Bringmann, Karl; Staals, Frank; Węgrzycki, Karol; van Wordragen, Geert

Fine-Grained Complexity of Earth Mover's Distance Under Translation

Published in:
40th International Symposium on Computational Geometry, SoCG 2024

DOI:
10.4230/LIPIcs.SoCG.2024.25

Published: 01/06/2024

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Bringmann, K., Staals, F., Węgrzycki, K., & van Wordragen, G. (2024). Fine-Grained Complexity of Earth Mover's Distance Under Translation. In W. Mulzer, & J. M. Phillips (Eds.), *40th International Symposium on Computational Geometry, SoCG 2024* Article 25 (Leibniz International Proceedings in Informatics, LIPIcs; Vol. 293). Schloss Dagstuhl - Leibniz-Zentrum für Informatik. https://doi.org/10.4230/LIPIcs.SoCG.2024.25

Fine-Grained Complexity of Earth Mover's Distance Under Translation

Karl Bringmann

Saarland University and Max-Planck-Institute for Informatics, Saarbrücken, Germany

Frank Staals

Department of Information and Computing Sciences, Utrecht University, The Netherlands

Karol Węgrzycki

Saarland University and Max Planck Institute for Informatics, Saarbrücken, Germany

Geert van Wordragen

Department of Computer Science, Aalto University, Espoo, Finland

Abstract

The Earth Mover's Distance is a popular similarity measure in several branches of computer science. It measures the minimum total edge length of a perfect matching between two point sets. The Earth Mover's Distance under Translation (EMDuT) is a translation-invariant version thereof. It minimizes the Earth Mover's Distance over all translations of one point set.

For EMDuT in $\mathbb{R}^1$, we present an $\widetilde{\mathcal{O}}(n^2)$-time algorithm. We also show that this algorithm is nearly optimal by presenting a matching conditional lower bound based on the Orthogonal Vectors Hypothesis. For EMDuT in $\mathbb{R}^d$, we present an $\widetilde{\mathcal{O}}(n^{2d+2})$-time algorithm for the L_1 and L_∞ metric. We show that this dependence on d is asymptotically tight, as an $n^{o(d)}$-time algorithm for L_1 or L_∞ would contradict the Exponential Time Hypothesis (ETH). Prior to our work, only approximation algorithms were known for these problems.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Earth Mover's Distance, Earth Mover's Distance under Translation, Fine-Grained Complexity, Maximum Weight Bipartite Matching

Digital Object Identifier 10.4230/LIPIcs.SoCG.2024.25

Related Version *Full Version*: `https://arxiv.org/abs/2403.04356` [13]

Funding *Karl Bringmann and Karol Węgrzycki:* This work is part of the project TIPEA that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 850979).

Acknowledgements This work was initiated at the Workshop on New Directions in Geometric Algorithms, May 14-19 2023, Utrecht, The Netherlands.

1 Introduction

Earth Mover's Distance (EMD). EMD, also known as geometric transportation or geometric bipartite matching, is a widely studied distance measure (see, e.g., [29, 6, 5, 7, 37, 31, 25, 24, 1, 3]) that has received significant interest in computer vision, starting with the work of [39]. Depending on the precise formulation, EMD is a distance measure on point sets, distributions, or functions. In this paper, we study the following formulation of EMD as measuring the distance from a set of blue points B to a set of red points R:

$$\mathrm{EMD}_p(B, R) = \min_{\text{injective } \phi\colon B \to R} \sum_{b \in B} \|b - \phi(b)\|_p.$$

Here, the minimization goes over all injective functions from B to R, i.e., ϕ encodes a perfect matching of the points in B to points in R, and the cost of a matching is the total length of all matching edges, with respect to the L_p metric, $1 \leqslant p \leqslant \infty$. When the value of p is irrelevant, we may drop the subscript p.

The EMD_p problem is to compute the value $\mathrm{EMD}_p(B, R)$ for given sets $B, R \subseteq \mathbb{R}^d$ of sizes $|B| \leqslant |R| = n$. This general problem is sometimes called the *asymmetric* EMD. The *symmetric* EMD is the special case with the additional restriction $|B| = |R|$. Intuitively, the asymmetric EMD asks whether B is similar to some subset of R, while the symmetric variant compares the full sets B and R. In this paper, we assume the dimension d to be constant.

We briefly discuss algorithms for EMD. Note that EMD can be formulated as a mincost matching problem on a bipartite graph with vertices $R \cup B$, where edge lengths are equal to the point-to-point distances. This graph has $|R| \cdot |B| = \mathcal{O}(n^2)$ edges and solving bipartite mincost matching by the Hungarian method yields an exact algorithm for EMD with running time $\mathcal{O}(n^3)$. Alternatively, by combining geometric spanners with recent advancements in (approximate) mincost flow solvers, one can obtain fast approximation algorithms for EMD. For instance, symmetric EMD in L_2 metric can be solved in time $n(\log(n)/\varepsilon)^{\mathcal{O}(d)}$ [31]. See also [29, 7, 25, 24, 1, 3] for more approximation algorithms. Conditional lower bounds are also known, but they apply only when the dimension is super-constant [37].

Earth Mover's Distance under Translation (EMDuT). We study a variant of EMD that is invariant under translations, and thus compares shapes of point sets, ignoring their absolute positions:

$$\mathrm{EMDuT}_p(B, R) = \min_{\tau \in \mathbb{R}^d} \mathrm{EMD}_p(B + \tau, R).$$

Here, $B + \tau = \{b + \tau \mid b \in B\}$ is the translated point set. See Figure 1 for an illustration of this distance measure. Again, we call asymmetric EMDuT_p the problem of computing $\mathrm{EMDuT}_p(B, R)$ for given sets B, R of size $|B| \leqslant |R| = n$, and the symmetric variant comes with the additional restriction $|B| = |R|$. This measure was introduced by Cohen and Guibas [18], who presented heuristics as well as an exact algorithm with respect to the squared Euclidean distance. Later, Klein and Veltkamp [32] designed a 2-approximation algorithm for symmetric EMDuT_p running in asymptotically the same time as any EMD algorithm. Cabello, Giannopoulos, Knauer, and Rote [14] designed $(1 + \varepsilon)$-approximation algorithms for EMDuT_2 in the plane, running in time $\widetilde{\mathcal{O}}(n^4/\varepsilon^4)$ for the asymmetric variant and $\widetilde{\mathcal{O}}(n^{3/2}/\varepsilon^{7/2})$ for the symmetric variant.[1] Eppstein et al. [22] proposed algorithms to solve the symmetric EMDuT_1 and symmetric EMDuT_∞ problems in the plane, that run in $\mathcal{O}(n^6 \log^3 n)$ time. We remark that most of these works also study variants of EMDuT under more general transformations than translations, but in this paper we focus on translations.

We are not aware of any other research on EMDuT, which is surprising, since translation-invariant distance measures are well motivated, and the analogous Hausdorff distance under translation [26, 38, 27, 2, 34, 33, 12, 15] and Fréchet distance under translation [4, 35, 30, 8, 10, 23, 11] have received considerably more attention.

[1] Here and throughout the paper we use $\widetilde{\mathcal{O}}$ notation to ignore logarithmic factors, i.e., $\widetilde{\mathcal{O}}(T) = \bigcup_{c \geqslant 0} \mathcal{O}(T(\log T)^c)$.

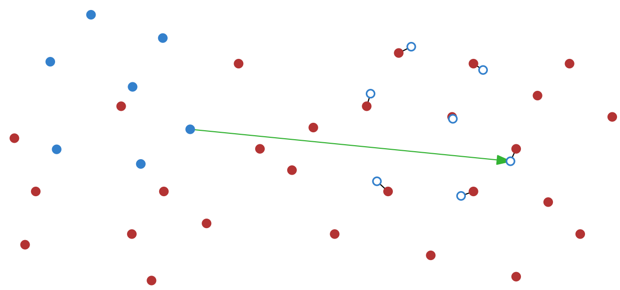

Figure 1 Given a set of (solid) blue points B and a set of red points R, our goal is to find a translation τ (shown in green) and a perfect matching from $B + \tau$ to R (shown in black) that minimizes the total distance of matched pairs.

1.1 Our results

We study EMDuT from the perspective of fine-grained complexity. We design new algorithms and prove conditional lower bounds over $\mathbb{R}^1$, as well as for L_1 and L_∞ over $\mathbb{R}^d$.

EMDuT in 1D. Over $\mathbb{R}^1$ all L_p metrics are equal. We present the following new algorithms.

▶ **Theorem 1** (1D Algorithms). *(Symmetric:) Given sets $B, R \subseteq \mathbb{R}$ of size $n = |B| = |R|$, $\mathrm{EMDuT}(B, R)$ can be computed in time $\mathcal{O}(n \log n)$. (Asymmetric:) Given sets $B, R \subseteq \mathbb{R}$ of size $m = |B| \leqslant n = |R|$, $\mathrm{EMDuT}(B, R)$ can be computed in time $\mathcal{O}(mn(\log n + \log^2 m))$.*

Note that for $m = \Omega(n)$, for the asymmetric variant we obtain near-quadratic time $\widetilde{\mathcal{O}}(n^2)$, while for the symmetric variant we obtain near-linear time $\widetilde{\mathcal{O}}(n)$. We fully explain this gap, by proving a matching conditional lower bound showing that no algorithm solves the asymmetric variant in strongly subquadratic time $\mathcal{O}(n^{2-\delta})$ for any $\delta > 0$, for $m = \Omega(n)$. In fact, we present a stronger lower bound that even rules out fast approximation algorithms, not only fast exact algorithms. Our lower bound assumes the Orthogonal Vectors Hypothesis (OVH), a widely-accepted conjecture from fine-grained complexity theory; for a definition see Section 4.

▶ **Theorem 2** (1D Lower Bound). *Assuming OVH, for any constant $\delta > 0$ there is no algorithm that, given $\varepsilon \in (0, 1)$ and sets $B, R \subseteq \mathbb{R}$ of size $n = |R| \geqslant |B| = \Omega(n)$, computes a $(1 + \varepsilon)$-approximation of $\mathrm{EMDuT}(B, R)$ in time $\mathcal{O}(n^{2-\delta}/\varepsilon^{o(1)})$.*

As a corollary, the same conditional lower bound holds for EMDuT_p over $\mathbb{R}^d$, for any $d \geqslant 1$ and $1 \leqslant p \leqslant \infty$, since subsets of $\mathbb{R}$ can be embedded into $\mathbb{R}^d$ for any dimension d and any L_p metric.

Let us give a brief overview of these results. In the symmetric setting, we establish that $f(\tau) := \mathrm{EMD}(B + \tau, R)$ is a unimodal function in τ, i.e., it is first monotone decreasing and then monotone increasing, and thus its minimum can be found easily. In contrast, in the asymmetric setting the function $f(\tau)$ can have up to $\Theta(n^2)$ disconnected global minima. Intuitively, our lower bound shows that any algorithm needs to consider each one of these global near-minima, and therefore the running time must be quadratic in order to determine which near-minimum is the actual global minimum. To obtain our algorithm in the asymmetric setting, we use a sweep algorithm with an intricate event handling data structure.

EMDuT for L_1 and L_∞ metric in higher dimensions. We extend the work of Eppstein et al. [22] for point sets in $\mathbb{R}^d$, leading to the following algorithms.

▶ **Theorem 3** (Algorithms for L_1 and L_∞ metric, Asymmetric)**.** *Given sets $B, R \subseteq \mathbb{R}^d$ of size $m = |B| \leqslant n = |R|$, $\mathrm{EMDuT}_1(B,R)$ and $\mathrm{EMDuT}_\infty(B,R)$ can be computed in time $\mathcal{O}(m^d n^{d+2} \log^{d+2} n)$.*

We explain that such a dependence on dimension is unavoidable, by establishing a more coarse-grained lower bound compared to our 1D results: We show that no algorithm can solve the problem in time $n^{o(d)}$. In fact, we present a stronger lower bound that even rules out fast approximation algorithms. Our lower bound assumes the Exponential Time Hypothesis (ETH) [28], which is a well-established conjecture from fine-grained complexity theory.

▶ **Theorem 4** (Lower Bound for L_1 and L_∞ metric, Symmetric)**.** *Assuming ETH, there is no algorithm that, given $\varepsilon \in (0,1)$ and sets $B, R \subseteq \mathbb{R}^d$ of size $n = |B| = |R|$, computes a $(1+\varepsilon)$-approximation of $\mathrm{EMDuT}_1(B,R)$ in time $(\frac{n}{\varepsilon})^{o(d)}$. The same holds for $\mathrm{EMDuT}_\infty(B,R)$.*

Note that our lower bound pertains to the symmetric setting, while our algorithm addresses the more general asymmetric setting. Hence, these results together cover both the symmetric and the asymmetric setting.

Let us give a brief overview of these results. For the algorithm, we establish an arrangement of complexity $\mathcal{O}(m^d n^d)$ such that the optimal translation τ is attained at one of the vertices within this arrangement. Our algorithm is obtained by computing the EMD at each vertex. The lower bound is proven via a reduction from the k-Clique problem. In our construction, each coordinate of the translation τ chooses one vertex from a given k-Clique instance. We design gadgets that verify that every pair of selected nodes indeed forms an edge.

1.2 Open problems

EMDuT in 1D. Over $\mathbb{R}^1$, we leave open whether there are fast approximation algorithms: Can a constant-factor approximation be computed in time $\mathcal{O}(n^{2-\delta})$ for some constant $\delta > 0$? Or even in time $\widetilde{\mathcal{O}}(n)$? Can a $(1+\varepsilon)$-approximation be computed in time $\widetilde{\mathcal{O}}(n^{2-\delta}/\mathrm{poly}(\varepsilon))$ for some constant $\delta > 0$ (independent of n and ε)? Or even in time $\widetilde{\mathcal{O}}(n/\mathrm{poly}(\varepsilon))$?

EMDuT for L_1 and L_∞ metric in higher dimensions. For the L_1 and L_∞ metric in dimension $d \geqslant 2$ we leave open to determine the optimal constant $c > 0$ such that the problem can be solved in time $n^{c \cdot d + o(d)}$.

EMDuT for L_2 metric in higher dimensions. The L_2 metric is the most natural measure in the geometric settings, making EMDuT_2 a well motivated problem. The most pressing open problem is to determine the complexity of the EMDuT_2 problem in any dimension $d \geqslant 2$.

It is known that the EMDuT_2 problem cannot be solved exactly. Namely, for any point set $R \subset \mathbb{R}^d$ of size n, if B consists of n copies of the point $(0, \dots, 0)$, then $\mathrm{EMDuT}_2(B,R)$ is the (cost of the) Geometric Median of R. Because the Geometric Median has no exact algebraic expression (even for $d = 2$) [9], there is no exact algorithm for EMDuT_2 in dimension $d \geqslant 2$.

We therefore need to relax the goal and ask for an approximation algorithm. Geometric Median has a very fast $(1+\varepsilon)$-approximation algorithm running in time $\mathcal{O}(nd \log^3(1/\varepsilon))$ [17], so the reduction from Geometric Median to EMDuT_2 does not rule out very fast approximation algorithms for EMDuT_2.

This is in stark contrast to what we know about the EMDuT_2 problem, as almost all of our techniques in this paper completely fail for this problem. We neither obtain an algorithm running in time $n^{\mathcal{O}(d)}$, nor can we prove a lower bound ruling out time $n^{o(d)}$. On the lower bound side, all we know is the lower bound from 1D, ruling out $(1+\varepsilon)$-approximation algorithms running in time $\mathcal{O}(n^{2-\delta}/\varepsilon^{o(1)})$ for any constant $\delta > 0$. On the algorithms side, one can observe that after fixing the matching from B to R, the problem of finding the optimal translation τ for this matching is the Geometric Median problem and thus has a $(1+\varepsilon)$-approximation algorithm running in time $\mathcal{O}(nd\log^3(1/\varepsilon))$. By trying out all $n^{\mathcal{O}(n)}$ possible matchings, one can obtain a $(1+\varepsilon)$-approximation algorithm for EMDuT_2 running in time $n^{\mathcal{O}(n)}\log^3(1/\varepsilon)$ for any constant d. We pose as an open problem to close this huge gap between the quadratic lower and exponential upper bound (for $(1+\varepsilon)$-approximation algorithms with a $1/\varepsilon^{o(1)}$ dependency on ε in the running time).

2 Preliminaries

We use $[n]$ to denote $\{1,\ldots,n\}$. All logarithms are base 2. For every $x \in \mathbb{R}$ we let $\lfloor x \rceil \in \mathbb{Z}$ be the unique integer such that $x - \lfloor x \rceil \in (-1/2, 1/2]$. Consider a set of blue points $B \subseteq \mathbb{R}^d$ and a set of red points $R \subseteq \mathbb{R}^d$. Fix an L_p norm, for any $1 \leqslant p \leqslant \infty$. Denote by Φ the set of all injective functions $\phi\colon B \to R$, i.e., Φ is the set of all perfect matchings from B to R. For any matching $\phi \in \Phi$ and any translation $\tau \in \mathbb{R}^d$ we define the cost

$$\mathcal{D}_{B,R,p}(\phi,\tau) = \sum_{b\in B} \|b+\tau-\phi(b)\|_p.$$

We will ignore the subscript p when it is clear from the context. Note that we can express EMD and EMDuT in terms of this cost function as

$$\mathrm{EMD}_p(B,R) = \min_{\phi\in\Phi} \mathcal{D}_{B,R,p}(\phi,(0,\ldots,0)) \text{ and } \mathrm{EMDuT}_p(B,R) = \min_{\phi\in\Phi}\min_{\tau\in\mathbb{R}^d} \mathcal{D}_{B,R,p}(\phi,\tau).$$

3 Algorithm in one dimension

We first consider computing $\mathrm{EMDuT}_p(B,R)$ for two point sets B, R in $\mathbb{R}^1$. For ease of presentation, assume that R and B are indeed sets, and thus there are no duplicate points. We can handle the case of duplicate points by symbolic perturbation. Observe, that the distance between a pair of points b, r in any L_p metric is simply $\|b-r\|_p = \|b-r\|_1 = |b-r|$. In Section 3.1, we describe a very simple $\mathcal{O}(n\log n)$ time algorithm to compute $\mathrm{EMDuT}_p(B,R)$ (as well as an optimal matching ϕ^* and translation τ^* that realize this distance) when B and R both contain exactly n points. In Section 3.2, we consider the much more challenging case where $|B| = m$ and $|R| = n$ differ. For this case we develop an $\mathcal{O}(nm(\log n + \log^2 m))$ time algorithm to compute $\mathrm{EMDuT}_p(B,R)$. Omitted proofs are included in the full version [13].

A matching ϕ is said to be *monotonically increasing* if and only if for every pair of blue points $b' < b$ we also have $\phi(b') < \phi(b)$. We show the following crucial property.

▶ **Lemma 5.** *When $B, R \subset \mathbb{R}$ there is an optimal matching ϕ that is monotonically increasing.*

3.1 Symmetric case

In the symmetric case ($|R| = |B|$), Lemma 5 uniquely defines an optimal matching. Let $B = \{b_1,\ldots,b_n\}$ and $R = \{r_1,\ldots,r_n\}$ be the points in increasing order. Now, the optimal translation τ^* is the value for τ that minimizes $\mathcal{D}_{B,R}(\phi,\tau) = \sum_{i=1}^{n} |b_i - r_i + \tau|$. Thus, it corresponds to the median of $b_1 - r_1, \ldots, b_n - r_n$, which we can compute in $O(n\log n)$ time.

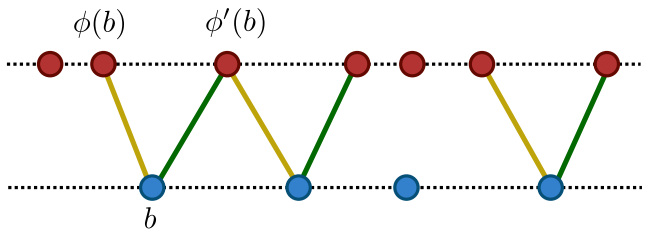

Figure 2 Schematic representation of the graph $G = \phi \otimes \phi'$ used in the proof of Lemma 7. Each edge exists if and only if exactly one edge from either ϕ or ϕ' is present. Green edges arise from the matching ϕ', while yellow edges arise from the matching ϕ. Initially, we demonstrate that the connected components of this graph are paths. Then, considering that the matchings are monotone, it follows that the edges of these paths are non-crossing. This implies that consecutive red vertices on these paths are monotone. Hence, if we translate to the right, the matching ϕ' is superior to ϕ.

▶ **Theorem 6.** *We can compute* $\mathrm{EMDuT}(R, B)$ *in 1D in* $\mathcal{O}(n \log n)$ *time when* $|R| = |B|$.

3.2 Asymmetric case

We will present an $\mathcal{O}(mn(\log n + \log^2 m))$ time algorithm to compute $\mathrm{EMDuT}(B, R)$, for the case that $m \leqslant n$. Consider the cost $f(\tau) = \min_{\phi\in\Phi} \mathcal{D}_{B,R}(\phi, \tau)$ as a function of τ. The minimum of this function is $\mathrm{EMDuT}(B, R)$. The main idea is then to sweep over the domain of f, increasing τ from $-\infty$ to ∞, while maintaining (a representation of) f and a matching ϕ that realizes cost $f(\tau) = \mathcal{D}_{B,R}(\phi, \tau)$. We also maintain the best translation $\tau^* \leqslant \tau$ (i.e. with minimal cost) among the translations considered so far (and if there are multiple such translations, the smallest one), so at the end of our sweep, τ^* is thus an optimal translation.

Properties of f. By Lemma 5, for any τ, there exists an optimal monotonically increasing matching between $B+\tau$ and R. So, we restrict our attention to such monotonically increasing matchings. Observe that any such matching ϕ corresponds to a partition of B into *runs*, i.e. maximal subsequences of consecutive points, $B_1, \ldots, B_z$, so that the points $b_{t-k}, \ldots, b_t$ in a run B_i are matched to consecutive red points $r_{u-k}, \ldots, r_r$, for some $r_u = \phi(b_t)$. Moreover, for any such a matching ϕ, the function $\mathcal{D}_{B,R}(\phi, \tau)$ is piecewise linear in τ, and each breakpoint is a translation τ for which there is a pair $(b, r) \in B \times R$ with $b + \tau = r$. It then follows that $f(\tau)$ is also piecewise linear in τ. Furthermore, the breakpoints of f are of two types. A type (i) breakpoint is a translation such that there is a pair $(b, r) \in B \times R$ with $b + \tau = r$, and a type (ii) breakpoint if there are two different matchings ϕ, ϕ' that both realize the same minimum cost $\mathcal{D}_{B,R}(\phi, \tau) = \mathcal{D}_{B,R}(\phi', \tau)$. We show the following key lemma, which lets us characterize the breakpoints of type (ii) more precisely.

▶ **Lemma 7.** *Let ϕ be an optimal monotone matching of* $\mathrm{EMD}_p(B + \tau, R)$*, and let ϕ' be an optimal monotone matching of* $\mathrm{EMD}_p(B + \tau', R)$ *for some $\tau' > \tau$. Then, $\phi'(b) \geqslant \phi(b)$ for all $b \in B$.*

See Figure 2 for a sketch of the proof of Lemma 7. The full proof is in the full version [13].

▶ **Corollary 8.** *A breakpoint τ of type (ii) corresponds to a pair of monotonically increasing matchings ϕ, ϕ' for which for all points $b \in B$ we have $\phi(b) \leqslant \phi'(b)$. Furthermore, consider a run $b_s, \ldots, b_t$ of ϕ and a point b_i with $i \in \{s, \ldots, t\}$. If $\phi(b_i) < \phi'(b_i)$, then $\phi(b_j) < \phi'(b_j)$ for all $j \in \{i, \ldots, t\}$.*

▶ **Lemma 9.** *The function $f(\tau)$ is piecewise linear, and consists of $\mathcal{O}(nm)$ pieces.*

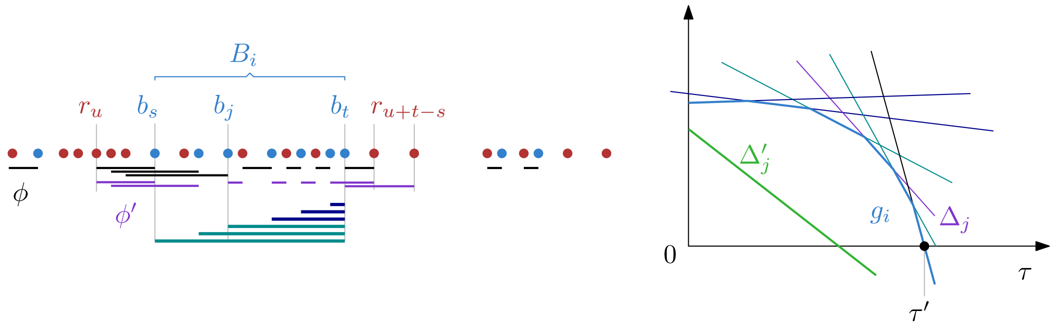

Figure 3 Each point b_j in a run $B_i = b_s, \ldots, b_t$ defines a (piecewise)-linear function Δ'_j. Each suffix $b_j, \ldots, b_t$ then defines a linear function Δ_j, expressing the cost of switching from matching ϕ to ϕ'. The lower envelope g_i of these functions then defines the first type (ii) event τ' of run B_i.

Proof. As argued above, f is piecewise linear. What remains is to argue that there are $\mathcal{O}(nm)$ breakpoints. For every pair of points $(b_i, r_j) \in B \times R$ there is only one translation τ such that $b + \tau = r$, so clearly there are at most $\mathcal{O}(nm)$ breakpoints of type (i). At every breakpoint of type (ii), there is at least one blue point b_i that was matched to r_j and gets matched to some r_k with $k > j$. This also happens at most once for every pair b_i, r_j. Hence, the number of breakpoints of type (ii) is also $\mathcal{O}(nm)$. ◀

In our sweep line algorithm we will maintain a current optimal matching ϕ. At each breakpoint of type (i) we will have an event to update the cost function of the matching. Furthermore, it follows from Corollary 8 that when we sweep over a breakpoint of type (ii), we can decompose the changes to the matching using a series of *atomic* events. In each such atomic event there is some suffix $b_j, \ldots, b_t$ of a run $b_s, \ldots, b_t$ that ϕ currently matches to $r_{u-j}, \ldots, r_u$ that will become matched to $r_{u-j+1}, \ldots, r_{u+1}$. As we argued in the proof of Lemma 9, the total number of such events is only $\mathcal{O}(nm)$. Next, we express how we can efficiently compute the next such atomic event, and handle it.

Consider a run $B_i = b_s, \ldots, b_t$ induced by ϕ at time τ. Our aim is to find the smallest $\tau' \geqslant \tau$ at which there is an atomic type (ii) event involving a suffix $b_j, \ldots, b_t$ of B_i. Hence, for a given suffix $b_j, \ldots, b_t$, we wish to maintain when it starts being beneficial to match $b_j, \ldots, b_t$ to $r_{u-j+1}, \ldots, r_{u+1}$ rather than to $r_{u-j}, \ldots, r_u$.

Let Δ'_j represent the change in cost when we match $b = b_j$ to $r' = r_{v+1}$ rather than to $r = r_v$, ignoring that r_{v+1} may already be matched to some other blue point. We have that

$$\Delta'_j(\tau) = |b - r' + \tau| - |b - r + \tau| = \begin{cases} r' - r & \text{if } b + \tau \leqslant r, \\ r + r' - 2b - 2\tau & \text{if } r < b + \tau < r', \\ r - r' & \text{if } b + \tau \geqslant r'. \end{cases}$$

Observe that this function is piecewise linear, and decreasing (more precisely, non-increasing). Moreover, the breakpoints coincide with type (i) breakpoints of f at which $b + \tau$ coincides with a red point. Hence, in between any two consecutive events, we can consider Δ'_j as a linear function. See Figure 3 for an illustration.

We can then express the cost of changing the matching for the entire suffix $b_j, \ldots, b_t$ as $\Delta_j(\tau) = \sum_{k=j}^{t} \Delta'_k(\tau)$. This function is again decreasing, piecewise linear, and has breakpoints that coincide with type (i) breakpoints of f. When $\Delta_j(\tau)$ becomes non-positive it becomes beneficial to match the suffix $b_j, \ldots, b_t$ to $r_{u-j+1}, \ldots, r_{u+1}$. Hence, the first such translation is given by a root of $\Delta_j(\tau)$. Note that there is at most one such root since Δ_j is decreasing.

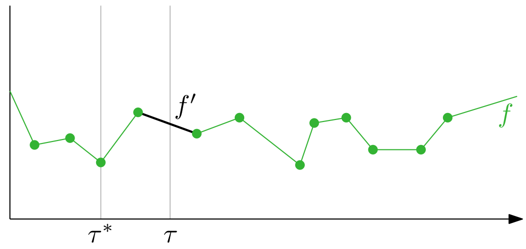

Figure 4 We sweep the domain of f, while maintaining a representation of the current piece f' of f, and the best translation $\tau^* \leqslant \tau$ found so far. Breakpoints correspond to type (i) or type (ii) events.

It now follows that (if it exists) the root τ' of the function $g_i(\tau) = \min_{j \in \{s,\ldots,t\}} \Delta_j(\tau)$ expresses the earliest time that there is a suffix $b_j, \ldots, b_t$ for which it is beneficial to update the matching. As before, this function is decreasing and piecewise linear. Hence, we obtain:

▶ **Lemma 10.** *Let $[\tau_1, \tau'] \ni \tau$ be a maximal interval on which $f(\tau)$ is linear, let τ' be a type (ii) breakpoint, and let ϕ be an optimal matching for τ. Then there is a run B_i induced by ϕ, and τ' is a root of the function $g_i(\tau)$.*

Representing the lower envelope g_i. At any moment of our sweep, we maintain a single piece of g_i. Hence, this piece is the lower envelope of a set of linear functions $\Delta_s, \ldots, \Delta_t$. We will maintain this lower envelope using an adapted version of the data structure by Overmars and van Leeuwen [36]. Ideally, we would maintain the lower envelope of $\Delta_s, \ldots, \Delta_t$ directly. However, reassigning a single blue point b_j in the matching ϕ, may cause many functions Δ_k to change. So, we implicitly represent each function Δ_j as a sum of Δ'_k functions.

▶ **Lemma 11.** *Let B_i be a run of size k. We can represent the current piece of the lower envelope g_i such that we can find the root of (this piece of) g_i in $\mathcal{O}(\log k)$ time, and insert or remove any point in B_i in $\mathcal{O}(\log^2 k)$ time.*

The main algorithm. Our main algorithm sweeps the space of all possible translations, while maintaining an optimal matching ϕ for the current translation τ, a representation of the current piece of the function f (i.e., the linear function f' for which $f(\tau) = f'(\tau)$), and the best translation $\tau^* \leqslant \tau$ found so far. To support the sweep, we also maintain a Lemma 11 data structure for each run B_i induced by ϕ, and a global priority queue. The Lemma 11 data structure allows us to efficiently obtain the next type (ii) event of a run B_i. The global priority queue stores all type (i) events, as well as the first type (ii) event of each run.

We initialize the priority queue by inserting all translations for which a pair $(b, r) \in B \times R$ coincide as type (i) events. Let τ_0 be the first such event. For a translation $\tau < \tau_0$, the matching ϕ that assigns b_i to r_i is optimal (by Lemma 5). Hence, we use ϕ as the initial matching. We compute the corresponding function f' expressing the cost of ϕ, construct the data structure of Lemma 11 on the single run induced by ϕ, and query it for its first type (ii) event. We add this event to the priority queue. All of this can be done in $\mathcal{O}(mn)$ time.

To handle a type (i) event involving point b_j, we remove it from the data structure for its run and add it back in the same place with its updated linear function $\Delta'_j(\tau)$. We query the data structure to find the next type (ii) event of the run B_i containing b_j, and update the event of B_i in the global priority queue if needed. Finally, if b_j is aligned with $\phi(b_j)$ in the event, we update f' by adding the function $2(b_j + \tau - \phi(b_j))$ and evaluate it. Handling an event of type (i) takes $\mathcal{O}(\log n + \log^2 m)$ time, as it involves a constant number of operations in the global priority queue, each taking $\mathcal{O}(\log(nm)) = \mathcal{O}(\log n)$ time, and a constant number of operations involving the Lemma 11 data structures, each taking $\mathcal{O}(\log^2 m)$ time.

To handle a type (ii) event where the matching changes for points $b_j, \ldots, b_t \in B_i$, we remove each point from the data structure for B_i and then add them to the run they are now a part of (which can be either the existing run B_{i+1} or a new run in between B_i and B_{i+1}). This takes $\mathcal{O}(\log^2 m)$ time per point, but as argued in Lemma 9 each point can only be involved in $\mathcal{O}(n)$ events of this type, so over all events, this takes $\mathcal{O}(nm \log^2 m)$ time. We then recompute the type (ii) events corresponding to the at most two affected runs in $\mathcal{O}(\log m)$ time, and update them in the global priority queue in $\mathcal{O}(\log n)$ time. Here, we update f' by adding the (linear) cost function $\Delta_i(\tau)$ associated with the event.

Thus, we handle a total of $\mathcal{O}(nm)$ events of type (i), each taking $\mathcal{O}(\log n + \log^2 m)$ time, and $\mathcal{O}(nm)$ events of type (ii), which take a total of $\mathcal{O}(nm(\log n + \log^2 m))$ time as well.

Once we have processed all events, the algorithm has found an optimal translation τ^*. We run the sweep once more from the start, and stop at translation τ^*, then report the current matching ϕ as an optimal matching. Together with Theorem 6, this thus establishes Theorem 1.

4 Lower bound in one dimension

In the Orthogonal Vectors problem (OV) we are given two sets of vectors $X, Y \subseteq \{0,1\}^d$ of size $|X| = |Y| = n$ and the task is to decide whether there exist $x \in X$ and $y \in Y$ with $x \cdot y = 0$, where $x \cdot y = \sum_{i=1}^{d} x[i] \cdot y[i]$. A naive algorithm solves this problem in time $\mathcal{O}(n^2 d)$.

▶ **Hypothesis 12** (Orthogonal Vectors Hypothesis (OVH) [42, 41]). *No algorithm solves the Orthogonal Vectors problem in time $\mathcal{O}(n^{2-\delta} d^c)$ for any constants $\delta, c > 0$.*

In this section we prove the following theorem.

▶ **Theorem 13.** *Assuming OVH, for any constant $\delta > 0$ there is no algorithm that, given sets $B, R \subseteq \mathbb{R}$ of size $n = |R| \geqslant |B| = \Omega(n)$, computes $\mathrm{EMDuT}(B, R)$ in time $\mathcal{O}(n^{2-\delta})$. This even holds with the additional restriction $B, R \subseteq \{0, 1, \ldots, \mathcal{O}(n^4)\}$.*

Observe that this immediately implies Theorem 2 because each coordinate is polynomially bounded. Hence, we focus on of Theorem 13. We only give the main ideas of the proof here. In particular, in Section 4.1 we construct the vector gadgets, and in Section 4.2 we present the reduction. Omitted details, and the full correctness argument are in the full version [13].

4.1 Vector gadgets

We have two different types of gadgets depending on whether a vector belongs to set X or Y (see Figure 5 for illustration):

▶ **Definition 14** (Red Vectors). *For a vector $x \in \{0,1\}^d$, create a group of points $R(x)$ to consist of: $8d$ points at the coordinate 0, and $8d$ points at the coordinate $4d + 1$. Next, for every $i \in \{1, \ldots, d\}$: (i) if $x[i] = 0$, we put points $\{4i-3, 4i-2, 4i-1, 4i\}$, and (ii) if $x[i] = 1$, we put points $\{4i-2, 4i-1\}$.*

▶ **Definition 15** (Blue Vectors). *For a vector $y \in \{0,1\}^d$, create a group of points $B(y)$: (i) if $y[i] = 0$ put points $\{4i-2, 4i-1\}$, and (ii) if $y[i] = 1$ put points $\{4i-3, 4i\}$.*

Next, we show that the above gadgets simulate the orthogonality.

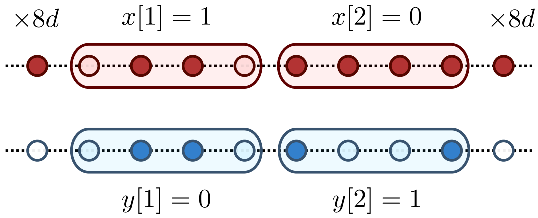

Figure 5 Gadgets for red and blue vectors in $d = 2$. The top figure shows $R(x)$ for $x = (1, 0)$, and the bottom figure illustrates $B(y)$ for $y = (0, 1)$. Since x and y are orthogonal, each blue point corresponds to a red point with the same coordinate.

▶ **Lemma 16.** *Let $x, y \in \{0, 1\}^d$ be d-dimensional vectors.*
1. *If x and y are orthogonal then* $\mathrm{EMD}(B(y), R(x)) = 0$.
2. *If x and y are not orthogonal then* $\mathrm{EMD}(B(y) + \tau, R(x)) \geqslant 1$ *for all* $\tau \in \mathbb{R}$.

Moreover, for every $\tau \in \mathbb{R}$, we have $\mathrm{EMD}(B(y) + \tau, R(x)) \geqslant |\tau|$. *If $|\tau| \geqslant 4d + 1$, then we even have* $\mathrm{EMD}(B(y) + \tau, R(x)) = |\tau| \cdot c_1 + c_2$, *where $c_1 = 2d$ and $c_2 = -4d^2 - d$.*

4.2 Reduction

Now we use the vector gadgets from the previous section to reduce from the Orthogonal Vectors problem to EMDuT. Specifically, given an OV instance $X, Y \subseteq \{0, 1\}^d$ of size n, we construct sets $B, R \subseteq \mathbb{R}$ such that from $\mathrm{EMDuT}(B, R)$ we can easily infer whether X, Y contains an orthogonal pair of vectors or not. Our reduction takes time $\mathcal{O}(nd)$ to construct the sets B, R, in particular the constructed sets have size $\mathcal{O}(nd)$. Hence, if there would be an algorithm computing $\mathrm{EMDuT}(B, R)$ in time $\mathcal{O}(|R|^{2-\delta})$ for some constant $\delta > 0$, then our reduction would yield an algorithm for OV running in time $\mathcal{O}((nd)^{2-\delta})$, which contradicts OVH (Hypothesis 12). That is, assuming OVH, $\mathrm{EMDuT}(B, R)$ cannot be computed in time $\mathcal{O}(|R|^{2-\delta})$ for any constant $\delta > 0$.

For the reduction we can assume that n is odd, because if n is even, one can simply add a vector consisting exclusively of 1s to both X and Y. We can also assume that $d \leqslant n$, since otherwise the naive algorithm for OV already runs in time $\mathcal{O}(n^2 d) = \mathcal{O}(nd^2)$. Our reduction constructs the following point sets, for $\Delta := 1000dn$:

- **Red Points**: For the i^{th} vector $x_i \in X$, we create five red gadgets $R(x_i)^{(1)}, \ldots, R(x_i)^{(5)}$. For each $k \in [5]$, we translate $R(x_i)^{(k)}$ by $(i + kn) \cdot (n - 1)\Delta$ and call it $(i + kn)^{\text{th}}$ *red cell.*
- **Blue Points**: For the j^{th} vector $y_j \in Y$, we create a blue gadget $B(y_j)$ and translate it by $j \cdot n\Delta$. This set of points is called the j^{th} *blue cell.*

We create five copies of red points for a technical reason that will become clear later (just three copies is enough, but then we would need to argue about two types of optimal translations in the analysis). We denote the set of all red points by R, and the set of all blue points by B. This concludes the construction. Observe that B, R can be constructed in time $\mathcal{O}(nd)$, as claimed, and that their coordinates are in $\{0, \ldots, \mathcal{O}(dn^3)\} \subseteq \{0, \ldots, \mathcal{O}(n^4)\}$. Let c_1 and c_2, where c_1, c_2 are the constants (that depend on d) from Lemma 16. Let

$$\Lambda := c_1 \Delta \cdot (n^2 - 1)/4 + c_2 \cdot (n - 1).$$

We now claim that the sets X, Y contain orthogonal vectors if and only if $\mathrm{EMDuT}(B, R) \leqslant \Lambda$. Thus, from the value $\mathrm{EMDuT}(B, R)$ we can then easily infer whether X, Y contain orthogonal vectors. In the full version [13], we analyze the properties of the construction, and formalize the proof. This leads to Theorem 13 as claimed.

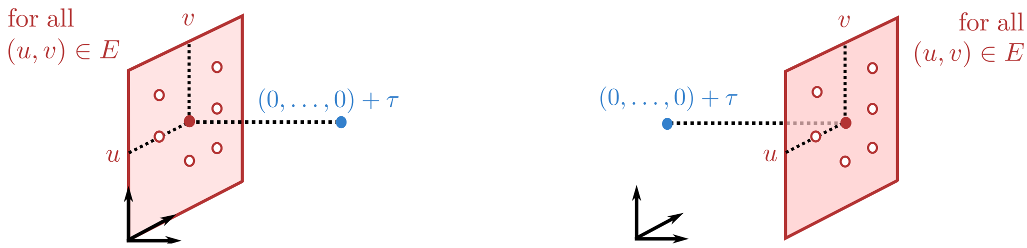

Figure 6 On the left the gadget $(B_{i,j}, R_{i,j})$, and on the right the gadget $(B'_{i,j}, R'_{i,j})$.

5 Lower bounds in higher dimension

In this section, we prove conditional lower bounds for approximating EMDuT with the L_1 or L_∞ norm. Our lower bounds assume the popular Exponential Time Hypothesis (ETH), which postulates that the 3-SAT problem on N variables cannot be solved in time $2^{o(N)}$ [28].

▶ **Theorem 17.** *Assuming ETH, there is no algorithm that, given $\varepsilon \in (0, 1)$ and $B, R \subseteq \mathbb{R}^d$ of size $|B| = |R| = n$, computes a $(1+\varepsilon)$-approximation of* $\mathrm{EMDuT}_1(B, R)$ *(or* $\mathrm{EMDuT}_\infty(B, R)$*) and runs in time $(\frac{n}{\varepsilon})^{o(d)}$.*

We prove our lower bounds by a reduction from the k-Clique problem: Given a graph $G = (V, E)$ with N nodes, decide whether there exist distinct nodes $v_1, \dots, v_k \in V$ such that $(v_i, v_j) \in E$ for all $1 \leqslant i < j \leqslant k$. Here, we always assume that k is constant. A naive algorithm solves the k-Clique problem in time $\mathcal{O}(N^k)$. It is well known that this running time cannot be improved to $N^{o(k)}$ assuming ETH.

▶ **Theorem 18** ([16])**.** *Assuming ETH, the k-Clique problem cannot be solved in time $N^{o(k)}$.*

In our lower bounds we will use the following lemma that combines gadgets $(B_1, R_1), \dots, (B_k, R_k)$ into a single instance (B, R) whose cost is essentially the total cost of all gadgets. To prove this lemma, we simply place the gadgets sufficiently far apart.

▶ **Lemma 19** (Gadget Combination Lemma)**.** *Let $1 \leqslant p \leqslant \infty$. Given sets $B_1, R_1, \dots, B_k, R_k \subset \mathbb{R}^d$ of total size n with $|B_i| \leqslant |R_i|$ for all $i \in [k]$, in time $\mathcal{O}(nd)$ we can compute sets $B, R \subset \mathbb{R}^d$ of total size n such that*

$$\mathrm{EMDuT}_p(B, R) = \min_{\tau \in \mathbb{R}^d} \sum_{i=1}^{k} \mathrm{EMD}_p(B_i + \tau, R_i).$$

Proof Sketch. For a sufficiently large number U we construct the sets $B := \bigcup_{i=1}^k B_i + (U \cdot i, 0, \dots, 0)$ and $R := \bigcup_{i=1}^k R_i + (U \cdot i, 0, \dots, 0)$, i.e., we place the gadgets sufficiently far apart. Then one can argue that any optimal matching must match points in B_i to points in R_i, and thus the EMDuT cost splits over the gadgets as claimed. ◀

In Section 5.1 we prove the lower bound for the L_1 norm in the asymmetric setting, i.e., we allow $|B|$ to be smaller than $|R| = n$. In the full version [13], we show that we can actually strengthen this lower bound to hold even in the symmetric setting $|B| = |R| = n$. Additionally, we prove the lower bound for the L_∞ norm in the symmetric setting.

5.1 Lower bound for L_1 asymmetric

In this section we prove Theorem 17 for the L_1 norm in the asymmetric setting, i.e., we relax the condition $|B| = |R|$ to $|B| \leqslant |R|$.

We are given a k-Clique instance $G = ([N], E)$. We set the dimension to $d \coloneqq k$. In what follows by $p_{i,u,j,v,b} \in \mathbb{R}^d$ we denote the point with coordinates, for $\ell \in [d]$,

$$(p_{i,u,j,v,b})_\ell = \begin{cases} u & \text{if } \ell = i, \\ v & \text{if } \ell = j, \\ b & \text{otherwise.} \end{cases}$$

We construct the following $2\binom{k}{2}$ gadgets. For any $1 \leqslant i < j \leqslant k$ we construct

$$B_{i,j} \coloneqq \{(0, \dots, 0)\}, \qquad R_{i,j} \coloneqq \{p_{i,u,j,v,0} \mid (u, v) \in E\},$$
$$B'_{i,j} \coloneqq \{(0, \dots, 0)\}, \qquad R'_{i,j} \coloneqq \{p_{i,u,j,v,N} \mid (u, v) \in E\}.$$

The cost of these gadgets has the following properties.[2]

▶ **Lemma 20.** *Let $1 \leqslant i < j \leqslant k$. For any $\tau \in \mathbb{R}^d$ we have*

$$\mathrm{EMD}_1(B_{i,j} + \tau, R_{i,j}) + \mathrm{EMD}_1(B'_{i,j} + \tau, R'_{i,j}) \geqslant (d-2)N,$$

and equality holds if $\tau \in [N]^d$ and $(\tau_i, \tau_j) \in E$. Moreover, for any $\tau \in \mathbb{R}^d$ with $(\lfloor \tau_i \rceil, \lfloor \tau_j \rceil) \notin E$ we have

$$\mathrm{EMD}_1(B_{i,j} + \tau, R_{i,j}) + \mathrm{EMD}_1(B'_{i,j} + \tau, R'_{i,j}) \geqslant (d-2)N + 1.$$

Proof. Observe that

$$\begin{aligned} \mathrm{EMD}_1(B_{i,j} + \tau, R_{i,j}) &= \min_{(u,v) \in E} \|(0, \dots, 0) + \tau - p_{i,u,j,v,0}\|_1 \\ &= \min_{(u,v) \in E} |\tau_i - u| + |\tau_j - v| + \sum_{\ell \neq i,j} |\tau_\ell| \\ &\geqslant \min_{(u,v) \in E} |\tau_i - u| + |\tau_j - v| + \sum_{\ell \neq i,j} \tau_\ell, \end{aligned}$$

where equality holds if $\tau \in [N]^d$. We similarly bound

$$\mathrm{EMD}_1(B'_{i,j} + \tau, R'_{i,j}) \geqslant \min_{(u,v) \in E} |\tau_i - u| + |\tau_j - v| + \sum_{\ell \neq i,j} N - \tau_\ell,$$

where equality holds if $\tau \in [N]^d$. Summing up and bounding the absolute values by 0, we obtain

$$\mathrm{EMD}_1(B_{i,j} + \tau, R_{i,j}) + \mathrm{EMD}_1(B'_{i,j} + \tau, R'_{i,j}) \geqslant (d-2)N.$$

If $\tau \in [N]^d$ and $(\tau_i, \tau_j) \in E$, then we can pick u, v with $|\tau_i - u| + |\tau_j - v| = 0$, and we obtain equality.

Moreover, for any $\tau \in \mathbb{R}^d$ with $(\lfloor \tau_i \rceil, \lfloor \tau_j \rceil) \notin E$, note that since (τ_i, τ_j) has L_∞ distance at most 1/2 to $(\lfloor \tau_i \rceil, \lfloor \tau_j \rceil)$, it has L_∞ distance at least 1/2 to any other grid point. In particular, (τ_i, τ_j) has L_∞ distance at least 1/2 to any $(u, v) \in E$. Since L_∞ distance lower bounds L_1 distance, we obtain $\min_{(u,v) \in E} |\tau_i - u| + |\tau_j - v| \geqslant 1/2$. This yields

$$\begin{pmatrix} \mathrm{EMD}_1(B_{i,j} + \tau, R_{i,j}) + \\ \quad \mathrm{EMD}_1(B'_{i,j} + \tau, R'_{i,j}) \end{pmatrix} \geqslant 2 \min_{(u,v) \in E} \Big(|\tau_i - u| + |\tau_j - v| \Big) + (d-2)N \geqslant (d-2)N + 1. \blacktriangleleft$$

[2] Recall that $\lfloor x \rceil$ denotes the closest integer to x, while $[x]$ denotes $\{1, \dots, x\}$.

We apply the Gadget Combination Lemma to the gadgets $B_{i,j}, R_{i,j}, B'_{i,j}, R'_{i,j}$ for $1 \leqslant i < j \leqslant d$. The EMDuT_1 of the resulting point sets B, R is the sum of the costs of the gadgets. Hence, the above lemma implies the following. If G has a k-Clique $v_1, \ldots, v_k$, then $\tau \coloneqq (v_1, \ldots, v_k) \in [N]^d$ has a total cost of $\binom{d}{2} \cdot (d-2)N =: \Lambda$. On the other hand, if G has no k-Clique, then for any $\tau \in \mathbb{R}^d$ there exist $1 \leqslant i < j \leqslant k$ with $(\lfloor \tau_i \rceil, \lfloor \tau_j \rceil) \notin E$ (as otherwise $(\lfloor \tau_1 \rceil, \ldots, \lfloor \tau_k \rceil)$ would form a k-Clique). Thus, each pair of gadgets contributes cost at least $(d-2)N$, and at least one pair of gadgets contributes cost at least $(d-2)N+1$, so the total cost is at least $\binom{d}{2} \cdot (d-2)N + 1 = \Lambda + 1$.

For any $\varepsilon < 1/\Lambda$, a $(1+\varepsilon)$-approximation algorithm for EMDuT_1 could distinguish cost at most Λ and cost at least $\Lambda + 1$, and thus would solve the k-Clique problem. Hence, if we would have a $(1+\varepsilon)$-approximation algorithm for EMDuT_1 running in time $(n/\varepsilon)^{o(d)}$, then by setting $\varepsilon \coloneqq 0.9/\Lambda$ and observing $n = \mathcal{O}(N^2)$, $1/\varepsilon = \mathcal{O}(\Lambda) = \mathcal{O}(N)$, and $d = k$, we would obtain an algorithm for k-Clique running in time $(n/\varepsilon)^{o(d)} = \mathcal{O}(N^3)^{o(k)} = N^{o(k)}$, which contradicts ETH by Theorem 18.

6 Algorithms in higher dimensions

Given two sets R and B of n points in the plane, Eppstein et al. [22] show how to compute a translation τ^* minimizing $\mathrm{EMDuT}_1(B, R)$ with respect to the L_1-distance in $\mathcal{O}(n^6 \log^3 n)$ time. We observe that their result can be generalized to point sets in arbitrary dimension d, leading to an $\mathcal{O}(m^d n^{d+2} \log^{d+2} n)$ time algorithm.

Furthermore, we show that our approach can also be used to obtain an $\mathcal{O}(m^d n^{d+2} \log^{d+2} n)$ time algorithm for finding a translation that minimizes $\mathrm{EMDuT}_\infty(B, R)$, i.e. the Earth Mover's Distance with respect to the L_∞ distance. For point sets in $\mathbb{R}^2$, this immediately follows by "rotating the plane by 45°" and using the algorithm for L_1. For higher dimensions this trick is no longer immediately applicable. However, we show that our algorithm can also directly be applied to the L_∞ distance, even for point sets in $\mathbb{R}^d$, with $d > 2$.

Earth Mover's Distance without Translation. We first describe an algorithm to compute $\mathrm{EMD}_p(B, R)$ in $\mathbb{R}^d$. Note that we assume to work in the Real RAM model, hence we need a strongly-polynomial algorithm. Naively, one can achieve that in $\mathcal{O}(m^2 n)$ time by computing the bipartite graph, and solving maximum weight matching in bipartite graph in strongly polynomial time by Edmonds and Karp [21]. Here, however, we can use the fact that points are in $\mathbb{R}^d$. To the best of our knowledge, the best algorithm in this setting is due to Vaidya [40]. However, he only considers the case when both point sets are in $\mathbb{R}^2$ and have size $n = m$ in $\mathbb{R}^2$. He shows that one can compute $\mathrm{EMD}_p(B, R)$ (with $p \in \{1, \infty\}$) in $\mathcal{O}(n^2 \log^3 n)$ time in this setting. Furthermore, he states (without proof) that for point sets in $\mathbb{R}^d$, that the running time increases by at most $\mathcal{O}(\log^d n)$. We briefly sketch the algorithm and fill in the missing details for the higher-dimensional setting in the full version [13].

▶ **Theorem 21.** *Given a set B of m points in $\mathbb{R}^d$, and a set of $n \geqslant m$ red points in $\mathbb{R}^d$, there is an $\mathcal{O}(n^2 \log^{d+2} n)$ time algorithm to compute $\mathrm{EMD}_p(B, R)$, for $p \in \{1, \infty\}$.*

Earth Mover's Distance under Translation in L_1. The sets B and R are aligned in dimension i, or *i-aligned* for short, if there is a pair of points $b \in B, r \in R$ for which $b_i = r_i$. Eppstein et al. [22] show that for two point sets in $\mathbb{R}^2$, there exists an optimal translation τ^* that aligns B and R in both dimensions. They explicitly consider all $\mathcal{O}((nm)^2)$ translations that both 1-align and 2-align $B + \tau$ and R. For each such a translation τ, computing an optimal matching can then be done in $\mathcal{O}(n^2 \log^3 n)$ time [40], thus leading to an $\mathcal{O}(n^4 m^2 \log^3 n)$ time algorithm. We now argue that we can generalize the above result to higher dimensions.

▶ **Theorem 22.** *Given B and R we can find an optimal translation τ^* that realizes* $\mathrm{EMDuT}_1(B,R)$ *in* $\mathcal{O}(m^d n^{d+2} \log^{d+2} n)$ *time.*

Proof. Recall the definition of the cost function

$$\mathcal{D}_{B,R,1}(\phi,\tau) = \sum_{b\in B} L_1(b+\tau,\phi(b)) = \sum_{b\in B}\sum_{i=1}^{d} |b_i+\tau_i-\phi(b)_i|.$$

For a fixed matching ϕ, this is a piecewise linear function in τ. In particular, $\mathcal{D}_{B,R,1}(\phi,\tau)$ is a sum of piecewise linear functions $f_{b,i}(\tau) = |b_i+\tau_i-\phi(b)_i|$. For each such a function there is a hyperplane $h_{b,\phi(b),i}$ in $\mathbb{R}^d$ given by the equation $\tau_i+b_i-\phi(b)_i = 0$, so that for a point (translation) $\tau\in\mathbb{R}^d$ on one side of (or on) the hyperplane, $f_{b,i}(\tau)$ is linear in τ (i.e. on one side we have $f(\tau) = \tau_i+b_i-\phi(b)_i$, whereas on the other side we have $f(\tau) = -\tau_i-b_i+\phi(b)_i$). Let $H_\phi = \{h_{b,\phi(b),i} \mid b\in B, i\in\{1,\ldots,d\}\}$ denote the set of all such hyperplanes, and consider the arrangement $\mathcal{A}(H_\phi)$. It follows that in each cell of $\mathcal{A}(H_\phi)$, the function $\mathcal{D}_{B,R,1}(\phi,\tau)$ is a linear function in τ, and that $\mathcal{D}_{B,R,1}(\phi,\tau)$ thus has its minimum at a vertex of $\mathcal{A}(H_\phi)$.

We extend the set of hyperplanes H_ϕ to include the hyperplane $h_{b,r,i}$ for *every* pair $(b,r)\in B\times R$, and every $i\in\{1,\ldots,d\}$, rather than just the pairs $(b,\phi(b))$. Let H be the resulting set. A minimum of $\mathcal{D}_{B,R,1}(\phi,\tau)$ still occurs at a vertex of $\mathcal{A}(H)$ (as $\mathcal{A}(H)$ includes all vertices of $\mathcal{A}(H_\phi)$). Moreover, observe that H now actually contains the hyperplanes H_ϕ, for *every* matching $\phi\in\Phi$, so also those of an optimal matching ϕ^*. It thus follows that such a global minimum $\mathcal{D}_1(\phi^*,\tau^*)$ occurs at a vertex τ^* of $\mathcal{A}(H)$.

So, to compute an optimal matching ϕ^* and its τ^* (and thus $\mathrm{EMDuT}(B,R)$) we can

1. explicitly compute (all vertices of) $\mathcal{A}(H)$,
2. for each such a vertex $\tau\in\mathcal{A}(H)$ (which is some candidate translation), compute an optimal matching ϕ_τ between the sets $B+\tau$ and R, and
3. report the matching (and corresponding translation) that minimizes total cost.

The set H contains mnd hyperplanes, and thus $\mathcal{A}(H)$ contains $\mathcal{O}((mnd)^d) = \mathcal{O}(m^d n^d)$ vertices. Computing $\mathcal{A}(H)$ takes $\mathcal{O}(m^d n^d)$ time [19, 20]. For each such a vertex (translation), we can compute an optimal matching in $\mathcal{O}(n^2\log^{d+2} n))$ time using the algorithm from Theorem 21. This thus yields an $\mathcal{O}(m^d n^{d+2}\log^{d+2} n)$ time algorithm in total. ◀

Earth Mover's Distance under Translation in L_∞. In the full version [13] we use a similar approach as in Theorem 22. We prove that there is a set H of $\mathcal{O}(mnd^2)$ hyperplanes in $\mathbb{R}^d$, so that for any matching ϕ, there is a minimum cost translation that is a vertex of the arrangement $\mathcal{A}(H)$. We can thus again compute such an optimal matching (and the translation) by trying all $\mathcal{O}(m^d n^d)$ vertices. This yields the following result, thereby also establishing Theorem 3.

▶ **Theorem 23.** *Given B and R we can find an optimal translation τ^* that realizes* $\mathrm{EMDuT}_\infty(B,R)$ *in* $\mathcal{O}(m^d n^{d+2}\log^{d+2} n)$ *time.*

References

1 Pankaj K. Agarwal, Hsien-Chih Chang, Sharath Raghvendra, and Allen Xiao. Deterministic, near-linear ϵ-approximation algorithm for geometric bipartite matching. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1052–1065. ACM, 2022. doi:10.1145/3519935.3519977.

2 Pankaj K. Agarwal, Sariel Har-Peled, Micha Sharir, and Yusu Wang. Hausdorff distance under translation for points and balls. *ACM Trans. Algorithms*, 6(4):71:1–71:26, 2010. doi:10.1145/1824777.1824791.

3 Pankaj K. Agarwal, Sharath Raghvendra, Pouyan Shirzadian, and Rachita Sowle. An Improved ϵ-Approximation Algorithm for Geometric Bipartite Matching. In Artur Czumaj and Qin Xin, editors, *18th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2022, June 27-29, 2022, Tórshavn, Faroe Islands*, volume 227 of *LIPIcs*, pages 6:1–6:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPICS.SWAT.2022.6`.

4 Helmut Alt, Christian Knauer, and Carola Wenk. Matching Polygonal Curves with Respect to the Fréchet Distance. In Afonso Ferreira and Horst Reichel, editors, *STACS 2001, 18th Annual Symposium on Theoretical Aspects of Computer Science, Dresden, Germany, February 15-17, 2001, Proceedings*, volume 2010 of *Lecture Notes in Computer Science*, pages 63–74. Springer, 2001. `doi:10.1007/3-540-44693-1_6`.

5 Alexandr Andoni, Khanh Do Ba, Piotr Indyk, and David P. Woodruff. Efficient Sketches for Earth-Mover Distance, with Applications. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 324–330. IEEE Computer Society, 2009. `doi:10.1109/FOCS.2009.25`.

6 Alexandr Andoni, Piotr Indyk, and Robert Krauthgamer. Earth mover distance over high-dimensional spaces. In Shang-Hua Teng, editor, *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008*, pages 343–352. SIAM, 2008. URL: `http://dl.acm.org/citation.cfm?id=1347082.1347120`.

7 Alexandr Andoni, Aleksandar Nikolov, Krzysztof Onak, and Grigory Yaroslavtsev. Parallel algorithms for geometric graph problems. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 574–583. ACM, 2014. `doi:10.1145/2591796.2591805`.

8 Rinat Ben Avraham, Haim Kaplan, and Micha Sharir. A faster algorithm for the discrete Fréchet distance under translation. *CoRR*, abs/1501.03724, 2015. `arXiv:1501.03724`.

9 Chandrajit L. Bajaj. The algebraic degree of geometric optimization problems. *Discret. Comput. Geom.*, 3:177–191, 1988. `doi:10.1007/BF02187906`.

10 Karl Bringmann, Marvin Künnemann, and André Nusser. When Lipschitz Walks Your Dog: Algorithm Engineering of the Discrete Fréchet Distance Under Translation. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 25:1–25:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPICS.ESA.2020.25`.

11 Karl Bringmann, Marvin Künnemann, and André Nusser. Discrete Fréchet Distance under Translation: Conditional Hardness and an Improved Algorithm. *ACM Trans. Algorithms*, 17(3):25:1–25:42, 2021. `doi:10.1145/3460656`.

12 Karl Bringmann and André Nusser. Translating Hausdorff Is Hard: Fine-Grained Lower Bounds for Hausdorff Distance Under Translation. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry, SoCG 2021, June 7-11, 2021, Buffalo, NY, USA (Virtual Conference)*, volume 189 of *LIPIcs*, pages 18:1–18:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPICS.SOCG.2021.18`.

13 Karl Bringmann, Frank Staals, Karol Węgrzycki, and Geert van Wordragen. Fine-grained complexity of earth mover's distance under translation. *CoRR*, abs/2403.04356, 2024. URL: `https://arxiv.org/abs/2403.04356`.

14 Sergio Cabello, Panos Giannopoulos, Christian Knauer, and Günter Rote. Matching point sets with respect to the Earth Mover's Distance. *Comput. Geom.*, 39(2):118–133, 2008. `doi:10.1016/J.COMGEO.2006.10.001`.

15 Timothy M. Chan. Minimum L_∞ Hausdorff Distance of Point Sets Under Translation: Generalizing Klee's Measure Problem. In Erin W. Chambers and Joachim Gudmundsson, editors, *39th International Symposium on Computational Geometry, SoCG 2023, June 12-15, 2023, Dallas, Texas, USA*, volume 258 of *LIPIcs*, pages 24:1–24:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. `doi:10.4230/LIPICS.SOCG.2023.24`.

16 Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.*, 72(8):1346–1367, 2006. doi:10.1016/J.JCSS.2006.04.007.

17 Michael B. Cohen, Yin Tat Lee, Gary L. Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 9–21. ACM, 2016. doi:10.1145/2897518.2897647.

18 Scott D. Cohen and Leonidas J. Guibas. The Earth Mover's Distance under Transformation Sets. In *Proceedings of the International Conference on Computer Vision, Kerkyra, Corfu, Greece, September 20-25, 1999*, pages 1076–1083. IEEE Computer Society, 1999. doi:10.1109/ICCV.1999.790393.

19 Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications.* Springer, Berlin, 3rd edition, 2008.

20 Herbert Edelsbrunner, Joseph O'Rourke, and Raimund Seidel. Constructing arrangements of lines and hyperplanes with applications. *SIAM J. Comput.*, 15(2):341–363, 1986. doi:10.1137/0215024.

21 Jack Edmonds and Richard M Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2):248–264, 1972.

22 David Eppstein, Marc J. van Kreveld, Bettina Speckmann, and Frank Staals. Improved grid map layout by point set matching. *Int. J. Comput. Geom. Appl.*, 25(2):101–122, 2015. doi:10.1142/S0218195915500077.

23 Omrit Filtser and Matthew J. Katz. Algorithms for the discrete Fréchet distance under translation. *J. Comput. Geom.*, 11(1):156–175, 2020. doi:10.20382/JOCG.V11I1A7.

24 Emily Fox and Jiashuai Lu. A deterministic near-linear time approximation scheme for geometric transportation. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 1301–1315. IEEE, 2023. doi:10.1109/FOCS57990.2023.00078.

25 Kyle Fox and Jiashuai Lu. A near-linear time approximation scheme for geometric transportation with arbitrary supplies and spread. *J. Comput. Geom.*, 13(1), 2022. doi:10.20382/JOCG.V13I1A8.

26 Daniel P. Huttenlocher and Klara Kedem. Computing the Minimum Hausdorff Distance for Point Sets Under Translation. In Raimund Seidel, editor, *Proceedings of the Sixth Annual Symposium on Computational Geometry, Berkeley, CA, USA, June 6-8, 1990*, pages 340–349. ACM, 1990. doi:10.1145/98524.98599.

27 Daniel P. Huttenlocher, William Rucklidge, and Gregory A. Klanderman. Comparing images using the Hausdorff distance under translation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 1992, Proceedings, 15-18 June, 1992, Champaign, Illinois, USA*, pages 654–656. IEEE, 1992. doi:10.1109/CVPR.1992.223209.

28 Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/JCSS.2000.1727.

29 Piotr Indyk. A near linear time constant factor approximation for Euclidean bichromatic matching (cost). In Nikhil Bansal, Kirk Pruhs, and Clifford Stein, editors, *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 39–42. SIAM, 2007. URL: http://dl.acm.org/citation.cfm?id=1283383.1283388.

30 Minghui Jiang, Ying Xu, and Binhai Zhu. Protein Structure-structure Alignment with Discrete Fréchet Distance. *J. Bioinform. Comput. Biol.*, 6(1):51–64, 2008. doi:10.1142/S0219720008003278.

31 Andrey Boris Khesin, Aleksandar Nikolov, and Dmitry Paramonov. Preconditioning for the geometric transportation problem. *J. Comput. Geom.*, 11(2):234–259, 2020. doi:10.20382/JOCG.V11I2A11.

32 Oliver Klein and Remco C. Veltkamp. Approximation algorithms for the Earth mover's distance under transformations using reference points. In *(Informal) Proceedings of the 21st European Workshop on Computational Geometry, Eindhoven, The Netherlands, March 9-11, 2005*, pages 53–56. Technische Universiteit Eindhoven, 2005. URL: `http://www.win.tue.nl/EWCG2005/Proceedings/14.pdf`.

33 Christian Knauer, Klaus Kriegel, and Fabian Stehn. Minimizing the weighted directed Hausdorff distance between colored point sets under translations and rigid motions. *Theor. Comput. Sci.*, 412(4-5):375–382, 2011. `doi:10.1016/J.TCS.2010.03.020`.

34 Christian Knauer and Marc Scherfenberg. Approximate Nearest Neighbor Search under Translation Invariant Hausdorff Distance. *Int. J. Comput. Geom. Appl.*, 21(3):369–381, 2011. `doi:10.1142/S0218195911003706`.

35 Axel Mosig and Michael Clausen. Approximately matching polygonal curves with respect to the Fréchet distance. *Comput. Geom.*, 30(2):113–127, 2005. `doi:10.1016/J.COMGEO.2004.05.004`.

36 Mark H. Overmars and Jan van Leeuwen. Maintenance of configurations in the plane. *J. Comput. Syst. Sci.*, 23(2):166–204, 1981. `doi:10.1016/0022-0000(81)90012-X`.

37 Dhruv Rohatgi. Conditional Hardness of Earth Mover Distance. In Dimitris Achlioptas and László A. Végh, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2019, September 20-22, 2019, Massachusetts Institute of Technology, Cambridge, MA, USA*, volume 145 of *LIPIcs*, pages 12:1–12:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. `doi:10.4230/LIPICS.APPROX-RANDOM.2019.12`.

38 Günter Rote. Computing the Minimum Hausdorff Distance Between Two Point Sets on a Line Under Translation. *Inf. Process. Lett.*, 38(3):123–127, 1991. `doi:10.1016/0020-0190(91)90233-8`.

39 Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *Int. J. Comput. Vis.*, 40(2):99–121, 2000. `doi:10.1023/A:1026543900054`.

40 Pravin M. Vaidya. Geometry helps in matching. *SIAM J. Comput.*, 18(6):1201–1225, 1989. `doi:10.1137/0218080`.

41 Virginia Vassilevska-Williams. On Some Fine-Grained Questions in Algorithms and Complexity. In *Proceedings of the International Congress of Mathematicians (ICM 2018)*, pages 3447–34, 2018.

42 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2-3):357–365, 2005. `doi:10.1016/j.tcs.2005.09.023`.