Griddings of Permutations and Hardness of Pattern Matching

Vít Jelínek
Computer Science Institute, Charles University, Prague, Czech Republic

Michal Opler
Computer Science Institute, Charles University, Prague, Czech Republic

Jakub Pekárek
Department of Applied Mathematics, Charles University, Prague, Czech Republic

Abstract

We study the complexity of the decision problem known as *Permutation Pattern Matching*, or PPM. The input of PPM consists of a pair of permutations τ (the "text") and π (the "pattern"), and the goal is to decide whether τ contains π as a subpermutation. On general inputs, PPM is known to be NP-complete by a result of Bose, Buss and Lubiw. In this paper, we focus on restricted instances of PPM where the text is assumed to avoid a fixed (small) pattern σ; this restriction is known as Av(σ)-PPM. It has been previously shown that Av(σ)-PPM is polynomial for any σ of size at most 3, while it is NP-hard for any σ containing a monotone subsequence of length four.

In this paper, we present a new hardness reduction which allows us to show, in a uniform way, that Av(σ)-PPM is hard for every σ of size at least 6, for every σ of size 5 except the symmetry class of 41352, as well as for every σ symmetric to one of the three permutations 4321, 4312 and 4231. Moreover, assuming the exponential time hypothesis, none of these hard cases of Av(σ)-PPM can be solved in time $2^{o(n/\log n)}$. Previously, such conditional lower bound was not known even for the unconstrained PPM problem.

On the tractability side, we combine the CSP approach of Guillemot and Marx with the structural results of Huczynska and Vatter to show that for any monotone-griddable permutation class $\mathcal{C}$, PPM is polynomial when the text is restricted to a permutation from $\mathcal{C}$.

2012 ACM Subject Classification Mathematics of computing → Permutations and combinations; Theory of computation → Pattern matching; Theory of computation → Problems, reductions and completeness

Keywords and phrases Permutation, pattern matching, NP-hardness

Digital Object Identifier 10.4230/LIPIcs.MFCS.2021.65

Related Version *Full Version*: https://arxiv.org/abs/2107.10897

Funding Supported by the GAUK project 1766318.
Vít Jelínek: Supported by project 18-19158S of the Czech Science Foundation.
Michal Opler: Supported by project 21-32817S of the Czech Science Foundation.

1 Introduction

Permutation Pattern Matching, or PPM, is one of the most fundamental decision problems related to permutations. In PPM, the input consists of two permutations: τ, referred to as the "text", and π, referred to as the "pattern". The two permutations are represented as sequences of distinct integers. The goal is to determine whether the text τ contains the pattern σ, that is, whether τ has a subsequence order-isomorphic to σ (see Section 2 for precise definitions).

Bose, Buss and Lubiw [7] have shown that the PPM problem is NP-complete. Thus, most recent research into the complexity of PPM focuses either on parametrized or superpolynomial algorithms, or on restricted instances of the problem.

For a pattern π of size k and a text τ of size n, a straightforward brute-force approach can solve PPM in time $O(n^{k+1})$. This was improved by Ahal and Rabinovich [1] to $O(n^{0.47k+o(k)})$, and then by Berendsohn, Kozma and Marx [6] to $O(n^{k/4})$.

When k is large in terms of n, a brute-force approach solves PPM in time $O(2^{n+o(n)})$. The first improvement upon this bound was obtained by Bruner and Lackner [8], whose algorithm achieves the running time $O(1.79^n)$, which was in turn improved by Berendsohn, Kozma and Marx [6] to $O(1.6181^n)$.

Guillemot and Marx [11] have shown, perhaps surprisingly, that PPM is fixed-parameter tractable with parameter k, via an algorithm with running time $n \cdot 2^{O(k^2 \log k)}$, later improved to $n \cdot 2^{O(k^2)}$ by Fox [10].

Restricted instances

Given that PPM is NP-hard on general inputs, various authors have sought to identify restrictions on the input permutations that would allow for an efficient pattern matching algorithm. These restrictions usually take the form of specifying that the pattern must belong to a prescribed set $\mathcal{C}$ of permutations (the so-called $\mathcal{C}$-Pattern PPM problem), or that both the pattern and the text must belong to a set $\mathcal{C}$ (known as $\mathcal{C}$-PPM problem). The most commonly considered choices for $\mathcal{C}$ are sets of the form $\mathrm{Av}(\sigma)$ of all the permutations that do not contain a fixed pattern σ.

Note that for the class Av(21), consisting of all the increasing permutations, Av(21)-Pattern PPM corresponds to the problem of finding the longest increasing subsequence in the given text, a well-known polynomially solvable problem [17]. Another polynomially solvable case is Av(132)-Pattern PPM, which follows from more general results of Bose et al. [7].

In contrast, for the class Av(321) of permutations avoiding a decreasing subsequence of length 3 (or equivalently, the class of permutations formed by merging two increasing sequences), Av(321)-Pattern PPM is already NP-complete, as shown by Jelínek and Kynčl [15]. In fact, Jelínek and Kynčl show that $\mathrm{Av}(\sigma)$-Pattern PPM is polynomial for $\sigma \in \{1, 12, 21, 132, 231, 312, 213\}$ and NP-complete otherwise.

For the more restricted $\mathrm{Av}(\sigma)$-PPM problem, a polynomial algorithm for $\sigma = 321$ was found by Guillemot and Vialette [12] (see also Albert et al. [2]), and it follows that $\mathrm{Av}(\sigma)$-PPM is polynomial for any σ of length at most 3. In contrast, the case $\sigma = 4321$ (and by symmetry also $\sigma = 1234$) is NP-complete [15]. It follows that $\mathrm{Av}(\sigma)$-PPM is NP-complete whenever σ contains 1234 or 4321 as subpermutation, and in particular, it is NP-complete for any σ of length 10 or more.

In this paper, our main motivation is to close the gap between the polynomial and the NP-complete cases of $\mathrm{Av}(\sigma)$-PPM. We develop a general type of hardness reduction, applicable to any permutation class that contains a suitable grid-like substructure. We then verify that for most choices of σ large enough, the class $\mathrm{Av}(\sigma)$ contains the required substructure. Specifically, we can prove that $\mathrm{Av}(\sigma)$-PPM is NP-complete in the following cases:

- Any σ of size at least 6.
- Any σ of size 5, except the symmetry type of 41352 (i.e., the two symmetric permutations 41352 and 25314).
- Any σ symmetric to one of 4321, 4312 or 4231.

Note that the list above includes the previously known case $\sigma = 4321$. Our hardness reduction, apart from being more general than previous results, has also the advantage of being more efficient: we reduce an instance of 3-SAT of size m to an instance of PPM of size $O(m \log m)$. This implies, assuming the exponential time hypothesis (ETH), that none of these NP-complete cases of Av(σ)-PPM can be solved in time $2^{o(n/\log n)}$. Previously, this lower bound was not known to hold even for the unconstrained PPM problem.

Grid classes

The sets of permutations of the form Av(σ), i.e., the sets determined by a single forbidden pattern, are the most common type of permutation sets considered; however, such sets are not necessarily the most convenient tools to understand the precise boundary between polynomial and NP-complete cases of PPM. We will instead work with the more general concept of *permutation class*, which is a set $\mathcal{C}$ of permutations with the property that for any $\pi \in \mathcal{C}$, all the subpermutations of π are in $\mathcal{C}$ as well.

A particularly useful family of permutation classes are the so-called grid classes. When dealing with grid classes, it is useful to represent a permutation $\pi = \pi_1\pi_2\cdots\pi_n$ by its *diagram*, which is the set of points $\{(i, \pi_i) \mid i = 1, \dots, n\}$. A grid class is defined in terms of a *gridding matrix* $\mathcal{M}$, whose entries are (possibly empty) permutation classes. We say that a permutation π has an $\mathcal{M}$*-gridding*, if its diagram can be partitioned, by horizontal and vertical cuts, into an array of rectangles, where each rectangle induces in π a subpermutation from the permutation class in the corresponding cell of $\mathcal{M}$. The permutation class Grid($\mathcal{M}$) then consists of all the permutations that have an $\mathcal{M}$-gridding.

To a gridding matrix $\mathcal{M}$ we associate a *cell graph*, which is the graph whose vertices are the entries in $\mathcal{M}$ that correspond to infinite classes, with two vertices being adjacent if they belong to the same row or column of $\mathcal{M}$ and there is no other infinite entry of $\mathcal{M}$ between them.

In the griddings we consider in this paper, a prominent role is played by four specific classes, forming two symmetry pairs: one pair are the monotone classes Av(21) and Av(12), containing all the increasing and all the decreasing permutations, respectively. Note that any infinite class of permutations contains at least one of Av(12) and Av(12) as a subclass, by the Erdős–Szekeres theorem [9].

The other relevant pair of classes involves the so-called *Fibonacci class*, denoted $\oplus 21$, and its mirror image $\ominus 12$. The Fibonacci class can be defined as the class of permutations avoiding the three patterns 321, 312 and 231, or equivalently, it is the class of permutations $\pi = \pi_1\pi_2\cdots\pi_n$ satisfying $|\pi_i - i| \leq 1$ for every i.

Griddings have been previously used, sometimes implicitly, in the analysis of special cases of PPM, where they were applied both in the design of polynomial algorithms [2, 12], and in NP-hardness proofs [15, 16]. In fact, all the known NP-hardness arguments for special cases of $\mathcal{C}$-Pattern PPM are based on the existence of suitable grid subclasses of the class $\mathcal{C}$. In particular, previous results of the authors [16] imply that for any gridding matrix $\mathcal{M}$ that only involves monotone or Fibonacci cells, Grid($\mathcal{M}$)-Pattern PPM is polynomial when the cell graph of $\mathcal{M}$ is a forest, and it is NP-complete otherwise. Of course, if Grid($\mathcal{M}$)-Pattern PPM is polynomial then Grid($\mathcal{M}$)-PPM is polynomial as well. However, the results in this paper identify a broad family of examples where Grid($\mathcal{M}$)-PPM is polynomial, while Grid($\mathcal{M}$)-Pattern PPM is known to be NP-complete.

Our main hardness result, Theorem 2, can be informally rephrased as a claim that $\mathcal{C}$-PPM is hard for a class $\mathcal{C}$ whenever $\mathcal{C}$ contains, for each n and a fixed $\varepsilon > 0$, a grid subclass whose cell graph is a path of length n, and at least εn of its cells are Fibonacci classes. A somewhat less technical consequence, Corollary 4, says that Grid($\mathcal{M}$)-PPM is NP-hard whenever the cell graph of $\mathcal{M}$ is a cycle with no three vertices in the same row or column and with at least one Fibonacci cell.

Corollary 4 is, in a certain sense, best possible, since our main tractability result, Theorem 10, states that $\mathcal{C}$-PPM is polynomial whenever $\mathcal{C}$ is *monotone-griddable*, that is, $\mathcal{C} \subseteq \mathrm{Grid}(\mathcal{M})$, where $\mathcal{M}$ contains only monotone (or empty) cells. Moreover, by a result of Huczynska and Vatter [13], every class $\mathcal{C}$ that does not contain $\oplus 21$ or $\ominus 12$ is monotone griddable. Taken together, these results show that $\mathrm{Grid}(\mathcal{M})$-PPM is polynomial whenever no cell of $\mathcal{M}$ contains $\oplus 21$ or $\ominus 12$ as a subclass.

2 Preliminaries

A *permutation of length* n is a sequence $\pi_1, \ldots, \pi_n$ in which each element of the set $[n] = \{1, 2, \ldots, n\}$ appears exactly once. When writing out short permutations explicitly, we shall omit all punctuation and write, e.g., 15342 for the permutation $1, 5, 3, 4, 2$. The *permutation diagram* of π is the set of points $S_\pi = \{(i, \pi_i) \mid i \in [n]\}$ in the plane. Observe that no two points from S_π share the same x- or y-coordinate. We say that such a set is in *general position*. Note that we blur the distinction between permutations and their permutation diagrams, e.g., we shall refer to "the point of π".

For a point p in the plane, we let $p.x$ denote its horizontal coordinate and $p.y$ its vertical coordinate. Two finite sets $S, R \subseteq \mathbb{R}^2$ in general position are *order-isomorphic*, or just *isomorphic* for short, if there is a bijection $f\colon S \to R$ such that for any pair of points $p \neq q$ of R we have $f(p).x < f(q).x$ if and only if $p.x < q.x$, and $f(p).y < f(p).y$ if and only if $p.y < q.y$; in such case, the function f is the *isomorphism* from S to R. The *reduction* of a finite set $S \subseteq \mathbb{R}^2$ in general position is the unique permutation π such that S is isomorphic to S_π.

A permutation τ *contains* a permutation π, denoted by $\pi \preceq \tau$, if there is a subset $P \subseteq S_\tau$ that is isomorphic to S_π. Such a subset is then called *an occurrence* of π in τ, and the isomorphism from S to P is an *embedding* of π into τ. If τ does not contain π, we say that τ *avoids* π.

A *permutation class* is any down-set $\mathcal{C}$ of permutations, i.e., a set $\mathcal{C}$ such that if $\pi \in \mathcal{C}$ and $\sigma \preceq \pi$ then also $\sigma \in \mathcal{C}$. For a permutation σ, we let $\mathrm{Av}(\sigma)$ denote the class of all σ-avoiding permutations. We shall throughout use the symbols ⧄ and ⧅ as short-hands for the class of increasing permutations Av(21) and the class of decreasing permutations Av(12).

Observe that for every permutation π of length at most m, the permutation diagram S_π is a subset of the set $\{p \mid \frac{1}{2} < p.x < m + \frac{1}{2} \wedge \frac{1}{2} < p.y < m + \frac{1}{2}\}$, called *$m$-box*. This fact motivates us to extend the usual permutation symmetries to bijections of the whole m-box. In particular, there are eight symmetries generated by:

reversal which reflects the m-box through its vertical axis, i.e., the image of a point p is the point $(m + 1 - p.x, p.y)$,

complement which reflects the m-box through its horizontal axis, i.e., the image of a point p is the point $(p.x, m + 1 - p.y)$,

inverse which reflects the m-box through its ascending diagonal axis, i.e., the image of a point p is the point $(p.y, p.x)$.

We say that a permutation π is *symmetric* to a permutation σ if π can be transformed into σ by any of the eight symmetries generated by reversal, complement and inverse. The *symmetry type*[1] of a permutation σ is the set of all the permutations symmetric to σ.

[1] We chose the term "symmetry type" over the more customary "symmetry class", to avoid possible confusion with the notion of permutation class.

The symmetries generated by reversal, complement and inverse can be applied not only to individual permutations but also to their classes. Formally, if Ψ is one of the eight symmetries and $\mathcal{C}$ is a permutation class, then $\Psi(\mathcal{C})$ refers to the set $\{\Psi(\sigma)|\ \sigma \in \mathcal{C}\}$. We may easily see that $\Psi(\mathcal{C})$ is again a permutation class.

Consider a pair of permutations π of length n and σ of length m. The *inflation* of a point p of π by σ is the reduction of the point set

$$S_\pi \setminus \{p\} \cup \left\{ \left(p.x + \frac{q.x}{m+1}, p.y + \frac{q.y}{m+1} \right) \middle| \, q \in S_\sigma \right\}.$$

Informally, we replace the point p with a tiny copy of σ.

The *direct sum* of π and σ, denoted by $\pi \oplus \sigma$, is the result of inflating the "1" in 12 with π and then inflating the "2" with σ. Similarly, the *skew sum* of π and σ, denoted by $\pi \ominus \sigma$, is the result of inflating the "2" in 21 with π and then inflating the "1" with σ. If a permutation τ cannot be obtained as direct sum of two shorter permutations, we say that τ is *sum-indecomposable* and if it cannot be obtained as a skew sum of two shorter permutations, we say that it is *skew-indecomposable*. Moreover, we say that a permutation class $\mathcal{C}$ is *sum-closed* if for any $\pi, \sigma \in \mathcal{C}$ we have $\pi \oplus \sigma \in \mathcal{C}$. We define *skew-closed* analogously.

We define the *sum completion* of a permutation π to be the permutation class

$$\oplus\pi = \{\sigma_1 \oplus \sigma_2 \oplus \cdots \oplus \sigma_k \mid \sigma_i \preceq \pi \text{ for all } i \leq k \in \mathbb{N}\}.$$

Analogously, we define the *skew completion* $\ominus\pi$ of π. The class $\oplus 21$ is known as the *Fibonacci class*.

2.1 Grid classes

When we deal with matrices, we number their rows from bottom to top to be consistent with the Cartesian coordinates we use for permutation diagrams. For the same reason, we let the column coordinates precede the row coordinates; in particular, a $k \times \ell$ *matrix* is a matrix with k columns and ℓ rows, and for a matrix $\mathcal{M}$, we let $\mathcal{M}_{i,j}$ denote its entry in column i and row j.

A matrix $\mathcal{M}$ whose entries are permutation classes is called a *gridding matrix*. Moreover, if the entries of $\mathcal{M}$ belong to the set $\{ ⧄, ⧅, \emptyset\}$ then we say that $\mathcal{M}$ is a *monotone gridding matrix*.

A $k \times \ell$*-gridding* of a permutation π of length n are two weakly increasing sequences $1 = c_1 \leq \cdots \leq c_{k+1} = n+1$ and $1 = r_1 \leq \cdots \leq r_{\ell+1} = n+1$. For each $i \in [k]$ and $j \in [\ell]$, we call the set of points $p \in S_\pi$ such that $c_i \leq p.x < c_{i+1}$ and $r_j \leq p.y < r_{j+1}$ the (i,j)*-cell* of π. An $\mathcal{M}$*-gridding* of a permutation π is a $k \times \ell$-gridding such that the reduction of the (i,j)-cell of π belongs to the class $\mathcal{M}_{i,j}$ for every $i \in [k]$ and $j \in [\ell]$. If π has an $\mathcal{M}$-gridding, then π is said to be $\mathcal{M}$*-griddable*, and the *grid class of* $\mathcal{M}$, denoted by $\mathrm{Grid}(\mathcal{M})$, is the class of all $\mathcal{M}$-griddable permutations.

The *cell graph* of the gridding matrix $\mathcal{M}$, denoted $G_\mathcal{M}$, is the graph whose vertices are pairs (i,j) such that $\mathcal{M}_{i,j}$ is an infinite class, with two vertices being adjacent if they share a row or a column of $\mathcal{M}$ and all the entries between them are finite or empty. See Figure 1. We slightly abuse the notation and use the vertices of $G_\mathcal{M}$ for indexing $\mathcal{M}$, i.e., for a vertex v of $G_\mathcal{M}$, we write $\mathcal{M}_v$ to denote the corresponding entry.

A *proper-turning path* in $G_\mathcal{M}$ is a path P such that no three vertices of P share the same row or column. Similarly, a *proper-turning cycle* in $G_\mathcal{M}$ is a cycle C such that no three vertices of C share the same row or column.

$$\mathcal{M} = \begin{pmatrix} \oplus 21 & \text{———} & \mathrm{Av}(21) \\ & & | \\ & \ominus 12 \text{ — } & \mathrm{Av}(12) \end{pmatrix}$$

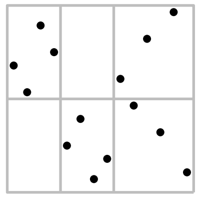

Figure 1 A gridding matrix $\mathcal{M}$ on the left and a permutation equipped with an $\mathcal{M}$-gridding on the right. Empty entries of $\mathcal{M}$ are omitted and the edges of $G_{\mathcal{M}}$ are displayed inside $\mathcal{M}$.

Let π be a permutation, and let (c, r) be its $k \times \ell$-gridding, where $c = (c_1, \ldots, c_{k+1})$ and $r = (r_1, \ldots, r_{\ell+1})$. A permutation π together with a gridding (c, r) form a *gridded permutation*. When dealing with gridded permutations, it is often convenient to apply symmetry transforms to individual columns or rows of the gridding. Specifically, the *reversal of the i-th column* of π is the operation which generates a new (c, r)-gridded permutation π' by taking the diagram of π, and then reflecting the rectangle $[c_i, c_{i+1} - 1] \times [1, n]$ in the diagram through its vertical axis, producing the diagram of the new permutation π'. Note that π' differs from π by having all the entries at positions $c_i, c_i + 1, \ldots, c_{i+1} - 1$ in reverse order. If $c_{i+1} \leq c_i + 1$, then $\pi' = \pi$.

Similarly, the *complementation of the j-th row* of the (c, r)-gridded permutation π is obtained by taking the rectangle $[1, n] \times [r_j, r_{j+1} - 1]$ and turning it upside down, obtaining a permutation diagram of a new permutation.

Column reversals and row complementations can also be applied to gridding matrices: a reversal of a column i in a gridding matrix $\mathcal{M}$ simply replaces all the classes appearing in the entries of the i-th column by their reverses; a row complementation is defined analogously.

We often need to perform several column reversals and row complementations at once. To describe such operations succinctly, we introduce the concept of $k \times \ell$-orientation. A *$k \times \ell$-orientation* is a pair of functions $\mathcal{F} = (f_c, f_r)$ with $f_c \colon [k] \to \{-1, 1\}$ and $f_r \colon [\ell] \to \{-1, 1\}$. To *apply* the orientation $\mathcal{F}$ to a $k \times \ell$-gridded permutation π means to create a new permutation $\mathcal{F}(\pi)$ by reversing in π each column i for which $f_c(i) = -1$ and complementing each row j for which $f_r(j) = -1$. Note that the order in which we perform the reversals and complementations does not affect the final outcome. Note also that $\mathcal{F}$ is an involution, that is, $\mathcal{F}(\mathcal{F}(\pi)) = \pi$ for any $k \times \ell$-gridded permutation π.

We may again also apply $\mathcal{F}$ to a gridding matrix $\mathcal{M}$. By performing, in some order, the row reversals and column complementations prescribed by $\mathcal{F}$ on the matrix $\mathcal{M}$, we obtain a new gridding matrix $\mathcal{F}(\mathcal{M})$. For instance, taking the gridding matrix $\begin{pmatrix} \boxbslash & \boxslash \\ \boxslash & \boxbslash \end{pmatrix}$ and applying reversal to its first column yields the gridding matrix $\begin{pmatrix} \boxslash & \boxslash \\ \boxbslash & \boxbslash \end{pmatrix}$. Observe that if (c, r) is an $\mathcal{M}$-gridding of a permutation π, then the same gridding (c, r) is also an $\mathcal{F}(\mathcal{M})$-gridding of the permutation $\mathcal{F}(\pi)$.

Let $\mathcal{M}$ be a monotone gridding matrix. An orientation $\mathcal{F}$ of $\mathcal{M}$ is *consistent* if all the nonempty entries of $\mathcal{F}(\mathcal{M})$ are equal to $\boxslash$. For instance, the matrix $\begin{pmatrix} \boxbslash & \boxslash \\ \boxslash & \boxbslash \end{pmatrix}$ has a consistent orientation acting by reversing the first column and complementing the first row, while the matrix $\begin{pmatrix} \boxslash & \boxslash \\ \boxslash & \boxbslash \end{pmatrix}$ has no consistent orientation. We remark that Vatter and Waton [18] have shown that any monotone gridding matrix whose cell graph is acyclic has a consistent orientation.

A vital role in our arguments is played by the concept of monotone griddability. We say that a class $\mathcal{C}$ is *monotone-griddable* if there exists a monotone gridding matrix $\mathcal{M}$ such that $\mathcal{C}$ is contained in $\mathrm{Grid}(\mathcal{M})$. Huczynska and Vatter [13] provided a neat and useful characterization of monotone-griddable classes.

▶ **Theorem 1** (Huczynska and Vatter [13]). *A permutation class $\mathcal{C}$ is monotone-griddable if and only if it contains neither the Fibonacci class $\oplus 21$ nor its symmetry $\ominus 12$.*

Finally, a monotone grid class $\mathrm{Grid}(\mathcal{C}\ \mathcal{D})$ where both $\mathcal{C}$ and $\mathcal{D}$ are non-empty is called a *horizontal monotone juxtaposition*. Analogously, a *vertical monotone juxtaposition* is a monotone grid class $\mathrm{Grid}\left(\begin{smallmatrix}\mathcal{C}\\ \mathcal{D}\end{smallmatrix}\right)$ with both $\mathcal{C}$ and $\mathcal{D}$ non-empty. A *monotone juxtaposition* is simply a class that is either a horizontal or a vertical monotone juxtaposition.

2.2 Pattern matching complexity

In this paper, we deal with the complexity of the decision problem known as $\mathcal{C}$-PPM. For a permutation class $\mathcal{C}$, the input of $\mathcal{C}$-PPM is a pair of permutations (π, τ) with both π and τ belonging to $\mathcal{C}$. An instance of $\mathcal{C}$-PPM is then accepted if τ contains π, and rejected if τ avoids π. In the context of pattern-matching, π is referred to as the *pattern*, while τ is the *text*.

Note that an algorithm for $\mathcal{C}$-PPM does not need to verify that the two input permutations belong to the class $\mathcal{C}$, and the algorithm may answer arbitrarily on inputs that fail to fulfill this constraint. Decision problems that place this sort of validity restrictions on their inputs are commonly known as *promise problems*.

Our NP-hardness results for $\mathcal{C}$-PPM are based on a general reduction scheme from the classical 3-SAT problem. Given that $\mathcal{C}$-PPM is a promise problem, the reduction must map instances of 3-SAT to valid instances of $\mathcal{C}$-PPM, i.e., the instances where both π and τ belong to $\mathcal{C}$.

On top of NP-hardness arguments, we also provide time-complexity lower bounds for the hard cases of $\mathcal{C}$-PPM. These lower bounds are conditioned on the *exponential-time hypothesis* (ETH), a classical hardness assumption which states that there is a constant $\varepsilon > 0$ such that 3-SAT cannot be solved in time $O(2^{\varepsilon n})$, where n is the number of variables of the 3-SAT instance. In particular, ETH implies that 3-SAT cannot be solved in time $2^{o(n)}$.

Given an instance (π, τ) of $\mathcal{C}$-PPM, we always use n to denote the length of the text τ. We also freely assume that π has length at most n since otherwise the instance can be straightforwardly rejected. Following established practice, we express our complexity bounds for $\mathcal{C}$-PPM in terms of n. Note that inputs of $\mathcal{C}$-PPM of size n actually require $\Theta(n \log n)$ bits to encode.

3 Hardness of PPM

In this section, we present the main technical hardness result and then derive its several corollaries. However, we first need to introduce one more definition.

We say that a permutation class $\mathcal{C}$ has the *$\mathcal{D}$-rich path property* for a class $\mathcal{D}$ if there is a positive constant ε such that for every k, the class $\mathcal{C}$ contains a grid subclass whose cell graph is a proper-turning path of length k with at least $\varepsilon \cdot k$ entries equal to $\mathcal{D}$. Moreover, we say that $\mathcal{C}$ has the *computable $\mathcal{D}$-rich path property*, if $\mathcal{C}$ has the $\mathcal{D}$-rich path property and there is an algorithm that, for a given k, outputs a witnessing proper-turning path of length k with at least $\varepsilon \cdot k$ copies of $\mathcal{D}$ in time polynomial in k.

▶ **Theorem 2.** *Let $\mathcal{C}$ be a permutation class with the computable $\mathcal{D}$-rich path property for a non-monotone-griddable class $\mathcal{D}$. Then $\mathcal{C}$-PPM is NP-complete, and unless ETH fails, there can be no algorithm that solves $\mathcal{C}$-PPM*

- *in time $2^{o(n/\log n)}$ if $\mathcal{D}$ moreover contains any monotone juxtaposition,*
- *in time $2^{o(\sqrt{n})}$ otherwise.*

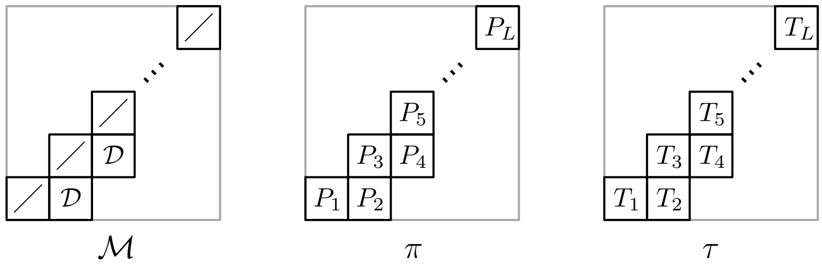

Figure 2 The gridding matrix $\mathcal{M}$, the gridded permutation π (the pattern) and the gridded permutation τ (the text), used in the simplified overview of the proof of Theorem 2.

We remark, without going into detail, that the two lower bounds we obtained under ETH are close to optimal. It is clear that the bound of $2^{o(n/\log n)}$ matches, up to the $\log n$ term in the exponent, the trivial $2^{O(n)}$ brute-force algorithm for PPM. Moreover, the lower bound of $2^{o(\sqrt{n})}$ for $\mathcal{C}$-PPM also cannot be substantially improved without adding assumptions about the class $\mathcal{C}$. Consider for instance the class $\mathcal{C} = \begin{pmatrix} \boxbslash & \boxslash \\ \oplus 21 & \boxbslash \end{pmatrix}$. As we shall see in Proposition 3, this class has the computable $\oplus 21$-rich path property, and therefore the $2^{o(\sqrt{n})}$ conditional lower bound applies to it. However, by using the technique of Ahal and Rabinovich [1], which is based on the concept of treewidth of permutations, we can solve $\mathcal{C}$-PPM (even $\mathcal{C}$-Pattern PPM) in time $n^{O(\sqrt{n})}$. This is because we can show that a permutation $\pi \in \mathcal{C}$ of size n has treewidth at most $O(\sqrt{n})$. We omit the details of the argument here.

3.1 Overview of the proof of Theorem 2

The proof of Theorem 2 is based on a reduction from the well-known 3-SAT problem. The individual steps of the construction are rather technical and in view of the space constraints, we only present here a high-level overview of the construction, while some of the more technical aspects are described in the appendix.

Suppose that $\mathcal{C}$ is a class with the computable $\mathcal{D}$-rich path property, where $\mathcal{D}$ is not monotone griddable. This means that $\mathcal{D}$ contains the Fibonacci class $\oplus 21$ or its reversal $\ominus 12$ as subclass. Suppose then, without loss of generality, that $\mathcal{D}$ contains $\oplus 21$.

To reduce 3-SAT to $\mathcal{C}$-PPM, consider a 3-SAT formula Φ, with n variables $x_1, \dots, x_n$ and m clauses. We may assume that each clause of Φ has exactly 3 literals.

Let $L = L(m, n)$ be an integer whose value will be specified later. By the $\mathcal{D}$-rich path property, $\mathcal{C}$ contains a grid subclass $\mathrm{Grid}(\mathcal{M})$ where the cell graph of $\mathcal{M}$ is a path of length L, in which a constant fraction of cells is equal to $\mathcal{D}$.

To simplify our notation in this high-level overview, we will assume that the cell graph of $\mathcal{M}$ corresponds to an increasing staircase. More precisely, the cells of $\mathcal{M}$ representing infinite classes can be arranged into a sequence $C_1, C_2, \dots, C_L$, where C_1 is the bottom-left cell $\mathcal{M}_{1,1}$ of $\mathcal{M}$, each odd-numbered cell C_{2i-1} corresponds to the diagonal cell $\mathcal{M}_{i,i}$, and each even numbered cell $C_{2i,2i}$ corresponds to $\mathcal{M}_{i+1,i}$. All the remaining cells of $\mathcal{M}$ are empty. To simplify the exposition even further, we will assume that each odd-numbered cell of the path is equal to $\boxslash$ and each even-numbered cell is equal to $\mathcal{D}$. See Figure 2.

With the gridding matrix $\mathcal{M}$ specified above, we will construct two $\mathcal{M}$-gridded permutations, the pattern π and the text τ, such that π can be embedded into τ if and only if the formula Φ is satisfiable. We will describe π and τ geometrically, as permutation diagrams, which are partitioned into blocks by the $\mathcal{M}$-gridding. We let P_i denote the part of π corresponding to the cell C_i of $\mathcal{M}$, and similarly we let T_i be the part of τ corresponding to C_i.

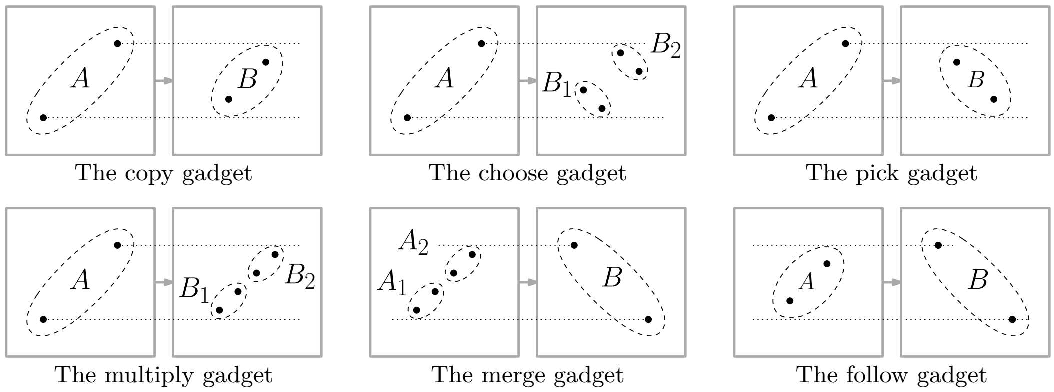

Figure 3 The constructions of simple gadgets. The tile Q_i is always on the left and the tile Q_{i+1} is on the right. The dotted lines show the relative vertical order of points.

To get an intuitive understanding of the reduction, it is convenient to first restrict our attention to *grid-preserving embeddings* of π into τ, that is, to embeddings which map the elements of P_i to elements of T_i for each i.

The basic building blocks in the description of π and τ are the *atomic pairs*, which are specific pairs of points appearing inside a single block P_i or T_i. It is a feature of the construction that in any grid preserving embedding of π into τ, an atomic pair inside a pattern block P_i is mapped to an atomic pair inside the corresponding text block T_i. Moreover, each atomic pair in π or τ is associated with one of the variables $x_1, \dots, x_n$ of Φ, and any grid-preserving embedding will maintain the association, that is, atomic pairs associated to a variable x_j inside π will map to atomic pairs associated to x_j in τ.

To describe π and τ, we need to specify the relative positions of the atomic pairs in two adjacent blocks P_i and P_{i+1} (or T_i and T_{i+1}). These relative positions are given by several typical configurations, which we call *gadgets*. Several examples of gadgets are depicted in Figure 3. In the figure, the pairs of points enclosed by an ellipse are atomic pairs. The choose, multiply and merge gadgets are used in the construction of τ, while the pick and follow gadgets are used in π. The copy gadget will be used in both. We also need more complicated gadgets, namely the *flip gadgets* of Figure 4, which span more than two consecutive blocks. In all cases, the atomic pairs participating in a single gadget are all associated to the same variable of Φ.

The sequence of pattern blocks $P_1, P_2, \dots, P_L$, as well as their corresponding text blocks $T_1, \dots, T_L$, is divided into several contiguous parts, which we call *phases*. We now describe the individual phases in the order in which they appear.

The initial phase and the assignment phase. The initial phase involves a single pattern block P_1 and the corresponding text block T_1. Both P_1 and T_1 consist of an increasing sequence of $2n$ points, divided into n consecutive atomic pairs $X_1^1, X_2^1, \dots, X_n^1 \subseteq P_1$ and $Y_1^1, Y_2^1, \dots, Y_n^1 \subseteq T_1$, numbered in increasing order. The pairs X_j^1 and Y_j^1 are both associated to the variable x_j. Clearly any embedding of P_1 into T_1 will map the pair X_j^1 to the pair Y_j^1, for each $j \in [n]$.

The initial phase is followed by the assignment phase, which also involves only one pattern block P_2 and the corresponding text block T_2. P_2 will consist of an increasing sequence of n atomic pairs $X_1^2, X_2^2, \dots, X_n^2$, where each X_j^2 is a decreasing pair, i.e., a copy of 21. Moreover, $X_j^1 \cup X_j^2$ forms the pick gadget, so the first two pattern blocks can be viewed as a sequence of n pick gadgets stacked on top of each other.

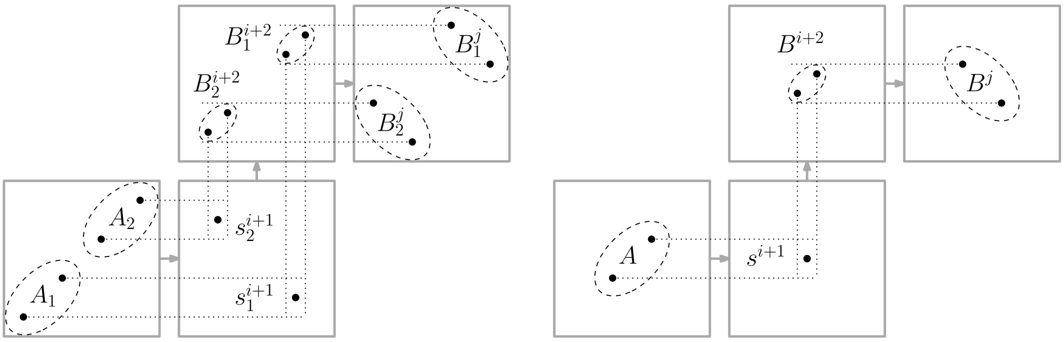

Figure 4 A flip text gadget on the left and a flip pattern gadget on the right. The first tile pictured is Q_i and the last tile is Q_j where $j = i + 3$. As before, the dotted lines show the relative order of points.

The block T_2 then consists of $2n$ atomic pairs $\{Y_j^2, Z_j^2;\ j \in [n]\}$, positioned in such a way that $Y_j^1 \cup Y_j^2 \cup Z_j^2$ is a choose gadget. Thus, $T_1 \cup T_2$ is a sequence of n choose gadgets stacked on top of each other, each associated with one of the variables of Φ.

In a grid-preserving embedding of π into τ, each pick gadget $X_j^1 \cup X_j^2$ must be mapped to the corresponding choose gadget $Y_j^1 \cup Y_j^2 \cup Z_j^2$, with X_j^1 mapped to Y_j^1, and X_j^2 mapped either to Y_j^2 or to Z_j^2. There are thus 2^n grid-preserving embeddings of $P_1 \cup P_2$ into $T_1 \cup T_2$, and these embeddings encode in a natural way to the 2^n assignments of truth values to the variables of Φ. Specifically, if X_j^2 is mapped to Y_j^2, we will say that x_j is false, while if X_j^2 maps to Z_j^2, we say that x_j is true. The aim is to ensure that an embedding of $P_1 \cup P_2$ into $T_1 \cup T_2$ can be extended to an embedding of π into τ if and only if the assignment encoded by the embedding satisfies Φ.

Each atomic pair that appears in one of the text blocks $T_2, T_3, \dots, T_L$ is not only associated with a variable of Φ, but also with its truth value; that is, there are "true" and "false" atomic pairs associated with each variable x_j. The construction of π and τ ensures that in an embedding of π into τ in which X_j^2 is mapped to Y_j^2 (corresponding to setting x_j to false), all the atomic pairs associated to x_j in the subsequent stages of π will map to false atomic pairs associated to x_j in τ, and conversely, if X_j^2 is mapped to Z_j^2, then the atomic pairs of π associated to x_j will only map to the true atomic pairs associated to x_j in τ.

The multiplication phase. The purpose of the multiplication phase is to "duplicate" the information encoded in the assignment phase. Without delving into the technical details, we describe the end result of the multiplication phase and its intended behaviour with respect to embeddings. Let d_j be the number of occurrences (positive or negative) of the variable x_j in Φ. Note that $d_1 + d_2 + \cdots + d_n = 3m$, since Φ has m clauses, each of them with three literals. Let P_k and T_k are the final pattern block and text block of the multiplication phase. Then P_k is an increasing sequence of $3m$ increasing atomic pairs, among which there are d_j atomic pairs associated to x_j. Moreover, the pairs are ordered in such a way that the d_1 pairs associated to x_1 are at the bottom, followed by the d_2 pairs associated to x_2 and so on. The structure of T_k is similar to P_k, except that T_k has $6m$ atomic pairs. In fact, we may obtain T_k from P_k by replacing each atomic pair $X_i^k \subseteq P_k$ associated to a variable x_j by two adjacent atomic pairs Y_i^k, Z_i^k, associated to the same variable, where Y_i^k is false and Z_i^k is true.

It is useful to identify each pair $X_i^k \subseteq P_k$ as well as the corresponding two pairs $Y_i^k, Z_i^k \subseteq T_k$ with a specific occurrence of x_j in Φ. Thus, each literal in Φ is represented by one atomic pair in P_k and two adjacent atomic pairs of opposite truth values in T_k.

The blocks $P_3, \ldots, P_k$ and $T_3, \ldots, T_k$ are constructed in such a way that any embedding of π into τ that encodes an assignment in which x_j is false has the property that all the atomic pairs in P_k associated to x_j are mapped to the false atomic pairs of T_k associated to x_j, and similarly, when x_j is encoded as true in the assignment phase, the pairs of P_k associated to x_j are only mapped to the true atomic pairs of T_k. Thus, the mapping of any atomic pair of P_k encodes the information on the truth assignment of the associated variable.

The multiplication phase is implemented by a combination of multiply gadgets and flip text gadgets in τ, and copy gadgets and flip pattern gadgets in π. It requires no more than $O(\log m)$ blocks in π and τ, i.e., $k = O(\log m)$.

The sorting phase. The purpose of the sorting phase is to rearrange the relative positions of the atomic pairs. While at the end of the multiplication phase, the pairs representing occurrences of the same variable appear consecutively, after the sorting phase, the pairs representing literals belonging to the same clause will appear consecutively. More precisely, letting P_ℓ and T_ℓ denote the last pattern block and the last text block of the sorting phase, P_ℓ has the same number of atomic pairs associated to a given variable x_j as P_k, and similarly for T_ℓ and T_k. If $K_1, \ldots, K_m$ are the clauses of Φ, then for each clause K_j, P_ℓ contains three consecutive atomic pairs corresponding to the three literals in K_j, and T_ℓ contains the corresponding six atomic pairs, again appearing consecutively. Similarly as in P_k and T_k, each atomic pair in P_ℓ must map to an atomic pair in T_ℓ representing the same literal and having the correct truth value encoded in the assignment phase.

To prove Theorem 2, we need two different ways to implement the sorting phase, depending on whether the class $\mathcal{D}$ contains a monotone juxtaposition or not. The first construction, which we call *sorting by gadgets*, does not put any extra assumptions on $\mathcal{D}$. However, it may require up to $\Theta(m)$ blocks to perform the sorting, that is $\ell = \Theta(m)$.

The other implementation of the sorting phase, which we call *sorting by juxtapositions* is only applicable when $\mathcal{D}$ contains a monotone juxtaposition, and it can be performed with only $O(\log m)$ blocks. The difference between the lengths of the two versions of sorting is the reason for the two different lower bounds in Theorem 2.

The evaluation phase. The final phase of the construction is the evaluation phase. The purpose of this phase is to ensure that for any embedding of π into τ, the truth assignment encoded by the embedding satisfies all the clauses of Φ. For each clause K_j, we attach suitable gadgets to the atomic pairs in P_ℓ and T_ℓ representing the literals of K_j. Using the fact that the atomic pairs representing the literals of a given clause are consecutive in P_j and T_j, this can be done for all the clauses simultaneously, with only $O(1)$ blocks in π and τ. This completes an overview of the hardness reduction proving Theorem 2.

When the reduction is performed with sorting by gadgets, it produces permutations π and τ of size $O(m^2)$, since we have $L = O(m)$ blocks and each block has size $O(m)$. When sorting is done by juxtapositions, the number of blocks drops to $L = O(\log m)$, hence π and τ have size $O(m \log m)$. ETH implies that 3-SAT with n variables and m clauses cannot be solved in time $2^{o(m+n)}$ [14]. From this, the lower bounds from Theorem 2 follow.

Further details of the reduction, as well as the correctness proof, are presented in the appendix A.

3.2 Consequences

In the rest of this section, we focus on presenting examples of classes that satisfy the technical "rich path" property, which is the backbone of all our hardness arguments.

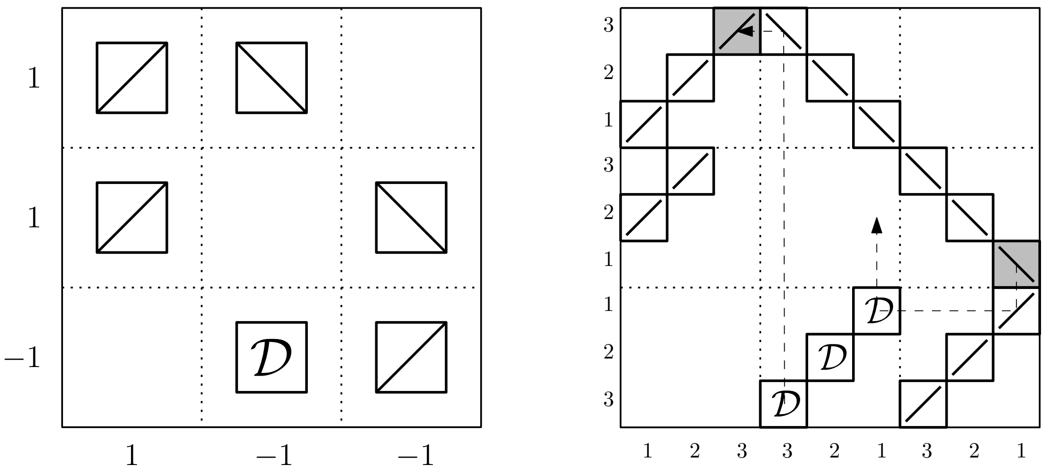

Figure 5 Illustration of the proof of Proposition 3. Left: a gridding matrix $\mathcal{M}$ whose cell graph is a cycle with a single entry equal to $\mathcal{D}$. The numbers along the bottom and the left edge form an orientation that maps each entry to a sum-closed class. Right: a gridding matrix whose grid class is contained in Grid($\mathcal{M}$) and whose cell graph is a path. The endvertices of the path are highlighted in gray, and the first and last few steps of the path are shown as dashed lines. The numbers along the edges are the labels forming the characteristic of each entry.

▶ **Proposition 3.** *Let $\mathcal{D}$ be a non-monotone-griddable class that is sum-closed or skew-closed. If $\mathcal{M}$ is a gridding matrix whose cell graph $G_{\mathcal{M}}$ contains a proper-turning cycle with at least one entry equal to $\mathcal{D}$, then* Grid($\mathcal{M}$) *has the computable $\mathcal{D}$-rich path property.*

Proof. We note that the proof closely follows a proof of a similar claim for monotone grid classes by Jelínek et al. [16, Lemma 3.5].

We may assume, without loss of generality, that the cell graph of $\mathcal{M}$ consists of a single cycle, that it contains a unique entry equal to $\mathcal{D}$, and that all the remaining nonempty entries are equal to ⧄ or to ⧅. This is because each infinite permutation class contains either ⧄ or ⧅ as a subclass, and replacing an entry of $\mathcal{M}$ by its infinite subclass can only change Grid($\mathcal{M}$) into its subclass. If we can establish the $\mathcal{D}$-rich path property for the subclass, then it also holds for the class Grid($\mathcal{M}$) itself.

We may also assume that $\mathcal{D}$ is sum-closed, since the skew-closed case is symmetric. In particular, $\mathcal{D}$ contains $\oplus 21$ as a subclass.

Let L be a given integer. We show how to obtain a grid subclass of Grid($\mathcal{M}$) whose cell graph is a proper-turning path of length at least L that contains a constant fraction of $\mathcal{D}$-entries. Refer to Figure 5. Suppose $\mathcal{N}$ is a $k \times \ell$ gridding matrix whose every entry is either sum-closed or skew-closed. The *refinement* $\mathcal{N}^{\times q}$ of $\mathcal{N}$ is the $qk \times q\ell$ matrix obtained from $\mathcal{N}$ by replacing the entry $\mathcal{N}_{i,j}$ with

- a $q \times q$ diagonal matrix with all the non-empty entries equal to $\mathcal{N}_{i,j}$ if $\mathcal{N}_{i,j}$ is sum-closed,
- a $q \times q$ anti-diagonal matrix with all the non-empty entries equal to $\mathcal{N}_{i,j}$ if $\mathcal{N}_{i,j}$ is skew-closed.

It is easy to see that Grid($\mathcal{N}^{\times q}$) is a subclass of Grid($\mathcal{N}$). We call the submatrix of $\mathcal{N}^{\times q}$ formed by the entries $\mathcal{N}^{\times q}_{a,b}$ for $q \cdot i < a \leq (q+1) \cdot i$ and $q \cdot j < b \leq (q+1) \cdot j$ the *(i,j)-block* of $\mathcal{N}^{\times q}$.

Importantly, it follows from the work of Albert et al. [3, Proposition 4.1] that for every monotone gridding matrix $\mathcal{N}$, there exists a consistent orientation of the refinement $\mathcal{N}^{\times 2}$. Translating it to our setting, we can assume that there is a $k \times \ell$ orientation $\mathcal{F}$ such that the image of $\mathcal{M}_{i,j}$ under $\mathcal{F}$ is sum-closed for every $i \in [k]$ and $j \in [\ell]$. If that is not the case for $\mathcal{M}$, we simply start with $\mathcal{M}^{\times 2}$ instead.

Given such an orientation $\mathcal{F} = (f_c, f_r)$, we label the rows and columns of the refinement $\mathcal{M}^{\times L}$ using the set $[L]$. The L-tuple of columns created from the i-th column of $\mathcal{M}$ is labeled in the increasing order from left to right if $f_c(i)$ is positive and right to left otherwise. Similarly, the L-tuple of rows created from the j-th row of $\mathcal{M}$ is labeled in the increasing order from bottom to top if $f_r(j)$ is positive and top to bottom otherwise. The *characteristic of an entry* in $\mathcal{M}^{\times L}$ is the pair of labels given to its column and row. Observe that each non-empty entry in $\mathcal{M}^{\times L}$ has a characteristic of the form (s, s) for some $s \in [L]$ by the choice of orientation. Therefore, $G_{\mathcal{M}^{\times L}}$ consists exactly of L connected components, each corresponding to a copy of $\mathcal{M}$.

We pick an arbitrary non-empty monotone entry $\mathcal{M}_{i,j}$ of $\mathcal{M}$ and obtain a matrix $\mathcal{M}_L$ by replacing the (i, j)-block in $\mathcal{M}^{\times q}$ with the $q \times q$ matrix whose only non-empty entries are the ones with characteristic $(s, s+1)$ for all $s \in [L-1]$ and they are all equal to $\mathcal{M}_{i,j}$. $\mathrm{Grid}(\mathcal{M}_L)$ is a subclass of $\mathrm{Grid}(\mathcal{M})$ since the modified (i, j)-block corresponds to shifting the original (anti-)diagonal matrix by one row either up or down, depending on the orientation of the j-th row of $\mathcal{M}$.

Observe that we connected all the L copies of $\mathcal{M}$ into a single long path. Moreover, the path contains $L - 1$ entries in the (i, j)-block and L entries in every other non-empty block. Therefore, a constant fraction of its entries belong to the (a, b)-block such that $\mathcal{M}_{a,b} = \mathcal{D}$ and thus are equal to $\mathcal{D}$. It is easy to see that the described procedure is constructive and can easily be implemented to run in polynomial time. Therefore, $\mathrm{Grid}(\mathcal{M})$ indeed has the computable $\mathcal{D}$-rich path property. ◀

Combining Proposition 3 with Theorem 2, we get the following corollary. Note that in the corollary, if $\mathcal{D}$ fails to be sum-closed or skew-closed, we may simply replace it with $\oplus 21$ or $\ominus 12$, since at least one of these two classes is its subclass by Theorem 1.

▶ **Corollary 4.** *Let $\mathcal{D}$ be a non-monotone-griddable class. If $\mathcal{M}$ is a gridding matrix whose cell graph contains a proper-turning cycle with one entry equal to $\mathcal{D}$, then* $\mathrm{Grid}(\mathcal{M})$*-PPM is NP-complete. Moreover, unless ETH fails, there can be no algorithm for* $\mathrm{Grid}(\mathcal{M})$*-PPM running*

- *in time $2^{o(n/\log n)}$ if $\mathcal{D}$ additionally contains any monotone juxtaposition and is either sum-closed or skew-closed,*
- *in time $2^{o(\sqrt{n})}$ otherwise.*

Three symmetry types of patterns of length 4 can be tackled with a special type of grid classes. The *k-step increasing $(\mathcal{C}, \mathcal{D})$-staircase*, denoted by $\mathrm{St}_k(\mathcal{C}, \mathcal{D})$ is a grid class $\mathrm{Grid}(\mathcal{M})$ of a $k \times (k + 1)$ gridding matrix $\mathcal{M}$ such that the only non-empty entries in $\mathcal{M}$ are $\mathcal{M}_{i,i} = \mathcal{C}$ and $\mathcal{M}_{i,i+1} = \mathcal{D}$ for every $i \in [k]$. In other words, the entries on the main diagonal are equal to $\mathcal{C}$ and the entries of the adjacent lower diagonal are equal to $\mathcal{D}$. The *increasing $(\mathcal{C}, \mathcal{D})$-staircase*, denoted by $\mathrm{St}(\mathcal{C}, \mathcal{D})$, is the union of $\mathrm{St}_k(\mathcal{C}, \mathcal{D})$ over all $k \in \mathbb{N}$.

Observe that if $\mathcal{C}$ and $\mathcal{D}$ are two infinite classes and one of them contains $\oplus 21$ or $\ominus 12$ then Theorem 2 applies and $\mathrm{St}(\mathcal{C}, \mathcal{D})$-PPM is NP-complete. Furthermore, if it also contains a monotone juxtaposition as a subclass, then the almost linear lower bound under ETH follows. We proceed to show that three symmetry types of classes avoiding a pattern of length 4 actually contain such a staircase subclass.

▶ **Proposition 5.** *For any sum-indecomposable permutation σ, the class* $\mathrm{St}(\boxslash, \mathrm{Av}(\sigma))$ *is contained in the class* $\mathrm{Av}(1 \ominus \sigma)$.

Proof. Suppose for a contradiction that $\sigma' = 1 \ominus \sigma$ belongs to $\mathrm{St}(\boxslash, \mathrm{Av}(\sigma))$. In particular it belongs to $\mathrm{St}_k(\boxslash, \mathrm{Av}(\sigma))$ for some k and there is a witnessing gridding. If the first element is not mapped to one of the $\boxslash$-entries on the upper diagonal, then the whole σ' must lie in a single $\mathrm{Av}(\sigma)$-entry on the lower diagonal, which is clearly not possible. Therefore, the first element must be mapped to one of the $\boxslash$-entries. Notice that the rest of σ' cannot be mapped to any of the $\boxslash$-entries as it lies below and to the right of the first element. However, it cannot lie in more than one $\mathrm{Av}(\sigma)$-entry; otherwise, we could express σ as a direct sum of two shorter permutations. Hence, there must be an occurrence of σ in an $\mathrm{Av}(\sigma)$-entry which is clearly a contradiction. ◀

A direct consequence of Proposition 5 is that taking σ to be 321, 312 or 231, we see that $\mathrm{St}(\boxslash, \mathrm{Av}(321)) \subseteq \mathrm{Av}(4321)$, $\mathrm{St}(\boxslash, \mathrm{Av}(231)) \subseteq \mathrm{Av}(4231)$ and $\mathrm{St}(\boxslash, \mathrm{Av}(312)) \subseteq \mathrm{Av}(4312)$. Note that the first inclusion is rather trivial and the latter two have been previously observed by Berendsohn [5].

We may easily observe that for any pattern σ of size 3, the class $\mathrm{Av}(\sigma)$ contains the Fibonacci class or its reversal, as well as a monotone juxtaposition. Combining Proposition 5 with Theorem 2 yields the following consequence.

▶ **Corollary 6.** *For any permutation σ that contains a pattern symmetric to* 4321*, to* 4231*, or to* 4312*, the problem* $\mathrm{Av}(\sigma)$*-PPM is NP-complete, and unless ETH fails, it cannot be solved in time* $2^{o(n/\log n)}$.

We verified by computer that there are only five symmetry types of patterns of length 5 that do not contain any of 4321, 4213, 4312 or their symmetries – represented by 14523, 24513, 32154, 42513 and 41352. Of these five, four can be handled by Corollary 4 since they contain a specific type of cyclic grid classes, as we now show.

▶ **Proposition 7.** *The class* $\mathrm{Av}(\sigma)$ *contains the class* $\mathrm{Grid}(\mathcal{M})$ *for the gridding matrix* $\mathcal{M} = \begin{pmatrix} \boxbslash & \boxslash \\ \mathrm{Av}(\pi) & \boxbslash \end{pmatrix}$ *whenever*

- $\pi = 132$ *and* $\sigma = 14523$*, or*
- $\pi = 231$ *and* $\sigma = 24513$*, or*
- $\pi = 321$ *and* $\sigma \in \{32154, 42513\}$.

Proof. Suppose that σ and π are one of the listed cases. Observe that $\mathrm{Grid}(\mathcal{M})$ is a subclass of $\mathrm{Av}(\sigma)$ if and only if σ is not in $\mathrm{Grid}(\mathcal{M})$. For contradiction, suppose that the class $\mathrm{Grid}(\mathcal{M})$ contains σ. Therefore, there exists a witnessing $\mathcal{M}$-gridding $1 = c_1 \le c_2 \le c_3 = 6$ and $1 = r_1 \le r_2 \le r_3 = 6$ of σ.

Let us consider the four choices of σ separately, starting with $\sigma = 14523$: if $c_2 \le 3$ and $r_2 \le 3$, the cell $(2, 2)$ of the gridding contains the pattern 21, if $c_2 \le 4$ and $r_2 \ge 4$, the cell $(2, 1)$ contains 12, if $c_2 \ge 4$ and $r_2 \le 4$, the cell $(1, 2)$ contains 12, and if $c_2 \ge 5$ and $r_2 \ge 5$, the cell $(1, 1)$ contains 132. In all cases we get a contradiction with the properties of the $\mathcal{M}$-gridding. The same argument applies to $\sigma = 14513$, except in the last case we use the pattern 231 instead of 132.

For $\sigma = 32154$, the four cases to consider are $c_2 \le 4 \wedge r_2 \le 4$, $c_2 \ge 5 \wedge r_2 \le 3$, $c_2 \le 3 \wedge r_2 \ge 5$, and $c_2 \ge 4 \wedge r_2 \ge 4$, in each case getting contradiction in a different cell of the gridding. For $\sigma = 42513$, the analogous argument distinguishes the cases $c_2 \le 3 \wedge r_2 \le 3$, $c_2 \ge 4 \wedge r_2 \le 4$, $c_2 \le 4 \wedge r_2 \ge 4$, and $c_2 \ge 5 \wedge r_2 \ge 5$. ◀

It is easy to see that every σ of length at least 6 contains a pattern of size 5 which is not symmetric to 41352. Therefore, $\mathrm{Av}(\sigma)$-PPM is NP-complete for all permutations σ of length at least 4 except for one symmetry type of length 5 and for four out of seven symmetry types of length 4. As $\mathrm{Av}(\sigma)$-PPM is polynomial-time solvable for any σ of length at most 3, these are, in fact, the only cases left unsolved.

▶ **Corollary 8.** *If σ is a permutation of length at least 4 that is not in symmetric to any of* $3412, 3142, 4213, 4123$ *or* 41352*, then* $\mathrm{Av}(\sigma)$*-PPM is NP-complete, and unless ETH fails, it cannot be solved in time* $2^{o(n/\log n)}$.

To conclude this section, we remark that the suitable grid subclasses were discovered via computer experiments facilitated by the Permuta library [4].

4 Polynomial-time algorithm

We say that a permutation π is *t-monotone* if there is a partition $\Pi = (S_1, \ldots, S_t)$ of S_π such that S_i is a monotone point set for each $i \in [t]$. The partition Π is called a *t-monotone partition*.

Given a t-monotone partition $\Pi = (S_1, \ldots, S_t)$ of a permutation π and a t-monotone partition $\Sigma = (S'_1, \ldots, S'_t)$ of τ, an embedding ϕ of π into τ is a *(Π, Σ)-embedding* if $\phi(S_i) \subseteq S'_i$ for every $i \in [t]$. Guillemot and Marx [11] showed that if we fix a t-monotone partitions of both π and τ, the problem of finding a (Π, Σ)-embedding is polynomial-time solvable.

▶ **Proposition 9** (Guillemot and Marx [11])**.** *Given a permutation π of length m with a t-monotone partition Π and a permutation τ of length n with a t-monotone partition Σ, we can decide if there is a (Π, Σ)-embedding of π into τ in time $O(m^2n^2)$.*

We can combine this result with the fact that there is only a bounded number of ways how to grid a permutation, and obtain the following counterpart to Corollary 4.

▶ **Theorem 10.** *$\mathcal{C}$-PPM is polynomial-time solvable for any monotone-griddable class $\mathcal{C}$.*

Proof. Let $\mathcal{M}$ be a $k \times \ell$ monotone gridding matrix such that $\mathrm{Grid}(\mathcal{M})$ contains the class $\mathcal{C}$. We have to decide whether π is contained in τ for two given permutations π of length m and τ of length n, both belonging to the class $\mathcal{C}$.

First, we find an $\mathcal{M}$-gridding of τ. We enumerate all possible $k \times \ell$ griddings and for each, we test if it is a valid $\mathcal{M}$-gridding. Observe that there are in total $O(n^{k+\ell-2})$ such griddings since they are determined by two sequences of values from the set $[n]$, one of length $k-1$ and the other of length $\ell - 1$. Moreover, it is straightforward to test in time $O(n^2)$ whether a given $k \times \ell$ gridding is in fact an $\mathcal{M}$-gridding. Note that we are guaranteed to find an $\mathcal{M}$-gridding as τ belongs to $\mathcal{C} \subseteq \mathrm{Grid}(\mathcal{M})$. We set Σ to be the $(k \cdot \ell)$-monotone partition of τ into the monotone sequences given by the individual cells of the gridding.

In the second step, we enumerate all possible $\mathcal{M}$-griddings of π. As with τ, we enumerate all possible $O(m^{k+\ell-2})$ $k \times \ell$ griddings of π and check for each gridding whether it is actually an $\mathcal{M}$-gridding in time $O(m^2)$. For each $\mathcal{M}$-gridding found, we let Π be the $(k \cdot \ell)$-monotone partition of π given by the gridding, and we apply Proposition 9 to test whether there is a (Π, Σ)-embedding in time $O(m^2n^2)$.

If there is an embedding ϕ of π into τ, there is a $(k \cdot \ell)$-monotone partition Σ' of π such that ϕ is a (Π, Σ')-embedding. Therefore, the algorithm correctly solves $\mathcal{C}$-PPM in time $O(n^{k+\ell} + m^{k+\ell}n^2)$ – polynomial in n, m. ◀

Notice that if $\mathcal{M}$ is a gridding matrix whose every entry is monotone griddable, or equivalently no entry contains the Fibonacci class or its reverse as a subclass, then the class Grid($\mathcal{M}$) is monotone griddable as well. It follows that for such $\mathcal{M}$, the Grid($\mathcal{M}$)-PPM problem is polynomial-time solvable. We also note that recent results on $\mathcal{C}$-Pattern PPM [16] imply that if a gridding matrix $\mathcal{M}$ has an acyclic cell graph, and if every nonempty cell is either monotone or symmetric to a Fibonacci class, then Grid($\mathcal{M}$)-Pattern PPM, and therefore also Grid($\mathcal{M}$)-PPM, is polynomial-time solvable as well. These two tractability results contrast with our Corollary 4, which shows that for any gridding matrix $\mathcal{M}$ whose cell graph is a cycle, and whose nonempty cells are all monotone except for one Fibonacci cell, Grid($\mathcal{M}$)-PPM is already NP-hard.

5 Open problems

We have presented a hardness reduction which allowed us to show that the Av(σ)-PPM problem is NP-complete for every permutation σ of size at least 6, as well as for most shorter choices of σ. Nevertheless, for several symmetry types of σ, the complexity of Av(σ)-PPM remains open. We collect all the remaining unresolved cases as our first open problem.

▶ **Open problem 1.** *What is the complexity of* Av(σ)*-PPM, when σ is a permutation from the set* $\{3412, 3142, 4213, 4123, 41352\}$*?*

Our hardness results are accompanied by time complexity lower bounds based on the ETH. Specifically, for our NP-hard cases, we show that under ETH, no algorithm may solve $\mathcal{C}$-PPM in time $2^{o(\sqrt{n})}$. The lower bound can be improved to $2^{o(n/\log n)}$ under additional assumptions about $\mathcal{C}$. This opens the possibility of a more refined complexity hierarchy within the NP-hard cases of $\mathcal{C}$-PPM. In particular, we may ask for which $\mathcal{C}$ can $\mathcal{C}$-PPM be solved in subexponential time.

▶ **Open problem 2.** *Which cases of $\mathcal{C}$-PPM can be solved in time $2^{O(n^{1-\varepsilon})}$? Can the general PPM problem be solved in time $2^{o(n)}$?*

References

1 Shlomo Ahal and Yuri Rabinovich. On complexity of the subpattern problem. *SIAM J. Discrete Math.*, 22(2):629–649, 2008. doi:10.1137/S0895480104444776.

2 Michael Albert, Marie-Louise Lackner, Martin Lackner, and Vincent Vatter. The complexity of pattern matching for 321-avoiding and skew-merged permutations. *Discrete Math. Theor. Comput. Sci.*, 18(2):Paper No. 11, 17, 2016. URL: https://dmtcs.episciences.org/2607.

3 Michael H. Albert, M. D. Atkinson, Mathilde Bouvel, Nik Ruškuc, and Vincent Vatter. Geometric grid classes of permutations. *Trans. Amer. Math. Soc.*, 365(11):5859–5881, 2013. doi:10.1090/S0002-9947-2013-05804-7.

4 Ragnar Pall Ardal, Tomas Ken Magnusson, Émile Nadeau, Bjarni Jens Kristinsson, Bjarki Agust Gudmundsson, Christian Bean, Henning Ulfarsson, Jon Steinn Eliasson, Murray Tannock, Alfur Birkir Bjarnason, Jay Pantone, and Arnar Bjarni Arnarson. Permuta, 2021. doi:10.5281/zenodo.4725759.

5 Benjamin Aram Berendsohn. Complexity of permutation pattern matching. Master's thesis, Freie Universität Berlin, Berlin, 2019. URL: https://www.mi.fu-berlin.de/inf/groups/ag-ti/theses/master_finished/berendsohn_benjamin/index.html.

6 Benjamin Aram Berendsohn, László Kozma, and Dániel Marx. Finding and counting permutations via CSPs. In Bart M. P. Jansen and Jan Arne Telle, editors, *14th International Symposium on Parameterized and Exact Computation, IPEC 2019, September 11-13, 2019, Munich, Germany*, volume 148 of *LIPIcs*, pages 1:1–1:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.IPEC.2019.1.

7 Prosenjit Bose, Jonathan F. Buss, and Anna Lubiw. Pattern matching for permutations. *Inform. Process. Lett.*, 65(5):277–283, 1998. doi:10.1016/S0020-0190(97)00209-3.

8 Marie-Louise Bruner and Martin Lackner. A fast algorithm for permutation pattern matching based on alternating runs. *Algorithmica*, 75(1):84–117, 2016. doi:10.1007/s00453-015-0013-y.

9 P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Math.*, 2:463–470, 1935. URL: http://www.numdam.org/item?id=CM_1935__2__463_0.

10 Jacob Fox. Stanley–Wilf limits are typically exponential. arXiv:1310.8378v1, 2013.

11 Sylvain Guillemot and Dániel Marx. Finding small patterns in permutations in linear time. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 82–101. ACM, New York, 2014. doi:10.1137/1.9781611973402.7.

12 Sylvain Guillemot and Stéphane Vialette. Pattern matching for 321-avoiding permutations. In *Algorithms and computation*, volume 5878 of *Lecture Notes in Comput. Sci.*, pages 1064–1073. Springer, Berlin, 2009. doi:10.1007/978-3-642-10631-6_107.

13 Sophie Huczynska and Vincent Vatter. Grid classes and the Fibonacci dichotomy for restricted permutations. *Electron. J. Combin.*, 13(1):Research Paper 54, 14, 2006. URL: http://www.combinatorics.org/Volume_13/Abstracts/v13i1r54.html.

14 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. System Sci.*, 62(2):367–375, 2001. Special issue on the Fourteenth Annual IEEE Conference on Computational Complexity (Atlanta, GA, 1999). doi:10.1006/jcss.2000.1727.

15 Vít Jelínek and Jan Kynčl. Hardness of permutation pattern matching. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 378–396. SIAM, Philadelphia, PA, 2017. doi:10.1137/1.9781611974782.24.

16 Vít Jelínek, Michal Opler, and Jakub Pekárek. A complexity dichotomy for permutation pattern matching on grid classes. In Javier Esparza and Daniel Kráľ, editors, *45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, August 24-28, 2020, Prague, Czech Republic*, volume 170 of *LIPIcs*, pages 52:1–52:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.MFCS.2020.52.

17 C. Schensted. Longest increasing and decreasing subsequences. *Canad. J. Math.*, 13:179–191, 1961. doi:10.4153/CJM-1961-015-3.

18 Vincent Vatter and Steve Waton. On partial well-order for monotone grid classes of permutations. *Order*, 28(2):193–199, 2011. doi:10.1007/s11083-010-9165-1.

A The proof of Theorem 2

Our job is to construct a pair of permutations π and τ, both having a gridding corresponding to a $\mathcal{D}$-rich path, with the property that the embeddings of π into τ will simulate satisfying assignments of a given 3-SAT formula. First, we introduce the concept of $\mathcal{F}$-assembly which enables us to describe constructions of gridded permutations from a grid class $\mathrm{Grid}(\mathcal{M})$ somewhat independently from the actual shape of $\mathcal{M}$.

A.1 $\mathcal{F}$-assembly

A finite subset P of the m-box in general position is called an *m-tile* and a *$k \times \ell$ family of m-tiles* is a set $\mathcal{P} = \{P_{i,j} \mid i \in [k], j \in [\ell]\}$ where each $P_{i,j}$ is an m-tile. Let $\mathcal{F} = (f_c, f_r)$ be a $k \times \ell$ orientation and let $\mathcal{P}$ be a family of m-tiles $P_{i,j}$ for $i \in [k]$, $j \in [\ell]$. The *$\mathcal{F}$-assembly of $\mathcal{P}$* is the point set S defined as follows.

We define for every $i \in [k]$, $j \in [\ell]$ the point set $P'_{i,j} = \{p + (i \cdot m, j \cdot m) \mid p \in \Phi_i(\Psi_j(P_{i,j}))\}$ where Φ_i is an identity if $f_c(i) = 1$ and reversal otherwise, while Ψ_j is an identity if $f_r(j) = 1$ and complement otherwise. We set $S = \bigcup P'_{i,j}$ to be the $\mathcal{F}$-assembly of $\mathcal{P}$. If S is not in general position, we rotate it clockwise by a tiny angle to a general position without changing the order of any points that originally did not share a common coordinate.

Let $\mathcal{M}$ be a $k \times \ell$ gridding matrix. We say that the *image of* $\mathcal{M}_{i,j}$ *under* $\mathcal{F}$ is the class $\Phi_i(\Psi_j(\mathcal{M}_{i,j}))$. The $\mathcal{F}$*-image of* $\mathcal{M}$, denoted by $\mathcal{F}(\mathcal{M})$, is then the $k \times \ell$ gridding matrix defined as $\mathcal{F}(\mathcal{M})_{i,j} = \Phi_i(\Psi_j(\mathcal{M}_{i,j}))$.

▶ **Observation 11.** *Let $\mathcal{M}$ be a $k \times l$ gridding matrix, let $\mathcal{F}$ be a $k \times \ell$ orientation and $\mathcal{P}$ a $k \times \ell$ family of m-tiles. If for every $i \in [k]$ and $j \in [\ell]$ the reduction of $P_{i,j}$ belongs to the class $\mathcal{F}(\mathcal{M})_{i,j}$ then the reduction of the $\mathcal{F}$-assembly of $\mathcal{P}$ belongs to* $\mathrm{Grid}(\mathcal{M})$.

Furthermore, if the cell graph $G_{\mathcal{M}}$ of a monotone gridding matrix $\mathcal{M}$ is acyclic, we can always find an orientation $\mathcal{F}$ such that the image $\mathcal{F}(\mathcal{M})_{i,j}$ of every non-empty entry $\mathcal{M}_{i,j}$ is the class $\boxslash$. We say that such $\mathcal{F}$ is a *consistent orientation* for the gridding matrix $\mathcal{M}$. The existence of consistent orientations is guaranteed for matrices with acyclic cell graphs.

▶ **Lemma 12** (Vatter and Waton [18]). *There exists a consistent $k \times \ell$ orientation $\mathcal{F}$ for any monotone $k \times \ell$ gridding matrix $\mathcal{M}$ whose cell graph $G_{\mathcal{M}}$ is acyclic.*

A.2 The reduction

Let Φ be a given 3-CNF formula with n variables $x_1, x_2, \dots, x_n$ and m clauses $K_1, K_2, \dots, K_m$ each containing exactly three literals. Let $\mathcal{M}$ be a $g \times h$ gridding matrix such that $\mathrm{Grid}(\mathcal{M})$ is a subclass of $\mathcal{C}$, the cell graph $G_{\mathcal{M}}$ is a proper-turning path of sufficient length to be determined later with a constant fraction of its entries is equal to $\mathcal{D}$.

First, we label the vertices of the path as $p_1, p_2, p_3, \dots$ choosing the direction such that at least half of the $\mathcal{D}$-entries share a row with their predecessor. By application of Lemma 12, there is a $g \times h$ orientation $\mathcal{F}$ such that the class $\mathcal{F}(\mathcal{M})_{i,j}$ is equal to $\boxslash$ for every monotone entry $\mathcal{M}_{i,j}$ and the class $\mathcal{F}(\mathcal{M})_{i,j}$ contains $\oplus 21$ for every $\mathcal{D}$-entry $\mathcal{M}_{i,j}$. Our plan is to simultaneously construct two $g \times h$ families of tiles $\mathcal{P}$ and $\mathcal{T}$ and then set π and τ to be the $\mathcal{F}$-assemblies of $\mathcal{P}$ and $\mathcal{T}$, respectively. We abuse the notation and for any $g \times h$ family of tiles $\mathcal{Q}$ (in particular for $\mathcal{P}$ and $\mathcal{T}$), we use Q_i instead of Q_{p_i} to denote the tile corresponding to the i-th cell of the path.

For now, we will only consider restricted embeddings. We say that an embedding of π into τ where π is an $\mathcal{F}$-assembly of $\mathcal{P}$ and τ is an $\mathcal{F}$-assembly of $\mathcal{T}$, is *grid-preserving* if the image of tile $P_{i,j}$ is mapped to the image of $T_{i,j}$ for every i and j. We slightly abuse the notation in the case of grid-preserving embeddings and say that a point q in the tile $P_{i,j}$ is mapped to a point r in the tile $T_{i,j}$ instead of saying that the image of q under the $\mathcal{F}$-assembly is mapped to the image of r under the $\mathcal{F}$-assembly. We say that a pair of points r, q in the tile Q_i *sandwiches* a set of points A in the tile Q_{i+1} if for every point $t \in A$ $r.y < t.y < q.y$ in case p_i and p_{i+1} occupy a common row or otherwise, if the same holds for the x-coordinates.

A.2.1 Gadgets

We construct the tiles from gadgets consisting of pairs of points that we call *atomic pairs*. We assume that the tiles are formed as direct sums of the individual gadgets. Consequently, if $A, B \subseteq Q_i$ are point sets of two different gadgets, then either whole A lies to the right and above B or vice versa. We describe the gadgets in the case when p_i and p_{i+1} share a common row and p_i is to the left of p_{i+1}, as the other cases are symmetric. The gadgets are fully described by the relative positions of their points, therefore we refer the reader to Figure 3 for their definitions.

We say that the copy, pick and follow gadgets connect the pair A to the pair B and the multiply gadget multiplies the pair A to B_1 and B_2. The choose gadget is said to branch the pair A to B_1 and B_2 and the merge gadget merges the pairs A_1 and A_2 into the pair B. We follow with two observations about the behavior of these gadgets.

▶ **Observation 13.** *Suppose there is a choose gadget branching an atomic pair A^T in the tile T_i to two atomic pairs B_1^T and B_2^T in the tile T_{i+1}, and a pick gadget in $\mathcal{P}$ connecting an atomic pair A^P in the tile P_i to an atomic pair B^P in the tile P_{i+1}. In any grid-preserving embedding of π into τ, if A^P is mapped to A^T then B^P is mapped either to B_1^T or to B_2^T.*

▶ **Observation 14.** *Suppose there is a merge gadget merging atomic pairs A_1^T and A_2^T in the tile T_i into an atomic pair B^T in the tile T_{i+1}, and a follow gadget connecting an atomic pair A^P in the tile P_i to an atomic pair B^P in the tile P_{i+1}. In any grid-preserving embedding of π into τ, if A^P is mapped to A_α^T for some $\alpha \in \{1, 2\}$ then B^P is mapped to B^T.*

The flip gadget

We proceed to define two gadgets – a flip text gadget and a flip pattern gadget. It is insufficient to consider just two neighboring tiles as we need two $\mathcal{D}$-entries for the construction. To that end, let i and j be indices such that both p_{i+1} and p_j are $\mathcal{D}$-entries and there is no other $\mathcal{D}$-entry between them.

As before, suppose that A_1 and A_2 are two atomic pairs in Q_i. The *flip text gadget* attached to the atomic pairs A_1 and A_2 consists of two points s_1^{i+1}, s_2^{i+1} in the tile Q_{i+1} and two atomic pairs B_1^k and B_2^k in each tile Q_k for every $k \in [i+2, j] = \{i+2, i+3, \dots, j\}$. The points s_1^{i+1}, s_2^{i+1} form an occurrence of 21 and s_α^{i+1} is sandwiched by A_α for each $\alpha \in \{1, 2\}$.

The atomic pairs B_1^k, B_2^k for $k \in [i+2, j-1]$ are set such that B_2^k lies to the left and below of B_1^k, and together, they form an occurrence of 1234. The only difference in the case of atomic pairs B_1^j, B_2^j is that they form an occurrence of 2143. For every $k \in [i+3, j]$ and $\alpha \in \{1, 2\}$, the pair B_α^k sandwiches the pair B_α^{k-1} and moreover, the pair B_α^{i+2} sandwiches the point s_α^{i+1}. We say that the flip text gadget flips the pairs A_1, A_2 in Q_i to the pairs B_2^j, B_1^j in Q_j. See the left part of Figure 4.

We define the *flip pattern gadget* as a set of points isomorphic to the pairs B_1^k for all k together with the point s_1^{i+2}. See the right part of Figure 4.

Observe that flip gadget propagates the mapping properties while switching the order of the pairs. However, it can also be used to test if only one of its initial atomic pairs is used in the embedding. We omit proofs of the following lemmas due to space constraints.

▶ **Lemma 15.** *Suppose there is a flip pattern gadget connecting an atomic pair $\overline{A}$ in P_i with an atomic pair $\overline{B}$ in P_j. Furthermore, suppose that there is a flip text gadget flipping atomic pairs A_1 and A_2 in T_i to atomic pairs B_2 and B_1 in T_j. In any grid-preserving embedding of π into τ, if $\overline{A}$ is mapped to A_α for some $\alpha \in \{1, 2\}$ then $\overline{B}$ is mapped to B_α.*

▶ **Lemma 16.** *Suppose that there are two flip pattern gadgets connecting an atomic pair A_α^P in P_i to an atomic pair B_α^P in P_j for $\alpha \in \{1, 2\}$. Suppose that there is a flip text gadget in $\mathcal{T}$ that flips atomic pairs A_1^T and A_2^T in T_i to atomic pairs B_2^T and B_1^T in T_j. There cannot exist a grid-preserving embedding ϕ of π into τ that maps A_α^P to A_α^T for each $\alpha \in \{1, 2\}$.*

Note that all the gadgets except for the copy and multiply ones require a non-monotone entry and thus, we need to somehow bridge the segments of the path consisting only of monotone entries. By *attaching a gadget* to the pair A, we mean connecting to A a chain of copy gadgets leading all the way to its first non-monotone entry and then attaching the desired gadget at the end of this chain.

A.2.2 Constructing the $\mathcal{C}$-PPM instance

We define the initial tile P_1 to contain atomic pairs X_k^0 for $k \in [n]$ and the initial tile T_1 to contain atomic pairs Y_k^0 for $k \in [n]$ where each X_k^0 and Y_k^0 form an occurrence of 12 and X_i^0 (Y_j^0) lies to the left and below of X_j^0 (Y_i^0) for every $i < j$. Any grid-preserving embedding of π into τ must obviously map X_k^0 to Y_k^0 for every $k \in [n]$. We describe the rest of the construction in four distinct phases.

Assignment phase. In the first phase, we simulate the assignment of truth values to the variables. To that end, we attach to each pair Y_k^0 for $k \in [n]$ a choose gadget that branches Y_k^0 to two atomic pairs $Y_{k,1}^1$ and $Z_{k,1}^1$ and we attach to each pair X_k^0 for $k \in [n]$ a pick gadget that connects X_k^0 to an atomic pair $X_{k,1}^1$. The properties of choose and pick gadgets imply that in any grid-preserving embedding, $X_{k,1}^1$ is either mapped to $Y_{k,1}^1$ or to $Z_{k,1}^1$.

Multiplication phase. Our next goal is to multiply the atomic pairs corresponding to a single variable into as many pairs as there are occurrences of this variable in the clauses. We describe the gadgets dealing with each variable individually.

Fix $k \in [n]$ and let m_k for $k \in [n]$ denote the total number of occurrences of x_k and $\neg x_k$ in Φ. We are going to describe the construction inductively in $\lceil \log m_k \rceil$ steps. Fix $i \geq 1$. We add for each $j \in [2^i]$ three multiply gadgets, one that multiplies the atomic pair $X_{k,j}^i$ to atomic pairs $\widetilde{X}_{k,2j-1}^{i+1}$ and $\widetilde{X}_{k,2j}^{i+1}$, one that multiplies the pair $Y_{k,j}^i$ to $\widetilde{Y}_{k,2j-1}^{i+1}$ and $\widetilde{Y}_{k,2j}^{i+1}$, and finally one that multiplies $Z_{k,j}^i$ to $\widetilde{Z}_{k,2j-1}^{i+1}$ and $\widetilde{Z}_{k,2j}^{i+1}$. Observe that the properties of gadgets imply that for arbitrary $j \in [2^{i+1}]$, $\widetilde{X}_{k,j}^{i+1}$ maps either to $\widetilde{Y}_{k,j}^{i+1}$ or to $\widetilde{Z}_{k,j}^{i+1}$. However in the text, we have the quadruple $\widetilde{Y}_{k,2j-1}^{i+1}, \widetilde{Y}_{k,2j}^{i+1}, \widetilde{Z}_{k,2j-1}^{i+1}, \widetilde{Z}_{k,2j}^{i+1}$ in this specific order.

To solve this, we add for each $j \in [2^i]$ a flip text gadget that flips $\widetilde{Y}_{k,2j}^{i+1}$, $\widetilde{Z}_{k,2j-1}^{i+1}$ to atomic pairs $Z_{k,2j-1}^{i+1}, Y_{k,2j}^{i+1}$. The properties of flip gadgets (Lemma 15) guarantee that for every $j \in [2^{i+1}]$, the pair $X_{k,j}^{i+1}$ is mapped either to $Y_{k,j}^{i+1}$ or to $Z_{k,j}^{i+1}$. Moreover, the order of atomic pairs in the text now alternates between Y and Z as desired. It follows that we need in total $O(\log m)$ $\mathcal{D}$-entries for the multiplication phase.

Sorting phase. The multiplication phase ended with atomic pairs $X_{k,j}^i$ in the pattern ordered lexicographically by (k, j), i.e., bundled in blocks by the variables. The goal of the sorting phase is to rearrange them such that they become bundled by clauses.

First, we remark that it is possible to flip the order of any two neighboring pairs in the pattern using only a $O(1)$ layers of gadgets. Unfortunately, the space constraints make it impossible to include the description here as it is quite involved and technical. Using this approach, we can flip an arbitrary set of neighboring pairs in $O(1)$ layers of gadgets and thus, we can arbitrarily reshuffle the atomic pairs of the pattern in $O(m)$ layers.

On the other hand, we describe how we can do the sorting phase using significantly fewer fewer tiles if the class $\mathcal{D}$ contains a monotone juxtaposition. We shall discuss here only the case when $\mathcal{D}$ contains the juxtaposition $\mathcal{B} = \text{Grid}(⧄\ ⧄)$ as the other cases can be solved using the same technique. Let p_i be an entry such that $\mathcal{F}(\mathcal{M})_{p_i}$ contains $\mathcal{B}$ and recall that we assumed that p_i shares a common row with p_{i-1}. We construct a tile Q_i from two increasing sets Q_i^1 and Q_i^2 placed next to each other. In particular, we can attach to any atomic pair A in Q_{i-1} a copy gadget connecting A to an atomic pair B and choose arbitrarily whether B lies in Q_i^1 or Q_i^2.

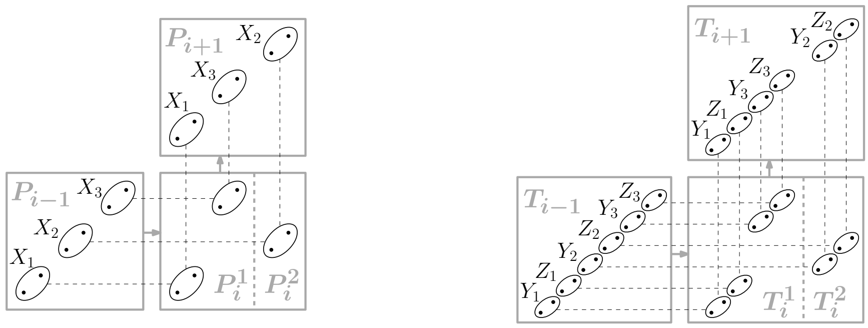

Figure 6 Example of one sorting step using Grid(⧄ ⧄) and a partition $\{1, 2, 3\} = \{1, 3\} \cup \{2\}$.

Let J_1 and J_2 be a partition of the set $[3m]$. We attach a copy gadget ending in Q_i^α to each X_j, Y_j and Z_j with $j \in J_\alpha$ for each $\alpha \in \{1, 2\}$. In this way, we rearranged the atomic pairs in $\mathcal{P}$ such that first we have all pairs X_j such that $j \in J_1$ followed by all pairs X_j for $j \in J_2$. Similarly in $\mathcal{T}$, we have Y_j, Z_j for $j \in J_1$ followed by Y_j, Z_j for $j \in J_2$. See Figure 6.

Notice that the described operation simulates a stable bucket sort with two buckets. Therefore, we can simulate radix sort and rearrange the atomic pairs into arbitrary order given by σ by iterating this operation $O(\log m)$ times. In this way, the whole sorting phase uses only $O(\log m)$ entries equal to $\mathcal{D}$.

Evaluation phase. In the evaluation phase, we test whether each clause $K_j = (x_a \vee x_b \vee x_c)$ is satisfied. We consider the case when K_j contains only positive literals as clauses with negative literals can be handled with minor modifications of the argument. Suppose X_a, X_b and X_c are the three neighboring atomic pairs in $\mathcal{P}$ that correspond to the three literals in K_j. In $\mathcal{T}$, there are six neighboring atomic pairs Y_a, Z_a, Y_b, Z_b, Y_c, Z_c such that in any grid-preserving embedding, the pair X_α is mapped to either Y_α or Z_α for every $\alpha \in \{a, b, c\}$.

We abuse the notation and use the same letters to denote atomic pairs in different tiles so that the gadgets carry the names through. First, we add (i) a choose gadget that branches Z_b to $\overline{Z}_b$ and $\widetilde{Z}_b$, and (ii) pick gadgets to $Y_a, Z_a, Y_b, Y_c, Z_c, X_a, X_b$ and X_c. We continue with adding two layers of flip gadgets, modifying the order of atomic pairs in the text as follows

$$Y_a Z_a Y_b \overline{Z}_b \widetilde{Z}_b Y_c Z_c \rightarrow Y_a Z_a \overline{Z}_b Y_b \widetilde{Z}_b Z_c Y_c \rightarrow Y_a \overline{Z}_b Z_a Y_b Z_c \widetilde{Z}_b Y_c.$$

Observe that either X_b is mapped to Y_b and thus K_j is satisfied, or X_b is mapped to one of $\overline{Z}_b$ and $\widetilde{Z}_b$. Subsequently, the order of pairs in the final tile guarantees by Lemma 16 that simultaneously, X_a cannot map to Z_a and X_c to Z_c and thus, K_j must be satisfied.

That concludes the construction of $\mathcal{P}$ and $\mathcal{T}$. Observe that each tile in both $\mathcal{P}$ and $\mathcal{T}$ contains $O(m)$ points. If $\mathcal{D}$ contains a monotone juxtaposition, then $|\pi|, |\tau| \in O(m \log m)$ and otherwise, $|\pi|, |\tau| \in O(m^2)$. This gives rise to the two different conditional lower bounds.

Beyond grid-preserving embeddings. First, we modify both π and τ such that any embedding that maps the image of P_1 to the image of T_1 must already be grid-preserving. To that end, we add atomic pairs A_1, A_2 to the initial tile P_1 such that A_1 is to the left and below everything else and A_2 is to the right and above everything else. We then attach to both A_1, A_2 a chain of copy gadgets going all the way to the last tile of the path. We modify $\mathcal{T}$, in the same way, using chains of copy gadgets originating in the atomic pairs B_1 and B_2. Observe that in any embedding that sends P_1 to T_1, the image of A_α is mapped to the image of B_α for each $\alpha \in \{1, 2\}$. The chain of copy gadgets attached to A_α then must map to the chain of gadgets attached to B_α and these chains force the embedding to be grid-preserving.

Finally, we modify π and τ to obtain permutations π' and τ' such that any embedding of π' into τ' can be translated to an embedding of π into τ that maps P_1 to T_1 and vice versa. Let r and q be the lowest and topmost points in P_1 and similarly, let s and t be the lowest and topmost points of T_1. The family of tiles $\mathcal{P}'$ is obtained from $\mathcal{P}$ by inflating both r and q with an increasing sequence of length $|\tau| + 1$ and similarly, the family $\mathcal{T}'$ is obtained from $\mathcal{T}$ by inflating both s and t. We call the points obtained by inflating r and q *lower anchors* and the ones obtained by inflating s and t *upper anchors*. We let π' and τ' be the $\mathcal{F}$-assemblies of $\mathcal{P}'$ and $\mathcal{T}'$. Observe that these modifications did not change the asymptotic size of the input as $|\pi'| = O(|\tau|)$ and $|\tau'| = O(|\tau'|) = O(|\tau|)$.

A.3 Correctness

The "only if" part. Let Φ be a satisfiable formula and fix arbitrary satisfying assignment. We map the image of P_1 to the image of T_1. In the assignment phase, we map the pair $X^1_{k,1}$ to $Y^1_{k,1}$ if x_k is set to true, otherwise we map it to $Z^1_{k,1}$. The embedding of the multiplication and sorting phase is uniquely determined by the gadgets. It is easy to check that for each satisfied clause K_j there is a way to extend the mapping to the evaluation phase.

The "if" part. Let ϕ be an embedding of π' into τ'. The total length of the anchors in both π' and τ' is $2|\tau| + 2$. Therefore, at least $|\tau| + 2$ points of the anchors in π' must be mapped to the anchors in τ' and in particular, there is at least one point in each anchor of π' that maps to corresponding anchor in τ'. The chains of copy gadgets attached to A_1 and A_2 force the rest of the embedding to be grid-preserving, and thus it straightforwardly translates to a grid-preserving embedding of π into τ.

Using the grid-preserving embedding, we define a satisfying assignment $\rho\colon [n] \to \{T, F\}$. We set $\rho(k) = T$ if the pair $X^1_{k,1}$ is mapped to $Y^1_{k,1}$ and we set $\rho(k) = F$ if it is mapped to $Z^1_{k,1}$. This property is clearly maintained throughout the multiplication and sorting phases due to the properties of the gadgets. Finally, we already argued that our construction of evaluation phase guarantees that all three literals in a given clause cannot be negative.