

How (and when) can you fit examples to logic-based hypothesis classes over infinite structures?

MICHAEL BENEDIKT, University of Oxford, UK

ALESSIO MANSUTTI, IMDEA Software Institute, Spain

We study fitting problems, sometimes called “training problems”, where given a finite sample of inputs and outputs, the goal is to determine whether a function in a certain class can produce those outputs, exactly or approximately, on the given inputs. We focus on the computational and descriptive complexity of fitting for logically-defined classes in common decidable structures, like the real ordered field and Presburger arithmetic. Our framework encompasses classes that arise in practice, including neural network classifiers.

We also study fitting for classes indexed by *distributions* over structures, where we learn a distribution over elements from the statistics it induces on definable subsets; a framework that connects to learning-based cardinality estimation in databases. In both settings, we characterize the complexity of fitting and show how these problems can often be decided through encodings into natural query languages over the sample, while establishing broad structural theorems based on combinatorial and model-theoretic properties.

CCS Concepts: • **Theory of computation** → **Machine learning theory; Logic and databases.**

Additional Key Words and Phrases: fitting problems, embedded finite models, RQC, NIP, automatic structures

ACM Reference Format:

Michael Benedikt and Alessio Mansutti. 2026. How (and when) can you fit examples to logic-based hypothesis classes over infinite structures?. *Proc. ACM Manag. Data* 4, 5 (PODS), Article 299 (November 2026), 20 pages. <https://doi.org/10.1145/3850134>

1 Introduction

A basic computational problem in machine learning is to fit sample inputs and outputs with a function in a given class. There are many notions of fitting possible, depending on the loss function used and whether the fit has to be exact or within some tolerance. For example, *linear regression* involves fitting samples with a linear function minimizing the aggregate error. Here we look at decision problems, asking if there is an exact fit for a set of input/output samples, or a fit with a certain bound on the error. The case of exact fitting is often called the *consistency problem*.

Our focus will be on *hypothesis classes arising from logical formulas over infinite structures*. We will first consider families of Boolean-valued functions (a.k.a. “concept classes”) defined by first-order formulas in a decidable infinite structure. The formula’s free variables are partitioned into two parts, the “object variables” and the “parameter variables”. By considering all instantiations of the parameter variables, one obtains families of Boolean-valued functions via logic.

EXAMPLE 1 (RECTANGLES). Our logic-based concept class could be the set of rectangles in the plane with sides parallel to the axes. This class is definable by a formula ϕ over the structure $(\mathbb{R}, <)$.

Authors’ Contact Information: Michael Benedikt, University of Oxford, UK, michael.benedikt@cs.ox.ac.uk; Alessio Mansutti, IMDEA Software Institute, Spain, alessio.mansutti@imdea.org.



This work is licensed under a Creative Commons Attribution 4.0 International License.

© 2026 Copyright held by the owner/author(s).

ACM 2836-6573/2026/11-ART299

<https://doi.org/10.1145/3850134>

There are four “parameter variables”, $y_1, \dots, y_4$, and two “object variables”, x_1 and x_2 . The formula ϕ constrains the point (x_1, x_2) to lie inside the rectangle defined by the parameter variables:

$$\phi(x_1, x_2; y_1, y_2, y_3, y_4) := y_1 < x_1 < y_3 < x_2 < y_4.$$

In the exact fitting problem for this class, we have sequences, $p_1, \dots, p_n$ and $q_1, \dots, q_m$, of pairs of reals: the *positive examples* and *negative examples*. We want to know if there is a rectangle (i.e., an instantiation of the parameter variables) that contains each p_i for $i \leq n$ and excludes each q_i for $i \leq m$. In the approximate fitting problem, we allow k errors, where k is an additional input. $\blacktriangleleft$

We also consider families of *real-valued functions* definable from logical formulas. A first way of generating such families is via variable-partitioned formulas over a numerical structure, like the reals or the integers, where for each parameter the instantiated formula defines a function.

EXAMPLE 2. Suppose our real-valued hypothesis class is the set of functions whose graph is a piecewise combination of two quadratics. This class can be constructed from a parametric function

$$f(x; t, a_1, b_1, c_1, a_2, b_2, c_2) = \begin{cases} a_1 \cdot x^2 + b_1 \cdot x + c_1 & \text{if } x \geq t \\ a_2 \cdot x^2 + b_2 \cdot x + c_2 & \text{otherwise,} \end{cases}$$

where the parameter variables are the threshold t and the coefficients of the two quadratic polynomials. This hypothesis class is definable by a formula over the real ordered field $(\mathbb{R}, +, \cdot, <)$. The exact fitting problem consists of a sequence of pairs of reals $(x_1, y_1), \dots, (x_n, y_n)$: we wish to know if there is an instantiation of the parameters such that $f(x_i) = y_i$ for each $i \leq n$. In the approximate fitting problem, deviations $f(x_i) \neq y_i$ are allowed, provided the total error across all $i \leq n$ stays within a given tolerance ϵ . $\blacktriangleleft$

A second way to obtain a real-valued hypothesis class, investigated in model theory [9, 33] and in data management [31], is to construct a *distribution class* on top of a (concept or real-valued) class. The idea is to start with a logically-defined hypothesis class over an *arbitrary* structure, and consider the family of real-valued functions obtained by “randomizing” its parameters or its objects:

EXAMPLE 3. Suppose our real-valued hypothesis class is parameterized by probability distributions over the points in the plane, with each function in the class mapping axis-aligned rectangles to their probability. The exact fitting problem will now consist of a sequence of pairs $(r_1, p_1), \dots, (r_n, p_n)$ where each r_i is a rectangle (specified by endpoints) and p_i is its desired probability. We wish to know if there is a probability distribution on the plane that can give rectangle r_i probability p_i . $\blacktriangleleft$

From the above we see that fitting problems have many variations. We will investigate two related questions for fitting problems over logically-defined hypothesis classes:

- What is the computational complexity of exact and approximate fitting problems, in terms of the sample size, that is, the *data complexity* of fitting problems?
- When can we solve fitting problems in polynomial time *by evaluating a first-order query over the sample*? Informally, we search for a first-order sentence quantifying only over the sample, expressed in the signature of the underlying structure (e.g., the real ordered field), along with predicates for accessing the sample, such that evaluating the sentence on a sample tells whether it is fittable. We formalize this with the notion of *restricted-quantifier formula* (Definition 9) that arises from earlier work in database theory [22, 38].

For instance, we will see that the fitting problem in Example 1 can be solved via “first-order query evaluation over the sample”. By extending this form of query evaluation with linear algebra, we can also solve fitting problems of distribution classes. For instance, we can show tractability of the fitting problem for Example 3 using first-order queries plus linear algebra.

Summary of Main Contributions. We overview our contributions, leaving more fine-grained TAKEAWAYS to the individual sections where the relevant formal definitions are in place. We formalize fitting problems (exact and approximate) for logically-defined classes over infinite structures. For many common decidable structures, we show that these fitting problems are tractable in the size of the sample, and can be reduced to *first-order query evaluation over the sample*. Although our results hold for concrete classes of interest, we provide a general property of a structure (*restricted quantifier collapse* [12]) that guarantees that this tractability-via-querying result holds.

In the case of logically-defined concept classes, we compare the phenomenon of fitting by first-order query evaluation over the sample with *finite VC dimension*, which characterizes efficiency in the number of samples needed to learn to some confidence and accuracy [17]. It is well-known that, in general, a concept class can have finite VC dimension but still have an intractable fitting problem (see Remark 3). We find that this discrepancy is more difficult to achieve for *logically-defined* concept classes: we show that any such example must come from a structure without restricted quantifier collapse, and it cannot be an *automatic structure* [27].

For real-valued function classes, formed using function classes as in Example 2 or via the randomization construction of [31] as in Example 3, we show decidability (and, often, tractability) of fitting in important cases. In the tractable cases the algorithms use a combination of first-order querying and linear algebra. Again, our results apply to particular function classes of interest (e.g., given by neural networks), but we also provide results that apply to a broad class of structures. We also identify classes of fitting problems where tractability cannot be shown without resolving longstanding open problems in theoretical computer science.

In studying these fitting problems, we require a mix of techniques. In some cases we use ideas from *classical model theory*, such as results relating combinatorial dimensions of families and definability of types (see UDTFS in Section 4). In other cases, fitting problems connect to core computational questions concerning numerical structures such as Presburger arithmetic and the real ordered field (see Section 6 and Section 7).

Organization. After preliminaries in Section 3, Section 4 establishes conditions on a structure implying that exact and approximate fitting for definable concept classes can be solved not only in PTIME, but via first-order queries over the sample. In Section 5 we look at automatic structures, where fitting may be intractable for some classes, but becomes tractable when the class has finite VC dimension. Sections 6 and 7 move to real-valued classes. Section 6 extends our analysis of tractable fitting to hypothesis classes given by formulas over a numerical structure (i.e., a structure whose domain is a subset of $\mathbb{R}$). Section 7 covers real-valued functions arising from randomizing a logically-defined class. We close with discussion in Section 8.

For space reasons, a number of proofs and examples are deferred to the extended version [14].

2 Related work

Our logic-based framework subsumes some standard constrained optimization problems like linear regression, which were already known to be tractable. While such problems are often attacked by imposing geometric conditions on the constraints and objective function, like convexity, here we consider arbitrary parameterized families, restricting only the underlying structure. Fitting problems, under the name of “training”, have been studied for standard neural architectures [1, 15, 26], and also in the presence of SoftMax and other exponential activation functions [29]. Our results deal with a different set of function classes, and our classes will fix the number of parameters that can be set, unlike [1, 15, 26, 29]. We discuss a few of the connections of our results to prior work in this area in Sections 4 and 6, see Remarks 6 and 8.

Our work has close connections with *embedded finite model theory* which emerged from the database theory community [10, 11, 13, 22, 36]: in fact, it can be seen as a restricted case of the evaluation problem dealt with in embedded finite model theory, as we will explain shortly. Our analysis of fitting using queries is in the same spirit as “in database learning”, e.g. [37]. Unlike those works, we consider hypothesis classes that may already involve quantification over an infinite set. And we are interested in training using queries directly on a materialized supervised sample, not how to do composition with queries that form the sample.

The complexity of fitting has also been studied for logical formulas on finite structures, with the focus being conjunctive queries [46]. In addition to the distinction in structures, in this prior line of work the logical formula again varies, while in our case it is fixed. Similarly [28] defines a learning framework allowing formulas referencing either finite or infinite structures; but the focus is not learning parameters in a fixed formula over an infinite structure, and thus both the results and the techniques are incomparable to ours.

Our approach to logic-based fitting problems includes distribution classes (Section 7), which were introduced in [31], motivated by machine learning approaches to cardinality estimation in databases. They are closely related to the “randomization of a structure” [8, 33], as explained in [3].

3 Preliminaries

3.1 Hypothesis classes and fitting problems

DEFINITION 1 (HYPOTHESIS CLASSES). A hypothesis class is a set $\mathcal{H} = \{f_y : y \in Y\}$ of real-valued functions $f_y : X \rightarrow \mathbb{R}$, for some set X , the range space. The functions in the class, also called hypotheses, are indexed by elements of a parameter space Y .

DEFINITION 2 (CONCEPT CLASSES). A concept class is a hypothesis class in which the image of all functions is restricted to $\{0, 1\}$. One can equivalently identify the functions in the class with subsets of the range space, i.e., a concept class C can be thought of as the set $\{\{x \in X : f(x) = 1\} : f \in C\}$.

DEFINITION 3 (SAMPLE). A sample is a finite collection $(s_1, t_1), \dots, (s_n, t_n)$ of pairs from $X \times \mathbb{R}$, where the training inputs $s_1, \dots, s_n$ are pairwise distinct, and $t_1, \dots, t_n$ are the training outputs.

Our focus is on the following computational problem:

DEFINITION 4 (APPROXIMATE FITTING PROBLEM). The approximate fitting problem for a hypothesis class $\mathcal{H}$ is the decision problem of determining, given as input a sample and a tolerance $\epsilon \geq 0$, whether there is a hypothesis $h \in \mathcal{H}$ that ϵ -fits the sample, that is, h achieves $\sum_{i=1}^n |h(s_i) - t_i| \leq \epsilon$.

We refer to the above problem as the ϵ -fitting problem when the tolerance ϵ is fixed rather than being part of the input. The 0-fitting problem requires an exact fit of the sample, and it is thus also called the *exact fitting problem*. In the case of fitting problems for *concept classes*, ϵ is a non-negative integer, and samples are of the form $(s_1, b_1) \dots (s_n, b_n)$, where $b_i \in \{0, 1\}$; so, they can equivalently be viewed as a collection $\{s_i : b_i = 1\}$ of *positive examples*, and $\{s_i : b_i = 0\}$ of *negative examples*.

EXAMPLE 4 (RECTANGLES, AGAIN). In the concept class of rectangles from Example 1, the *range space* is $\mathbb{R}^2$ (all points in the plane) and the *parameter space* is $\mathbb{R}^4$ (each parameter describes a rectangle). The approximate fitting problem for this class asks, given positive S^+ and negative S^- examples and a tolerance ϵ , if there is a rectangle $R \subseteq \mathbb{R}^2$ such that $|(R^2 \setminus R) \cap S^+| + |R \cap S^-| \leq \epsilon$. ◀

REMARK 1. While Definition 4 uses the L^1 norm $\sum_{i=1}^n |h(s_i) - t_i|$, all our complexity upper bounds also hold for the L^2 norm $\sqrt{\sum_{i=1}^n (h(s_i) - t_i)^2}$ and the hinge loss $\sum_{i=1}^n \max(0, 1 - t_i \cdot h(s_i))$. See [14].

Fitting problems for logic-based concept classes. We are interested in approximate fitting problems for hypothesis classes that are defined in terms of first-order predicate logic on some

infinite structure. For concept classes, this means considering *partitioned formulas*, that is, formulas $\phi(\vec{x}; \vec{y})$ with free variables partitioned into *object variables* $\vec{x}$ and *parameter variables* $\vec{y}$.

DEFINITION 5 (DEFINABLE CONCEPT CLASS). *Let $\mathfrak{M}$ be a structure with domain M . Let $\phi(\vec{x}; \vec{y})$ be a partitioned formula over (the vocabulary of) $\mathfrak{M}$, with j object variables and k parameter variables. Associated to ϕ is the concept class C_ϕ with range space M^j and parameter space M^k , defined as*

$$C_\phi := \{ \{\vec{q} \in M^j : \mathfrak{M} \models \phi(\vec{q}; \vec{p})\} : \vec{p} \in M^k \}.$$

Concept classes that are definable in terms of partitioned formulas are sometimes called *definable families*. We sometimes abuse notation by abbreviating “partitioned formula” by “formula”, and we sometimes leave the background structure $\mathfrak{M}$ implicit. Given a formula $\phi(\vec{x}; \vec{y})$, we say that a sample with positive examples S^+ and negative examples S^- is ϕ -*fittable* if there is a parameter $\vec{p}$ such that $\phi(\vec{q}; \vec{p})$ holds for every $\vec{q} \in S^+$, and $\neg\phi(\vec{q}; \vec{p})$ holds for every $\vec{q} \in S^-$.

REMARK 2. *Classical examples of concept classes arise as C_ϕ for suitable ϕ . For example, a binary classifier given by a feedforward neural network with ReLU activations can be described by a formula of the real ordered field $(\mathbb{R}, +, \cdot, <)$, where the inputs are objects and the weights are parameters. Our later results (e.g., Corollary 1 and Theorem 4) will show that these families have tractable fitting.*

3.2 Computational tameness assumptions on structures

To obtain complexity bounds for approximate fitting problems of definable families in the standard Turing machine model, we need two computational assumptions about the structure on which the formulas are defined. We write $\langle w \rangle$ for the length of a finite word w over some finite alphabet Σ .

ASSUMPTION 1 (EFFECTIVE CONSTANTS). *We assume that each structure $\mathfrak{M}$ is equipped with an (infinite) distinguished subset $E \subseteq M$ of effective constants and a surjection $\rho: D \rightarrow E$ from a set $D \subseteq \Sigma^*$ of finite words over some alphabet Σ . We assume membership in D is decidable in polynomial time: given $s \in \Sigma^*$, we can check if $s \in D$ in time polynomial in $\langle s \rangle$.*

We tacitly identify effective constants with their *representations*. For example, in saying that a procedure “takes as input an effective constant c and runs in time polynomial in the size of c ”, we mean that the procedure takes as input an $s \in D$ such that $\rho(s) = c$, and its running time is polynomial in $\langle s \rangle$. Representations extend to collections (tuples, sets, etc.) of effective constants, by defining their size as the sum of the sizes of all elements (from D) appearing in the collection. In fitting problems, we assume the *training inputs* s_i are tuples of (elements in D representing) effective constants and the *training outputs* t_i are rationals encoded as a pair of binary integers. Similarly, we assume the tolerance ϵ to be a non-negative rational, again encoded as a pair of binary integers. With these representations, we can now talk about the complexity of fitting:

DEFINITION 6 (PTIME APPROXIMATE FITTING). *A partitioned formula ϕ over a structure $\mathfrak{M}$ has PTIME approximate fitting whenever there is an algorithm for deciding the approximate fitting problem for C_ϕ that runs in polynomial time in the size of the sample and of the tolerance.*

We stress that throughout the paper we always look at the complexity of fitting problems *when formulas are fixed* (the input consists only of a sample and a tolerance). In particular, our fitting algorithms can perform any normalization on this formula (e.g., quantifier elimination) “for free”. Note also that having PTIME approximate fitting for each partitioned formula $\phi(\vec{x}; \vec{y})$ does not imply even decidability of the approximate fitting problem when ϕ is not fixed.

ASSUMPTION 2 (FIXED-FORMULA TRACTABLE EVALUATION). *We assume that for any first-order formula $\phi(x_1, \dots, x_k)$, there is an algorithm that takes as input a k -tuple of effective constants $\vec{c}$, and*

determines whether $\phi(\vec{c})$ holds in the structure, running in polynomial time in the total size of $\vec{c}$. We refer to this property of a structure as *fixed-formula tractable evaluation (FFTE)*.

EXAMPLE 5. Many commonly studied decidable structures have a natural set of effective constants where **FFTE** holds. In particular, this property holds for any structure with quantifier elimination where checking of quantifier-free formulas is tractable. Examples include the real ordered field $(\mathbb{R}, +, \cdot, <)$, where the effective constants are the rationals, or additive natural number arithmetic $(\mathbb{N}, +, <)$, also known as *Presburger arithmetic*, where the effective constants are all naturals. $\blacktriangleleft$

Unless otherwise stated, we always assume that structures have effective constants and **FFTE**.

3.3 Combinatorial and definability properties of structures

Two model-theoretic properties will play a key role: finite VC dimension and **RQC**.

DEFINITION 7 (VC DIMENSION). We say a concept class C on range space X shatters a finite subset S of X whenever, for every $E \subseteq S$, there is $h \in C$ such that $h(s) = 1$ for $s \in E$ and $h(s) = 0$ for $s \in S \setminus E$. The VC dimension of C is the supremum of the sizes of subsets of X shattered by C . A concept class has finite VC dimension whenever there is a finite bound on the sizes of sets it can shatter.

EXAMPLE 6 (RECTANGLES HAVE FINITE VC DIMENSION). The class of rectangles from Example 1 has VC dimension 4: there is a set of size 4 that you can shatter, namely a diamond-shaped set in the plane. However, it is rather simple to see that there is no such set of size 5. $\blacktriangleleft$

REMARK 3. Finite VC dimension does not imply a tractable fitting problem. There are effectively presented classes with finite VC dimension which do not have PTIME fitting (under standard complexity-theoretic assumptions), see for instance [39, Section 3], [4, Section 7].

REMARK 4. Finite VC dimension is necessary and sufficient for Probably Approximately Correct (PAC) learnability from random examples [17]. The definition of PAC learning will not be important in our work but if a class has finite VC dimension, any fitting algorithm suffices to PAC learn. Conversely, infinite VC dimension makes PAC learning impossible, even when not requiring an efficient algorithm.

DEFINITION 8 (FINITE VC DIMENSION OF FORMULAS, **NIP**, AND **IP**). A partitioned formula $\phi(\vec{x}; \vec{y})$ over a structure $\mathfrak{M}$ is said to have finite VC dimension if the corresponding concept class C_ϕ does. We say that a structure $\mathfrak{M}$ is **NIP** (short for “Not having the Independence Property”) when every partitioned formula over $\mathfrak{M}$ has finite VC dimension. Otherwise, $\mathfrak{M}$ is said to be **IP**.

EXAMPLE 7 (EXAMPLES OF **NIP** STRUCTURES). Both the real ordered field and Presburger arithmetic are **NIP** [45, App A]. The extension of Presburger arithmetic with the power predicate $2^{\mathbb{N}} := \{x \in \mathbb{N} : x \text{ is a power of two}\}$, denoted **PrA**($2^{\mathbb{N}}$), is also **NIP** [34]. $\blacktriangleleft$

We mentioned in the introduction that we are interested in whether we can solve fitting problems with a *first-order query that quantifies only over the elements in the sample*, not over the infinite structure. This is a special case of the notion of *restricted quantifier formula* from the database literature [22, 38] which we review. Below, $\mathfrak{M}$ is a structure with domain M and signature L .

DEFINITION 9 (RESTRICTED QUANTIFIER FORMULA). Let V be a (finite) relational signature disjoint from L . A restricted quantifier formula (**RQ** formula) in the signature $L \cup V$ is a formula built up inductively from (possibly quantified) first-order formulas in the signature L , via Boolean operators and restricted quantifiers of the form $\exists \vec{x} (T(\vec{x}, \vec{y}) \wedge \phi(\vec{x}, \vec{y}))$, where T is a relation symbol from V , $\vec{x}$ and $\vec{y}$ are tuples of variables, and ϕ is an **RQ** formula.

An *embedded finite model* for V over $\mathfrak{M}$ is an interpretation of each relation in V by a finite set of tuples in $\mathfrak{M}$. Let $\mathcal{E}$ be such a model. By expanding $\mathfrak{M}$ with $\mathcal{E}$ we obtain a structure, denoted $\mathfrak{M}, \mathcal{E}$,

for the signature $L \cup V$. We say that a formula ϕ in this signature *holds in* $\mathcal{E}$ whenever the structure $\mathfrak{M}, \mathcal{E}$ satisfies ϕ in the usual sense, i.e., $\mathfrak{M}, \mathcal{E} \models \phi$. Two formulas ϕ_1 and ϕ_2 in the signature $L \cup V$ are *equivalent over embedded finite models for V over $\mathfrak{M}$* whenever $\mathfrak{M}, \mathcal{E} \models \phi_1 \leftrightarrow \phi_2$ holds for every embedded finite model $\mathcal{E}$ for V over $\mathfrak{M}$. We omit “over $\mathfrak{M}$ ” when clear from context.

DEFINITION 10 (ACTIVE DOMAIN). *The active domain of an embedded finite model $\mathcal{E}$ for V is the finite subset of M obtained by taking the projections of each interpretation of each relation symbol in $\mathcal{E}$. Without loss of generality, we can let **RQ** formulas replace restricted quantifiers with quantifiers over the active domain **ADom**. For example, if $V = \{R(x_1), S(x_1, x_2)\}$, the formula $\exists x \in \mathbf{ADom} \phi$ stands for $\exists x (R(x) \wedge \phi) \vee \exists x, y (S(x, y) \wedge \phi) \vee \exists x, y (S(y, x) \wedge \phi)$. We call these active domain quantifiers.*

DEFINITION 11 (RESTRICTED QUANTIFIER COLLAPSE). *We say that $\mathfrak{M}$ has Restricted Quantifier Collapse (“ $\mathfrak{M}$ is **RQC**”) whenever for every relational signature V disjoint from L , every first-order formula in the signature $L \cup V$ is equivalent, over embedded finite models for V , to an **RQ** formula.*

REMARK 5. *It is known that **RQC** implies **NIP**, see [13, Proof of Lemma 3.9].*

EXAMPLE 8 ((NON-)EXAMPLES OF **RQC STRUCTURES).** All structures listed in Example 7 are **RQC**. For **PrA**($2^{\mathbb{N}}$) we give an argument in [14]. The other examples are well-known, and follow from general model-theoretic criteria that imply **RQC**, like o-minimality, NFCP [23], and distality [7].

Some **NIP** structures are not **RQC**. One example is the structure on the natural numbers with the equivalence relation $E(n, m)$ holding when the largest power of two below n is the same as the largest below m . Note that equivalence classes of this relation are finite, but the size of classes is unbounded. Let $V = \{C(x)\}$ and consider the sentence $\exists x \forall y [E(x, y) \rightarrow C(y)]$ stating that C contains an equivalence class. One can show that this is not equivalent to an **RQ** sentence. $\triangleleft$

4 RQ fitting for definable classes

TAKEAWAY. *We show that **RQC** implies PTIME approximate fitting. Via the notion of Uniform Definability of Types over Finite Sets (**UDTFS**), we prove that **RQC** is equivalent to “**RQ** fitting”—fitting via an **RQ** sentence—even when restricting **RQ** fitting to partitioned formulas with a single parameter.*

Throughout the section, let $\mathfrak{M}$ be a structure with domain M and signature L , and V be a relational signature disjoint from L . Our study of fitting problems for definable concept classes starts from the following motivating example, which illustrates why **RQC** is a very natural notion in this context.

EXAMPLE 9. Consider the partitioned formula $\phi(x; y) := x < y$ over the structure $(\mathbb{R}, <)$. Introducing two unary predicates P and N , to be interpreted as the sets of positive and negative examples of an input sample, the 0-fitting problem for ϕ can be expressed as the following first-order sentence:

$$\exists y (\forall x (P(x) \rightarrow \phi(x, y)) \wedge \forall x (N(x) \rightarrow \neg \phi(x, y))) .$$

This general pattern works for any ϕ , with the obvious adjustments to the number of object variables and parameter variables. But a sample is $(x < y)$ -fittable if and only if the positive examples lie below the negative ones, and so the problem can alternatively be encoded as the **RQ** sentence $\forall x (P(x) \rightarrow \forall z (N(z) \rightarrow x < z))$. This sentence describes a PTIME algorithm for the 0-fitting problem for ϕ : iterate through all positive examples x and all negative ones z , checking $x < z$. $\triangleleft$

Example 9 motivates the following notion:

DEFINITION 12 (RQ** FITTING).** *Consider a partitioned formula $\phi(x_1, \dots, x_j; \vec{y})$ over $\mathfrak{M}$, and let $V = \{P, N\}$ be a relational signature with two j -ary relations. We say that ϕ has **RQ** fitting if there is an **RQ** sentence ϕ_{fit} in the signature $L \cup V$, with the following property: consider a finite sample S of positive examples S^+ and negative examples S^- , and let $\mathcal{E}$ be the embedded finite model for V over $\mathfrak{M}$ that interprets $P = S^+$ and $N = S^-$. Then, ϕ_{fit} holds in $\mathcal{E}$ if and only if S is ϕ -fittable.*

We say that $\mathfrak{M}$ has **RQ** fitting whenever every partitioned formula over $\mathfrak{M}$ has **RQ** fitting.

We remark that, as with **RQC** (Definition 11), the notion of **RQ** fitting is semantic and not algorithmic: it must hold for all finite sets S , not only those composed of effective constants.

Recalling that we assume effective constants and **FFTE**, the problem of evaluating a fixed **RQ** sentence on a given embedded finite model is in PTIME:

PROPOSITION 1. *Fix an **RQ** sentence ϕ in $L \cup V$. There is a PTIME algorithm that given an embedded finite model $\mathcal{E}$ for V over $\mathfrak{M}$, decides $\mathfrak{M}, \mathcal{E} \models \phi$. Thus, any formula with **RQ** fitting has PTIME 0-fitting.*

By “takes as input an embedded finite model” we mean that it takes as input lists of tuples of (representations of) effective constants from M , encoding the interpretation of the relations in V .

We extend Proposition 1 from exact to approximate fitting, at the cost of assuming **RQ** fitting for the whole structure instead of single formulas.

THEOREM 1. *If $\mathfrak{M}$ has **RQ** fitting, then any formula over $\mathfrak{M}$ has PTIME approximate fitting.*

By definition, **RQC** implies **RQ** fitting, and so from Theorem 1 we obtain:

COROLLARY 1. *If $\mathfrak{M}$ is **RQC**, then every formula ϕ over $\mathfrak{M}$ has **RQ** fitting and PTIME approximate fitting. In particular, this holds when ϕ is a formula over any of the structures in Example 7.*

REMARK 6 (CONNECTION TO NEURAL NETWORK CLASSIFIERS). *Consider the family of feedforward neural network classifiers with ReLU activations: a network which is supplemented with a threshold on a distinguished output gate to produce a Boolean. If we fix the number of network weights, such a family is definable in the real ordered field, even when the threshold is left as an additional parameter. Corollary 1 shows that we can solve the approximate fitting problem for this family in polynomial time in the size of the sample. For exact fitting, we can use a very basic kind of database query to check whether the sample is fittable (Proposition 1). But note that the exponent in the polynomial time algorithm will be proportional to the number of parameters. Similarly, the query will have as many table aliases as there are parameters in the architecture. Thus even for moderately-sized architectures, these results do not provide practical algorithms.*

4.1 Ideas behind the proof of Theorem 1

The proof of Theorem 1 has two steps. First, we show **RQ** fitting for all formulas implies $\mathfrak{M}$ is **NIP**. We prove something stronger: **RQ** fitting for a single parameter suffices. This will be useful later.

LEMMA 1. *If every formula $\phi(\vec{x}; y)$ over $\mathfrak{M}$ has **RQ** fitting, then $\mathfrak{M}$ is **NIP**.*

PROOF IDEA. We assume for contradiction that $\mathfrak{M}$ has a partitioned formula, which we can take to be of the form $\phi(\vec{x}; y)$, with infinite VC dimension. From this we construct a formula $\gamma(\vec{z}; y)$ (still with a single parameter) capturing 3-colorability: finite graphs can be encoded as samples so that a sample is γ -fittable if and only if the graph is 3-colorable. We then show that γ having **RQ** fitting implies that there is an **RQ** sentence γ' over the simpler structure $(\mathbb{Z}, +, <)$ with additional binary relation E such that, on embedded finite models, γ' holds exactly when the graph induced by E is 3-colorable. This gives a contradiction: γ' is an **RQ** sentence whose L -subformulas are quantifier-free and use only order and addition, so evaluating γ' on an embedded finite model is in AC^0 , but 3-colorability is not in AC^0 [25]. $\square$

In the second step, we use the fact that $\mathfrak{M}$ is **NIP** and has PTIME 0-fitting to provide a PTIME algorithm for approximate fitting. By the Shelah-Sauer lemma (Fact 1, below), **NIP** implies that the number of *realizable partitions* of a set S of elements in the range space—i.e., the partitions of S into positive and negative examples that can be fitted exactly—is polynomial in the size of S .

FACT 1 ([43]). *Let C be a concept class on range space X and with VC dimension $d \in \mathbb{N}$. For every set $S \subseteq X$ of size m , the set $\{ \{s \in S : f(s) = 1\} : f \in C \}$ has size at most $\sum_{i=0}^d \binom{m}{i}$.*

Fact 1 does not provide a method for constructing the realizable partitions, but we can use PTIME 0-fitting to circumvent this problem: by issuing polynomially many evaluations to **RQ** sentences (in PTIME by Proposition 1) it is possible to compute all realizable partitions in polynomial time.

LEMMA 2. *Let C be a concept class with finite VC dimension and PTIME 0-fitting. There is a polynomial-time algorithm for computing the realizable partitions of an input finite subset S of X .*

PROOF SKETCH. Let $S = \{s_1, \dots, s_n\}$. We compute the realizable partitions of $\{s_1, \dots, s_{j+1}\}$ from those of $\{s_1, \dots, s_j\}$: each realizable partition (S^+, S^-) yields the two candidates $(S^+ \cup \{s_{j+1}\}, S^-)$ and $(S^+, S^- \cup \{s_{j+1}\})$, and each candidate is kept if it passes the 0-fitting test. By Fact 1, at every step there are only polynomially many realizable partitions. $\square$

Then, approximate fitting reduces to determining whether the smallest distance between any realizable partition and the given sample is at most the tolerance ϵ , establishing Theorem 1.

4.2 The relationship between RQC and RQ fitting

RQ fitting captures the idea of being able to solve a fitting problem by evaluating an **RQ** sentence over the positive and negative examples. A priori, this seems like a much weaker property than **RQC**, which allows for more general queries over (arbitrary) embedded structures. Surprisingly, this intuition turns out to be wrong: **RQ** fitting is *equivalent* to **RQC**, with this equivalence holding even when **RQ** fitting is considered *only for formulas with a single parameter variable*.

THEOREM 2. $\mathfrak{M}$ has **RQ** fitting $\iff$ every $\phi(\vec{x}; y)$ over $\mathfrak{M}$ has **RQ** fitting $\iff \mathfrak{M}$ is **RQC**.

PROOF SKETCH. The interesting direction is from one-parameter **RQ** fitting to **RQC**. By Lemma 1, one-parameter **RQ** fitting implies that $\mathfrak{M}$ is **NIP**. We rely on a result in model theory of Chernikov and Simon [21], showing that **NIP** structures have a property called “Uniform Definability of Types over Finite Sets” (**UDTFS**). Formally, a partitioned formula $\gamma(\vec{x}; \vec{y})$ with j object variables and k parameter variables has **UDTFS** (with respect to $\mathfrak{M}$) whenever:

There is an L formula $\delta(\vec{y}; \vec{p})$ such that for any finite set S of k -tuples in $\mathfrak{M}$ of cardinality at least two, for any j -tuple $\vec{x}_0$, there is $\vec{p}_0$ from S such that $\mathfrak{M} \models \forall \vec{y} \in S \ \gamma(\vec{x}_0, \vec{y}) \leftrightarrow \delta(\vec{y}, \vec{p}_0)$.

Let V be a relational signature disjoint from the signature L of $\mathfrak{M}$. By induction on the number of unrestricted quantifiers, we show that every first-order formula in $L \cup V$ is equivalent over embedded finite models for $\mathfrak{M}$ to an **RQ** formula. The induction step considers a formula

$$\psi(\vec{z}) := \exists y \ Q_1(u_1) \dots Q_k(u_k) \ \gamma(y, \vec{u}, \vec{z}),$$

where γ is a Boolean combination of L -formulas and V -atoms, $\exists y$ is an unrestricted quantifier, and each $Q_i(u_i)$ is an active domain quantifier (the induction hypothesis ensures we can restrict our attention to these quantifiers). For simplicity, we restrict the proof sketch to embedded finite models with active domain of at least two elements, and assume that γ is an L -formula. We apply **UDTFS** on γ to get a formula $\delta(\vec{u}; \vec{p})$. Let $\eta(\vec{z}, \vec{p}) := \exists y \forall u_1 \in \mathbf{ADom} \dots \forall u_k \in \mathbf{ADom} : \gamma(y, \vec{u}, \vec{z}) \leftrightarrow \delta(\vec{u}, \vec{p})$.

The formula $\psi(\vec{z})$ is equivalent, over embedded finite models with active domain at least two, to

$$\exists p_1 \in \mathbf{ADom} \dots \exists p_r \in \mathbf{ADom} (\eta(\vec{z}, \vec{p}) \wedge \exists y \ Q_1(u_1) \dots Q_k(u_k) : \delta(\vec{u}, \vec{p})).$$

The variable y does not appear in δ , and thus $\exists y$ can be eliminated from the rightmost conjunct. It remains to convert η into an **RQ** formula. We consider the fitting problem for the one-parameter formula $\eta_{qf}(\vec{u}, \vec{z}, \vec{p}; y) := (\gamma(y, \vec{u}, \vec{z}) \leftrightarrow \delta(\vec{u}, \vec{p}))$, on positive examples only. This problem corresponds to the sentence $\exists y \forall \vec{u}, \vec{z}, \vec{p} : P(\vec{u}, \vec{z}, \vec{p}) \rightarrow \eta_{qf}(\vec{u}, \vec{z}, \vec{p}; y)$, for which we can find an equivalent

RQ sentence χ , by the **RQ** fitting hypothesis. Wlog., we can assume that the variables $\vec{z}$ and $\vec{p}$ do not occur in the sentence χ . We replace all occurrences of $P(\vec{w}_0, \vec{w}_1, \vec{w}_2)$ in χ (where $\vec{w}_0$ and $\vec{w}_1$ have the same cardinality as $\vec{u}$ and $\vec{z}$, respectively) with the conjunction $\mathbf{ADom}(\vec{w}_0) \wedge \vec{w}_1 = \vec{z} \wedge \vec{w}_2 = \vec{p}$. The proof concludes by showing that the resulting **RQ** formula is equivalent to η . $\square$

5 PTIME fitting beyond RQC

TAKEAWAY. *We show that **IP** structures do not necessarily have PTIME 0-fitting (unless $P = NP$); yet formulas of finite VC dimension over an automatic structure have PTIME approximate fitting.*

Theorem 1 shows that **RQC** (equivalently, **RQ** fitting) suffices for PTIME approximate fitting. Recall that the first step of the proof of this theorem shows that **IP** structures cannot be **RQC** by showing an encoding of 3-colorability. Intuitively, this tells us that for many **IP** structures, there are fitting problems that are NP-hard (although formalizing a general statement requires further computational assumptions). A concrete example of this fact is given by *Büchi arithmetic*, the expansion of Presburger arithmetic with the binary relation $V_2 := \{(x, y) \in \mathbb{N}^2 : y \text{ is the largest power of 2 that divides } x\}$. This structure is **IP**: the partitioned formula $\text{bit}(x; y) := V_2(x, x) \wedge \exists y_1, y_2 : y = y_1 + x + y_2 \wedge y_1 < x \wedge (y_2 = 0 \vee \exists z : x < z \wedge V_2(y_2, z))$ holds exactly when x is a power of two appearing in the binary expansion of y , and thus shatters the powers of two. It also has **FFTE** and effective constants (all naturals encoded in binary); this follows from the fact that it is an automatic structure (defined below) [18, 19]. By reduction from 3-colorability, we see that there are formulas over Büchi arithmetic for which the 0-fitting problem is NP-hard. Moreover, we can easily show that all formulas over Büchi arithmetic have approximate fitting problems in NP (see [14]).

PROPOSITION 2. *The approximate (and exact) fitting problem of any formula over Büchi arithmetic is in NP. There are formulas for which the 0-fitting problem is NP-complete.*

The structure $\mathbf{PrA}(2^{\mathbb{N}})$ is a reduct of Büchi arithmetic with **RQC** (Example 8), and there are a number of other reducts of Büchi arithmetic known to be **RQC** [13]. Consequently, we already know that many definable concept classes over Büchi arithmetic admit PTIME approximate fitting. By examining their automata representation, we now show that, for all definable concept classes over Büchi arithmetic, *finite VC dimension implies tractable fitting*.

Automatic structures. For a sequence $w_1, \dots, w_k$ of finite words over an alphabet Σ , their *convolution* $w_1 \otimes \dots \otimes w_k$ is the word over the alphabet $(\Sigma \cup \{\square\})^k$ obtained by first padding each w_i with a fresh symbol $\square \notin \Sigma$ to a common length, to then take the component-wise product. For instance, for $w_1 = a$ and $w_2 = bcd$, we have $w_1 \otimes w_2 = \begin{bmatrix} a \\ \square \end{bmatrix} \cdot \begin{bmatrix} \square \\ b \end{bmatrix} \cdot \begin{bmatrix} \square \\ c \end{bmatrix} \cdot \begin{bmatrix} \square \\ d \end{bmatrix}$, where $\cdot$ denotes concatenation.

A relation $R \subseteq (\Sigma^*)^k$ is *regular* if the language $\{(w_1 \otimes \dots \otimes w_k) \cdot p : (w_1, \dots, w_k) \in R, p \in \{\square\}^*\}$ is regular. A relational structure $\mathfrak{M}$ is *automatic* if there is a regular language $D \subseteq \Sigma^*$ and a surjection $r : D \rightarrow M$ such that, for every relation $R \subseteq M^k$ in the structure, the relation $\{(w_1, \dots, w_k) \in D^k : (r(w_1), \dots, r(w_k)) \in R\}$ is regular. See [27] for an overview on automatic structures. For an automatic structure, we take M as the set of effective constants, with D providing their encodings.

THEOREM 3. *Consider a partitioned formula $\phi(\vec{x}; \vec{y})$ over an automatic structure $\mathfrak{M}$. If ϕ has finite VC dimension, then ϕ has PTIME approximate fitting.*

PROOF. We assume a single object variable and a single parameter variable. Let $\mathcal{A}$ be the minimal DFA recognizing the relation defined by $\phi(x; y)$, and Q be its set of states. Let $S = (s_1, b_1), \dots, (s_n, b_n)$ be the sample and ϵ be the tolerance. By a pumping argument, if some parameter ϵ -fits S , then one such parameter has an encoding of length polynomial in the size of S . So, we can pad examples and candidate parameters with $\square$ to a common polynomial length ℓ .

For $k \in [0..\ell]$, let s_i^k be the prefix of s_i of length k . A tuple $(q_1, \dots, q_n) \in Q^n$ is *S-reachable at k* if there is a word y of length k such that, for every $i \in [1..n]$, $\mathcal{A}$ reaches q_i after reading $s_i^k \otimes y$. The algorithm computes the set of all *S-reachable tuples at k*, for $k = 0, 1, \dots, \ell$. For $k = 0$, the only such tuple is $(q_0, \dots, q_0)$, where q_0 is the initial state of $\mathcal{A}$. For $k \geq 1$, the set is obtained from the one at $k - 1$ by reading one more letter. Together with each tuple, the algorithm also stores the state reached by y in a DFA recognizing the padded encodings of parameters (to make sure that y belongs to $D \cdot \square^*$). Finally, it checks whether some tuple at ℓ , reached by the encoding of a parameter, agrees with S (in terms of accepting/rejecting states) on all but at most ϵ examples.

The algorithm runs in polynomial time because there are polynomially many *S-reachable tuples* at each k . To see this, for every two distinct states $p, q \in Q$, fix a word that leads to an accepting state from exactly one of p and q ; this word exists because $\mathcal{A}$ is minimal. Let $u_1 \otimes v_1, \dots, u_h \otimes v_h$ be these words; so, $h \leq \binom{|Q|}{2}$. Every state q is then uniquely determined by the set of indices $j \in [1..h]$ such that $u_j \otimes v_j$ leads from q to an accepting state (this set may be empty for one state). Fix $k \in [0..\ell]$. Let $(q_1, \dots, q_n)$ be the tuple reached at k via a word y . Since $(s_i^k \otimes y) \cdot (u_j \otimes v_j) = (s_i^k \cdot u_j) \otimes (y \cdot v_j)$, the word $u_j \otimes v_j$ leads from q_i to an accepting state if and only if $\mathcal{A}$ accepts $(s_i^k \cdot u_j) \otimes (y \cdot v_j)$. Hence, the tuple only depends on the h sets

$$X_j(y) := \{t \in T_j : \mathcal{A} \text{ accepts } t \otimes (y \cdot v_j)\}, \quad \text{where } T_j := \{s_i^k \cdot u_j : i \in [1..n]\} \text{ and } j \in [1..h].$$

That is, if two words y and y' of length k satisfy $X_j(y) = X_j(y')$ for every $j \in [1..h]$, then they reach the same tuple. Let C be the class of all sets $\{t : \mathcal{A} \text{ accepts } t \otimes z\}$, where z ranges over words. Since $\mathcal{A}$ only accepts padded encodings of pairs satisfying ϕ , the VC dimension of C is at most the VC dimension d of ϕ . As $|T_j| \leq n$, Fact 1 applied to C and T_j gives at most $\sum_{i=0}^d \binom{n}{i} \leq (n+1)^d$ choices for each $X_j(y)$. So, there are at most $(n+1)^{dh}$ *S-reachable tuples at k*, which is polynomial in n since d and h only depend on ϕ . $\square$

Note that the converse of Theorem 3 fails: there are definable concept classes in Büchi arithmetic with infinite VC dimension but tractable fitting: *bit*($x; y$) can easily be seen to give such an example.

Our results leave open the question of whether **NIP** and **FFTE** alone guarantee PTIME (exact or approximate) fitting, with Theorems 1 and 3 telling us that a counterexample to this would have to be a structure that is neither automatic nor with **RQC**. One way such a **NIP** structure might still have PTIME fitting is if it can be expanded into an **RQC** one (while retaining **FFTE**):

EXAMPLE 10. Consider the last structure from Example 8, involving an equivalence relation on the natural numbers with classes of unbounded finite size. As remarked there, this structure is not **RQC**, and hence lacks **RQ** fitting by Theorem 2. However, it can be expanded to $\mathbf{PrA}(2^{\mathbb{N}})$, which is **RQC** and thus does have PTIME fitting. $\blacktriangleleft$

Whether every **NIP** structure can be expanded to be **RQC** is open. We note that the structure from Example 10 can alternatively be expanded by a total order to get **RQC**. However, it is known [44] that there are **NIP** structures that cannot be expanded to have a linear order without losing **NIP**, thus we cannot hope to be able to add a total order in general. In Example 8 we recalled that there are some broad model-theoretic properties that imply **RQC**, most notably distality [7, 21]. Unfortunately, it has been shown that there are **NIP** structures that cannot be expanded to a distal structure [30]; but these counterexamples are still **RQC**.

6 Fitting for logically-defined real-valued functions

TAKEAWAY. We move our attention to hypothesis classes given by definable real-valued functions. If the function is “piecewise”, we show that the approximate fitting problem is tractable over many **RQC** structures. Otherwise, one can readily obtain fitting problems that are sum-of-square-roots-hard.

We now consider fitting problems for hypothesis classes of real-valued functions. As mentioned in the introduction, such classes can be built from logic in several ways. In this section, we take the most direct approach: definable families of functions over a *numerical structure*, i.e., one whose domain is a subset of $\mathbb{R}$. One can represent such a family through a partitioned formula $\phi(\vec{x}; z; \vec{y})$ with variables split into object variables $\vec{x}$, a single *output variable* z , and parameter variables $\vec{y}$. The formula ϕ is assumed to be functional: for every parameter $\vec{y}^*$ and object $\vec{x}^*$, there is a unique z^* such that $\phi(\vec{x}^*; z^*; \vec{y}^*)$. We write f_ϕ for the function defined by ϕ , i.e., the function satisfying $f_\phi(\vec{x}, \vec{y}) = z \leftrightarrow \phi(\vec{x}; z; \vec{y})$. The *definable function class* $\mathcal{H}_{f_\phi}$ given by f_ϕ is the hypothesis class containing all functions $f_{\vec{p}} := (\vec{x} \mapsto f_\phi(\vec{x}, \vec{p}))$, where $\vec{p}$ ranges over the parameters.

REMARK 7. *It is easy to see that the 0-fitting problem for definable function classes trivially reduces to the concept class case: given an input sample $S := (\vec{s}_1, t_1), \dots, (\vec{s}_n, t_n)$, we have*

$$(\exists \vec{y} \sum_{i=1}^n |f_\phi(\vec{s}_i, \vec{y}) - t_i| \leq 0) \leftrightarrow (\exists \vec{y} \bigwedge_{i=1}^n \phi(\vec{s}_i; t_i; \vec{y})),$$

and so we can alternatively solve the 0-fitting problem for $\phi(\vec{x}, z; \vec{y})$, treating S as the set of positive examples. In particular, by Corollary 1 this implies that for every **RQC** numerical structure, like the real ordered field $(\mathbb{R}, +, \cdot, <)$ or Presburger arithmetic, definable function classes have PTIME 0-fitting. In general, we will not obtain an analogous result for approximate fitting.

6.1 Approximate fitting for piecewise functions

Consider a formula $\phi(\vec{x}; z; \vec{y})$. We say that f_ϕ (and ϕ) are *piecewise* whenever ϕ is of the form

$$\bigvee_{i=1}^m \psi_i(\vec{x}, \vec{y}) \wedge z = \ell_i(\vec{x}, \vec{y}), \quad (1)$$

where the *guards* $\psi_1, \dots, \psi_m$ form a partition of the range and parameter spaces, and each ℓ_i is a term. In this case, we say that $\mathcal{H}_{f_\phi}$ is a *piecewise function class*. We show that, for piecewise function classes, the landscape of approximate fitting problems reduces to satisfiability in fixed dimension.

LEMMA 3. *Let $\mathfrak{M}$ be a numerical structure with domain M containing the naturals, and with signature L containing $0, 1, +$, and $<$, interpreted as usual. Assume $\mathfrak{M}$ is **NIP**, and all its partitioned formulas have PTIME 0-fitting. Let f_ϕ be a piecewise function given by a formula $\phi(\vec{x}; z; \vec{y})$ over $\mathfrak{M}$. There is a polynomial-time algorithm that given as input a sample S and a tolerance ϵ , computes $\phi'(\vec{y})$ over $\mathfrak{M}$, with the property that $\mathcal{H}_{f_\phi}$ ϵ -fits S if and only if $\mathfrak{M} \models \exists \vec{y} \phi'(\vec{y})$. If ϕ is quantifier-free, so is ϕ' .*

PROOF SKETCH. Let ϕ be as in Formula (1), and let $S = (\vec{s}_1, t_1), \dots, (\vec{s}_n, t_n)$ be the sample. Writing down the definition of the approximate fitting problem, we see that it is equivalent to deciding

$$\exists \vec{y} : \bigvee_{\mu: [1..n] \rightarrow [1..m]} \left(\bigwedge_{j=1}^n \psi_{\mu(j)}(\vec{s}_j, \vec{y}) \wedge \sum_{j=1}^n |\ell_{\mu(j)}(\vec{s}_j, \vec{y}) - t_j| \leq \epsilon \right).$$

We show that only polynomially many pairs of a map μ and a combination of signs for the terms inside absolute values are realized by some $\vec{y}$, and that these pairs can be computed in polynomial time. For this, we add two object variables u and v , and consider the formula $\chi(\vec{x}, z, u, v; \vec{y})$ stating that $\psi_u(\vec{x}, \vec{y})$ holds and that $v \in \{0, 1\}$ encodes the sign of $\ell_u(\vec{x}, \vec{y}) - z$; this is expressible since M contains the naturals. The pairs above correspond to the realizable partitions of $\{(\vec{s}_j, t_j, i, b) : j \in [1..n], i \in [1..m], b \in \{0, 1\}\}$ for χ , and we conclude by Fact 1 and Lemma 2. Therefore, the algorithm from the statement of the lemma computes a formula ϕ' equivalent to the one above, but with only polynomially many disjuncts and no absolute values. $\square$

Thanks to Lemma 3, we obtain the following:

THEOREM 4. *The approximate fitting problem is in PTIME for every definable function class over the real ordered group or over Presburger arithmetic. For the real ordered field, the problem is in PTIME for every piecewise function class.*

PROOF SKETCH. First, we note that every definable function over $(\mathbb{R}, +, <)$ is a piecewise function f_ϕ given by a quantifier-free formula ϕ from the expanded structure $(\mathbb{R}, +, -, 0, 1, \{\frac{\cdot}{d}\}_{d \geq 2}, <)$, where $\frac{\cdot}{d}$ denotes division by d . A similar result holds for Presburger arithmetic, with respect to the expanded structure $(\mathbb{N}, +, -, 0, 1, \{\frac{\cdot}{d}\}_{d \geq 2}, <)$ where $\frac{\cdot}{d}$ is interpreted as *integer* division by d (i.e., $\frac{n}{d} = r$ if and only if $r \cdot d \leq n < (r+1) \cdot d$). Both results ultimately follow from quantifier elimination in these linear structures. Moreover, both structures inherit **RQC** and **FFTE**. Thus, we can restrict our attention to piecewise functions also in the first statement of the theorem.

From **RQC** we conclude that all considered structures are **NIP** and have PTIME 0-fitting (Theorem 1). Moreover, we can assume the formula ϕ' from Lemma 3 to be quantifier-free (recall from Example 5: the real ordered field has quantifier elimination). The sentence $\exists \vec{y} \phi'(\vec{y})$ in that lemma has size polynomial in the sample and tolerance, and has a fixed number of (existential) quantifiers. The result follows because, for both Presburger arithmetic and the real ordered field, one can solve existential formulas in PTIME when the number of variables is fixed [35, 41]. $\square$

REMARK 8 (CONNECTION TO TRAINING PROBLEMS FOR NEURAL NETWORKS). *We return to the connection with feedforward networks, now moving from classifiers (as in Remark 6) to networks without a threshold, which output a real value. As noted before, feedforward neural networks with ReLU activation functions and a single output neuron can be encoded as piecewise functions over the real ordered field. By Theorem 4, the approximate fitting problem for any fixed network of this kind is in PTIME. This result also applies to the L^2 norm and the hinge loss function (Remark 1).*

As in Remark 6, we do not claim that these results are practical, due to an exponential dependency on the number of parameters. Related work [5, 24] shows that approximate fitting for two-layer networks with a single output is in PTIME when the network is fixed, and NP-complete otherwise. When the network is not fixed and has at least two outputs, the 0-fitting problem is already $\exists\mathbb{R}$ -complete [1, 15]. Here, $\exists\mathbb{R}$ represents the complexity of the existential theory of the reals, see Definition 13 below.

6.2 Approximate fitting outside piecewise functions

The real ordered field has definable functions that are not piecewise. An example is the parameter-free square root function given by formula $\text{sqrt}(x; z) := (x < 0 \wedge z = 0) \vee (z \geq 0 \wedge z^2 = x)$. This function violates the piecewise property because the second disjunct contains a guard involving the output ($z \geq 0$), and an equality non-linear in z ($z^2 = x$ instead of $z = \ell(x)$). Showing PTIME approximate fitting for sqrt would constitute a breakthrough in computational geometry, as it would yield a PTIME algorithm for the *sum-of-square-roots problem* (SSR). Given non-negative integers $a_1, \dots, a_n$ encoded in binary, and a rational b , this problem asks if $\sum_{i=1}^n \sqrt{a_i} \leq b$. The problem is not known to be in PTIME, but the best known upper bound places it in the counting hierarchy [2]. The reduction to approximate fitting for sqrt is straightforward: simply supply the sample $(a_1, 0), \dots, (a_n, 0)$ and the tolerance b to this problem. Proposition 3 (stated below) shows that, in fact, the tolerance can be fixed to any $\epsilon > 0$.

While we conjecture the approximate fitting problem for the real ordered field to be in the counting hierarchy, at the moment the best known upper bound is given by the straightforward reduction to the existential theory of the reals.

DEFINITION 13 ($\exists\mathbb{R}$). *The existential theory of the reals $\exists\mathbb{R}$ is the closure of the satisfiability problem for existential sentences over the real ordered field under polynomial-time many-one reductions.*

The class $\exists\mathbb{R}$ is contained in PSPACE [41], but strictness of this containment is open [6].

PROPOSITION 3. *The approximate fitting problem of any definable function class over the real ordered field is in $\exists\mathbb{R}$. There are classes for which the ϵ -fitting problem is SSR-hard, for every fixed $\epsilon > 0$.*

PROOF. Membership in $\exists\mathbb{R}$ is given by the reduction mentioned above. For hardness, fix $\epsilon > 0$, and let $a_1, \dots, a_n, b$ be an instance of SSR. In polynomial time, compute rational approximations q_i of $\sqrt{a_i}$ from below, with an error within $\frac{\epsilon}{n}$. If $\sum_{i=1}^n q_i < b - \epsilon$ (resp. $\sum_{i=1}^n q_i > b$), the answer is yes (resp. no). Otherwise, decrease q_1 so that $\sum_{i=1}^n q_i = b - \epsilon$. Then, $\sum_{i=1}^n |\sqrt{a_i} - q_i| = \sum_{i=1}^n \sqrt{a_i} - b + \epsilon$, and so sqrt ϵ -fits the sample $(a_1, q_1), \dots, (a_n, q_n)$ if and only if $\sum_{i=1}^n \sqrt{a_i} \leq b$. $\square$

7 Fitting for randomized classes

TAKEAWAY. We look at hypothesis classes formed by “randomizing” a base concept class, considering distributions over the parameter space. If a concept class comes from an **RQC** structure, its randomized class has PTIME approximate fitting; the algorithm combines **RQ** querying with linear algebra.

We now turn to a second natural approach to creating a real-valued hypothesis class from logic, which works on top of an arbitrary structure. The idea, which was independently investigated in model theory [9, 33] and in data management [31], is to “randomize a hypothesis class”. We use the variant from [3, 31]: starting from a “base” hypothesis class we form a new real-valued hypothesis class by looking at probability distributions.

Randomizing a concept class. Let $\Delta(A)$ denote all probability distributions on a σ -algebra A .

DEFINITION 14 (DISTRIBUTION CLASS). Consider a concept class $C := \{f_y : y \in Y\}$ with range space X , and let Y_Σ be the σ -algebra on Y generated by the sets $P_x := \{y \in Y : f_y(x) = 1\}$ for all $x \in X$. The distribution class of C is the (real-valued) hypothesis class $\mathcal{D}_C := \{h_\mu : \mu \in \Delta(Y_\Sigma)\}$ on the same range space X , such that each h_μ maps $x \in X$ to $\mu(P_x)$.

EXAMPLE 11. We return to Example 3. Let C be the dual of the concept class of rectangles from Example 1: the range space is $\mathbb{R}^4$ (all rectangles) and the parameter space is $\mathbb{R}^2$ (points in the plane). Its distribution class is parameterized by distributions μ over a σ -algebra on $\mathbb{R}^2$. Given such a probability distribution μ , the hypothesis h_μ maps a rectangle to its μ -probability.

In the approximate fitting problem for $\mathcal{D}_C$, samples are sequences $(\vec{s}_1, t_1), \dots, (\vec{s}_n, t_n)$, where each $\vec{s}_i$ is a (representation of a) rectangle, and t_i is a target probability. Given a tolerance $\epsilon \geq 0$, we seek a distribution μ satisfying $\sum_{i=1}^n |h_\mu(\vec{s}_i) - t_i| \leq \epsilon$. $\blacktriangleleft$

We remark that [31] defines the *dual distribution class* of a concept class by taking distributions μ on the range space X , where each hypothesis h_μ maps a parameter $y \in Y$ to $\mu(\{x \in X : f_y(x) = 1\})$. For a concept class given by a partitioned formula $\phi(\vec{x}; \vec{y})$, the dual distribution class coincides with the distribution class of the formula obtained from ϕ by swapping object and parameter variables. All our results on distribution classes carry over to dual distribution classes.

7.1 Fitting problems for distribution classes

We show that when the 0-fitting problem for the underlying concept class is decidable, so is the fitting problem for its distribution class, via a reduction from the latter to the former.

PROPOSITION 4. Let C be a concept class given by a partitioned formula $\phi(\vec{x}; \vec{y})$. The approximate fitting problem for the distribution class $\mathcal{D}_C$ reduces in non-deterministic polynomial time to deciding polynomially many instances of the 0-fitting problem for C . The instance of the fitting problem for $\mathcal{D}_C$ is a yes-instance if and only if all the 0-fitting problems are yes-instances.

PROOF. Let $(\vec{s}_1, t_1), \dots, (\vec{s}_n, t_n)$ be the input sample, and ϵ be the tolerance. The approximate fitting problem for $\mathcal{D}_C$ asks for a distribution μ such that $\sum_{i=1}^n |\mu(\{\vec{y} \in Y : \phi(\vec{s}_i, \vec{y})\}) - t_i| \leq \epsilon$. All that is relevant for specifying μ is its output on the finite Boolean algebra generated by the sets $\{\vec{p} \in Y : \phi(\vec{s}_i, \vec{p})\}$ for $i \in [1..n]$, so it suffices to know the probability of formulas $\bigwedge_{i=1}^n \psi_i(\vec{s}_i, \vec{y})$ where each ψ_i is either ϕ or its negation. There are at most 2^n of these formulas. Let F be the set

of these formulas that are satisfiable. We write F_i for the formulas in F in which $\phi(\vec{s}_i, \vec{y})$ occurs positively. Introducing a variable x_ψ for every formula $\psi \in F$, the approximate fitting problem reduces to solving the following problem over the real ordered group:

$$\sum_{i=1}^n |t_i - \sum_{\psi \in F_i} x_\psi| \leq \epsilon \wedge \sum_{\psi \in F} x_\psi = 1 \wedge \bigwedge_{\psi \in F} x_\psi \geq 0. \quad (2)$$

The main observation is that Formula (2) is severely underconstrained, and so by appealing to Carathéodory's theorem [20] for convex cones, one can show that if there is a solution, then there is one where at most polynomially many variables x_ψ are non-zero. The constraint featuring absolute values is unproblematic when it comes to applying Carathéodory's theorem: it can be replaced by $\sum_{i=1}^n z_i \leq \epsilon \wedge \bigwedge_{i=1}^n (y_i = t_i - \sum_{\psi \in F_i} x_\psi \wedge y_i \leq z_i \wedge -y_i \leq z_i)$, where y_i, z_i are fresh variables.

The non-deterministic polynomial-time reduction simply guesses polynomially many satisfiable formulas of the form $\bigwedge_{i=1}^n \psi_i(\vec{s}_i, \vec{y})$ described above, and solves in polynomial time the system obtained from Formula 2 by keeping only the variables corresponding to the guessed formulas. Each formula $\bigwedge_{i=1}^n \psi_i(\vec{s}_i, \vec{y})$ corresponds to an instance of the 0-fitting problem for C . $\square$

It is easy to see that *the 0-fitting problem for the distribution class over a concept class C is at least as hard as the 0-fitting problem for C* : for samples where the associated target probabilities are in $\{0, 1\}$, if they can be fit by a randomized parameter, they can also be fit by a deterministic one. Then, the corollary below follows from Proposition 2 and Proposition 4.

COROLLARY 2. *Approximate fitting for distribution classes of formulas in Büchi arithmetic is in NP, and there is a distribution class over Büchi arithmetic with an NP-complete 0-fitting problem.*

If we assume that the concept class has finite VC dimension and PTIME 0-fitting, then its distribution class has PTIME approximate fitting.

THEOREM 5. *Let $\phi(\vec{x}; \vec{y})$ be a partitioned formula with finite VC dimension and PTIME 0-fitting. Then, the distribution class $\mathcal{D}_{C_\phi}$ has PTIME approximate fitting.*

PROOF. We use the same reduction outlined in the proof of Proposition 4. In that proof, F is the set of satisfiable conjunctions $\bigwedge_{i=1}^n \psi_i(\vec{s}_i, \vec{y})$ with $\psi_i \in \{\phi, \neg\phi\}$. Such a conjunction is satisfiable if and only if the partition of $\{\vec{s}_1, \dots, \vec{s}_n\}$ it induces is 0-fitted by some hypothesis in C_ϕ . Hence F corresponds to the realizable partitions of the sample inputs. By Lemma 2, these can be computed in polynomial time, since C_ϕ has finite VC dimension and PTIME 0-fitting. In particular, $|F|$ is polynomial by Fact 1. Thus, Formula 2 has polynomially many variables. After eliminating the absolute values as in the proof of Proposition 4, it becomes a linear program of polynomial size, solvable in polynomial time with the ellipsoid method [42]. $\square$

Theorem 5 allows us to lift Theorems 1 and 3, and Corollary 1, to distribution classes:

COROLLARY 3. *If a structure $\mathfrak{M}$ is RQC, then for any partitioned formula ϕ , the approximate fitting problem for the distribution class $\mathcal{D}_{C_\phi}$ is in PTIME. The same result holds if $\mathfrak{M}$ is an automatic structure and ϕ has finite VC dimension.*

7.2 Randomizing a definable function class

Thus far, we have considered two ways of creating logic-based real-valued hypothesis classes: via formulas over a numerical structure that define functions, and by randomizing an arbitrary logically-defined hypothesis class. It is possible to combine these approaches. Starting from a definable function class over a numerical structure such as the reals, we “randomize the function class” to form a new real-valued class parameterized by distributions over the parameter space. Each distribution induces a hypothesis by taking the expected value of the underlying function over the parameters. We formalize this construction below.

A real-valued hypothesis class $\mathcal{H} := \{f_y : y \in Y\}$ with range space X is said to be *bounded* whenever, for every $x \in X$, the set $\{f_y(x) : y \in Y\}$ is included in an interval $[a, b]$ of $\mathbb{R}$.

DEFINITION 15 (EXPECTATION CLASS). Consider a bounded hypothesis class $\mathcal{H} := \{f_y : y \in Y\}$ with range space X , and the σ -algebra Y_Σ generated by the sets $\{y \in Y : f_y(x) \leq r\}$ for all $x \in X$ and $r \in \mathbb{R}$. The expectation class of $\mathcal{H}$ is the (real-valued) hypothesis class $\mathcal{E}_\mathcal{H} := \{h_\mu : \mu \in \Delta(Y_\Sigma)\}$ on the same range space X , such that each h_μ maps $x \in X$ to the expectation $\mathbb{E}_{y \sim \mu}[f_y(x)] := \int_Y f_y(x) d\mu(y)$.

Note that the hypothesis that $\mathcal{H}$ is bounded is both a necessary and sufficient condition for the expectation $\mathbb{E}_{y \sim \mu}[f_y(x)]$ to be well-defined (i.e., for every $x \in X$, the map $y \mapsto f_y(x)$ is μ -integrable) on all probability distributions $\mu \in \Delta(Y_\Sigma)$.

We can show that fitting problems are decidable for expectation classes that are built over definable function classes over classical numerical structures:

THEOREM 6. *The approximate fitting problem is in NP for every expectation class over a bounded function class over the real ordered group and over Presburger arithmetic. For expectation classes based on the real ordered field, the problem is in $\exists\mathbb{R}$.*

To prove this theorem, we first reformulate the approximate fitting problem for expectation classes to replace continuous integration with linear algebra.

LEMMA 4. *Let $\mathcal{H} := \{f_y : y \in Y\}$ be a bounded hypothesis class with range space X , $S := (s_1, t_1), \dots, (s_n, t_n)$ be a sample of pairs from $X \times \mathbb{R}$, and ϵ be a tolerance. For $i \in [1..n]$, let f_i denote the map $y \mapsto f_y(s_i)$. Then, $\mathcal{E}_\mathcal{H}$ ϵ -fits S if and only if*

$$\sum_{i=1}^n |t_i - \sum_{j=1}^{n+1} p_j \cdot f_i(y_j)| \leq \epsilon \wedge \sum_{i=1}^{n+1} p_i = 1, \quad (3)$$

for some $y_1, \dots, y_{n+1} \in Y$ and non-negative $p_1, \dots, p_{n+1} \in \mathbb{R}$.

PROOF. By definition, $\mathcal{E}_\mathcal{H}$ ϵ -fits S iff $|\sum_{i=1}^n |t_i - \int_Y f_i(y) d\mu(y)| \leq \epsilon$ for some distribution $\mu \in \Delta(Y_\Sigma)$.

Consider a distribution $\mu \in \Delta(Y_\Sigma)$. Since $\mathcal{H}$ is bounded, every f_i is μ -integrable. We now rely on the following mean value theorem for integrals:

FACT 2 ([32]). *Let $f_1, \dots, f_n : Y \rightarrow \mathbb{R}$ be μ -integrable functions. Then, the convex hull of the set $\{(f_1(y), \dots, f_n(y)) : y \in Y\} \subseteq \mathbb{R}^n$ contains the vector $(\int_Y f_1(y) d\mu(y), \dots, \int_Y f_n(y) d\mu(y))$.*

From Carathéodory's theorem [20], any vector $(v_1, \dots, v_n) \in \mathbb{R}^n$ in the convex hull of the set $C := \{(f_1(y), \dots, f_n(y)) : y \in Y\}$ can be expressed as a convex combination of at most $n + 1$ elements from C , that is, $(v_1, \dots, v_n) = \sum_{i=1}^{n+1} p_i \cdot (f_1(y_i), \dots, f_n(y_i))$, for some $y_1, \dots, y_{n+1} \in Y$ and non-negative $p_1, \dots, p_{n+1} \in \mathbb{R}$ with $\sum_{i=1}^{n+1} p_i = 1$. Observe that this means that the converse of Fact 2 also holds: there is a probability distribution μ^* such that

$$(v_1, \dots, v_n) = (\int_Y f_1(y) d\mu^*(y), \dots, \int_Y f_n(y) d\mu^*(y)).$$

It is the distribution $\mu^* := \sum_{i=1}^{n+1} p_i \delta_{y_i}$, where δ_y is the Dirac measure at y ; that is, $\mu^*(B) = \sum_{i: y_i \in B} p_i$ for every $B \in Y_\Sigma$. By Fact 2, we conclude that the fitting problem reduces to solving System 3. $\square$

PROOF OF THEOREM 6. The theorem refers to three structures, and we handle each separately.

CASE: REAL ORDERED FIELD. In this case, one simply encodes System 3 as an existential formula.

CASE: PRESBURGER ARITHMETIC. The NP bound for *Presburger arithmetic* relies on the fact that bounded function families on the naturals have finite range, so they are not far from concept classes.

Let $\mathcal{H}$ be a bounded function class definable in Presburger arithmetic. Recall that we are looking at the approximate fitting problem for $\mathcal{E}_\mathcal{H}$. Since $\mathcal{H}$ is bounded, the image $\text{Im}(f_i) \subseteq \mathbb{N}$ of every function in System 3 is finite. Note that f_i only depends on s_i , and by applying well-known bounds

on the solutions to integer linear programs (see [42, Chapter 17]) one can show that the bit length of every natural in $\text{Im}(f_i)$ is polynomial in the bit length of s_i .

The NP algorithm guesses a polynomial-size natural k_{ij} for each term $f_i(y_j)$ occurring in System 3. For each $j \in [1..n+1]$, the algorithm solves the 0-fitting problem for $\mathcal{H}$ (in PTIME, by Theorem 4) with input the sample $(s_1, k_{1j}), \dots, (s_n, k_{nj})$. This ensures the existence of a $y_j \in Y$ such that $f_i(y_j) = k_{ij}$ for all $i \in [1..n]$. Afterwards, the algorithm updates System 3, by replacing each $f_i(y_j)$ by k_{ij} , and removing the absolute value as explained in the proof of Proposition 4. The result is a linear program that can be solved in polynomial time.

CASE: REAL ORDERED GROUP. For obtaining an NP bound for the *real ordered group*, the main challenge lies in handling the non-linear terms $p_j \cdot f_i(y_j)$ from System 3, given that $\text{Im}(f_i)$ can be infinite. Fix a partitioned formula $\phi(\vec{x}; z; \vec{y})$ from the real ordered group. We consider the approximate fitting problem for $\mathcal{E}_{\mathcal{H}_\phi}$. Every function definable in the real ordered group is known to be piecewise, and in fact, wlog., we can assume $\phi(\vec{x}; z; \vec{y}) := \bigvee_{i=1}^m (\psi_i(\vec{x}, \vec{y}) \wedge z = \ell_i(\vec{x}, \vec{y}))$ to be such that, for $i \in [1..m]$, $\psi_i(\vec{x}; \vec{y}) := A_i \cdot \begin{bmatrix} \vec{x} \\ \vec{y} \end{bmatrix} \leq \vec{b}_i \wedge C_i \cdot \begin{bmatrix} \vec{x} \\ \vec{y} \end{bmatrix} < \vec{d}_i$, where A_i, C_i are rational matrices, and $\vec{b}_i, \vec{d}_i$ rational vectors, and each ℓ_i is a linear polynomial with rational coefficients.

Let $(\vec{s}_1, t_1), \dots, (\vec{s}_n, t_n)$ be a sample and ϵ a tolerance. In System 3, wlog. we can assume $p_j > 0$ for every $j \in [1..n+1]$. Indeed, consider a solution $\vec{y}_1^*, \dots, \vec{y}_{n+1}^*$ and $p_1^*, \dots, p_{n+1}^*$ to System 3, and suppose $p_j^* = 0$ for some j . Pick an index r with $p_r^* > 0$. Replace every $\vec{y}_j^*$ for which $p_j^* = 0$ by $\vec{y}_r^*$, and split the weight p_r^* equally among the index r and these indices j . This is still a solution, as we have not changed the sums $\sum_{j=1}^{n+1} p_j^* \cdot f_i(\vec{y}_j^*)$.

By guessing an index $k_{ij} \in [1..m]$ for all $(i, j) \in [1..n] \times [1..n+1]$, the algorithm non-deterministically rewrites System 3 as

$$\begin{aligned} \sum_{i=1}^n |t_i - \sum_{j=1}^{n+1} p_j \cdot \ell_{k_{ij}}(\vec{s}_i, \vec{y}_j)| &\leq \epsilon \\ \sum_{j=1}^{n+1} p_j &= 1 \\ p_j &> 0 & j \in [1..n+1] \\ \psi_{k_{ij}}(\vec{s}_i, \vec{y}_j) & & i \in [1..n], j \in [1..n+1] \end{aligned} \quad (4)$$

I.e., the algorithm guesses which formula among $\psi_1, \dots, \psi_m$ the parameter $\vec{y}_j$ satisfies, for every object $\vec{s}_i$ in the sample. System 4 is still non-linear.

We linearize System 4 by a change of variables: for every $j \in [1..n+1]$, we replace the vector $p_j \cdot \vec{y}_j$ by a vector of fresh variables $\vec{w}_j$. Observe that the variables p_j still appear in the system. However, since $p_j > 0$, this change of variables is invertible, with $\vec{y}_j = \vec{w}_j / p_j$. For $k \in [1..m]$, write $\ell_k(\vec{x}, \vec{y}) = \vec{\alpha}_k \cdot \vec{x} + \vec{\beta}_k \cdot \vec{y} + \gamma_k$. Then $p_j \cdot \ell_k(\vec{s}_i, \vec{y}_j) = p_j \cdot (\vec{\alpha}_k \cdot \vec{s}_i + \gamma_k) + \vec{\beta}_k \cdot \vec{w}_j$. Moreover, multiplying the inequalities of ψ_k by $p_j > 0$, we see that $\psi_k(\vec{s}_i, \vec{y}_j)$ holds if and only if $A_k \cdot \begin{pmatrix} p_j \cdot \vec{s}_i \\ \vec{w}_j \end{pmatrix} \leq p_j \cdot \vec{b}_k$ and $C_k \cdot \begin{pmatrix} p_j \cdot \vec{s}_i \\ \vec{w}_j \end{pmatrix} < p_j \cdot \vec{d}_k$. Hence, System 4 is feasible if and only if the following system is feasible:

$$\begin{aligned} \sum_{i=1}^n \left| t_i - \sum_{j=1}^{n+1} (p_j \cdot (\vec{\alpha}_{k_{ij}} \cdot \vec{s}_i + \gamma_{k_{ij}}) + \vec{\beta}_{k_{ij}} \cdot \vec{w}_j) \right| &\leq \epsilon \\ \sum_{j=1}^{n+1} p_j &= 1 \\ p_j &> 0 & j \in [1..n+1] \\ A_{k_{ij}} \cdot \begin{pmatrix} p_j \cdot \vec{s}_i \\ \vec{w}_j \end{pmatrix} &\leq p_j \cdot \vec{b}_{k_{ij}} \wedge C_{k_{ij}} \cdot \begin{pmatrix} p_j \cdot \vec{s}_i \\ \vec{w}_j \end{pmatrix} < p_j \cdot \vec{d}_{k_{ij}} & i \in [1..n], j \in [1..n+1] \end{aligned} \quad (5)$$

This is a linear system, in the variables p_j and $\vec{w}_j$ (the absolute value can be removed as in the proof of Proposition 4), and hence it is solvable in polynomial time. $\square$

Concept Class	Exact fitting	Approximate fitting
RQC structure (Presburger, real field, ...)	RQ (Corollary 1)	PTIME (Corollary 1)
Büchi arithmetic, finite VC dimension	PTIME (Theorem 3)	PTIME (Theorem 3)
Büchi arithmetic, general	NP , tight (Proposition 2)	NP , tight (Proposition 2)
Definable Function Class		
over $(\mathbb{R}, +, <)$ or $(\mathbb{N}, +, <)$	RQ (Remark 7)	PTIME (Theorem 4)
Real ordered field, piecewise	RQ (Remark 7)	PTIME (Theorem 4)
Real ordered field, general	RQ (Remark 7)	$\exists \mathbb{R}$, SSR-hard (Prop. 3)
Distribution Class		
RQC structure	PTIME (Corollary 3)	
Büchi arithmetic, finite VC dimension	PTIME (Corollary 3)	
Büchi arithmetic, general	NP , tight (Corollary 2)	
Expectation Class		
over $(\mathbb{R}, +, <)$ or $(\mathbb{N}, +, <)$	NP (Theorem 6)	
Real ordered field	$\exists \mathbb{R}$ (Theorem 6)	

Table 1. Logic-based fitting problems (assuming effective constants and **FFTE**): a recap.

8 Discussion

Our work initiates a study of fitting problems for logic-based classes over infinite structures. Table 1 summarizes some of our results. We highlight that, for several natural examples, exact fitting can be performed with a first-order query over the active domain of the sample—**RQ** entries in the table—while approximate fitting reduces to generating polynomially many such queries. For real-valued classes, we show that fitting is decidable for many natural classes; the algorithms often involve equation solving in addition to querying, and tractability remains open in some cases.

For general hypothesis classes, it is known that “well-behaved sample complexity” (characterized by finite VC dimension) and “well-behaved computational complexity of fitting” are unrelated (Remark 3). However, our exploration of fitting problems reveals that these two are linked when focusing on *logically-defined classes*. Indeed, for these classes we know no counterexample to the question “does every **NIP** structure with **FFTE** have **PTIME** fitting?”, and we proved that any such counterexample would need to come from a structure that is neither **RQC** nor automatic. This is perhaps the major question we leave open. For an affirmative answer, it would suffice to show that every **NIP** structure with **FFTE** can be expanded to have **RQC** while retaining **FFTE**.

In Table 1, all results involving the existential theory of the reals are *not* known to be tight. These problems are syntactically restricted instances of the existential theory of the reals. The reduction from the sum-of-square-roots problem to fitting for definable functions in the real field indicates that one should not hope to prove a better bound for this problem than a fixed level of the counting hierarchy [2]. The **NP** upper bounds for expectation classes over the real ordered group and Presburger arithmetic from Section 7 are also not known to be tight.

All of our complexity results use standard bit complexity, assuming effective constants for the samples. It would be possible to restate some results using the abstract Blum–Shub–Smale model of computation [16, 40], but in many cases this would weaken the statements (e.g., for **NP** bounds).

In this work, we have only briefly explored the connection to neural network fitting problems (e.g., [1, 5, 15, 24, 29]). In the extended version [14], we show that all our upper bounds extend to both the L^2 norm and the hinge loss function. A future direction is to examine approximate fitting under other loss functions. Additionally, we have focused solely on *decision problems* in this work, leaving open the task of finding a witness parameter. Addressing this search problem will require further restrictions on the computation model to specify how parameters are represented.

References

- [1] M. Abrahamsen, L. Kleist, and T. Miltzow. Training Neural Networks is $\exists\mathbb{R}$ -Complete. In *NeurIPS*, 2021.
- [2] E. Allender, P. Bürgisser, J. Kjeldgaard-Pedersen, and P. B. Miltersen. On the complexity of numerical analysis. *SIAM J. Comput.*, 38(5):1987–2006, 2009.
- [3] A. Anderson and M. Benedikt. From learnable objects to learnable random objects. *JMLR*, 2026.
- [4] M. Anthony. Some connections between learning and optimization. *Discrete Applied Mathematics*, 144(1):17–26, 2004.
- [5] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee. Understanding deep neural networks with rectified linear units. In *ICLR*, 2018.
- [6] S. Basu, R. Pollack, and M.-F. Roy. Existential theory of the reals. In *Algorithms in Real Algebraic Geometry*, pages 465–492. Springer, 2003.
- [7] O. V. Belegardek, A. P. Stolboushkin, and M. A. Taitlin. Extended order-generic queries. *Annals of Pure and Applied Logic*, 97(1):85–125, 1999.
- [8] I. Ben Yaacov. On theories of random variables. *Israel Journal of Mathematics*, 194:957–1012, 2013.
- [9] I. Ben Yaacov and H. J. Keisler. Randomizations of models as metric structures. *Confluentes Mathematici*, 1(2):197–223, 2009.
- [10] M. Benedikt. Generalizing finite model theory. In *Logic Colloquium '03*, Lecture Notes in Logic, pages 3–24. Cambridge University Press, 2006.
- [11] M. Benedikt and E. Hrushovski. Embedded finite models beyond restricted quantifier collapse. In *LICS*, 2023.
- [12] M. Benedikt and L. Libkin. Relational queries over interpreted structures. *J. ACM*, 47(4):644–680, July 2000.
- [13] M. Benedikt, L. Libkin, T. Schwentick, and L. Segoufin. Definable relations and first-order query languages over strings. *J. ACM*, 50(5):694–751, 2003.
- [14] M. Benedikt and A. Mansutti. How (and when) can you fit examples to logic-based hypothesis classes over infinite structures? *CoRR*, abs/2606.01107, 2026.
- [15] D. Bertschinger, C. Hertrich, P. Jungeblut, T. Miltzow, and S. Weber. Training Fully Connected Neural Networks is $\exists\mathbb{R}$ -Complete. In *NeurIPS*, 2023.
- [16] 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):1–46, 1989.
- [17] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik–Chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989.
- [18] V. Bruyère, G. Hansel, C. Michaux, and R. Villemaire. Logic and p -recognizable sets of integers. *Bull. Belg. Math. Soc. Simon Stevin*, 1(2):191–238, 1994.
- [19] J. R. Büchi. Weak second-order arithmetic and finite automata. *Math. Logic Quart.*, 6(1-6):66–92, 1960.
- [20] C. Carathéodory. Über den Variabilitätsbereich der Koeffizienten von Potenzreihen, die gegebene Werte nicht annehmen. *Mathematische Annalen*, 64(1):95–115, 1907.
- [21] A. Chernikov and P. Simon. Externally definable sets and dependent pairs II. *Transactions of the American Mathematical Society*, 367(7):5217–5235, 2015.
- [22] D. Figueira, A. Jez, and A. W. Lin. Data path queries over embedded graph databases. In *PODS*, 2022.
- [23] J. Flum and M. Ziegler. Pseudo-finite homogeneity and saturation. *The Journal of Symbolic Logic*, 64(4):1689–1699, 1999.
- [24] V. Froese and C. Hertrich. Training neural networks is NP-hard in fixed dimension. In *NeurIPS*, 2023.
- [25] M. L. Furst, J. B. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Math. Syst. Theory*, 17(1):13–27, 1984.
- [26] S. Goel, A. R. Klivans, P. Manurangsi, and D. Reichman. Tight Hardness Results for Training Depth-2 ReLU Networks. In *ITCS*, 2021.
- [27] E. Grädel. Automatic structures: Twenty years later. In *LICS*, 2020.
- [28] M. Grohe and M. Ritzert. Learning first-order definable concepts over structures of small degree. In *LICS*, 2017.
- [29] T. Hankala, M. Hannula, J. Kontinen, and J. Virtema. Complexity of neural network training and ETR: extensions with effectively continuous functions. In *AAAI*, 2024.
- [30] P. Hieronymi, T. Nell, and E. Walsberg. Wild theories with o-minimal open core. *Annals of Pure and Applied Logic*, 169(2):146–163, 2018.
- [31] X. Hu, Y. Liu, H. Xiu, P. K. Agarwal, D. Panigrahi, S. Roy, and J. Yang. Selectivity functions of range queries are learnable. In *SIGMOD*, 2022.
- [32] S. Janković and M. Merkle. A mean value theorem for systems of integrals. *Journal of Mathematical Analysis and Applications*, 342(1):334–339, 2008.
- [33] H. J. Keisler. Randomizing a model. *Advances in Mathematics*, 143(1):124–158, 1999.
- [34] Q. Lambotte and F. Point. On expansions of $(\mathbb{Z}, +, 0)$. *Annals of Pure and Applied Logic*, 171(8):102809, 2020.
- [35] H. W. Lenstra. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4):538–548, 1983.

- [36] L. Libkin. Embedded finite models and constraint databases. In *Finite Model Theory and Its Applications*, Texts in Theoretical Computer Science. An EATCS Series, pages 257–337. Springer, 2007.
- [37] D. Olteanu and M. Schleich. F: regression models over factorized views. *Proc. VLDB Endow.*, 9(13):1573–1576, 2016.
- [38] J. Paredaens, J. Van den Bussche, and D. Van Gucht. First-order queries on finite structures over the reals. *SIAM J. Comput.*, 27(6):1747–1763, 1998.
- [39] L. Pitt and L. G. Valiant. Computational limitations on learning from examples. *J. ACM*, 35(4):965–984, 1988.
- [40] B. Poizat. *Les Petits Cailloux: Une approche modèle-théorique de l’algorithmie*. Aléas, Lyon, 1995.
- [41] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals, Part I: Introduction. Preliminaries. The geometry of semi-algebraic sets. The decision problem for the existential theory of the reals. *J. Symb. Comput.*, 13(3):255–299, 1992.
- [42] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, 1986.
- [43] S. Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41:247–261, 1972.
- [44] S. Shelah and P. Simon. Adding linear orders. *The Journal of Symbolic Logic*, 77(2):717–725, 2012.
- [45] P. Simon. *A Guide to NIP Theories*. Cambridge University Press, 2015.
- [46] B. ten Cate, M. Funk, J. C. Jung, and C. Lutz. Fitting algorithms for conjunctive queries. *SIGMOD Rec.*, 52(4):6–18, 2023.

Received 30 May 2026; revised 25 August 2026; accepted 1 September 2026