(\beta_i)}$. Then, we can choose the corresponding subcurves of P as $P[\alpha'_i, \beta'_i]$, with

$$\alpha'_{i-1} \leq \beta'_{i-1} = \alpha'_i \leq \beta'_i$$

for any $1 < i \leq s$, with $\alpha'_1 = 0$ and $\beta'_s = 1$. Thus, we obtain a decomposition of P. Now, if P' is a range-preserving simplification of P, then the above construction induces a decomposition of P' into subcurves $P'[\alpha''_i, \beta''_i]$ and we can apply Observation 4.

As noted above, P' is not necessarily range-preserving on all edges since it is a signature. In particular, it may not be range-preserving on the first edge (or the last edge, or neither). This could lead to $P(\alpha'_2)$ (resp. $P(\alpha'_s)$ for the last edge) not being included in the image of the signature edge of P' that corresponds to the subcurve of P containing α'_2 (resp., α'_s). Note that if $P(\alpha'_2)$ is not contained in the image of the first signature edge, then it must be that $|P(\alpha'_2) - P(0)| \leq \delta$, and in fact, it must be that this holds for the entire subcurve, that is $|P(t) - P(0)| \leq \delta$ for any $t \in [0, \alpha'_2]$. We claim that in this case we can simply set α''_2, and β''_1 to 0 (resp., we can set β''_{s-1}, and α''_s to 1). We argue that this way of choosing the decomposition leads to $\mathrm{d_F}(P'[\alpha''_1, \beta''_1], X_1) \leq 3\delta$ and $\mathrm{d_F}(P'[\alpha''_2, \beta''_2], X_2) \leq 3\delta$ so that the above arguments can be applied (for the last two edges of Q' a symmetric argument can be applied and we will omit the explicit analysis).

By the triangle inequality, we have that

$$|P'(0) - Q(\alpha_2)| \leq |P(0) - P(\alpha'_2)| + |P(\alpha'_2) - Q(\alpha_2)| \leq 2\delta.$$

Together with

$$|P'(0) - Q(\alpha_1)| = |P(0) - Q(0)| \leq \delta$$

this implies by Observation 3 that $\mathrm{d_F}(P'(0), X_1) \leq 3\delta$ since X_1 is a line segment and $X_1 = \overline{Q(0)Q(\alpha_2)}$. Applying the triangle inequality again, we obtain for any $t \in [0, \alpha'_2]$ that

$$|P(t) - Q(\alpha_2)| \leq |P(t) - P(0)| + |P(0) - Q(\alpha_2)| \leq 3\delta$$

By Observation 4 and since $X_2 = \overline{Q(\alpha_2)Q(\beta_2)}$, this implies that

$$\mathrm{d_F}(P[0, \beta'_2], X_2) \leq \max\Big(\mathrm{d_F}(P[0, \alpha'_2], Q(\alpha_2)) \ , \ \mathrm{d_F}(P[\alpha'_2, \beta'_2], \overline{Q(\alpha_2)Q(\beta_2)}) \Big) \leq 3\delta$$

By Observation 5 it follows that $\mathrm{d_F}(P'[0, \beta''_2], X_2) \leq 3\delta$.

□

8 Lower Bounds

In this section we show several conditional lower bounds for $(2-\varepsilon)$ and $(3-\varepsilon)$-approximate nearest neighbor data structures. We use the well-known Orthogonal Vectors problem as the problem that we base our hardness results on.

Definition 46 (Orthogonal Vectors (OV))**.** *Given two sets of vectors $A, B \subseteq \{0,1\}^d$, do there exist two vectors $a \in A, b \in B$ such that $\langle a, b\rangle = 0$?*

Definition 47 (Orthogonal Vectors Hypothesis (OVH))**.** *For all $\varepsilon > 0$ there exists a $c > 0$ such that there is no algorithm solving OV instances $A, B \subset \{0,1\}^d$ with $|A| = |B|$ and $d = c \log |A|$ in time $\mathcal{O}(|A|^{2-\varepsilon})$.*

The above hypothesis is also sometimes called the Low-Dimensional Orthogonal Vectors Hypothesis and it is implied by the Strong Exponential Time Hypothesis [Wil05]. We use this version of the Orthogonal Vectors Hypothesis as it allows us to rule out running times using an arbitrarily small ε while still reducing from an instance where vectors have a logarithmic dimension. It is well known that *balanced* OV with sets of the same size is equally hard as *unbalanced* OV [AW14, BK18].

Lemma 48 (Unbalanced Orthogonal Vectors Hypothesis)**.** *Assume OVH holds true. For every $\alpha \in (0,1)$ and $\varepsilon > 0$ there exists a $c > 0$ such that there is no algorithm solving OV instances $A, B \subset \{0,1\}^d$ with $|B| = |A|^\alpha$ and $d = c \log |A|$ in time $\mathcal{O}(|A|^{1+\alpha-\varepsilon})$.*

Proof sketch. We briefly outline why this hardness holds. To that end, assume that we can solve the unbalanced case in time $\mathcal{O}(|A|^{1+\alpha-\varepsilon})$ for some $\varepsilon > 0$. Then we could solve the balanced case by splitting B into $|A|^{1-\alpha}$ parts of size $|A|^\alpha$, solve these instances in time $\mathcal{O}(|A|^{1+\alpha-\varepsilon})$, and thus solve the balanced problem in time $\mathcal{O}(|A|^{1-\alpha} \cdot |A|^{1+\alpha-\varepsilon}) = \mathcal{O}(|A|^{2-\varepsilon})$. □

Leveraging this insight, we later reduce from unbalanced OV instances to show stronger hardness results. For convenience, we introduce some additional notation. For a vector $a \in \{0,1\}^d$, we use $a[i]$ to refer to its ith entry, where the entries are 0-index, i.e., $a = (a[0], \ldots, a[d-1])$. Recall that we use the "$\circ$" operator to concatenate curves and that the curve P where each point is translated by τ is denoted as $P + \tau$.

Instead of reducing directly from OV, we introduce a novel problem called OneSidedSparseOV and show that it is hard under OV. Subsequently, we reduce from this problem to the ANN problems introduced above.

8.1 OneSidedSparseOV

This problem can be thought of as a variant of OV with an additional restriction on one of the input sets. More precisely, for one set we allow at most k non-zero entries in each vector.

Definition 49 (OneSidedSparseOV)**.** *Given a value $k \in \mathbb{N}$ and two sets of vectors $A, B \subseteq \{0,1\}^d$ where each $a \in A$ contains at most k non-zero entries, do there exist two vectors $a \in A, b \in B$ such that $\langle a, b\rangle = 0$?*

We also refer to OneSidedSparseOV with parameter k as OneSidedSparseOV(k). We now show that this problem is hard under OV, interestingly, this is already the case for $k \in \omega(1)$.

Lemma 50. *Assume OVH holds true. For every $\alpha \in (0,1)$, $\varepsilon > 0$ there is a $c > 0$ such that for any $k \in \omega(1) \cap o(\log |A|)$ there is no algorithm solving* OneSidedSparseOV(k) *instances $A, B \subset \{0,1\}^d$ with $|B| = |A|^\alpha$ and $d = k \cdot |A|^{c/k}$ in time $\mathcal{O}(|A|^{1+\alpha-\varepsilon})$.*

Proof. For any $\alpha \leq 1, \varepsilon > 0$, let $c > 0$ be the constant from Lemma 48. Thus, unless OVH fails, we cannot solve OV instances $A, B \subset \{0,1\}^d$ with $|B| = |A|^\alpha$ and $d = c \log |A|$ in time $\mathcal{O}(|A|^{1+\alpha-\varepsilon})$. For any $k \in \omega(1) \cap o(\log |A|)$, we now reduce to $\textsc{OneSidedSparseOV}(k)$ as follows. We convert A to a *set of sparse vectors* A' and B to a set B' such that A', B' is an equivalent $\textsc{OneSidedSparseOV}(k)$ instance. To achieve this, we increase the dimensionality of the vectors in the $\textsc{OneSidedSparseOV}$ instance. Given a vector $a \in A$, partition the dimensions of a into k blocks of size d/k.[1] More precisely, let

$$a_i = \left(a\left[i \cdot \frac{d}{k}\right], a\left[i \cdot \frac{d}{k} + 1\right], \ldots, a\left[i \cdot \frac{d}{k} + \frac{d}{k} - 1\right]\right)$$

for $i \in \{0, \ldots, k-1\}$. Let $\hat{a}_i \in \left[2^{d/k}\right]$ be defined as the binary vector a_i interpreted as a binary number. We now construct the corresponding $a' \in A'$ as follows. We choose the dimension of the vectors in A', B' as $d' = k \cdot 2^{d/k}$ — note that this equals $k \cdot |A|^{c/k}$ as stated in the lemma. For each $i \in \{0, \ldots, k-1\}$, we set $a'[i \cdot 2^k + \hat{a}_i] = 1$. All other entries of a' are set to 0. Thus, each vector $a' \in A'$ contains exactly k 1-entries. The vectors $b' \in B'$ we construct as follows. Given a vector $b \in B$, we also partition its dimensions the same way as we did for $a \in A$ and obtain vectors $b_0, \ldots, b_{k-1}$. For each $i \in \{0, \ldots, k-1\}$ and all $\beta \in \{0,1\}^{d/k}$ — where we again use $\hat{\beta}$ to denote β being interpreted as a binary number — we set $b'[i \cdot 2^k + \hat{\beta}] = 1$ if $\langle b_i, \beta \rangle > 0$, otherwise we set it to zero. This completes the description of the reduction. Note that while we changed the dimension of the vectors, the size of the sets remained the same, that is $|A'| = |A|$ and $|B'| = |B|$.

Note that for any vectors $a \in A$ and $b \in B$ with $\langle a, b \rangle > 0$ there exist parts a_i, b_i and a coordinate ℓ such that $a_i[\ell] = b_i[\ell] = 1$, and thus $\langle a_i, b_i \rangle > 0$. Hence, by construction of b', there exists a dimension in a' and b' where both have a 1. On the other hand, if a' and b' contain a 1 in the same dimension, then by construction of b' there have to be two parts a_i, b_i such that $\langle a_i, b_i \rangle > 0$ and thus $\langle a, b \rangle > 0$.

The total running time of this reduction consists of constructing the vectors in A' — which takes time proportional to the number of entries — and the inner product computation between vectors of dimensionality d/k for each of the $k \cdot 2^{d/k}$ dimensions of each vector in B':

$$\mathcal{O}\left(|A'| \cdot k \cdot 2^{d/k} + |B'| \cdot k \cdot 2^{d/k} \cdot \frac{d}{k}\right) = \mathcal{O}\left(|A| \cdot 2^{c \log |A| / k} \cdot c \log |A|\right) = \mathcal{O}\left(|A|^{1+c/k} \cdot c \log |A|\right),$$

which simplifies to $|A|^{1+o(1)}$ as $k \in \omega(1)$ and $\log |A| = \mathcal{O}(|A|^{o(1)})$. Thus, if indeed we can solve $\textsc{OneSidedSparseOV}(k)$ in time $\mathcal{O}(|A'|^{1+\alpha-\varepsilon})$ and add the running time of the reduction, then we can solve unbalanced OV in time

$$\mathcal{O}(|A'|^{1+\alpha-\varepsilon}) + |A|^{1+o(1)} = \mathcal{O}(|A|^{1+\alpha-\varepsilon}),$$

which would refute OVH. □

Using this insight, we now proceed to proving hardness results for different approximation ratios for ANN under the continuous Fréchet distance.

8.2 Hardness of $(2-\varepsilon)$-Approximation in 1D

In this section we present our first hardness result. We note that the gadgets that we use to encode our vectors are inspired by [DP20].

[1] If d is not divisible by k, increase the dimension until this is the case and fill these dimensions with zeros.

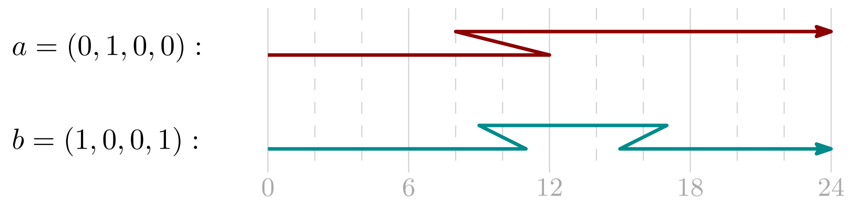

Figure 4: Visualization of the $2 - \varepsilon$ lower bound in 1D.

Theorem 51. *Assume OVH holds true. For any $\varepsilon, \varepsilon' > 0$ there is a $c > 0$, such that there is no $(2 - \varepsilon)$-ANN for the continuous Fréchet distance supporting query curves of any complexity $k \in \omega(1) \cap o(\log n)$ and storing n one-dimensional curves of complexity $m = k \cdot n^{c/k}$ with preprocessing time poly(n) and query time $\mathcal{O}(n^{1-\epsilon'})$.*

Proof. We show the hardness by a reduction from OneSidedSparseOV(k). To that end, let $A, B \subset \{0,1\}^d$ be a OneSidedSparseOV(k) instance with $|B| = |A|^\alpha$ for a constant $\alpha \le 1$ that we specify later, $k \in \omega(1) \cap o(\log |A|)$, and $d = k \cdot |A|^{c/k}$ with a constant $c > 0$ that we later choose sufficiently large. Recall that, by Lemma 50, there exists a $c > 0$ such that OneSidedSparseOV(k) is OV-hard in this regime. The goal is to use the k-sparsity of the vectors in A to obtain short query curves of length $\mathcal{O}(k)$.

Let us first give the reduction. To that end, we define the following subcurves:

$$0_A := \langle 0, 6 \rangle\,, \quad 1_A := \langle 0, 6, 2, 6 \rangle\,, \quad 0_B := \langle 0, 5, 3, 6 \rangle\,, \quad 1_B := \langle 0, 6 \rangle$$

Now, given a OneSidedSparseOV(k) instance A, B, we create the input set $\mathcal{P}$ and the query set $\mathcal{Q}$ of a $(2 - \varepsilon)$-ANN instance with distance threshold $\delta = 1$ as follows. For each vector $a \in A$, we add the curve Q_a to $\mathcal{Q}$ which is defined as

$$Q_a := \bigcirc_{i=0}^{d-1} V_a^i \quad \text{with} \quad V_a^i := a[i]_A + 6i,$$

where $a[i]_A$ is either 0_A or 1_A, depending on the value of $a[i]$, and the "$+6i$" is a translation of each point of the curve by $6i$. For each vector $b \in B$, we add the curve P_b to $\mathcal{P}$ which is defined as

$$P_b := \bigcirc_{i=0}^{d-1} V_b^i \quad \text{with} \quad V_b^i := b[i]_B + 6i,$$

where $b[i]_B$ is either 0_B or 1_B, depending on the value of $b[i]$. It is crucial that we make the resulting curves non-degenerate by removing all degenerate vertices. In particular, all connecting vertices between gadget curves will be removed and any sequence of consecutive gadgets 0_A will be turned into a single line segment. Thus, the curves in $\mathcal{Q}$ will have complexity $\mathcal{O}(k)$. See Figure 4 for an example of the construction.

We now show correctness of the reduction. Let $P_b \in \mathcal{P}$ and $Q_a \in \mathcal{Q}$ be any curves in these sets. Note that if $\mathrm{d_F}(P_b, Q_a) < 2$, then if the traversal is a distance 2 into the gadget V_a^i, then we also have to be in the gadget V_b^i, as there is no other gadget in distance less than 2. The same statement holds for V_a^i and V_b^i exchanged. Thus, we traverse the gadgets synchronously. Now consider the case $\langle a, b \rangle = 0$. As $\mathrm{d_F}(0_A, 0_B) = \mathrm{d_F}(0_A, 1_B) = \mathrm{d_F}(1_A, 0_B) = 1$, also $\mathrm{d_F}(P_b, Q_a) = 1$, as there is no $i \in \{0, \dots, d-1\}$ for which the gadget V_a^i is of type 1_A and V_b^i is of type 1_B. Conversely, consider the case $\langle a, b \rangle = 1$. Then there exist an $i \in \{0, \dots, d-1\}$ such that V_a^i is of type 1_A and V_b^i is of type 1_B. As we traverse the gadgets synchronously and as $\mathrm{d_F}(V_b^i, V_a^i) = 2$, we have $\mathrm{d_F}(P_b, Q_a) = 2$.

Thus, if we have a $(2-\varepsilon)$-ANN, then we can use it to check if there exist orthogonal vectors $a \in A$ and $b \in B$ by the above reduction.

It remains to show that this reduction implies the claimed lower bound. The time to compute the reduction is linear in the output size and thus negligible. Recall that $\mathcal{P}$ is the input set, i.e., it is the set that we preprocess, and we run a query for each curve in $\mathcal{Q}$. Note that by the construction of the above reduction we have $|\mathcal{P}| = |A|^\alpha$, and $|\mathcal{Q}| = |A|$. Towards a contradiction, assume that we can solve $(2-\varepsilon)$-ANN with preprocessing time $\mathcal{O}(|\mathcal{P}|^{\alpha'})$ for some $\alpha' > 0$ and query time $\mathcal{O}(|\mathcal{P}|^{1-\varepsilon'})$ for some $\varepsilon' > 0$. Choosing $\alpha = 1/\alpha'$, we obtain preprocessing time $\mathcal{O}(|\mathcal{P}|^{\alpha'}) = \mathcal{O}(|A|^{\alpha\alpha'}) = \mathcal{O}(|A|)$ and total query time

$$\mathcal{O}(|\mathcal{Q}| \cdot |\mathcal{P}|^{1-\varepsilon'}) = \mathcal{O}(|A| \cdot |A|^{\alpha(1-\varepsilon')}) = \mathcal{O}(|A|^{1+\alpha-\varepsilon'\alpha}).$$

Thus, we could solve OneSidedSparseOV(k) in time $\mathcal{O}(|A|^{1+\alpha-\varepsilon'\alpha})$. However, by Lemma 50, there exists a $c > 0$ such that this contradicts OVH. □

8.3 Hardness of $(3-\varepsilon)$-Approximation in 1D

We now show the first of two hardness results that rule out certain preprocessing and query times for $(3-\varepsilon)$-approximations. Note that ruling out higher approximation ratios is not possible using gadgets that encode the single coordinates, as the distance between the gadgets that encode 1-entries can be at most 3 times the threshold distance due to the triangle inequality between the other gadgets, for details see [BOS19]. For one-dimensional curves we obtain the following lower bound. We note that the gadgets that we use to encode our vectors are inspired by [BOS19].

Theorem 52. *Assume OVH holds true. For any $\varepsilon, \varepsilon' > 0$ there is a $c > 0$, such that there is no $(3-\varepsilon)$-ANN for the continuous Fréchet distance storing n one-dimensional curves of complexity m and supporting query curves of complexity k with $m = k = c \log n$ such that we have preprocessing time poly(n) and query time $\mathcal{O}(n^{1-\epsilon'})$.*

Proof. We show the hardness by a reduction from OV. To that end, let $A, B \subset \{0,1\}^d$ be an OV instance with $|B| = |A|^\alpha$ for a constant $\alpha \leq 1$ that we specify later and $d = c \log |A|$ for a constant $c > 0$ that we later choose sufficiently large. Recall that, by Lemma 48, there exists a $c > 0$ such that this problem is OV-hard. We now create the input set $\mathcal{P}$ and query set $\mathcal{Q}$ of a $(3-\varepsilon)$-ANN instance with distance threshold $\delta = 1$ as follows. For convenience, we define the curves

$$1_A \coloneqq \langle 0, 6, 0 \rangle, \quad 0_B \coloneqq \langle 0, 7, 0 \rangle, \quad 0_A \coloneqq \langle 0, 8, 0 \rangle, \quad 1_B \coloneqq \langle 0, 9, 0 \rangle.$$

First, for each vector $a \in A$ we create a new curve $Q_a \in \mathcal{Q}$ defined as

$$Q_a \coloneqq \bigcirc_{i=0}^{d-1} V_a^i \quad \text{with} \quad V_a^i \coloneqq a[i]_A,$$

where $a[i]_A$ is either 0_A or 1_A, depending on the value of $a[i]$. Second, for each vector $b \in B$ we create a new curve $P_b \in \mathcal{P}$ defined as

$$P_b \coloneqq \bigcirc_{i=0}^{d-1} V_b^i \quad \text{with} \quad V_b^i \coloneqq b[i]_B,$$

where $b[i]_B$ is either 0_B or 1_B, depending on the value of $b[i]$. See Figure 5 for examples of these curves.

We now prove the correctness of the reduction. Consider any $Q_a \in \mathcal{Q}$ and $P_b \in \mathcal{P}$. We first show that if $\mathrm{d_F}(Q_a, P_b) < 3$, then any traversal realizing this distance has to visit vertices of both

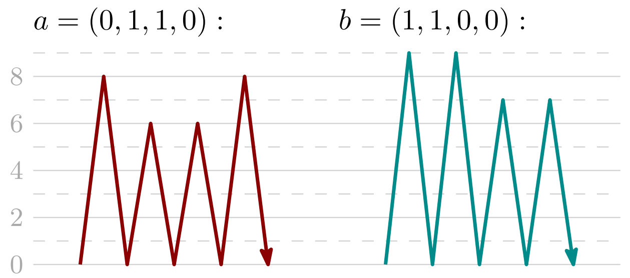

Figure 5: Visualization of the $3 - \varepsilon$ lower bound in 1D.

curves synchronously. More precisely, a traversal can be in the gadgets V_a^i and V_b^j with $i \neq j$ only if the positions on both curves are strictly less than 6 in image space. Towards a contradiction, consider the first point in the traversal where this occurs and without loss of generality let the traversal be at position 6 in Q_a. As the traversal on Q_a visited 0 before, the traversal on P_b has to be below 3 and thus the positions on Q_a and P_b are within distance more than 3, which is a contradiction. Thus, when traversing gadgets V_a^i and V_b^j above 6, then $i = j$.

We now proceed with showing that for all $a \in A$ and $b \in B$ it holds that $\mathrm{d_F}(Q_a, P_b) \leq 1$ if and only if $\langle a, b \rangle = 0$, and $\mathrm{d_F}(Q_a, P_b) \geq 3$ otherwise. Assume that $\langle a, b \rangle = 0$, then, by traversing all V_a^i, V_b^i for $i \in \{0, \dots, d-1\}$ synchronously, they can always stay within distance at most 1, as $\mathrm{d_F}(0_A, 0_B) = \mathrm{d_F}(0_A, 1_B) = \mathrm{d_F}(1_A, 0_B) = 1$. However, if $\langle a, b \rangle > 0$, then there exists an index $i \in \{0, \dots, d-1\}$ such that $a[i] = b[i] = 1$. If $\mathrm{d_F}(Q_a, P_b) < 3$, then we have to traverse these V_a^i and V_b^i synchronously but as $\mathrm{d_F}(1_A, 1_B) = 3$, there is a point in the traversal where the curves have distance at least 3 and thus $\mathrm{d_F}(Q_a, P_b) \geq 3$. It follows that, if we have a $(3-\varepsilon)$-ANN, then it would find if there exists orthogonal vectors in A and B by querying each $Q \in \mathcal{Q}$.

Let us now show that this implies the desired lower bounds. The time to compute the reduction is linear in the output size and thus negligible. Note that by construction we have $m = k = \mathcal{O}(c \log |A|) \leq c' \log |A|$ for some constant $c' > 0$. By adding dummy vertices, say many points close to the starting point, we can ensure $m = k = c' \log |A|$ (we could also achieve any intended value $m \geq k$, but this is not necessary for the theorem statement). Moreover, $|\mathcal{P}| = |A|^\alpha$ and $|\mathcal{Q}| = |A|$. Towards a contradiction, assume that we can solve $(3-\varepsilon)$-ANN with preprocessing time $\mathcal{O}(|\mathcal{P}|^{\alpha'})$ for some $\alpha' > 0$ and query time $\mathcal{O}(|\mathcal{P}|^{1-\varepsilon'})$ for some $\varepsilon' > 0$. Choosing $\alpha = 1/\alpha'$, we obtain preprocessing time $\mathcal{O}(|\mathcal{P}|^{\alpha'}) = \mathcal{O}(|A|^{\alpha\alpha'}) = \mathcal{O}(|A|)$ and total query time

$$\mathcal{O}(|\mathcal{Q}| \cdot |\mathcal{P}|^{1-\varepsilon'}) = \mathcal{O}(|A| \cdot |A|^{\alpha(1-\varepsilon')}) = \mathcal{O}(|A|^{1+\alpha-\varepsilon'\alpha}).$$

Thus, we could solve unbalanced OV in time $\mathcal{O}(|A|^{1+\alpha-\varepsilon'\alpha})$. However, by Lemma 48, there exists a $c > 0$ such that this contradicts OVH. □

8.4 Hardness of $(3 - \varepsilon)$-Approximation in 2D

While until here we only considered algorithmic and hardness results for one-dimensional curves, we now show a hardness result for *two-dimensional* curves. This is the only technical section in this paper where we consider two-dimensional curves. Note that in Section 2 we defined most of our notation for curves in $\mathbb{R}^d$ and thus the notation of the previous hardness results carries over. For two-dimensional curves we obtain the following lower bound.

Theorem 53. *Assume OVH holds true. For any $\varepsilon, \varepsilon' > 0$ there is a $c > 0$, such that there is no $(3-\varepsilon)$-ANN for the continuous Fréchet distance supporting query curves of any complexity $k \in$*

$\omega(1) \cap o(\log n)$ and storing n two-dimensional curves of complexity $m = k \cdot n^{c/k}$ with preprocessing time poly(n) and query time $\mathcal{O}(n^{1-\epsilon'})$.

Proof. This proof is very similar to the proof of Theorem 51. The significant difference is the gadgets that we construct. To this end, consider a OneSidedSparseOV(k) instance A, B, where we again use the k-sparsity of the vectors in A to obtain short query curves of length $\mathcal{O}(k)$. We define the generic subcurve

$$V(y) \coloneqq \langle (0,0), (3,0), (3,y), (6,y), (6,0) \rangle$$

to then define the usual gadgets

$$0_A \coloneqq V(0), \quad 1_A \coloneqq V(2), \quad 0_B \coloneqq V(1), \quad 1_B \coloneqq V(-1).$$

Now, given a OneSidedSparseOV(k) instance A, B, we create the input set $\mathcal{P}$ and query set $\mathcal{Q}$ of a $(3-\varepsilon)$-ANN with distance threshold $\delta = 1$ as follows. For each vector $a \in A$, we add the curve Q_a to $\mathcal{Q}$ which is defined as

$$Q_a \coloneqq \bigcirc_{i=0}^{d-1} V_a^i \quad \text{with} \quad V_a^i \coloneqq a[i]_A + (6i, 0),$$

where $a[i]_A$ is either 0_A or 1_A, depending on the value of $a[i]$, and the "$+(6i,0)$" is a translation of each point of the curve by $(6i, 0)$. For each vector $b \in B$, we add the curve P_b to $\mathcal{P}$ which is defined as

$$P_b \coloneqq \bigcirc_{i=0}^{d-1} V_b^i \quad \text{with} \quad V_b^i \coloneqq b[i]_B + (6i, 0),$$

where $b[i]_B$ is either 0_B or 1_B, depending on the value of $b[i]$. It is crucial that we make the resulting curves non-degenerate by removing all degenerate vertices. In particular, any sequence of consecutive gadgets 0_A will be turned into a single line segment. Thus, the curves in $\mathcal{Q}$ will have complexity $\mathcal{O}(k)$. See Figure 6 for an example of the construction.

We now prove correctness of the reduction. Consider the case of two orthogonal vectors $a \in A$ and $b \in B$ such that there is an $i \in \{0, \ldots, d-1\}$ with $a[i] = b[i] = 1$. Note that for $\mathrm{d_F}(Q_a, P_b) < 3$, there has to be a point in the traversal where we are in some point $(x_a, 2)$ in Q_a and in some point $(x_b, -1)$ in P_b as otherwise the distance of the x-coordinate would be at least 3. However, the y-distance of these points is 3 and thus $\mathrm{d_F}(Q_a, P_b) \geq 3$. On the other hand, if $a \in A$ and $b \in B$ are orthogonal, then we can traverse the two curves with the same speed in x-direction — i.e., staying at the same x-coordinate at every point in time — and obtain a Fréchet distance at most 1 as $\mathrm{d_F}(0_A, 0_B) = \mathrm{d_F}(0_A, 1_B) = \mathrm{d_F}(1_A, 0_B) = 1$, where the described traversal realizes these distances.

The remainder of the proof, i.e., the derivation of the claimed lower bound, is the same as in the proof of Theorem 51 and we thus omit it for brevity. ◻

9 Conclusions and Open Problems

In this work we largely resolve the α-ANN problem under the continuous Fréchet distance for one-dimensional curves from a fine-grained perspective for $1 < \alpha < 3$. We show that, in general, most of the running times presented in this work cannot be improved significantly, however, other tradeoffs between preprocessing time and query time are still possible, and other parameter regimes might be shown hard or more tractable, e.g., for $k \in \mathcal{O}(1)$. Indeed, there is a line of work on related data structure problems using the continuous Fréchet distance for the specific value of $k = 2$, which

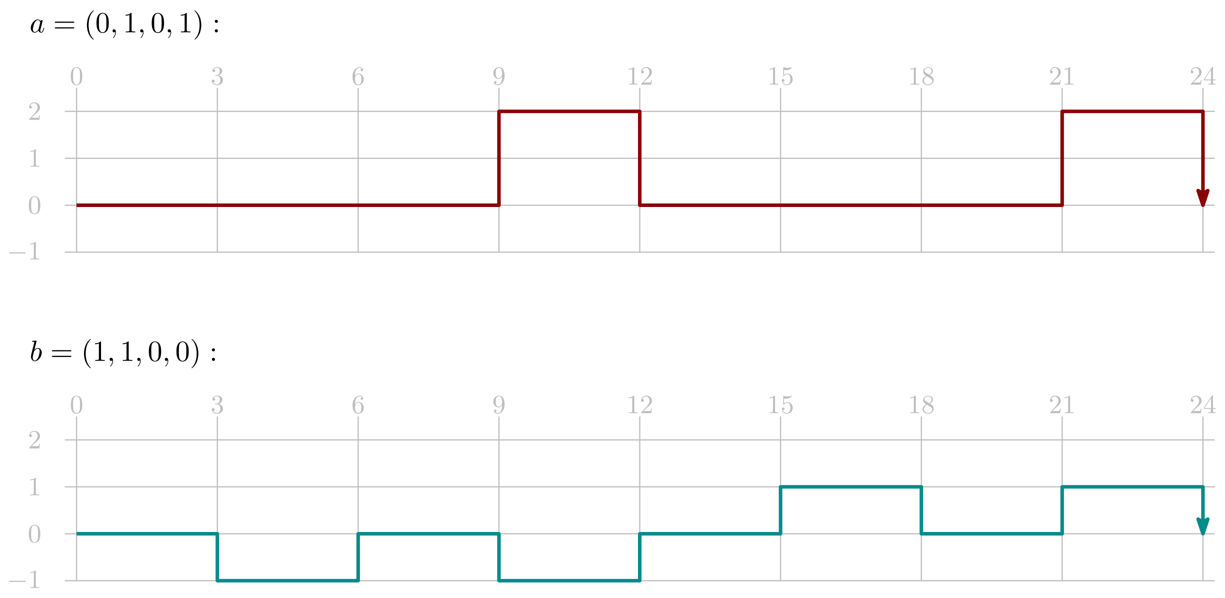

Figure 6: Visualization of the $3 - \varepsilon$ lower bound in 2D.

corresponds to queries with line segments, see [dBIG13, GvRSW21]. It also remains a fundamental problem to show fine-grained lower bounds for approximation factor larger than 3 for a metric problem, which seems to require fundamentally different techniques, cf. [Rub18].

As for the continuous Fréchet distance, our new upper and lower bounds show that the case of one-dimensional curves provides a kaleidoscopic view into the computational complexity and the underlying challenges posed by the general problem for polygonal curves in $\mathbb{R}^d$. The obvious way forward in this line of research is to show upper and lower bounds for dimension 2 and higher. Some of our ideas might translate directly, such as the idea to generate candidate curves at query time in order to achieve a tradeoff between preprocessing and query time. While our lower bounds also hold in higher dimension, it is conceivable that higher lower bounds can be shown already in the plane. In fact, we already initiate this line of work by showing an equally high lower bound for $(3-\varepsilon)$-ANN in the plane as we have for $(2-\varepsilon)$-ANN for one-dimensional curves. This lower bound already hints at techniques that can potentially achieve a matching upper bound. We leave this as an open problem. Our notions of straightenings and signatures, which capture the approximate shape of one-dimensional curves in a best-possible way, currently do not exist in dimension 2 or higher. Extending these notions to the plane by itself would be very interesting.

References

[AD18] Peyman Afshani and Anne Driemel. On the complexity of range searching among curves. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 898–917, 2018.

[AG95] Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geom. Appl.*, 5:75–91, 1995.

[ARW17] Amir Abboud, Aviad Rubinstein, and R. Ryan Williams. Distributed PCP theorems for hardness of approximation in P. In *58th IEEE Annual Symposium on Foundations* *of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 25–36, 2017.

[AW14] Amir Abboud and Virginia Vassilevska Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 434–443, 2014.

[AW15] Josh Alman and Ryan Williams. Probabilistic polynomials and hamming nearest neighbors. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 136–150, 2015.

[BB17] Julian Baldus and Karl Bringmann. A fast implementation of near neighbors queries for Fréchet distance (GIS Cup). In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'17, pages 99:1–99:4, 2017.

[BBW08] Kevin Buchin, Maike Buchin, and Carola Wenk. Computing the Fréchet distance between simple polygons. *Computational Geometry*, 41(1-2):2–20, 2008.

[BDvDM17] Kevin Buchin, Yago Diez, Tom van Diggelen, and Wouter Meulemans. Efficient trajectory queries under the Fréchet distance (GIS Cup). In *Proc. 25th Intern. Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, pages 101:1–101:4, 2017.

[BK18] Karl Bringmann and Marvin Künnemann. Multivariate fine-grained complexity of longest common subsequence. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1216–1235, 2018.

[BKN19] Karl Bringmann, Marvin Künnemann, and André Nusser. Walking the dog fast in practice: Algorithm engineering of the Fréchet distance. In Gill Barequet and Yusu Wang, editors, *35th International Symposium on Computational Geometry, SoCG 2019, June 18-21, 2019, Portland, Oregon, USA*, volume 129 of *LIPIcs*, pages 17:1–17:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[BM16] Karl Bringmann and Wolfgang Mulzer. Approximability of the discrete Fréchet distance. *J. Comput. Geom.*, 7(2):46–76, 2016.

[BOS19] Kevin Buchin, Tim Ophelders, and Bettina Speckmann. SETH says: Weak Fréchet distance is faster, but only if it is continuous and in one dimension. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2887–2901, 2019.

[Bri14] Karl Bringmann. Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 661–670, 2014.

[CGL$^+$19] Lijie Chen, Shafi Goldwasser, Kaifeng Lyu, Guy N. Rothblum, and Aviad Rubinstein. Fine-grained complexity meets IP = PSPACE. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1–20. SIAM, 2019.

[CW19] Lijie Chen and Ryan Williams. An equivalence class for orthogonal vectors. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 21–40. SIAM, 2019.

[dBGM17] Mark de Berg, Joachim Gudmundsson, and Ali D. Mehrabi. A dynamic data structure for approximate proximity queries in trajectory data. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2017, Redondo Beach, CA, USA, November 7-10, 2017*, pages 48:1–48:4, 2017.

[dBIG13] Mark de Berg, Atlas F. Cook IV, and Joachim Gudmundsson. Fast Fréchet queries. *Comput. Geom.*, 46(6):747–755, 2013.

[dBMO17] Mark de Berg, Ali D. Mehrabi, and Tim Ophelders. Data structures for Fréchet queries in trajectory data. In *Proceedings of the 29th Canadian Conference on Computational Geometry, CCCG 2017, July 26-28, 2017, Carleton University, Ottawa, Ontario, Canada*, pages 214–219, 2017.

[DHP13] Anne Driemel and Sariel Har-Peled. Jaywalking your dog: computing the Fréchet distance with shortcuts. *SIAM Journal on Computing*, 42(5):1830–1866, 2013.

[DKS16] Anne Driemel, Amer Krivosija, and Christian Sohler. Clustering time series under the Fréchet distance. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 766–785, 2016.

[DP20] Anne Driemel and Ioannis Psarros. (2+ϵ)-ANN for time series under the Fréchet distance. *CoRR*, abs/2008.09406, 2020.

[DP21] Anne Driemel and Ioannis Psarros. ANN for time series under the fréchet distance. In Anna Lubiw and Mohammad R. Salavatipour, editors, *Algorithms and Data Structures - 17th International Symposium, WADS 2021, Virtual Event, August 9-11, 2021, Proceedings*, volume 12808 of *Lecture Notes in Computer Science*, pages 315–328. Springer, 2021.

[DS17] Anne Driemel and Francesco Silvestri. Locality-sensitive hashing of curves. In *33rd International Symposium on Computational Geometry, SoCG 2017, July 4-7, 2017, Brisbane, Australia*, pages 37:1–37:16, 2017.

[DV17] Fabian Dütsch and Jan Vahrenhold. A filter-and-refinement- algorithm for range queries based on the Fréchet distance (GIS Cup). In *Proc. 25th Int. Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, pages 100:1–100:4, 2017.

[EP20] Ioannis Z. Emiris and Ioannis Psarros. Products of euclidean metrics, applied to proximity problems among curves: Unified treatment of discrete Fréchet and dynamic time warping distances. *ACM Trans. Spatial Algorithms Syst.*, 6(4):27:1–27:20, 2020.

[FFK20] Arnold Filtser, Omrit Filtser, and Matthew J. Katz. Approximate nearest neighbor for curves - simple, efficient, and deterministic. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual*

Conference), volume 168 of *LIPIcs*, pages 48:1–48:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[FKS84] Michael L. Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with $O(1)$ worst case access time. *J. ACM*, 31(3):538–544, 1984.

[GHPS20] Joachim Gudmundsson, Michael Horton, John Pfeifer, and Martin P. Seybold. A practical index structure supporting Fréchet proximity queries among trajectories, 2020.

[GvRSW21] Joachim Gudmundsson, André van Renssen, Zeinab Saeidi, and Sampson Wong. Translation invariant fréchet distance queries. *CoRR*, abs/2102.05844, 2021.

[HP11] Sariel Har-Peled. *Geometric Approximation Algorithms.* American Mathematical Society, Boston, MA, USA, 2011.

[Ind02] Piotr Indyk. Approximate nearest neighbor algorithms for Fréchet distance via product metrics. In *Proceedings of the 18th Annual Symposium on Computational Geometry, Barcelona, Spain, June 5-7, 2002*, pages 102–106, 2002.

[IP01] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

[Mil94] Peter Bro Miltersen. Lower bounds for union-split-find related problems on random access machines. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '94, page 625–634, New York, NY, USA, 1994. Association for Computing Machinery.

[Mir20] Majid Mirzanezhad. On the approximate nearest neighbor queries among curves under the Fréchet distance. *CoRR*, abs/2004.08444, 2020.

[Rub18] Aviad Rubinstein. Hardness of approximate nearest neighbor search. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1260–1268, 2018.

[SBL20] Roniel S. De Sousa, Azzedine Boukerche, and Antonio A. F. Loureiro. Vehicle trajectory similarity: Models, methods, and applications. *ACM Comput. Surv.*, 53(5), September 2020.

[SLZ$^+$20] Han Su, Shuncheng Liu, Bolong Zheng, Xiaofang Zhou, and Kai Zheng. A survey of trajectory distance measures and performance evaluation. *The VLDB Journal*, 29(1):3–32, 2020.

[Wil05] Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2-3):357–365, 2005.