Representing Matroids over the Reals is $\exists \mathbb{R}$*-complete*

Eun Jung Kim[1]   Arnaud de Mesmay[2]   Tillmann Miltzow[3*]

[1] *KAIST, Daejeon, South Korea and CNRS, Paris, France*
[2] *Univ Gustave Eiffel, CNRS, LIGM, F-77454 Marne-la-Vallée, France*
[3] *Department of Information and Computing Sciences, Utrecht University, The Netherlands*

A *matroid* M is an ordered pair $(E, \mathcal{I})$, where E is a finite set called the *ground set* and a collection $\mathcal{I} \subset 2^E$ called the *independent sets* which satisfy the conditions: (i) $\emptyset \in \mathcal{I}$, (ii) $I' \subset I \in \mathcal{I}$ implies $I' \in \mathcal{I}$, and (iii) $I_1, I_2 \in \mathcal{I}$ and $|I_1| < |I_2|$ implies that there is an $e \in I_2$ such that $I_1 \cup \{e\} \in \mathcal{I}$. The *rank* $\mathsf{rk}(M)$ of a matroid M is the maximum size of an independent set. We say that a matroid $M = (E, \mathcal{I})$ is *representable* over the reals if there is a map $\varphi \colon E \to \mathbb{R}^{\mathsf{rk}(M)}$ such that $I \in \mathcal{I}$ if and only if $\varphi(I)$ forms a linearly independent set.

We study the problem of MATROID $\mathbb{R}$-REPRESENTABILITY over the reals. Given a matroid M, we ask whether there is a set of points in the Euclidean space representing M. We show that MATROID $\mathbb{R}$-REPRESENTABILITY is $\exists \mathbb{R}$-complete, already for matroids of rank 3. The complexity class $\exists \mathbb{R}$ can be defined as the family of algorithmic problems that is polynomial-time equivalent to determining if a multivariate polynomial with integer coefficients has a real root.

Our methods are similar to previous methods from the literature. Yet, the result itself was never pointed out and there is no proof readily available in the language of computer science.

Keywords: Computer Science - Computational Complexity, Mathematics - Combinatorics

1   Introduction

Many articles on matroids assume that the matroid is representable, see for example Lovász (1980); Cameron, Dinu, Michałek, and Seynnaeve (2022); Cameron and Mayhew (2021). Representability either heavily simplifies proofs and definitions or is even essential. We show that the question of representability over the reals is as difficult as the existential theory of the reals, that is $\exists \mathbb{R}$-complete. The complexity class $\exists \mathbb{R}$ can be defined as the family of algorithmic problems that is polynomial-time equivalent to determining if a multivariate polynomial (with integer coefficients) has a real root, see Section 2 for an introduction and overview of this complexity class.

Definitions. Before we give a general definition of a matroid, we introduce *vector matroids*. Given a matrix A over a field $\mathbb{F}$, we can define the corresponding vector matroid $M[A] = (E, \mathcal{I})$ as follows. The ground set E of $M[A]$ is formed by the columns of A and we say that a subset $I \subset E$ is independent in $M[A]$, i.e., $I \in \mathcal{I}$, if the columns are linearly independent over $\mathbb{F}$, that is, the zero vector can be linearly combined by the vectors in I only trivially. A set of elements of E which is not independent is said to be dependent. Note that any set of columns containing a zero column is dependent. The independent sets of a vector matroid satisfy three simple properties (see below). One way to look at matroids is to see them as abstract set systems that have those three properties. A *matroid* is an ordered pair $(E, \mathcal{I})$, where E is a finite set called the *ground set* and a collection $\mathcal{I} \subset 2^E$ called the *independent sets* which satisfy the conditions:

   (i) $\emptyset \in \mathcal{I}$,

   (ii) $I' \subset I \in \mathcal{I}$ implies $I' \in \mathcal{I}$,

   (iii) $I_1, I_2 \in \mathcal{I}$ and $|I_1| < |I_2|$ implies that there is an $e \in I_2$ such that $I_1 \cup \{e\} \in \mathcal{I}$.

The *rank* $\mathsf{rk}(M)$ of a matroid M is the maximum size of an independent set. We say that a matroid $M = (E, \mathcal{I})$ is *representable* over $\mathbb{F}$ if there is a matrix A over $\mathbb{F}$ such that $M = M[A]$. Note that all columns of A live in a subspace of dimension at most $\mathsf{rk}(M[A])$. Therefore, we can assume without loss of generality that the columns of A have dimension $\mathsf{rk}(M[A])$. We refer to Oxley (2006) for more background on matroids.

Given a matroid M, if there exists a matrix A over $\mathbb{R}$ such that $M = M[A]$, we say that M is representable over the reals. The algorithmic problem of MATROID $\mathbb{R}$-REPRESENTABILITY is to test whether a given matroid is representable over the reals. Since we discuss only the real case in this article, we will sometimes say representable as a shorthand. Note that we also need to specify how the matroid M is given. In the literature on matroids, one has often an oracle such that one can ask the oracle for each set I whether $I \in \mathcal{I}$. We will deviate from this practice, as it might be unclear how to describe the oracle. Instead, we will just list all sets in $\mathcal{I}$ explicitly. This does not blow up the description complexity too much in our case, as we will deal mainly with constant rank matroids.

We refer to Section 2 for more background, motivation and application of matroids.

Geometric Interpretation. If we have a representation $A \in \mathbb{R}^{3 \times n}$ of a matroid $M = M[A]$, we can scale an arbitrary column of A by a nonzero number and it stays a valid representation. This is also the case if we scale by -1. Furthermore, if we rotate A (i.e., multiply it on the left with an orthogonal transformation) it still stays a valid representation. Thus, in case we have a representable rank-3 matroid over $\mathbb{R}$, we can assume that there is a representation in which all nonzero vectors have their third coordinate equal to 1. In this way, we can consider the columns of A as a point configuration in the plane with $z = 1$. The property of three vectors being dependent is then equivalent to the corresponding three points lying on a common line.
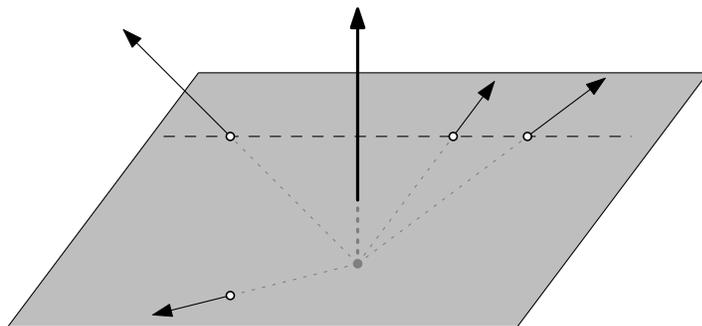
In this geometric interpretation, we could have considered any plane different from the one with $z = 1$, as long as it does not contain the origin. This would have yielded a different point configuration. The resulting transformation is called a *projective transformation*. It maps lines to lines, except for one line that disappears. We say that this line is *sent to infinity*. Conversely, for any line in the plane, there exists a projective transformation that sends it to infinity. Throughout this article, we think of representations of rank-3 matroids via these point configurations, and thus we will slightly abuse language by calling such a point configuration a representation.

A motivating example. One of the standard examples to illustrate realizability is the so-called Fano plane. It is the matroid on seven elements whose maximal independent sets are all the triples except

$$\{1, 4, 7\}, \{1, 2, 3\}, \{1, 5, 6\}, \{3, 6, 7\}, \{2, 5, 7\}, \{3, 4, 5\}, \{2, 4, 6\}.$$

This can be represented pictorially as in the figure below, where the lines and the circle denote the dependencies.
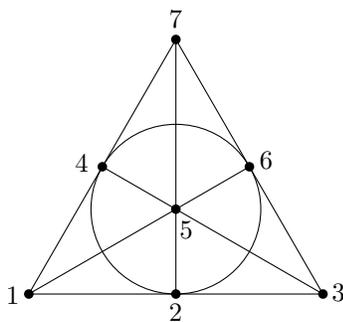
An immediate question that arises from this picture is whether a picture exists where the circle is not used and the dependencies are all pictured by lines. This is equivalent to asking whether the Fano matroid is representable over the reals. It is well-known not to be (Oxley, 2006, Proposition 6.4.8). The problem MATROID ℝ-REPRESENTABILITY addresses the general question of deciding whether a given matroid can be represented like that, and our main result is that this problem is ∃ℝ-complete.

Order Types. We saw above that matroids are an abstraction to describe point collinearities in the plane. I.e., if we have a rank-3 matroid then every dependent set corresponds to three collinear points.

Now given a set of points, we are often also interested in the orientation of each triple: either clockwise, counter-clockwise, or collinear.
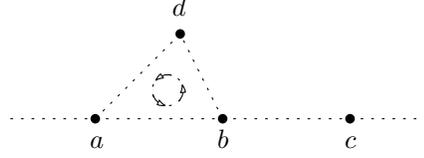
In this example, $\{a, b, c\}$ is collinear and (a, b, d), (a, c, d), and (b, c, d) are oriented counter-clockwise. This leads to the definition of *(abstract) order types*, which is a pair $O = (E, \chi)$. Again, E is a finite set called the ground set. Then

$$\chi : \binom{E}{3} \to \{-1, 0, 1\}$$

is a function called a *chirotope* satisfying a few simple properties that are derived from the intuition given above. We say that a point set $P \subset \mathbb{R}^2$ *represents* a given order type $O = (E, \chi)$ if P has, for each element $e \in E$, a corresponding point $e' \in P$. Furthermore, for each triple $a, b, c \in E$ the corresponding points $a', b', c' \in P$ are oriented according to $\chi(\{a, b, c\})$. Note that if we lift every point in P to the plane with $z = 1$ as a subset of $\mathbb{R}^3$, then we get the following correspondence between $(p'_x, p'_y, 1), (q'_x, q'_y, 1), (r'_x, r'_y, 1)$ and the elements $p, q, r \in E$:

$$\operatorname{sign} \det \begin{pmatrix} p'_x & q'_x & r'_x \\ p'_y & q'_y & r'_y \\ 1 & 1 & 1 \end{pmatrix} = \chi(p, q, r).$$

Note that in this specific setup a realization of a rank 3 matroid and order types are closely related. While a matroid determines only the collinearities, the order type also determines the orientation of each triple. We want to point out that every abstract order type can be represented by a pseudoline arrangement. A pseudoline arrangement can be defined as a collection of x-monotone curves such that any pair of curves intersects exactly once. The orientation of a triple of pseudolines is defined by the orientation of the triangle that they form (a degenerate triangle corresponding to a zero orientation).
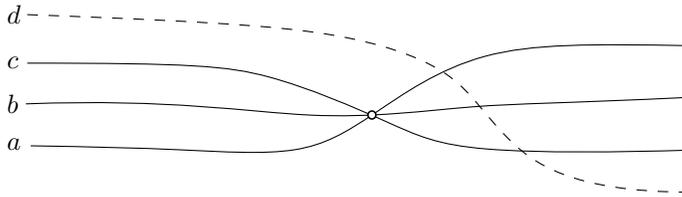
Note that this pseudoline arrangement corresponds to the order type example given above: $\chi(abc) = 0$, $\chi(abd) = 1$, $\chi(bcd) = 1$ and $\chi(acd) = 1$. Now, the realizability of the order types is equivalent to the STRETCHABILITY of pseudoline arrangements, that is, finding a line arrangement with the same combinatorics as the pseudoline arrangement. It is one of the central theorems in the field of the existential theory of the reals that STRETCHABILITY is $\exists \mathbb{R}$-complete. We will use many ideas of that proof for our main result.
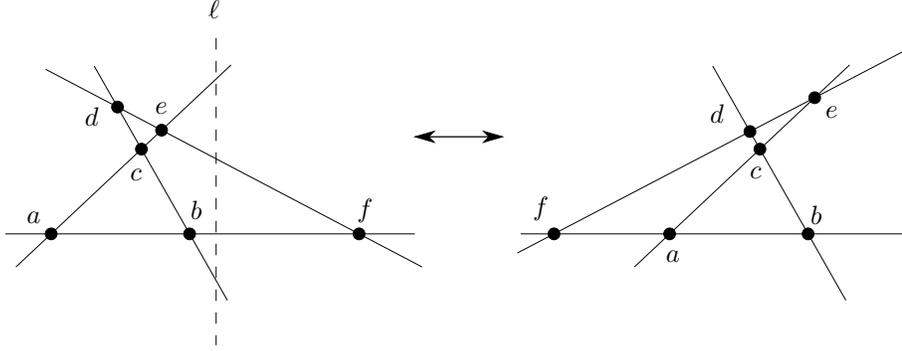
Fig. 1: As ℓ separates f from the other points, a projective transformation sending the line ℓ to infinity will flip the orientation of the triangles involving f while keeping the other orientations unchanged.

We also want to point out that the notion of an order type can be easily generalized to dimension d. The chirotope becomes a function of all $(d+1)$ tuples and tells us the orientation in d dimensions. To illustrate this, if we have four points a, b, c, e in 3-space, then the points a, b, c lie on a hyperplane H. Then the chirotope tells us on which side of H the point e lies, for an orientation of H defined by the three points a, b and c.

It is important to note that if we take a projective transformation of the plane, we preserve the represented matroid. This is because lines are mapped to lines and points to points. However, projective transformations do not preserve the order type of a point set, since a point may end up on the other side of some line, as depicted in Figure 1. In order to get a closer relationship, we work (sometimes) with matroids endowed with a distinguished line *at infinity* ℓ_∞. Then we consider *valid* representations of such matroids, which are those where this line is at infinity (i.e., all the points lie on one side of it). This definition extends to rank-k matroids using a hyperplane at infinity and leads to the following definition.

Given an order type O, we say that a matroid M *simulates* O if the underlying matroid of O is a subset of M and if the following conditions are met:

- Any representation of the matroid underlying O extends to a representation of M.

- Any valid representation of M induces a point set representing O.

Note that when M simulates O, then M has a representation if and only if O has an oriented representation: indeed, starting with a representation of M, one can always send the line at infinity to infinity using a projective transformation and thus obtain a valid representation..

Our results

Our main theorem is that MATROID ℝ-REPRESENTABILITY is complete for the existential theory of the reals.

Theorem 1. MATROID ℝ-REPRESENTABILITY *is* ∃ℝ*-complete.*

We provide two proofs of Theorem 1. The first one relies on simulating arbitrary ETR-formulas using addition and multiplication and some technical assumptions (see the overview in Section 2). The second

proof is somewhat easier, starting from the fact that order type realizability is $\exists\mathbb{R}$-complete, and then simulating order types using normal matroids.

Theorem 2. *Let $k \geq 3$ be a fixed integer. Given a rank-k order type O, we can compute in linear time a rank-k matroid M such that M simulates O.*

Theorem 1 easily follows from Theorem 2 as deciding whether an order type is representable over the reals is $\exists\mathbb{R}$-complete. This follows from techniques dating back to the proof of the Mnëv Universality Theorem (Matoušek, 2014; Schaefer, 2010). However, as explained in Matoušek (2014), the proof that STRETCHABILITY is $\exists\mathbb{R}$-hard requires a significant number of intricate steps, some of which can be simplified in the setting of MATROID $\mathbb{R}$-REPRESENTABILITY. Indeed, the need for different scales (see for example (Matoušek, 2014, Proof of Theorem 4.6)), which is the main difficulty in the oriented case, can be completely circumvented in our case. Therefore, for the sake of completeness and simplicity, we also provide a self-contained proof of Theorem 1. We think that it might be educational to first understand the proof of Theorem 1, before one tries to understand the $\exists\mathbb{R}$-completeness of STRETCHABILITY.

2   Proof Overview and Background

Proof Overview

In order to give an idea of the direct proof of Theorem 1, we first sketch an incorrect proof. Then we point out the issues with this sketch and how we can fix them.

It is folklore that in order to prove $\exists\mathbb{R}$-completeness, it is sufficient to find a way to encode variables and some basic operations like addition ($x + y = z$) and multiplication ($xy = z$). We can force points to lie on a specific line ℓ to represent our variables. Furthermore, using the well-known *von Staudt constructions*, we can simulate all the basic constraints, see Figure 2 for the construction to simulate addition. This (almost) describes a rank-3 matroid M. Furthermore, in any realization of M, we can read a valid variable assignment.
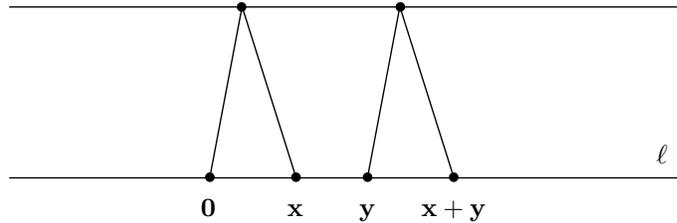
Fig. 2: Encoding addition geometrically.

The issue with this basic approach is that it could be that there is only one realization of M such that two points coincide, or three points lie on a common line, accidentally. If we were able to anticipate this, we could easily specify this in the description of the matroid, but in general, this is not easy.

We circumvent this general position issue with two fixes. The first fix is to reduce from a version of ETR where we can assume that all variable values are distinct. We call this variant DISTINCT-ETR, see Section 3. The second fix is to observe that when we build the von Staudt construction, we can ensure that all helper points have enough *freedom* to avoid any coincidences or collinearities with previously defined

points, see Section 4. With these two fixes, the above proof sketch works as is explained in the full proof of Theorem 1.

The proof idea of Theorem 2 goes as follows. Using arithmetic operations, we can give a variable the value $y = x^2 \geq 0$. Geometrically, this implies that the point representing y is on the same side as the point representing 1 within the common line ℓ (with respect to the point respresenting 0). In other words, we can enforce two points to lie on the same side of a line with respect to a given point. We lift this to half-spaces in the plane and higher dimensions. In this way, we can enforce consistent orientations of the matroid with the given order type.

Results on DISTINCT-ETR. We define the problem DISTINCT-ETR as a variant of ETR as follows (see later for a proper definition of ETR). We are given variables $X = \{x_1, \ldots, x_n\}$ and constraints of the form

$$x + y = z, \quad x \cdot y = z, \quad x = 1, \quad x > 0,$$

for $x, y, z \in X$. Furthermore, we are promised that there is either no solution at all or there is a solution $(x_1, \ldots, x_n) \in \mathbb{R}^n$ such that $x_i \neq x_j$ for all $i \neq j$. We show the following theorem in Section 3, which might be of independent interest.

Theorem 3. DISTINCT-ETR *is ∃ℝ-complete.*

While we could not find any prior proof of Theorem 1 in the literature (hence this work), there are many related works from at least two perspectives. First, as already mentioned, when one considers order-types instead of unoriented matroids, Theorem 1 is very well-known. Second, topological universality theorems have been proved for the real-representability of matroids in the algebraic geometry literature, see for example Lafforgue (2003) and Lee and Vakil (2013). While our work uses tools that are similar in spirit to those papers, it differs in that our constructions are arguably simpler and that we specifically focus on proving the computational hardness result, which is not the point of focus of those previous works, and is not entirely equivalent (see discussion below). Furthermore, we believe that it is worthwhile to have a complete proof of Theorem 1 in a purely combinatorial language, as opposed to previous works which employ the language of schemes.

Shor's proof of Mnëv's universality theorem (Shor, 1991, Section 4) introduces an intermediate problem called *the existential theory of totally real ordered variables*, which is very similar to DISTINCT-ETR but features totally ordered variables $x_1 < x_2 \ldots < x_n$ as opposed to just requiring distinctness as in our problem. This ordering is desirable when one investigates oriented matroids, and unneeded for unoriented ones. This difference allows for a proof that is arguably simpler than his, or at least different.

Background and Related Work

Matroids and Greedy A practical reason why matroids are relevant for computational purposes is that they capture in a simple way the class of discrete objects where greedy algorithms are successful in finding an optimal solution. For example, the standard Kruskal and Prim algorithms to compute a Minimum Spanning Tree in a weighted graph can be abstracted by considering the vector matroid defined by an oriented incidence matrix of the graph (called a *graphic matroid*), and then generalized to compute in polynomial time a maximum or minimum-weight basis for any matroid. This property actually characterizes matroids, see for example (Oxley, 2006, Section 1.8)

Some applications of representability.   For an algorithm on matroids, a suitable encoding scheme of the input matroid is needed. A common way is to take the input matroid $M = (E, \mathcal{I})$ in the form of an independence oracle which answers whether a given subset of the ground elements E is independent or not. There are algorithms which run with a polynomial number of queries to such an oracle. For example, a maximum weight independent set of a given matroid can be computed in this way. However, for many natural matroid properties, it is known that there is no algorithm with polynomially bounded queries to an independence oracle (Jensen and Korte, 1982) including representability over the finite field with two elements $GF(2)$ and connectivity of a matroid.

A vector representation of a matroid offers a compelling alternative to an independence oracle as matroid operations can be substantially more efficient using matrix operations. The Matroid parity problem, a common generalization of graph matching and matroid intersection, is solvable in polynomial time given a vector representation (Lovász, 1980) while a super-polynomial number of calls is needed under the independence oracle model (Jensen and Korte, 1982). Deciding whether the branch-width of a matroid is at most k is a common generalization of computing the branch-width, rank-width and carving-width of a graph. While there is an algorithm with $n^{O(k)}$ queries on an n element matroid for this problem (Oum and Seymour, 2007) under the oracle model, it remains unknown whether the dependency on k in the exponent can be replaced by a uniform constant. In contrast, the branch-width of a vector matroid can be computed in $f(k) \cdot n^3$ time (Jeong, Kim, and Oum, 2021) when the given representation is over a finite field $\mathbb{F}$.

Another powerful application of a vector representation can be found in the theory of kernelization in parameterized complexity. A surprising discovery of Kratsch and Wahlströhm (Kratsch and Wahlström, 2020) is that for many graph cut problems, compressing the input boils down to finding a so-called representative set of a matroid. When the said matroid is a vector matroid, a representative set of bounded size can be efficiently computed in polynomial time (László, 1977; Marx, 2009). It turns out that solutions to graph cut problems can be encoded as independent sets in *gammoids*, which form a well-known class of representable matroids and of which a vector representation can be constructed in randomized polynomial time.

Oriented Matroids.   One might wonder why we jumped from realizability of matroids to realizability of abstract order types, instead of using the perhaps closer notion of oriented matroids (Björner, Las Vergnas, Sturmfels, White, and Ziegler, 1999), for which one can also define a realizability problem and investigate its complexity. The reason is that in our arguments, we reason extensively with point configurations, and the geometric interpretation described above does not adapt directly to oriented matroids, as scaling a column by a negative number could lead to a change of the underlying oriented matroid. The correct framework to connect oriented matroids to point configurations is to only consider *acyclic* oriented matroids (Björner, Las Vergnas, Sturmfels, White, and Ziegler, 1999, Section 1.2.b), that is, those for which the geometric interpretation works readily without a need for rescaling by a negative number. This notion of acyclic, oriented matroids coincides with the notion of abstract order types, and so do their realizability problems. Note that in some of the existing literature, oriented matroids and abstract order types are sometimes described as equivalent. Therefore, we stress this subtle difference here.

The existential theory of the reals.   The complexity class $\exists \mathbb{R}$ (pronounced as 'ER', 'exists R', or 'ETR') has gained a lot of interest in recent years. See the compendium by Cardinal, Miltzow, and Schaefer (2024) for a comprehensive overview. The complexity class is defined via its canonical complete problem ETR (short for *Existential Theory of the Reals*. ETR refers to a geometric problem and $\exists \mathbb{R}$ refers to the

complexity class. While there are several different variants of ETR, there is only one complexity class.) and contains all problems that polynomial-time many-to-one reduce to it. In an ETR instance, we are given a sentence of the form

$$\exists x_1, \ldots, x_n \in \mathbb{R} : \varphi(x_1, \ldots, x_n),$$

where φ is a well-formed and quantifier-free formula consisting of polynomial equations and inequalities in the variables and the logical connectives $\{\wedge, \vee, \neg\}$. The goal is to decide whether this sentence is true. As an example consider the formula $\varphi(X, Y) :\equiv X^2 + Y^2 \leq 1 \wedge Y^2 \geq 2X^2 - 1$; among (infinitely many) other solutions, $\varphi(0, 0)$ evaluates to true, witnessing that this is a yes-instance of ETR. We use $|\varphi|$ to denote the *length* of φ, that is, the number of bits necessary to write down φ. The solution set of an ETR-formula is called a semi-algebraic set. The *(bit)-complexity* of a semi-algebraic set is the shortest length of any formula defining the set. It is known that

$$\text{NP} \subseteq \exists\mathbb{R} \subseteq \text{PSPACE}.$$

Here the first inclusion follows because a SAT instance can trivially be written as an equivalent ETR instance. The second inclusion is highly non-trivial and was first proven by Canny in his seminal paper (Canny, 1988).

Note that the complexity of working with continuous numbers was studied in various contexts. To avoid confusion, let us make some remarks on the underlying machine model. The underlying machine model for $\exists\mathbb{R}$ (over which sentences need to be decided and where reductions are performed) is the word RAM (or equivalently, a Turing machine) and not the real RAM (Erickson, van der Hoog, and Miltzow, 2020) or the Blum-Shub-Smale model (Blum, Shub, and Smale, 1989).

The complexity class $\exists\mathbb{R}$ gains its importance by numerous important algorithmic problems that have been shown to be complete for this class in recent years. The name $\exists\mathbb{R}$ was introduced in Schaefer (2010) who also pointed out that several NP-hardness reductions from the literature actually implied $\exists\mathbb{R}$-hardness. For this reason, several important $\exists\mathbb{R}$-completeness results had been obtained before the need for a dedicated complexity class became apparent.

Common features of $\exists\mathbb{R}$-complete problems are their continuous solution space and the nonlinear relations between their variables. Important $\exists\mathbb{R}$-completeness results include the realizability of abstract order types (Mnëv, 1988; Shor, 1991) and geometric linkages (Schaefer, 2013), as well as the recognition of geometric segment (Kratochvíl and Matoušek, 1994; Matoušek, 2014), unit-disk (Kang and Müller, 2012; McDiarmid and Müller, 2013), and ray intersection graphs (Cardinal, Felsner, Miltzow, Tompkins, and Vogtenhuber, 2018). More results appeared in the graph drawing community (Dobbins, Kleist, Miltzow, and Rzążewski, 2023; Erickson, 2019; Lubiw, Miltzow, and Mondal, 2022; Schaefer, 2021), regarding the Hausdorff distance (Jungeblut, Kleist, and Miltzow, 2022), regarding polytopes (Dobbins, Holmsen, and Miltzow, 2019; Richter-Gebert and Ziegler, 1995), the study of Nash-equilibria (Berthelsen and Hansen, 2019; Bilò and Mavronicolas, 2016, 2017; Garg, Mehta, Vazirani, and Yazdanbod, 2018; Schaefer and Štefankovič, 2017), training neural networks (Abrahamsen, Kleist, and Miltzow, 2021; Bertschinger, Hertrich, Jungeblut, Miltzow, and Weber, 2022), matrix factorization (Chistikov, Kiefer, Marusic, Shirmohammadi, and Worrell, 2016; Schaefer and Štefankovič, 2018; Shitov, 2016, 2017; Tuncel, Vavasis, and Xu, 2022), or continuous constraint satisfaction problems (Miltzow and Schmiermann, 2022). In computational geometry, we would like to mention geometric packing (Abrahamsen, Miltzow, and Seiferth, 2020), the art gallery problem (Abrahamsen, Adamaszek, and Miltzow, 2018), and covering polygons with convex polygons (Abrahamsen, 2022).

Recall that NP is usually described using a witness and a verification algorithm. The same characterization exists for $\exists \mathbb{R}$. Instead of the witness consisting of binary words of polynomial length, we allow in addition using real-valued numbers as a witness. Furthermore, in order to be able to use those real numbers, we are allowed to work on the so-called real RAM model of computation. The real RAM allows arithmetic operations with real numbers in constant time (Erickson, van der Hoog, and Miltzow, 2020).

Topological Universality. Many results and techniques on the existential theory of the reals actually precede the study of this complexity class. The underlying idea was to study how complicated solution spaces can be from a topological perspective. For example, if we want to study convex polytopes, we are often interested in the properties of their face lattice, which is the family of faces of different dimension together with their inclusion order The face lattice is a purely combinatorial object. Therefore, it is natural to ask which face lattices are realizable by polytopes. If there existed an easy combinatorial description of realizable face lattices, convex polytopes would be much better understood. Given a specific face lattice L, we can study its suitably defined solution space $S(L)$. As the realizability question can be formulated as an ETR-formula, it follows that $S(L)$ is a semi-algebraic set. Now, let T be a different semi-algebraic set, we wonder whether there exists a face lattice L such that $S(L)$ is homotopy-equivalent to T. Maybe surprisingly topological universality states that there is such an L for any semi-algebraic set T. This type of property feels very strong, as it intuitively states that we can encode the vast complexity of semi-algebraic sets into the problem of realizing convex polytopes. Indeed, many of the results that establish such topological universality also imply $\exists \mathbb{R}$-completeness (Matoušek, 2014).

However, topological universality can also be established for NP-complete problems as has been shown by Bertschinger, El Maalouly, Miltzow, Schnider, and Weber (2023). As they showed, for simplicial complexes, it is sufficient to encode the topology as it is possible to triangulate semi-algebraic sets (Hironaka, 1975). In other words, the difference between the *wild* semi-algebraic sets and the *tame* simplicial complexes is not that the former emit a more complicated topological structure. The difference comes from the ability of semi-algebraic sets to encode complicated topological spaces in a much more *concise* manner. (To be precise the description complexity of a topological space might be exponentially smaller using the language of semi-algebraic sets, compared to simplicial complexes.) $\exists \mathbb{R}$-completeness can be interpreted as giving a concise encoding of semi-algebraic sets into a different domain. To make our life easier, we do not care about preserving the complete topology, but merely the property of being empty or not. Still, in order to do so one usually also preserves topological properties. This is the reason why there is a close connection between topological universality and $\exists \mathbb{R}$-completeness. But given that NP-complete problems may also admit universality theorems $\exists \mathbb{R}$-completeness may arguably be considered the more interesting finding.

Stronger Universality Results. We want to point out that previous universality results often showed stronger results than mere topological universality. For instance, Richter-Gebert and Ziegler (1995) showed such a stronger universality theorem for polytopes. Given a face lattice F, we can define the set of polytopes $V(F)$ having face lattice F. Richter-Gebert showed that for every semi-algebraic set S there exists a face-lattice F of a polytope such that $V(F)$ is stably-equivalent to S. To define the notion of stable-equivalence goes above the scope of this paper. We just note that stable-equivalence encapsulates more than just the topology of S, but also to a degree the "geometry" of S.

Discussion on Matroid Input Encodings

In this paper, we study the $\mathbb{R}$-realizability, where the input matroid is given with all base, that is, maximal independent sets. This would seem unconventional at first glimpse, especially for those familiar with matroid theory. We would like to address the subtleties around our problem setting.

Types of encodings. For matroids, three possible descriptions are examined in the literature, namely an explicit description of sets, a description via an oracle, and a succinct description with a matrix.

Recall that an input graph for a graph problem can be given as an adjacency matrix, or equivalently the family of vertex subsets of size two. Similarly, an input hypergraph is typically given as a set family with an explicit description of all hyperedges. An immediate analogue of such an hypergraph description for a matroid is an explicit enumeration of all bases (maximal independent sets) or all circuits (minimally dependent sets). However, explicitly stating all independent sets, bases, or circuits is unconventional; the most common size measure of a matroid is the number of elements, which is polynomially bounded by the size of a graph or matrix that is generalized by a matroid. On the other hand, the number of bases or circuits can be prohibitively large in comparison to the number of elements. Specifically, it is known that the number of distinct matroids on n elements is doubly exponential in n (Knuth, 1974), hence in space of size polynomial in n one cannot describe an arbitrary input matroid. For further details about explicit matroid encoding, see Mayhew (2008).

When the matroids under consideration are representable over a field $\mathbb{F}$, a matrix over $\mathbb{F}$ provides a succinct description of a matroid. There are well-studied matroid classes that are representable such as uniform matroids, graphic matroids, and transversal matroids. However, not all matroids are representable. Hence, an important question is to decide whether an input matroid is representable over a specific field, or over any field at all, and to find a representation if one exists.

Due to the limitations of the above two explicit descriptions, the most common way to encode an input matroid without any restriction is with an oracle, often an independence oracle. One can view an independence oracle as a black box expressing a boolean function on n variables. The boolean function is on n input variables and outputs 0 or 1 depending on whether the input corresponds to (a characteristic vector of) an independent set of the said matroid. Problems with black box functions, e.g., an implicit input with oracle access are studied in the context of learning a function with a small number of queries, e.g. Polynomial Identity Testing, and also in the context of search problems where a graph is accessed by adjacency query. Such a problem does not fit in the classic computational complexity, where an explicit string of numbers is expected as an input. Moreover, learning a black box (boolean) function cannot be done efficiently. As mentioned previously, even when the boolean input functions are restricted to be matroid oracles, deciding whether a nontrivial matroid property holds or not requires $2^{\Theta(n)}$ queries (Truemper, 1982) even for basic properties such as connectivity and representability over $\mathbb{F} = GF(2)$, and even with a randomized algorithm.

Therefore, an oracle encoding of the $\mathbb{F}$-representability problem does not appear to be a fruitful setting to better understand the algorithmic aspects of the problem. Moreover, we shall argue below that, with an explicit matroid description, there is an intriguing difference in the computational complexity of $\mathbb{F}$-representability between the cases when $\mathbb{F}$ is finite and when $\mathbb{F} = \mathbb{R}$.

$\mathbb{F}$-representability with explicit bases description. Let us consider a matroid description that provides a matroid M as a pair $(E, \mathcal{B})$, where all bases of M are stated in the collection $\mathcal{B}$. In this setting, we shall argue that the problem of deciding whether M is $\mathbb{F}$-representable is in NP for a finite field $\mathbb{F}$ and in $\exists\mathbb{R}$ for

$\mathbb{F} = \mathbb{R}$. We shall also argue that $\mathbb{F}$-representability is likely to be in co-NP for a finite field $\mathbb{F}$. This makes an interesting contrast with the case $\mathbb{F} = \mathbb{R}$, for which the corresponding non-representability problem is unlikely to be in ETR due to our main result.

First, let us see that $\mathbb{F}$-representability is in NP for finite $\mathbb{F}$. Indeed, a matrix A (whose columns are labeled by the elements of E) over $\mathbb{F}$ with $M[A] = M$ can be taken as a witness for $\mathbb{F}$-representability of M. Moreover, one can conceive a polynomial-time verification algorithm for the pair $M = (E, \mathcal{B})$ and A as follows. Let $\mathcal{B}(A)$ be the set of bases of $M[A]$ and recall that $M = M[A]$ if and only if $\mathcal{B} = \mathcal{B}(A)$. Whether $\mathcal{B} \subseteq \mathcal{B}(A)$ can be easily verified in time polynomial in $|\mathcal{B}| + \mathsf{rk}(M)$. For this, we first compute the column rank of A. If it is different from $\mathsf{rk}(M)$, this trivially implies $M[A] \neq M$. Henceforth, let us assume that the column rank of A equals $\mathsf{rk}(M)$. Now, for each base $B \in \mathcal{B}$, one checks whether the submatrix $A[B]$, the submatrix of A consisting of all columns labeled by the elements of B, is full rank. If any $B \in \mathcal{B}$ fails the test, we know that A is not a representation of M.

Therefore, we may assume that $\mathcal{B} \subseteq \mathcal{B}(A)$. To verify whether equality holds, we rely on the following Lemma.

Lemma 4. *Let $\mathcal{B}$ be the set of all bases of a matroid M and let $\mathcal{B}' \subsetneq \mathcal{B}$. Then there exist bases $B' \in \mathcal{B}'$ and $B \in \mathcal{B}' \setminus \mathcal{B}$ with $|B' \triangle B| = 2$.*

Proof: Choose $B' \in \mathcal{B}'$ and $B \in \mathcal{B} \setminus \mathcal{B}'$ so that $|B' \cap B|$ is maximized. Let $x \in B' \setminus B$. Recall the Basis Exchange Property:

> For any distinct bases W', W of a matroid and an element $x \in W' \setminus W$, there exists an element $y \in W \setminus W'$ such that $W' - x + y$ is a basis of the matroid.

By the Basis Exchange Property, there exists $y \in B \setminus B'$ such that $B'' := B' - x + y$ is a basis of M. If B'' belongs to $\mathcal{B}'$, we have $B'' \cap B = (B' \cap B) + y$, which contradicts the choice of B' and B. Therefore, B'' belongs to $\mathcal{B} \setminus \mathcal{B}'$. Note that $|B' \cap B''| = |B' - x| = rk(M) - 1$. Thus $B'' = B$ by the choice of B' and B and the claim follows. $\square$

Hence, the last step of the verification algorithm tests if there exist a basis $B \in \mathcal{B}$ and two elements $x \in B$ and $y \in E - B$ such that $B - x + y$ is not a basis of M but the corresponding set of columns of A is independent, which precisely tests if $B - x + y \in \mathcal{B}(A) \setminus \mathcal{B}$. If such a triple B, x, y exists, clearly $\mathcal{B} \subsetneq \mathcal{B}(A)$ and thus $M[A] \neq M$. Conversely if $\mathcal{B} \subsetneq \mathcal{B}(A)$, there exists such a triple B, x, y by Lemma 4. Therefore, we examine all triples (B, x, y) with $B \in \mathcal{B}$, $x \in B$ and $y \in E - B$ and certify that $B - x + y$ is either in $\mathcal{B}$ or dependent, in which case the verification algorithm can correctly conclude that $\mathcal{B}(A) = \mathcal{B}$, and thus $M[A] = M$. Otherwise, the verification algorithm concludes $M[A] \neq M$ and rejects the witness A.

The presented verification algorithm shows that $\mathbb{F}$-representability is in NP for each finite field $\mathbb{F}$. A matrix over $\mathbb{F}$ as witness and the polynomial-time verification algorithm naturally extend to the case when $\mathbb{F} = \mathbb{R}$, where the verification algorithm works on a real RAM. Therefore, $\mathbb{R}$-representability is in $\exists \mathbb{R}$. For details on the characterization of $\exists \mathbb{R}$ via a witness and a polynomial-time verification algorithm on a real RAM, see Erickson, van der Hoog, and Miltzow (2020), and also the discussion in the paragraph above about the existential theory of the reals.

Furthermore, it is likely that $\mathbb{F}$-representability is in co-NP for each finite field $\mathbb{F}$. It is known (Geelen and Whittle, 2016) that for any prime field $\mathbb{F}$, non-$\mathbb{F}$-representability can be certified by evaluating the ranks of $O(n^2)$ subsets of an n-element matroid. Notice that when the matroid $M = (E, \mathcal{B})$ is given

with an explicit description of all the bases $\mathcal{B}$ of M, evaluating $\mathsf{rk}(X)$ for $X \subseteq E$ can be done in time polynomial in the input size because $\mathsf{rk}(X)$ equals the maximum of $|X \cap B|$ over all $B \in \mathcal{B}$. Therefore, $\mathbb{F}$-representability is in co-NP under the explicit bases description for each prime field $\mathbb{F}$. For an arbitrary finite field $\mathbb{F}$, not necessarily prime, up to our best knowledge there is no published result which establishes that a polynomial number of rank evaluations suffices for non-$\mathbb{F}$-representability. However, it is known that a positive resolution of Rota's conjecture implies that only a constant, depending on $|\mathbb{F}|$ only, number of rank evaluations would suffice (Oxley, 2006) to certify that a given matroid is not $\mathbb{F}$-representable. The proof of Rota's conjecture was announced in 2014 by Geelen, Gerards, and Whittle (2014) although it is expected to take a few more years for the full proof to be written for publication.

Therefore, $\mathbb{F}$-representability appears to be in NP $\cap$ co-NP when the input is given as the exhaustive list of bases for each finite $\mathbb{F}$ given the claimed proof of Rota's conjecture. Given that, deciding the computational complexity of $\mathbb{F}$-representability for each finite $\mathbb{F}$ with explicit bases description is an intriguing question. For $\mathbb{F} = GF(2)$, a polynomial-time algorithm is straightforward from the uniqueness of a binary representation (up to linear transformation) and the fact that such a representation can be efficiently obtained (Oxley, 2006); after constructing a matrix over $GF(2)$, we apply the above verification algorithm for NP membership. However, even for $\mathbb{F} = GF(3)$ it is not clear whether a matrix over GF(3) can be efficiently constructed although it is known that there is a unique representation over $GF(3)$ for a matroid representable over $GF(3)$. As far as we are aware, there is no efficient procedure known for constructing the representation of a matroid M with a promise that M is representable over $GF(3)$, when M is given with an independence oracle or even given as a matrix over the rationals $\mathbb{Q}$ (Hlinený, 2006). Getting an input matroid as explicit bases description might help to circumvent this obstacle.

In contrast to the case of finite fields, it is impossible to certify non-$\mathbb{R}$-representability with a polynomial number of rank evaluations (Oxley, 2006). Finally, for $\mathbb{R}$-representability we showed $\exists\mathbb{R}$-completeness, exhibiting a noticeable diversion from $\mathbb{F}$-representability for finite $\mathbb{F}$ which appears neither NP-complete nor co-NP-complete with explicit bases description under the assumption NP $\neq$ co-NP. Therefore, our result highlights the recurring contrast between representability over a finite field and over the reals.

3 Distinct-ETR

This section serves as a preparation for the later reduction. Specifically, we show in Lemma 10 the $\exists\mathbb{R}$-completeness of the variant of ETR called DISTINCT-ETR that we defined in the introduction: in this variant, we are guaranteed that either there is no solution or there is a solution with all variables holding *distinct* values. This property will be key for encoding into matroids. Most of this section follows standard techniques.

Overview. This section is dedicated to the proof of the lemma. The idea is that we first establish the hardness of an ETR variant (STRICT-INEQ) with an open solution space. It is clear that all variables can be assumed there to have distinct values. Then, we reduce again to a variant where we use only the basic constraints.

The reduction goes in four steps.

1. From ETR to ETRAMI.

2. Then from ETRAMI to FEASIBILITY.

3. Then from FEASIBILITY to STRICT-INEQ.

4. And at last from STRICT-INEQ to DISTINCT-ETR.

Note that steps 1, 2, and 3 have already been done (among others) by Schaefer and Štefankovič (2017). We sketch the main steps of their reduction. Specifically, we point out some properties that were not explicitly emphasized. We start to explain a simple trick that is excessively used in those types of constructions in order to build small, very small, and very large numbers.

Number Constructions. Before we describe the reduction, we show how to construct variables that must have specific rational values.

If we want to build integers of polynomial size, we can do this by simply repeatedly adding a one.

$$a_1 = 1, \quad a_{i+1} = a_i + a_1.$$

It is easy to see that $a_i = i$, for all a_i that are defined in this way.

If we want to build a very large number, say 2^{2^k}, the previous approach cannot be done in a polynomial number of steps. Instead, we can use repeated squaring as follows.

$$x_0 = a_2 + a_0 (= 2), \quad x_{i+1} = x_i^2,$$

for $i = 1, \ldots, k$. It holds inductively that $x_i = 2^{2^i}$. Similarly, we can construct very small numbers say $2^{2^{-k}}$, as follows:

$$y_0 + y_0 = 1, \quad y_{i+1} = y_i^2,$$

for $i = 1, \ldots, k$. It holds inductively that $y_i = 2^{-2^i}$. Note that we can also use strict inequalities to build large and small numbers. For example,

$$x_0 > 2, \quad x_{i+1} > x_i^2,$$

for $i = 1, \ldots, k$ implies that $x_i > 2^{2^i}$.

We will use these standard tricks repeatedly later in the reduction.

Reduction from ETR **to** ETRAMI. We define the problem ETRAMI (see Abrahamsen and Miltzow (2019) for more background on this problem) as a variant of ETR as follows. We are given a set of variables $X = \{x_1, \ldots, x_n\}$ and constraints of the form

$$x + y = z, \quad x \cdot y = z, \quad x = 1,$$

for $x, y, z \in X$. Therefore, ETRAMI is a variant of ETR without negations, inequalities, disjunctions, conjunctions and the only constant is 1. It is folklore that ETRAMI is $\exists \mathbb{R}$-complete and follows implicitly from various papers, for example Matoušek (2014); Schaefer and Štefankovič (2017); Shor (1991).

Lemma 5 (folklore)**.** ETRAMI *is* $\exists \mathbb{R}$-complete.

$\exists \mathbb{R}$-membership follows from the definition. The idea of the reduction is to simplify an ETR-formula in each step. For instance, we can remove negations by replacing $\neg p > 0$ by $p \leq 0$, where p is a polynomial. In this way, we can remove all negations. We can replace inequalities by observing that $p \geq 0$ is equivalent to $\exists x: p = x^2$ and $p > 0$ is equivalent to $\exists x : px^2 = 1$. We can replace $p = 0 \wedge q = 0$ by $p^2 + q^2 = 0$. Similarly, we can replace $p = 0 \vee q = 0$ by $pq = 0$. Thereafter, we end with a single polynomial

equation $p = 0$. We construct variables for each coefficient value. Then we replace each coefficient with an appropriate variable. At last, we replace each occurrence of multiplication and addition inductively by introducing one more variable and one more constraint. We end with a single equation of the form $x = 0$, which can be replaced by $z = 1$ and $z + x = z$.

Reduction from ETRAMI **to** FEASIBILITY. In FEASIBILITY, we are given a single polynomial $p \in \mathbb{Z}[x_1, \ldots, x_n]$ of degree at most four. We are asked if there exists some $x \in \mathbb{R}^n$ such that $p(x) = 0$. Furthermore, we require each coefficient to be of absolute value at most $36n^3$. Below, we will show how to achieve this upper bound.

Lemma 6. FEASIBILITY *is $\exists\mathbb{R}$-complete.*

Again, $\exists\mathbb{R}$-membership follows from the fact that FEASIBILITY is a special case of ETR. To show hardness we sketch a reduction from ETRAMI that is already known (Matoušek, 2014; Schaefer and Štefankovič, 2017). Let $f_1 = 0, \ldots, f_m = 0$ be the equations of some ETRAMI instance φ. (For example $x + y = z$ becomes $x + y - z = 0$.) Let

$$p = f_1^2 + \ldots + f_m^2.$$

Clearly, φ is satisfiable if and only if p has a zero. As each f_i has a degree at most two, it holds that p has degree at most four. Since there are three types of equations, each involving at most three variables, there are only $3n^3$ possible distinct constraints in φ. Thus we have $m \leq 3n^3$. Note that each term f_i^2 gives rise to at most six monomials and each coefficient is at most two. (For example, $(x + y - z)^2 = x^2 + 2xy - 2xz + y^2 - 2yz + z^2$.) Therefore each coefficient has absolute value at most $12m = 36n^3$. Thus, we can rewrite p as a sum of monomials with bounded-sized coefficients, as claimed.

Reduction from FEASIBILITY **to** STRICT-INEQ. In a STRICT-INEQ instance, we are given a sentence of the form

$$\exists x_1, \ldots, x_n \in \mathbb{R} : \varphi(x_1, \ldots, x_n),$$

where φ is a well-formed and quantifier-free formula consisting of polynomial strict-inequalities in the variables and the logical connectives $\{\wedge, \vee\}$.

Note that the solution space $\{x \in \mathbb{R}^n : \varphi(x)\}$ is always open.

Lemma 7. STRICT-INEQ *is $\exists\mathbb{R}$-complete.*

Again membership follows from the fact that ETR is more general then STRICT-INEQ. To show $\exists\mathbb{R}$-hardness we reduce from FEASIBILITY. The idea of the reduction is to replace $p(x) = 0$ by $-\delta < p(x) < \delta$, for some sufficiently small δ. To this end, we employ two lemmas as formulated in Schaefer and Štefankovič (2017). Note that the actual proof comes from real algebraic geometry and can be found for instance in Basu and Roy (2010) and Jeronimo, Perrucci, and Tsigaridas (2013).

Lemma 8. *Every non-empty semi-algebraic set in $\mathbb{R}^n$ of complexity at most $L \geq 4$ contains a point of distance at most $2^{L^{8n}}$ from the origin.*

Lemma 9. *If two semi-algebraic sets in $\mathbb{R}^n$ each of complexity at most $L \geq 5n$ have positive distance (for example, if they are disjoint and compact), then that distance is at least $2^{-2^{L+5}}$.*

In this context, the distance between two sets A and B in $\mathbb{R}^n$ is defined as

$$d(A, B) = \inf_{a \in A, b \in B} \|a - b\|,$$

where that $\| \cdot \|$ denotes the Euclidean norm.

In order to apply Lemmas 8 and 9, we define the following three semi-algebraic sets. First, we define the solution set for p.

$$S = \{x \in \mathbb{R}^n : p(x) = 0\}.$$

Let $R = 2^{L^{8n}}$, where L is the bit-complexity of S. Note that L upper bounded by the length of p. Using Lemma 8, we know that S is empty if and only if $S \cap B(R)$ is empty. (We denote by $B(R)$ the ball of radius R around the origin.) This motivates us to define the sets

$$S_1' = \{(x, z) \in \mathbb{R}^{n+1} : p(x) = z \wedge \|x\|^2 \leq R^2\},$$

and

$$S_2' = \{(x, z) \in \mathbb{R}^{n+1} : z = 0 \wedge \|x\|^2 \leq R^2\}.$$

Note that $S_1' \cap S_2' = (S \cap B(R)) \times \{0\}$. Furthermore, S_1' and S_2' are compact and thus we can apply Lemma 9. Unfortunately, the description complexities of S_1' and S_2' are exponential if we write R out in binary. Therefore, we define S_1 and S_2 slightly differently. Namely, we add some extra variables, whose sole purpose is to encode R using repeated squaring. Let $\overline{L}$ be the maximum of the bit complexities of S_1 and S_2. Note that $\overline{L} = O(L + n \log L)$. Let δ be a lower bound on the distance between S_1 and S_2, which is provided by Lemma 9 applied to S_1 and S_2. It holds that $S = \emptyset$ is equivalent to $S \cap B(R) = \emptyset$. This in turn is equivalent to $S_1 \cap S_2 = \emptyset$. And this is equivalent to

$$p(x) \leq -\delta \text{ or } \delta \leq p(x),$$

for all $x \in B(R)$. In other words, we have

$$\exists x : \ -\delta < p(x) < \delta \text{ and } \|x\|^2 < R^2 \tag{1}$$

if and only if

$$\exists x : \ p(x) = 0$$

This obviously also works if we would use *any* smaller δ. Maybe not so obviously, this also works for any R between $2^{L^{8n}}$ and $2^{L^{8n+1}}$. The lower bound allows us to apply Lemma 8. The upper bound allows us to apply Lemma 9. Using repeated squaring, we create numbers $a > 2^{L^{8n}}$ and $b < 2^{L^{8n+1}}$. Furthermore, we add the inequalities $a < R < b$. Note that we can construct a number δ that is at most $2^{-2^{\overline{L}+5}}$. Our STRICT-INEQ instance φ consists of the three inequalities from Equation (1) and some extra variables and constraints to bound R and δ as described above.

Reduction from STRICT-INEQ to DISTINCT-ETR. We can now finally show that DISTINCT-ETR is $\exists \mathbb{R}$-complete.

Lemma 10. DISTINCT-ETR *is $\exists \mathbb{R}$-complete*

Proof: We are not actually reducing from STRICT-INEQ but from the instance φ described in Equation (1). Let δ, R be the given numbers, $x_1, \ldots, x_n$ the variables and p the polynomial as in the previous paragraph. Recall that p has a degree at most four and the coefficients are bounded integers. We will introduce new variables in order to construct δ, R, $\|x\|^2$, and $p(x)$. Then, we will argue about distinctness.

First, we construct variables holding the values of the integers $-36n^3, \ldots, 36n^3$. Those variables are meant to represent the coefficients of p. Recall that the values of those coefficients were in this range. It has been described above how to construct those integers. Furthermore, we add variables holding the values $R = 2^{L^{8n}}$ and $\delta = 2^{-2^{\overline{L}+5}}$. (As we reduce from Equation (1), we do not need to approximate the values δ and R, but can construct them directly, as DISTINCT-ETR allows us to use equations.) If in this process two variables hold the same value, we can detect this and remove one of the variables.

Now, we construct $p(x)$ and $\|x\|^2$. First, we construct all possible $\binom{n}{4}$ monomials of degree at most four. For example, $N = xyzw$ is constructed in three steps. $N_1 = xy$, $N_2 = N_1 z$, and $N = N_2 w$. Again, whenever two identical monomials appear, we are able to notice this and make an appropriate replacement. Let us denote

$$p(x) = \sum_{i=1}^{m} a_i M_i.$$

Here, a_i is the coefficient of the monomial M_i. We construct $P_k = \sum_{i=1}^{k} a_i M_i$ inductively as follows:

$$P_1 = a_1 M_1 \text{ and } P_k = P_{k-1} + T_k,$$

with $T_k = a_k M_k$.

We denote by $P = P_m$ the variable holding the value of $p(x)$. We construct $X = \|x\|^2 = x_1^2 + \ldots + x_n^2$ in the same way.

At last, we add the variables a, b, c, and the constraints

$$P + \delta = a, \quad b + P = \delta, \quad R_2 = R \cdot R, \quad c + X = R_2.$$

Now we can enforce the inequalities

$$-\delta < p(x) < \delta \text{ and } \|x\|^2 < R^2$$

by

$$a > 0, \quad b > 0, \quad c > 0.$$

To summarize, we started with the variables $x_1, \ldots, x_n$. We have created variables $C_1, \ldots, C_s$ that each holds a different integer/rational number. Furthermore, we constructed some variables $V_1, \ldots, V_t$ such that each V_i is a polynomial function $g_i(x)$. Note that all the g_i have a degree of at most four and small integer coefficients. If for two variables V_i and V_j, we have that $g_i = g_j$, we can detect this and remove one of them and replace each occurrence with the other one. This finishes the description of the reduction. We denote this instance ψ.

To show correctness, we observe that φ has a solution if and only if ψ has a solution as well. Indeed, all new variables and constraints only "build" the correct polynomials and the numbers δ and R. Thus it remains to show that φ has a solution if and only if ψ has a solution with all variables taking distinct values. The backward direction is trivial. Therefore, we assume that φ has at least one solution $x \in \mathbb{R}^n$. As the solution space of φ is open, there is an open ball B fully contained in the solution space. Clearly, the variables $C_1, \ldots, C_s$ have all fixed distinct values by construction. Every other variable V_i can be expressed as a polynomial function $g_i(x)$. As all the g_i are distinct, there must be some $x \in B$ such that $g_i(x) \neq g_j(x)$, for all $i \neq j$ and $g_i(x) \neq C_j$, for all i, j. Otherwise, two of the polynomials would be identical. This finishes the proof. □

4   Arithmetic Using Matroids

In this section, we describe how to encode addition and multiplication of real numbers using rank-3 matroids. We rely on the *von Staudt constructions*, which are very well-known, perhaps with the caveat that they are usually stated for oriented matroids. But as we shall see, no orientedness is actually required to make them work. We follow the presentation of Matoušek (2014).

The setup for both operations is as follows. We have a line ℓ containing three distinct distinguished points called $\mathbf{0}$, $\mathbf{1}$, and ∞, and a fixed second line ℓ_∞ crossing ℓ at ∞. Given the variable x, we denote the corresponding point representing it by $\mathbf{x}$ in boldface. In this way, we easily distinguish between a point and the corresponding variable. A point $\mathbf{x}$ on the line ℓ can be interpreted as a real number using cross-ratios: if we denote by $d(\mathbf{a}, \mathbf{b})$ the oriented distance between two points $\mathbf{a}$ and $\mathbf{b}$ on the line ℓ, then the quantity

$$(\mathbf{x}, \mathbf{1}; \mathbf{0}, \infty) := \frac{d(\mathbf{x}, \mathbf{0}) \cdot d(\mathbf{1}, \infty)}{d(\mathbf{x}, \infty) \cdot d(\mathbf{1}, \mathbf{0})}$$

is a real number invariant under projective transformations, which, by a slight abuse of notation, we simply denote by x. Note that if ∞ is progressively sent to infinity using a projective transformation and $d(\mathbf{0}, \mathbf{1})$ is scaled to one in this formula, x converges to $d(0, x)$. Thus this cross-ratio matches the geometric location of $\mathbf{x}$ on ℓ under some projective transformation.

Now, given two points $\mathbf{x}$ and $\mathbf{y}$ on ℓ, we describe geometric operations to compute points $x + y$ and $x \cdot y$ on ℓ representing their addition and their multiplication.
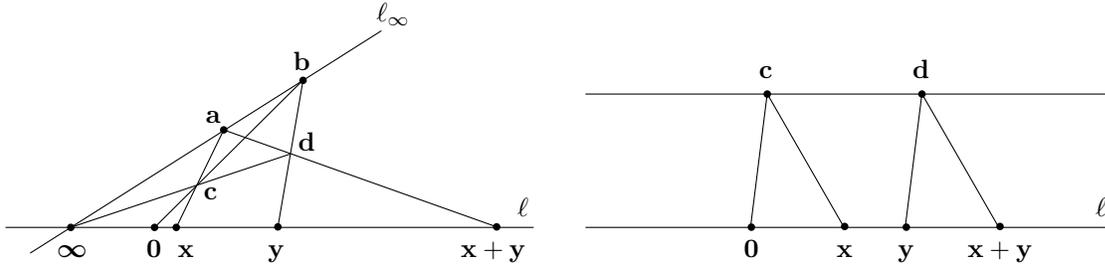
Fig. 3: Encoding addition geometrically.

Addition.   The construction is shown in Figure 3, left. We first introduce two distinct auxiliary helper points $\mathbf{a}$ and $\mathbf{b}$ situated anywhere on ℓ_∞. The line connecting $\mathbf{0}$ to $\mathbf{b}$ crosses the line connecting $\mathbf{x}$ to $\mathbf{a}$ in a point $\mathbf{c}$, then the line connecting ∞ to $\mathbf{c}$ crosses the line connecting $\mathbf{y}$ to $\mathbf{b}$ in a point $\mathbf{d}$. Similarly, the line connecting $\mathbf{a}$ to $\mathbf{d}$ crosses ℓ in a point that we define to be $x + y$. The rationale behind this construction is that by a projective transformation we can consider ℓ_∞ to be a line at infinity, leading us to Figure 3, right. Now the line containing $\mathbf{c}$ and $\mathbf{d}$ crosses ℓ at a point at infinity, i.e., these two lines are parallel. Likewise, the line containing $\mathbf{c}$ and $\mathbf{d}$ is parallel to ℓ. Then the parallelity of the lines $\mathbf{0cb}$ and $\mathbf{ydb}$ as well as $\mathbf{xca}$ and $(x+y)\mathbf{da}$ immediately shows that $d(\mathbf{0}, x+y) = d(\mathbf{0}, \mathbf{x}) + d(\mathbf{0}, \mathbf{y})$. In other words, the point $x + y$ has value $x + y$, justifying the notation.

Multiplication.   The construction is shown in Figure 4, left. As before, we first introduce two distinct auxiliary helper points $\mathbf{a}$ and $\mathbf{b}$ situated anywhere on ℓ_∞. The line connecting $\mathbf{1}$ and $\mathbf{b}$ crosses the line

connecting **x** and **a** at a point **c**, and the line connecting **0** and **c** crosses the line connecting **b** and **y** at a point **d**. Finally, the line connecting **a** and **d** crosses the line ℓ at a point that we define to be $\boldsymbol{xy}$. By sending the line ℓ_∞ at infinity using a projective transformation, we obtain Figure 4, right, where one can readily show using the parallel lines that $d(\mathbf{0}, \boldsymbol{xy}) = d(\mathbf{0}, \mathbf{x})d(\mathbf{0}, \mathbf{y})$. In other words, the point $\boldsymbol{xy}$ has value xy, justifying the notation.
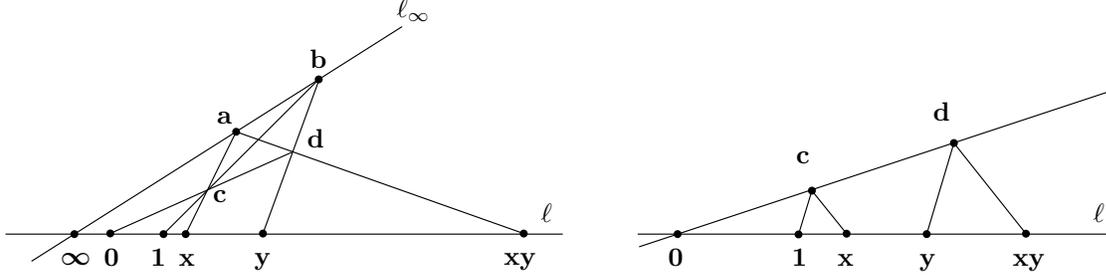
Fig. 4: Encoding multiplication geometrically.

Encoding these geometric constructions using matroids. The addition and multiplication constructions defined above can be entirely encoded using matroids: the independent sets are exactly the empty set, all the singletons, all the pairs of points, and all the triples of non-aligned points. We point out that no orientation was ever enforced during the constructions, and thus we do not need oriented matroids to describe them. Furthermore, these operations can be chained arbitrarily, allowing to encode polynomials using matroids. However, some very important care needs to be taken here: while it is clear from the construction which points should be aligned, we also need to make sure that points that should not be aligned can be assumed to not be aligned. For example, it could a priori happen that a line going through two helper points c_1 and c_2 somehow accidentally happens to pass through a variable x of the polynomial we are encoding. In that case, the matroid would not properly encode the geometric construction.

This motivates the following definition: we say that the set S of helper points used during an addition or multiplication construction is *free* if for any finite set of lines L and points P not involved in the construction, the points in S can be perturbed so that:

- the incidences required by the addition or multiplication construction still hold,

- no point of S lies on a line L, and

- no pair of points of S is aligned with a point of P.
  (In particular, no point of S coincides with a point of P.)

A key property of the addition and multiplication construction is that the four helper points that they rely on form a free set. Indeed, the points a and b can be placed freely on the line ℓ_∞, and thus can be perturbed so as to avoid the lines and points of L and P. Such a perturbation induces a perturbation of c and d in a two-dimensional open set, therefore allowing them to avoid lines of P and L, but also ensuring that no line going through a pair of points in $\{a, b, c, d\}$ also goes through a point in P.

This freedom will be leveraged in the proofs of Theorem 2 and Theorem 1, to ensure that the matroid correctly encodes orientation predicates and systems of polynomial equations, respectively.

Strict inequalities.   The multiplication construction can be leveraged to simulate a strict inequality constraint $x > 0$. Indeed, $x > 0$ if and only if there exists $z \in \mathbb{R}$ such that $z \neq 0$ and $x = z^2 = zz$, which can thus be simulated using a helper point z distinct from 0 and the above multiplication construction. However, this construction would require us to use as a helper point a fixed point z on the line ℓ, which could not be perturbed and thus would not be free. This can be resolved by using an additional variable: we first introduce a helper point y for which we ensure that $y > 0$ using the multiplication gadget. Then we use another multiplication gadget to ensure that $z > y$: note that this amounts to enforcing that z lies on the same side of y as 1 does, i.e., this can be tested by using another multiplication gadget where 0 is replaced by y. Now, in these two multiplication gadgets, neither y nor any of the other helper points is fixed, and therefore we can perturb them to avoid any fixed set of lines and points, showing that they form a free set of points.

5   Proof of Theorem 2

For convenience, we first restate Theorem 2

Theorem 2. *Let $k \geq 3$ be a fixed integer. Given a rank-k order type O, we can compute in linear time a rank-k matroid M such that M simulates O.*

Theorem 2 is proved by using the arithmetic constructions described in the previous section, in particular the one for strict inequalities, to simulate orientation predicates.

Simulating rank-3 order types.   We first consider the case of rank equal to 3 and treat the general case later. Let $O = (E, \chi)$ be an order type on n elements of rank 3. (We assume that $E = \{e_1, \ldots, e_n\}$.) We construct a matroid M from O inductively. To be more precise, we construct matroids, $M_3, \ldots, M_n = M$ such that M_i simulates O_i. Here, $O_i = (E_i, \chi_i)$ is the order type formed by the first i elements of O. All of our matroids M_i will feature a distinguished line ℓ_∞.

The matroid M_3 merely contains e_1, e_2, e_3. Without loss of generality we assume that the triple $t = \{e_1, e_2, e_3\}$ is independent in O. (Otherwise, all triples in O are dependent, which can trivially be simulated.) We first add a line at infinity, on which none of e_1, e_2 or e_3 lies. We can assume that t is oriented correctly in any representation of M_3, as otherwise, we can just reflect the representation and get a correct representation. Therefore, M_3 satisfies the induction hypothesis.

Now, let us assume that we have already constructed M_{i-1}. We construct M_i from M_{i-1}, by adding the element $e = e_i$ from O. Furthermore, for each triple $t = \{a, b, e\} \subset E_i$, we need to ensure that t is oriented correctly. In case that $\chi(t) = 0$, we can encode this directly by specifying that $\{a, b, e\}$ forms a rank-2 dependent set in the matroid M_i. Thus it remains to consider the case $\chi(t) \neq 0$. Let $t' = \{a, b, c\} \subseteq E_{i-1}$ be such that $\chi(t') \neq 0$. Note that such a triple t' must exists, as otherwise all points of E_i lie on a common line. This would be a contradiction to the fact that M_3 is formed by an independent triple. Using the orientation of t', we add a small constant number of helper points and we will enforce the correct orientation of t in M_i as well. Thus it remains to show the following lemma, where we use the notion of a free set of helper points defined in Section 4.

Lemma 11. *Given the independent triple $t' = \{a, b, c\}$ in a matroid M and another point e, we can enforce that in any valid representation of M, the triple $t = \{a, b, e\}$ is oriented identically to t', or that it is oriented opposite to t'. We do this by adding a constant number of helper points to M. Furthermore, this set of helper points is free.*

Proof: This construction goes in essentially two steps. First, we show how we can enforce an element x on the line $\ell = \ell(a, b)$ to be on the same side of a as b.
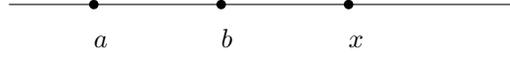
This first step relies on standard constructions to encode arithmetic operations as explained in Section 4. Although those constructions can be well described and understood, without any reference to arithmetic operations, the language of arithmetic operations gives a better intuition. The underlying idea is that we interpret a as zero, and b as one. Indeed, the constraint that x lies on the same side of a as b in any representation where the line ℓ_∞ is sent to infinity amounts to enforcing that $x > 0$ when interpreting a as zero and b as one. As explained in Section 4, this can be encoded using multiplication gadgets, in such a way that none of the helper points accidentally lie on a previously used line or point.

In the second step, we use the previous tool to enforce that e lies on the same (or the opposite) side of the line $\ell(a, b)$ as x.
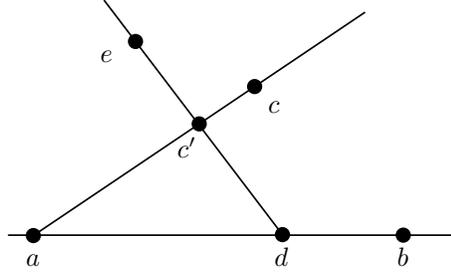
Fig. 5: Forcing c to be on the same side of $\ell(a, b)$ as e. Rays ending at a point denote the strict inequality constraints.

To this end, we first define a point c' for which we enforce that on the line $\ell(a, c)$, it lies on the same side of a as c. Then we define a point d situated on the line $\ell(a, b)$ and the line ℓ' through e and c', see Figure 5 The condition that e, c' are on the same side of d on the line ℓ' can be enforced using the previous gadgets. This condition is equivalent to c' and e being on the same side of $\ell(a, b)$. Lastly, by construction the triple $\{a, b, c'\}$ is oriented identically to $\{a, b, c\}$.

We can use the same tool to enforce that e and c lie on the opposite side of the line $\ell(a, b)$. We define c' as before, so that it lies on the same side of $\ell(a, b)$ as c, and then we simply need to enforce instead that c' lies on the opposite side of ℓ' with respect to d (where ℓ' and d are defined as above).

As explained in Section 4, the strict inequalities gadgets can be directly encoded into the independent sets of the matroid, and by construction, the helper points always have at least one degree of freedom, and thus can form a free set of points. Therefore, the matroid M_i is entirely defined by the dependency constraints indicated by the lines in the geometric constructions. This concludes the proof. □

We can now conclude the proof of Theorem 2 for rank-3 matroids. Given an order type O of rank 3, for which we can assume that there is at least one independent set of size 3 (otherwise the orientation predicates are trivial), we inductively encode the orientation predicates into a matroid using the helper points provided by Lemma 11. At each stage of the induction, the freedom of the set of helper points

can be used to ensure that the helper points do not yield any accidental dependencies with all the points and lines previously placed. At the end of the induction, by Lemma 11, any valid representation of the resulting matroid M induces a representation of O. Conversely, any representation of O can be extended to a representation of M. Therefore M simulates O. All the constructions can clearly be done in linear time, which concludes the proof.

Simulating Rank-k Matroids.   The construction for rank k is identical to that of rank 3 as explained above, with two exceptions. First, the induction basis starts with a matroid on k elements instead of 3. Second, we have to describe the simulation of an oriented k-tuple. For this, we use the same trick that forces a point to lie on a specific side of a line, where we just replace a line with a hyperplane P, as shown in Figure 6.
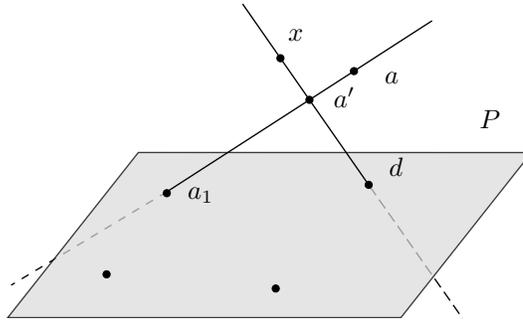
Fig. 6: Forcing x to be on the same side of P as a.

More precisely, in an inductive step where we add a point x, we need to enforce the orientation of all the independent k-tuples $t = \{a_1, \ldots a_{k-1}, x\}$. For this, we take another k-tuple $t' = \{a_1, \ldots a_{k-1}, a\}$ for which $\chi(t') \neq 0$, and we devise a gadget to encode the constraint that t is oriented identically, or opposite to t'. This will be done using the gadget described above that enforces that in any valid representation, for three aligned points a, b and c, c lies on the same side of a as b. That gadget was defined for rank 3 matroids and thus representations into $\mathbb{R}^2$, but readily works in higher dimensions: one must simply ensure using dependency constraints that all the points involved in the gadget lie on a common plane. The hyperplane at infinity will intersect this common plane in a line, which takes the role of the line at infinity in the gadgets.

Now, we proceed as in the rank 3 case: considering the hyperplane P generated by $\{a_1, \ldots a_{k-1}\}$, we first define a point a' that lies on the same side as a on the line $\ell(a_1, a)$. We introduce the point d at the intersection of P and the line going through a' and x. Then we enforce that x is on the same side of d as a'. In order to enforce that a' and x are on opposite sides of P, we instead enforce that x and a' are on opposite sides of d. Finally, we observe that all the helper points that we have introduced are free, where the notion of freedom is generalized to also disallow k-dimensional dependencies: once again this follows from the fact that all the helper points that we introduce have some wiggle room to be perturbed. The rest of the proof proceeds identically to the rank-3 case.

6 Proof of Theorem 1

For convenience, let us first restate Theorem 1

Theorem 1. MATROID $\mathbb{R}$-REPRESENTABILITY *is* $\exists\mathbb{R}$-*complete.*

The proof of Theorem 1 is by a direct reduction from the problem DISTINCT-ETR, using the arithmetic constructions described in Section 4. Let us start with an instance of DISTINCT-ETR given by variables $X = \{x_1, \ldots, x_n\}$ and constraints of the form

$$x + y = z, \quad x \cdot y = z, \quad x = 1, \quad x > 0,$$

for $x, y, z \in X$, with the promise that if there is a solution, then there is one where all the variables are pairwise different.

As in the proof of Theorem 2, we can construct the constant 0 using a constraint $x + y = x$, and thus, without loss of generality, by the distinctness assumption, we can assume that there are exactly two variables in X equal to respectively 0 or 1, and that the others are different from 0 and 1. By a slight abuse of language, we remove those from X and denote them directly by 0 and 1 in the rest of the description. We define a rank-3 matroid M as follows. First, we have a line ℓ consisting of three distinguished distinct points 0, 1 and ∞, as well as n distinct points (and distinct from $\{0, 1, \infty\}$) corresponding to the variables $\{x_1, \ldots x_n\}$. We also add a line at infinity ℓ_∞ going through ∞ and no other point.

We then use the gadgets from Section 4 to inductively encode all the constraints. Note that there is at most one constraint $x = 1$ which can be hardcoded from the start. Then let us assume inductively that we have defined a matroid M_i encoding the first i constraints. The $(i+1)$-th constraint is an addition, a multiplication, or a strict inequality, which can be encoded using a geometric construction as described in Section 4. Now, the key property of these constructions is that the set of helper points is free. Therefore, the only linear dependencies involved in the construction are those of that construction, which can be readily encoded into a matroid M_{i+1}.

We now prove that the DISTINCT-ETR instance has a solution with distinct variables if and only if the matroid M is representable over the reals. First, if DISTINCT-ETR has a solution, we obtain a representation of M over the reals by placing all the variables $x_1, \ldots, x_n$ on the line ℓ at the values indicated by the solution, and by sending the line at infinity to infinity. The geometric constructions are then represented one by one, and since the helper points are free, by perturbing them if needed we can ensure that they are all distinct, that no three of them are colinear, and that they do not form colinearities with previously placed points. Therefore, this constitutes a correct representation of the matroid. Conversely, given a representation of the matroid M over the reals, we read the values of the variables on the line ℓ using cross-ratios as explained in Section 4 (or equivalently we send ℓ_∞ to ∞ and use the oriented distance to 0). This gives us values for the variables of the DISTINCT-ETR instance. The definition of the addition, multiplication and strict inequality constructions ensures that each of the constraints will be satisfied. Furthermore, by definition of the matroid, all of the variables are distinct. This finishes the proof.

Acknowledgements

References

M. Abrahamsen.   Covering Polygons is Even Harder.   In N. K. Vishnoi, editor, *Proc. 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 375–386, 2022. doi:10.1109/FOCS52979.2021.00045.

M. Abrahamsen and T. Miltzow. Dynamic toolbox for ETRINV, 2019.

M. Abrahamsen, A. Adamaszek, and T. Miltzow. The Art Gallery Problem is ∃ℝ-complete. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 65–73, 2018. doi:10.1145/3188745.3188868.

M. Abrahamsen, T. Miltzow, and N. Seiferth. Framework for ER-Completeness of Two-Dimensional Packing Problems. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1014–1021, 2020. doi:10.1109/FOCS46700.2020.00098.

M. Abrahamsen, L. Kleist, and T. Miltzow. Training Neural Networks is ER-complete. In M. A. Ranzato, A. Beygelzimer, K. Nguyen, P. Liang, J. W. Vaughan, and Y. Dauphin, editors, *Advances in Neural Information Processing Systems (NeurIPS 2021)*, volume 34, 2021.

S. Basu and M.-F. Roy.   Bounding the radii of balls meeting every connected component of semi-algebraic sets.   *Journal of Symbolic Computation*, 45(12):1270–1279, 2010. doi:https://doi.org/10.1016/j.jsc.2010.06.009.

M. L. T. Berthelsen and K. A. Hansen. On the Computational Complexity of Decision Problems About Multi-player Nash Equilibria. In D. Fotakis and E. Markakis, editors, *International Symposium on Algorithmic Game Theory*, volume 11801 of *Lecture Notes in Computer Science*, pages 153–167, 2019. doi:10.1007/978-3-030-30473-7_11.

D. Bertschinger, C. Hertrich, P. Jungeblut, T. Miltzow, and S. Weber. Training fully connected neural networks is ∃ℝ-complete, 2022.

D. Bertschinger, N. El Maalouly, T. Miltzow, P. Schnider, and S. Weber. Topological art in simple galleries. *Discrete & Computational Geometry*, 2023. doi:https://doi.org/10.1007/s00454-023-00540-x.

V. Bilò and M. Mavronicolas. A Catalog of EXISTS-R-Complete Decision Problems About Nash Equilibria in Multi-Player Games. In N. Ollinger and H. Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016)*, Leibniz International Proceedings in Informatics (LIPIcs), pages 17:1–17:13, 2016. doi:10.4230/LIPIcs.STACS.2016.17.

V. Bilò and M. Mavronicolas. Existential-R-Complete Decision Problems about Symmetric Nash Equilibria in Symmetric Multi-Player Games. In V. Heribert and B. Vallée, editors, *34th Symposium on Theoretical Aspects of Computer Science (STACS 2017)*, volume 66 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 13:1–13:14, 2017. doi:10.4230/LIPIcs.STACS.2017.13.

A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G. M. Ziegler. *Oriented matroids*. Number 46 in Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1999.

L. Blum, M. Shub, and S. Smale. On a Theory of Computation and Complexity over the Real Numbers: NP-Completeness, Recursive Functions and Universal Machines. *Bulletin of the American Mathematical Society*, 21:1–46, 1989. doi:10.1090/S0273-0979-1989-15750-9.

A. Cameron. Kinser inequalities and related matroids. arXiv preprint, 2014.

A. Cameron. *Polytopal and structural aspects of matroids and related objects*. PhD thesis, Queen Mary University of London, 2017.

A. Cameron and D. Mayhew. Excluded minors for matroids satisfying Kinser's inequalities. *Graphs and Combinatorics*, 32(1):31–47, 2016. doi:https://doi.org/10.1007/s00373-015-1555-0.

A. Cameron and D. Mayhew. Excluded minors for the class of split matroids. *Australasian Journal of Combinatorics*, 79(2):195–204, 2021.

A. Cameron, R. Dinu, M. Michałek, and T. Seynnaeve. Flag matroids: Algebra and geometry. In *International Conference on Interactions with Lattice Polytopes*, pages 73–114. Springer, 2022. doi:10.1007/978-3-030-98327-7_4.

J. Canny. Some Algebraic and Geometric Computations in PSPACE. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing (STOC '88))*, pages 460–467, 1988. doi:10.1145/62212.62257.

J. Cardinal, S. Felsner, T. Miltzow, C. Tompkins, and B. Vogtenhuber. Intersection Graphs of Rays and Grounded Segments. *Journal of Graph Algorithms and Applications*, 22(2):273–294, 2018. doi:10.7155/jgaa.00470.

J. Cardinal, T. Miltzow, and M. Schaefer. A compendium of ER-complete problems. *in preparation*, 2024.

D. Chistikov, S. Kiefer, I. Marusic, M. Shirmohammadi, and J. Worrell. On Restricted Nonnegative Matrix Factorization. In I. Chatzigiannakis, M. Mitzenmacher, Y. Rabani, and D. Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 103:1–103:14, 2016. doi:10.4230/LIPIcs.ICALP.2016.103.

M. G. Dobbins, A. Holmsen, and T. Miltzow. A Universality Theorem for Nested Polytopes. arXiv:1908.02213, 2019.

M. G. Dobbins, L. Kleist, T. Miltzow, and P. Rzążewski. Completeness for the complexity class ∀∃ℝ and area-universality. *Discrete & Computational Geometry*, 70(1):154–188, 2023. doi:https://doi.org/10.1007/s00454-022-00381-0.

J. Erickson. Optimal Curve Straightening is ∃ℝ-Complete. arXiv preprint, 2019.

J. Erickson, I. van der Hoog, and T. Miltzow. Smoothing the gap between NP and ER. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1022–1033, 2020. doi:10.1109/FOCS46700.2020.00099.

J. Garg, R. Mehta, V. V. Vazirani, and S. Yazdanbod. ∃ℝ-Completeness for Decision Versions of Multi-Player (Symmetric) Nash Equilibria. *ACM Transactions on Economics and Computation*, 6(1):1:1–1:23, 2018. doi:10.1145/3175494.

J. Geelen and G. Whittle. Certifying non-representability of matroids over prime fields. *J. Comb. Theory, Ser. B*, 117:22–33, 2016. doi:10.1016/j.jctb.2015.11.005.

J. Geelen, B. Gerards, and G. Whittle. Solving Rota's conjecture. *Notices of the American Mathematical Society*, 61:736–743, 2014.

H. Hironaka. Triangulations of algebraic sets. In *Algebraic Geometry (Proc. Sympos. Pure Math., Vol. 29, Humboldt State Univ., Arcata, Calif., 1974)*, volume 29, pages 165–185, 1975.

P. Hlinený. On matroid representability and minor problems. In R. Kralovic and P. Urzyczyn, editors, *31st International Symposium on Mathematical Foundations of Computer Science 2006 (MFCS 2006)*, volume 4162 of *Lecture Notes in Computer Science*, pages 505–516. Springer, 2006. doi:10.1007/11821069_44.

P. M. Jensen and B. Korte. Complexity of matroid property algorithms. *SIAM Journal on Computing*, 11 (1):184–190, 1982. doi:10.1137/0211014.

J. Jeong, E. J. Kim, and S. Oum. Finding branch-decompositions of matroids, hypergraphs, and more. *SIAM J. Discret. Math.*, 35(4):2544–2617, 2021. doi:10.1137/19M1285895.

G. Jeronimo, D. Perrucci, and E. Tsigaridas. On the minimum of a polynomial function on a basic closed semialgebraic set and applications. *SIAM Journal on Optimization*, 23(1):241–255, 2013. doi:https://doi.org/10.1137/110857751.

P. Jungeblut, L. Kleist, and T. Miltzow. The complexity of the Hausdorff distance. In X. Goaoc and M. Kerber, editors, *38th International Symposium on Computational Geometry (SoCG 2022)*, volume 224 of *LIPIcs*, pages 48:1–48:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.SoCG.2022.48.

R. Kang and T. Müller. Sphere and Dot Product Representations of Graphs. *Discrete & Computational Geometry*, 47(3):548–569, 2012. doi:10.1007/s00454-012-9394-8.

D. E. Knuth. The asymptotic number of geometries. *Journal of Combinatorial Theory, Series A*, 16(3): 398–400, 1974. ISSN 0097-3165. doi:https://doi.org/10.1016/0097-3165(74)90063-6.

J. Kratochvíl and J. Matoušek. Intersection Graphs of Segments. *Journal of Combinatorial Theory, Series B*, 62(2):289–315, 1994. doi:10.1006/jctb.1994.1071.

S. Kratsch and M. Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. *J. ACM*, 67(3):16:1–16:50, 2020. doi:10.1145/3390887.

L. Lafforgue. *Chirurgie des grassmanniennes*. Number 19 in CRM Monograph Series. Amer. Math. Soc., 2003.

S. H. Lee and R. Vakil. Mnëv-Sturmfels universality for schemes. *A Celebration of Algebraic Geometry*, 18:457–468, 2013.

L. Lovász. Matroid matching and some applications. *J. Comb. Theory, Ser. B*, 28(2):208–236, 1980. doi:10.1016/0095-8956(80)90066-0.

L. László. Flats in matroids and geometric graphs. *Combinatorial Surveys*, 01 1977.

A. Lubiw, T. Miltzow, and D. Mondal. The complexity of drawing a graph in a polygonal region. *J. Graph Algorithms Appl.*, 26(4):421–446, 2022. doi:http://dx.doi.org/10.7155/jgaa.00602.

D. Marx. A parameterized view on matroid optimization problems. *Theor. Comput. Sci.*, 410(44):4471–4479, 2009. doi:10.1016/j.tcs.2009.07.027.

J. Matoušek. Intersection graphs of segments and ∃ℝ. arXiv preprint, 2014.

D. Mayhew. Matroid complexity and nonsuccinct descriptions. *SIAM J. Discret. Math.*, 22(2):455–466, 2008. doi:10.1137/050640576.

C. McDiarmid and T. Müller. Integer realizations of disk and segment graphs. *Journal of Combinatorial Theory, Series B*, 103(1):114–143, 2013. doi:10.1016/j.jctb.2012.09.004.

T. Miltzow and R. F. Schmiermann. On Classifying Continuous Constraint Satisfaction Problems. In N. K. Vishnoi, editor, *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 781–791, 2022. doi:10.1109/FOCS52979.2021.00081.

N. E. Mnëv. The Universality Theorems on the Classification Problem of Configuration Varieties and Convex Polytopes Varieties. In O. Y. Viro and A. M. Vershik, editors, *Topology and Geometry — Rohlin Seminar*, volume 1346 of *Lecture Notes in Mathematics*, pages 527–543. Springer, 1988. doi:10.1007/BFb0082792.

S. Oum and P. D. Seymour. Testing branch-width. *J. Comb. Theory, Ser. B*, 97(3):385–393, 2007. doi:10.1016/j.jctb.2006.06.006.

J. G. Oxley. *Matroid theory*, volume 3. Oxford University Press, USA, 2006.

J. Richter-Gebert and G. M. Ziegler. Realization Spaces of 4-Polytopes are Universal. *Bulletin of the American Mathematical Society*, 32(4):403–412, 1995. doi:10.1090/S0273-0979-1995-00604-X.

M. Schaefer. Complexity of Some Geometric and Topological Problems. In D. Eppstein and E. R. Gansner, editors, *Graph Drawing (GD 2009)*, volume 5849 of *Lecture Notes in Computer Science*, pages 334–344, 2010. doi:10.1007/978-3-642-11805-0_32.

M. Schaefer. *Realizability of Graphs and Linkages*, pages 461–482. Thirty Essays on Geometric Graph Theory. Springer, 2013. doi:10.1007/978-1-4614-0110-0_24.

M. Schaefer. Complexity of Geometric k-Planarity for Fixed k. *Journal of Graph Algorithms and Applications*, 25(1):29–41, 2021. doi:10.7155/jgaa.00548.

M. Schaefer and D. Štefankovič. Fixed Points, Nash Equilibria, and the Existential Theory of the Reals. *Theory of Computing Systems*, 60:172–193, 2017. doi:10.1007/s00224-015-9662-0.

M. Schaefer and D. Štefankovič. The Complexity of Tensor Rank. *Theory of Computing Systems*, 62(5): 1161–1174, 2018. doi:10.1007/s00224-017-9800-y.

Y. Shitov. A Universality Theorem for Nonnegative Matrix Factorizations. arXiv preprint, 2016.

Y. Shitov. The complexity of positive semidefinite matrix factorization. *SIAM Journal on Optimization*, 27(3):1898–1909, 2017. doi:10.1137/16M1080616.

P. W. Shor. Stretchability of Pseudolines is NP-Hard. In P. Gritzmann and B. Sturmfels, editors, *Applied Geometry And Discrete Mathematics*, volume 4 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 531–554, 1991. doi:10.1090/dimacs/004/41.

K. Truemper. On the efficiency of representability tests for matroids. *European Journal of Combinatorics*, 3(3):275–291, 1982. ISSN 0195-6698. doi:10.1016/S0195-6698(82)80039-5.

L. Tuncel, S. Vavasis, and J. Xu. Computational complexity of decomposing a symmetric matrix as a sum of positive semidefinite and diagonal matrices, 2022.