Fréchet Distance Under Translation: Conditional Hardness and an Algorithm via Offline Dynamic Grid Reachability

Karl Bringmann* Marvin Künnemann* André Nusser*

Abstract

The discrete Fréchet distance is a popular measure for comparing polygonal curves. An important variant is the discrete Fréchet distance under translation, which enables detection of similar movement patterns in different spatial domains. For polygonal curves of length n in the plane, the fastest known algorithm runs in time $\tilde{\mathcal{O}}(n^5)$ [Ben Avraham, Kaplan, Sharir '15]. This is achieved by constructing an arrangement of disks of size $\mathcal{O}(n^4)$, and then traversing its faces while updating reachability in a directed grid graph of size $N = \mathcal{O}(n^2)$, which can be done in time $\tilde{\mathcal{O}}(\sqrt{N})$ per update [Diks, Sankowski '07]. The contribution of this paper is two-fold.

First, although it is an open problem to solve dynamic reachability in directed grid graphs faster than $\tilde{\mathcal{O}}(\sqrt{N})$, we improve this part of the algorithm: We observe that an offline variant of dynamic s-t-reachability in directed grid graphs suffices, and we solve this variant in amortized time $\tilde{\mathcal{O}}(N^{1/3})$ per update, resulting in an improved running time of $\tilde{\mathcal{O}}(n^{4.66...})$ for the discrete Fréchet distance under translation. Second, we provide evidence that constructing the arrangement of size $\mathcal{O}(n^4)$ is necessary in the worst case, by proving a conditional lower bound of $n^{4-o(1)}$ on the running time for the discrete Fréchet distance under translation, assuming the Strong Exponential Time Hypothesis.

1 Introduction

Fréchet distance. Modern tracking devices yield an abundance of movement data, e.g., in the form of GPS trajectories. This data is usually given as a sequence of points in $\mathbb{R}^d$ for some small dimension d like 2 or 3. By interpolating linearly between consecutive points, we obtain a corresponding polygonal curve. One of the most fundamental tasks on such objects is to measure similarity between two curves π, σ. A popular approach is to measure their distance using the Fréchet distance, which has two important variants: The classic *continuous Fréchet distance* is the minimal length of a leash connecting a dog and its owner as they continuously walk along the interpolated curves π and σ, respectively, from the startpoints to the endpoints without backtracking. In the *discrete Fréchet distance*, at any time step the dog and its owner must be at vertices of their curves and may jump to the next vertex. This discrete version is well motivated when we think of the inputs as sequences of points rather than polygonal curves, i.e., if the interpolated line segments between input points have no meaning in the underlying application. In comparison to other similarity measures such as the Hausdorff distance, the Fréchet distance considers the ordering of the vertices along the curves, thus reflecting an intuitive property of curve similarity.

The time complexity of the Fréchet distance is well understood. For the continuous Fréchet distance, Alt and Godau designed an $\mathcal{O}(n^2 \log n)$-time algorithm for polygonal curves π, σ consisting of n vertices [AG95]. Buchin et al. [BBMM14] improved on this result by giving an algorithm that runs in time $\mathcal{O}(n^2 \sqrt{\log n}(\log\log n)^{3/2})$ on the Real RAM and $\mathcal{O}(n^2(\log\log n)^2)$ on the Word RAM. The first algorithm for the discrete Fréchet distance ran in time $\mathcal{O}(n^2)$ [EM94], which was later improved to $\mathcal{O}\big(n^2 \frac{\log\log n}{\log n}\big)$ [AAKS13]. On the hardness side, conditional on the Strong Exponential Time Hypothesis, Bringmann [Bri14] ruled out $\mathcal{O}(n^{2-\varepsilon})$-time algorithms for any $\varepsilon > 0$, for both variants of the Fréchet distance. Recently, Abboud and Bringmann [AB18] showed that any $\mathcal{O}(n^2/\log^{17+\varepsilon} n)$-time algorithm for the discrete Fréchet distance would prove novel circuit lower bounds.

Many extensions and variants of the Fréchet distance have been studied, e.g., generalizing from curves to other types of objects, replacing the ground space $\mathbb{R}^d$ by more complex spaces, and many more (see, e.g., [Ind02, BBW09, AB10, CdVE+10, CW10, MSSZ11, DH13, AFK+15]). Applications of the Fréchet distance range from moving objects analysis (see, e.g., [BBG+11]) through map-matching tracking data (see, e.g., [BPSW05]) to signature verification (see, e.g., [MP99]).

Fréchet distance under translation. For some applications, it is useful to change the definition of the Fréchet distance slightly. In particular, several applications on curves evolve around the theme of *detecting movement patterns*. For instance, given GPS trajectories of an animal, we might want to detect different running styles by chopping the trajectories

*Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany, {kbringma,marvin,anusser}@mpi-inf.mpg.de

into smaller pieces and clustering these pieces according to some distance measure. For such applications, it is inconvenient that the Fréchet distance is not invariant under translation.[1] Indeed, the same running style performed at different spatial locations would result in a large Fréchet distance. In order to overcome this issue, the *Fréchet distance under translation* between curves π, σ is defined as the minimal Fréchet distance between π and any translation of σ, i.e., we minimize over all possible translations of σ. Clearly, this yields a translation-invariant distance measure, and thus enables the above application.

The *continuous* Fréchet distance under translation was independently introduced by Efrat et al. [EIV01] and Alt et al. [AKW01], who designed algorithms in the plane with running time $\tilde{\mathcal{O}}(n^{10})$ and $\tilde{\mathcal{O}}(n^8)$, respectively[2]. Both groups of researchers also presented approximation algorithms, e.g., a $(1+\varepsilon)$-approximation running in time $\mathcal{O}(n^2/\varepsilon^2)$ in the plane [AKW01]. This line of work was extended to three dimensions with a running time of $\tilde{\mathcal{O}}(n^{11})$ [Wen03].

The *discrete* Fréchet distance under translation was first studied by Jiang et al. [JXZ08] who designed an $\tilde{\mathcal{O}}(n^6)$-time algorithm in the plane. Mosig et al. [MC05] presented an approximation algorithm that computes the discrete Fréchet distance under translation, rotation, and scaling in the plane, up to a factor close to 2, and runs in time $\mathcal{O}(n^4)$. The best known exact algorithm for the discrete Fréchet distance under translation in the plane is due to Ben Avraham et al. [AKS15]. It is an improvement of the algorithm by Jiang et al. [JXZ08] and runs in time $\tilde{\mathcal{O}}(n^5)$.

Our contribution. In this paper, we further study the time complexity of the discrete Fréchet distance under translation. First, we improve the running time from $\tilde{\mathcal{O}}(n^5)$ to $\tilde{\mathcal{O}}(n^{4.66\ldots})$. This is achieved by designing an improved algorithm for a subroutine of the previously best algorithm, namely offline dynamic s-t-reachability in directed grid graphs, see Section 1.1 below for a more detailed overview.

Theorem 1.1. *The discrete Fréchet distance under translation on curves of length n in the plane can be computed in time $\tilde{\mathcal{O}}(n^{14/3}) = \tilde{\mathcal{O}}(n^{4.66..})$.*

Our second main result is a lower bound of $n^{4-o(1)}$, conditional on the standard Strong Exponential Time Hypothesis. The Strong Exponential Time Hypothesis essentially asserts that Satisfiability requires time $2^{n-o(n)}$; see Section 2 for a definition. This (conditionally) separates the discrete Fréchet distance under translation from the classic Fréchet distance, which can be computed in time $\tilde{\mathcal{O}}(n^2)$. Moreover, the first step of all known algorithms for the discrete Fréchet distance under translation is to construct an arrangement of disks of size $\mathcal{O}(n^4)$. Our conditional lower bound shows that this is essentially unavoidable.

Theorem 1.2. *The discrete Fréchet distance under translation of curves of length n in the plane requires time $n^{4-o(1)}$, unless the Strong Exponential Time Hypothesis fails.*

We leave closing the gap between $\tilde{\mathcal{O}}(n^{4.66..})$ and $n^{4-o(1)}$ as an open problem.

1.1 Technical Overview

Previous algorithms for the discrete Fréchet distance under translation. Let us sketch the algorithms by Jiang et al. [JXZ08] and Ben Avraham et al. [AKS15]. Given sequences $\pi = (\pi_1, \ldots, \pi_n)$ and $\sigma = (\sigma_1, \ldots, \sigma_n)$ in $\mathbb{R}^2$ and a number $\delta \geq 0$, we want to decide whether the discrete Fréchet distance under translation of π and σ is at most δ. From this decision procedure one can obtain an algorithm to compute the actual distance via standard techniques (i.e., parametric search).

The translations τ for which the distance of π_i and $\sigma_j + \tau$ is at most δ form a disk in $\mathbb{R}^2$. Over all pairs (π_i, σ_j) this yields $\mathcal{O}(n^2)$ disks, all of them having radius δ. We construct their arrangement $\mathcal{A}$, which is guaranteed to have $\mathcal{O}(n^4)$ faces. Within each face of $\mathcal{A}$, any two translations are equivalent, in the sense that they leave the same pairs (π_i, σ_j) in distance at most δ. Thus, whether the discrete Fréchet distance is at most δ is constant in each face. Hence, it suffices to compute the discrete Fréchet distance between π and σ translated by τ over $\mathcal{O}(n^4)$ choices for τ, one for each face of $\mathcal{A}$. Since the discrete Fréchet distance can be computed in time $\mathcal{O}(n^2)$, this yields an $\mathcal{O}(n^6)$-time algorithm, which is essentially the algorithm by Jiang et al. [JXZ08].

To improve this further, first we consider the Fréchet distance more closely. Denote by M the $n \times n$ matrix with $M_{i,j} = 1$ if the points π_i, σ_j are in distance at most δ, and $M_{i,j} = 0$ otherwise (M is called the "free-space diagram"). It is well-known that the discrete Fréchet distance of π, σ is at most δ if and only if there exists a monotone path from the lower left to the upper right corner of M using only 1-entries. Equivalently, consider a *directed grid graph* G_M on $n \times n$ vertices, where each

[1] In this context one could even ask for a version of the Fréchet distance that is translation- and rotation-invariant, but we focus on the former in this paper.

[2] By $\tilde{\mathcal{O}}(\cdot)$ we hide polylogarithmic factors in n.

node (i,j) has directed edges to $(i+1,j),(i,j+1)$, and $(i+1,j+1)$, and the nodes (i,j) of G_M with $M_{i,j}=0$ are "deactivated" (i.e., removed). Then the discrete Fréchet distance of π,σ is at most δ if and only if node (n,n) is reachable from node $(1,1)$ in G_M. See Figure 1 on page 7 for an example of a pair of curves and its corresponding free-space diagram M and directed grid graph G_M.

With this preparation, we start from a sequence of $\mathcal{O}(n^4)$ faces $f_1,\ldots,f_L$ of the arrangement $\mathcal{A}$ such that (1) each face of $\mathcal{A}$ is visited at least once and (2) f_ℓ and $f_{\ell+1}$ are neighboring in $\mathcal{A}$ for all ℓ. Such a sequence can be constructed by building a spanning tree of the dual graph of the arrangement, doubling any edge of the spanning tree, and then computing an Euler tour in the resulting graph. Since consecutive faces in this sequence are neighbors, only one pair (π_i,σ_j) changes its distance, i.e., either π_i,σ_j are in distance at most δ in f_ℓ and in distance larger than δ in $f_{\ell+1}$, or vice versa. This corresponds to one activation or deactivation of a node in G_M. After this update, we want to again check whether node (n,n) is reachable from node $(1,1)$ in G_M. That is, using a dynamic algorithm for s-t-reachability in directed grid graphs, we can maintain whether the Fréchet distance is at most δ. The best-known solution to dynamic reachability in directed $n\times n$ grids runs in time $\tilde{\mathcal{O}}(n)$ [DS07].[3] Over all $\mathcal{O}(n^4)$ faces, this yields time $\tilde{\mathcal{O}}(n^5)$ for the discrete Fréchet distance under translation in the plane [AKS15].

Intuition. There are two parts to the above algorithm: (1) Constructing the arrangement $\mathcal{A}$ and iterating over its faces, and (2) maintaining reachability in the grid graph G_M. Both parts could potentially be improved.

The natural first attempt seems to attack the arrangement enumeration (1). The size of the arrangement is $\mathcal{O}(n^4)$, and for no other computational problem it is known – to the best of our knowledge – that any optimal algorithm must construct such a large arrangement, so this part seems intuitively wasteful. Surprisingly, our conditional lower bound of Theorem 1.2 shows that constructing the arrangement is essentially unavoidable.

The remaining part (2) at first sight seems much less likely to be improvable, since it is a well-known open problem to find a faster dynamic algorithm for reachability in directed grid graphs. Nevertheless, we managed to improve the running time of this part of the algorithm, as sketched in what follows.

[3]This algorithm even works more generally for dynamic reachability in directed planar graphs.

Our algorithm. We observe that we do not need the full power of dynamic reachability, since we can precompute all $\mathcal{O}(n^4)$ updates. This leaves us with the following problem.

Offline Dynamic Grid Reachability: We start from the directed $n\times n$-grid graph G in which all nodes are deactivated. We are given a sequence of updates $u_1,\ldots,u_U$, where each u_ℓ is of the form "activate node (i,j)" or "deactivate node (i,j)". The goal is to compute for each $1\le\ell\le U$ whether node $(1,1)$ can reach node (n,n) in G after performing the updates $u_1,\ldots,u_\ell$.

Our main algorithmic contribution is an algorithm for Offline Dynamic Grid Reachability in amortized time $\tilde{\mathcal{O}}(n^{2/3})$ per update. This is faster than the time $\tilde{\mathcal{O}}(n)$ obtained by using a dynamic algorithm for reachability in directed planar graphs [DS07].

THEOREM 1.3. *Offline Dynamic Grid Reachability can be solved in time $\tilde{\mathcal{O}}(n^2+U\cdot n^{2/3})$.*

We give a short overview of this algorithm. Start with the block $[n]\times[n]$ corresponding to the matrix M. Repeatedly split every block horizontally in the middle, and then split every block vertically in the middle, until we end up with constant-size blocks. We call all the blocks considered during this process (not just the constant-size blocks!) the "canonical" blocks, see Figure 2 on page 9. Ben Avraham et al. [AKS15] showed that one can store for each canonical block of sidelength s reachability information for each pair of boundary nodes, succinctly represented using only $\tilde{\mathcal{O}}(s)$ bits of space, and efficiently computable in time $\tilde{\mathcal{O}}(s)$ from the information of the two canonical child-blocks. In particular, over all blocks this information can be maintained in time $\tilde{\mathcal{O}}(n)$ per update u_i.

We extend their algorithm to show that given k updates $u_1,\ldots,u_k$, we can directly compute the reachability information after all k updates in time $\tilde{\mathcal{O}}(n\sqrt{k})$. To understand this better, observe that each update "touches" roughly $2\cdot\log n$ blocks – all those that contain the node which is activated or deactivated. Our approach now uses that among the canonical blocks containing an update, the large blocks must be shared by many updates. More concretely, instead of recomputing the reachability information of the large blocks at the top of the hierarchy k times, we perform those updates jointly and thus avoid the runtime of k explicit updates of large blocks. This result then allows us to split the updates $u_1,\ldots,u_U$ into *chunks* of size $k=\tilde{\mathcal{O}}(n^{2/3})$ and compute the above reachability information for all startpoints of chunks in total time $\tilde{\mathcal{O}}(\frac{U}{k}\cdot n\sqrt{k}) = \tilde{\mathcal{O}}(U\cdot n^{2/3})$.

Now fix a chunk $C = u_\ell, \ldots, u_{\ell+k-1}$. Denote by $\mathcal{T}$ ("terminals") the entries that get activated or deactivated during this chunk C, and also add $(1,1)$ and (n,n) to the set of terminals. We first deactivate all terminals, obtaining a matrix M and a corresponding grid graph G_M. The basic idea now is to determine for each pair of terminals $t, t' \in \mathcal{T}$ whether t' is reachable from t in G_M.

Assuming we have this reachability information among terminals, we now show that this yields a speedup for Offline Dynamic Grid Reachability. We describe a simplified algorithm here, which will be improved later in the paper. Build a graph H with vertex set $\mathcal{T}$, containing a directed edge (t, t') if and only if t' is reachable from t in G_M. Activate or deactivate the nodes of H according to the state at the beginning of the chunk C. Then iteratively perform each update of the chunk C by activating or deactivating the corresponding node of H, and check whether (n,n) is reachable from $(1,1)$ in H. Disregarding the time it takes to construct H, this reachability check can be performed in time $\mathcal{O}(k^2)$ per update, or total time $\mathcal{O}(k^3)$ over the chunk C since H has $\mathcal{O}(k)$ nodes and thus $\mathcal{O}(k^2)$ edges. This solves the Offline Dynamic Grid Reachability problem in time $\tilde{\mathcal{O}}(n\sqrt{k} + k^3)$ per chunk, or $\tilde{\mathcal{O}}(\frac{U}{k}(n\sqrt{k} + k^3)) = \tilde{\mathcal{O}}(U(n/\sqrt{k} + k^2))$ in total. Setting $k = n^{2/5}$ optimizes this time to $\tilde{\mathcal{O}}(Un^{4/5})$, again ignoring the preprocessing time. This is a simplified variant of our algorithm. We will later show how to improve the time per reachability check from $\mathcal{O}(k^2)$ to $\tilde{\mathcal{O}}(k)$, by working directly on the graph G_M instead of constructing the graph H. This yields total time $\tilde{\mathcal{O}}(\frac{U}{k}(n\sqrt{k}+k^2)) = \tilde{\mathcal{O}}(U(n/\sqrt{k}+k))$, which is $\tilde{\mathcal{O}}(Un^{2/3})$ for $k = \tilde{\mathcal{O}}(n^{2/3})$. Note that the above analysis ignores all preprocessing terms as they are dominated. These details are given in the subsequent sections.

It remains to describe how to determine reachability information among terminals. To this end, we designed a surprisingly succinct representation of reachability from terminals to block boundaries. Consider a canonical block B and let $\mathcal{T}_B$ be the terminals in B. For each terminal $t \in \mathcal{T}_B$ let $A(t)$ be the lowest/rightmost point on the right/upper boundary of B that is reachable from t, and similarly let $Z(t)$ be the highest/leftmost reachable point, see Figure 4 on page 11. We label any terminal $t = (x, y)$ by $L(t) := x + y$, i.e., the anti-diagonal that t is contained in. For any right/upper boundary point q of B, let $\ell(q)$ be the minimal label of any terminal in $\mathcal{T}_B$ from which q is reachable, see Figure 3 on page 9. We prove the following succinct representation of reachability (see Corollary 4.1).

For any right/upper boundary point q of B and any terminal $t \in \mathcal{T}_B$, q is reachable from t if and only if $q \in [A(t), Z(t)]$ and $\ell(q) \le L(t)$.

Here, $q \in [A(t), Z(t)]$ is to be understood as "q lies between $A(t)$ and $Z(t)$ along the boundary of B", which can be expressed using a constant number of inequalities. The "only if" part is immediate, since t can only reach boundary vertices in $[A(t), Z(t)]$, and $\ell(q)$ is the minimal label of any terminal reaching q; the "if" part is surprising.

Assume we can maintain the information $A(t), Z(t), \ell(q)$. Then using this characterization we can determine all terminals reaching a boundary point q by a single call to *orthogonal range searching*, since we can express the characterization using a constant number of inequalities. A complex extension of this trick allows us to determine reachability among terminals (indeed, this technical overview is missing many details of Section 4). This yields our algorithm, see Sections 3 and 4 for details.

Conditional lower bound. Our reduction starts from the k-OV problem, which asks for k vectors from k given sets such that in no dimension all vectors are 1. More formally:

k-Orthogonal Vectors (k-OV): Given sets $V_1, \ldots, V_k$ of N vectors in $\{0,1\}^D$, are there $v_1 \in V_1, \ldots, v_k \in V_k$ such that for any $j \in [D]$ there exists an $i \in [k]$ with $v_i[j] = 0$?

A naive algorithm solves k-OV in time $\mathcal{O}(N^k D)$. It is well-known that the Strong Exponential Time Hypothesis implies that k-OV has no $\mathcal{O}(N^{k-\varepsilon}\mathrm{poly}(D))$-time algorithm for all $\varepsilon > 0$ and $k \ge 2$ [Wil05].

In our reduction we set $k = 4$. An overview of our construction can be found in Figure 7 on page 16. We consider *canonical translations* of the form $\tau = (\varepsilon \cdot h_1, \varepsilon \cdot h_2) \in \mathbb{R}^2$ with $h_1, h_2 \in \{0, \ldots, N^2 - 1\}$. By a simple gadget, we ensure that any translation resulting in a Fréchet distance of at most 1 must be close to a canonical translation. For simplicity, here we restrict our attention to exactly the canonical translations. Note that there are N^4 canonical translations, and thus they are in one-to-one correspondence to choices of vectors $(v_1, \ldots, v_4) \in V_1 \times \ldots \times V_4$. In other words, the outermost existential quantifier in the definition of 4-OV corresponds to the existential quantifier over the translation τ in the Fréchet distance under translation.

The next part in the definition of 4-OV is the universal quantifier over all dimensions $j \in [D]$. For this, our constructed curves π, σ are split into $\pi = \pi^{(1)} \ldots \pi^{(D)}, \sigma = \sigma^{(1)} \ldots \sigma^{(D)}$ such that $\pi^{(i)}, \sigma^{(j)}$ are very far for $i \ne j$. This ensures that the Fréchet distance of π, σ is the maximum over all Fréchet distances of $\pi^{(i)}, \sigma^{(i)}$, and thus simulates a universal quantifier.

The next part is an existential quantifier over $i \in [k]$. Here we need an OR-gadget for the Fréchet distance. Such a construction in principle exists in previous work [Bri14, AB18], however, no previous construction would work with translations, in the sense that a translation in y-direction could only decrease the Fréchet distance. By constructing a more complex OR-gadget, we avoid this monotonicity, see Figure 8 on page 18.

Finally, we need to implement a check whether the translation τ corresponds to a particular choice of vectors. We exemplify this with the first dimension of the translation, which we call τ_1, explaining how it corresponds to choosing (v_1, v_2). Let $\mathrm{ind}(v_1), \mathrm{ind}(v_2) \in \{0, \ldots, N-1\}$ be the indices of these vectors in their sets V_1, V_2, respectively. We want to test whether $\tau_1 = \varepsilon \cdot (\mathrm{ind}(v_1) + \mathrm{ind}(v_2) \cdot N)$. We split this equality into two inequalities. For the inequality $\tau_1 \geq \varepsilon \cdot (\mathrm{ind}(v_1) + \mathrm{ind}(v_2) \cdot N)$, in one curve we place a point at $\pi_1 = (1 + \varepsilon \cdot \mathrm{ind}(v_1), -1 - \eta)$, and in the other we place a point at $\sigma_1 = (-1 - \varepsilon \cdot \mathrm{ind}(v_2) \cdot N, -1 - \eta)$, for some $\eta > 0$ which we specify later in this work. Then the distance of π_1 to the translated σ_1 is essentially their difference in x-coordinates, which is $(1 + \varepsilon \cdot \mathrm{ind}(v_1)) - (-1 - \varepsilon \cdot \mathrm{ind}(v_2) \cdot N + \tau_1) = 2 + \varepsilon \cdot (\mathrm{ind}(v_1) + \mathrm{ind}(v_2) \cdot N) - \tau_1$. This is at most 2 if and only if the inequality for τ_1 holds. We handle the opposite inequality similarly, and we concatenate the constructed points for both inequalities in order to test equality.

In total, our construction yields curves π, σ such that their discrete Fréchet distance under translation is at most 1 if and only if $V_1, \ldots, V_4$ contain orthogonal vectors. The curves π, σ consist of $n = \mathcal{O}(D \cdot N)$ vertices. Hence, an algorithm for the discrete Fréchet distance under translation in time $\mathcal{O}(n^{4-\varepsilon})$ would yield an algorithm for 4-OV in time $\mathcal{O}(N^{4-\varepsilon}\mathrm{poly}(D))$, and thus violate the Strong Exponential Time Hypothesis. See Section 5 for details.

1.2 Further related work

On directed planar/grid graphs. In this paper we improve offline dynamic s-t-reachability in directed grid graphs. The previously best algorithm for this problem came from a more general solution to dynamic reachability in directed planar graphs. For this problem, a solution with $\tilde{\mathcal{O}}(N^{2/3})$ update time was given by Subramanian [Sub93], which was later improved to update time $\tilde{\mathcal{O}}(\sqrt{N})$ by Diks and Sankowski [DS07]. In particular, our work yields additional motivation to study offline variants of classic dynamic graph problems.

Related work on dynamic directed planar or grid graphs includes, e.g., shortest path computation [KS98, ACG12, INSW11], reachability in the decremental setting [IKLS17], or computing the transitive closure [DS07]. Recently, the first conditional lower bounds for dynamic problems on planar graphs were shown by Abboud and Dahlgaard [AD16], however, they did not cover dynamic reachability in directed planar graphs.

2 Preliminaries

We let $[n]$ denote the set $\{1, \ldots, n\}$. Furthermore, for convenience, we use as convention that $\min \emptyset = \infty$ and $\max \emptyset = -\infty$.

Curves, Traversals, Fréchet distances, and more. A polygonal curve π of length n over $\mathbb{R}^d$ is a sequence of points $\pi_1, \ldots, \pi_n \in \mathbb{R}^d$. Throughout the paper, we only consider polygonal curves in the Euclidean plane, i.e., $d = 2$. Given any translation vector $\tau \in \mathbb{R}^2$, we denote by $\pi + \tau$ the polygonal curve $\pi' = (\pi'_1, \ldots, \pi'_n)$ given by $\pi'_i = \pi_i + \tau$.

We now define two types of concatenations: a concatenation of curves and a concatenation of traversals. Let $\pi = (\pi_1, \ldots, \pi_n), \sigma = (\sigma_1, \ldots, \sigma_n)$ be polygonal curves of lengths n. We define the concatenation of π and σ as $\pi \circ \sigma \coloneqq (\pi_1, \ldots, \pi_n, \sigma_1, \ldots, \sigma_n)$. The resulting curve has length $2n$. Now defining the concatenation of traversals, we call any pair $(i, j) \in [n] \times [n]$ a *position*. A *traversal* T is a sequence $t_1, \ldots, t_\ell$ of positions, where $t_k = (i, j)$ implies that t_{k+1} is either $(i+1, j)$ (that is, we advance one step in π while staying in σ_j), $(i, j+1)$ (we advance in σ while staying in π_i), or $(i+1, j+1)$ (we advance in both curves simultaneously). We call $T = (t_1, \ldots, t_\ell)$ a traversal of π, σ, if $t_1 = (1, 1)$ and $t_\ell = (n, n)$. Given two traversals $T = (t_1, \ldots, t_\ell)$ and $T' = (t'_1, \ldots, t'_{\ell'})$ with $t_\ell = t'_1$, we define the concatenated traversal as $T \circ T' \coloneqq (t_1, \ldots, t_\ell = t'_1, t'_2, \ldots, t'_{\ell'})$. Note that we obtain a traversals from t_1 to $t'_{\ell'}$.

The *discrete Fréchet distance* is defined as

$$\delta_F(\pi, \sigma) \coloneqq \min_{T=((i_1,j_1),\ldots,(i_\ell,j_\ell))} \max_{1 \leq k \leq \ell} \|\pi_{i_k} - \sigma_{j_k}\|,$$

where T ranges over all traversals of π, σ and $\|\cdot\|$ denotes the Euclidean distance in $\mathbb{R}^2$.

We obtain a well-known equivalent definition as follows: Fix some distance $\delta \geq 0$. We call a position (i, j) *free* if $\|\pi_i - \sigma_j\| \leq \delta$. We say that a traversal $T = (t_1, \ldots, t_\ell)$ of π, σ is a *valid traversal* for δ if $t_1, \ldots, t_\ell$ are all free positions. The discrete Fréchet distance of π, σ is then the smallest δ such that there is a valid traversal of π, σ for δ.

Analogously, consider the $n \times n$ matrix M with

$M_{i,j} = 1$ if (i,j) is free, and $M_{i,j} = 0$ otherwise. We call any traversal $T = (t_1, \dots, t_\ell)$ a *monotone path from* t_1 *to* t_ℓ. If all positions (i,j) visited by T satisfy $M_{i,j} = 1$, we call T a *monotone 1-path* from t_1 to t_ℓ in M. As yet another formulation, consider the $n \times n$ grid graph G_M where vertex (i,j) has directed edges to all of $(i, j+1), (i+1, j)$, and $(i+1, j+1)$ (in case they exist). Deactivate (i.e., remove) all non-free vertices (i,j) from G_M. Then a monotone 1-path in M corresponds to a (directed) path in G_M. Hence, $\delta_F(\pi, \sigma) \le \delta$ is equivalent to the existence of a valid traversal of π, σ for δ, which in turn is equivalent to the existence of a monotone 1-path from $(1,1)$ to (n,n) in the matrix M, and to vertex (n,n) being reachable from $(1,1)$ in G_M.

Finally, we define the discrete Fréchet distance under translation as $\min_{\tau \in \mathbb{R}^2} \delta_F(\pi, \sigma + \tau)$, i.e., the smallest discrete Fréchet distance of π to any translation of σ.

Hardness assumptions. The *Strong Exponential Time Hypothesis (SETH)* was introduced by Impagliazzo and Paturi [IP01] and essentially postulates that there is no exponential-time improvement over exhaustive search for the Satisfiability Problem. More formally, the Strong Exponential Time Hypothesis states that for any $\epsilon > 0$ there exists $k \ge 3$ such that k-SAT has no $\mathcal{O}((2-\epsilon)^n)$-time algorithm.

In fact, our reductions even hold under a weaker assumption, specifically, the *k-OV Hypothesis*.[4] Recall the k-OV problem: Given sets $V_1, \dots, V_k$ of N vectors in $\{0,1\}^D$, the task is to determine whether there are $v_1 \in V_1, \dots, v_k \in V_k$ such that for all $j \in [D]$ there exists an $i \in [k]$ with $v_i[j] = 0$. The k-OV Hypothesis states that for any $k \ge 2$ and $\epsilon > 0$, there is no $\mathcal{O}(N^{k-\epsilon}\mathrm{poly}(D))$-time algorithm for k-OV. The well-known split-and-list technique due to Williams [Wil05] shows that SETH implies the k-OV Hypothesis. Thus, any conditional lower bound that holds under the k-OV hypothesis also holds under SETH.

Orthogonal range data structures. We will use a tool from geometric data structures, namely *(dynamic) orthogonal range data structures*. Let S be a set of key-value pairs $s = (k_s, v_s) \in \mathbb{Z}^d \times \mathbb{Z}$. An orthogonal range data structure on S allows to query the maximal value of any pair in S whose key lies in a given orthogonal range. Formally, we say $\mathcal{OR}$ *stores* v_s *under the key* k_s *for* $s \in S$ *for minimization queries*, if $\mathcal{OR}$ supports, for any $\ell_1, u_1, \ell_2, u_2, \dots, \ell_d, u_d \in \mathbb{Z} \cup \{-\infty, \infty\}$ and $R \coloneqq [\ell_1, u_1] \times \cdots \times [\ell_d, u_d]$, queries of the form

$$\mathcal{OR}.\min(R): \text{ return } \min\{v_s \mid s \in S, k_s \in R\}.$$

We will also consider analogous maximization queries.

Classic results [GBT84, Cha88] show that for any set S of size n and $d = 2$, we can construct such a data structure $\mathcal{OR}$ in time and space $\mathcal{O}(n \log n)$, supporting minimization (or maximization) queries in time $\mathcal{O}(\log n)$.

At one point in the paper we will also use an orthogonal range searching data structure that allows (1) to *report* all values of pairs in S whose keys lie in a given orthogonal range, and (2) to *remove* a key-value pair from S. Formally, we say that $\mathcal{OR}$ *stores* v_s *under the key* k_s *for* $s \in S$ *for decremental range reporting queries*, if $\mathcal{OR}$ supports, for any $\ell_1, u_1, \ell_2, u_2, \dots, \ell_d, u_d \in \mathbb{Z} \cup \{-\infty, \infty\}$ and $R \coloneqq [\ell_1, u_1] \times \cdots \times [\ell_d, u_d]$, queries of the form

$$\mathcal{OR}.\mathrm{report}(R): \text{ return } \{v_s \mid s \in S, k_s \in R\},$$

as well as deletions from the set S.

Mortensen [Mor06] and Chan and Tsakalidis [CT17] showed how to construct such a data structure $\mathcal{OR}$ for any set S of size n in time and space $\mathcal{O}(n \log^{d-1} n)$, deletion time $\mathcal{O}(\log^{d-1} n)$ and query time $\mathcal{O}(\log^{d-1} n + k)$, where k denotes the output size of the query. (These works obtain even stronger results, however, we use simplified bounds for ease of presentation.)

3 Algorithm: Reduction to Grid Reachability

In this section, we prove our algorithmic result by showing how a certain grid reachability data structure (that we give in Section 4) yields an $\tilde{\mathcal{O}}(n^{4+2/3})$-time algorithm for computing the discrete Fréchet distance under translation.

We start with a formal overview of the algorithm. First, we reduce the decision problem (i.e., is the discrete Fréchet distance under translation of π, σ at most δ?) to the problem of determining reachability in a dynamic grid graph, as shown by Ben Avraham et al. [AKS15]. However, noting that all updates and queries are known in advance, we observe that the following *offline* version suffices.

Problem 3.1. (Offline Dynamic Grid Reachability) *Let M be an $n \times n$ matrix over $\{0,1\}$. We call $u = (p, b)$ with $p \in [n] \times [n]$ and $b \in \{0,1\}$ an* update *and define $M[\![u]\!]$ as the matrix obtained by setting the bit at position p to b, i.e.,*

$$M[\![u]\!]_{i,j} = \begin{cases} b & \text{if } p = (i,j), \\ M_{i,j} & \text{otherwise.} \end{cases}$$

[4] In fact, we only need the corresponding hypothesis for 4-OV.

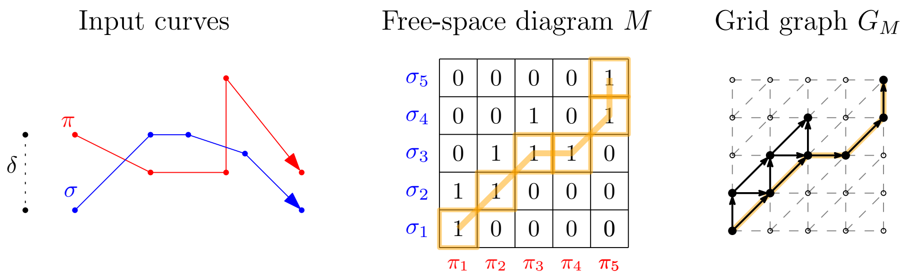

Figure 1: Two input curves π, σ and a distance δ, the corresponding free-space diagram M, and the grid graph G_M corresponding to M. A monotone traversal of M and G_M is marked in orange.

For any sequence of updates $u_1, \dots, u_k \in ([n] \times [n]) \times \{0,1\}$ with $k \geq 2$, we define $M[\![u_1, \dots, u_k]\!] := (M[\![u_1]\!])[\![u_2, \dots, u_k]\!]$.

The Offline Dynamic Grid Reachability *problem asks to determine, given M and any sequence of updates $u_1, \dots, u_U \in ([n] \times [n]) \times \{0,1\}$, whether there is a monotone 1-path from $(1,1)$ to (n,n) in $M[\![u_1, \dots, u_k]\!]$ for any $1 \leq k \leq U$.*

We state the following lemma, whose proof is deferred to the full version of this paper [BKN18] due to space constraints.

Lemma 3.1. *Assume there is an algorithm solving Offline Dynamic Grid Reachability in time $T(n,U)$. Then there is an algorithm that, given $\delta > 0$ and polygonal curves π, σ of length n over $\mathbb{R}^2$, determines whether $\delta_F(\pi, \sigma + \tau) \leq \delta$ for some $\tau \in \mathbb{R}^2$ in time $\mathcal{O}(T(n, n^4))$.*

Our speed-up is achieved by solving Offline Dynamic Grid Reachability in time $T(n,U) = \tilde{\mathcal{O}}(n^2 + Un^{2/3})$ (Ben Avraham et al. [AKS15] achieved $T(n,U) = \mathcal{O}(n^2 + Un)$). To this end, we devise a *grid reachability data structure*, which is our central technical contribution.

Lemma 3.2. (Grid reachability data structure) *Given an $n \times n$ matrix M over $\{0,1\}$ and a set of* terminals *$\mathcal{T} \subseteq [n] \times [n]$ of size $k > 0$, there is a data structure $\mathcal{D}_{M,\mathcal{T}}$ with the following properties.*

i) (Construction:) We can construct $\mathcal{D}_{M,\mathcal{T}}$ in time $\mathcal{O}(n^2 + k \log^2 n)$.

ii) (Reachability Query:) Given $F \subseteq \mathcal{T}$, we can determine in time $\mathcal{O}(k \log^3 n)$ whether there is a monotone path from $(1,1)$ to (n,n) using only positions (i,j) with $M_{i,j} = 1$ or $(i,j) \in F$.

iii) (Update:) Given $\mathcal{T}' \subseteq [n] \times [n]$ of size k and an $n \times n$ matrix M' over $\{0,1\}$ differing from M in at most k positions, we can update $\mathcal{D}_{M,\mathcal{T}}$ to $\mathcal{D}_{M',\mathcal{T}'}$ in time $\mathcal{O}(n\sqrt{k} \log n + k \log^2 n)$. Here, we assume M' to be represented by the set Δ of positions in which M and M' differ.

Section 4 is dedicated to devising this data structure, i.e., proving Lemma 3.2. Equipped with this data structure, we can efficiently *batch* updates and queries to the data structure. Specifically, we obtain the following theorem.

Theorem 3.1. *We can solve Offline Dynamic Grid Reachability in time $\mathcal{O}(n^2 + Un^{2/3} \log^2 n)$.*

We prove this theorem in Section 3.1. Finally, it remains to use standard techniques of parametric search to transform the decision algorithm to an algorithm computing the discrete Fréchet distance under translation. This has already been shown by Ben Avraham et al. [AKS15]; for all details, we also refer to the full version of this paper [BKN18].

Lemma 3.3. *Let $T_{\text{dec}}(n)$ be the running time to decide, given $\delta > 0$ and polygonal curves π, σ of length n over $\mathbb{R}^2$, whether $\delta_F(\pi, \sigma + \tau) \leq \delta$ for some $\tau \in \mathbb{R}^2$. Then there is an algorithm computing the discrete Fréchet distance under translation for any curves π, σ of length n over $\mathbb{R}^2$ in time $\mathcal{O}((n^4 + T_{\text{dec}}(n)) \log n)$.*

Combining Lemma 3.3, Lemma 3.1 and Theorem 3.1, we obtain an algorithm computing the discrete Fréchet distance under translation in time

$$\mathcal{O}((n^4 + T(n, n^4)) \log n) = \mathcal{O}(n^{4+2/3} \log^3 n),$$

as desired. In the remainder of this section, we provide the details of solving Offline Dynamic Grid Reachability assuming Lemma 3.2, which we show in Section 4.

3.1 Solving Offline Dynamic Grid Reachability

We prove Theorem 3.1 using the grid reachability data structure given in Lemma 3.2. Specifically, we claim that the following algorithm solves Offline Dynamic Grid Reachability in time $\mathcal{O}(n^2 + Un^{2/3} \log^2 n)$.

We partition our updates $u_1, \ldots, u_U$ into groups $u_1, \ldots, u_{\mathcal{O}(U/k)}$ containing k updates each. For any group $\bar{u}_i$, let $\mathcal{T}_i$ denote the set of positions of updates in $\bar{u}_i$ and consider the grid reachability data structure $\mathcal{D}_i = \mathcal{D}_{M_i,\mathcal{T}_i}$ with terminal set $\mathcal{T}_i$ and matrix M_i obtained from M by performing all updates prior to $\bar{u}_i$ and setting the positions of all terminals $\mathcal{T}_i$ to 0. For each update within $\bar{u}_i$, we can determine whether it creates a monotone 1-path from $(1,1)$ to (n,n) by simply determining the set $F \subseteq \mathcal{T}_i$ of terminals that are set to 1 (at the point of this update) and performing the corresponding reachability query in $\mathcal{D}_i$. It is straightforward to argue that the resulting algorithm correctly solves Offline Dynamic Grid Reachability.

To analyze the running time, note that by Lemma 3.2, we need time $\mathcal{O}(n^2 + k\log^2 n)$ to build $\mathcal{D}_1 = \mathcal{D}_{M_1,\mathcal{T}_1}$. The time spent for handling a single group $\bar{u}_i$ is bounded by the time to perform k queries in $\mathcal{D}_i = \mathcal{D}_{M_i,\mathcal{T}_i}$ plus the time to update $\mathcal{D}_i = \mathcal{D}_{M_i,\mathcal{T}_i}$ to $\mathcal{D}_{i+1} = \mathcal{D}_{M_{i+1},\mathcal{T}_{i+1}}$, which amounts to $\mathcal{O}(k^2\log^3 n + n\sqrt{k}\log n + k\log^2 n) = \mathcal{O}(k^2\log^3 n + n\sqrt{k}\log n)$ by Lemma 3.2. Thus, in total, we obtain a running time of

$$\mathcal{O}\left(n^2 + U\left(k\log^3 n + \frac{n}{\sqrt{k}}\log n\right)\right).$$

This expression is minimized by setting $k := n^{2/3}/\log^{4/3} n$, resulting in a total running time of $\mathcal{O}(n^2 + Un^{2/3}\log^{1+2/3} n) = \mathcal{O}(n^2 + Un^{2/3}\log^2 n)$, as desired.

4 Grid Reachability Data Structure

In this section, we prove Lemma 3.2. First, we state some basic definitions and then present the details of our construction.

4.1 Preparation Without loss of generality, we may assume that $n = 2^\kappa + 1$ for some integer $\kappa \in \mathbb{N}$.

Canonical blocks. Let I, J be intervals in $[n]$. We call $I \times J \subseteq [n]\times[n]$ a *block*. In particular, we only consider blocks obtained by splitting the square $[n] \times [n]$ alternately horizontally and vertically until we are left with 2×2 blocks. More concretely, we define $\mathcal{B}_0 := \{([n],[n])\}$ and construct $\mathcal{B}_{\ell+1}$ inductively by splitting each block $B \in \mathcal{B}_\ell$ as follows. If the last split was horizontally, then we now split vertically and vice versa. The block B is split into two equally sized blocks B_1, B_2 which overlap in the middle column when split vertically, or the middle row when split horizontally. We call B_1, B_2 the *children* of B. We then let $\mathcal{B} := \bigcup_{\ell=0}^{2\kappa} \mathcal{B}_\ell$ be the set of *canonical blocks*, and call each block $B \in \mathcal{B}_\ell$ a *canonical block on level* ℓ. See Figure 2.

Boundaries. For any $B = (I,J) \in \mathcal{B}$, we denote the lower left boundary of B as $B^- = \{\min I\} \times J \cup I \times \{\min J\}$, and call each $p \in B^-$ an *input* of B. Analogously, we denote the upper right boundary of B as $B^+ = \{\max I\} \times J \cup I \times \{\max J\}$, and call each $q \in B^+$ an *output* of B. By slight abuse of notation, we define $|\partial B| = |B^- \cup B^+|$ as the size of the boundary of B, i.e., the number of inputs and outputs of B.

If B splits into children B_1, B_2, we call $B^{\mathrm{mid}} = B_1^+ \cap B_2^-$ the *splitting boundary* of B.

Indices. To prepare the description of this information, we first define, for technical reasons, *indices* for all positions in $[n]\times[n]$. It allows us to give each position a unique identifier with the property that for any canonical block B, the indices yield a local ordering of the boundaries.

Observation 4.1. *Let* $\mathrm{ind} : [n]\times[n] \to \mathbb{N}$, *where for any point* $p = (x,y) \in [n]\times[n]$, *we set* $\mathrm{ind}(p) := (y-x)(2n) + x$. *We call* $\mathrm{ind}(p)$ *the* index *of* p. *This function satisfies the following properties:*

1. *The function* ind *is injective, can be computed in constant time, and given* $i = \mathrm{ind}(p)$, *we can determine* $\mathrm{ind}^{-1}(i) := p$ *in constant time.*
2. *For any* $B \in \mathcal{B}$, ind *induces an ordering of* B^+ *in counter-clockwise order and an ordering of* B^- *in clockwise order.*

4.2 Connectivity characterization Our aim is to construct a data structure $\mathcal{D}_{M,\mathcal{T}} = (\mathcal{D}_{M,\mathcal{T}}(B))_{B\in\mathcal{B}}$, where $\mathcal{D}_{M,\mathcal{T}}(B)$ succinctly describes connectivity (via monotone 1-paths) between the boundaries B^-, B^+ and the terminals $\mathcal{T}_B := \mathcal{T} \cap B$ inside B. In particular, we show that we only require space $\mathcal{O}(|\partial B| + |\mathcal{T}_B|)$ to represent this information.

To prepare this, we start with a few simple observations that yield a surprisingly simple characterization of connectivity from any terminal to the boundary.

Compositions of crossing paths. We say that we reach q from p, written $p \rightsquigarrow q$, if there is a traversal $T = (t_1, \ldots, t_\ell)$ with $t_1 = p$, $t_\ell = q$, and t_i is free for all $1 < i < \ell$ (note that we do not require t_1 and t_ℓ to be free). We call such a slightly adapted notion of traversal a *reach traversal*. By connecting the points of T by straight lines, we may view T as a polygonal curve in $\mathbb{R}^2$. To avoid confusion, we denote this polygonal line as $P(T)$.

Observation 4.2. *Let* T_1, T_2 *be reach traversals from* p_1 *to* q_1 *and from* p_2 *to* q_2, *respectively. Then if* $P(T_1)$ *and* $P(T_2)$ *intersect, we have* $p_1 \rightsquigarrow q_2$ *(and, symmetrically,* $p_2 \rightsquigarrow q_1$*).*

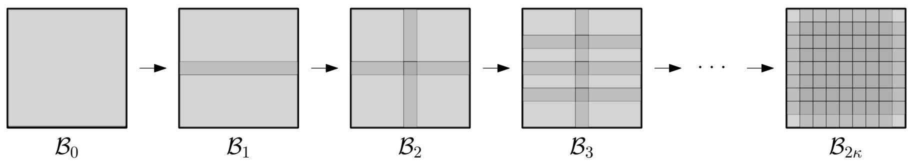

Figure 2: The sets of canonical blocks $\mathcal{B}_0, \dots, \mathcal{B}_{2\kappa}$. We alternate between horizontal and vertical splits. Note that the blocks overlap which is visualized by darker gray tones.

Proof. Let $t \in [n] \times [n]$ be a free position in which $P(T_1), P(T_2)$ intersect (observe that such a point with integral coordinates must exist unless $p_1 = p_2$ or $q_1 = q_2$; in the latter case, the claim is trivial). Note that t splits T_1, T_2 into $T_1 = T_1^a \circ T_1^b$ and $T_2 = T_2^a \circ T_2^b$ such that T_1^a, T_2^a are reach traversals ending in t and T_1^b, T_2^b are reach traversal starting in t. By concatenating T_1^a and T_2^b, we obtain a reach traversal from p_1 to q_2. Symmetrically, $T_2^a \circ T_1^b$ proves $p_2 \rightsquigarrow q_1$.

Let $B \in \mathcal{B}$ and recall that $\mathrm{ind}(\cdot)$ orders B^+ counter-clockwise. For any $p \in B$, we define $\mathrm{A}(p) := \min\{\mathrm{ind}(q) \mid q \in B^+, p \rightsquigarrow q\}$, and analogously $\mathrm{Z}(p) := \max\{\mathrm{ind}(q) \mid q \in B^+, p \rightsquigarrow q\}$. Note that in the following analysis we slightly abuse notation by also using $\mathrm{ind}(p)$ to denote the corresponding (unique) position $p \in [n] \times [n]$.

Definition 4.1. *Let $p \in B$ with $\infty > \mathrm{A}(p), \mathrm{Z}(p) > -\infty$ and fix any reach traversals T_A, T_Z from p to $\mathrm{A}(p)$ and $\mathrm{Z}(p)$. We write*

$$P(T_A) = P_{\text{com}} \circ P'_A, \quad P(T_Z) = P_{\text{com}} \circ P'_Z,$$

for some polygonal curves $P_{\text{com}}, P'_A, P'_Z$ with P'_A, P'_Z non-intersecting. Let $\mathcal{F}$ be the face enclosed by P'_A, P'_Z and the path from $\mathrm{A}(p)$ to $\mathrm{Z}(p)$ on B^+ (if $\mathrm{A}(p) = \mathrm{Z}(p)$, we let $\mathcal{F}$ be the empty set). We define the reach region *of p as*

$$\mathcal{R}(p) := \mathcal{F} \cup P_{\text{com}}.$$

We refer to Figure 3 for an illustration. Observe that $\mathcal{R}(p)$ is indeed well-defined: For any reach traversals T'_A, T'_Z from p to $\mathrm{A}(p)$ and $\mathrm{Z}(p)$, respectively, consider the latest point in which $P(T'_A), P(T'_B)$ intersect, say t. We can define reach traversals T_A and T_Z by following T'_A until t and then following the remainder of T'_A or T'_Z to reach $\mathrm{A}(p)$ or $\mathrm{Z}(p)$, respectively. These traversals satisfy the conditions by construction.

Proposition 4.1. *Let $p, p' \in B$, $q \in B^+$ with $\mathrm{ind}(q) \in [\mathrm{A}(p), \mathrm{Z}(p)]$ and $p' \notin \mathcal{R}(p)$. Then $p' \rightsquigarrow q$ implies $p \rightsquigarrow q$.*

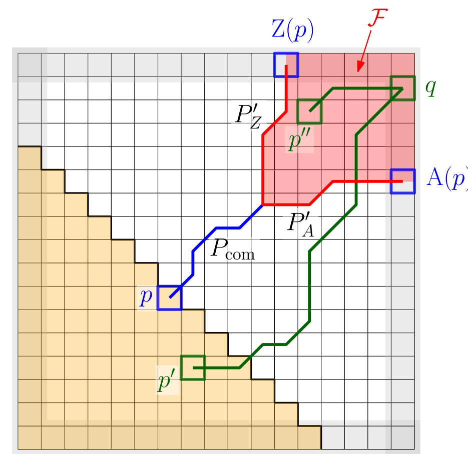

Figure 3: Illustration of $\mathcal{R}(p)$, Proposition 4.1 and Lemma 4.1: Any reach traversal from $p' \notin \mathcal{R}(p)$ must cross P'_A or P'_Z to reach q. However, if $p'' \rightsquigarrow q$ but $p'' \in \mathcal{R}(p)$, then q might not be reachable from p. A sufficient condition for $p' \notin \mathcal{R}(p)$ is that $p' \neq p$ and $L(p') \leq L(p)$ (indicated by the orange triangular area).

Proof. The claim holds trivially if $\mathrm{ind}(q) = \mathrm{A}(p)$ or $\mathrm{ind}(q) = \mathrm{Z}(p)$. Thus, we may assume that $\mathrm{A}(p) < \mathrm{Z}(p)$, which implies that the face $\mathcal{F}$ in $\mathcal{R}(p)$ is nonempty with $q \in \mathcal{F}$ and $p' \notin \mathcal{F}$. Hence any reach traversal T from p' to q must cross the boundary of $\mathcal{F}$, in particular, the path $P(T_A)$ or $P(T_Z)$, where T_A, T_Z both originate in p. By Observation 4.2, this yields $p \rightsquigarrow q$.

Reachability Labelling. We define a total order on nodes in B that allows us to succinctly represent reachability on B^+ for any subset $S \subseteq B$ in space $\tilde{O}(|S|+|B^+|)$. The key is a labelling $L : [n] \times [n] \to \mathbb{N}$, defined by $L((x, y)) = x + y$, that we call the *reachability labelling*.

Lemma 4.1. *Let $p = (x, y), p' = (x', y') \in B$ with $L(p') \leq L(p)$ and $q \in B^+$ with $\mathrm{ind}(q) \in [\mathrm{A}(p), \mathrm{Z}(p)]$. Then $p' \rightsquigarrow q$ implies $p \rightsquigarrow q$.*

Proof. The proof idea is to show that $L(p') \leq L(p)$ implies that $p' \notin \mathcal{R}(p)$, and hence Proposition 4.1 shows the claim. Note that by monotonicity of reach traversals, any point $r = (r_x, r_y) \in \mathcal{R}(p)$ satisfies $r_x \geq x$ and $r_y \geq y$. Thus, $p' \in \mathcal{R}(p)$ only if $x' \geq x$, $y' \geq y$, but this together with $x' + y' = L(p') \leq L(p) = x + y$ implies $(x', y') = (x, y)$. Summarizing, we either have $p = p'$, which trivially satisfies the claim, or $p' \notin \mathcal{R}(p)$, which yields the claim by Proposition 4.1.

For any $S \subseteq B$, this labelling enables a surprising characterization of which terminals in S have reach traversals to which outputs in B^+ by the following lemma (greatly generalizing a simpler characterization due to Ben Avraham et al. [AKS15, implicit in Lemma 4.4] for the case of $S = B^-$). This is one of our key insights.

Corollary 4.1. *Let $q \in B^+$ and define $\ell(q) := \min\{L(p) \mid p \in B, p \rightsquigarrow q\}$. Then for any $p \in B$, we have $p \rightsquigarrow q$ if and only if* $\mathrm{ind}(q) \in [\mathrm{A}(p), \mathrm{Z}(p)]$ *and $\ell(q) \leq L(p)$.*

Proof. Clearly, $p \rightsquigarrow q$ implies, by definition of $\mathrm{A}(p), \mathrm{Z}(p)$, and $\ell(q)$, that $\mathrm{A}(p) \leq \mathrm{ind}(q) \leq \mathrm{Z}(p)$ and $\ell(q) \leq L(p)$.

Conversely, assume that $\mathrm{ind}(q) \in [\mathrm{A}(p), \mathrm{Z}(p)]$ and $\ell(q) \leq L(p)$. Take any $p' \in B$ with $p' \rightsquigarrow q$ and $\ell(q) = L(p')$. Thus we have $L(p') = \ell(q) \leq L(p)$, $\mathrm{ind}(q) \in [\mathrm{A}(p), \mathrm{Z}(p)]$ and $p' \rightsquigarrow q$, which satisfies the requirements of Lemma 4.1, yielding $p \rightsquigarrow q$.

Given this characterization, we obtain a highly succinct representation of connectivity information. Specifically, to represent the information which terminals in S have reach traversals to which outputs in B^+, we simply need to store $\ell(q)$ for all $q \in B^+$ as well as the interval $[\mathrm{A}(p), \mathrm{Z}(p)]$ for all $p \in S$. Thus, the space required to store this information amounts to only $\mathcal{O}(|\partial B| + |S|)$, which greatly improves over a naive $\mathcal{O}(|\partial B| \cdot |S|)$-sized tabulation.

Reverse Information. By defining $L^{\mathrm{rev}}((x, y)) = -L((x, y)) = -x - y$, we obtain a labelling with symmetric properties. In particular, define $\mathrm{A}^{\mathrm{rev}}(q) := \min\{\mathrm{ind}(p) \mid p \in B^-, p \rightsquigarrow q\}$ and $\mathrm{Z}^{\mathrm{rev}}(q) := \max\{\mathrm{ind}(p) \mid p \in B^-, p \rightsquigarrow q\}$. It is straightforward to prove the following symmetric variant of Corollary 4.1.

Corollary 4.2. *Let $p \in B^-$ and define $\ell^{\mathrm{rev}}(p) := \min\{L^{\mathrm{rev}}(q) \mid q \in B, p \rightsquigarrow q\}$. Then for any $q \in B$, we have $p \rightsquigarrow q$ if and only if* $\mathrm{ind}(q) \in [\mathrm{A}^{\mathrm{rev}}(p), \mathrm{Z}^{\mathrm{rev}}(p)]$ *and $\ell^{\mathrm{rev}}(q) \leq L^{\mathrm{rev}}(p)$.*

4.3 Information stored at canonical block B

Using the characterization given in Corollaries 4.1 and 4.2, we can now describe which information we need to store for any canonical block $B \in \mathcal{B}$.

Definition 4.2. *Let $B \in \mathcal{B}$. The* information stored *at B (which we denote as $\mathcal{D}_{M,\mathcal{T}}(B)$) consists of the following information: First, we store* forward connectivity information *consisting of,*

- *for every $p \in B^- \cup \mathcal{T}_B$, the interval $\mathcal{I}(p) := [\mathrm{A}(p), \mathrm{Z}(p)]$, where $\mathrm{A}(p) = \min\{\mathrm{ind}(q) \mid q \in B^+, p \rightsquigarrow q\}$, and $\mathrm{Z}(p) = \max\{\mathrm{ind}(q) \mid q \in B^+, p \rightsquigarrow q\}$ (note that $\mathcal{I}(p)$ might be empty if $\mathrm{A}(p) = \infty, \mathrm{Z}(p) = -\infty$),*
- *for every $q \in B^+$, the* reachability level *$\ell(q) = \min\{L(p) \mid p \in B, p \rightsquigarrow q\}$.*

Symmetrically, we store the reverse connectivity information $\mathcal{I}_{\mathrm{rev}}(q) := [A_{\mathrm{rev}}(q), Z_{\mathrm{rev}}(q)]$ for all $q \in B^+ \cup \mathcal{T}_B$ and $\ell_{\mathrm{rev}}(p)$ for all $p \in B^-$.

Finally, if B has children $B_1, B_2 \in \mathcal{B}$, where B_1 is the lower or left sibling of B_2, we additionally store an orthogonal range minimization data structure $\mathcal{OR}_B$ storing, for each free *$q \in B^{\mathrm{mid}} = B_1^+ \cap B_2^-$, the value $\ell_2^{\mathrm{rev}}(q)$ under the key $(\mathrm{ind}(q), \ell_1(q))$. Here $\ell_1(q)$ denotes the forward reachability level in B_1, and $\ell_2^{\mathrm{rev}}(q)$ denotes the reverse reachability level in B_2.*

4.4 Computing Information at Parent From Information at Children

We show how to construct the information stored at the blocks quickly in a recursive fashion.

Lemma 4.2. *Let $B \in \mathcal{B}$ with children B_1, B_2. Given the information stored at B_1 and B_2, we can compute the information stored at B in time $\mathcal{O}((|\partial B| + |\mathcal{T}_B|) \log |\partial B|)$.*

Proof. Without loss of generality, we assume that B_1, B_2 are obtained from B by a vertical split (the other case is analogous) – let $B_{\mathbf{l}}, B_{\mathbf{r}}$ denote the left and right child, respectively. As a convention, we equip the information stored at $B_{\mathbf{l}}, B_{\mathbf{r}}$ with the subscript $\mathbf{l}, \mathbf{r}$, respectively, and write the information stored at B without subscript. Furthermore, we let $B^{\mathrm{mid}}_{\mathrm{free}}$ denote the set of *free* positions of the splitting boundary $B^{\mathrm{mid}} = B_{\mathbf{l}}^+ \cap B_{\mathbf{r}}^-$.

Computation of $\mathcal{I}(p)$. Let $p \in B^- \cup \mathcal{T}_B$ be arbitrary. We first explain how to compute $\mathrm{A}(p)$. If $p \in B_{\mathbf{r}}$, then $\mathrm{A}(p) = \mathrm{A}_{\mathbf{r}}(p)$, since by monotonicity any $q \in B^+$ with $p \rightsquigarrow q$ satisfies $q \in B_{\mathbf{r}}^+$. Thus, it remains to consider $p \notin B_{\mathbf{r}}$.

We claim that for $p \notin B_{\mathbf{r}}$, we have $A(p) = \min\{A_1(p), A_2(p)\}$, where

$$A_1(p) := \min_{\substack{q \in B^+ \cap B_{\mathbf{l}},\\ p \rightsquigarrow q}} \mathrm{ind}(q)$$

$$A_2(p) := \min_{\substack{j \in B^{\mathrm{mid}}_{\mathrm{free}},\\ p \rightsquigarrow j}} \min_{\substack{q \in B^+ \cap B_{\mathbf{r}},\\ j \rightsquigarrow q}} \mathrm{ind}(q)$$

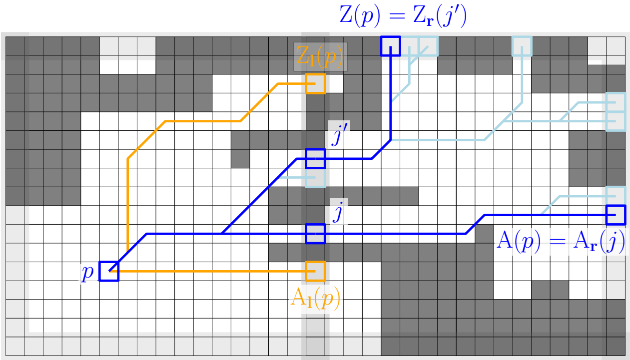

Figure 4: Computation of $\mathcal{I}(p)$. To determine the smallest (largest) reachable index on $B^+ \cap B_r^+$, we optimize, over all $j \in B^{\mathrm{mid}}$ with $p \rightsquigarrow j$, the smallest (largest) reachable index $A_r(j)$ ($Z_r(j)$) on B_r^+.

Indeed, this follows since each path starting in $p \in B_l$ and ending in B^+ must end in B_l, or cross B^{mid} at some free $j \in B^{\mathrm{mid}}$ and end in B_r.

To compute $A_1(p)$ note that Corollary 4.1 yields $A_1(p) = \min\{\mathrm{ind}(q) \mid q \in B^+ \cap B_l, \mathrm{ind}(q) \in [A_l(p), Z_l(p)], \ell_l(q) \leq L(p)\}$, which can be expressed as an orthogonal range minimization query. Likewise, to compute $A_2(p)$, note that $B^+ \cap B_r = B_r^+$. Thus,

$$A_2(p) = \min_{\substack{j \in B^{\mathrm{mid}}_{\mathrm{free}},\\ p \rightsquigarrow j}} \min_{\substack{q \in B_r^+,\\ j \rightsquigarrow q}} \mathrm{ind}(q) = \min_{\substack{j \in B^{\mathrm{mid}}_{\mathrm{free}},\\ \mathrm{ind}(j) \in \mathcal{I}_l(p), \ell_l(j) \leq L(p)}} A_r(j),$$

where the last equality follows from the definition of A_r and Corollary 4.1. Thus, we can compute $A_2(p)$ using a simple orthogonal range minimization query.

Switching the roles of minimization and maximization, we obtain the analogous statements for computing $Z(p)$. We summarize the algorithm formally in Algorithm 1. Its correctness follows from the arguments above and the total running time amounts to $\mathcal{O}((|\partial B| + |\mathcal{T}_B|) \log |\partial B|)$.

Computation of $\ell(q)$. Let $q \in B^+$ be arbitrary. If $q \in B_l$, then $\ell(q) = \ell_l(p)$, since by monotonicity every $p \in B$ with $p \rightsquigarrow q$ is contained in B_l. Thus, we may assume that $q \notin B_l$.

We claim that for $q \notin B_l$, we have $\ell(q) = \min\{\ell_1(q), \ell_2(q)\}$, where

$$\ell_1(q) := \min_{\substack{p \in B_r,\\ p \rightsquigarrow q}} L(p), \qquad \ell_2(q) := \min_{\substack{j \in B^{\mathrm{mid}}_{\mathrm{free}},\\ j \rightsquigarrow q}} \min_{\substack{p \in B_l,\\ p \rightsquigarrow j}} L(p)$$

Indeed, this follows since each path starting in B and ending in $q \in B_r$ must start in B_r, or start in B_l and cross B^{mid} at some free $j \in B^{\mathrm{mid}}$.

Observe that the definition of $\ell_1(q)$ coincides with the definition of $\ell_r(q)$. Thus it only remains

Algorithm 1 Computing $\mathcal{I}(p) = [A(p), Z(p)]$ for all $p \in B^- \cup \mathcal{T}_B$.

Build $\mathcal{OR}_A$ storing $A_r(j)$ under the key $(\mathrm{ind}(j), \ell_l(j))$ for $j \in B^{\mathrm{mid}}_{\mathrm{free}}$ (for min queries)
Build $\mathcal{OR}_Z$ storing $Z_r(j)$ under the key $(\mathrm{ind}(j), \ell_l(j))$ for $j \in B^{\mathrm{mid}}_{\mathrm{free}}$ (for max queries)
Build $\mathcal{OR}_{\mathrm{top}}$ storing $\mathrm{ind}(q)$ under the key $(\mathrm{ind}(q), \ell_l(q))$ for $q \in B^+ \cap B_l$ (for both queries)
for $p \in (B^- \cup \mathcal{T}_B)$ **do**
  if $p \in B_r$ **then**
    $\mathcal{I}(p) \leftarrow \mathcal{I}_r(p)$
  else
    range $\leftarrow [A_l(p), Z_l(p)] \times (-\infty, L(p)]$
    $A_1(p) \leftarrow \mathcal{OR}_{\mathrm{top}}.\min(\mathrm{range})$
    $A_2(p) \leftarrow \mathcal{OR}_A.\min(\mathrm{range})$
    $A(p) \leftarrow \min\{A_1(p), A_2(p)\}$
    $Z_1(p) \leftarrow \mathcal{OR}_{\mathrm{top}}.\max(\mathrm{range})$
    $Z_2(p) \leftarrow \mathcal{OR}_Z.\max(\mathrm{range})$
    $Z(p) \leftarrow \max\{Z_1(p), Z_2(p)\}$

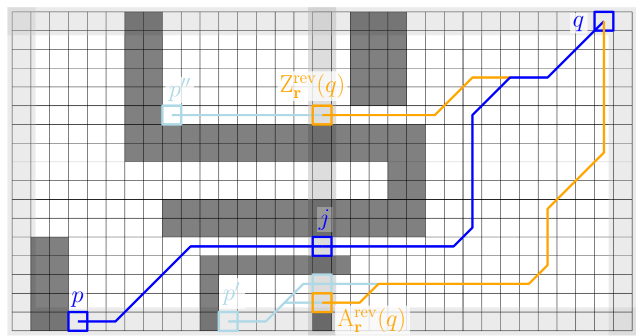

Figure 5: Computation of $\ell(q)$. To determine the smallest label of a position in B_l reaching q, we optimize, over all $j \in B^{\mathrm{mid}}$ with $j \rightsquigarrow q$, the smallest label $\ell_l(j)$ of a position $p \in B_l$ reaching j.

to compute $\ell_2(q)$. We write

$$\ell_2(q) = \min_{\substack{j \in B^{\mathrm{mid}}_{\mathrm{free}},\\ j \rightsquigarrow q}} \min_{\substack{p \in B_l,\\ p \rightsquigarrow j}} L(p) = \min_{\substack{j \in B^{\mathrm{mid}}_{\mathrm{free}},\\ \mathrm{ind}(j) \in \mathcal{I}_r^{\mathrm{rev}}(q), \ell_r^{\mathrm{rev}}(j) \leq L^{\mathrm{rev}}(q)}} \ell_l(j),$$

where the last equality follows from the definition of $\ell_l(j)$ and Corollary 4.2. It follows that we can compute $\ell_2(p)$ using a simple orthogonal range minimization query. For an illustration of $\ell_2(q)$, we refer to Figure 5.

We summarize the resulting algorithm formally in Algorithm 2 and illustrate it in Figure 5. Its correctness follows from the arguments above and the total running time amounts to $\mathcal{O}(|\partial B| \log |\partial B|)$.

Computation of reverse information. Switching the direction of reach traversals (which switches roles of inputs and outputs, B_l and B_r, etc.) as well as L and L^{rev}, we can use the same algorithms

Algorithm 2 Computing $\ell(q)$ for all $q \in B^+$.

Build $\mathcal{OR}_\ell$ storing $\ell_{\mathbf{l}}(j)$ under the key $(\mathrm{ind}(j), \ell_{\mathbf{r}}^{\mathrm{rev}}(j))$ for $j \in B_{\mathrm{free}}^{\mathrm{mid}}$ (for min queries)
for $q \in B^+$ **do**
  if $q \in B_{\mathbf{l}}$ **then**
    $\ell(q) \leftarrow \ell_{\mathbf{l}}(q)$
  else
    range $\leftarrow [\mathrm{A}_{\mathbf{r}}^{\mathrm{rev}}(q), \mathrm{Z}_{\mathbf{r}}^{\mathrm{rev}}(q)] \times (-\infty, L^{\mathrm{rev}}(q)]$
    $\ell_2(q) \leftarrow \mathcal{OR}_\ell.\min(\mathrm{range})$
    $\ell(q) \leftarrow \min\{\ell_{\mathbf{r}}(q), \ell_2(q)\}$

to compute the reverse connectivity information in the same running time of $\mathcal{O}((|\partial B| + |\mathcal{T}_B|) \log |\partial B|)$.

Computation of $\mathcal{OR}_B$. Finally, we need to construct the two-dimensional orthogonal range minimization data structure $\mathcal{OR}_B$: Recall that $\mathcal{OR}_B$ stores, for each $q \in B_{\mathrm{free}}^{\mathrm{mid}}$, the value $\ell_{\mathbf{r}}^{\mathrm{rev}}(q)$ under the key $(\mathrm{ind}(q), \ell_{\mathbf{l}}(q))$ for minimization queries. Since $|B_{\mathrm{free}}^{\mathrm{mid}}| \leq |\partial B|$, this can be done in time $\mathcal{O}(|\partial B| \log |\partial B|)$ (cf. Section 2).

In summary, we can compute the information stored at B from the information stored at B_1 and B_2 in time $\mathcal{O}((|\partial B| + |\mathcal{T}_B|) \log |\partial B|)$, as desired.

4.5 Initialization and Updates We have to show how to construct and update our reachability data structure (using Lemma 4.2 that shows how to compute the information stored at some canonical block B given the information stored at both children). However, due to space constraints we omit the proof for i) of Lemma 3.2 and instead refer to the proof of iii) of Lemma 3.2, which is very similar, and also to the full version of this paper [BKN18]. Thus, we now prove iii) of Lemma 3.2.

Proof. [Proof of iii) of Lemma 3.2] Set $X \coloneqq \Delta \cup \mathcal{T} \cup \mathcal{T}'$ and note that $|X| = \mathcal{O}(k)$. Observe that for any $B \in \mathcal{B}$ with $B \cap X = \emptyset$, we have $\mathcal{D}_{M,\mathcal{T}}(B) = \mathcal{D}_{M',\mathcal{T}'}(B)$, since the information stored at this block does not depend on any changed entry in M and does not contain any of the old or new terminals. Thus, we only need to update $\mathcal{D}_{M,\mathcal{T}}(B)$ to $\mathcal{D}_{M',\mathcal{T}'}(B)$ for all $B \in \mathcal{B}$ with $B \cap X \neq \emptyset$. We do this analogously to i) of Lemma 3.2 in a bottom-up fashion. Specifically, for any lowest-level block $B \in \mathcal{B}_{2\kappa}$ with $B \cap X \neq \emptyset$, we can compute the information stored in B in constant time. Since there are at most $4|X|$ such blocks, this step takes time $\mathcal{O}(|X|) = \mathcal{O}(k)$ in total.

It remains to bound the running time to compute $\mathcal{D}_{M,\mathcal{T}}(B)$ for $B \in \mathcal{B}_\ell$ with $B \cap X \neq \emptyset$, where $0 \leq \ell < 2\kappa$. For any such B, we let again $c_B \coloneqq |\partial B| \log |\partial B| + |\mathcal{T}_B| \log |\partial B|$. Observe that the running time for the remaining task is thus bounded by $\mathcal{O}(\sum_{\ell=0}^{2\kappa-1} \sum_{B \in \mathcal{B}_\ell, B \cap X \neq \emptyset} c_B)$ by Lemma 4.2.

We do a case distinction into $0 \leq \ell < \bar{\ell}$ and $\bar{\ell} \leq \ell < 2\kappa$ where $\bar{\ell} \coloneqq \lfloor \log k \rfloor$. For the first case, we bound

$$\begin{aligned}
\sum_{\ell=0}^{\bar{\ell}-1} \sum_{\substack{B \in \mathcal{B}_\ell, \\ B \cap X \neq \emptyset}} |\partial B| \log |\partial B| &\leq \sum_{\ell=0}^{\bar{\ell}-1} \sum_{B \in \mathcal{B}_\ell} |\partial B| \log |\partial B| \\
&\leq \sum_{i=0}^{\bar{\ell}-1} 2^{\kappa+i/2+3}(\kappa - i/2 + 3) \leq \left(\sum_{i=0}^{\bar{\ell}-1} 2^{i/2} \right) 2^{\kappa+3}\kappa \\
&= (1+\sqrt{2})(2^{\bar{\ell}/2} - 1) 2^{\kappa+3}\kappa = \mathcal{O}(\sqrt{k} n \log n).
\end{aligned}$$

Recall that for any $0 \leq \ell < 2\kappa$, there are at most $4|X|$ blocks $B \in \mathcal{B}_\ell$ with $B \cap X \neq \emptyset$ and for any $B \in \mathcal{B}_\ell$, we have $|\partial B| \leq 2^{\kappa - \ell/2+3}$. We compute

$$\begin{aligned}
&\sum_{\ell=\bar{\ell}}^{2\kappa-1} \sum_{\substack{B \in \mathcal{B}_\ell, \\ B \cap X \neq \emptyset}} |\partial B| \log |\partial B| \\
&\leq \sum_{\ell=\bar{\ell}}^{2\kappa-1} 4|X| 2^{\kappa-\ell/2+3}(\kappa - \ell/2 + 3) \\
&\leq 4|X| 2^{\kappa - \bar{\ell}/2+3} \kappa \cdot \sum_{\ell=0}^{2\kappa-\bar{\ell}-1} 2^{-\ell/2} \\
&= \mathcal{O}\left(|X| \frac{n}{\sqrt{k}} \log n \right) = \mathcal{O}(\sqrt{k} n \log n).
\end{aligned}$$

Furthermore, as in the proof of i) of Lemma 3.2, we again compute

$$\begin{aligned}
\sum_{\ell=0}^{2\kappa-1} \sum_{\substack{B \in \mathcal{B}_\ell, \\ B \cap X \neq \emptyset}} |\mathcal{T}_B| \log |\partial B| &\leq \sum_{\ell=0}^{2\kappa-1} 4|\mathcal{T}|(\kappa - \ell/2 + 3) \\
&= \mathcal{O}(|\mathcal{T}|\kappa^2) = \mathcal{O}(|\mathcal{T}| \log^2 n).
\end{aligned}$$

Thus, in total we obtain a running time of $\mathcal{O}(k + n\sqrt{k} \log n + |\mathcal{T}| \log^2 n) = \mathcal{O}(n\sqrt{k} \log n + k \log^2 n)$.

4.6 Reachability queries It remains to show how to use the information stored at all canonical blocks to answer reachability queries quickly. Specifically, we now prove ii) of Lemma 3.2.

Recall that we aim to determine whether there is a monotone path in M using only positions (i,j) with $M_{i,j} = 1$ or $(i,j) \in F$, i.e., we view F as a set of *free terminals* (typically, $(i,j) \in F$ is a *non-free* position). In this section we assume, without loss of generality, that $(1,1), (n,n) \in \mathcal{T}_B$ (whenever we construct/update to the data structure $\mathcal{D}_{M,\mathcal{T}}$, we may construct/update to $\mathcal{D}_{M,\mathcal{T} \cup \{(1,1),(n,n)\}}$ in the same asymptotic running time).

For any block $B \in \mathcal{B}$, $S \subseteq F \subseteq \mathcal{T}_B$, we define the function $\mathrm{Reach}(B, S, F)$ that returns the set

$$R := \{t \in F \mid \exists f_1, \dots, f_\ell \in F : \\ f_1 \in S, f_\ell = t, f_1 \rightsquigarrow f_2 \rightsquigarrow \cdots \rightsquigarrow f_\ell\},$$

i.e., we interpret S as a set of admissible *starting positions* for a reach traversal and ask for the set of positions reachable from S using only free positions or free terminals. We call any such position *F-reachable from S*. (Recall that in the definition of $p \rightsquigarrow q$, only the intermediate points on a reach traversal from p and q are required to be free, while the endpoints p and q are allowed to be non-free.)

We show that $\mathrm{Reach}(B, S, F)$ can be computed in time $\mathcal{O}(|\mathcal{T}_B| \log^3 n)$. Given this, we can answer any reachability query in the same asymptotic running time: the reachability query asks whether there is a sequence $f_1, \dots, f_\ell \in F \cup \{(1,1),(n,n)\}$ such that (i) $f_1 = (1,1)$ and $f_\ell = (n,n)$, (ii) both $(1,1)$ and (n,n) are free positions or contained in F and (iii) $f_1 \rightsquigarrow f_2 \rightsquigarrow \cdots \rightsquigarrow f_\ell$. Since (ii) can be checked in constant time, it remains to determine whether

$$(n,n) \in \mathrm{Reach}([n]\times[n],\ \{(1,1)\},\ F \cup \{(1,1),(n,n)\}).$$

4.6.1 Computation of Reach(B, S, F) To compute $\mathrm{Reach}(B, S, F)$, we work on the recursive block structure of $\mathcal{D}_{M,\mathcal{T}}$. Specifically, consider any canonical block $B \in \mathcal{B}$ (containing some free terminal) with children B_1, B_2. The (somewhat simplified) approach is the following: We first (recursively) determine all free terminals that are F-reachable from S in B_1 and call this set R_1. Then, we determine all free terminals in B_2 that are (directly) reachable from R_1 and call this set T_2. Finally, we (recursively) determine all free terminals in B_2 that are F-reachable from $T_2 \cup (S \cap B_2)$ and call this set R_2. The desired set of free terminals that are F-reachable from S is then $R_1 \cup R_2$. The main challenge in this process is the computation of the set T_2; this task is solved by the following lemma.

Lemma 4.3. *Let $B \in \mathcal{B}$ be a block with children B_1, B_2. Given $S \subseteq B_1 \setminus B^{\mathrm{mid}}$ and $F \subseteq B_2 \setminus B^{\mathrm{mid}}$ with $S, F \subseteq \mathcal{T}_B$, we can compute the set*

$$T = \{t \in F \mid \exists s \in S : s \rightsquigarrow t\}$$

in time $\mathcal{O}(|\mathcal{T}_B| \log^2 n)$. We call this procedure $\mathrm{SingleStepReach}(B, S, F)$.

This lemma yields an algorithm for Reach, and thus, for reachability queries. Algorithm 3 computes the result in time $\mathcal{O}(|\mathcal{T}| \log^3 n)$. For the proof of correctness and runtime of ii) of Lemma 3.2, see the full version [BKN18].

Algorithm 3 Computing $\mathrm{Reach}(B, S, F)$ for $B \in \mathcal{B}$, $S \subseteq F \subseteq \mathcal{T}_B$.

```
function Reach(B, S, F)
    if F = ∅ then
        return ∅
    else if B is a 2 × 2 block then
        Compute R by checking all possibilities
        return R
            ▷ Else B splits into child blocks B_1, B_2
    S_1 ← S ∩ B_1, S_2 ← S ∩ B_2
    R_1 ← Reach(B_1, S_1, F ∩ B_1)
    T_2 ← SingleStepReach(B, R_1 \ B^mid, F \ B_1)
    R_2 ← Reach(B_2, S_2 ∪ T_2 ∪ (R_1 ∩ B^mid), F ∩ B_2)
    return R_1 ∪ R_2
```

4.6.2 Computing SingleStepReach(B, S, F) It remains to prove Lemma 4.3 to conclude the proof of ii) of Lemma 3.2.

Proof. [Proof of Lemma 4.3] Consider $B \in \mathcal{B}$. We only consider the case that B is split vertically (the other case is symmetric); let $B_{\mathbf{l}}, B_{\mathbf{r}}$ denote its left and right sibling, respectively. Let $S \subseteq B_{\mathbf{l}} \setminus B^{\mathrm{mid}}$, $F \subseteq B_{\mathbf{r}} \setminus B^{\mathrm{mid}}$ with $S, F \subseteq \mathcal{T}_B$ be arbitrary. We use notation (subscripts $\mathbf{l}, \mathbf{r}$, etc.) as in the proof of Lemma 4.2.

Observe that for any $s \in S, f \in F$, we have that $s \rightsquigarrow f$ if and only if there exists some $j \in B^{\mathrm{mid}}_{\mathrm{free}}$ with $s \rightsquigarrow j$ and $j \rightsquigarrow f$. To introduce some convenient conventions, let $J^{\mathrm{mid}} = \{j_1, \dots, j_N\}$, where $j_1, \dots, j_N$ is the sorted sequence of $\mathrm{ind}(q)$ with $q \in B^{\mathrm{mid}}_{\mathrm{free}}$. We call $J \subseteq J^{\mathrm{mid}}$ an interval of J^{mid} if $J = \{j_a, j_{a+1}, \dots, j_b\}$ for some $1 \le a \le b \le N$ and write it as $J = [j_a, j_b]_{J^{\mathrm{mid}}}$ (i.e., $[j_a, j_b]_{J^{\mathrm{mid}}}$ simply disregards any indices in $[j_a, j_b]$ representing positions outside of $B^{\mathrm{mid}}_{\mathrm{free}}$).

Consider any interval J of J^{mid} with the property that for all $s \in S$ we either have $J \cap \mathcal{I}_{\mathbf{l}}(s) = J$ or $J \cap \mathcal{I}_{\mathbf{l}}(s) = \emptyset$ and for all $f \in F$ we either have $J \cap \mathcal{I}^{\mathrm{rev}}_{\mathbf{r}}(f) = J$ or $J \cap \mathcal{I}^{\mathrm{rev}}_{\mathbf{r}}(f) = \emptyset$. We call such a J an *(S, F)-reach-equivalent*. Note that by splitting J^{mid} right before and right after all points $\mathrm{A}(s), \mathrm{Z}(s)$ with $s \in S$ and $\mathrm{A}^{\mathrm{rev}}(f), \mathrm{Z}^{\mathrm{rev}}(f)$ with $f \in F$, we obtain a partition of J^{mid} into (S, F)-reach-equivalent intervals $J_1, \dots, J_\ell$ with $\ell = \mathcal{O}(|S \cup F|) = \mathcal{O}(|\mathcal{T}_B|)$.[5]

Claim 4.1. *Let J be an (S, F)-reach-equivalent interval J. Let R^J be the set of $t \in F$ reachable from S*

[5]To be more precise, we start with the partition $\mathcal{J}$ consisting of the singleton J^{mid}. We then iterate over any point j among $\mathrm{A}(s), \mathrm{Z}(s), s \in S$ and $\mathrm{A}^{\mathrm{rev}}(f), \mathrm{Z}^{\mathrm{rev}}(f), f \in F$, and replace the interval $J = [j_a, j_b]_{J^{\mathrm{mid}}} \in \mathcal{J}$ containing j by the three intervals $[j_a, j)_{J^{\mathrm{mid}}}, \{j\}, (j, j_b]_{J^{\mathrm{mid}}}$, where the first and the last interval may be empty.

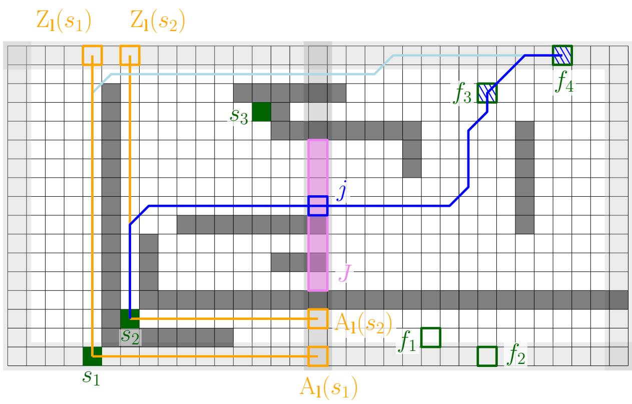

Figure 6: Computation of R^J for an (S, F)-reach equivalent interval J. Intuitively, we first determine, among indices in J reachable from some $s \in S$, the index $j \in J$ with the best connectivity towards F. We then determine all $f \in F$ reachable from j.

via J*, i.e.,* $R_J := \{t \in F \mid \exists s \in S, j \in J : s \rightsquigarrow j \rightsquigarrow t\}$. *Define* $\ell^J := \min_{\substack{j \in J, \\ \exists s \in S: s \rightsquigarrow j}} \ell_{\mathbf{r}}^{\text{rev}}(j)$. *We have*

$$R^J = \{t \in F \mid J \subseteq \mathcal{I}_{\mathbf{r}}^{\text{rev}}(t), \ell^J \leq L^{\text{rev}}(t)\}. \tag{4.1}$$

Proof. See Figure 6 for an illustration. Indeed, for any $t \in F$ with $J \subseteq \mathcal{I}_{\mathbf{r}}^{\text{rev}}(t)$ and $\ell^J \leq L^{\text{rev}}(t)$, consider any $j \in J$ with $\ell_{\mathbf{r}}^{\text{rev}}(j) = \ell^J$ and $s \rightsquigarrow j$ for some $s \in S$. Then we have $j \in J \subseteq \mathcal{I}_{\mathbf{r}}^{\text{rev}}(t)$ and $\ell_{\mathbf{r}}^{\text{rev}}(j) = \ell^J \leq L^{\text{rev}}(t)$. Thus by Corollary 4.2, $j \rightsquigarrow t$, which together with $s \rightsquigarrow j$ implies $s \rightsquigarrow j \rightsquigarrow t$, as desired. For the converse, let $t \in F$ with $s \rightsquigarrow j \rightsquigarrow t$ for some $s \in S, j \in J$. Then by definition of ℓ^J, we obtain $\ell^J \leq \ell_{\mathbf{r}}^{\text{rev}}(j)$. Furthermore, by Corollary 4.2, $j \rightsquigarrow t$ implies that $j \in \mathcal{I}_{\mathbf{r}}^{\text{rev}}(t)$ with $L^{\text{rev}}(t) \geq \ell_{\mathbf{r}}^{\text{rev}}(j) \geq \ell^J$. Note that $j \in \mathcal{I}_{\mathbf{r}}^{\text{rev}}(t)$ implies $J \subseteq \mathcal{I}_{\mathbf{r}}^{\text{rev}}(t)$ (as J is (S, F)-reach-equivalent), thus we obtain that $J \subseteq \mathcal{I}_{\mathbf{r}}^{\text{rev}}(t)$ and $\ell^J \leq \ell_{\mathbf{r}}^{\text{rev}}(j)$, as desired.

Thus, after computing ℓ^J, an orthogonal range reporting query can be used to report all $t \in F$ reachable from S via J. To compute ℓ^J, we observe that for any $j \in J$, we have

$$\begin{aligned} \exists s \in S : s \rightsquigarrow j &\overset{\text{Cor. 4.1}}{\iff} \exists s \in S : j \in \mathcal{I}_{\mathbf{l}}(s), \ell_{\mathbf{l}}(j) \leq L(s) \\ &\iff \ell_{\mathbf{l}}(j) \leq \max_{\substack{s \in S, \\ j \in \mathcal{I}_{\mathbf{l}}(s)}} L(s) =: L^j. \end{aligned}$$

Noting (by (S, F)-reach-equivalence of J) that $j \in \mathcal{I}_{\mathbf{l}}(s)$ if and only if $J \subseteq \mathcal{I}_{\mathbf{l}}(s)$, we have that L^j is independent of $j \in J$, and, in particular, equal to

$$L^J := \max_{\substack{s \in S, \\ J \subseteq \mathcal{I}_{\mathbf{l}}(s)}} L(s), \tag{4.2}$$

which can be computed by a single orthogonal range minimization query. Equipped with this value, we may determine ℓ^J as

$$\ell^J = \min_{\substack{j \in J, \\ \ell_{\mathbf{l}}(j) \leq L^J}} \ell_{\mathbf{r}}^{\text{rev}}(j). \tag{4.3}$$

Note that given ℓ^J, we may determine R^J by a single orthogonal range reporting query; by (4.1). We obtain Algorithm 4, whose correctness follows from above and the runtime amounts to $\mathcal{O}(|\mathcal{T}_B| \log^2 n)$. For the detailed analysis we refer to the full version of this paper [BKN18].

Algorithm 4 Computing SingleStepReach(B, S, F) for $B \in \mathcal{B}$, $S \subseteq B_{\mathbf{l}} \setminus B^{\text{mid}}$, $F \subseteq B_{\mathbf{r}} \setminus B^{\text{mid}}$.

function SingleStepReach(B, S, F)
  Compute a partitioning of J^{mid} into (S, F)-reach-equivalent intervals $J_1, \ldots, J_\ell$
  Build $\mathcal{OR}_S$ storing $L(s)$ under the key $(\mathrm{A}_{\mathbf{l}}(s), \mathrm{Z}_{\mathbf{l}}(s))$ for $s \in S$ (for max queries)
  Build $\mathcal{OR}_F$ storing $\mathrm{ind}(f)$ under the key $(\mathrm{A}_{\mathbf{r}}^{\text{rev}}(f), \mathrm{Z}_{\mathbf{r}}^{\text{rev}}(f), L^{\text{rev}}(f))$ for $f \in F$ (for dynamic dominance reporting queries)
  for $i = 1, \ldots, \ell$ **do** ▷ consider $J = [a_i, b_i]_{J^{\text{mid}}}$
    range $= (-\infty, a_i] \times [b_i, \infty)$
    $L^J \leftarrow \mathcal{OR}_S.\max(\text{range})$
    $\ell^J \leftarrow \mathcal{OR}_B.\min([a_i, b_i] \times (-\infty, L^J])$
    $R_i \leftarrow \mathcal{OR}_F.\text{report}(\text{range} \times [\ell^J, \infty))$
    $\mathcal{OR}_F.\text{delete}(R_i)$
  return $\bigcup_{i=1}^{\ell} R_i$

5 Conditional Lower Bound

In this section we prove a lower bound of $n^{4-o(1)}$ for the discrete Fréchet distance under translation for two curves of length $n \in \mathbb{R}^2$ conditional on the Strong Exponential Time Hypothesis, or more precisely the 4-OV Hypothesis. To this end, we reduce 4-OV to the discrete Fréchet distance under translation.

Let us first have a closer look at 4-OV. Given four sets of N vectors $V_1, \ldots, V_4 \subseteq \{0, 1\}^D$, the 4-OV problem can be expressed as

$$\begin{aligned} &\exists v_1 \in V_1, \ldots, v_4 \in V_4 \; \forall j \in [D] \\ &\qquad \exists i \in \{1, \ldots, 4\} : v_i[j] = 0. \end{aligned} \tag{5.4}$$

Recall from the introduction that we encode choosing the vectors $v_1, \ldots, v_4$ by the translation $\tau = (h_1 \cdot \epsilon, h_2 \cdot \epsilon)$ with $h_1, h_2 \in \{0, \ldots, N^2 - 1\}$ for some constant $\epsilon > 0$ which is sufficiently small, e.g., $\epsilon = 0.001/N^4$. Choosing $v_1 \in V_1$ and $v_2 \in V_2$, we define $h_1 := h(v_1, v_2) := \mathrm{ind}(v_1) + \mathrm{ind}(v_2) \cdot N$, where $\mathrm{ind}(v_i)$ is the index of vector v_i in the set V_i; similarly for

$v_3 \in V_3, v_4 \in V_4$ we define $h_2 := h(v_3, v_4)$. To perform the reduction, we want to construct two curves π and σ that implement the following expression, which is equivalent to (5.4):

$$\exists \tau \in \mathbb{R}^2 \; \forall j \in [D] \; \exists i \in \{1,2\}, v \in V_{2i-1}, v' \in V_{2i} : \\ (v[j] = 0 \vee v'[j] = 0) \wedge (h(v, v') \cdot \epsilon = \tau_i).$$

We can further transform this expression to make it easier to create gadgets for the reduction:

$$\exists \tau \in [0, (N^2-1) \cdot \epsilon] \times [0, (N^2-1) \cdot \epsilon] \\ \forall j \in [D] \bigvee_{\substack{i \in \{1,2\} \\ v \in V_{2i-1}, v' \in V_{2i}: \\ v[j]=0 \text{ or } v'[j]=0}} h(v, v') \cdot \epsilon = \tau_i.$$

According to this formula, we will construct the following gadgets:

- *Translation gadget:* It ensures that $\tau \in [-\frac{1}{4} \cdot \epsilon, (N^2 - \frac{3}{4}) \cdot \epsilon] \times [-\frac{1}{4} \cdot \epsilon, (N^2 - \frac{3}{4}) \cdot \epsilon]$, i.e. we are always close to the points in the ϵ-grid of translations that choose our vectors $v_1, \ldots, v_4$.
- *OV-dimension gadget:* AND over all $j \in [D]$.
- *OR gadget:* The big OR in the formula.
- *Equality gadget:* This gadget is only traversable if the two vectors it was created for correspond to τ, i.e., it ensures that $h(v, v') \cdot \epsilon \approx \tau_i$.

We use the above mentioned gadgets as follows. The constructed curves π and σ start with the translation gadget consisting of the curves $\pi^{(0)}, \sigma^{(0)}$. They are followed by D different parts that form the OV-dimension gadget. Each of the D parts is an OR gadget and we call the respective curves $\pi^{(j)}$ and $\sigma^{(j)}$ for $j \in [D]$. Each of the OR gadgets $(\pi^{(j)}, \sigma^{(j)})$ contains several equality gadgets. We will use different variations of the equality gadget (one for each set of vectors $V_1, \ldots, V_4$) but they are all of very similar structure. We need four different types of equality gadgets because for a certain $v_i \in V_i$ a part of the gadget is only inserted if $v_i[d] = 0$. Thus, if we traverse an equality gadget later, we know that it corresponds to one zero entry and also to the current translation. See Figure 7 for an overview.

Without loss of generality, assume that for all dimensions $j \in [D]$ at least one vector in $V_1 \cup \cdots \cup V_4$ contains a 0 in dimension j. Now we give the detailed construction of the gadgets and the proofs of correctness. The instance of the discrete Fréchet distance under translation that we construct in the reduction uses a threshold distance of $\delta = 2 + \frac{1}{4}\epsilon$, i.e. we want to know for the constructed curves π and σ if their discrete Fréchet distance under translation is not more than δ.

Translation Gadget. This gadget is also depicted in Figure 7. First we have to restrict the possible translations to ensure that $\tau \in [-\frac{1}{4} \cdot \epsilon, (N^2 - \frac{3}{4}) \cdot \epsilon] \times [-\frac{1}{4} \cdot \epsilon, (N^2 - \frac{3}{4}) \cdot \epsilon]$. This is realized by a gadget where curve $\pi^{(0)}$ consists of one and curve $\sigma^{(0)}$ of four vertices with $a = 2 - (N^2 - 1)\epsilon$:

$$\pi^{(0)} := \langle (0,0) \rangle, \; \sigma^{(0)} := \langle (a,0), (0,a), (-2,0), (0,-2) \rangle.$$

Lemma 5.1. *Given two curves π, σ with prefixes $\pi^{(0)}, \sigma^{(0)}$, such that all remaining points are in distance greater than 8 of the prefixes, the following holds:*

(i) if $\tau \in [0, (N^2-1)\epsilon] \times [0, (N^2-1)\epsilon]$, then $\delta_F(\pi^{(0)}, \sigma^{(0)} + \tau) \leq \delta$

(ii) if $\delta_F(\pi, \sigma + \tau) \leq \delta$, then $\tau \in [-\frac{1}{4} \cdot \epsilon, (N^2 - \frac{3}{4}) \cdot \epsilon] \times [-\frac{1}{4} \cdot \epsilon, (N^2 - \frac{3}{4}) \cdot \epsilon]$

Proof. We start with showing (i), so assume $\tau \in [0, (N^2-1)\epsilon] \times [0, (N^2-1)\epsilon]$. Note that the maximal distance $\max_{q \in \sigma^{(0)}} \max_\tau \left\| \pi^{(0)} - (q + \tau) \right\|$ is an upper bound on $\delta_F(\pi^{(0)}, \sigma^{(0)} + \tau)$. We obtain:

$$\max_{q \in \sigma^{(0)}} \max_\tau \left\| \pi^{(0)} - (q + \tau) \right\| < \sqrt{2^2 + \epsilon^2 N^4} \leq 2 + \frac{1}{4}\epsilon,$$

where we used $\epsilon \leq N^{-4}$.

Now we prove (ii). Note that the start points of π and σ have to be in distance $\leq \delta$, thus $\tau \in [-4, 1] \times [-2, 2]$ (using a very rough estimate). Using this and the fact that all points on π except $\pi^{(0)}$ are further than 8 from $\sigma^{(0)}$, we have to stay in $\pi^{(0)}$ while traversing $\sigma^{(0)}$. Thus, the following inequality holds for $\tau_i > (N^2 - \frac{3}{4})\epsilon$ or $\tau_i < -\frac{1}{4}\epsilon$ and $i \in \{1,2\}$ (where $\|v\|_\infty$ denotes the infinity norm of v):

$$\delta_F(\pi, \sigma + \tau) \geq \max_{i \in [4]} \left\{ \left\| \pi_1^{(0)} - (\sigma_i^{(0)} + \tau) \right\|_\infty \right\} > \delta,$$

which is the contrapositive of (ii).

For the remainder of this section we restrict τ to the range from the previous lemma, and thus for convenience define

$$\mathcal{T} := [-\frac{1}{4} \cdot \epsilon, (N^2 - \frac{3}{4}) \cdot \epsilon] \times [-\frac{1}{4} \cdot \epsilon, (N^2 - \frac{3}{4}) \cdot \epsilon].$$

OV-dimension Gadget. For every 4-OV dimension $j \in [D]$, we construct separate gadgets $\pi^{(1)}, \ldots, \pi^{(D)}$ for π and $\sigma^{(1)}, \ldots, \sigma^{(D)}$ for σ. We want to connect these gadgets in a way that the whole curve has distance not more than δ if and only if

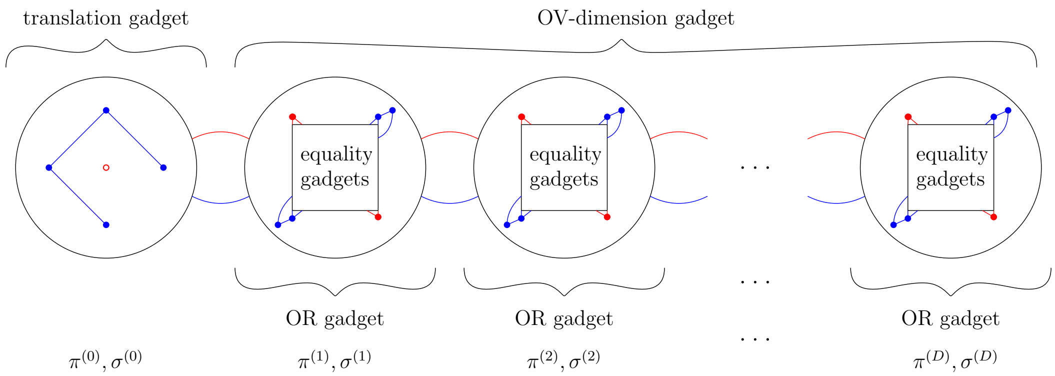

Figure 7: Overview of how the different gadgets are used in the curves that result from the reduction. We use one translation gadget, one OV-dimension gadget, D OR gadgets, and $\mathcal{O}(ND)$ equality gadgets.

all gadgets have distance not more than δ for a given translation τ. This is done by simply placing the gadgets in distance greater than $\delta + N^2 \cdot \epsilon$ from each other and concatenating them.

Lemma 5.2. *Given a translation $\tau \in \mathcal{T}$ and curves $\pi = \pi^{(1)}, \dots, \pi^{(D)}$ and $\sigma = \sigma^{(1)}, \dots, \sigma^{(D)}$ where for all $j \in [D]$ all points of $\pi^{(j)}$ are further than $\delta + 2N^2 \cdot \epsilon$ from each point of $\sigma^{(j')}$ with $j \neq j'$, then $\delta(\pi, \sigma + \tau) \leq \delta$ if and only if $\delta_F(\pi^{(j)}, \sigma^{(j)} + \tau) \leq \delta$ for all $j \in [D]$.*

Proof. First, note that whatever τ we choose in the given range, $\sigma^{(j)} + \tau$ is still in distance greater than δ from every $\pi^{(j')}$ with $j' \neq j$.

Now, assume that for all $j \in [D]$ the curves $\pi^{(j)}, \sigma^{(j)} + \tau$ have distance at most δ. Then we can traverse the gadgets in order and do simultaneous jumps between them. Note that those jumps do not change the distance. Thus, also the distance of the whole curves π and $\sigma + \tau$ is at most δ. For the other direction, assume that at least one distance is greater than δ. If we do not traverse simultaneously (i.e., at one point the traversal is in $\pi^{(j)}$ and $\sigma^{(j')}$ for $j \neq j'$), then due to large distances of $\pi^{(j)}, \sigma^{(j')} + \tau$ for $j \neq j'$ we have distance greater than δ for this traversal. On the other hand, a simultaneous traversal traverses $\pi^{(j)}$ and $\sigma^{(j)}$ together for all j, so we also have distance greater than δ due to the gadget with distance greater than δ.

Equality Gadget. For the remaining gadgets we define for convenience $\eta := 3 \cdot N^2\epsilon$. An equality gadget $F(v_1, v_2)$ for the vectors $v_1 \in V_1, v_2 \in V_2$ is a pair of two curves, $\pi_F(v_1)$ and $\sigma_F(v_2)$:

$$\begin{aligned}\pi_F(v_1) &:= \langle (1 + \operatorname{ind}(v_1)\epsilon, -1 - \eta),\\ &\qquad (-1 + \operatorname{ind}(v_1)\epsilon, 1 + \eta) \rangle,\\ \sigma_F(v_2) &:= \langle (-1 - \operatorname{ind}(v_2)N\epsilon, -1 - \eta),\\ &\qquad (1 - \operatorname{ind}(v_2)N\epsilon, 1 + \eta) \rangle.\end{aligned}$$

Note that this gives us N^2 different gadgets consisting of $2N$ different curves. We later use the curves $\pi_F(v_1)$ in π and the curves $\sigma_F(v_2)$ in σ where they can be matched to form a gadget.

Lemma 5.3. *Given curves $\pi_F(v_1), \sigma_F(v_2)$ for some $v_1 \in V_1$ and $v_2 \in V_2$, and given a translation $\tau \in \mathcal{T}$, the following properties hold:*

(i) if $\tau_1 = \epsilon \cdot (\operatorname{ind}(v_1) + \operatorname{ind}(v_2) \cdot N)$, then $\delta_F(\pi_F(v_1), \sigma_F(v_2) + \tau) \leq \delta$

(ii) if $\delta_F(\pi_F(v_1), \sigma_F(v_2) + \tau) \leq \delta$, then $|\epsilon \cdot (\operatorname{ind}(v_1) + \operatorname{ind}(v_2) \cdot N) - \tau_1| \leq \frac{1}{3}\epsilon$

Proof. To prove (i), it suffices to give a valid traversal. We traverse $\pi_F(v_1) = (p_1, p_2)$ and $\sigma_F(v_2) = (q_1, q_2)$ simultaneously. Thus, we just want an upper bound on the distance between the (translated) first points $p_1, q_1 + \tau$ and the distance between the (translated) second points $p_2, q_2 + \tau$ to get an upper bound on $\delta_F(\pi_F(v_1), \sigma_F(v_2) + \tau)$. These distances are

$$\begin{aligned}\|p_1 - (q_1 + \tau)\|^2 &= 4 + \tau_2^2 \leq 4 + \epsilon + \frac{1}{16}\epsilon^2 = \delta^2,\\ \|p_2 - (q_2 + \tau)\|^2 &= 4 + \tau_2^2 \leq \delta^2,\end{aligned}$$

where we used $|\tau_2| \leq N^2\epsilon$ and thus $\tau_2^2 \leq N^4\epsilon^2 \leq \epsilon$ since $\epsilon \leq N^{-4}$. Both distances are at most δ and thus the discrete Fréchet distance is at most δ as well.

For proving (ii), first note that the first (respectively second) point of $\pi_F(v_1)$ is far from the second (respectively first) point of $\sigma_F(v_2)$, due to $\eta \geq N^2\epsilon$. Thus, we have to traverse the gadget simultaneously. It remains to show that if the first two points are in distance not more than δ and the same holds for the second points, then τ_1 is close to $\epsilon \cdot (\operatorname{ind}(v_1) + \operatorname{ind}(v_2) \cdot N)$. In the following calculations let $\Delta := \epsilon \cdot \operatorname{ind}(v_1) + \epsilon \cdot \operatorname{ind}(v_2) \cdot N - \tau_1$. For p_1, q_1 we then get $\|p_1 - (q_1+\tau)\|^2$ is equal to

$$(2+\mathrm{ind}(v_1)\epsilon + \mathrm{ind}(v_2)N\epsilon - \tau_1)^2 + \tau_2^2 \le (2+\frac{1}{4}\epsilon)^2$$
$$\Leftrightarrow (2+\Delta)^2 + \tau_2^2 \le 4+\epsilon+\frac{1}{16}\epsilon^2$$
$$\Rightarrow 4\Delta \le \epsilon + \frac{1}{16}\epsilon^2 \Rightarrow \Delta \le \frac{1}{3}\epsilon.$$

With a similar calculation for p_2, q_2 we obtain that $\Delta \ge -\frac{1}{3}\epsilon$, and thus $|\Delta| \le \frac{1}{3}\epsilon$.

Now we introduce three gadgets which have the same properties as the equality gadget but are slightly different. The aim is to have four types of gadgets which are pairwise further than a discrete Fréchet distance of δ apart such that we can use them together in one big OR expression.

Shifted Equality Gadget. As described in the introduction of this section, we want to use the curves $\pi_F(v_1), \sigma_F(v_2)$ in case $v_1[j] = 0$ and we need an additional gadget for $v_2[j] = 0$. However, those two gadgets should not be too close such that the curves cannot be matched but also not too far such that the OR gadget (which we introduce later) still works. Thus, we introduce another gadget $F'(v_1, v_2)$ which consists of a pair of curves $\pi_{F'}(v_1), \sigma_{F'}(v_2)$ that are just shifted versions of $\pi_F(v_1), \sigma_F(v_2)$; shifted by $N^2\epsilon$ in the first dimension. More formally,

$$\pi_{F'}(v_1) := \pi_F(v_1) + (N^2\epsilon, 0),$$
$$\sigma_{F'}(v_2) := \sigma_F(v_2) + (N^2\epsilon, 0).$$

Before proving the desired properties, we introduce the remaining two variants of the equality gadget.

Equality Gadget for V_3 and V_4. The above introduced equality gadgets only work for vectors in V_1 and V_2 but we also need a gadget for vectors in V_3 and V_4. Therefore, we introduce the gadget $G(v_3, v_4)$, which is a mirrored equality gadget consisting of the curves $\pi_G(v_3)$ and $\sigma_G(v_4)$:

$$\begin{aligned}\pi_G(v_3) &:= \langle(-1-\eta, 1+\mathrm{ind}(v_3)\epsilon),\\ &\qquad (1+\eta, -1+\mathrm{ind}(v_3)\epsilon)\rangle,\\ \sigma_G(v_4) &:= \langle(-1-\eta, -1-\mathrm{ind}(v_4)N\epsilon),\\ &\qquad (1+\eta, 1-\mathrm{ind}(v_4)N\epsilon)\rangle.\end{aligned}$$

Shifted Equality Gadget for V_3 and V_4. We define $G'(v_3, v_4)$ similarly to $F'(v_1, v_2)$, i.e., we shift the curves of G by $N^2\epsilon$, but in contrast to F' we shift it in the second dimension. More formally:

$$\pi_{G'}(v_3) := \pi_G(v_3) + (0, N^2\epsilon),$$
$$\sigma_{G'}(v_4) := \sigma_G(v_4) + (0, N^2\epsilon).$$

Due to the similar structure of the curve pairs of $F(v_1, v_2)$ and $F'(v_1, v_2), G(v_3, v_4), G'(v_3, v_4)$, analogous statements to Lemma 5.3 also hold for the curve pairs from $F'(v_1, v_2)$, $G(v_3, v_4)$, and $G'(v_3, v_4)$. We now show that all subcurves of different equality gadgets are pairwise further apart than δ.

LEMMA 5.4. *For any vectors $v_1 \in V_1, \ldots, v_4 \in V_4$ and any translation $\tau \in \mathcal{T}$, each π-subcurve from any of $F(v_1, v_2), F'(v_1, v_2), G(v_3, v_4), G'(v_3, v_4)$ is in discrete Fréchet distance greater than δ from each σ-subcurve of any equality gadget of different type.*

Proof. We first show that this holds for F and F'. Consider the first point of $\sigma_{F(\cdot,\cdot)}$ which we call q. This point is further than $2+N^2\epsilon$ from both points of $\pi_{F'(\cdot,\cdot)}$. When translating σ with τ, the distance is still greater than $2+\frac{3}{4}\epsilon$. Thus, $\sigma_{F(\cdot,\cdot)}$ and $\pi_{F'(\cdot,\cdot)}$ are in discrete Fréchet distance greater than δ for any valid τ. Now let p be the second point of $\pi_{F(\cdot,\cdot)}$. The point p has distance greater than $2+\epsilon$ from $\sigma_{F'(\cdot,\cdot)}$. With translation τ this distance is still greater than $2+\frac{3}{4}\epsilon$ and thus $\pi_{F(\cdot,\cdot)}$ and $\sigma_{F'(\cdot,\cdot)}$ are in discrete Fréchet distance greater than δ for any valid τ. The proof for G and G' is symmetric.

Now we prove the lemma for F and G, G'. First note that every point of F is in distance $1+\eta$ of the first coordinate axis and every point of G, G' is in distance $1+\eta$ of the second coordinate axis. Additionally, no point of F is closer than $1-2N^2\epsilon$ to the second coordinate axis while no point of G, G' is closer than $1-2N^2\epsilon$ to the first coordinate axis. This means that every point of a π-curve of F is in distance at least $2+\eta-2N^2\epsilon = 2+N^2\epsilon$ of any point of a σ-curve of G or G'. Even with translation this distance is at least $2+\frac{3}{4}\epsilon \ge \delta$. Thus, also the discrete Fréchet distance is greater than δ. The proof for F' is similar and the proofs for G and G' are symmetric.

We moreover observe that our equality gadgets lie in very restricted regions. Specifically, call a curve *diagonal* if all of its vertices are in

$$[-1-2\eta, -1+2\eta]^2 \ \cup\ [1-2\eta, 1+2\eta]^2,$$

and call it *anti-diagonal* if all of its vertices are in

$$\begin{aligned}&[-1-2\eta, -1+2\eta] \times [1-2\eta, 1+2\eta] \cup\\ &\qquad [1-2\eta, 1+2\eta] \times [-1-2\eta, -1+2\eta].\end{aligned}$$

OBSERVATION 5.1. *For all gadgets $F(v_1, v_2)$, $F'(v_1, v_2)$, $G(v_3, v_4)$, $G'(v_3, v_4)$ the σ-parts are diagonal while the π-parts are anti-diagonal.*

We are now ready to describe the last gadget. For proving its correctness, we will essentially only use the diagonal and anti-diagonal property of the curves.

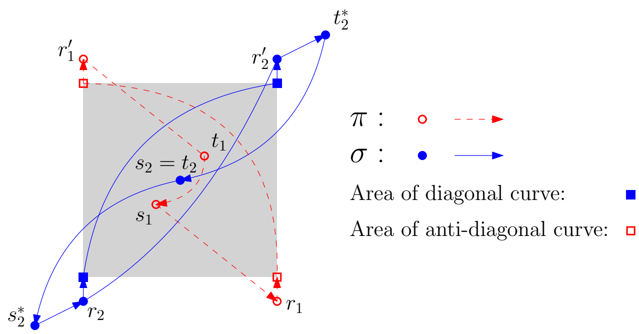

Figure 8: The OR gadget for general diagonal and anti-diagonal curves. The gray square is the centered square with diameter 2.

OR Gadget. We construct an OR gadget over diagonal and anti-diagonal curves which we will later apply to equality gadgets. Before introducing the gadget itself, we define various auxiliary points whose meaning will become clear later. Here we keep notation close to [Bri14], although the details of our construction are quite different.

$$s_1 := \left(-\tfrac{1}{4}, -\tfrac{1}{4}\right), \; t_1 := \left(\tfrac{1}{4}, \tfrac{1}{4}\right),$$
$$r_1 := \left(\tfrac{99}{100}, -\tfrac{5}{4}\right), \; r_1' := \left(-\tfrac{99}{100}, \tfrac{5}{4}\right),$$
$$s_2 := (0,0), \; s_2^* := \left(-\tfrac{3}{2}, -\tfrac{3}{2}\right), \; t_2^* := \left(\tfrac{3}{2}, \tfrac{3}{2}\right),$$
$$t_2 := (0,0), \; r_2 := \left(-\tfrac{99}{100}, -\tfrac{5}{4}\right), \; r_2' := \left(\tfrac{99}{100}, \tfrac{5}{4}\right).$$

Now, given diagonal curves $\hat{\sigma}^1, \dots, \hat{\sigma}^\ell$ and anti-diagonal curves $\hat{\pi}^1, \dots, \hat{\pi}^k$, we define the two curves of the OR gadget as

$$\pi_{\mathrm{OR}} := \bigcirc_{i \in [k]} s_1 \circ r_1 \circ \hat{\pi}^i \circ r_1' \circ t_1,$$
$$\sigma_{\mathrm{OR}} := s_2 \circ s_2^* \circ \Big(\bigcirc_{j \in [\ell]} r_2 \circ \hat{\sigma}^j \circ r_2' \Big) \circ t_2^* \circ t_2.$$

See Figure 8 for a visualization. Now let us prove correctness of the gadget.

LEMMA 5.5. *Given an OR gadget over diagonal curves $\hat{\sigma}^1, \dots, \hat{\sigma}^\ell$ and anti-diagonal curves $\hat{\pi}^1, \dots, \hat{\pi}^k$, for any translation $\tau \in \mathcal{T}$ we have $\delta_F(\pi_{\mathrm{OR}}, \sigma_{\mathrm{OR}} + \tau) \le \delta$ if and only if $\delta_F(\hat{\pi}^i, \hat{\sigma}^j + \tau) \le \delta$ for some i, j.*

Proof. First observe that no two auxiliary points have distance close to 2 and thus the translation τ does not change whether auxiliary points are closer than $\delta = 2 + \frac{1}{4}\epsilon$ or not. Thus, we can ignore the translation for distances between auxiliary points in this proof.

We first show that if $\delta_F(\hat{\pi}^i, \hat{\sigma}^j + \tau) \le \delta$ for some i, j, then $\delta_F(\pi_{\mathrm{OR}}, \sigma_{\mathrm{OR}} + \tau) \le \delta$ by giving a valid traversal. We start in s_1, s_2. Then we traverse π_{OR} until the copy of s_1 which comes before the subcurve $\hat{\pi}^i$. While staying in s_1, we traverse σ_{OR} until we reach the copy of r_2 right before the subcurve $\hat{\sigma}^j$. We then do one step on π_{OR} to r_1. Now we step to the first nodes of $\hat{\pi}^i$ and $\hat{\sigma}^j$ simultaneously, and then traverse these two subcurves in distance δ, which is possible due to $\delta_F(\hat{\pi}^i, \hat{\sigma}^j + \tau) \le \delta$. We then step to the copies of r_1' and r_2' simultaneously. We then step to t_1 on π_{OR}, while staying at r_2' at σ_{OR}. Subsequently, while staying in t_1, we traverse σ_{OR} until we reach its last point, namely t_2. Now we can traverse the remainder of π_{OR}. One can check that this traversal stays within distance δ.

We now show that if $\delta_F(\pi_{\mathrm{OR}}, \sigma_{\mathrm{OR}} + \tau) \le \delta$, then there exist i, j such that $\delta_F(\hat{\pi}^i, \hat{\sigma}^j + \tau) \le \delta$. We prove this by reconstructing how a valid traversal, which exists due to $\delta_F(\pi_{\mathrm{OR}}, \sigma_{\mathrm{OR}} + \tau) \le \delta$, must have passed through π_{OR} and σ_{OR}. Consider the point when s_2^* is reached. At that point, we have to be in some copy of s_1 as this is the only type of node of π_{OR} which is in distance at most δ from s_2^*. Let $\hat{\pi}^i$ be the subcurve right after this copy of s_1. When we step to the copy of r_1 right after this s_1, there are only three types of nodes from σ_{OR} in distance δ: s_2, t_2, r_2. Note that we already passed s_2, and we cannot have reached t_2 yet, as t_2^* is neither in reach of s_1 nor r_1. Thus, we are in r_2. Let the curve right after r_2 be $\hat{\sigma}^j$. The only option now is to do a simultaneous step to the first nodes of $\hat{\pi}^i$ and $\hat{\sigma}^j$. Now, consider the point when either r_1' or r_2' is first reached. All points of $\hat{\pi}^i$ are far from r_2' and all points of $\hat{\sigma}^j$ are far from r_1' and thus we have to be in r_1' and r_2' at the same time. This implies that we traversed $\hat{\pi}^i$ and $\hat{\sigma}^j$ from the start to the end nodes in distance δ and therefore $\delta_F(\hat{\pi}^i, \hat{\sigma}^j + \tau) \le \delta$.

Assembling $\boldsymbol{\pi^{(j)}}$ and $\boldsymbol{\sigma^{(j)}}$. Now we can apply the OR gadget to the equality gadgets in the following way. For each of the D dimensions we construct an OR gadget. The OR gadget for dimension $j \in [D]$ contains as anti-diagonal curves all $\pi_F(v_1)$ with $v_1[j] = 0$, all $\pi_{F'}(v_1)$, all $\pi_G(v_3)$ with $v_3[j] = 0$, and all $\pi_{G'}(v_3)$; and as diagonal curves it contains all $\sigma_F(v_2)$, all $\sigma_{F'}(v_2)$ with $v_2[j] = 0$, all $\sigma_G(v_4)$, and all $\sigma_{G'}(v_4)$ with $v_4[j] = 0$. By Observation 5.1 these curves are suited for the OR gadget. We denote the resulting curves by $\pi^{(j)}$ and $\sigma^{(j)}$, and we write $H(j) = (\pi^{(j)}, \sigma^{(j)})$. This yields the following lemma.

LEMMA 5.6. *Given a 4-OV instance $V_1, \dots, V_4$, and consider the corresponding OR gadget $H(j) = (\pi^{(j)}, \sigma^{(j)})$ for some $j \in [D]$. It holds that:*

(i) For any vectors $v_1 \in V_1, \dots, v_4 \in V_4$ with $v_1[j] \cdot v_2[j] \cdot v_3[j] \cdot v_4[j] = 0$ we have $\delta_F(\pi^{(j)}, \sigma^{(j)} + \tau) \le \delta$ for $\tau = ((\mathrm{ind}(v_1) + \mathrm{ind}(v_2) \cdot N) \cdot \epsilon, (\mathrm{ind}(v_3) + \mathrm{ind}(v_4) \cdot N) \cdot \epsilon)$.

(ii) If $\delta_F(\pi^{(j)}, \sigma^{(j)} + \tau) \leq \delta$ *for some* $\tau \in \mathcal{T}$*, then*

$$(\exists v_1 \in V_1, v_2 \in V_2 : v_1[j] \cdot v_2[j] = 0 \quad \textit{and}$$
$$|\epsilon \cdot (\mathrm{ind}(v_1) + \mathrm{ind}(v_2) \cdot N) - \tau_1| \leq \frac{1}{3}\epsilon)$$

or

$$(\exists v_3 \in V_3, v_4 \in V_4 : v_3[j] \cdot v_4[j] = 0 \quad \textit{and}$$
$$|\epsilon \cdot (\mathrm{ind}(v_3) + \mathrm{ind}(v_4) \cdot N) - \tau_2| \leq \frac{1}{3}\epsilon).$$

Proof. For (i), from $v_1[j] \cdot v_2[j] \cdot v_3[j] \cdot v_4[j] = 0$ it follows that at least one gadget of $F(v_1, v_2)$, $F'(v_1, v_2)$, $G(v_3, v_4)$, $G'(v_3, v_4)$ is contained in $H(j)$. By Lemma 5.3 and its analogous versions, we know that the discrete Fréchet distance of this gadget is small. By Lemma 5.5 it then follows that $\delta_F(\pi^{(j)}, \sigma^{(j)} + \tau) \leq \delta$.

For (ii), from $\delta_F(\pi^{(j)}, \sigma^{(j)} + \tau) \leq \delta$ it follows by Lemmas 5.5 and 5.4 that there exists a gadget Γ for which the discrete Fréchet distance is at most δ. From Lemma 5.3 and its analogous versions it follows that

$$|\epsilon \cdot (\mathrm{ind}(v_1) + \mathrm{ind}(v_2) \cdot N) - \tau_1| \leq \frac{1}{3}\epsilon \quad \text{or}$$
$$|\epsilon \cdot (\mathrm{ind}(v_3) + \mathrm{ind}(v_4) \cdot N) - \tau_2| \leq \frac{1}{3}\epsilon.$$

for some vectors $v_1 \in V_1, \ldots, v_4 \in V_4$. As Γ is contained in the OR gadget, we additionally have that $v_1[j] \cdot v_2[j] = 0$ or $v_3[j] \cdot v_4[j] = 0$, respectively.

Final Curves. The final curves π and σ are now defined as follows. We start with the translation gadget $\pi^{(0)}$ ($\sigma^{(0)}$). Then the curves $\pi^{(j)}$ ($\sigma^{(j)}$) follow for $j \in [D]$. Note that we have to translate these curves to fulfill the requirements of Lemmas 5.1 and 5.2, thus, we translate $\pi^{(j)}$ ($\sigma^{(j)}$) by $(100 \cdot j, 0)$.

We are now ready to prove Theorem 1.2. We split the proof into Lemma 5.7 and Lemma 5.8 which together imply Theorem 1.2.

Lemma 5.7. *Given a YES-instance of 4-OV, the curves* π *and* σ *constructed in our reduction have discrete Fréchet distance under translation at most* δ*, i.e.* $\min_\tau \delta_F(\pi, \sigma + \tau) \leq \delta$.

Proof. Let $v_1 \in V_1, \ldots, v_4 \in V_4$ be orthogonal vectors and let $\tau = ((\mathrm{ind}(v_1) + \mathrm{ind}(v_2) \cdot N) \cdot \epsilon, (\mathrm{ind}(v_3) + \mathrm{ind}(v_4) \cdot N) \cdot \epsilon)$ be the corresponding translation to those vectors. From Lemma 5.1 we know that $\delta_F(\pi^{(0)}, \sigma^{(0)} + \tau) \leq \delta$, and thus there is a valid traversal to the endpoints of the translation gadget. Then we simultaneously step to the start of $\pi^{(1)}$ and $\sigma^{(1)}$. From Lemma 5.6 we know that there also exist traversals of $\pi^{(1)}, \ldots, \pi^{(D)}$ and $\sigma^{(1)} + \tau, \ldots, \sigma^{(D)} + \tau$ of distance at most δ. It follows from Lemma 5.2 that we can also traverse those gadgets sequentially in distance δ and thus $\delta_F(\pi, \sigma + \tau) \leq \delta$.

Lemma 5.8. *If the curves* π *and* σ *constructed in our reduction have discrete Fréchet distance under translation at most* δ*, then the given 4-OV instance is a YES-instance.*

Proof. Let τ be a translation such that $\delta_F(\pi, \sigma + \tau) \leq \delta$. We know from Lemma 5.1 that $\tau \in \mathcal{T}$. Furthermore, from Lemma 5.2 we know that for all $j \in [D]$ it holds that $\delta_F(\pi^{(j)}, \sigma^{(j)} + \tau) \leq \delta$. It follows from Lemma 5.6 that for every $j \in [D]$ there exist $v_1 \in V_1, v_2 \in V_2$ such that $v_1[j] \cdot v_2[j] = 0$ and $|\epsilon \cdot (\mathrm{ind}(v_1) + \mathrm{ind}(v_2) \cdot N) - \tau_1| \leq \frac{1}{3}\epsilon$ or there exist $v_3 \in V_3, v_4 \in V_4$ such that $v_3[j] \cdot v_4[j] = 0$ and $|\epsilon \cdot (\mathrm{ind}(v_3) + \mathrm{ind}(v_4) \cdot N) - \tau_2| \leq \frac{1}{3}\epsilon$. Therefore, every dimension $j \in [D]$ gives us constraints on either v_1, v_2 or v_3, v_4. Note that those constraints have to be consistent. If in total this gives us constraints for $v_1, \ldots, v_4$, then we are done. Otherwise, if this only gives us constraints for v_1, v_2, then we already found v_1, v_2 which are orthogonal and thus we can pick arbitrary $v_3 \in V_3, v_4 \in V_4$ to obtain an orthogonal set of vectors. The case of v_3, v_4 is symmetric.

Proof. [Proof of Theorem 1.2.] SETH implies the k-OV hypothesis. The reduction above from a 4-OV instance of size N over $\{0,1\}^D$ to an instance of the discrete Fréchet distance under translation in $\mathbb{R}^2$ results in two curves of length $\mathcal{O}(D \cdot N)$. Lemmas 5.7 and 5.8 show correctness of this reduction. Hence, any $\mathcal{O}(n^{4-\epsilon})$-time algorithm for the discrete Fréchet distance under translation would imply an algorithm for 4-OV in time $\mathcal{O}((D \cdot N)^{4-\epsilon}) = \mathcal{O}(\mathrm{poly}(D) \cdot N^{4-\epsilon})$, refuting the k-OV hypothesis.

References

[AAKS13] P. K. Agarwal, R. Ben Avraham, H. Kaplan, and M. Sharir. Computing the discrete fréchet distance in subquadratic time. In *SODA*, pages 156–167. SIAM, 2013.

[AB10] H. Alt and M. Buchin. Can we compute the similarity between surfaces? *Discrete & Computational Geometry*, 43(1):78–99, 2010.

[AB18] A. Abboud and K. Bringmann. Tighter connections between formula-sat and shaving logs. In *ICALP*, volume 107 of *LIPIcs*, pages 8:1–8:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

[ACG12] I. Abraham, S. Chechik, and C. Gavoille. Fully dynamic approximate distance oracles for planar graphs via forbidden-set distance labels. In *STOC*, pages 1199–1218. ACM, 2012.

[AD16] A. Abboud and S. Dahlgaard. Popular conjectures as a barrier for dynamic planar graph algorithms. In *FOCS*, pages 477–486. IEEE Computer Society, 2016.

[AFK$^+$15] R. Ben Avraham, O. Filtser, H. Kaplan, M. J. Katz, and M. Sharir. The discrete and semicontinuous fréchet distance with shortcuts via approximate distance counting and selection. *ACM Trans. Algorithms*, 11(4):29:1–29:29, 2015.

[AG95] H. Alt and M. Godau. Computing the fréchet distance between two polygonal curves. *Int. J. Comput. Geometry Appl.*, 5:75–91, 1995.

[AKS15] R. Ben Avraham, H. Kaplan, and M. Sharir. A faster algorithm for the discrete fréchet distance under translation. *CoRR*, abs/1501.03724, 2015.

[AKW01] H. Alt, C. Knauer, and C. Wenk. Matching polygonal curves with respect to the fréchet distance. In *STACS*, volume 2010 of *Lecture Notes in Computer Science*, pages 63–74. Springer, 2001.

[BBG$^+$11] K. Buchin, M. Buchin, J. Gudmundsson, M. Löffler, and J. Luo. Detecting commuting patterns by clustering subtrajectories. *Int. J. Comput. Geometry Appl.*, 21(3):253–282, 2011.

[BBMM14] K. Buchin, M. Buchin, W. Meulemans, and W. Mulzer. Four soviets walk the dog - with an application to alt's conjecture. In *SODA*, pages 1399–1413. SIAM, 2014.

[BBW09] K. Buchin, M. Buchin, and Y. Wang. Exact algorithms for partial curve matching via the fréchet distance. In *SODA*, pages 645–654. SIAM, 2009.

[BKN18] K. Bringmann, M. Künnemann, and A. Nusser. Fréchet Distance Under Translation: Conditional Hardness and an Algorithm via Offline Dynamic Grid Reachability. *CoRR*, abs/1810.10982, 2018.

[BPSW05] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On map-matching vehicle tracking data. In *VLDB*, pages 853–864. ACM, 2005.

[Bri14] K. Bringmann. Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless SETH fails. In *FOCS*, pages 661–670. IEEE Computer Society, 2014.

[CdVE$^+$10] E. W. Chambers, É. Colin de Verdière, J. Erickson, S. Lazard, F. Lazarus, and S. Thite. Homotopic fréchet distance between curves or, walking your dog in the woods in polynomial time. *Comput. Geom.*, 43(3):295–311, 2010.

[Cha88] B. Chazelle. A functional approach to data structures and its use in multidimensional searching. *SIAM J. Comput.*, 17(3):427–462, 1988.

[CT17] T. M. Chan and K. Tsakalidis. Dynamic orthogonal range searching on the ram, revisited. In *Symposium on Computational Geometry*, volume 77 of *LIPIcs*, pages 28:1–28:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[CW10] A. F. Cook and C. Wenk. Geodesic fréchet distance inside a simple polygon. *ACM Trans. Algorithms*, 7(1):9:1–9:19, 2010.

[DH13] A. Driemel and S. Har-Peled. Jaywalking your dog: Computing the fréchet distance with shortcuts. *SIAM J. Comput.*, 42(5):1830–1866, 2013.

[DS07] K. Diks and P. Sankowski. Dynamic plane transitive closure. In *ESA*, volume 4698 of *Lecture Notes in Computer Science*, pages 594–604. Springer, 2007.

[EIV01] A. Efrat, P. Indyk, and S. Venkatasubramanian. Pattern matching for sets of segments. In *SODA*, pages 295–304. ACM/SIAM, 2001.

[EM94] T. Eiter and H. Mannila. Computing discrete Fréchet distance. Technical Report CD-TR 94/64, Christian Doppler Laboratory for Expert Systems, TU Vienna, Austria, 1994.

[GBT84] H. N. Gabow, J. L. Bentley, and R. E. Tarjan. Scaling and related techniques for geometry problems. In *STOC*, pages 135–143. ACM, 1984.

[IKLS17] G. F. Italiano, A. Karczmarz, J. Lacki, and P. Sankowski. Decremental single-source reachability in planar digraphs. In *STOC*, pages 1108–1121. ACM, 2017.

[Ind02] P. Indyk. Approximate nearest neighbor algorithms for frechet distance via product metrics. In *Symposium on Computational Geometry*, pages 102–106. ACM, 2002.

[INSW11] G. F. Italiano, Y. Nussbaum, P. Sankowski, and C. Wulff-Nilsen. Improved algorithms for min cut and max flow in undirected planar graphs. In *STOC*, pages 313–322. ACM, 2011.

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

[JXZ08] M. Jiang, Y. Xu, and B. Zhu. Protein structure-structure alignment with discrete fréchet distance. *J. Bioinformatics and Computational Biology*, 6(1):51–64, 2008.

[KS98] P. N. Klein and S. Subramanian. A fully dynamic approximation scheme for shortest paths in planar graphs. *Algorithmica*, 22(3):235–249, 1998.

[MC05] A. Mosig and M. Clausen. Approximately matching polygonal curves with respect to the fre'chet distance. *Comput. Geom.*, 30(2):113–127, 2005.

[Mor06] C. W. Mortensen. Fully dynamic orthogonal range reporting on RAM. *SIAM J. Comput.*, 35(6):1494–1525, 2006.

[MP99] M. E. Munich and P. Perona. Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification. In *ICCV*, pages 108–115, 1999.

[MSSZ11] A. Maheshwari, J. Sack, K. Shahbaz, and H. Zarrabi-Zadeh. Fréchet distance with speed limits. *Comput. Geom.*, 44(2):110–120, 2011.

[Sub93] S. Subramanian. A fully dynamic data structure for reachability in planar digraphs. In *ESA*, volume 726 of *Lecture Notes in Computer Science*, pages 372–383. Springer, 1993.

[Wen03] C. Wenk. *Shape matching in higher dimensions*. PhD thesis, Free University of Berlin, Dahlem, Germany, 2003.

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