

A Constant-Factor Approximation for Continuous Dynamic Time Warping in 2D

Kevin Buchin ✉ 

Technical University Dortmund, Germany

Maike Buchin ✉ 

Ruhr University Bochum, Germany

Jan Erik Swiadek ✉ 

Ruhr University Bochum, Germany

Sampson Wong ✉ 

University of Copenhagen, Denmark

Abstract

Continuous Dynamic Time Warping (CDTW) is a robust similarity measure for polygonal curves that has recently found a variety of applications. Despite its practical use, not much is known about the algorithmic complexity of computing it in 2D, especially when one requires either an exact solution or strong approximation guarantees. We fill this gap by introducing a 5-approximation algorithm with running time $O(n^5)$ under the 1-norm.

This is the first constant-factor approximation for 2D CDTW with polynomial running time. We extend our algorithm to all polygonal norms on $\mathbb{R}^2$, which we subsequently use in order to achieve a $(5 + \varepsilon)$ -approximation with time complexity $O(n^5/\varepsilon^{1/2})$ for CDTW in 2D under any fixed norm. The latter result in particular includes the usual Euclidean 2-norm.

2012 ACM Subject Classification Theory of computation $\rightarrow$ Computational geometry; Theory of computation $\rightarrow$ Approximation algorithms analysis

Keywords and phrases Continuous Dynamic Time Warping, Curve Similarity, Geometric Approximation Algorithm

Digital Object Identifier 10.4230/LIPIcs.ICALP.2026.47

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version:* <https://arxiv.org/abs/2605.05917>

Funding *Sampson Wong:* Supported by the European Union’s Marie Skłodowska-Curie Actions Postdoctoral Fellowship, grant no. 101146276.

1 Introduction

There exist several desirable properties for curve similarity measures [22, 23], two of which are tolerance to outliers and robustness to different sampling rates. As the Fréchet distance fails to provide the former, whereas Dynamic Time Warping struggles with the latter, these popular measures have drawbacks for applications. Continuous Dynamic Time Warping (CDTW) is an alternative that combines both abovementioned properties by minimising a path integral of distances between continuously and monotonically matched points on two polygonal curves. Our considered CDTW formulation [16, 15, 3, 7, 2, 5] originates from a summed version of the Fréchet distance introduced in [8]. There exist other related definitions [20, 21, 17, 1, 11, 14], and we refer to [7, Section 1] as well as [5, Section 1.1] for brief overviews of these.

On the practical side, the robustness of CDTW and its ability to yield high-quality solutions across different application domains have been observed in experiments [1, 11, 3]. Also, reasonable running times on realistic data have been reported [3, 2]. On the theoretical side,



© Kevin Buchin, Maike Buchin, Jan Erik Swiadek, and Sampson Wong;
licensed under Creative Commons License CC-BY 4.0

53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026).

Editors: Sayan Bhattacharya, Danupon Nanongkai, Michael Benedikt, and Gabriele Puppis;
Article No. 47; pp. 47:1–47:22



Leibniz International Proceedings in Informatics

LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



however, CDTW still lacks algorithms that guarantee both a good approximation factor and an efficient running time in 2D and beyond. Many existing approaches are approximations or heuristics that proceed via discretising the input curves [1, 11, 16, 15, 3, 14]. These include a generic additive approximation algorithm [15, 3], and a pseudo-polynomial-time $(1+\varepsilon)$ -approximation for 2D CDTW under the Euclidean 2-norm [16]. The latter has running time $O(\zeta^4 n^4 / \varepsilon^2 \cdot \log(\zeta n / \varepsilon))$, where n is the complexity and ζ is the spread of the given polygonal curves. That means the running time depends on the maximum ratio ζ of a pair of curve segment lengths. Such dependencies are typical for discretisation approaches.

A different type of approach, which enables exact computations, propagates functions arising from integration within a dynamic program [6, 15, 7, 2, 5]. This results in a polynomial running time of $O(n^5)$ for CDTW in 1D [7]. However, in 2D the algorithmic complexity of computing CDTW is as yet unknown. On the one hand, it has been shown that 2D CDTW cannot even be computed exactly under the 2-norm when restricted to algebraic operations [5]. On the other hand, function propagation approaches have only been made to work with polygonal norms like the 1-norm [2, 5]. These give piecewise linear integrands and piecewise quadratic cost functions similar to 1D. By converging to the unit disk, polygonal norms provide a $(1+\varepsilon)$ -approximation for CDTW under the 2-norm without a dependency on the curves' spread. However, the 2D dynamic program may involve more complicated propagation patterns [5, Section 4.2]. Those have so far prevented any 2D generalisation of the quite technical polynomial-time result that was achieved in 1D.

Our Contributions

We show that we can circumvent the 2D propagation patterns that potentially have an exponential complexity, in exchange for an approximation factor $\beta \leq 5$. This relies on a new bound for certain CDTW integrals, which we establish in Section 3.1. It is open whether the bound is tight, and any improvement to β would transfer to the following results. Applying this building block, we develop a CDTW algorithm that is outlined in Section 3.2 and then analysed in Section 4. After proving its approximation factor, we bound its time complexity by combining ideas from [7] with novel insights. For CDTW in $(\mathbb{R}^2, \|\cdot\|_1)$ we achieve a 5-approximation with running time $O(n^5)$. This is the first polynomial-time constant-factor result for 2D CDTW. We subsequently extend our algorithm from the 1-norm $\|\cdot\|_1$ to the class of polygonal norms. This is based on the framework of [5], which we generalise to facilitate approximations for not only the 2-norm $\|\cdot\|_2$ but also any other norm on $\mathbb{R}^2$. To that end, we exploit that any fixed norm $\|\cdot\|$ on $\mathbb{R}^d$ is $(1+\varepsilon)$ -approximated by a polyhedral norm of complexity $O(\varepsilon^{(1-d)/2})$. Here, fixed means that we treat the description complexity of $\|\cdot\|$, including the dimension d , as a constant. Putting things together, we obtain an approximation algorithm with factor $5+\varepsilon$ and running time $O(n^5/\varepsilon^{1/2})$ for 2D CDTW under any fixed norm and for any $\varepsilon > 0$. In the imbalanced case, where one of the given polygonal curves has some smaller complexity $m \in o(n)$, the term n^5 in our time bounds may be replaced by $n^4 m$. Note that the guaranteed bounds might be pessimistic. In particular, even if $\beta = 5$ is a tight bound for the approximated CDTW integrals, it is unclear whether there are any curves for which our algorithm approximates within factor close to β .

On a technical level, we expand the toolbox for CDTW algorithms by demonstrating that an integration-based building block and a tailored function propagation scheme can enable CDTW approximations without discretisations of any form. Combining our techniques with the approach of approximating the Euclidean 2-norm via polygonal norms yields polynomial-time computations for 2D CDTW under $\|\cdot\|_2$ without any significant loss of accuracy for the first time. Even though polygonal norms have already been used in [5], no attempt at a

fine-grained running time analysis was made there, after highlighting the obstacle that we deal with approximatively here. Furthermore, our running time analysis is more streamlined and insightful than that from [7]. We avoid a large case distinction by identifying core principles behind the complexity of function propagations, which generalise beyond some properties unique to 1D. In addition to advancing the understanding of propagation patterns, this enables an improvement upon previous algorithms: We compute lower envelopes of quadratic pieces in a single pass instead of multiple passes.

2 Preliminaries

A polygonal curve P in a normed real vector space $(\mathbb{R}^d, \|\cdot\|)$ consists of $n \in \mathbb{N}$ consecutive line segments induced by a sequence $\langle p_0, \dots, p_n \rangle$ of vertices, where $p_i \neq p_{i-1}$ for all $i \in \{1, \dots, n\}$. We write $P_{\leq i} := \langle p_0, \dots, p_i \rangle$, where $i \in \{0, \dots, n\}$, for the prefix subcurves of P . The following definitions are based on [5], which provides a robust CDTW formulation under any norm.

► **Definition 1** ([5, Definition 1]). *Let P, Q be two polygonal curves in $(\mathbb{R}^d, \|\cdot\|)$. The measure Continuous Dynamic Time Warping (CDTW) of P, Q under $\|\cdot\|$ is defined by*

$$\text{cdtw}_{\|\cdot\|}(P, Q) := \inf_{(f,g) \in \Pi_{[0,1]}(P) \times \Pi_{[0,1]}(Q)} \int_0^1 \|f(t) - g(t)\| \cdot \left\| \begin{pmatrix} \|f'(t)\| \\ \|g'(t)\| \end{pmatrix} \right\|_1 dt,$$

where the sets $\Pi_{[a,b]}(P)$ and $\Pi_{[a,b]}(Q)$ contain all piecewise continuously differentiable functions defined on the interval $[a, b]$ that monotonically parametrise P or Q respectively.

The arc length of P under $\|\cdot\|$ is $\|P\| := \sum_{i=1}^n \|p_i - p_{i-1}\|$. As the arc length is invariant to reparametrisation, we have $\int_a^b \|f'(t)\| dt = \|P\|$ for all $f \in \Pi_{[a,b]}(P)$. We denote the arc length parametrisation of P with constant speed 1 under $\|\cdot\|$ by $P_{\|\cdot\|}: [0, \|P\|] \rightarrow \mathbb{R}^d$.

► **Definition 2** ([5, Definitions 2 and 12]). *The parameter space of two polygonal curves P, Q under $\|\cdot\|$ is $[0, \|P\|] \times [0, \|Q\|]$. Each segment pair of $P = \langle p_0, \dots, p_n \rangle$ and $Q = \langle q_0, \dots, q_m \rangle$ is associated with a cell $C_{i,j} := [\|P_{\leq i-1}\|, \|P_{\leq i}\|] \times [\|Q_{\leq j-1}\|, \|Q_{\leq j}\|]$ in their parameter space, where $(i, j) \in \{1, \dots, n\} \times \{1, \dots, m\}$. The north/east/south/west border of $C_{i,j}$ parametrises along that side of $C_{i,j}$, e.g. its north border is $[\|P_{\leq i-1}\|, \|P_{\leq i}\|] \rightarrow \mathbb{R}^2, t \mapsto (t, \|Q_{\leq j}\|)^T$.*

The monotone matchings of P and Q correspond to monotone paths in parameter space. Let $\sigma := \|P\| + \|Q\|$, and let $\Gamma_{\|\cdot\|}(P, Q)$ be the set of functions $\gamma: [0, \sigma] \rightarrow [0, \|P\|] \times [0, \|Q\|]$ such that there are parametrisations $(f, g) \in \Pi_{[0,\sigma]}(P) \times \Pi_{[0,\sigma]}(Q)$ with $\gamma_1(s) = \int_0^s \|f'(t)\| dt$ and $\gamma_2(s) = \int_0^s \|g'(t)\| dt$ as well as $\|\gamma'(s)\|_1 = \|f'(s)\| + \|g'(s)\| = 1$ for all $s \in [0, \sigma]$.

► **Definition 3** ([5, Definition 3]). *Let $x, y \in [0, \|P\|] \times [0, \|Q\|]$ be two points with $x \preceq y$, i.e. $x_1 \leq y_1$ and $x_2 \leq y_2$, for polygonal curves P, Q . An (x, y) -path is a restriction $\hat{\gamma} := \gamma|_{[\|x\|_1, \|y\|_1]}$ of any function $\gamma \in \Gamma_{\|\cdot\|}(P, Q)$ with $\gamma(\|x\|_1) = x$ and $\gamma(\|y\|_1) = y$. It is called optimal for P, Q under $\|\cdot\|$ if it minimises $\text{cost}_{\|\cdot\|}(\hat{\gamma}) := \int_{\|x\|_1}^{\|y\|_1} \|P_{\|\cdot\|}(\hat{\gamma}_1(t)) - Q_{\|\cdot\|}(\hat{\gamma}_2(t))\| dt$ among all (x, y) -paths. We write $\text{opt}_{\|\cdot\|}(x, y)$ for the cost of such an optimal (x, y) -path.*

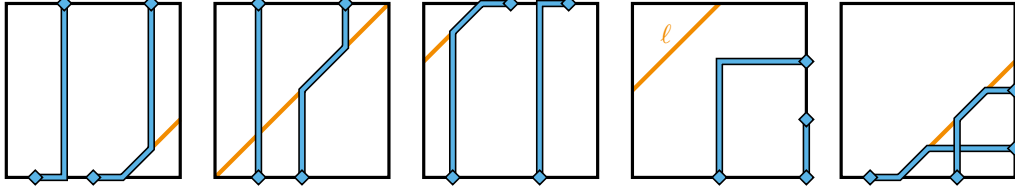
It is $\text{cdtw}_{\|\cdot\|}(P, Q) = \text{opt}_{\|\cdot\|}(\mathbf{0}, (\|P\|, \|Q\|)^T)$ by construction, where $\mathbf{0}$ denotes the origin. Given a cell border $\mathcal{B}: \text{dom}(\mathcal{B}) \rightarrow \mathbb{R}^2$, we define its *optimum function* $\text{opt}_{0,\mathcal{B}}: \text{dom}(\mathcal{B}) \rightarrow \mathbb{R}_{\geq 0}$ by $\text{opt}_{0,\mathcal{B}}(t) := \text{opt}_{\|\cdot\|}(\mathbf{0}, \mathcal{B}(t))$ for $t \in \text{dom}(\mathcal{B})$. In a cell under $\|\cdot\|$ we further define for each pair of south/west *input border* $\mathcal{A}$ and north/east *output border* $\mathcal{B}$ a function $\text{opt}_{\mathcal{A},\mathcal{B}}$ by

$$\text{opt}_{\mathcal{A},\mathcal{B}}(s, t) := \text{opt}_{\|\cdot\|}(\mathcal{A}(s), \mathcal{B}(t)) \quad \text{for } (s, t) \in \text{dom}(\mathcal{A}) \times \text{dom}(\mathcal{B}) \text{ with } \mathcal{A}(s) \preceq \mathcal{B}(t).$$

The optimal paths realising $\text{opt}_{\mathcal{A},\mathcal{B}}$ are characterised by Theorem 4, and possible resulting shapes of these optimal paths within different cells are illustrated in Figure 1.

► **Theorem 4** ([5, Section 2]). *Let P, Q be two polygonal curves in $(\mathbb{R}^2, \|\cdot\|)$, and let C be a parameter space cell. There exists a line $\ell \subseteq \mathbb{R}^2$ of positive slope such that for any choice of two points $x, y \in C$ with $x \preceq y$ the following (x, y) -path γ is optimal for P, Q under $\|\cdot\|$.*

- (a) *If ℓ intersects the bounding box $\Xi := [x_1, y_1] \times [x_2, y_2]$, then γ traces line segments from x to ξ_x to ξ_y to y , where $\xi_x, \xi_y \in \ell \cap \Xi$ share a coordinate with x, y respectively.*
- (b) *If ℓ does not intersect the bounding box Ξ , then γ traces line segments from x to ξ to y , where the single bending point $\xi \in \{(x_1, y_2)^\top, (y_1, x_2)^\top\}$ is the point closest to ℓ in Ξ .*



■ **Figure 1** Optimal paths from different cells' south borders to their output borders.

Note that Theorem 4 relies on the usage of norms in Definition 1: The equipped norm $\|\cdot\|$ measures both distances between curve points and speeds of curve parametrisations [5], while the 1-norm $\|\cdot\|_1$ combines these speeds within the CDTW path integral [8, 16, 7, 2, 5]. Also, Theorem 4 only holds in 1D and 2D. See the full version for a 3D counterexample.

3 Towards a Polynomial Bound

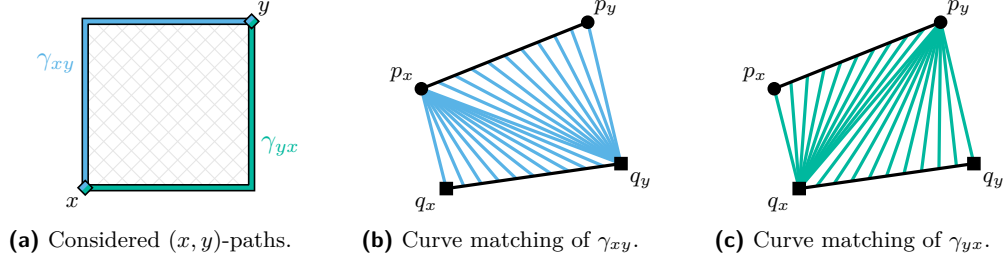
The idea behind function propagation approaches for exact CDTW algorithms is propagating costs of optimal paths through the parameter space cells and successively computing optimum functions in a dynamic program. After $\text{opt}_{0,\mathcal{A}}$ has been computed for each input border $\mathcal{A}$ of some cell C , one can use Theorem 4 to determine $\text{opt}_{0,\mathcal{B}}$ for each output border $\mathcal{B}$ of C . If the CDTW integrands are piecewise linear, as it is the case in 1D [7] or under polygonal norms in 2D such as the 1-norm or ∞ -norm [2, 5], then all optimum functions of borders are piecewise quadratic (cf. [5, Lemma 14]). The central challenge for the running time analysis lies in bounding the total number of quadratic pieces over all optimum functions [20, 17, 7, 5].

3.1 Building Block

In general, the possible propagation patterns and their complexity in terms of how the number of quadratic pieces is growing are still not well understood. The polynomial bound for the exact 1D algorithm relies heavily on the fact that the quadratic pieces on the output borders of a cell can be assigned monotonically to *parent* pieces on the cell's input borders (cf. [7, Lemma 13 and Definition 18]). However, this is insufficient on its own already in 2D under the 1-norm: Certain optimal paths originating from Theorem 4b may hypothetically cause an exponential growth of propagated pieces in 2D, even though there is a monotone assignment to parent pieces. In short, the obstacle for bounding the number of pieces occurs when the paths' bending points move freely within the cell. This yet unresolved issue was highlighted in [5, Section 4.2]. To tackle the absence of means for finding general tight bounds, our approach instead aims to ensure that the complexity created by such paths during the propagation of path costs in the dynamic program remains low. We henceforth call them *unhappy* paths within this paper.

► **Definition 5.** *Let γ be an optimal (x, y) -path as in Theorem 4. If each bending point of γ lies on ℓ or on the boundary of the cell C , then γ is happy w.r.t. ℓ . Else, γ is unhappy.*

See the fourth cell of Figure 1 for an unhappy path. As our first contribution, we show that the cost of any (unhappy) optimal (x, y) -path γ from Theorem 4b is approximated within factor at most 5 by choosing the worse of the two possible options for the single bending point ξ of γ . This changes whether a path first travels vertically and then horizontally or vice versa, as shown in Figure 2. Note that the paths from Theorem 4a are always happy.



■ **Figure 2** Approximable paths and their corresponding curve matchings.

Our technical proof of the following lemma relies – similar to Theorem 4 – on Definition 1 and Definition 3 employing the equipped norm $\|\cdot\|$ for distances between curve points as well as speeds and arc lengths on each curve. In particular, this allows for a versatile use of the triangle inequality and symmetry of the metric $(p, q) \mapsto \|p - q\|$ induced by $\|\cdot\|$.

► **Lemma 6.** *Let $x, y \in C$ be two points with $x \preceq y$, where C is a parameter space cell of polygonal curves P, Q in $(\mathbb{R}^d, \|\cdot\|)$. Consider the (x, y) -path γ_{xy} tracing line segments from x to $(x_1, y_2)^\top$ to y , and the (x, y) -path γ_{yx} tracing line segments from x to $(y_1, x_2)^\top$ to y . Their costs under $\|\cdot\|$ satisfy $1/5 \cdot \text{cost}_{\|\cdot\|}(\gamma_{xy}) \leq \text{cost}_{\|\cdot\|}(\gamma_{yx}) \leq 5 \cdot \text{cost}_{\|\cdot\|}(\gamma_{xy})$.*

Proof. We show only the second inequality, as the first one is entirely symmetric. To make the notation of this proof more concise, we write $p_s := P_{\|\cdot\|}(s) \in \mathbb{R}^d$ and $q_t := Q_{\|\cdot\|}(t) \in \mathbb{R}^d$ for parameters $(s, t) \in [x_1, y_1] \times [x_2, y_2]$ as well as $\overline{pq} := \|p - q\|$ for points $p, q \in \mathbb{R}^d$.

As γ_{xy} and γ_{yx} trace horizontal and vertical line segments, Definition 3 implies that their cost integrands on these respective segments correspond to $s \mapsto \overline{p_s q_t}$, where $s \in [x_1, y_1]$ varies and $t \in \{x_2, y_2\}$ is fixed, and to $t \mapsto \overline{p_s q_t}$, where $t \in [x_2, y_2]$ varies and $s \in \{x_1, y_1\}$ is fixed. We thus also use the shorthands $p_x := p_{x_1}$, $p_y := p_{y_1}$, $q_x := q_{x_2}$, and $q_y := q_{y_2}$. By applying the triangle inequality and symmetry that are induced by the norm $\|\cdot\|$, we then obtain

$$\begin{aligned}
 \text{cost}_{\|\cdot\|}(\gamma_{yx}) - \text{cost}_{\|\cdot\|}(\gamma_{xy}) &= \int_{x_1}^{y_1} \overline{p_s q_x} \, ds + \int_{x_2}^{y_2} \overline{p_y q_t} \, dt - \int_{x_2}^{y_2} \overline{p_x q_t} \, dt - \int_{x_1}^{y_1} \overline{p_s q_y} \, ds \\
 &= \int_{x_1}^{y_1} \overline{p_s q_x} - \overline{p_s q_y} \, ds + \int_{x_2}^{y_2} \overline{p_y q_t} - \overline{p_x q_t} \, dt \\
 &\leq \int_{x_1}^{y_1} \overline{q_x q_y} \, ds + \int_{x_2}^{y_2} \overline{p_x p_y} \, dt = 2 \cdot \overline{p_x p_y} \cdot \overline{q_x q_y}.
 \end{aligned}$$

The final equality holds because the integrands do not depend on the variables s and t , while the integration intervals are of length $y_1 - x_1 = \overline{p_x p_y}$ and $y_2 - x_2 = \overline{q_x q_y}$ respectively. Note that we have $y_1 - x_1 = \overline{p_x p_y}$ due to $x, y \in C$ for the given cell C , which means that p_x and p_y lie on a single curve segment of P . We similarly have $y_2 - x_2 = \overline{q_x q_y}$. Intuitively, the cost difference between γ_{yx} and γ_{xy} is at most two times the bounding box area of x and y . Together with $0 \leq (\overline{p_x p_y} - \overline{q_x q_y})^2 = \overline{p_x p_y}^2 + \overline{q_x q_y}^2 - 2 \cdot \overline{p_x p_y} \cdot \overline{q_x q_y}$, we therefore arrive at

$$\text{cost}_{\|\cdot\|}(\gamma_{yx}) \leq \text{cost}_{\|\cdot\|}(\gamma_{xy}) + 2 \cdot \overline{p_x p_y} \cdot \overline{q_x q_y} \leq \text{cost}_{\|\cdot\|}(\gamma_{xy}) + \overline{p_x p_y}^2 + \overline{q_x q_y}^2,$$

so it remains to show $\overline{p_x p_y}^2 + \overline{q_x q_y}^2 \leq 4 \cdot \text{cost}_{\|\cdot\|}(\gamma_{xy})$ in order to get the claimed result.

We consider $\int_{x_2}^{y_2} \overline{p_x q_t} dt$, which is the first term of $\text{cost}_{\|\cdot\|}(\gamma_{xy})$, and pick a $\lambda \in [x_2, y_2]$ such that both $\overline{q_x q_\lambda} \leq \overline{p_x q_x}$ and $\overline{q_\lambda q_y} \leq \overline{p_x q_y}$ hold. This is feasible as follows: If $\overline{q_x q_y} \leq \overline{p_x q_x}$, then $\lambda := y_2$ with $q_\lambda = q_y$ works. Else, we have $\overline{q_x q_y} > \overline{p_x q_x}$ and can pick λ with $\overline{q_x q_\lambda} = \overline{p_x q_x}$, which implies $\overline{q_\lambda q_y} = \overline{q_x q_y} - \overline{q_x q_\lambda} = \overline{q_x q_y} - \overline{p_x q_x} \leq \overline{p_x q_y}$ via $x, y \in C$ and the triangle inequality.

We now split the interval $[x_2, \lambda]$ into $k \in \mathbb{N}$ subintervals, each of length $\frac{1}{k} \cdot (\lambda - x_2) = \frac{1}{k} \cdot \overline{q_x q_\lambda}$. For every value t on the i -th subinterval this gives $\overline{q_x q_t} \leq \frac{i}{k} \cdot \overline{q_x q_\lambda}$, so together we have

$$\overline{p_x q_t} \geq \overline{p_x q_x} - \overline{q_x q_t} \geq \overline{q_x q_\lambda} - \frac{i}{k} \cdot \overline{q_x q_\lambda} = \frac{k-i}{k} \cdot \overline{q_x q_\lambda} \quad \text{if } t - x_2 \in \left[\frac{i-1}{k} \cdot \overline{q_x q_\lambda}, \frac{i}{k} \cdot \overline{q_x q_\lambda} \right],$$

which bounds the integrand from below. Summing this over all $i \in \{1, \dots, k\}$ yields

$$\int_{x_2}^{\lambda} \overline{p_x q_t} dt \geq \frac{1}{k} \cdot \overline{q_x q_\lambda} \cdot \sum_{i=1}^k \frac{k-i}{k} \cdot \overline{q_x q_\lambda} = \frac{1}{k} \cdot \overline{q_x q_\lambda} \cdot \frac{(k-1) \cdot k}{2k} \cdot \overline{q_x q_\lambda} = \frac{k-1}{2k} \cdot \overline{q_x q_\lambda}^2.$$

Since we also have $\overline{q_\lambda q_y} \leq \overline{p_x q_y}$ by choice of λ , we similarly obtain $\int_{\lambda}^{y_2} \overline{p_x q_t} dt \geq \frac{k-1}{2k} \cdot \overline{q_\lambda q_y}^2$. Putting these together gives a lower bound for the first term of $\text{cost}_{\|\cdot\|}(\gamma_{xy})$. That means

$$\int_{x_2}^{y_2} \overline{p_x q_t} dt = \int_{x_2}^{\lambda} \overline{p_x q_t} dt + \int_{\lambda}^{y_2} \overline{p_x q_t} dt \geq \frac{k-1}{2k} \cdot (\overline{q_x q_\lambda}^2 + \overline{q_\lambda q_y}^2) \geq \frac{k-1}{4k} \cdot \overline{q_x q_y}^2$$

by utilising $2 \cdot (\overline{q_x q_\lambda}^2 + \overline{q_\lambda q_y}^2) = (\overline{q_x q_\lambda} + \overline{q_\lambda q_y})^2 + (\overline{q_x q_\lambda} - \overline{q_\lambda q_y})^2 \geq (\overline{q_x q_\lambda} + \overline{q_\lambda q_y})^2 = \overline{q_x q_y}^2$ in the final inequality. Proceeding analogously for the second term of $\text{cost}_{\|\cdot\|}(\gamma_{xy})$ results in

$$\text{cost}_{\|\cdot\|}(\gamma_{xy}) = \int_{x_2}^{y_2} \overline{p_x q_t} dt + \int_{x_1}^{y_1} \overline{p_s q_y} ds \geq \frac{k-1}{4k} \cdot (\overline{q_x q_y}^2 + \overline{p_x p_y}^2) \xrightarrow{k \rightarrow \infty} \frac{1}{4} \cdot (\overline{p_x p_y}^2 + \overline{q_x q_y}^2)$$

and hence $\overline{p_x p_y}^2 + \overline{q_x q_y}^2 \leq 4 \cdot \text{cost}_{\|\cdot\|}(\gamma_{xy})$ as required above, which completes the proof. ◀

► **Remark 7.** It remains open whether the factor of 5 is tight in general. The greatest lower bound that we know of occurs already in 1D: Consider $p_x := 0$, $p_y := \phi + 1$, $q_x := \phi$, and $q_y := 2\phi + 1$, where $\phi := (1 + \sqrt{5})/2$ is the golden ratio. This yields a factor of $2\phi + 1 \approx 4.236$. By analysing the trade-off between the inequalities in the above proof, one can likely improve the upper bound in 1D from 5 to $2\phi + 1$, but closing the gap for $d \geq 2$ seems difficult.

Lemma 6 for the first time allows CDTW approximations without discretisations of any form. Our approximation algorithm is based on the fact that at least one of the paths γ_{xy} and γ_{yx} is a happy path whenever both x and y lie on borders of the cell C . At first glance, it thus might even seem like simply omitting all unhappy paths in the propagation procedure of the exact algorithm from [5] yields a polynomial-time result for CDTW in 2D with approximation factor $\beta \leq 5$. However, this is not the case because minimum-cost happy paths lack an important property of optimal paths: One can always choose non-crossing optimal paths (cf. [7, Lemma 11]), whereas minimum-cost happy paths may cross, e.g. as depicted in the final cell of Figure 1. This prevents the previously mentioned monotone assignment of quadratic pieces. To achieve a polynomial bound, we will propagate some unhappy paths while still keeping the resulting complexity low.

3.2 Algorithm Outline

Having established one of the main building blocks, we now provide a conceptual outline of our approximation algorithm, which uses a propagation scheme tailored to Definition 5 and Lemma 6. Although the outline has no requirements on the norm $\|\cdot\|$, note that it

only seems feasible to computationally deal with the propagated cost functions if they are well-behaved, such as under polygonal norms. We defer the implementation details along with the analysis to Section 4. The dynamic program in Algorithm 1 visits all parameter space cells in layers, starting at $C_{1,1}$ and ending at $C_{n,m}$. For each cell C we store its north, east, south and west border as $C.\mathcal{N}$, $C.\mathcal{E}$, $C.\mathcal{S}$ and $C.\mathcal{W}$ respectively. An output border $\mathcal{B}$ of C knows its *adjoining* input border $\mathcal{B}.\text{adj}$ and its *opposing* input border $\mathcal{B}.\text{opp}$. That means $C.\mathcal{N}.\text{opp} := C.\mathcal{S} := C.\mathcal{E}.\text{adj}$, so the first three cells of Figure 1 show paths between opposing borders, and the final two cells show the adjoining case. Symmetrically, we thus have $C.\mathcal{N}.\text{adj} := C.\mathcal{W} := C.\mathcal{E}.\text{opp}$ as well. (Cf. [5, Definition 13].)

Our algorithm computes for every cell border $\mathcal{B}$ a cost function $\mathcal{B}.\text{apx}$ that 5-approximates the optimum function $\text{opt}_{0,\mathcal{B}}$. We also store a value $h_{i,j}$ for all $(i,j) \in \{0, \dots, n\} \times \{0, \dots, m\}$, which is set to the minimum computed cost from $\mathbf{0}$ to the north-east corner $(\|P_{\leq i}\|, \|Q_{\leq j}\|)^T$ of the cell $C_{i,j}$. The base case of the dynamic program in lines 1–2 is as usual: It initialises all costs given by paths that travel either straight horizontally or straight vertically from $\mathbf{0}$. Then the main loop initialises in lines 6–7 for each cell C the approximate cost functions of its output borders $C.\mathcal{N}$ and $C.\mathcal{E}$ via straight paths from the cell corners.

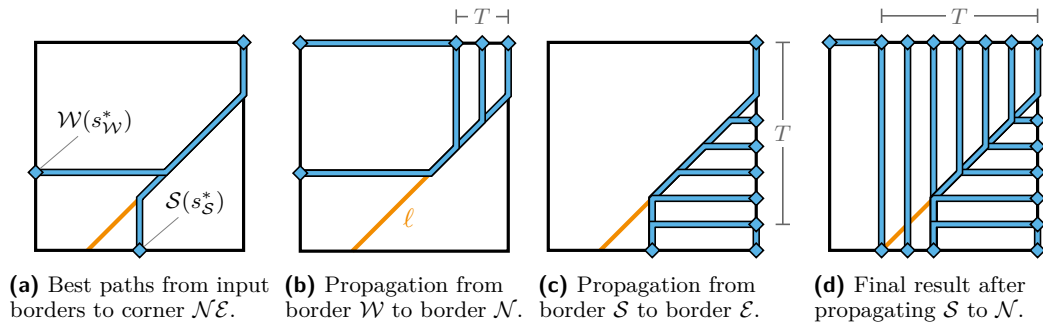
■ **Algorithm 1** A 5-approximation for CDTW of polygonal curves P, Q in $(\mathbb{R}^2, \|\cdot\|)$.

```

1. for  $i \leftarrow 1$  to  $n$  do  $C_{i,0}.\mathcal{N}.\text{apx} \leftarrow [t \mapsto \text{opt}_{\|\cdot\|}(\mathbf{0}, (t, 0)^T)]$ ;  $h_{i,0} \leftarrow C_{i,0}.\mathcal{N}.\text{apx}(\|P_{\leq i}\|)$ 
2. for  $j \leftarrow 1$  to  $m$  do  $C_{0,j}.\mathcal{E}.\text{apx} \leftarrow [t \mapsto \text{opt}_{\|\cdot\|}(\mathbf{0}, (0, t)^T)]$ ;  $h_{0,j} \leftarrow C_{0,j}.\mathcal{E}.\text{apx}(\|Q_{\leq j}\|)$ 
3. for  $l \leftarrow 2$  to  $n + m$  do
4.   foreach  $C \leftarrow C_{i,j}$  with  $i + j = l$  do // visit the layer- $l$  cells
5.      $C.\mathcal{S}.\text{apx} \leftarrow C_{i,j-1}.\mathcal{N}.\text{apx}$ ;  $C.\mathcal{W}.\text{apx} \leftarrow C_{i-1,j}.\mathcal{E}.\text{apx}$ 
6.      $C.\mathcal{N}.\text{apx} \leftarrow [t \mapsto h_{i-1,j} + \text{opt}_{C.\mathcal{W},C.\mathcal{N}}(\|Q_{\leq j}\|, t)]$  // propagate corner NW
7.      $C.\mathcal{E}.\text{apx} \leftarrow [t \mapsto h_{i,j-1} + \text{opt}_{C.\mathcal{S},C.\mathcal{E}}(\|P_{\leq i}\|, t)]$  // propagate corner SE
8.      $H_S \leftarrow \{C.\mathcal{S}.\text{apx}(s) + \text{opt}_{C.\mathcal{S},C.\mathcal{E}}(s, \|Q_{\leq j}\|) \mid s \in \text{dom}(C.\mathcal{S})\}$ 
9.      $H_W \leftarrow \{C.\mathcal{W}.\text{apx}(s) + \text{opt}_{C.\mathcal{W},C.\mathcal{N}}(s, \|P_{\leq i}\|) \mid s \in \text{dom}(C.\mathcal{W})\}$ 
10.     $C.\text{PROPAGATECOSTS}(\min H_S, \arg \min H_S, \min H_W, \arg \min H_W)$  // call subroutine
11.     $h_{i,j} \leftarrow \min H_S \cup H_W$  // store cost of corner NE
12. return  $h_{n,m}$ 

```

In the following, the costs of optimal paths within C are implicitly given by some choice of ℓ as in Theorem 4. Next, lines 8–11 of Algorithm 1 identify the best path from each of the two input borders to the north-east corner of C . See Figure 3a for a possible configuration. Our propagation scheme utilises these paths' costs and starting points, which are passed to the subroutine $C.\text{PROPAGATECOSTS}$. This is a difference to previous propagation-based CDTW algorithms [15, 7, 2, 5], although a related idea was stated in [6, Observation 3.3] within the context of the partial Fréchet similarity.



■ **Figure 3** Example of algorithm steps in case of $\min H_S = h_S^* < h_W^* = \min H_W$.

As outlined in Algorithm 2, every input border $\mathcal{A}$ of C is then propagated to its adjoining output border $\mathcal{B}$ using the subroutine $\mathcal{B}.\text{PROPAGATEFROMADJOINING}$, which updates the cost function $\mathcal{B}.\text{apx}$. It only propagates paths with the same starting point on $\mathcal{A}$ as its best path to the north-east corner, see Figure 3b. This may include unhappy paths with that starting point, e.g. as in Figure 3c, but it omits all other unhappy paths in contrast to exact algorithms. In a final step, at most one input border $\mathcal{A}$ is propagated to its opposing output border $\mathcal{B}$, namely if its best path to the north-east corner is better than the other one. Here, $\mathcal{B}.\text{PROPAGATEFROMOPPOSING}$ updates $\mathcal{B}.\text{apx}$ by computing a lower envelope over all points located on $\mathcal{A}$ up to its best path's starting point. See Figure 3d for an example result.

■ **Algorithm 2** Subroutines for propagating costs from input to output borders.

```

1. procedure  $C.\text{PROPAGATECOSTS}(h_S^*, s_S^*, h_W^*, s_W^*)$  // propagate...
2.    $C.\mathcal{N}.\text{PROPAGATEFROMADJOINING}(s_W^*)$  // ...W to N
3.    $C.\mathcal{E}.\text{PROPAGATEFROMADJOINING}(s_S^*)$  // ...S to E
4.   if  $h_S^* < h_W^*$  then  $C.\mathcal{N}.\text{PROPAGATEFROMOPPOSING}(s_S^*)$  // ...S to N
5.   if  $h_W^* < h_S^*$  then  $C.\mathcal{E}.\text{PROPAGATEFROMOPPOSING}(s_W^*)$  // ...W to E


---


6. procedure  $\mathcal{B}.\text{PROPAGATEFROMADJOINING}(s^*)$ 
7.    $\mathcal{A} \leftarrow \mathcal{B}.\text{adj}; \text{apx}_{s^*} \leftarrow [t \mapsto \mathcal{A}.\text{apx}(s^*) + \text{opt}_{\mathcal{A},\mathcal{B}}(s^*, t)]$  // use only  $s^*$ 
8.    $T \leftarrow \{t \in \text{dom}(\mathcal{B}) \mid \text{apx}_{s^*}(t) < \mathcal{B}.\text{apx}(t)\}; \mathcal{B}.\text{apx}|_T \leftarrow \text{apx}_{s^*}|_T$  // update costs


---


9. procedure  $\mathcal{B}.\text{PROPAGATEFROMOPPOSING}(s^*)$ 
10.   $\mathcal{A} \leftarrow \mathcal{B}.\text{opp}; \text{apx}_{\leq s^*} \leftarrow [t \mapsto \min_{s \leq s^* \wedge s \leq t} \mathcal{A}.\text{apx}(s) + \text{opt}_{\mathcal{A},\mathcal{B}}(s, t)]$  // use all  $s \leq s^*$ 
11.   $T \leftarrow \{t \in \text{dom}(\mathcal{B}) \mid \text{apx}_{\leq s^*}(t) < \mathcal{B}.\text{apx}(t)\}; \mathcal{B}.\text{apx}|_T \leftarrow \text{apx}_{\leq s^*}|_T$  // update costs

```

4 Algorithm Analysis and Specifics

In Section 4.1 we show that the above algorithm outline gives a 5-approximation for CDTW. Its implementation details and the resulting running time depend on the (polygonal) norm $\|\cdot\|$. We first consider $\|\cdot\|_1$ in Section 4.2 and afterwards extend our results in Section 4.3. Both the proof of approximation factor and the running time analysis build upon the fact that all approximate cost functions computed by our algorithm are continuous. To show that, one can apply a useful lemma, which says that converging paths have converging costs.

► **Lemma 8** ([5, Lemma 19]). *Let γ be an arbitrary (x, y) -path, and let $(\gamma_k)_{k \in \mathbb{N}}$ be a sequence of (x_k, y_k) -paths converging to γ , which means $\lim_{k \rightarrow \infty} (x_k, y_k) = (x, y)$ and $\lim_{k \rightarrow \infty} \gamma_k(\lambda) = \gamma(\lambda)$ for all $\lambda \in (\|x\|_1, \|y\|_1)$. The limit $\lim_{k \rightarrow \infty} \text{cost}_{\|\cdot\|}(\gamma_k)$ exists and is equal to $\text{cost}_{\|\cdot\|}(\gamma)$.*

► **Proposition 9.** *For every border $\mathcal{B}$ the function $\mathcal{B}.\text{apx}$ computed by our algorithm is continuous. Moreover, all $t, t' \in \text{dom}(\mathcal{B})$ with $t < t'$ satisfy $\mathcal{B}.\text{apx}(t) + \text{opt}_{\|\cdot\|}(\mathcal{B}(t), \mathcal{B}(t')) \geq \mathcal{B}.\text{apx}(t')$.*

The proof of Proposition 9 is deferred to the full version. Showing the continuity of $\mathcal{B}.\text{apx}$ through Lemma 8 is similar to the corresponding proof for the optimum function $\text{opt}_{0,\mathcal{B}}$ [5, Theorem 20]. The second property implies that travelling on a border does not improve the computed costs, which justifies the use of the corner costs $h_{i-1,j}, h_{i,j-1}$ in Algorithm 1.

4.1 Proof of Approximation Factor

We want to show that the cost of every optimal path from $\mathbf{0}$ to a cell border is 5-approximated. To that end, we give a formal and more general definition of best paths to a point. We also establish the existence of best paths and that they are closed under taking prefix paths.

► **Definition 10.** Let $\mathcal{A}$ be an input border of a cell C , and let $y \in C$ be some point. A best path from $\mathcal{A}$ to y is an optimal $(\mathcal{A}(s_y^*), y)$ -path γ_y^* such that $s_y^* \in \text{dom}(\mathcal{A})$ satisfies $\mathcal{A}(s_y^*) \preceq y$ and minimises the function $s \mapsto \mathcal{A}.\text{apx}(s) + \text{opt}_{\|\cdot\|}(\mathcal{A}(s), y)$. If the function value at s_y^* is less than the analogous minimum for the other input border, we call γ_y^* a dominating best path.

► **Lemma 11.** For each input border $\mathcal{A}$ of a cell C and each point $y \in C$ there is a best path γ_y^* from $\mathcal{A}$ to y . Now let $\mathcal{A}(s_y^*)$ with $s_y^* \in \text{dom}(\mathcal{A})$ be the starting point of γ_y^* , and let $\lambda \in [\lambda_y^*, \|y\|_1]$ be arbitrary, where $\lambda_y^* := \|\mathcal{A}(s_y^*)\|_1$. Then the prefix path $\gamma_y^*|_{[\lambda_y^*, \lambda]}$ is a best path from $\mathcal{A}$ to $\gamma_y^*(\lambda)$. If γ_y^* is dominating, we have that $\gamma_y^*|_{[\lambda_y^*, \lambda]}$ is dominating as well.

Proof. Consider the function $s \mapsto \mathcal{A}.\text{apx}(s) + \text{opt}_{\|\cdot\|}(\mathcal{A}(s), y)$ from Definition 10. By Proposition 9, $\mathcal{A}.\text{apx}$ is continuous. Lemma 8 and Theorem 4 imply that $s \mapsto \text{opt}_{\|\cdot\|}(\mathcal{A}(s), y)$ is also continuous. Basic calculus says that their sum is a continuous function attaining its minimum on the closed interval $\{s \in \text{dom}(\mathcal{A}) \mid \mathcal{A}(s) \preceq y\}$ at some s_y^* .

If any prefix path $\gamma_y^*|_{[\lambda_y^*, \lambda]}$ of an optimal $(\mathcal{A}(s_y^*), y)$ -path γ_y^* were not a best path, there would be a better path γ_λ from $\mathcal{A}$ to $\gamma_y^*(\lambda)$. Concatenating γ_λ with the suffix of γ_y^* would yield a better path from $\mathcal{A}$ to y than γ_y^* , a contradiction. Similarly, the existence of a non-dominating prefix path of γ_y^* would imply that γ_y^* is non-dominating. ◀

This enables us to show that for paths whose costs are not propagated by the subroutines from Algorithm 2 there are propagations of paths with better or approximating costs. In case of opposing borders, we always find better paths and thus do not require any approximations.

► **Proposition 12.** Let $\mathcal{B}$ be an output border of a cell C with opposing input border $\mathcal{A} := \mathcal{B}.\text{opp}$. It is $\mathcal{B}.\text{apx}(t) \leq \mathcal{A}.\text{apx}(s) + \text{opt}_{\mathcal{A}, \mathcal{B}}(s, t)$ for all $(s, t) \in \text{dom}(\mathcal{A}) \times \text{dom}(\mathcal{B})$ with $s \leq t$.

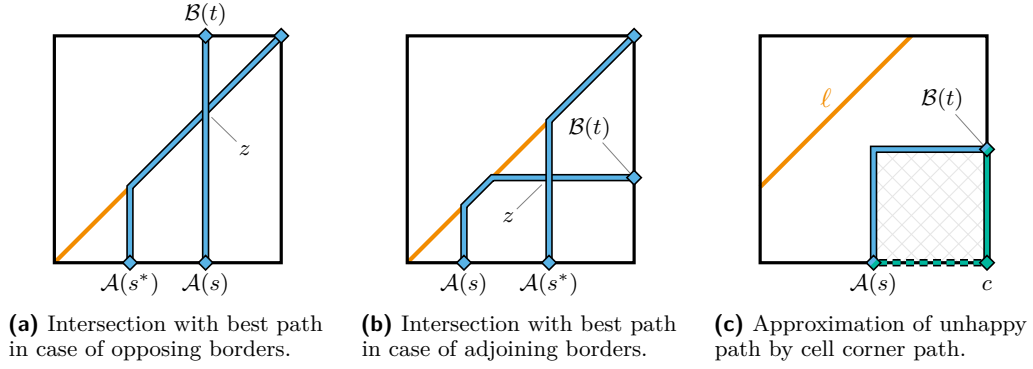
Proof. Let γ be an optimal $(\mathcal{A}(s), \mathcal{B}(t))$ -path. We next distinguish between whether the subroutine $\mathcal{B}.\text{PROPAGATEFROMOPPOSING}$ from Algorithm 2 is called for the given border $\mathcal{B}$ or not. Assume first that it is called. If $s \leq s^*$ holds, where $s^* := \arg \min H_{\mathcal{A}}$ is given by lines 8–10 of Algorithm 1, then this call yields $\mathcal{B}.\text{apx}(t) \leq \text{apx}_{\leq s^*}(t) \leq \mathcal{A}.\text{apx}(s) + \text{opt}_{\mathcal{A}, \mathcal{B}}(s, t)$. Else, we have $s > s^*$ and it follows that γ intersects a best path γ^* from $\mathcal{A}$ to the north-east corner of C in some point $z \in C$, see Figure 4a. Lemma 11 says $\mathcal{A}.\text{apx}(s^*) + \text{opt}_{\|\cdot\|}(\mathcal{A}(s^*), z) \leq \mathcal{A}.\text{apx}(s) + \text{opt}_{\|\cdot\|}(\mathcal{A}(s), z)$, so that the call yields

$$\mathcal{B}.\text{apx}(t) \leq \text{apx}_{\leq s^*}(t) \leq \mathcal{A}.\text{apx}(s^*) + \text{opt}_{\mathcal{A}, \mathcal{B}}(s^*, t) \leq \mathcal{A}.\text{apx}(s) + \text{opt}_{\mathcal{A}, \mathcal{B}}(s, t).$$

Assume now that $\mathcal{B}.\text{PROPAGATEFROMOPPOSING}$ is not called. Because of the decisions made in lines 4–5 of Algorithm 2, there necessarily exists a dominating best path γ_0^* from the other input border $\mathcal{A}_0 := \mathcal{B}.\text{adj}$ to the north-east corner of C . The path γ_0^* intersects all paths from $\mathcal{A}$ to $\mathcal{B}$, including γ . Let $z_0 \in C$ be an intersection point of these two paths, and let $\mathcal{A}_0(s_0^*)$ be the starting point of γ_0^* . Lemma 11 says $\mathcal{A}_0.\text{apx}(s_0^*) + \text{opt}_{\|\cdot\|}(\mathcal{A}_0(s_0^*), z_0) < \mathcal{A}.\text{apx}(s) + \text{opt}_{\|\cdot\|}(\mathcal{A}(s), z_0)$. Hence, the subroutine call $\mathcal{A}_0.\text{PROPAGATEFROMADJOINING}$, which occurs in lines 2–3 of Algorithm 2, now results in

$$\mathcal{B}.\text{apx}(t) \leq \text{apx}_{s_0^*}(t) = \mathcal{A}_0.\text{apx}(s_0^*) + \text{opt}_{\mathcal{A}_0, \mathcal{B}}(s_0^*, t) < \mathcal{A}.\text{apx}(s) + \text{opt}_{\mathcal{A}, \mathcal{B}}(s, t). \quad \blacktriangleleft$$

In case of adjoining borders, we get a similar result for happy paths, whereas unhappy paths might not intersect any propagated best paths. We can, however, use Lemma 6 to show that all unhappy paths are already approximated by the propagation of the cell corner that is shared by the adjoining borders.



■ **Figure 4** Different configurations yielding a path with at worst approximating cost.

► **Proposition 13.** *Let $\mathcal{B}$ be an output border of a cell C with adjoining input border $\mathcal{A} := \mathcal{B}.\text{adj}$. It is $\mathcal{B}.\text{apx}(t) \leq \mathcal{A}.\text{apx}(s) + 5 \cdot \text{opt}_{\mathcal{A},\mathcal{B}}(s, t)$ for all $(s, t) \in \text{dom}(\mathcal{A}) \times \text{dom}(\mathcal{B})$.*

Proof. Let γ be an optimal $(\mathcal{A}(s), \mathcal{B}(t))$ -path as in Theorem 4. Assume first that γ is of the form from Theorem 4a. This implies that γ intersects all paths from $\mathcal{A}$ to the north-east corner of C , either on ℓ or otherwise like in Figure 4b. Let γ^* be a best such path with starting point $\mathcal{A}(s^*)$, where $s^* := \arg \min H_{\mathcal{A}}$ is given by lines 8–10 of Algorithm 1, and let $z \in C$ be an intersection point of γ and γ^* . Lemma 11 says $\mathcal{A}.\text{apx}(s^*) + \text{opt}_{\|\cdot\|}(\mathcal{A}(s^*), z) \leq \mathcal{A}.\text{apx}(s) + \text{opt}_{\|\cdot\|}(\mathcal{A}(s), z)$. Thus, the call $\mathcal{B}.\text{PROPAGATEFROMADJOINING}$ from lines 2–3 of Algorithm 2 results in $\mathcal{B}.\text{apx}(t) \leq \text{apx}_{s^*}(t) = \mathcal{A}.\text{apx}(s^*) + \text{opt}_{\mathcal{A},\mathcal{B}}(s^*, t) \leq \mathcal{A}.\text{apx}(s) + \text{opt}_{\mathcal{A},\mathcal{B}}(s, t)$.

Assume now that γ is of the form from Theorem 4b. By Lemma 6, we then have

$$\mathcal{A}.\text{apx}(s) + 5 \cdot \text{opt}_{\mathcal{A},\mathcal{B}}(s, t) = \mathcal{A}.\text{apx}(s) + 5 \cdot \text{cost}_{\|\cdot\|}(\gamma) \geq \mathcal{A}.\text{apx}(s) + \text{cost}_{\|\cdot\|}(\gamma_c),$$

where γ_c is the $(\mathcal{A}(s), \mathcal{B}(t))$ -path through the shared cell corner $c = \mathcal{A}(s_c)$ of the adjoining borders $\mathcal{A}$ and $\mathcal{B}$, see Figure 4c. We obtain $\mathcal{A}.\text{apx}(s) + \text{cost}_{\|\cdot\|}(\gamma_c) \geq \mathcal{A}.\text{apx}(s_c) + \text{opt}_{\|\cdot\|}(c, \mathcal{B}(t)) \geq \mathcal{B}.\text{apx}(t)$ due to Proposition 9 and the initialisation of $\mathcal{B}.\text{apx}$ in lines 6–7 of Algorithm 1. ◀

We conclude that the conceptual algorithm outline from Section 3.2 indeed gives a CDTW approximation within factor $\beta \leq 5$. Using the above propositions, this follows by induction. See the full version for a proof of Theorem 14.

► **Theorem 14.** *Any implementation of Algorithm 1 and Algorithm 2 under a norm $\|\cdot\|$ returns a factor-5 approximation of $\text{cdtw}_{\|\cdot\|}(P, Q)$ for all polygonal curves P, Q in $(\mathbb{R}^2, \|\cdot\|)$.*

As discussed in Remark 7, we do not know whether the factor is tight. Even the value β that is tight for Lemma 6 might not be tight for Theorem 14. It would only transfer if there are curves P, Q with an optimal $(\mathbf{0}, (\|P\|, \|Q\|)^T)$ -path that is predominantly unhappy.

4.2 Running Time under the 1-Norm

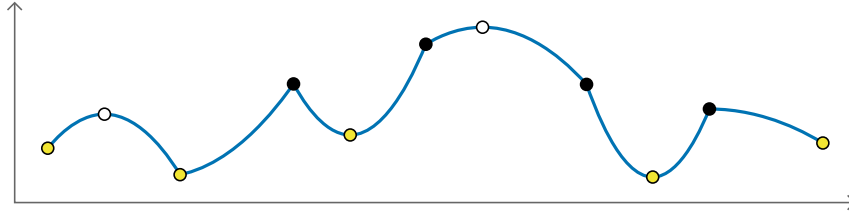
In this section we specify the details of how to implement our approximation algorithm under the 1-norm $\|\cdot\|_1$, and analyse its running time on two curves P, Q of complexity n and m respectively. By definition, we have $\|z\|_1 = |z_1| + |z_2|$ for $z \in \mathbb{R}^2$, meaning the 1-norm is linear on each quadrant of the plane. Every cell border $\mathcal{B}$ hence contains only $O(1)$ breakpoints where the function $t \mapsto \|P_{\|\cdot\|_1}(\mathcal{B}_1(t)) - Q_{\|\cdot\|_1}(\mathcal{B}_2(t))\|_1$ switches from one linear function piece to another. It follows that the base case in lines 1–2 of Algorithm 1, which uses straight paths, has piecewise linear integrands and yields $O(1)$ quadratic pieces on each initialised border $\mathcal{B}$. Computing all pieces requires $O(n + m)$ time.

Similarly, propagating the costs of straight paths from the cell corners in lines 6–7 yields $O(1)$ initial quadratic pieces per output border of the cell C . We now claim that lines 8–11 of Algorithm 1 take $O(N_{C,S} + N_{C,W})$ time, where N_B denotes the number of (maximal) quadratic pieces making up the computed cost function $\mathcal{B}.\text{apx}$ of a border $\mathcal{B}$. Note that the costs of optimal paths within C are given by Theorem 4, and a suitable ℓ is computable in $O(1)$ time under $\|\cdot\|_1$ [5, Corollary 9]. First, we consider the possible starting points of best paths to the north-east corner of C .

► **Definition 15.** Let $\mathcal{A}, \mathcal{B}$ be a pair of input and output border, and let $(s_0, t_0) \in \text{dom}(\mathcal{A}) \times \text{dom}(\mathcal{B})$ with $\mathcal{A}(s_0) \preceq \mathcal{B}(t_0)$ be arbitrary. If the function $s \mapsto \mathcal{A}.\text{apx}(s) + \text{opt}_{\mathcal{A},\mathcal{B}}(s, t_0)$ attains a semistrict¹ local minimum at s_0 , we say that $\mathcal{A}(s_0)$ is a parent candidate for $\mathcal{B}(t_0)$. If $\mathcal{B}.\text{apx}(t_0) = \mathcal{A}.\text{apx}(s_0) + \text{opt}_{\mathcal{A},\mathcal{B}}(s_0, t_0)$ holds, we say that $\mathcal{A}(s_0)$ is a parent point of $\mathcal{B}(t_0)$. We call an interval $S \subseteq \text{dom}(\mathcal{A})$ a parent interval of $T \subseteq \text{dom}(\mathcal{B})$ if each $\mathcal{B}(t)$ with $t \in T$ has a parent point $\mathcal{A}(s)$ with $s \in S$.

This definition builds upon [7, Definition 18], but allows for a more fine-grained analysis: Whereas [7] collects consecutive quadratic pieces into specific *subsegments* and then introduces a parent relationship for these subsegments, we deal with arbitrary points and intervals.

For a fixed t_0 , as in lines 8–9 of Algorithm 1, the given function $s \mapsto \mathcal{A}.\text{apx}(s) + \text{opt}_{\mathcal{A},\mathcal{B}}(s, t_0)$ is piecewise quadratic with $N_{\mathcal{A}} + O(1)$ pieces under $\|\cdot\|_1$. This is because $\mathcal{A}.\text{apx}$ has $N_{\mathcal{A}}$ pieces, and the optimal $(\mathcal{A}(s), \mathcal{B}(t_0))$ -paths from Theorem 4 have $O(1)$ bending points whose coordinates may depend linearly on the variable s . Hence, $s \mapsto \text{opt}_{\mathcal{A},\mathcal{B}}(s, t_0)$ has $O(1)$ quadratic pieces by the linearity of $\|\cdot\|_1$ on each quadrant. Basic calculus says that (semistrict) local minima of a piecewise quadratic function can only occur at domain endpoints, at breakpoints between pieces, or at points where the derivative is 0. The latter correspond to parabola vertices, see Figure 5.



■ **Figure 5** Local minima of a continuous and piecewise quadratic function.

Overall, there are $O(N_{C,S} + N_{C,W})$ parent candidates for the north-east corner of C . We can check all of them to find starting points of best paths, as the global minima that exist by Lemma 11 are among the local minima. Thus, computing the required values from lines 10–11 of Algorithm 1 takes $O(N_{C,S} + N_{C,W})$ time under $\|\cdot\|_1$. Moreover, we can reuse this candidate set for opposing propagation since parent points that do not share a coordinate with the child can be chosen from the same set. This is due to the following key lemma that pertains to happy optimal paths.

► **Lemma 16 (Happiness Lemma).** Let $X := \{(s, t) \in S \times T \mid \mathcal{A}(s) \preceq \mathcal{B}(t)\}$, where $\mathcal{A}, \mathcal{B}$ is a pair of input and output border with $S \times T \subseteq \text{dom}(\mathcal{A}) \times \text{dom}(\mathcal{B})$. If all $(s, t) \in X$ have a happy optimal $(\mathcal{A}(s), \mathcal{B}(t))$ -path, it is $\text{opt}_{\mathcal{A},\mathcal{B}}(s, t) = \rho_{\mathcal{A}}^>(s) + \rho_{\mathcal{B}}^>(t)$ on X for two univariate functions $\rho_{\mathcal{A}}^>, \rho_{\mathcal{B}}^>$. This always applies in case of $\mathcal{A} = \mathcal{B}.\text{opp}$ and $S \times T = \text{dom}(\mathcal{A}) \times \text{dom}(\mathcal{B})$.

¹ That means strict to the left and/or to the right, ruling out interior points of constant-valued pieces.

Proof. The function $\text{opt}_{\mathcal{A},\mathcal{B}}$ is obtained by evaluating costs of optimal $(\mathcal{A}(s), \mathcal{B}(t))$ -paths in terms of the parameter pair $(s, t) \in X$, and we assume all paths given by Theorem 4 to be happy on X . Consequently, each bending point ξ_0 lies on ℓ or on the cell's boundary according to Definition 5. This implies that ξ_0 can be expressed in terms of only s or in terms of only t , cf. Figure 1. We evaluate optimal path costs by summing the costs of subpaths, each of which traces a line segment between two points ξ and ξ' that are among $\mathcal{A}(s)$, $\mathcal{B}(t)$ and the paths' bending points. Definition 3 says that every such subpath cost is a definite integral of some function $\|\cdot\| \circ \psi$ on the interval $[\|\xi\|_1, \|\xi'\|_1]$, where ψ is an affine map. We distinguish the following two cases.

1. If the traced line segment lies on ℓ or on the cell's boundary, the corresponding function ψ is fixed and does not depend on s and t . By the Fundamental Theorem of Calculus, the subpath cost is equal to $\Psi(\|\xi'\|_1) - \Psi(\|\xi\|_1)$, where Ψ is an arbitrary antiderivative of the continuous integrand $\|\cdot\| \circ \psi$. Due to happiness, the term $\Psi(\|\xi\|_1)$ may depend only on s and can thus be assigned to $\rho_{\mathcal{A}}^{\triangleright}$, while $\Psi(\|\xi'\|_1)$ may depend only on t and can be assigned to $\rho_{\mathcal{B}}^{\triangleright}$.
2. Otherwise, the traced segment is horizontal/vertical with $\xi = \mathcal{A}(s)$ or $\xi' = \mathcal{B}(t)$ such that ψ depends only on s or only on t . Then the other endpoint must depend on the same parameter, i.e. ξ' depends on s or ξ depends on t . Due to happiness, it can be expressed in terms of only this parameter. The subpath cost can hence be assigned to $\rho_{\mathcal{A}}^{\triangleright}$ or $\rho_{\mathcal{B}}^{\triangleright}$ respectively.

It remains to show that every (x, y) -path γ for x, y on opposing borders is happy. Since paths from Theorem 4a are always happy, we assume that γ is as in Theorem 4b and consider the single breaking point ξ of γ . By construction, ξ shares one coordinate value with x and another with y . Opposing borders parametrise via the same coordinate and keep the other fixed, cf. Definition 2. It follows that the value of ξ in the non-parametric coordinate is equal to one border's fixed value. This means that ξ lies on the related border and thus on the boundary of C , so γ is happy. $\blacktriangleleft$

Being able to split costs into two univariate functions simplifies the behaviour of local minima in terms of one variable, which yields the desired characterisation of parent candidates.

► **Proposition 17.** *Let $\mathcal{A}$ be an input border of a cell C , and let $S_{\mathcal{A}}^*$ be defined as follows:*

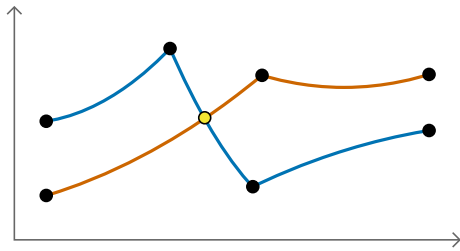
$$S_{\mathcal{A}}^* := \{s \in \text{dom}(\mathcal{A}) \mid \mathcal{A}(s) \text{ is a parent candidate for the north-east corner of } C\}.$$

Let $t_0 \in \text{dom}(\mathcal{B})$, where $\mathcal{B}$ is the opposing output border of $\mathcal{A}$. For every $s \in S_{\mathcal{A}}^$ with $s \leq t_0$ we have that $\mathcal{A}(s)$ is a parent candidate for $\mathcal{B}(t_0)$. Conversely, each parent candidate $\mathcal{A}(s_0)$ for $\mathcal{B}(t_0)$ satisfies $s_0 \in \{s \in S_{\mathcal{A}}^* \mid s \leq t_0\} \cup \{t_0\}$. It is $|S_{\mathcal{A}}^*| \in O(N_{\mathcal{A}})$ under $\|\cdot\|_1$.*

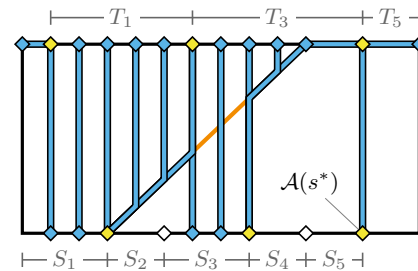
Proof. We start with the bound for $|S_{\mathcal{A}}^*|$ under $\|\cdot\|_1$. As argued above, for a fixed $t_0 \in \text{dom}(\mathcal{B})$ the function $s \mapsto \mathcal{A}.\text{apx}(s) + \text{opt}_{\mathcal{A},\mathcal{B}}(s, t_0)$ from Definition 15 is piecewise quadratic with $N_{\mathcal{A}} + O(1)$ pieces under $\|\cdot\|_1$. Apart from domain endpoints and piece breakpoints, each piece can contribute at most one semistrict local minimum with derivative 0, so we have $|S_{\mathcal{A}}^*| \in O(N_{\mathcal{A}})$. In general, we have $\mathcal{A}.\text{apx}(s) + \text{opt}_{\mathcal{A},\mathcal{B}}(s, t_0) = \mathcal{A}.\text{apx}(s) + \rho_{\mathcal{A}}^{\triangleright}(s) + \rho_{\mathcal{B}}^{\triangleright}(t_0)$ for all $s \in \text{dom}(\mathcal{A})$ with $s \leq t_0$, where $\rho_{\mathcal{A}}^{\triangleright}, \rho_{\mathcal{B}}^{\triangleright}$ are as in Lemma 16. Because $\rho_{\mathcal{B}}^{\triangleright}(t_0)$ does not depend on the variable s , all local minima that the function attains on the interior of its domain are entirely determined by $\mathcal{A}.\text{apx}$ and $\rho_{\mathcal{A}}^{\triangleright}$. The same applies to the fixed left endpoint of the domain, while the right endpoint is exactly t_0 per the constraint $s \leq t_0$ of monotone paths. Hence, it is the only parent candidate that can vary. The north-east corner of C gives the largest domain, so the stated results follow. $\blacktriangleleft$

The intuition behind $s_0 \in S_{\mathcal{A}}^*$ versus $s_0 = t_0$ for opposing propagation is that the corresponding paths may switch between the two types depicted in the second cell of Figure 1: Either they are straight with $s_0 = t_0$, or they travel on ℓ after starting at a point $\mathcal{A}(s_0)$ with $s_0 \in S_{\mathcal{A}}^*$. In fact, this switching is a major reason behind the $O((n+m)^5)$ time complexity of the exact 1D algorithm [7]. Note that in 1D the CDTW integrand is 0 everywhere on ℓ . The procedure from [7] propagates from $\mathcal{A}$ to ℓ , performs a cumulative minimum operation on ℓ , and then propagates from ℓ to $\mathcal{B}$. In 2D a cell C is generally associated with two non-parallel curve segments, so that the CDTW integrand is non-zero within C except at the point that corresponds to their intersection. Although the cumulative minimum operation can be adapted to non-zero weightings (cf. [7, Observation 46]), Lemma 16 reveals the underlying principles and generalises them to all happy paths. This technical contribution of ours advances the understanding of function propagations. It allows us to streamline and enhance the running time analysis compared to [7], yet it does not hold for unhappy paths. Their propagation complexity is still unknown in 2D, see [5, Section 4.2].

We now continue with the implementation of the subroutines from Algorithm 2 under the 1-norm $\|\cdot\|_1$. In $\mathcal{B}.\text{PROPAGATEFROMADJOINING}$ the function $\mathcal{B}.\text{apx}$ gets updated through taking its lower envelope with the function apx_{s^*} defined by $t \mapsto \mathcal{A}.\text{apx}(s^*) + \text{opt}_{\mathcal{A},\mathcal{B}}(s^*, t)$ on $\text{dom}(\mathcal{B})$. The given value s^* provides a fixed starting point for propagated paths, which may include unhappy paths. As fixing the first argument of $\text{opt}_{\mathcal{A},\mathcal{B}}$ is symmetric to fixing its second argument, we have that the function apx_{s^*} is piecewise quadratic with $O(1)$ pieces under $\|\cdot\|_1$. Moreover, $\mathcal{B}.\text{apx}$ has $O(1)$ pieces before the subroutine calls in lines 2–3 of Algorithm 2 occur, so taking the lower envelope requires finding only a constant number of intersections between pieces as in Figure 6.



■ **Figure 6** Intersection between piecewise quadratic functions.



■ **Figure 7** Monotone assignment of intervals on $\mathcal{B}$ to parents on $\mathcal{A}$.

In $\mathcal{B}.\text{PROPAGATEFROMOPPOSING}$ the cost function $\mathcal{B}.\text{apx}$ then gets updated through taking its lower envelope with the function $\text{apx}_{\leq s^*}$ defined by $t \mapsto \min_{s \leq s^* \wedge s \leq t} \mathcal{A}.\text{apx}(s) + \text{opt}_{\mathcal{A},\mathcal{B}}(s, t)$ on $\text{dom}(\mathcal{B})$. The function $\text{apx}_{\leq s^*}$ is itself a lower envelope, and Proposition 17 implies that it is covered by optimal paths starting at the parent candidates from $\{s \in S_{\mathcal{A}}^* \mid s \leq s^*\}$ together with straight paths starting before $\mathcal{A}(s^*)$. The costs of the former paths are given by $O(N_{\mathcal{A}})$ functions under $\|\cdot\|_1$, each of which again has $O(1)$ quadratic pieces. The latter paths' costs are given by the function $t \mapsto \mathcal{A}.\text{apx}(t) + \text{opt}_{\mathcal{A},\mathcal{B}}(t, t)$ for $t \leq s^*$, which has at most $N_{\mathcal{A}} + O(1)$ pieces.

The lower envelope of these pieces with the existing $O(1)$ pieces on $\mathcal{B}$ is computable in linear time since the new pieces are given monotonically by parent points. Those induce intervals on $\mathcal{B}$ that have parent intervals on $\mathcal{A}$ separated by $S_{\mathcal{A}}^*$, see Figure 7. We traverse $\mathcal{A}$ in ascending order, and compare pieces on $\mathcal{B}$ given by the current and an earlier interval on $\mathcal{A}$. In contrast to previous algorithms [7, 5], we do a single bottom-up pass on $\mathcal{B}$ instead of repeated top-down passes. As it takes $O(N_{\mathcal{A}})$ time to perform an opposing propagation under $\|\cdot\|_1$ with this method, lines 8–11 of Algorithm 1 take $O(N_{C.S} + N_{C.W})$ time in total. We can sum this bound over all cells.

► **Lemma 18.** *Let $t_0 \in \text{dom}(\mathcal{B})$ for an output border $\mathcal{B}$ with opposing input border $\mathcal{A} := \mathcal{B}.\text{opp}$, and let $\mathcal{A}(s_0)$ be the starting point of a best path from $\mathcal{A}$ to $\mathcal{B}(t_0)$ with smallest s_0 . It is*

$$\mathcal{A}.\text{apx}(s_0) + \text{opt}_{\mathcal{A},\mathcal{B}}(s_0, t) < \mathcal{A}.\text{apx}(s) + \text{opt}_{\mathcal{A},\mathcal{B}}(s, t) \quad \text{whenever } s < s_0 \leq t.$$

In particular, for all $t \geq s_0$ we have that every parent point $\mathcal{A}(s)$ of $\mathcal{B}(t)$ satisfies $s \geq s_0$.

Proof. Because $\mathcal{A}(s_0)$ is the starting point of a best path to $\mathcal{B}(t_0)$, Definition 10 says $\mathcal{A}.\text{apx}(s_0) + \text{opt}_{\mathcal{A},\mathcal{B}}(s_0, t_0) \leq \mathcal{A}.\text{apx}(s) + \text{opt}_{\mathcal{A},\mathcal{B}}(s, t_0)$ for all $s \leq t_0$. If $s < s_0$, this holds with strict inequality since there is no such point $\mathcal{A}(s)$ by choice of s_0 as the smallest possible value. Using Lemma 16 therefore gives $\mathcal{A}.\text{apx}(s_0) + \rho_{\mathcal{A}}^{\mathcal{B}}(s_0) + \rho_{\mathcal{B}}^{\mathcal{A}}(t_0) < \mathcal{A}.\text{apx}(s) + \rho_{\mathcal{A}}^{\mathcal{B}}(s) + \rho_{\mathcal{B}}^{\mathcal{A}}(t_0)$ for all $s < s_0$. By subtracting $\rho_{\mathcal{B}}^{\mathcal{A}}(t_0)$ from that inequality, adding $\rho_{\mathcal{B}}^{\mathcal{A}}(t)$ for a $t \geq s_0$, and using Lemma 16 once more, we get $\mathcal{A}.\text{apx}(s_0) + \text{opt}_{\mathcal{A},\mathcal{B}}(s_0, t) < \mathcal{A}.\text{apx}(s) + \text{opt}_{\mathcal{A},\mathcal{B}}(s, t)$ for all $s < s_0$ as desired. Thus, there is no parent point $\mathcal{A}(s)$ of $\mathcal{B}(t)$ with $s < s_0$ due to Proposition 12. ◀

► **Proposition 19.** *Let $\mathcal{B}$ be an output border with opposing input border $\mathcal{A} := \mathcal{B}.\text{opp}$. We can perform $\mathcal{B}.\text{PROPAGATEFROMOPPOSING}$ in $O(N_{\mathcal{A}})$ time under $\|\cdot\|_1$. Algorithm 1 then has a running time in $O(N)$, where N is the total number of quadratic pieces over all cell borders.*

Proof. In the following, we describe a procedure that computes the lower envelope $\text{apx}_{\leq s^*}$ for the subroutine, where $s^* \in \text{dom}(\mathcal{A})$ is the value passed to its call. Let $s_1^*, \dots, s_{\lambda}^*$ be the elements of the set $\{s \in S_{\mathcal{A}}^* \mid s \leq s^*\}$ in ascending order, where $S_{\mathcal{A}}^*$ is as in Proposition 17. We have w.l.o.g. that the inclusion $s^* \in S_{\mathcal{A}}^*$ holds, which further implies $s_{\lambda}^* = s^*$. This is since s^* gives a global and thus also a local minimum: Its computation in lines 8–10 of Algorithm 1 and the decisions made in lines 4–5 of Algorithm 2 mean $\mathcal{A}(s^*)$ is a parent point of the north-east cell corner.

We partition the interval $\{s \in \text{dom}(\mathcal{A}) \mid s \leq s_{\lambda}^*\}$, which contains all starting points considered by $\text{apx}_{\leq s^*}$, into subintervals $S_1 := \{s \in \text{dom}(\mathcal{A}) \mid s \leq s_1^*\}$ and $S_{\kappa} := (s_{\kappa-1}^*, s_{\kappa}^*]$ for $\kappa \in \{2, \dots, \lambda\}$. Given a subset $T_{\kappa} \subseteq \text{dom}(\mathcal{B})$ with parent interval S_{κ} for any $\kappa \in \{1, \dots, \lambda\}$, Proposition 17 implies that the pieces of $\text{apx}_{\leq s^*}$ on T_{κ} are those of the function τ_{κ} defined by $t \mapsto \mathcal{A}.\text{apx}(t) + \text{opt}_{\mathcal{A},\mathcal{B}}(t, t)$ for $t \in S_{\kappa}$ and $t \mapsto \mathcal{A}.\text{apx}(s_{\kappa}^*) + \text{opt}_{\mathcal{A},\mathcal{B}}(s_{\kappa}^*, t)$ for $t > s_{\kappa}^*$. See Figure 7 for an example. To compute the lower envelope of $\tau_1, \dots, \tau_{\lambda}$, it is not necessary to create all pieces of these functions. Instead, we traverse $S_1, \dots, S_{\lambda}$ and create pieces on the fly through maintaining an *active* function.

Initially, τ_1 is active on all of S_1 . For $\kappa \in \{2, \dots, \lambda\}$ we compare the restriction $\tau_{\kappa}|_{S_{\kappa}}$ with the yet active function τ_{μ} by checking their overlapping pairs of pieces for intersections as in Figure 6. This takes $O(1)$ time per pair. If τ_{μ} is nowhere greater than τ_{κ} on S_{κ} , then τ_{μ} remains active there. Otherwise, τ_{μ} only stays active up to the earliest point after which τ_{κ} is smaller. At this point τ_{κ} becomes active and stays so on the remainder of S_{κ} . After S_{λ} has been processed, the traversal is complete and the last active function τ_{μ} stays so on the remaining part of the domain.

This procedure correctly constructs the lower envelope: Let $\mathcal{A}(s_0)$ be the earliest starting point of a best path to an arbitrary point $\mathcal{B}(t_0)$, and let $\kappa \in \{1, \dots, \lambda\}$ with $s_0 \in S_{\kappa}$. By Proposition 17, all sets of parent candidates are finite. Hence, Lemma 18 and continuity of $s \mapsto \mathcal{A}.\text{apx}(s) + \text{opt}_{\mathcal{A},\mathcal{B}}(s, t)$, cf. the proof of Lemma 11, imply that each $t \geq s_0$ has a smallest $s_t \in [s_0, t]$ such that $\mathcal{A}(s_t)$ is a parent candidate for $\mathcal{B}(t)$. Together with the definition of S_{κ} , we get $\tau_{\kappa}(t) = \mathcal{A}.\text{apx}(s_t) + \text{opt}_{\mathcal{A},\mathcal{B}}(s_t, t) \leq \mathcal{A}.\text{apx}(s_0) + \text{opt}_{\mathcal{A},\mathcal{B}}(s_0, t) < \min\{\tau_1(t), \dots, \tau_{\kappa-1}(t)\}$ for all $t \geq s_0$, while τ_{κ} becomes active.

Furthermore, at most $N_{\mathcal{A}} + O(|S_{\mathcal{A}}^*|)$ overlapping pairs of pieces are created under $\|\cdot\|_1$, which yields a running time in $O(N_{\mathcal{A}})$ by Proposition 17. This is because $\tau_1|_{S_1}, \dots, \tau_{\lambda}|_{S_{\lambda}}$ together have at most $N_{\mathcal{A}} + |S_{\mathcal{A}}^*| + O(1)$ pieces, and we create at most $|S_{\mathcal{A}}^*| + O(1)$ pieces of $\tau_{\mu}|_{S_{\kappa}}$

with $\mu < \kappa$ over all active τ_μ due to the starting points then being fixed to s_μ^* . It remains to update $\mathcal{B}.\text{apx}$, which already contains $O(1)$ pieces, through taking its lower envelope with the pieces of $\text{apx}_{\leq s^*}$ in $O(N_{\mathcal{A}})$ time.² Altogether, we perform $\mathcal{B}.\text{PROPAGATEFROMOPPOSING}$ in $O(N_{\mathcal{A}})$ time, so that Algorithm 1 takes $O(N_{C,S} + N_{C,W})$ time per cell C by what has been argued above in this section. Summing this bound over all cells finally yields a total running time in $O(N)$ under $\|\cdot\|_1$. $\blacktriangleleft$

The central challenge is to bound the number N polynomially in the complexities n, m of the input curves. As indicated by the above proof, Lemma 18 implies $N_{\mathcal{B}} \leq N_{\mathcal{A}} + O(|S_{\mathcal{A}}^*|)$ under $\|\cdot\|_1$, but combining that with $|S_{\mathcal{A}}^*| \in O(N_{\mathcal{A}})$ from Proposition 17 is still too weak. It cannot rule out an exponential growth since it allows constants $c > 1$ with $N_{\mathcal{B}} \geq c \cdot N_{\mathcal{A}}$. This necessitates a better bound for $|S_{\mathcal{A}}^*|$ that does not depend on $N_{\mathcal{A}}$. We show that $|S_{\mathcal{A}}^*|$ grows at worst quadratically over successive propagations between opposing borders. The resulting inductive bound for $N_{\mathcal{B}}$ is our main ingredient for establishing that N is polynomial.

► **Definition 20.** We assign each border $\mathcal{B}$ a rank. If $\mathcal{B}.\text{PROPAGATEFROMOPPOSING}$ is not called for $\mathcal{B}$, it has rank 0. Else, it has rank $r := r_0 + 1$, where $r_0 \in \mathbb{N}_0$ is the rank of $\mathcal{B}.\text{opp}$.

► **Lemma 21.** Let $\mathcal{B}$ be an output border of rank $r > 0$ with opposing input border $\mathcal{A} := \mathcal{B}.\text{opp}$. Then the number $N_{\mathcal{B}}$ is bounded by $N_{\mathcal{B}} \leq N_{\mathcal{A}} + O(|S_{\mathcal{A}}^*|)$ with $|S_{\mathcal{A}}^*| \in O(r^2)$ under $\|\cdot\|_1$.

Lemma 21 follows from the more general Lemma 24 that we will prove below in Section 4.3. In comparison to [7, Lemma 14], the proof of our bounds avoids a distinction between different types of subsegments. It instead uses the Happiness Lemma to implicitly handle all types that are relevant for our algorithm. With these bounds in place, it remains to count the number of quadratic pieces on a border as well as the number of borders for each rank.

► **Proposition 22.** In a parameter space with $n \times m$ cells the ranks are at most $\max\{n, m\}$. There are $O(nm)$ borders of rank 0, each of which has $O(1)$ quadratic pieces under $\|\cdot\|_1$. There are $O(nm/r)$ borders of any rank $r > 0$, each of which has $O(r^3)$ pieces under $\|\cdot\|_1$.

Proof. The first claim is true since ranks can only accumulate along a row or column of cells by Definition 20. As there are $O(nm)$ borders in total, this particularly holds for borders of rank 0, and we have already argued in this section that these borders have $O(1)$ pieces under $\|\cdot\|_1$. Each border $\mathcal{B}$ of rank $r > 0$ has r predecessor borders of ranks $0, \dots, r-1$ that are disjoint from all the other predecessors, so the number of rank- r borders is in $O(nm/r)$. Finally, we show $N_{\mathcal{B}} \leq c^2 \cdot r^3$ by induction, where $c \in \mathbb{N}$ is a universal constant for the bounds of Lemma 21. In the base case for $r = 1$, only one propagation from a border of rank 0 with $O(1)$ pieces has occurred. For the inductive step we assume $r = r_0 + 1$, where $r_0 > 0$ is the rank of $\mathcal{A} := \mathcal{B}.\text{opp}$. By using Lemma 21 and the induction hypothesis for $\mathcal{A}$, we get

$$N_{\mathcal{B}} \leq N_{\mathcal{A}} + c \cdot |S_{\mathcal{A}}^*| \leq c^2 \cdot r_0^3 + c \cdot (c \cdot r_0^2) \leq c^2 \cdot r^2(r_0 + 1) = c^2 \cdot r^3. \quad \blacktriangleleft$$

When putting it together, we assume $n \geq m$ in order to bound the maximum rank by n . This is not a restriction since the input curves P, Q can be swapped: We have $\text{cdtw}_{\|\cdot\|}(P, Q) = \text{cdtw}_{\|\cdot\|}(Q, P)$ under any norm $\|\cdot\|$, as Definition 1 inherits the norm-induced symmetry.

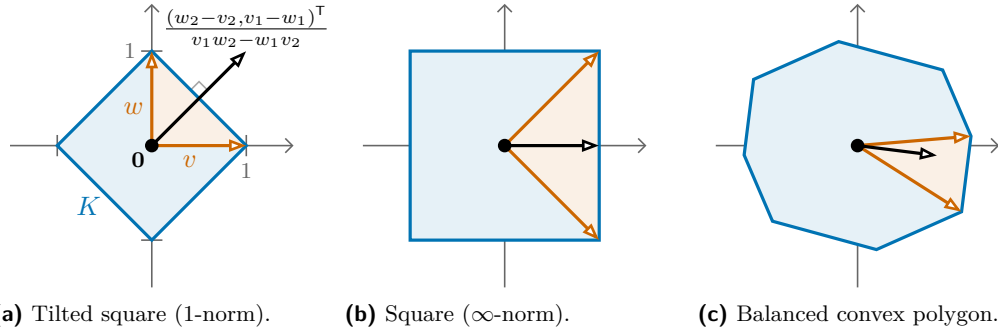
► **Theorem 23.** Given polygonal curves $P = \langle p_0, \dots, p_n \rangle$ and $Q = \langle q_0, \dots, q_m \rangle$ in $(\mathbb{R}^2, \|\cdot\|_1)$, where $n \geq m$, our 5-approximation for CDTW of P, Q under $\|\cdot\|_1$ takes $O(n^4 m)$ time.

² Alternatively to updating $\mathcal{B}.\text{apx}$ at the end, one can instead construct an initial active function τ_0 using its existing pieces, and start comparisons on S_1 . (Due to the separation of pieces that is explained in the proof of Lemma 24).

Proof. Proposition 19 says that the running time is linear in N . By Proposition 22, there is a constant $c \in \mathbb{N}$ with $N \leq c \cdot (nm + \sum_{r=1}^n nm/r \cdot r^3) = c \cdot nm \cdot (1 + \sum_{r=1}^n r^2) \leq 2c \cdot n^4 m$. ◀

4.3 Extension to Other Norms

We proceed to extend the above implementation and running time analysis of our algorithm from the 1-norm to general polygonal norms, which we will subsequently use in order to approximate any given norm on $\mathbb{R}^2$ such as the 2-norm. For these purposes, we employ the following characterisation: A function is a norm on $\mathbb{R}^d$ if and only if it is the *gauge* $\mathcal{G}_K: \mathbb{R}^d \rightarrow \mathbb{R}_{\geq 0}$ of a suitable set $K \subseteq \mathbb{R}^d$, defined by $\mathcal{G}_K(z) := \inf\{\lambda \geq 0 \mid z \in \lambda K\}$. E.g., the 2-norm is the gauge of the Euclidean unit ball, and the 1-norm on $\mathbb{R}^2$ is the gauge of the convex polygon that has vertices $(\pm 1, 0)^\top$ and $(0, \pm 1)^\top$. The precise conditions for K are that it is absorbing, balanced, bounded and convex. Because the gauge of any such set is equal to the gauges of its closure and interior, one may optionally further require that K is closed or open. (See [19, pp. 39–40].) We defer formal definitions to later.



■ **Figure 8** Polygons whose gauges are norms, along with evaluation vectors for specific cones.

For now, we only need the linearity property of polygonal norms that allows for the extension. While $\|\cdot\|_1$ is linear on each quadrant of the plane, any norm $\mathcal{G}_K$ for a suitable polygon $K \subseteq \mathbb{R}^2$ is linear on each cone $\Delta_{v,w} := \{\lambda v + \mu w \mid \lambda, \mu \geq 0\}$, where $v, w \in \mathbb{R}^2$ are two adjacent vertices of K . In fact, we can evaluate $\mathcal{G}_K$ on the cone $\Delta_{v,w}$ through the scalar product $\mathcal{G}_K(z) = \frac{(w_2 - v_2, v_1 - w_1)}{v_1 w_2 - w_1 v_2} \cdot z$, which uses a vector that is orthogonal to $v - w$. See Figure 8 for examples. This property yields piecewise linear CDTW integrands, and it suffices for proving the generalised inductive bound that we desire. If K is a polygon of complexity $k \in \mathbb{N}$, i.e. it has k vertices and k cones, then the piecewise quadratic functions that had $O(1)$ pieces under $\|\cdot\|_1$ have $O(k)$ pieces under $\mathcal{G}_K$.

► **Lemma 24.** *Consider any norm $\mathcal{G}_K$ on $\mathbb{R}^2$ such that K is a polygon of complexity $k \in \mathbb{N}$, and let $\mathcal{B}$ be an output border of rank $r > 0$ with opposing input border $\mathcal{A} := \mathcal{B}.\text{opp}$. Then the number $N_{\mathcal{B}}$ is bounded by $N_{\mathcal{B}} \leq N_{\mathcal{A}} + O(|S_{\mathcal{A}}^*| + k)$ with $|S_{\mathcal{A}}^*| \in O(r^2 \cdot k)$ under $\mathcal{G}_K$.*

Proof. As the considered norm $\mathcal{G}_K$ is linear on k cones partitioning the plane, fixing an argument of $\text{opt}_{\mathcal{A}, \mathcal{B}}$ now yields a piecewise quadratic function with $O(k)$ pieces, which is one of $\rho_{\mathcal{A}}^>, \rho_{\mathcal{B}}^>$ from Lemma 16 plus the fixed offset resulting from the other. This is again because the $O(1)$ bending points of the optimal paths from Theorem 4 may depend linearly on the non-fixed argument, so that the arguments of $\mathcal{G}_K$ in the integrand for the path costs can switch cones $O(k)$ times.

For the same reason, the initialisation of $\mathcal{B}.\text{apx}$ in lines 6–7 of Algorithm 1 and the subroutine call $\mathcal{B}.\text{PROPAGATEFROMADJOINING}$ from Algorithm 2 result in $O(k)$ quadratic pieces. Some of these pieces may remain after the call $\mathcal{B}.\text{PROPAGATEFROMOPPOSING}$, but

they cannot result in an increase of complexity: Every dominating best path from $\mathcal{A}$ to a point $\mathcal{B}(t)$ intersects all paths from $\mathcal{B}.\text{adj}$ to points $\mathcal{B}(t')$ for $t' > t$. By applying Lemma 11 as in the proof of Proposition 12, it then follows that there are dominating best paths from $\mathcal{A}$ to all such $\mathcal{B}(t')$. Thus, the pieces that the two input borders give are separated, and $\mathcal{B}.\text{adj}$ contributes at most $O(k)$ pieces to $N_{\mathcal{B}}$. To show $N_{\mathcal{B}} \leq N_{\mathcal{A}} + O(|S_{\mathcal{A}}^*| + k)$, we need to bound the number of pieces given by $\mathcal{A} = \mathcal{B}.\text{opp}$. For this, we consider the causes that can contribute pieces of $\text{apx}_{\leq s^*}$. Recall that we have

$$\text{apx}_{\leq s^*}(t) = \min_{s \leq s^* \wedge s \leq t} \mathcal{A}.\text{apx}(s) + \text{opt}_{\mathcal{A}, \mathcal{B}}(s, t) = \rho_{\triangleright}^{\mathcal{B}}(t) + \min_{s \leq s^* \wedge s \leq t} \mathcal{A}.\text{apx}(s) + \rho_{\triangleleft}^{\mathcal{A}}(s)$$

for all $t \in \text{dom}(\mathcal{B})$ due to its definition and Lemma 16, so one possible cause is a switch to another piece of $\rho_{\triangleright}^{\mathcal{B}}$. Apart from this, Proposition 17 implies that any new piece of $\text{apx}_{\leq s^*}$ may only begin when our assignment of parent points needs to switch to another piece of $\mathcal{A}.\text{apx}$ or to one of $\rho_{\triangleleft}^{\mathcal{A}}$, or when it needs to switch to or from a parent point among the candidates contained in $S_{\mathcal{A}}^*$. These switches never occur in reverse direction since Lemma 18 says that parent points can be assigned monotonically. Hence, $\rho_{\triangleleft}^{\mathcal{A}}$ and $\rho_{\triangleright}^{\mathcal{B}}$ contribute $O(k)$ pieces to $N_{\mathcal{B}}$, while $\mathcal{A}.\text{apx}$ contributes at most $N_{\mathcal{A}}$ pieces. We charge each parent point from $S_{\mathcal{A}}^*$ twice to cover the case that it causes a new piece both when switching to it and when switching away from it in a monotone assignment. This contributes at most $2 \cdot |S_{\mathcal{A}}^*|$ pieces, so we obtain $N_{\mathcal{B}} \leq N_{\mathcal{A}} + O(|S_{\mathcal{A}}^*| + k)$ as desired.

We next bound $|S_{\mathcal{A}}^*|$, which is the number of parent candidates for the north-east cell corner. By Definition 15 and Lemma 16, these candidates correspond to the semistrict local minima of the piecewise quadratic function $s \mapsto \mathcal{A}.\text{apx}(s) + \rho_{\triangleleft}^{\mathcal{A}}(s)$. They can thus only occur at the two domain endpoints, at the breakpoints between pieces, or at points where the derivative is 0. More precisely, a local minimum can occur at some breakpoint only if the left piece's derivative is at most 0 and the right piece's derivative is at least 0, see Figure 5. This is because the function is additionally continuous, cf. the proof of Lemma 11. Its left and right derivatives are defined everywhere. Let ∂_- and ∂_+ respectively denote the left and right differentiation operator. In the following, we split $\text{dom}(\mathcal{A})$ into open intervals such that $\partial_-[\mathcal{A}.\text{apx}(s) + \rho_{\triangleleft}^{\mathcal{A}}(s)] \geq \partial_+[\mathcal{A}.\text{apx}(s) + \rho_{\triangleleft}^{\mathcal{A}}(s)]$ holds on every interval, which implies that local minima attained within them must have derivative 0. For this, we consider the predecessor borders $\mathcal{A}_0, \dots, \mathcal{A}_{r-1}$ of $\mathcal{B}$, where $\mathcal{A}_{r-1} := \mathcal{A}$ has rank $r-1$, and $\mathcal{A}_{i-1} := \mathcal{A}_i.\text{opp}$ has rank $i-1$ for $i \in \{1, \dots, r-1\}$. They all share the same domain.

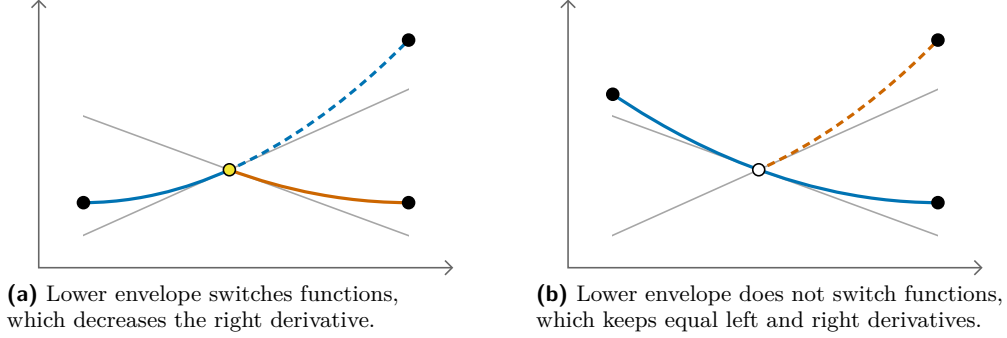
We introduce partition points for our open intervals at the breakpoints of the functions $\mathcal{A}_0.\text{apx}$ and $\rho_{\triangleleft}^{\mathcal{A}}$ as well as, for all $i \in \{1, \dots, r-1\}$, at the breakpoints of $\rho_{\triangleleft}^{\mathcal{A}_{i-1}}$ and $\rho_{\triangleright}^{\mathcal{A}_i}$ from Lemma 16. These are $O(r)$ functions and each has $O(k)$ pieces, so their breakpoints split $\text{dom}(\mathcal{A})$ into $O(r \cdot k)$ open intervals. In addition to proving the above property, we determine the number of semistrict local minima with derivative 0 to be at most r on each interval, so we get $|S_{\mathcal{A}}^*| \in O(r^2 \cdot k)$. Let $I \subseteq \text{dom}(\mathcal{A})$ be one of the open intervals. As $\mathcal{A}_0.\text{apx}$ has a single piece on I by construction, its left and right derivatives match on all of I . Given $i \in \{1, \dots, r-1\}$, we therefore inductively assume $\partial_- \mathcal{A}_{i-1}.\text{apx}(s) \geq \partial_+ \mathcal{A}_{i-1}.\text{apx}(s)$ for all $s \in I$, and show that $\mathcal{A}_i.\text{apx}$ also has this property. For $\mathcal{A}_{r-1} = \mathcal{A}$ it then transfers to $s \mapsto \mathcal{A}.\text{apx}(s) + \rho_{\triangleleft}^{\mathcal{A}}(s)$ since $\rho_{\triangleleft}^{\mathcal{A}}$ has a single piece on I . As for the inductive step: Proposition 17 implies that $\mathcal{A}_i.\text{apx}|_I$ is the lower envelope of

$$\alpha_i: I \rightarrow \mathbb{R}_{\geq 0}, \quad \text{defined by} \quad \alpha_i(t) := \mathcal{A}_{i-1}.\text{apx}(t) + \rho_{\triangleleft}^{\mathcal{A}_{i-1}}(t) + \rho_{\triangleright}^{\mathcal{A}_i}(t),$$

$$\text{as well as} \quad \alpha_{i,s_0}: \{t \in I \mid t \geq s_0\} \rightarrow \mathbb{R}_{\geq 0} \quad \text{for all } s_0 \in S_{\mathcal{A}_{i-1}}^*,$$

$$\text{defined by} \quad \alpha_{i,s_0}(t) := \mathcal{A}_{i-1}.\text{apx}(s_0) + \rho_{\triangleleft}^{\mathcal{A}_{i-1}}(s_0) + \rho_{\triangleright}^{\mathcal{A}_i}(t).$$

Due to $\rho_{\mathcal{A}_{i-1}}^{\triangleright}$ and $\rho_{\triangleright}^{\mathcal{A}_i}$ each having a single piece on I by construction, it follows $\partial_- \alpha_i(t) \geq \partial_+ \alpha_i(t)$ for all $t \in I$ via the induction hypothesis. Meanwhile, for all $s_0 \in S_{\mathcal{A}_{i-1}}^*$ and all $t \in I$ with $t > s_0$ we even have $\partial_- \alpha_{i,s_0}(t) = \partial_+ \alpha_{i,s_0}(t)$ since the argument of $\mathcal{A}_{i-1}.\text{apx}$ in α_{i,s_0} is fixed to s_0 . The lower envelope retains this property: It only switches from one of the functions to another if the next function's right derivative is less than or equal to the previous one's. Else, the previous function would still yield smaller values beyond the breakpoint, see Figure 9. Thus, $\partial_- \mathcal{A}_i.\text{apx}(t) \geq \partial_+ \mathcal{A}_i.\text{apx}(t)$ holds for all $t \in I$, including the breakpoints introduced by the lower envelope.



■ **Figure 9** Behaviour of a continuous lower envelope depending on the functions' derivatives.

It remains to bound the number of semistrict local minima with derivative 0 occurring on I . For this, we count the distinct (a, b) -coefficient pairs over all the quadratic pieces $s \mapsto as^2 + bs + c$ of the function $\mathcal{A}.\text{apx}$ on I . Such a pair determines the derivative of the pieces to be $s \mapsto 2as + b$, and thereby fixes the sole point with derivative 0 to be $-b/(2a)$ in case of $a \neq 0$. That means the number of distinct (a, b) -pairs gives an upper bound for the number of parabola vertices. Note that the case of $a = 0$ does not contribute additional semistrict local minima, as it yields a derivative of constant value b . Local minima on I have derivative 0 as argued before, requiring also $b = 0$ and thus constant-valued pieces. Any semistrict minimum on those pieces can only be attained at a piece breakpoint. Such a point then has to coincide either with an endpoint of the open interval I , or with a point of derivative 0 that is fixed by a neighbouring piece's pair.

We use induction again to prove that $\mathcal{A}_i.\text{apx}$ has at most $i + 1$ distinct (a, b) -coefficient pairs on I for all $i \in \{0, \dots, r - 1\}$. The base case for $i = 0$ is true since $\mathcal{A}_0.\text{apx}$ has a single piece on I . For $i > 0$ we inductively assume that $\mathcal{A}_{i-1}.\text{apx}$ has at most i such pairs, and we consider $\mathcal{A}_i.\text{apx}$ in form of the lower envelope from above. Adding $\rho_{\mathcal{A}_{i-1}}^{\triangleright}$ and $\rho_{\triangleright}^{\mathcal{A}_i}$, which both have a single piece on I , to $\mathcal{A}_{i-1}.\text{apx}$ preserves the number of (a, b) -pairs. Therefore, α_i has at most i pairs on I . Moreover, all functions α_{i,s_0} for $s_0 \in S_{\mathcal{A}_{i-1}}^*$ share the single (a, b) -pair of $\rho_{\triangleright}^{\mathcal{A}_i}$ on I because the terms with fixed arguments contribute to c instead of a and b . Together, we have that $\mathcal{A}_i.\text{apx}$ has at most $i + 1$ distinct (a, b) -coefficient pairs on I . For $\mathcal{A}_{r-1} = \mathcal{A}$ the number r is again preserved by adding the single piece of $\rho_{\mathcal{A}}^{\triangleright}$ on I , so that we overall have $|S_{\mathcal{A}}^* \cap I| \leq r$ as claimed above. ◀

Note that the ideas of counting (a, b) -coefficient pairs [7, Lemma 14] as well as utilising left and right derivatives [7, Lemma 17] originate from the analysis of the exact 1D algorithm. There, the desired properties and bounds were shown separately on subsegments, which were associated with different types of optimal paths as in Figure 1. This required a distinction into a lot of cases, and it caused the analysis to become large and repetitive. One difficulty was the lack of tools for taking a unified approach and exploiting common principles behind

function propagations. Lemma 16 now provides such a tool. Although the subsegments are still reflected by the above proof splitting the domain into intervals, our analysis is much shorter and more straightforward than that from [7]. We believe that this constitutes an insightful contribution to the understanding of propagation complexity. A general analysis of unhappy paths in 2D remains open for future work. The costs of these paths cannot be split into two univariate functions (cf. [7, Section 4.5]).

We next use Lemma 24 to establish that under a polygonal norm $\mathcal{G}_K$ of complexity $k \in \mathbb{N}$ our algorithm can be implemented with running time $O(n^4 m \cdot k)$. This extension is not difficult, yet there are two important aspects for achieving a linear factor in k instead of a worse factor: First, the fact that the procedure for opposing propagation from Proposition 19 uses a single pass on the output border $\mathcal{B}$, which is an algorithmic contribution of ours. Doing multiple passes as in [7, 5] would lead to a higher running time (cf. [5, Proposition 18]). Second, evaluating $\mathcal{G}_K$ at a point $z \in \mathbb{R}^2$ requires finding the cone of K that contains z . This is possible in $O(\log(k))$ time via binary search. A naive implementation would do this $O(k)$ times per cell to compute costs of optimal paths, which we avoid by keeping track of how the arguments of $\mathcal{G}_K$ switch cones. See the full version for a proof of Theorem 25.

► **Theorem 25.** *Consider any norm $\mathcal{G}_K$ on $\mathbb{R}^2$ such that K is a polygon of complexity $k \in \mathbb{N}$. Given polygonal curves $P = \langle p_0, \dots, p_n \rangle$ and $Q = \langle q_0, \dots, q_m \rangle$ in $(\mathbb{R}^2, \mathcal{G}_K)$, where $n \geq m$, our 5-approximation for CDTW of P, Q under $\mathcal{G}_K$ takes $O(n^4 m \cdot k)$ time.*

To approximate CDTW in 2D under the 2-norm, we may use a regular polygon inscribed in the Euclidean unit disk. This was done in [5, Corollary 17], and a factor of $1 + \varepsilon$ requires polygon complexity $k \in O(\varepsilon^{-1/2})$. In fact, results from convex geometry imply that we can approximate any fixed norm on $\mathbb{R}^2$ by some polygonal norm of the same complexity. More generally, any fixed norm on $\mathbb{R}^d$ is $(1 + \varepsilon)$ -approximated by some polyhedral norm of complexity $O(\varepsilon^{(1-d)/2})$. In the following, we outline this polyhedral approximation and its application to our results. Its first ingredient is the geometric characterisation of norms, which we have introduced before: A function is a norm on $\mathbb{R}^d$ if and only if it is the gauge $\mathcal{G}_K$ of an absorbing, balanced, bounded, convex and closed set K . The first two properties are defined as follows (see [19, p. 11]).

1. A set $K \subseteq \mathbb{R}^d$ is called *absorbing* (or *radial*) if for every point $z \in \mathbb{R}^d$ there exists some $\lambda_0 \geq 0$ such that $z \in \lambda K$ holds whenever $|\lambda| \geq \lambda_0$. This ensures that $\mathcal{G}_K$ indeed assigns every $z \in \mathbb{R}^d$ a real number $\mathcal{G}_K(z) = \inf\{\lambda \geq 0 \mid z \in \lambda K\}$. It further implies $\mathbf{0} \in K$.
2. A set $K \subseteq \mathbb{R}^d$ is called *balanced* (or *circled*) in case of $\lambda K \subseteq K$ for all $\lambda \in [-1, 1]$. This yields absolute homogeneity of $\mathcal{G}_K$, i.e. $\mathcal{G}_K(\lambda z) = |\lambda| \mathcal{G}_K(z)$ for all $z \in \mathbb{R}^d$ and all $\lambda \in \mathbb{R}$. If K is a polyhedron, it further implies that K must have an even number of vertices.

The other three properties are defined as usual, and we do not require those definitions here. Every absorbing, convex and closed set in $\mathbb{R}^d$ contains the origin $\mathbf{0}$ in its interior. When a bounded, convex and closed set in $\mathbb{R}^d$ has a non-empty interior, it is also called a *convex body*. Moreover, gauges allow describing any norm $\|\cdot\|$ via its unit sublevel set: We have that $\|\cdot\|$ is the gauge $\mathcal{G}_M$ of the balanced convex body $M := \{z \in \mathbb{R}^d \mid \|z\| \leq 1\}$. (See [19, p. 40].)

The second ingredient is the approximation of convex bodies by (convex) polyhedra, for which there is a vast amount of literature, see [4] for a survey. Note that a d -dimensional polyhedron K generally has different numbers of vertices and facets when $d > 2$, where the facets correspond to the cones on which the gauge $\mathcal{G}_K$ is linear. The specific result that we need holds independent of whether one desires a bound on the number of vertices or facets, though this may lead to distinct polyhedra. Because we do not consider the equipped norm to be an input for CDTW, we suppress all constants in the size and dimension of the given convex body by treating it as fixed.

► **Lemma 26** ([4, Section 4.4]). *Fix any balanced convex body $M \subseteq \mathbb{R}^d$. For every $\varepsilon > 0$ there is a convex polyhedron $K \subseteq \mathbb{R}^d$ with $O(\varepsilon^{(1-d)/2})$ facets (or vertices) and $M \subseteq K \subseteq (1+\varepsilon)M$.*

For e.g. the sublevel sets of p -norms, which are hyperellipsoids if $1 < p < \infty$, it is not difficult to determine these approximating polyhedra. Constructions for the general case may involve very large constants, see [4, Section 8] for algorithmic considerations. A simple approach, albeit with a higher worst-case asymptotic complexity, is to choose roughly $\varepsilon^{-d} \log(\varepsilon^{-d})$ points uniformly at random from the boundary of $(1+\varepsilon)M$. Then their convex hull is a $(1+\varepsilon)$ -approximation of M with high probability [18]. Also, note that Proposition 27, which we prove in the full version, might be applicable to other results that work for polyhedral norms, such as the subquadratic DTW algorithm from [12].

► **Proposition 27.** *Fix any norm $\|\cdot\|$ on $\mathbb{R}^d$. For every $\varepsilon > 0$ there is another norm $\mathcal{G}_K$ on $\mathbb{R}^d$, where $K \subseteq \mathbb{R}^d$ is a balanced convex polyhedron with $O(\varepsilon^{(1-d)/2})$ facets (or vertices), such that $\|z\| \leq \mathcal{G}_K(z) \leq (1+\varepsilon)\|z\|$ holds for all $z \in \mathbb{R}^d$.*

► **Theorem 28.** *Fix any norm $\|\cdot\|$ on $\mathbb{R}^2$, and let $\varepsilon > 0$ be arbitrary. Given two polygonal curves $P = \langle p_0, \dots, p_n \rangle$ and $Q = \langle q_0, \dots, q_m \rangle$ in $(\mathbb{R}^2, \|\cdot\|)$, where $n \geq m$, we can compute a factor- $(5+\varepsilon)$ approximation for CDTW of P, Q under $\|\cdot\|$ in running time $O(n^4 m / \varepsilon^{1/2})$.*

Proof. We use Proposition 27 for $\varepsilon_0 := \min\{\frac{\varepsilon}{15}, 1\}$ to get a norm $\mathcal{G}_K$. By Definition 1, it is

$$\text{cdtw}_{\|\cdot\|}(P, Q) \leq \text{cdtw}_{\mathcal{G}_K}(P, Q) \leq (1+\varepsilon_0)^2 \cdot \text{cdtw}_{\|\cdot\|}(P, Q) \leq (1+\frac{\varepsilon}{5}) \cdot \text{cdtw}_{\|\cdot\|}(P, Q).$$

Applying Theorem 25 to $\mathcal{G}_K$ thus returns a $(5+\varepsilon)$ -approximation in $O(n^4 m / \varepsilon^{1/2})$ time. ◀

5 Conclusion

We established the first constant-factor approximation algorithm for CDTW in 2D that has a polynomial running time. It relies on a new integration-based building block and uses a tailored propagation scheme. Another technical contribution of ours is a unified approach to the running time analysis, thanks to novel insights into core principles behind propagation complexity. It is still open whether the additional propagation patterns occurring in exact 2D algorithms, which we circumvented, have polynomial complexity. We also do not know if the approximation factor of 5 is tight, see Remark 7 and Section 4.1. To evaluate efficiency and solution quality in practice, an experimental comparison of an exact algorithm and our approximation algorithm would be useful.

Our algorithm works for all polygonal norms. We demonstrated that this allows to approximate CDTW in 2D under any desired norm, making our results applicable in a very broad 2D setting. Recently, there has been an increased interest in solving not only geometric but also combinatorial optimisation problems under more general norm objective functions, see for example [9, 10]. One advantage of measuring the similarity of curves under a norm of choice is the possibility to assign the curve dimensions, which may carry different types of data, suitable weightings [13]. Thus, higher dimensions are a natural direction for future work on CDTW. Known fundamentals do not suffice there since Theorem 4 does not hold in 3D and beyond, see the full version for a counterexample.

References

- 1 Sotiris Brakatsoulas, Dieter Pfoser, Randall Salas, and Carola Wenk. On map-matching vehicle tracking data. In *Proceedings of the 31st International Conference on Very Large Data Bases*, pages 853–864. VLDB Endowment, 2005. URL: <https://dl.acm.org/doi/10.5555/1083592.1083691>.

- 2 Milutin Brankovic. *Graphs and Trajectories in Practical Geometric Problems*. PhD thesis, University of Sydney, 2022.
- 3 Milutin Brankovic, Kevin Buchin, Koen Klaren, André Nusser, Aleksandr Popov, and Sampson Wong. (k, l) -medians clustering of trajectories using Continuous Dynamic Time Warping. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, pages 99–110. ACM, 2020. doi:10.1145/3397536.3422245.
- 4 Efim M. Bronstein. Approximation of convex sets by polytopes. *Journal of Mathematical Sciences*, 153(6):727–762, 2008. doi:10.1007/S10958-008-9144-X.
- 5 Kevin Buchin, Maike Buchin, Jan Erik Swiadek, and Sampson Wong. Fundamentals of computing Continuous Dynamic Time Warping in 2D under different norms. In *20th International Conference and Workshops on Algorithms and Computation*, pages 467–482. Springer, 2026. doi:10.1007/978-981-95-7127-7_31.
- 6 Kevin Buchin, Maike Buchin, and Yusu Wang. Exact algorithms for partial curve matching via the Fréchet distance. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 645–654. SIAM, 2009. doi:10.1137/1.9781611973068.71.
- 7 Kevin Buchin, André Nusser, and Sampson Wong. Computing Continuous Dynamic Time Warping of time series in polynomial time. *Journal of Computational Geometry*, 16(1):765–799, 2025. doi:10.20382/JOCG.V16I1A21.
- 8 Maike Buchin. *On the Computability of the Fréchet Distance between Triangulated Surfaces*. PhD thesis, Freie Universität Berlin, 2007. URL: <https://refubium.fu-berlin.de/handle/fub188/1909>.
- 9 Deeparnab Chakrabarty and Chaitanya Swamy. Approximation algorithms for minimum norm and ordered optimization problems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 126–137. ACM, 2019. doi:10.1145/3313276.3316322.
- 10 Kuowen Chen, Jian Li, Yuval Rabani, and Yiran Zhang. New results on a general class of minimum norm optimization problems. In *52nd International Colloquium on Automata, Languages, and Programming*, pages 50:1–50:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.ICALP.2025.50.
- 11 Alon Efrat, Quanfu Fan, and Suresh Venkatasubramanian. Curve matching, time warping, and light fields: New algorithms for computing similarity between curves. *Journal of Mathematical Imaging and Vision*, 27(3):203–216, 2007. doi:10.1007/S10851-006-0647-0.
- 12 Omer Gold and Micha Sharir. Dynamic Time Warping and Geometric Edit Distance: Breaking the quadratic barrier. *ACM Transactions on Algorithms*, 14(4):50:1–50:17, 2018. doi:10.1145/3230734.
- 13 Theodor Gutschlag and Sabine Storandt. On the generalized Fréchet distance and its applications. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, pages 35:1–35:10. ACM, 2022. doi:10.1145/3557915.3560970.
- 14 Sarel Har-Peled, Benjamin Raichel, and Eliot W. Robson. The Fréchet distance unleashed: Approximating a dog with a frog. In *41st International Symposium on Computational Geometry*, pages 54:1–54:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.SOCG.2025.54.
- 15 Koen Klaren. Continuous Dynamic Time Warping for clustering curves. Master’s thesis, Eindhoven University of Technology, 2020. URL: <https://research.tue.nl/en/studentTheses/continuous-dynamic-time-warping-for-clustering-curves>.
- 16 Anil Maheshwari, Jörg-Rüdiger Sack, and Christian Scheffer. Approximating the integral Fréchet distance. *Computational Geometry*, 70–71:13–30, 2018. doi:10.1016/J.COMGEO.2018.01.001.
- 17 Mario E. Munich and Pietro Perona. Continuous Dynamic Time Warping for translation-invariant curve alignment with applications to signature verification. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 108–115. IEEE, 1999. doi:10.1109/ICCV.1999.791205.

- 18 Márton Naszódi. Approximating a convex body by a polytope using the Epsilon-Net Theorem. *Discrete & Computational Geometry*, 61(3):686–693, 2019. doi:10.1007/S00454-018-9977-0.
- 19 Helmut H. Schaefer and Manfred P. Wolff. *Topological Vector Spaces*. Springer, 2nd edition, 1999. doi:10.1007/978-1-4612-1468-7.
- 20 Bruno Serra and Marc Berthod. Subpixel contour matching using continuous dynamic programming. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 202–207. IEEE, 1994. doi:10.1109/CVPR.1994.323830.
- 21 Bruno Serra and Marc Berthod. Optimal subpixel matching of contour chains and segments. In *Proceedings of IEEE International Conference on Computer Vision*, pages 402–407. IEEE, 1995. doi:10.1109/ICCV.1995.466911.
- 22 Han Su, Shuncheng Liu, Bolong Zheng, Xiaofang Zhou, and Kai Zheng. A survey of trajectory distance measures and performance evaluation. *The VLDB Journal*, 29(1):3–32, 2020. doi:10.1007/S00778-019-00574-9.
- 23 Yaguang Tao, Alan Both, Rodrigo I. Silveira, Kevin Buchin, Stef Sijben, Ross S. Purves, Patrick Laube, Dongliang Peng, Kevin Toohey, and Matt Duckham. A comparative analysis of trajectory similarity measures. *GIScience & Remote Sensing*, 58(5):643–669, 2021. doi:10.1080/15481603.2021.1908927.