Computing the bridge length: the key ingredient in a continuous isometry classification of periodic point sets

Jonathan McManus[a]* and Vitaliy Kurlin[b]

[a] *Computer Science department and Materials Innovation Factory, University of Liverpool, Liverpool L69 3BX UK. E-mail: {j.d.mcmanus,vkurlin}@liverpool.ac.uk*

Abstract

The fundamental model of any periodic crystal is a periodic set of points at all atomic centres. Since crystal structures are determined in a rigid form, their strongest equivalence is rigid motion (composition of translations and rotations) or isometry (also including reflections). The recent isometry classification of periodic point sets used a complete invariant isoset whose size essentially depends on the bridge length, defined as the minimum 'jump' that suffices to connect any points in the given periodic set.

We propose a practical algorithm to compute the bridge length of any periodic point set given by a motif of points in a periodically translated unit cell. The algorithm has been tested on a large crystal dataset and is required for an efficient continuous classification of all periodic crystals. The exact computation of the bridge length is a key step to realising the inverse design of materials from new invariant values.

1. Introduction: practical motivations and the problem statement

All solid crystalline materials can be modelled at the atomic level as periodic sets of points (with the chemical attributes if desired) at all atomic centres, defined below.

Definition 1 (lattice, unit cell, motif, periodic point set)**.** *Any vectors* $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}$ *that form a linear basis of* $\mathbb{R}^n$ *generate the* lattice $\Lambda = \{\sum\limits_{i=1}^{n} c_i \boldsymbol{v_i} \mid c_i \in \mathbb{Z}\}$ *and the* unit cell $U = \{\sum\limits_{i=1}^{n} t_i \boldsymbol{v_i} \mid 0 \le t_i < 1\}$*. For any finite set of points* $M \subset U$ *(called a* motif*), the* periodic point set $S = \Lambda + M = \{\boldsymbol{v} + p \mid \boldsymbol{v} \in \Lambda, p \in M\}$ *is a union of finitely many lattices whose origins are shifted to all points of the motif* M*, see Fig. 1 (left).*

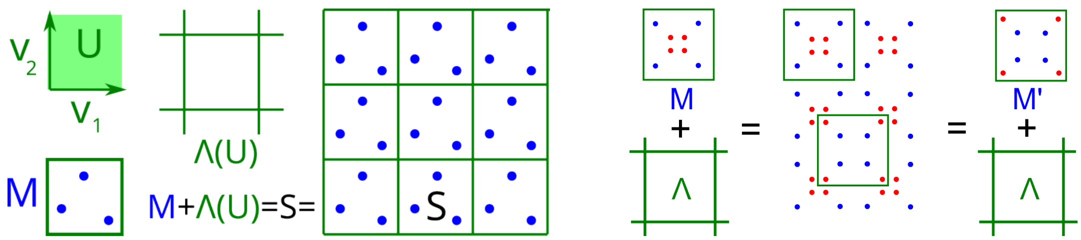

Fig. 1. **Left**: the orthonormal basis v_1, v_2 generates the green lattice Λ and the unit cell U containing the blue motif M of three points. The periodic point set $S = \Lambda + M$ is obtained by periodically repeating M along all vectors of Λ. **Right**: different motifs M, M' in the same cell generate periodic sets that differ by only translation.

Any unit cell U is a parallelepiped on basis vectors $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}$. If we translate the unit cell U by all vectors $v \in \Lambda$, the resulting cells tile $\mathbb{R}^n$ without overlaps. Motif points represent atomic centres in a real crystal. The same lattice can be generated by infinitely many different bases that are all related under multiplication by $n \times n$ matrices with integer elements and determinant 1. Even if we fix a basis of $\mathbb{R}^n$ and hence a unit cell U, different motifs in U can define periodic point sets that differ only by Euclidean *isometry* defined as any distance-preserving transformation of $\mathbb{R}^n$.

Since crystal structures are determined in a rigid form, their slightly stronger equivalence is *rigid motion* defined as any orientation-preserving isometry without reflections or as a composition of translations and rotations. After many years of discussing definitions of a "crystal" (Brock, 2021), a *crystal structure* was recently defined in the periodic case by (Anosova *et al.*, 2024) as a rigid class of all periodic sets of atoms.

This rigid class consists of all (infinitely many) periodic point sets that are equivalent to each other under some rigid motions. Though two mirror images of a crystal belong to the same isometry class, they can be distinguished by a suitably defined sign of crystal orientation, which makes a classification of periodic crystals under isometry practically sufficient. However, almost any perturbation of atoms discontinuously changes some inter-atomic distances and hence the isometry class with all cell-based descriptors such as symmetry groups. Even in dimension 1, for any integer $m > 0$ and small $\epsilon > 0$ all integers of the sequence $\mathbb{Z}$ with period 1 are ϵ-pointwise close to the sequence with the motif $M = \{0, 1+\epsilon, \dots, m+\epsilon\}$ and arbitrarily large period $m+1$.

This inherent discontinuity of all cell-based descriptors was resolved by Pointwise Distance Distributions (PDD) in (Widdowson & Kurlin, 2022), which defined geographic-style coordinates on the first continuous projections of the Cambridge Structural Database (CSD) in (Widdowson & Kurlin, 2024). Though PDDs distinguish all periodic crystals in the CSD within an hour on a modest desktop, the only theoretically complete and continuous invariant descriptor that uniquely identifies any periodic point set under isometry in $\mathbb{R}^n$ is the *isoset* from (Anosova & Kurlin, 2021). This isoset invariant requires the bridge length whose definition is reminded below.

Definition 2 (bridge length $\beta(S)$)**.** *For any finite or periodic set of points $S \subset \mathbb{R}^n$, the* bridge length $\beta(S)$ *is the minimum distance such that any points $p, q \in S$ can be connected by a finite sequence of points $p = p_1, p_2, \dots, p_k = q$ in S, such that every Euclidean distance $|p_i - p_{i+1}| \leq \beta(S)$ for all $i = 1, \dots, k-1$.*

Equivalently, the *bridge length* is the minimum double radius such that the union of the closed balls of the radius $\frac{1}{2}\beta(S)$ around all points of S is connected. The lattice $S = \mathbb{Z}^n$ of all points with integer coordinates has $\beta(S) = 1$. If we add to $\mathbb{Z}^3$ all points whose coordinates are half-integer, the resulting BCC (body-centred cubic) lattice has $\beta(S) = \frac{\sqrt{3}}{2}$ (half-diagonal of the unit cube). The bridge length β also makes sense for any finite set S, e.g. S coincides with a motif M. Now we state the main problem.

Problem 3. *Design an algorithm to compute the bridge length in polynomial time in the motif size of any periodic point set S with a fixed unit cell in $\mathbb{R}^n$.*

For any finite set M of points in $\mathbb{R}^n$, a *Minimum Spanning Tree* $\mathrm{MST}(M)$ is a tree that has the vertex set M and a minimum total length of straight-line edges measured by Euclidean distance. By Definition 2 the bridge length $\beta(M)$ of any finite set $M \subset \mathbb{R}^n$ of points equals the length of the longest edge of $\mathrm{MST}(M)$.

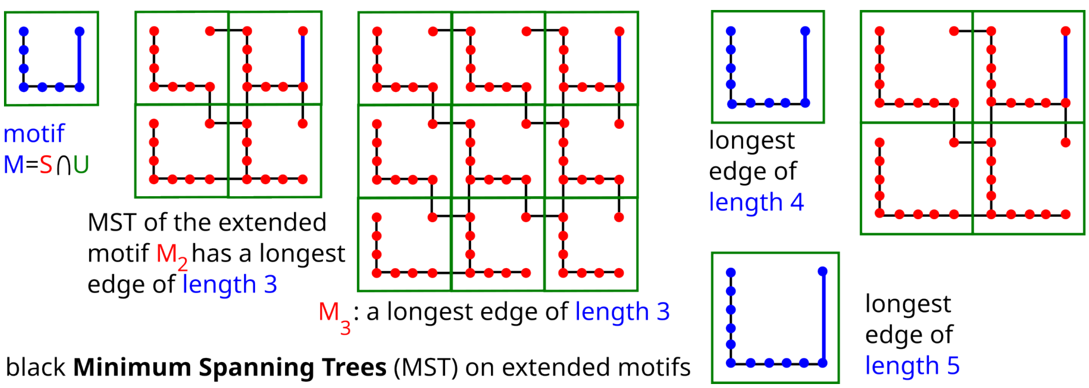

Fig. 2. All Minimum Spanning Trees on extended motifs of a periodic point set S have the longest edge (in blue) of length 3, which could be made arbitrarily long, relative to a preserved minimum inter-point distance of 1 and bridge length $\beta(S) = 2$ due to shorter edges from the top right point in every cell across a cell boundary.

For any periodic point set S with a unit cell U on a basis $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}$ in $\mathbb{R}^n$, one can consider the extended motifs $M_k = S \cap U_k$, where the extended cell U_k is defined by the basis $kv_1, \ldots, kv_n$ for any integer $k > 1$. The Minimum Spanning Trees provide the upper bounds $\beta(S) \leq \beta(M_k)$ for $k > 1$, which can be unnecessarily high, see Fig. 2, so Problem 3 is much harder for periodic sets than for finite sets of points.

Definition 4 (parameters $r(U)$, $R(S)$, and $a(U)$)**.** *For any periodic point set $S \subset \mathbb{R}^n$, (Anosova & Kurlin, 2022)[Lemma 3.7] provided another upper bound $\beta(S) \leq r(U) =$*

$\max\{b, \frac{d}{2}\}$, *where* $b = \max\limits_{i=1,\dots,n} |\boldsymbol{v_i}|$ *is the maximum edge-length of a unit cell* U *of* S *with a basis* $\boldsymbol{v_1}, \dots, \boldsymbol{v_n}$, *and* d *is the length of the longest diagonal of* U. *The upper bound* $r(U)$ *can be improved to the covering radius* $R(S)$ *defined as the smallest radius such that the union of closed balls of the radius* $R(S)$ *around all points of* S *covers* $\mathbb{R}^n$.

For any unit cell $U \subset \mathbb{R}^n$, *let* $h(U)$ *be the shortest* height *of* U, *which can be computed as* $h = \text{vol}(U)/\max\limits_{1\leq i<j\leq n} |\boldsymbol{v_i} \times \boldsymbol{v_j}|$. *Define the* aspect ratio $a(U) = r(U)/h(U)$.

Main Theorem 5 below guarantees an exact computation of the bridge length $\beta(S)$ in a time that only quadratically depends on the motif size m of a periodic set S.

Theorem 5. *For any periodic point set* $S \subset \mathbb{R}^n$ *with a motif of* m *points in a unit cell* U, *the bridge length* $\beta(S)$ *can be computed in time* $O(m^2 a(U)^n N)$, *where* N *is the time complexity of the Smith Normal Form,* $a(U)$ *is the aspect ratio from Definition 4.*

2. Auxiliary concepts for the bridge length algorithm

This section introduces a few auxiliary structures to describe the exact algorithm for the bridge length in section 3 and to prove main Theorem 5 in section 4.

Definition 6 (periodic graph)**.** *Let* $S \subset \mathbb{R}^n$ *be a periodic point set with a lattice* Λ. *A* periodic graph $G \subset \mathbb{R}^n$ *is an infinite graph with the vertex set* S *and straight-line edges such that the translation by any vector* $\boldsymbol{v} \in \Lambda$ *defines a* self-isomorphism *of* G, *which is a bijection* $S \to S$ *that also induces a bijection on the edges of* G, *see Fig. 3.*

If straight-line edges meet at interior points, they are not considered vertices of G.

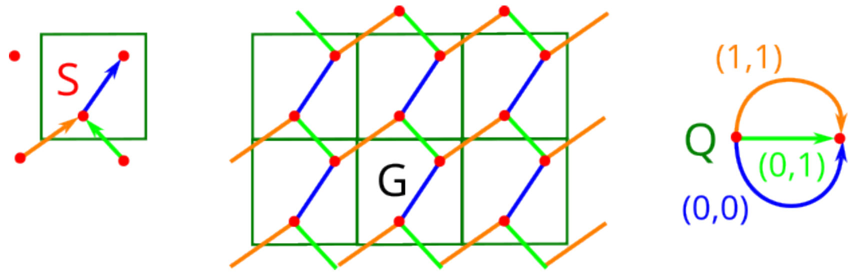

Fig. 3. The periodic graph $G \subset \mathbb{R}^2$ on a periodic point set S has the labelled quotient graph Q with translational vectors on directed edges, see Definitions 6, 7, 8.

Fig. 3 shows a connected periodic graph G but G can also be disconnected. For example, let S be the square lattice $\mathbb{Z}^2$, then the graph G consisting of all horizontal edges connecting the points (m, n) and $(m+1, n)$ for $m, n \in \mathbb{Z}$ is periodic but not connected. If we add to G all vertical edges connecting (m, n) and $(m, n+1)$ for $m, n \in \mathbb{Z}$, the resulting infinite square grid is a connected periodic graph on $\mathbb{Z}^2$.

Definition 7 (quotient graph)**.** *Let G be a periodic graph on a periodic point set S with a lattice Λ in $\mathbb{R}^n$. Two points of S (also vertices or edges of G) are called Λ-equivalent if they are related by a translation along a vector $\boldsymbol{v} \in \Lambda$. The* quotient graph *$G/\Lambda$ is an abstract undirected graph obtained as the quotient of G under the Λ-equivalence. Then G is called a* lifted graph *of G/Λ. Any vertex of G/Λ is a Λ-equivalence class $p + \Lambda$ represented by a point $p \in S$. Any edge e of the quotient graph G/Λ is a Λ-equivalence class $[p, q] + \Lambda$ represented by a straight-line edge $[p, q]$ of G. We define the length of any edge e in G/Λ as the Euclidean distance $|p - q|$.*

The quotient graph G/Λ can have multiple edges between the same pair of vertices as shown in Fig. 3, which all can be distinguished by the labels defined below.

Definition 8 (labelled quotient graph)**.** *Let $S \subset \mathbb{R}^n$ be a periodic point set with a lattice Λ defined by a basis $\boldsymbol{v_1}, \dots, \boldsymbol{v_n}$. Let G be a periodic graph on S. For an edge e of the quotient graph G/Λ, choose any of two directions and a representative edge*

$[p, q]$ in the lifted graph G. Let $U(p), U(q)$ be the unit cells containing p, q, respectively. There is a unique vector $\boldsymbol{v} = \sum_{i=1}^{n} c_i \boldsymbol{v_i} \in \Lambda$ such that $U(q) = U(p) + v$ and $c_i \in \mathbb{Z}$.

A labelled quotient graph (LQG) *is G/Λ whose every edge e has a direction (say, from the Λ-equivalence class of p to Λ-equivalence class of q) and the* translational vector $\boldsymbol{v}(e) = (c_1, \dots, c_n) \in \mathbb{Z}^n$*, see Fig. 3. Changing the direction of e multiplies each coordinate of $\boldsymbol{v}(e)$ by (-1). An equivalence of LQGs is a composition of a graph isomorphism and changes in edge directions that match all translational vectors.*

In crystallography, labelled quotient graphs have been studied by many authors. Section 6 in (Chung *et al.*, 1984) generated 3-periodic nets by considering used LQGs whose translational vectors have coordinates $\{-1, 0, 1\}$. Section 2 in (Cohen & Megiddo, 1990) described an algorithm to find connected components of a fixed periodic graph in terms of its LQG. Proposition 5.1 in (Eon, 2011) showed how to reconstruct a periodic graph up to translations from LQG and a lattice basis, which we also prove in Lemma 9 in our notations for completeness. Section 3 in (Eon, 2016*a*) described surgeries on building units of LQGs. Theorem 6.1 in (Eon, 2016*b*) characterised 3-connected minimal periodic graphs. (McColm, 2024) initiated a search for systematic periodic graphs realisable by real crystal nets, see also (Edelsbrunner & Heiss, 2024).

The quotient graph G/Λ in Fig. 3 has two vertices p, q. If we orient the three edges of G/Λ from p to q, the translational vector $(0, 0)$ of the blue edge in G/Λ means that the corresponding straight-line blue edge in the lifted graph G connects points of S within the same unit cell U with a basis $\boldsymbol{v_1}, \boldsymbol{v_2}$. The orange edge with the translational vector $(1, 1)$ means that each of its infinitely many liftings in G joins a point in a cell U to another point in the cell $U + \boldsymbol{v_1} + \boldsymbol{v_2}$.

Lemma 9 (lifting)**.** *Let G be a periodic graph on a periodic point set S with a motif M in a unit cell U defined by a basis $\boldsymbol{v_1}, \dots, \boldsymbol{v_n}$ in $\mathbb{R}^n$. Let Q be a labelled quotient graph of G. Then G can be reconstructed from only Q, the basis $\boldsymbol{v_1}, \dots, \boldsymbol{v_n}$, and a* *bijection between all vertices of Q and all points of the motif $M \subset U$.*

Proof. The basis $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}$ is needed to define a unit cell U with the given points of M, which are in 1-1 correspondence with all vertices of Q. The full periodic point set S, which is the vertex set of the periodic graph G, is obtained from M by translations along the vectors $\sum_{i=1}^{n} c_i \boldsymbol{v_i}$ for all $c_i \in \mathbb{Z}$. By Definitions 7 and 8, every edge e of the labelled quotient graph Q has a translational vector $\boldsymbol{v}(e) = (c_1, \ldots, c_n)$ and is a Λ-equivalence class $[p, q] + \Lambda$ for some $p, q \in S$ whose unit cells $U(p), U(q)$ are related by the translation along $\sum_{i=1}^{n} c_i \boldsymbol{v_i}$. Then we can lift the edge e to the periodically translated straight-line edges $[p + v, q + v + \sum_{i=1}^{n} c_i \boldsymbol{v_i}]$ in the periodic graph G for all $v \in \Lambda$. □

Definition 10 (path/cycle sum)**.** *For a path (sequence of consecutive edges) in a labelled quotient graph Q, we make all directions of edges consistent in the sequence and define the* path sum *in $\mathbb{Z}^n$ as the sum of the resulting translational vectors along the path. If the path is a closed cycle, then the path sum is called the* cycle sum.

In the labelled quotient graph in Fig. 3, the upper cycle consisting of the directed orange edge (from p to q) and the inverted green edge (from q to p) has the cycle sum $(1, 1) + (0, -1) = (1, 0)$. This cycle sum means that a lifting of the cycle to the periodic graph G in $\mathbb{R}^2$ produces a polygonal path connecting a point to its translate by the vector $\boldsymbol{v_1} = (1, 0)$ in the next cell to the right.

Definition 11 (minimal tree $\mathrm{MST}(S/\Lambda)$)**.** *For a periodic point set $S \subset \mathbb{R}^n$ with a lattice Λ, the* minimal tree *is a Minimum Spanning Tree* $\mathrm{MST}(S/\Lambda)$ *on the set S/Λ of Λ-equivalence classes of points, where the distance between any classes in S/Λ is the minimum Euclidean distance between their representatives in the periodic set S.*

In Fig. 3, $\mathrm{MST}(S/\Lambda)$ consists of the shortest green edge in G/Λ.

3. Algorithm for the bridge length of a periodic point set

This section will describe main Algorithm 15 for solving Problem 3, which will call auxiliary Algorithm 12 several times. Algorithm 12 starts from a conventional representation of a periodic set $S \subset \mathbb{R}^n$ with a motif M of points given by coordinates in a basis $\boldsymbol{v_1}, \dots, \boldsymbol{v_n}$ of a lattice Λ as in a Crystallographic Information File (CIF).

At every call, Algorithm 12 returns the next shortest edge e between points of S in increasing order of length. This edge e will be represented by an ordered pair of points $p, q \in M$ and a translational vector $(c_1, \dots, c_n) \in \mathbb{Z}^n$ so that the actual straight-line edge in the lifted periodic graph $G \subset \mathbb{R}^n$ is from p to $q + \sum\limits_{i=1}^{n} c_i \boldsymbol{v_i}$. For convenience, we also include the Euclidean distance $d = |q - p + \sum\limits_{i=1}^{n} c_i \boldsymbol{v_i}|$ between these endpoints. Then Algorithm 12 outputs any edge e as a tuple $(p, q; c_1, \dots, c_n; d)$.

Algorithm 12 maintains the list of already found edges in increasing order of length. If the next required edge e is already in the list, Algorithm 12 simply returns e. This shortcut is implemented in Python with the keyword 'Yield', see the documentation at https://docs.python.org/3/glossary.html#term-generator-iterator.

Rather than starting from line 1 every time Algorithm 12 runs, each 'Yield e' returns e, then temporarily suspends processing, remembering the location execution state including all local variables. When the generator is called again, Algorithm 12 picks up where it left off in contrast to functions that start fresh on every invocation.

If the next edge e is not yet found, Algorithm 12 adds more points from a shell of unit cells surrounding the previously considered cells. This *shell* contains the extended motif M_k without the smaller motif M_{k-1} for $k > 1$, see Fig. 2. For any new point p, it suffices to consider only edges to points $q \in M \subset U$ because any edge e can be periodically translated by $\boldsymbol{v} \in \Lambda$ so that one of the endpoints of e belongs to U.

Algorithm 12. *Input: basis $\boldsymbol{v_1}, \dots, \boldsymbol{v_n}$ defining a unit cell U, a motif $M \subset U$.*

next_edge ***runs only until the next*** **Yield*, and outputs the yielded edge*.**

```
supercell_size=0, current_batch=[], next_batch=[], next_batch_min_len=infinity
while True do
    for transl_vector in [−supercell_size, supercell_size]^n do
        if max([abs(i) for i in transl_vector])!=supercell_size then Continue
        end if
        for source in the motif M do
            for dest in the motif M do
                true_dest = dest + basis · transl_vector
                length = distance(source, dest)
                next_batch.add(length,M.index(source),M.index(dest),transl_vector)
                if length < next_batch_min_len then next_batch_min_len=length
                end if
            end for
        end for
    end for
    while current_batch do
        next = min(current_batch)
        if next ≥ next_batch_min_len then Break
        end if
        current_batch.pop(next)
        Yield(next)
    end while
    current_batch.extend(next_batch), next_batch=[], supercell_size++
end while
```

There is a faster way of checking a condition equivalent to *next_batch_min_len* by using the cell geometry. Then in the vast majority of cases the algorithm can stop at a supercell one size smaller, which dramatically speeds up the calculation. This calculation is described in Remark 13. However, due to the possibility of that not being the case (upon which the algorithm would just default to the same supercell size), we will keep this simpler idea and use it for the time complexity calculations.

Remark 13 (a faster way to compute *next_batch_min_len* in Algorithm 12)**.** *For a unit cell with a basis* $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}$*, let* $\boldsymbol{a}_i$ *and* $\boldsymbol{b}_i$ *be the shortest vectors parallel and antiparallel to* $\boldsymbol{v_i}$ *from any point of a motif* $M \subset U$ *to the opposite boundary faces of the unit cell* U*. Then the faster alternative for next_batch_min_len is*

$$\min_{i=1,\ldots,n}(|\boldsymbol{a_i}| + |\boldsymbol{b_i}| + supercell\_size * |\boldsymbol{v_i}|).$$

As all the vector lengths $|\boldsymbol{a_i}|, |\boldsymbol{b_i}|$*,* $i = 1, \ldots, n$ *can be pre-computed, we get a massive improvement over the calculation of next_batch_min_len in Algorithm 12.*

Algorithm 15 will be building a labelled quotient graph Q by adding (or ignoring) edges found by Algorithm 12 and monitoring the connectivity of the growing lifted graph G whose quotient G/Λ is Q. For a basis $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}$ of a unit cell U of the lattice Λ of S, the edge e between points p and $q + \sum_{i=1}^{n} c_i\boldsymbol{v_i} \in S$ is added to Q as the edge between the Λ-equivalence classes of p and q, with the translational vector $\boldsymbol{v}(e) = (c_1, \ldots, c_n) \in \mathbb{Z}^n$. As soon as G becomes connected, the length of the last added edge is the bridge length $\beta(S)$, which will be proved in Theorem 23 later.

In comparison with a Minimum Spanning Tree built on a finite set of points, verifying the connectivity of the lifted periodic graph requires a much more complicated check that a set of translational vectors with integer coordinates is a basis in $\mathbb{Z}^n$ (not $\mathbb{R}^n$), which can have more than n vectors. For example, the vectors $(1, 0)$, $(0, 2)$, $(0, 3)$ form a basis of $\mathbb{Z}^2$ because $(0, 3) - (0, 2) = (0, 1)$ but the vectors $(0, 2)$ and $(0, 3)$ are not proportional by an integer, so no subset of these 3 vectors is a basis of $\mathbb{Z}^2$.

Algorithm 15 will use the Smith Normal Form (SNF) of a matrix of vectors $(c_1, \dots, c_n)$ in $\mathbb{Z}^n$, see p. 26 in (Newman, 1972), (Cohn, 1985), and chapter 3.6 in (Van der Waerden, 2003) for finitely generated modules over a Principal Ideal Domain (PID).

Definition 14 (Smith Normal Form and invariant factors)**.** *For integers $m \geq n$, let A be a non-zero $n \times m$ matrix over a Principal Ideal Domain P, for example, $P = \mathbb{Z}$. Then there exist invertible $n \times n$ and $m \times m$-matrices L,R, respectively, with coefficients in P, such that the product LAR is an $n \times m$ matrix whose only non-zero entries are diagonal elements a_i such that a_i divides a_{i+1} for $i = 1, \dots, j-1$, and $a_i = 0$ for $i = j, \dots, n$ for some $1 < j \leq n$. This diagonal matrix LAR is the* Smith Normal Form (SNF) *of A, and the diagonal elements a_i are the* invariant factors *of A.*

Let 1 denote the unit element of a Principal Ideal Domain P. If $P = \mathbb{Z}$, then 1 is the usual integer 1. The simplest SNF has all invariant factors equal to 1, which happens if and only if the last factor $a_n = 1$ because all previous a_i should divide a_n.

Algorithm 15 (Finding the bridge length $\beta(S)$ of any periodic point set $S \subset \mathbb{R}^n$)**.**
Initialisation. *A labelled quotient graph Q and a forest $F \subset Q$ initially consist of m isolated vertices, each representing a Λ-equivalence class of a point of the motif of S. We will build a* translational matrix *A with columns in $\mathbb{Z}^n$, which is initially empty.*

Loop stage. *Consider the next edge $e = next\_edge()$ found by Algorithm 12.*

Case 1. *If adding the edge e to the current forest F would not form a closed cycle (ignoring all edge directions), then add e to F and Q as an edge with an arbitrarily chosen direction and corresponding translational vector $\boldsymbol{v}(e)$ found by Algorithm 12.*

Case 2. *If adding the edge e to F does form a cycle, find its cycle sum $c \in \mathbb{Z}^n$ from Definition 10. If c is not $0 \in \mathbb{Z}^n$ and cannot be expressed as an integer linear combination of the columns from the current translational matrix A, then add e to Q (but not to the forest F) and add the vector c as a new column to the matrix A.*

Termination. *We stop if both conditions below hold, otherwise continue the loop.*

(1) the labelled quotient graph Q (hence the forest F) becomes connected; and

(2) the matrix A (whose columns are cycle sums of cycles created by adding edges) has n invariant factors equal to 1, see Definition 14.

The necessity of termination condition 1 in Algorithm 15 means that if the lifted periodic graph G is connected then so is its quotient $Q = G/\Lambda$. The inverse implication (sufficiency) may not hold. For example, in Fig. 3, the minimal tree $\mathrm{MST}(S/\Lambda)$ is a single green edge e_g, whose preimage under the quotient map $G \to G/\Lambda$ is the disconnected set of all green straight-line edges in the periodic graph $G \subset \mathbb{R}^2$.

Example 16 (running Algorithm 15 on the periodic point set S in Fig. 3)**.** *The first addition to the quotient graph Q and forest F, which initially had two isolated vertices p, q, is the shortest green edge e_g from p to q (case 1 in the loop stage) with the translational vector $c(e_g) = (0, 1) \in \mathbb{Z}^2$. The matrix A remains empty.*

Adding the next (by length) blue edge e_b with $c(e_b) = (0, 0)$ to $F = \{e_g\}$ creates a cycle with the cycle sum $c = c(e_g) - c(e_b) = (0, 1)$. According to case 2 in the loop stage, the quotient graph Q becomes the cycle of two edges $e_g \cup e_b$ but the forest remains $F = \{e_g\}$. The matrix A becomes one column $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$, which has only one invariant factor equal to 1. The 2nd termination condition is not yet satisfied and the current lifted graph consisting of all green and blue segments is still disconnected.

Adding the orange edge e_o with $c(e_o) = (1, 1)$ to F creates another cycle with the cycle sum $c' = c(e_g) - c(e_o) = (-1, 0)$. The quotient graph $Q = e_g \cup e_b \cup e_o$ is now full but $F = \{e_g\}$ is still one edge. The matrix A becomes $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ whose $\mathrm{SNF} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ shows that A has 2 invariant factors equal to 1. Both termination conditions hold and the lifted graph G of all green, blue, and orange edges is connected. The bridge length $\beta(S)$ equals the length of the last (orange) edge as expected.

4. Correctness and time complexity of the bridge length algorithm

This section proves the correctness of Algorithm 15 in Theorem 23 finding the bridge length and main Theorem 5 about its time complexity. Lemmas 17-18 will prove the necessity of termination condition 2 in Algorithm 15. Both conditions 1 and 2 will guarantee the connectedness of the lifted periodic graph G due to Lemma 20.

Lemma 17 (basis in $\mathbb{Z}^n \Leftrightarrow n$ invariant factors equal 1)**.** *The columns of any $n \times m$ matrix A form a basis of $\mathbb{Z}^n$ if and only if A has n invariant factors equal to 1.*

Proof. Let I_n be the identity $n \times n$ matrix whose columns $\boldsymbol{w_1}, \dots, \boldsymbol{w_n}$ form the standard orthonormal basis of $\mathbb{Z}^n$. If the columns $\boldsymbol{u_1}, \dots, \boldsymbol{u_m}$ of A form a basis of $\mathbb{Z}^n$, then $\boldsymbol{w_i} = \sum\limits_{j=1}^{m} \boldsymbol{u_j} r_{ij}$, $i = 1 \dots, n$, for some $r_{ij} \in \mathbb{Z}$. These $n \times m$ values r_{ij} complemented by the $m - n$ zero columns form the $m \times m$ matrix R such that by Definition 14 the product $I_n AR$ is the Smith Normal Form of A with all invariant factors equal to 1. Conversely, if the Smith Normal Form LAR from Definition 14 has all invariant factors equal to 1, the m columns of the $n \times m$ matrix AR and hence the m columns of the matrix A form a basis of $\mathbb{Z}^n$. Indeed, transforming these m columns by the invertible $n \times n$ matrix L gives the standard orthonormal basis $\boldsymbol{w_1}, \dots, \boldsymbol{w_n}$ of $\mathbb{Z}^n$. □

Lemma 18 (connected $G \Rightarrow n$ invariant factors equal 1)**.** *In Algorithm 15, if the lifted periodic graph $G \subset \mathbb{R}^n$ becomes connected, then the translational matrix A has n invariant factors equal to 1.*

Proof. By Lemma 17 it suffices to show that any vector $\boldsymbol{v} \in \mathbb{Z}^n$ is an integer linear combination of columns of A. Choose any point $p \in S$. Then the points p and $p + \boldsymbol{v}$ are connected in the lifted periodic graph $G \subset \mathbb{R}^n$ by a polygonal path of straight-line edges. Under $G \to G/\Lambda$, this path projects to a closed cycle C at the vertex (Λ-equivalence class) $p + \Lambda$ in the labelled quotient graph $Q = G/\Lambda$.

Let the cycle C pass through edges $e_1, \dots, e_k$ (with integer multiplicities) in the complement $Q - F$ of the forest F in the quotient graph Q. These edges were added only to Q in case 2 of the loop stage. When we tried to add every edge e_j to F, the edge e_j created a cycle C_j whose cycle sum appeared as a column in the translational matrix A (if this cycle sum was not yet an integer combination of the previous columns). Then the vector $\boldsymbol{v}$ equals the sum of the cycle sums of all the cycles C_j for $j = 1, \dots, k$, which is an integer combination of the columns of A as required. $\square$

Lemma 19 (connected $G/\Lambda \Rightarrow \exists$ a tree of representatives $T \subset G$)**.** *If a labelled quotient graph $Q = G/\Lambda$ is connected, its lifted graph $G \subset \mathbb{R}^n$ on a periodic point set S with a motif of m points and a lattice Λ includes a straight-line* tree of representatives *$T \subset G$ whose m vertices are not Λ-equivalent to each other.*

Proof. Since Q is connected, we can choose a spanning tree $F \subset Q$ on the m vertices of Q. The required tree $T \subset G$ will be a connected union of straight-line edges of G that map 1-1 to all edges of F under the quotient $G \to Q$. Start from any point $p \in S$ and take any edge e at the vertex (Λ-equivalence class) $p + \Lambda$ of $F \subset Q$. The preimage of e under $G \to Q$ contains a unique straight-line edge $[p, q] \subset G$, which we add to T. After adding to T all edges at p that project to all edges of F at $p + \Lambda$, choose another point $p' \in T$ such that the vertex $p' + \Lambda$ has an edge of F not yet covered by T under $G \to Q$. We continue adding edges to T by using their projections in $F \subset Q$ until we get a full tree $T \subset G$ that spans all m non-Λ-equivalent points of S. $\square$

Lemma 20 (termination conditions in Algorithm 15 $\Rightarrow$ connected G)**.** *Let Q be a labelled quotient graph with a translational matrix A and a lifted graph G on a periodic point set $S \subset \mathbb{R}^n$ with a lattice Λ. If Q is connected and the matrix A has n invariant factors equal to 1, then the lifted periodic graph G is connected.*

Proof. For any points $p, q \in S$, we find a path of straight-line edges in G as follows.

By Lemma 19, the connectedness of the quotient graph $Q = G/\Lambda$ guarantees a tree $T \subset G$ whose vertices represent all Λ-equivalences classes of points of S. Let p', q' be the vertices of T that are Λ-equivalent to p, q, respectively.

Since p', q' are connected by a path in T, it suffices to find a path from p to its Λ-translate $p' = p + \boldsymbol{v}$ (then similarly from q to q') in the graph G for any $\boldsymbol{v} \in \Lambda$. By Lemma 17 the columns of A form a basis of $\mathbb{Z}^n$, so $\boldsymbol{v}$ is an integer combination of these columns. It remains to find a path in G by assuming that $\boldsymbol{v}$ is one column of A. This column can appear in A only in case 2 of the loop stage in Algorithm 15 as a cycle sum of a cycle $C \subset Q$ that was created by trying to add an edge e from Algorithm 12 to a forest $F \subset Q$. If we order all edges of C from the vertex $p + \Lambda$ as $e_1, \dots, e_k$, the sum of their translation vectors equals $\boldsymbol{v}$. We build a path from p to $p + \boldsymbol{v}$ in G by finding a unique edge $[p, p_1] \subset G$ that projects to e_1, then a unique edge $[p_1, p_2] \subset G$ that projects to e_2 and so on until we cover all $e_1, \dots, e_k$ and arrive at $p + \boldsymbol{v}$. □

Remark 21. *The paper (Onus & Robins, 2022) discusses connected components of a periodic graph K in terms of homology, namely Theorem 1(1) proves that $H_0(K)$ has a basis of $\sum_{i=1}^{N}[\mathbb{Z}^d : W_{Q_i}]$ elements, see details in their section 3.1, but without describing an algorithm for finding such a basis. Our results complement their approach by proving the time complexity for checking the connectivity of a dynamic periodic graph in Theorem 5 whilst keeping track of its connected components.*

Lemma 22 (ignored edges)**.** *Let an edge e be a Λ-equivalence class of a straight-line edge $[p, q] + \Lambda$ in a lifted periodic graph G for some points $p, q \in S$. If Algorithm 15 does not add the edge e to a labelled quotient graph Q, then the points p, q are already connected by a path in the graph G lifted from Q by Lemma 9.*

Proof. The loop stage in Algorithm 15 ignores an edge e in the cases below.

Case 1. The edge e forms a cycle in Q whose cycle sum is the zero vector in $\mathbb{Z}^n$.

Case 2. The edge e forms a cycle whose cycle sum equals an integer linear combination of pre-existing *cycle sums* from the translational set B.

In both cases, we have either one cycle (in case 1) containing e, whose cycle sum is $0 \in \mathbb{Z}^n$, or several cycles (in case 2), one (up to multiplicity) of which contains e, whose total sum of translational vectors is $0 \in \mathbb{Z}^n$. By Definition 8 each edge of Q involved in this zero total sum can be lifted to a straight-line edge in the periodic graph $G \subset \mathbb{R}^n$. If we start from the given point $p \in S$, a cycle in Q and its sum 0 of translational vectors guarantees that the sequence of the lifted edges in G finishes at the same point p and hence forms a cycle C. This cycle C has the edge $[p, q]$ whose exclusion keeps the points $p, q \in S$ connected by the complementary path $C - [p, q]$. □

Theorem 23. *Algorithm 15 finds the bridge length $\beta(S)$ from Definition 2 for any periodic point set $S \subset \mathbb{R}^n$ with a motif M of points given in a basis $\boldsymbol{v_1}, \dots, \boldsymbol{v_n}$.*

Proof. Within Algorithm 15, let d be the length of the last added edge e after which both termination conditions finally hold. By Lemma 22, all ignored edges do not create extra connections in the graph G. By Lemmas 18 and 19, the graph G obtained before adding the last edge e is disconnected. Lemma 20 guarantees that, after e is added, the graph G becomes connected. Because Algortihm 12 yields edges in increasing order, e is the shortest edge that could have this property, so the bridge length is $\beta(S) = d$. □

***Proof of Theorem** 5.* Algorithm 15 has the initialisation of a constant time $O(1)$ and the loop stage. We will multiply an upper bound for the number of loops by the time complexity of each loop. One loop in Algorithm 15 consists of the checks below.

- *(Check1)* Does adding an edge e to a forest F create a cycle?
- *(Check2)* Is the cycle sum an integer combination of previous cycle sums?
- *(CheckT)* After appending a cycle sum c to the matrix A and calculating the Smith Normal Form of A, does A have n invariant factors equal to 1?

Check1 and *Check2* can be done in $O(m^2)$ time, by using an antisymmetric additive matrix R, where R_{ij} is the sum for the path from p_i to p_j in the LQG Q.

In the naive case, *CheckT* has complexity $O(N + m^2) = O(N)$, where $O(N)$ is the complexity of the Smith Normal Form. Since this term is dominant, we will use $O(N)$ to represent the complexity of a single loop iteration of Algorithm 15. The complexity of updating the matrix R is dominated by steps that have $O(N)$ complexity.

A different way of performing *CheckT* is to append columns to A in an 'online' fashion. This avoids the need to calculate the Smith Normal Form from scratch every time (or often at all), and reduces the complexity to a time close to $O(Mul \cdot E \cdot n)$, where $O(Mul) = O(m^{\log_2 7})$ is the complexity of multiplication of $m \times m$ matrices, see (Karstadt & Schwartz, 2020), and $O(E)$ is the complexity of the Extended Euclidean Algorithm. As this reduction in complexity is dominated by the price of populating the edges with Algorithm 12, this will be irrelevant for most use cases (and is not used in the experiments shown later). If a use case involves, say, a large or high-dimensional pre-populated set of edges, then more information can be found in Appendix A.

Every loop iteration calls Algorithm 12. If we consider all calls to Algorithm 12 as running sequentially, then the main loop will run at most $a(U) + 1$ times, where $a(U)$ is the aspect ratio from Definition 4. Each loop runs through the unit cells that are '*supercell_size*' away from the central cell U_1. By the end, we will have run through and *yielded* $(a(U) + 1)^n$ unit cells. For each unit cell U_i, we find all distances between the m points in U_i and m points in the central cell. The required time is $O(m^2)$ for two cells and $O(m^2 a(U)^n)$ for all cells.

Algorithm 15 does not actually run for every edge found by Algorithm 12 but we assume this for simplicity. All other operations in Algorithm 12 are dominated by the complexity $O(N)$ for each iteration of Algorithm 15, for example checking if Q is connected, which requires $O(m)$ time by using the matrix R. The worst-case complexity for the naive implementation of Algorithm 15 is $O(m^2 a(U)^n N)$. □

5. Experiments on real and simulated crystals, and a discussion

This section shows that exact Algorithm 15 for the bridge length substantially improves the upper bound $\beta(S) \leq r(U)$ from Lemma 3.7 in (Anosova & Kurlin, 2022).

Table 1 reports the bridge lengths computed by Algorithm 15 on the real crystals from Fig. 4. The names of T2 polymorphs refer to the crystalline forms $\alpha, \beta, \gamma, \delta, \epsilon$ based on the same molecule T2. The IDs starting from 6-letter codes in the first column of Table 1 refer to the Cambridge Structural Database (Taylor & Wood, 2019).

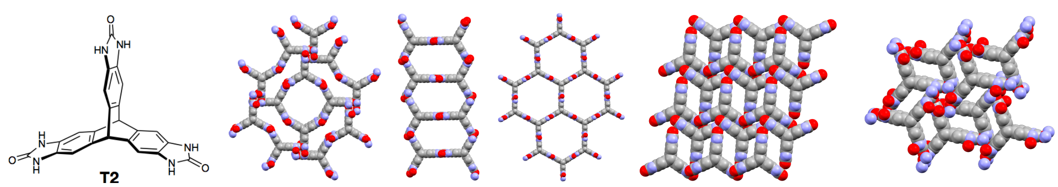

Fig. 4. T2 molecule and 5 crystals synthesized from T2. The first four T2-α, T2-β, T2-γ, T2-δ were reported in (Pulido *et al.*, 2017), the last T2-ϵ in (Zhu *et al.*, 2022).

Note that the polymorph T2-γ contains four slightly different versions in the CSD (DEBXIT01…04) because their crystal structures were determined at different temperatures. The seven versions DEBXIT01…07 with the same 6-letter code may look similar even for experts. Actually, T2-δ (SEMDIA) was deposited later than others because even the original authors confused this polymorph with earlier crystals, which was detected by the structural invariants in (Edelsbrunner *et al.*, 2021).

The bridge length is also an isometry invariant of periodic point sets and distinguishes all crystals in Table 1. Table 1 includes upper bounds of $r(U)$ and $R(S)$ from Definition 4, which were proved in Lemma 3.7 from (Anosova & Kurlin, 2022). The run times in Table 1 are recorded on a laptop with Intel i5, one 1GHz core, 8Gb RAM.

Table 1. *The bridge length $\beta(S)$ of experimental T2 crystals from (Pulido* et al.*, 2017).*

CSD ref codes of experimental T2 crystals	number of atoms in a cell	bridge length $\beta(S)$, Å	upper bound $r(U)$, Å	upper bound $R(S)$, Å	best upper bound over exact value $R(S)/\beta(S)$	running time, seconds
T2-α NAVXUG	184	2.028	22.325	15.609	7.695	4.337
T2-δ SEMDIA	92	2.713	14.401	8.350	3.077	0.671
T2-γ DEBXIT01	92	1.879	23.224	23.224	12.358	0.706
T2-γ DEBXIT02	92	1.926	23.226	23.226	12.061	0.636
T2-γ DEBXIT03	92	1.902	23.230	23.230	12.216	0.653
T2-γ DEBXIT04	92	1.970	23.290	23.290	11.824	0.649
T2-β DEBXIT05	92	3.163	20.665	12.906	4.080	0.664
T2-β DEBXIT06	92	3.188	20.694	12.884	4.042	0.657
T2-ϵ DEBXIT07	92	2.062	12.608	5.707	2.768	0.641
T2 average	295.8	2.293	15.203	9.110	3.973	31.653

The final row contains the averages for 5,679 simulated T2 crystals, which which are publicly available in the supplementary materials of (Pulido *et al.*, 2017) and were used for predicting the 5 experimental polymorphs represented by 9 entries in the CSD. For all crystals in Table 1, the translational basis size never exceeded 3.

We thank all reviewers for their valuable time and helpful suggestions.

Appendix A
A faster 'online' algorithm for the Smith Normal Form

Recalling from Definition 14 that the diagonal of LAR is made up of the invariant factors of A. To progressively calculate the Smith Normal Form, we must only keep track of the right-multiplying unimodular matrix R, and the invariant factors themselves (a vector $F \in \mathbb{Z}^n$). To run the main algorithm here, we do have to begin with a matrix with n integer linearly independent rows. 'Adding' a vector $\boldsymbol{v}$ to F is where the process changes. We treat R and F as mutable, meaning each value is not necessarily fixed to its original assignment. The first step is to define $x := \boldsymbol{v} * R$, then we find $g_i = gcd(x_i, F_i)$. If $F_i = g_i$ (i.e., F_i *divides* x_i), we can continue with $i := i + 1$, with no need to change R as it only keeps track of columns (for context, if we were keeping track of L, too, we would have to subtract the i^{th} row from the last row x_i/F_i times).

If F_i divides x_i for all i, we would know that including the vector changes nothing, therefore the relative edge is also irrelevant and can be discarded (this reduces the complexity of most *CheckT* from $O(N)$ to $O(Mul\ + log^2(n))$. However, if $g_i < F_i$, then F_i not only becomes g_i, we also know that the SNF will change, and that we must add the edge relative to $\boldsymbol{v}$. We must also alter R, accounting for the fact that F represents the diagonal of a matrix. We can do this by any typical process of 'changing the pivot' in your standard SNF algorithm, ensuring that we update R in tandem.

As accounting for the previous values of i is trivial, this is worst-case equivalent to calculating the SNF of an $(n-i) \times (n-i)$ matrix $= O(N_{n-i})$, which improves upon the naive method of calculating the SNF from scratch upon every alteration of A.

Lemma 24. *Updating the Smith Normal Form as above preserves its properties.*

Proof. As we only alter with elementary row and column operations, this preserves the Smith Normal Form. By multiplying the to-be-added row $\boldsymbol{v}$ by R before concatenating it as a new row to F, it is the same as performing those same elementary column operations upon a new matrix: $[\boldsymbol{A}_0, .., \boldsymbol{A}_n, \boldsymbol{v}]$ (i.e. $\boldsymbol{v}$ concatenated as a row onto A). We then continue to perform only elementary row and column operations, and we end with a matrix that satisfies the conditions of and SNF noted in Definition 14. □

To discuss this process any further is beyond the scope of this paper, though there are still some small tricks that take advantage of the way the 'new' rows for consideration are intrinsically related to v, and how F_{i+1} divides F_i.

References

Anosova, O. & Kurlin, V. (2021). In *Proceedings of Discrete Geometry and Mathematical Morphology*, pp. 229–241.

Anosova, O. & Kurlin, V. (2022). *arXiv:2205.15298.*
URL: *https://kurlin.org/projects/periodic-geometry/near-duplicate-periodic-patterns.pdf*

Anosova, O., Kurlin, V. & Senechal, M. (2024). *IUCrJ*, **11**, 453–463.

Brock, C. P., (2021). Change to the definition of "crystal" in the IUCr Online Dictionary of Crystallography. https://www.iucr.org/news/newsletter/etc/articles?issue=151351&result_138339_result_page=17.

Chung, S. J., Hahn, T. & Klee, W. (1984). *Acta Crystallographica Section A: Foundations of Crystallography*, **40**(1), 42–50.

Cohen, E. & Megiddo, N. (1990). In *Applied geometry and discrete mathematics*, pp. 135–146. Citeseer.

Cohn, P. M. (1985). *Free rings and their relations*. Academic Press.

Edelsbrunner, H. & Heiss, T. (2024). *arXiv:2408.16575*.

Edelsbrunner, H., Heiss, T., Kurlin, V., Smith, P. & Wintraecken, M. (2021). In *Proceedings of SoCG*, pp. 32:1–32:16.

Eon, J.-G. (2011). *Acta Crystallographica Section A*, **67**(1), 68–86.

Eon, J.-G. (2016*a*). *Acta Cryst A*, **72**(3), 268–293.

Eon, J.-G. (2016*b*). *Acta Cryst A*, **72**(3), 376–384.

Karstadt, E. & Schwartz, O. (2020). *Journal of the ACM*, **67**(1), 1–31.

McColm, G. (2024). *Acta Crystallographica Section A*, **80**(1).

Newman, M. (1972). *Integral matrices*. Academic Press.

Onus, A. & Robins, V. (2022). *arXiv:2208.09223*.

Pulido, A. *et al.* (2017). *Nature*, **543**, 657–664.

Taylor, R. & Wood, P. A. (2019). *Chemical reviews*, **119**(16), 9427–9477.

Van der Waerden, B. L. (2003). *Algebra*, vol. 1. Springer Science & Business Media.

Widdowson, D. & Kurlin, V. (2022). *Advances in Neural Information Processing Systems (NeurIPS)*, **35**, 24625–24638.

Widdowson, D. & Kurlin, V. (2024). *Crystal Growth and Design*, **24**, 5627–5636.

Zhu, Q. *et al.* (2022). *J Amer. Chem. Soc.* **144**, 9893–9901.

iucr

Synopsis

We describe an efficient algorithm to compute the bridge length estimating the size of a complete isoset invariant, which classifies all periodic point sets under Euclidean isometry.
