Approximately Counting Answers to Conjunctive Queries with Disequalities and Negations

Jacob Focke*
CISPA
Helmholtz Center for Information Security
Saarbrücken, Germany

Leslie Ann Goldberg*
Department of Computer Science,
University of Oxford
Oxford, U.K.

Marc Roth*
Department of Computer Science and Merton College,
University of Oxford
Oxford, U.K.

Stanislav Živný*
Department of Computer Science,
University of Oxford
Oxford, U.K.

ABSTRACT

We study the complexity of approximating the number of answers to a small query φ in a large database $\mathcal{D}$. We establish an exhaustive classification into tractable and intractable cases if φ is a conjunctive query possibly including disequalities and negations:

- If there is a constant bound on the arity of φ, and if the randomised Exponential Time Hypothesis (rETH) holds, then the problem has a fixed-parameter tractable approximation scheme (FPTRAS) if and only if the treewidth of φ is bounded.
- If the arity is unbounded and φ does not have negations, then the problem has an FPTRAS if and only if the adaptive width of φ (a width measure strictly more general than treewidth) is bounded; the lower bound relies on the rETH as well.

Additionally we show that our results cannot be strengthened to achieve a fully polynomial randomised approximation scheme (FPRAS): We observe that, unless NP = RP, there is no FPRAS even if the treewidth (and the adaptive width) is 1.

However, if there are neither disequalities nor negations, we prove the existence of an FPRAS for queries of bounded fractional hypertreewidth, strictly generalising the recently established FPRAS for conjunctive queries with bounded hypertreewidth due to Arenas, Croquevielle, Jayaram and Riveros (STOC 2021).

CCS CONCEPTS

• **Theory of computation → Design and analysis of algorithms**;
• **Information systems → Relational database query languages**.

*The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 714532). The research was also supported by the European Research Council (ERC) consolidator grant No 725978 SYSTEMATICGRAPH. Stanislav Živný was supported by a Royal Society University Research Fellowship. The paper reflects only the authors' views and not the views of the ERC or the European Commission. The European Union is not liable for any use that may be made of the information contained therein. A full version of this work can be found at [22].

PODS '22, June 12–17, 2022, Philadelphia, PA, USA

ACM ISBN 978-1-4503-9260-0/22/06…$15.00
https://doi.org/10.1145/3517804.3526231

KEYWORDS

approximate counting, conjunctive queries, fully polynomial randomised approximation scheme (FPRAS), fixed-parameter tractable randomised approximation scheme (FPTRAS)

ACM Reference Format:
Jacob Focke, Leslie Ann Goldberg, Marc Roth, and Stanislav Živný. 2022. Approximately Counting Answers to Conjunctive Queries with Disequalities and Negations. In *Proceedings of the 41st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS '22), June 12–17, 2022, Philadelphia, PA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3517804.3526231

1 INTRODUCTION

The evaluation of conjunctive queries is amongst the most central and well-studied problems in database theory [1, 5, 9, 24]. These queries are also called *select-project-join queries* in relational algebra and *select-from-where* queries in SQL. In this work, we study the *counting* problem associated with conjunctive queries and with extensions to conjunctive queries allowing negations, disequalities, and unions of queries. Given a query φ and a database $\mathcal{D}$, an "answer" of φ in $\mathcal{D}$ is an assignment of values from the universe of $\mathcal{D}$ to the free variables of φ that can be extended (by also assigning values to the existential variables of φ) to an assignment satisfying φ. For example, the universe of the database $\mathcal{D}$ could be a set of people U, and $\mathcal{D}$ has an entry $F(a, b)$ whenever two people $a, b \in U$ are "friends". Then an answer to the query

$$\varphi(x) = \exists y \exists z \, F(x, y) \wedge F(x, z) \wedge (y \neq z) \tag{1}$$

is a person that has at least two friends (from the people in U).

The counting problem is to compute the number of answers of φ in $\mathcal{D}$. Our goal is to determine the parameterised complexity of this counting problem in the situation where the query φ is significantly smaller than the database $\mathcal{D}$; a formal exposition is given in Section 2.

Previous work [10, 16, 17] established that the problem of exactly counting answers to conjunctive queries is *extremely* difficult: Even very simple queries, such as acyclic conjunctive queries, which can be evaluated in polynomial time [23, 44], are sufficiently powerful to encode intractable problems in their counting versions, making any non-trivial improvement over the brute-force algorithm impossible under the Strong Exponential Time Hypothesis [16].

Therefore, the relaxation to approximate counting is necessary if efficient algorithms are sought. In this work, we quantify the complexity of approximating the number of answers to conjunctive queries with negations and disequalities, and unions thereof, in terms of several natural width measures of the queries, such as treewidth, fractional hypertreewidth, and adaptive width. This leads to a complete classification (and a new approximation algorithm) in the bounded-arity case, to a complete classification (and another new approximation algorithm) in the unbounded-arity case when negations are excluded, and to a new FPRAS in the unparameterised setting. The formal setup, including the definitions of the problems and the approximation schemes, are introduced in Section 2 and we present our results in Section 3.

2 TECHNICAL BACKGROUND

A *signature* σ consists of a finite set of relation symbols with specified positive arities. A *(relational) database* $\mathcal{D}$ with signature $\mathrm{sig}(\mathcal{D})$ consists of a finite *universe* $U(\mathcal{D})$[1] together with, for each relation symbol $R \in \mathrm{sig}(\mathcal{D})$, a relation $R^{\mathcal{D}}$ over the universe $U(\mathcal{D})$ with the same arity that $\mathrm{sig}(\mathcal{D})$ specifies for R. The tuples in the relations $R^{\mathcal{D}}$ are called the *facts* of $\mathcal{D}$. A *conjunctive query* (CQ) φ with signature $\mathrm{sig}(\varphi)$ is a formula of the form

$$\varphi(x_1, \ldots, x_\ell) = \exists x_{\ell+1} \cdots \exists x_{\ell+k} \psi(x_1, \ldots, x_{\ell+k}),$$

where $\mathrm{vars}(\varphi)$ denotes the set of variables $\{x_1, \ldots, x_{k+\ell}\}$ of φ, $\mathrm{free}(\varphi)$ denotes the set of free (output) variables $\{x_1, \ldots, x_\ell\}$ of φ, and ψ is a conjunction of a finite number of atoms, which are predicates of the form $R(y_1, \ldots, y_j)$, where R is an arity-j relation symbol in $\mathrm{sig}(\varphi)$ and each y_i is a variable in $\mathrm{vars}(\varphi)$. Each element of $\mathrm{sig}(\varphi)$ appears in at least one predicate. Also, each variable in $\mathrm{vars}(\varphi)$ appears in at least one atom.

In an *extended* conjunctive query (ECQ) there are three more types of allowable atoms.

- Equality: $y_i = y_j$.
- Disequality: $y_i \neq y_j$.
- Negated predicate: $\neg R(y_1, \ldots, y_j)$, where R is an arity-j relation symbol in $\mathrm{sig}(\varphi)$ and each y_i is a variable in $\mathrm{vars}(\varphi)$.

Again, each element of $\mathrm{sig}(\varphi)$ appears at least once in φ (as a predicate, as a negated predicate, or both).

In fact, without loss of generality we can assume that φ has no equalities, since we can re-write φ to avoid these by replacing equal variables with a single variable. Thus, from now on, we assume that ECQs do not have equalities.

It is natural to extend conjunctive queries by adding disequalities and negations. Such extended queries were studied for instance in [4, 11, 27, 31, 32, 38]. Sometimes we will be interested in extending CQs by adding disequalities but not negations. We refer to these partially-extended queries as DCQs.

Consider an ECQ φ. The following notation of "Solution" captures the assignments (of elements in $U(\mathcal{D})$ to the variables of φ) that satisfy φ. It does not distinguish between existential and free variables, but we do that later in Definition 2.

Definition 1. (solution, $\mathsf{Sol}(\varphi, \mathcal{D})$) Let φ be an ECQ and let $\mathcal{D}$ be a database with $\mathrm{sig}(\varphi) \subseteq \mathrm{sig}(\mathcal{D})$. A *solution* of $(\varphi, \mathcal{D})$ is an assignment $\alpha\colon \mathrm{vars}(\varphi) \to U(\mathcal{D})$ which has the following property.

- For every predicate $R(y_1, \ldots, y_j)$ of φ, we have that the tuple $(\alpha(y_1), \ldots, \alpha(y_j))$ is in $R^{\mathcal{D}}$,
- For every negated predicate $\neg R(y_1, \ldots, y_j)$ of φ, the tuple $(\alpha(y_1), \ldots, \alpha(y_j))$ is not in $R^{\mathcal{D}}$, and
- For every disequality $y_i \neq y_j$ of φ we have $\alpha(y_i) \neq \alpha(y_j)$.

We use $\mathsf{Sol}(\varphi, \mathcal{D})$ to denote the set of solutions of $(\varphi, \mathcal{D})$.

In this work, we will not be interested so much in the solutions of $(\varphi, \mathcal{D})$ but rather in their projections onto the free (output) variables of φ.

Definition 2. (proj, answer, $\mathsf{Ans}(\varphi, \mathcal{D})$) Let φ be an ECQ and let $\mathcal{D}$ be a database with $\mathrm{sig}(\varphi) \subseteq \mathrm{sig}(\mathcal{D})$. Let $\alpha\colon \mathrm{vars}(\varphi) \to U(\mathcal{D})$ be an assignment of elements in $U(\mathcal{D})$ to the variables of φ. We use $\mathrm{proj}(\alpha, \mathrm{free}(\varphi))$ to denote α's projection onto the free variables of φ. That is, $\mathrm{proj}(\alpha, \mathrm{free}(\varphi))$ is the assignment from $\mathrm{free}(\varphi)$ to $U(\mathcal{D})$ that agrees with α. An *answer* of $(\varphi, \mathcal{D})$ is an assignment $\tau\colon \mathrm{free}(\varphi) \to U(\mathcal{D})$ of elements in $U(\mathcal{D})$ to the free variables of φ which can be extended to a solution in the sense that there is a solution α of $(\varphi, \mathcal{D})$ with $\mathrm{proj}(\alpha, \mathrm{free}(\varphi)) = \tau$. We write $\mathsf{Ans}(\varphi, \mathcal{D})$ for the set of all answers of $(\varphi, \mathcal{D})$.

Our main focus is on the problem of approximately counting answers to extended conjunctive queries φ, parameterised[2] by the size of φ, which is denoted by $\|\varphi\|$, and is defined to be the sum of $|\mathrm{vars}(\varphi)|$ and the sum of the arities of the atoms in φ.

The formal problem definition is as follows. Let Φ be a class of ECQs.

Name: #ECQ (Φ)
Input: An ECQ $\varphi \in \Phi$ and a database $\mathcal{D}$ with $\mathrm{sig}(\varphi) \subseteq \mathrm{sig}(\mathcal{D})$.
Parameter: $\|\varphi\|$.
Output: $|\mathsf{Ans}(\varphi, \mathcal{D})|$.

We define the problems #CQ (Φ) and #DCQ (Φ) analogously, by requiring the input to be a CQ (in the case of #CQ (Φ)) or a DCQ (in the case of #DCQ (Φ)).

The size of the encoding of the input pair $(\varphi, \mathcal{D})$ is taken to be the sum of $\|\varphi\|$ and the size of the encoding of $\mathcal{D}$ (written $\|\mathcal{D}\|$) which is defined to be $|\mathrm{sig}(\mathcal{D})| + |U(\mathcal{D})|$ plus the sum of the lengths of the tuples in the relations of $\mathcal{D}$.

Note that singleton unary relations in $\mathcal{D}$ can be used to implement "constants" in φ. To see this, for any $v \in U(\mathcal{D})$ let $R_v^{\mathcal{D}}$ denote the relation $\{v\}$. It is possible to refer to the constant v in φ by including R_v in $\mathrm{sig}(\varphi)$ and constraining some variable $x \in \mathrm{vars}(\varphi)$ with the predicate $R_v(x)$. Of course the size $\|\varphi\|$ increases by an additional constant amount by the introduction of the variable x and the predicate $R_v(x)$. Adding all singleton unary relations to the signature of $\mathcal{D}$ does not increase $\|\mathcal{D}\|$ significantly, since $|U(\mathcal{D})|$ is already included in $\|\mathcal{D}\|$.

Randomised Approximation Schemes and Fixed-Parameter Tractability. Given a value V and $\varepsilon, \delta \in (0, 1)$, an (ε, δ)-*approximation* of V is a random variable X that satisfies $\Pr(|X - V| \leq \varepsilon V) \geq 1 - \delta$.

Let $\#A$ be a counting problem that, when given input x, asks for the value $V(x)$. Slightly overloading notation, an (ε, δ)-*approximation*

[1] $U(\mathcal{D})$ is also often referred to as the "domain" of $\mathcal{D}$.

[2] This choice of the parameter allows one to produce fine-grained complexity results which are appropriate for instances in which the query size is (significantly) smaller than the size of the database (see for example the section "Why Fixed-Parameter Tractability" in [36, Section 1] for a longer discussion of this point). The notion of fixed-parameter tractability is made formal below.

for $\#A$ is a randomised algorithm that, given an input x to $\#A$, outputs an (ε, δ)-approximation of $V(x)$. A *fully polynomial randomised approximation scheme* (FPRAS) for $\#A$ is an (ε, δ)-approximation that runs in time polynomial in $\|x\|$, $1/\varepsilon$, and $\log(1/\delta)$.

Suppose that the counting problem $\#A$ is parameterised by a parameter k (as the problem #ECQ (Φ) is parametetrised by $\|\varphi\|$). A randomised approximation scheme for $\#A$ is a *fixed-parameter tractable randomised approximation scheme* (FPTRAS) if there is a function $f\colon \mathbb{R} \to \mathbb{R}$ and a polynomial p such that the running time is at most $f(k) \cdot p(\|x\|, 1/\varepsilon, \log(1/\delta))$.

Applying this definition, note that an FPTRAS for #ECQ (Φ) has a running time bound of $f(\|\varphi\|) \cdot p(\|\mathcal{D}\|, 1/\varepsilon, \log(1/\delta))$. In other words, relative to the definition of FPRAS, the definition of FPTRAS relaxes the condition that the algorithm must run in polynomial time by allowing a super-polynomial factor in the size of the query. Since the query is assumed to be significantly smaller than the database, this is a very natural notion of an efficient algorithm. Indeed, we will show that all FPTRASes for #ECQ (Φ) constructed in this work cannot be upgraded to FPRASes (subject to natural complexity hypotheses). The reason that they cannot be upgraded is that, even for very restricted query classes Φ, there are reductions from NP-hard problems to the problem of producing an FPRAS for #ECQ (Φ).

Considering #ECQ (Φ) as a parameterised problem allows one to interpolate between the classical complexity of the problem, in which no assumptions are made regarding the size of the input query, and its *data complexity*, in which the input query is fixed.

From the viewpoint of data complexity, there is a brute-force polynomial-time algorithm for counting answers to a query, by iterating through all assignments of the variables, roughly in time $\|\mathcal{D}\|^{\mathrm{O}(\|\varphi\|)}$. If the query φ is fixed, the running time of this brute-force algorithm is bounded by a polynomial in the input size. In the fixed-parameter setting the goal is instead to separate the (potentially exponential) running time in the query size from the (polynomial) running time, in the size of the database.

3 OUR RESULTS

In order to give an overview of our results, we provide an illustration in Figure 1.

Bounded-Treewidth ECQs. The tractability criteria in our results will depend on the hypergraph associated with a conjunctive query (Definition 3, see also [24]). A *hypergraph* H consists of a (finite) set of vertices $V(H)$ and a set $E(H) \subseteq 2^{V(H)}$ of non-empty hyperedges. The *arity* of a hypergraph is the maximum size of its hyperedges.

Definition 3. $(H(\varphi), \Phi_C)$ Given an ECQ φ, the *hypergraph of* φ, denoted $H(\varphi)$, has vertex set $V(H(\varphi)) = \mathrm{vars}(\varphi)$. For each predicate of φ there is a hyperedge in $E(H(\varphi))$ containing the variables appearing in the predicate. For each negated predicate of φ, there is a hyperedge in $E(H(\varphi))$ containing the variables appearing in the negated predicate. For any class of hypergraphs C, Φ_C denotes the class of all ECQs φ with $H(\varphi) \in C$.

We emphasise that $H(\varphi)$ does **not** contain any hyperedges corresponding to the disequalities of φ; note that this makes positive results in terms of $H(\varphi)$ stronger, but it also makes these results harder to prove.[3]

Our first result uses the treewidth of a hypergraph (Definition 4, originally from [40]). For the definitions of other width measures such as fractional hypertreewidth and adaptive width, which we will only use in a blackbox manner throughout this work, we refer the reader for instance to [36] and to the full version [22].

Definition 4. (tree decomposition, treewidth) A *tree decomposition* of a hypergraph H is a pair $(T, \mathbf{B})$ where T is a (rooted) tree and $\mathbf{B}$ assigns a subset $B_t \subseteq V(H)$ (called a *bag*) to each $t \in V(T)$. The following two conditions are satisfied: (i) for each $e \in E(H)$ there is a vertex $t \in V(T)$ such that $e \subseteq B_t$, and (ii) for each $v \in V(H)$ the set $\{t \in V(T) \mid v \in B_t\}$ induces a (connected) subtree of T. The treewidth $\mathrm{tw}(T, \mathbf{B})$ of the tree decomposition $(T, \mathbf{B})$ is $\max_{t \in V(T)} |B_t| - 1$. The *treewidth* $\mathrm{tw}(H)$ of H is the minimum of $\mathrm{tw}(T, \mathbf{B})$, minimised over all tree decompositions $(T, \mathbf{B})$ of H.

Our first theorem is as follows.

Theorem 5. *Let t and a be positive integers. Let C be a class of hypergraphs such that every member of C has treewidth at most t and arity at most a. Then* #ECQ (Φ_C) *has an FPTRAS, running in time* $\exp(\mathrm{O}(||\varphi||^2)) \cdot \mathrm{poly}(\log(1/\delta), \varepsilon^{-1}, ||\mathcal{D}||)$.

Technical Challenges. While, in the case of bounded arity, negated predicates can be simulated by adding a negated relation (symbol) $\overline{R}$ for each relation (symbol) R, the disequalities have to be treated separately since we do not include them in the query hypergraph (see the discussion below Definition 3).

However, the main difficulty stems from the fact that our queries have both quantified and free variables; recall e.g. the query in (1). We note that the restricted case in which there are no quantified variables can be dealt with in a much easier way by using standard and well-established reductions from approximate counting to decision. For example, in the special case of arity 2, and with all disequalities present, approximate counting of answers to queries without quantified variables can be encoded as approximate counting subgraphs, which can be done efficiently for bounded treewidth graphs using colour-coding [3, 6].

If quantified variables are allowed, the situation changes drastically: While it does not matter for the decision version which variables are quantified and which are not, even quantifying a single variable can transform the counting version from easy to infeasible.[4] The core technical difficulty arising in both the recent

[3] If we included hyperedges corresponding to disequalities, we could just model these disequalities as (binary) relations and reduce to the case of conjunctive queries without disequalities. However, adding those hyperedges can increase the treewidth of the hypergraph (see Definition 4) significantly, so it would weaken the results significantly.

[4] Consider for example the following query in the signature of graphs

$$\varphi(x_1, \ldots, x_k) = \exists y \bigwedge_{i=1}^{k} E(y, x_i)\,.$$

Deciding whether φ has an answer is computationally trivial, since it is equivalent to deciding whether there are k (not necessarily distinct) vertices that have a common neighbour. In other words, we can always return "Yes" if the graph has at least one edge. However, (exactly) counting answers to φ cannot be done in time $O(|V(G)|^{k-\varepsilon})$ unless the Strong Exponential Time Hypothesis fails [16], ruling out any improvement over the brute-force algorithm for the counting version. In the case of approximate counting, the result of Arenas et al. [5] yields an FPRAS for counting answers to φ. Also, our result, Theorem 5, yields an FPTRAS even if we additionally add the constraint that the x_is should be pairwise different. Note that our result relaxes the

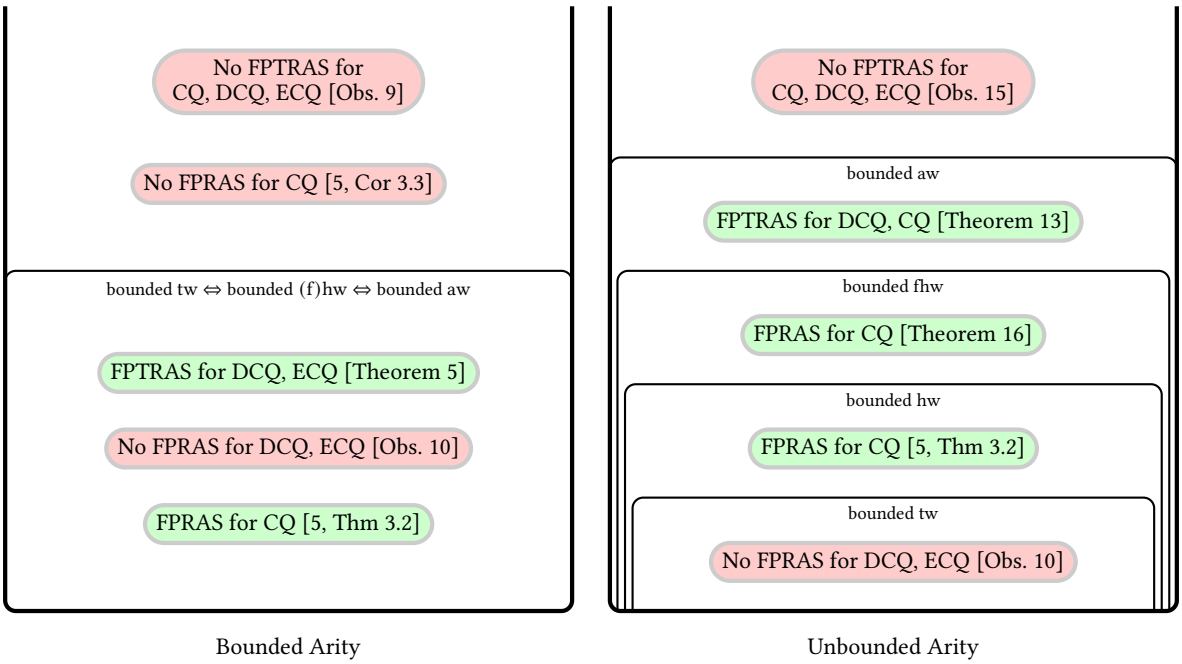

Figure 1: Overview of our results on approximately counting answers to conjunctive queries (CQs), to conjunctive queries with disequalities (DCQs), and to conjunctive queries with disequalities and negations (ECQs). Upper and lower bounds depend on a variety of width measures of the input queries, namely, treewidth (tw), hypertreewidth (hw), fractional hypertreewidth (fhw), and adaptive width (aw). The equivalence of the width parameters in the case of bounded arity is well known; we provide an explicit argument in the full version [22]. For completeness, we also compare our results to recent work of Arenas et al. [5]. The lower bounds either rely on the assumption that NP ≠ RP or on the rETH. All referenced theorems and observations are stated in Section 3. Note that, while our results complete the picture of the complexity of CQ, DCQ, and ECQ in the bounded arity case, two questions remain open for the unbounded arity case: Assuming the adaptive width is bounded, does ECQ have an FPTRAS, and does CQ have an FPRAS?

result of Arenas et al. [5] and also in our work is the problem of handling quantified variables in the context of approximate counting. While Arenas et al. were able to establish an FPRAS for the case of conjunctive queries of bounded (hyper)treewidth, we show that an FPRAS is not possible if disequalities are allowed (see Observation 10). For this reason, we relax the condition of feasibility of approximation by aiming for an FPTRAS.

In this work, we solve the problem of handling quantified variables by relying on the recent k-Hypergraph framework of Dell, Lapinskas and Meeks [15], which, in combination with colour-coding, will ultimately establish Theorem 5. We note that the presentation of our methods could be streamlined if we would only consider the bounded arity case. However, since understanding the unbounded arity case is also part of our goal, we present the framework in slightly more generality than we need for the bounded arity case.

notion of feasibility from an FPRAS to an FPTRAS since it turns out that the former is not always possible if disequalities are allowed (see Observation 10).

Finally, we point out that even exact counting becomes easy if we make y a free variable, that is, if we modify the query as follows: $\varphi'(x_1, \ldots, x_k, y) = \bigwedge_{i=1}^{k} E(y, x_i)$. The reason for this easiness is that counting answers to φ' in G is equivalent to counting homomorphisms from $H(\varphi')$ to G, which can be done efficiently as $H(\varphi')$ has treewidth 1 [13].

Before we continue with the presentation of further results, we will present a useful application of Theorem 5.

Application to Locally Injective Homomorphisms. Given graphs G and G', a homomorphism from G to G' is a mapping from the vertices of G to the vertices of G' that maps the edges of G to edges of G'. A homomorphism h is *locally injective* if for each vertex v of G, the restriction of h to the neighbourhood $N_G(v)$ of v in G is injective. Locally injective homomorphisms have been studied extensively, see [21] for an overview, and they can be applied, for instance, to model interference-free assignments of frequencies (of networks such as wireless networks) [20]. Some recent works on locally injective homomorphisms include [42] and [7]. The complexity of exactly counting locally injective homomorphisms from a fixed graph G to an input graph G' has been considered and is fully classified [41].

Given G and G', it is easy to construct an ECQ $\varphi(G)$ and a database $\mathcal{D}(G')$ such that locally injective homomorphisms from G to G' are in one-to-one correspondence with $\mathsf{Ans}(\varphi(G), \mathcal{D}(G'))$. For this, the signature of $\varphi(G)$ and $\mathcal{D}(G')$ has one binary relation E (representing the edge set of a graph). $\mathcal{D}(G')$ is the structure representing G' — its universe $U(\mathcal{D})$ is $V(G')$ and its relation $E^{\mathcal{D}(G')}$ is the edge set of G'. The query $\varphi(G)$ is constructed as follows.

For convenience, let $k = |V(G)|$ and assume that $V(G) = [k]$. Let $\mathrm{cn}(G)$ be the set of pairs $i \neq j$ of vertices of G such that i and j have a common neighbour. Then $\varphi(G)$ is the following query φ (which has no existential variables).

$$\varphi(x_1, \ldots, x_k) = \bigwedge_{\{i,j\} \in E(G)} E(x_i, x_j) \ \wedge \bigwedge_{(i,j) \in \mathrm{cn}(G)} x_i \neq x_j.$$

Consequently, Theorem 5 also gives an FPTRAS for counting locally injective homomorphisms from graphs G with bounded treewidth. Let C and C' be two classes of graphs.

Name: #LIHom(C, C')
Input: Graphs $G \in C$ and $G' \in C'$.
Parameter: $|V(G)|$.
Output: The number of locally injective homomorphisms from G to G'.

Corollary 6. *Let t be a positive integer. Let C_t be the class of all graphs with treewidth at most t, and let C be the class of all graphs. Then #LIHom(C_t, C) has an FPTRAS.*

Matching Lower Bound. Theorem 5 is optimal under standard assumptions — subject to the randomised Exponential Time Hypothesis (rETH) there is a matching lower bound showing that there is no FPTRAS for #ECQ (Φ_C) if the treewidth of C is unbounded. In fact, there is no FPTRAS for #CQ (Φ_C) in this case. In order to explain the lower bound, we first state the rETH.

Conjecture 7 (rETH, [28])**.** *There is a positive constant c such that no algorithm, deterministic or randomised, can decide the satisfiability of an n-variable 3-SAT instance in time $\exp(c \cdot n)$ (with failure probability at most $1/4$).*

The lower bound relies on a result by Marx [34, Theorem 1.3]. We state it here using our notation — in addition we allow randomised algorithms and therefore replace ETH by rETH.[5]

Theorem 8 ([34, Theorem 1.3])**.** *Let a be positive integer. Let C be a recursively enumerable class of hypergraphs such that every member of C has arity at most a. Suppose that the treewidth of hypergraphs in C is unbounded. If there is a computable function f and a randomised algorithm that, given a CQ $\varphi \in \Phi_C$ and a database $\mathcal{D}$ with $\mathrm{sig}(\varphi) \subseteq \mathrm{sig}(\mathcal{D})$, decides in time $f(H(\varphi)) \cdot (\|\varphi\| + \|\mathcal{D}\|)^{o(\mathrm{tw}(H(\varphi))/\log \mathrm{tw}(H(\varphi)))}$ whether $(\varphi, \mathcal{D})$ has an answer, then rETH fails.*

In order to apply Theorem 8, note that an FPTRAS for #CQ (Φ_C) provides a $(1/2, 1/4)$-approximation for #CQ (Φ_C) that runs in time $f(\|\varphi\|) \cdot \mathrm{poly}(\|\varphi\| + \|\mathcal{D}\|)$. It follows from the definition of $\|\varphi\|$ that $\|\varphi\|$ is a function of $H(\varphi)$. Thus, the FPTRAS provides a $(1/2, 1/4)$-approximation for #CQ (Φ_C) that runs in time $f(H(\varphi)) \cdot \mathrm{poly}(\|\varphi\| + \|\mathcal{D}\|)$. This approximation algorithm can be used to solve the decision problem of determining whether the output of #CQ (Φ_C) is nonzero. Thus, by Theorem 8, we obtain the following observation.

Observation 9. *Let a be positive integer. Let C be a recursively enumerable class of hypergraphs such that every member of C has arity at most a. If the treewidth of hypergraphs in C is unbounded then #CQ (Φ_C) does not have an FPTRAS, unless rETH fails.*

Observation 9 shows that Theorem 5 provides a tight result for hypergraph classes with bounded arity in the sense that, assuming rETH, there is an FPTRAS if and only if the treewidth is bounded. Note that the stated lower bound is slightly stronger than required since it also applies to conjunctive queries without extensions.

Theorem 5 shows that if C is any class of hypergraphs such that every member of C has treewidth at most t and arity at most a then there is an (ε, δ)-approximation algorithm for #ECQ (Φ_C) that runs in time $f(\|\varphi\|) \cdot \mathrm{poly}(\log(1/\delta), \varepsilon^{-1}, ||\mathcal{D}||))$, where f is exponential in $\|\varphi\|^2$. A natural question is whether the function f can be improved. Specifically, a polynomial f would imply the existence of an FPRAS for #ECQ (Φ_C). However, the following observation, proved by reduction from the Hamilton path problem, shows that there is no such FPRAS unless NP = RP, even when $t = 1$ and $a = 2$.

Observation 10. *Let C be the class of all hypergraphs with treewidth at most 1 and arity at most 2. Then there is no FPRAS for #DCQ (Φ_C), unless NP = RP.*

Proof. Given an n-vertex graph G, we will show how to construct (in time $\mathrm{poly}(n)$) an instance $(\varphi, \mathcal{D})$ of #DCQ (Φ_C) such that the answers in $\mathsf{Ans}(\varphi, \mathcal{D})$ are in one-to-one correspondence with the Hamiltonian paths of G. This implies (e.g., [19, Theorem 1]) that there is no FPRAS for #DCQ (Φ_C) unless NP = RP.

The construction is as follows. $U(\mathcal{D}) = V(G)$. The signature $\mathrm{sig}(\varphi) = \mathrm{sig}(\mathcal{D})$ contains a single binary relation symbol E. The relation $E^{\mathcal{D}}$ is the edge set $E(G)$. The query φ is defined as follows.

$$\varphi(x_1, \ldots, x_n) = \bigwedge_{i \in [n-1]} E(x_i, x_{i+1}) \ \wedge \bigwedge_{1 \leq i < j \leq n} x_i \neq x_j.$$

Note that φ has no existential variables so the solutions of $(\varphi, \mathcal{D})$ are in one-to-one correspondence with $\mathsf{Ans}(\varphi, \mathcal{D})$. It is clear from the definition of φ that these are also in one-to-one correspondence with the Hamilton paths of G.

It remains to show that $\varphi \in \Phi_C$, that is that $H(\varphi)$ has treewidth 1 and arity 2. Both of these follow from the fact that $H(\varphi)$ is the path $x_1, \ldots, x_n$, which follows from the definition of $H(\varphi)$ (Definition 3). □

Interestingly, the situation changes if we consider CQs without extensions. Arenas, Croquevielle, Jayaram, and Riveros [5] give an FPRAS for #CQ (Φ_C) for any bounded-treewidth class C of hypergraphs. We refer the reader to the full version [22] for the formal statement of their result. Observation 10 ensures that their result cannot be generalised to cover extended conjunctive queries (unless NP = RP). However, their result does apply to classes of hypergraphs with unbounded arity.

Beyond Bounded Arity. Even though Observation 9 gives a tight lower bound for classes C of hypergraphs with bounded arity, there is room for improvement if the arity in C is unbounded. In this setting, it is worth considering other notions of hypergraph width, such as hypertreewidth, fractional hypertreewidth, adaptive width and submodular width. As mentioned before, we refer the reader to [36] and the full version [22] for the definitions of these width measures, since we will only use them in a blackbox manner. Here we just give the relationships between them, from [36, Figure 2].

[5] In case it is not clear to the reader how Theorem 8 matches [34, Theorem 1.3], we refer to the introduction of [36] where this result is stated as Theorem 1.2.

Definition 11 (weakly dominated, strongly dominated, weakly equivalent)**.** A width measure is a function from hypergraphs to $\mathbb{R}_{\geq 0}$. Given two width measures w_1 and w_2, we say that w_1 is *weakly dominated* by w_2 if there is a function f such that every hypergraph H has $w_2(H) \leq f(w_1(H))$. We say that w_1 is *strongly dominated* by w_2 if w_1 is weakly dominated by w_2 and there is a class of hypergraphs that has unbounded w_1-width, but bounded w_2-width. We say that w_1 and w_2 are *weakly equivalent* if they weakly dominate each other.

If w_1 is strongly dominated by w_2 then the class of hypergraphs with bounded w_1-width is strictly contained in the class of hypergraphs with bounded w_2-width. Thus, algorithmic results based on bounded w_2-width have strictly greater applicability than algorithmic results based on bounded w_1-width. If w_1 and w_2 are weakly equivalent then algorithmic results for bounded w_1-width and bounded w_2-width are equivalent.

Lemma 12 ([36])**.** *Treewidth is strongly dominated by hypertreewidth. Hypertreewidth is strongly dominated by fractional hypertreewidth. Fractional hypertreewidth is strongly dominated by adaptive width, which is weakly equivalent to submodular width.*

Note that all of the width measures from Lemma 12 are weakly equivalent if we assume an overall bound on the arity of hypergraphs.

When restricting the queries to DCQs instead of ECQs, we can improve Theorem 5 by allowing unbounded arity, extending it to adaptive width. The following Theorem is proved in the full version [22]

Theorem 13. *Let b be a positive integer. Let C be a class of hypergraphs such that every member of C has adaptive width at most b. Then* #DCQ (Φ_C) *has an FPTRAS.*

Further Technical Challenges. In addition to the challenges that arose in the bounded arity case, a further issue that arises in the unbounded arity case is finding the right notion of treewidth on hypergraphs; recall that the notions of treewidth, hypertreewidth, fractional hypertreewidth and adaptive width are all equivalent in the bounded arity setting, but not in the unbounded arity setting.

We ultimately ended up with adaptive width as the most general width measure for which we can establish the existence of an FPTRAS. Despite the fact that approximate counting is often harder than decision [8, 19], we find that in the current setting, the criterion for tractability is the same for decision and approximate counting. In fact, it turns out that Theorem 13, i.e., the choice of adaptive width, is optimal, unless the rETH fails. This matching lower bound comes from another result of Marx [36, Theorem 7.1], which we express using our notation; for what follows, we write $\mathrm{aw}(H)$ for the adaptive width of a hypergraph H:

Theorem 14 ([36, Theorem 7.1])**.** *Let C be a recursively enumerable class of hypergraphs with unbounded adaptive width. If there is a computable function f and a randomised algorithm that, given a CQ $\varphi \in \Phi_C$ and a database $\mathcal{D}$ with $\mathrm{sig}(\varphi) \subseteq \mathrm{sig}(\mathcal{D})$, decides in time $f(H(\varphi)) \cdot (\|\varphi\| + \|\mathcal{D}\|)^{o(\mathrm{aw}(H(\varphi))^{1/4})}$ whether $(\varphi, \mathcal{D})$ has an answer, then rETH fails.*

As we noted earlier, an FPTRAS for #CQ (Φ_C) provides a $(1/2, 1/4)$-approximation for #CQ (Φ_C) that runs in time $f(\|\varphi\|) \cdot \mathrm{poly}(\|\varphi\| + \|\mathcal{D}\|)$. It follows from the definition of $\|\varphi\|$ that $\|\varphi\|$ is a function of $H(\varphi)$. Thus, the FPTRAS provides a $(1/2, 1/4)$-approximation for #CQ (Φ_C) that runs in time $f(H(\varphi)) \cdot \mathrm{poly}(\|\varphi\| + \|\mathcal{D}\|)$. This approximation algorithm can be used to solve the decision problem of determining whether the output of #CQ (Φ_C) is nonzero. Thus, by Theorem 14, we obtain the following observation.

Observation 15. *Let C be a recursively enumerable class of hypergraphs with unbounded adaptive width. Then* #CQ (Φ_C) *does not have an FPTRAS, unless rETH fails.*

Clearly, Observation 15 also rules out an FPTRAS for #DCQ (Φ_C) (matching Theorem 13) or an FPTRAS for #ECQ (Φ_C) when C has unbounded adaptive width.

As stated in Observation 10, there is little hope of improving Theorem 13 to obtain an FPRAS instead of an FPTRAS. However, as mentioned previously, if disequalities are not part of the query, the result by Arenas et al. [5] also applies to classes of hypergraphs with unbounded arity — they give an FPRAS for #CQ (Φ_C) if C is a class of hypergraphs with bounded hypertreewidth. We improve this result by showing that it suffices to require bounded fractional hypertreewidth — the following theorem is proved in the full version [22].

Theorem 16. *Let b be a positive integer. Let C be a class of hypergraphs such that every member of C has fractional hypertreewidth at most b. Then* #CQ (Φ_C) *has an FPRAS.*

We don't know whether Theorem 16 can be extended to instances with bounded adaptive width (closing the gap between Theorem 16 and the lower bound presented in Observation 15). However, this situation now mirrors the situation in the decision world: Bounded fractional hypertreewidth is the most general property of the class of underlying hypergraphs, for which the conjunctive query decision problem is known to be polynomial-time solvable, see [33]. Fixed-parameter tractability for this problem is also known for classes with bounded adaptive width [36], complemented by the matching lower bound stated in Theorem 14. The question whether bounded fractional hypertreewidth is the right answer for the existence of a polynomial-time algorithm remains open and we refer to the conclusion of [36] for further discussion regarding this question. With our results from Theorems 13 and 16 we now arrive at precisely the same gap for the existence of an FPRAS for the corresponding counting problem #CQ.

Extensions: Sampling and Unions. Using standard methods, the algorithmic results presented in this work can be extended in two ways. First, approximate counting algorithms can be lifted to obtain algorithms for approximately uniformly sampling answers. This is based on the fact that the problems that we considered are self-reducible (self-partitionable) [18, 29, 43]. Second, instead of single (extended) conjunctive queries, one can also consider unions thereof. We refer to the full version [22] for further details.

4 RELATED WORK

The first systematic study of the complexity of *exactly* counting answers to conjunctive queries is due to Pichler and Skritek [39], and Durand and Mengel [17]. Their results have been extended to unions of conjunctive queries by Chen and Mengel [10] and

to extended[6] conjunctive queries by Dell, Roth and Wellnitz [16]. Unfortunately, all of those results have shown that exact counting is infeasible even for very restricted classes of queries, indicating that relaxing to approximate counting is necessary if efficient algorithms are sought for wide classes of queries. As highlighted before, a recent result by Arenas et al. [5] shows that there is an FPRAS for approximately counting answers to conjunctive queries whenever the hypergraphs of the queries have bounded hypertreewidth, yielding a significantly wider class of tractable instances than for exact counting.

If we restrict to instances without existential quantifiers then counting answers to conjunctive queries is equivalent to the problem of counting homomorphisms from a small relational structure to a large one, the complexity of which was investigated by Dalmau and Jonsson [13] in the case of exact counting and by Bulatov and Živný [8] in the case of approximate counting.

The notion of an FPTRAS was introduced by Arvind and Raman [6] and has since been established as the standard notion for tractability of parameterised approximate counting problems (see [37] for an overview).

5 ALGORITHMIC METHODS AND PROOF TECHNIQUES

We conclude by presenting an overview of the most important tools and arguments used towards achieving our main results. The reader is referred to the full version [22] for a detailed and formal presentation.

At the heart of our approximate counting algorithms for counting answers to bounded-arity extended conjunctive queries (Theorem 5) and for counting answers to unbounded-arity conjunctive queries with disequalities (Theorem 13) lies a randomised reduction from these problems to the problem of determining whether there is a homomorphism between two associated relational structures. While reductions from approximate counting to decision are common, (e.g. approximate subgraph counting via colour-coding [3, 6]), the main difficulty in the current setting is the presence of quantified variables which make the usual reductions fail. We solve this problem by finding an algorithm which uses an oracle for the homomorphism decision problem between relational structures to solve the CQ counting problem by identifying and using carefully-tuned auxiliary structures. In the following lemma, the sizes $\|\cdot\|$ of queries, databases, and signatures are defined in the usual way (see the full version [22] for details).

Lemma 17 (informal version; see [22] for details)**.** *There is a randomised algorithm that is equipped with oracle access to the homomorphism decision problem over relational structures*

$$(\mathcal{A}, \mathcal{B}) \mapsto \begin{cases} 1 & \text{there is a homomorphism from } \mathcal{A} \text{ to } \mathcal{B} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

and takes as inputs an extended conjunctive query φ, a database $\mathcal{D}$ and $\varepsilon > 0$. The algorithm computes w.h.p. an ε-approximation of the number of answers of φ in $\mathcal{D}$ in time

$$\exp(\mathrm{O}(\|\varphi\|^2)) \cdot \mathrm{poly}(\varepsilon^{-1}, \|\mathcal{D}\|, \nu\|\mathcal{D}\|^a), \tag{3}$$

where a is the arity and ν is the number of negated predicates of φ.

Each oracle query $(\widehat{\mathcal{A}}, \widehat{\mathcal{B}})$ that is made by the algorithm has the property that $\widehat{\mathcal{A}}$ can be obtained from the structure representing the hypergraph of φ by adding unary relations. It also satisfies $\|\widehat{\mathcal{A}}\| \leq \mathrm{poly}(\|\varphi\|)$.

We next discuss how to prove Lemma 17. Recall that the hypergraph of φ contains a hyperedge for every atom and for every negated atom of φ. Crucially, it does *not* contain hyperedges for disequalities. Hence, proving Lemma 17 requires us not only to solve the problem of quantified variables, but also to deal with disequalities.

Our proof builds on the recent *k-hypergraph framework* of Dell, Lapinskas and Meeks [15], which shows how to reduce approximate counting problems to *coloured* or *partitioned* decision problems. It is not immediately clear how this coloured version is applicable (both because of existential variables and because of the colouring) but we will see how to use it after stating an informal version of their theorem. In the following theorem, $V_1, \ldots, V_\ell$ are disjoint sets of vertices of a hypergraph H and the hypergraph $H[V_1, \ldots, V_\ell]$ has their union as its vertex set and contains all hyperedges of H that intersect each V_i (precisely) once.

Theorem 18 (informal version of Theorem 1 of [15])**.** *There is an efficient algorithm $A(H, \varepsilon)$ with the following behaviour. Suppose that ε is a positive real number, H is an ℓ-uniform hypergraph, and that (in addition to learning $V(H)$ and ℓ) the algorithm A has access to an oracle for computing the following function*

$$V_1, \ldots, V_\ell \mapsto \begin{cases} 1 & \text{if } H[V_1, \ldots, V_\ell] \text{ has no hyperedges} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

With high probability, $A(H, \varepsilon)$ computes an ε-approximation to $|E(H)|$.

We emphasise that, while learning $V(H)$, ℓ, and ε, the algorithm in Theorem 18 does not learn $E(H)$ as part of the input. The only access that the algorithm has to $E(H)$ is via the oracle which evaluates the predicate $\mathrm{EdgeFree}(H[V_1, \ldots, V_\ell])$.

With Theorem 18 in hand, we proceed towards proving Lemma 17 as follows: Given an extended conjunctive query φ with ℓ free variables and a database $\mathcal{D}$ the goal is to approximately count the number of answers of φ in $\mathcal{D}$. We define a corresponding hypergraph $H(\varphi, \mathcal{D})$ to which we can then apply Theorem 18. The vertex set of $H(\varphi, \mathcal{D})$ consists of ℓ copies of the universe of $\mathcal{D}$ (one copy for each free variable of φ). The purpose of the copies is to ensure that $H(\varphi, \mathcal{D})$ is ℓ-uniform. Naturally, the hyperedges of $H(\varphi, \mathcal{D})$ are chosen to correspond to the answers that we wish to count.

The key question is how to provide the oracle used by Theorem 18. We will provide it using the oracle for (2) given by Lemma 17. To do this, we want to reduce the decision problem solved by the oracle in Theorem 18 (as applied to the hypergraph $H(\varphi, \mathcal{D})$) to the problem of detecting homomorphisms from $\widehat{\mathcal{A}}$ to $\widehat{\mathcal{B}}$, where $\widehat{\mathcal{A}}$ and $\widehat{\mathcal{B}}$ are some appropriately-defined structures associated with the query φ and the database $\mathcal{D}$; in what follows, we elaborate on our construction.

We start with $\widehat{\mathcal{A}}$, which will only depend on φ; we make this explicit and write $\widehat{\mathcal{A}} = \widehat{\mathcal{A}}(\varphi)$ where

- The universe of $\widehat{\mathcal{A}}(\varphi)$ is the set of variables of φ.

[6] "Disequalities" are sometimes referred to as "Inequalities", and "Negations" are sometimes referred to as "Non-monotone Constraints".

- For each atom "$(v_1, \ldots, v_j) \in R$" of φ, we add the tuple $(v_1, \ldots, v_j)$ to the relation R of $\widehat{\mathcal{A}}$.
- For each negated atom "$(v_1, \ldots, v_j) \notin R$" of φ, we add the tuple $(v_1, \ldots, v_j)$ to the relation $\overline{R}$ of $\widehat{\mathcal{A}}$.
- For each variable v_i of φ, we add a unary relation $P_i = \{v_i\}$.
- For each disequality $\eta =$ "$v_i \neq v_j$" (with $i < j$), we add two unary relations $R_\eta = \{v_i\}$ and $B_\eta = \{v_j\}$.

We refer the reader to the full version [22] for a detailed construction. Note that $\widehat{\mathcal{A}}$ has the same treewidth and adaptive width, respectively, as the hypergraph $H(\varphi)$: Essentially, $\widehat{\mathcal{A}}$ can be obtained from $H(\varphi)$ by labelling the hyperedges and adding unary predicates.

Now, ideally, for each tuple $(V_1, \ldots, V_\ell)$ of disjoint vertex subsets of $H(\varphi, \mathcal{D})$, we would like to also construct a structure $\widehat{\mathcal{B}}$ of size bounded by a polynomial in $\|\varphi\|$ and $\|\mathcal{D}\|$ such that there is a homomorphism from $\widehat{\mathcal{A}}(\varphi)$ to $\widehat{\mathcal{B}}$ if and only if $H(\varphi, \mathcal{D})[V_1, \ldots, V_\ell]$ has a hyperedge; this would conclude the construction by Theorem 18 and the fact that $H(\varphi, \mathcal{D})$ has as many hyperedges as φ has answers in $\mathcal{D}$.

Unfortunately, the presence of disequalities in φ enforces partial injectivity constraints and hence prevents us from constructing a *single* structure $\widehat{\mathcal{B}}$ with the above property.[7] However, at this point, we can rely on colour-coding to circumvent this problem. More precisely, given the tuple $(V_1, \ldots, V_\ell)$ we consider multiple structures $\widehat{\mathcal{B}}_{\mathbf{f}}$, each corresponding to a collection $\mathbf{f}$ of colouring functions which assign colours to the elements of the database $\mathcal{D}$ as defined below. The objective of the construction is that $H(\varphi, \mathcal{D})[V_1, \ldots, V_\ell]$ has a hyperedge if and only if, for at least one of the $\widehat{\mathcal{B}}_{\mathbf{f}}$, there is a homomorphism from $\widehat{\mathcal{A}}(\varphi)$ to $\widehat{\mathcal{B}}_{\mathbf{f}}$. We will also show that a random choice of the structure $\widehat{\mathcal{B}}_{\mathbf{f}}$ has the desired property with high probability.

We now present the construction of the structure $\widehat{\mathcal{B}}_{\mathbf{f}}$ in more detail. Let φ be an extended conjunctive query and let $\mathcal{D}$ be a database. Let ℓ and k be the number of free and quantified variables of φ, respectively. Recall that the vertex set of $H(\varphi, \mathcal{D})$ is the disjoint union of ℓ copies of the universe of $\mathcal{D}$. Therefore, let us assume that each vertex of $H(\varphi, \mathcal{D})$ is of the form (x, i) where x is in the universe of $\mathcal{D}$ and i is the index of the copy. We focus here on the most important case, which is where each set V_j is from a distinct copy (without loss of generality, from the j'th copy).

Let $\mathbf{f}$ be a collection of colouring functions that contains, for each disequality η of φ, a function f_η from the universe of $\mathcal{D}$ to $\{r, b\}$. We construct a structure $\widehat{\mathcal{B}} := \widehat{\mathcal{B}}(\varphi, \mathcal{D}, V_1, \ldots, V_\ell, \mathbf{f})$ as follows:

- For each $i \in \{1, \ldots, \ell\}$ let $S_i = V_i$ (recall that the elements of V_i are of the form (x, i)), and for each $i \in \{\ell + 1, \ldots, \ell + k\}$ let $S_i = U(\mathcal{D}) \times \{i\}$, that is, S_i contains all pairs (x, i) with $x \in U(\mathcal{D})$. Define the universe of $\widehat{\mathcal{B}}$ as $U(\widehat{\mathcal{B}}) = \bigcup_{i=1}^{\ell+k} S_i$.
- For each arity-a relation R of $\mathcal{D}$, $\widehat{\mathcal{B}}$ has the arity-a relation $R^{\widehat{\mathcal{B}}}$ containing each tuple $((w_1, i_1), \ldots, (w_a, i_a)) \in S_{i_1} \times \ldots \times S_{i_a}$ for all $(i_1, \ldots, i_a)$ and $(w_1, \ldots, w_a) \in R$.
- For each arity-a relation R of $\mathcal{D}$, $\widehat{\mathcal{B}}$ has the arity-a relation $\overline{R}^{\widehat{\mathcal{B}}}$ containing each tuple $((w_1, i_1), \ldots, (w_a, i_a)) \in S_{i_1} \times \ldots \times S_{i_a}$ for all $(i_1, \ldots, i_a)$ and $(w_1, \ldots, w_a) \notin R$.
- For each variable x_i of φ, $\widehat{\mathcal{B}}$ has a unary relation $P_i^{\widehat{\mathcal{B}}} := S_i$.
- For each disequality η of φ, we add to $\widehat{\mathcal{B}}$ the unary relations

$$R_\eta^{\widehat{\mathcal{B}}} := \{(x_i, j) \in U(\widehat{\mathcal{B}}) \mid f_\eta(x_i) = r\},$$

and

$$B_\eta^{\widehat{\mathcal{B}}} := \{(x_i, j) \in U(\widehat{\mathcal{B}}) \mid f_\eta(x_i) = b\}.$$

For a more comprehensive presentation of the construction of $\widehat{\mathcal{B}}(\varphi, \mathcal{D}, V_1, \ldots, V_\ell, \mathbf{f})$, including a discussion on its size, we refer the reader to the full version [22].

The following result establishes correctness of our construction; again, we refer the reader to the full version [22] for the formal statement and the proof.

Lemma 19 (informal version). *Let $\varphi, \mathcal{D}, \ell, (V_1, \ldots, V_\ell)$, and $H(\varphi, \mathcal{D})$ as above. Then $H(\varphi, \mathcal{D})[V_1, \ldots, V_\ell]$ has a hyperedge if and only if there exists a collection $\mathbf{f}$ of colouring functions such that there is a homomorphism from $\widehat{\mathcal{A}}(\varphi)$ to $\widehat{\mathcal{B}}(\varphi, \mathcal{D}, V_1, \ldots, V_\ell, \mathbf{f})$.*

With the above lemma in hand, we are now able to sketch the proof of Lemma 17.

Proof Sketch of Lemma 17. Given ε, φ, and $\mathcal{D}$, we first construct $\widehat{\mathcal{A}}(\varphi)$. Next we consider (but we do not construct) the hypergraph $H(\varphi, \mathcal{D})$, the number of edges of which is the number of answers of φ in $\mathcal{D}$.

We run the algorithm given by Theorem 18 with input ε to obtain, with high probability, an ε-approximation of $|E(H(\varphi, \mathcal{D})|$. This, however, requires us to (efficiently) simulate the oracle

$$V_1, \ldots, V_\ell \mapsto \begin{cases} 1 & H(\varphi, \mathcal{D})[V_1, \ldots, V_\ell] \text{ has no hyperedges} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Given $(V_1, \ldots, V_\ell)$ for which an oracle call must be simulated, we observe that, by Lemma 19, $H(\varphi, \mathcal{D})[V_1, \ldots, V_\ell]$ has an hyperedge if and only if there exists a collection of colouring functions $\mathbf{f}$ such that there is a homomorphism from $\widehat{\mathcal{A}}(\varphi)$ to $\widehat{\mathcal{B}}(\varphi, \mathcal{D}, V_1, \ldots, V_\ell, \mathbf{f})$. We take Q random guesses for the collection $\mathbf{f}$, where

$$Q = 4^{|\Delta|} \cdot \mathrm{poly}(\varepsilon^{-1}, \ell, \log n).$$

Here, Δ is the set of disequalities of φ, ℓ is the number of free variables of φ and n is the number of elements of $\mathcal{D}$.

For each guess of $\mathbf{f}$, we construct $\widehat{\mathcal{B}}(\varphi, \mathcal{D}, V_1, \ldots, V_\ell, \mathbf{f})$, and we use the homomorphism decision oracle (2) from the assumptions of Lemma 17 to decide whether there is a homomorphism from $\widehat{\mathcal{A}}(\varphi)$ to $\widehat{\mathcal{B}}(\varphi, \mathcal{D}, V_1, \ldots, V_\ell, \mathbf{f})$. If for at least one of the guesses, our oracle returns 1, we determine that $H(\varphi, \mathcal{D})[V_1, \ldots, V_\ell]$ has a hyperedge, and thus we answer the oracle query (5) by returning 0. Otherwise, we return 1.

The choice of Q allows us to prove that, with high probability,[8] one of our guesses yields an $\mathbf{f}$ for which a homomorphism exists (given that $H(\varphi, \mathcal{D})[V_1, \ldots, V_\ell]$ has a hyperedge; otherwise, no guess can yield a homomorphisms by Lemma 19).

[7] As mentioned previously, we *could* add binary relations for the disequalities to $\widehat{\mathcal{A}}(\varphi)$ to naively solve this problem. However, with those additional relations, the treewidth (and the adaptive width) of $\widehat{\mathcal{A}}(\varphi)$ could become arbitrarily large, even when $H(\varphi)$ itself has low treewidth (even when it has treewidth 1).

[8] We wish to emphasise that we swept under the rug many technicalities concerning the success probabilities of our nested oracle simulations. A comprehensive treatment of the necessary probability amplifications can be found in the full version [22].

Finally, this allows us to return w.h.p. an ε-approximation of the the number of edges of $H(\varphi, \mathcal{D})$. Since the edge count of $H(\varphi, \mathcal{D})$ is equal to the number of answers of φ in $\mathcal{D}$ this gives the sought-for result.

We point out that the overall running time is dominated by an exponential function in $\|\varphi\|^2$ which is due to the running time of the algorithm in Theorem 18, and by a polynomial in the size of $\widehat{\mathcal{B}}(\varphi, \mathcal{D}, V_1, \ldots, V_\ell, \mathbf{f})$ which crucially depends on both, the arity and the number of disequalities of φ. More precisely, the full proof will establish the overall running time bound given in Equation (3). □

With Lemma 17 in hand, we now sketch the proofs of Theorems 5 and 13.

Proof Sketch of Theorem 5. We use Lemma 17 and simulate the oracle queries to the homomorphism decision problem by running the algorithm of Dalmau et al. [14]. Since each oracle query $(\widehat{\mathcal{A}}, \widehat{\mathcal{B}})$ has the property that $\widehat{\mathcal{A}}$ can be obtained from the hypergraph of φ by adding unary relations, it is easy to see that the treewidth does not increase and is thus still bounded by the constant t in the statement of the theorem. Therefore, the algorithm in [13] runs in polynomial time. The overall running time is obtained by observing that

$$\nu\|\mathcal{D}\|^a \leq \mathsf{poly}(\|\varphi\|, \|\mathcal{D}\|)\,,$$

since $a \in O(1)$. □

Proof Sketch of Theorem 13. We proceed similarly to the previous proof, with three exceptions:

First, we do not use the algorithm in [14], but instead the algorithm of Marx for the unbounded arity case [36].

Second, similarly to treewidth, the adaptive width does not increase if unary predicates are added.

Third, we obtain the desired running time by observing that $\nu\|\mathcal{D}\|^a = 0$ since we do not have negated predicates; note that this is important as $\|\mathcal{D}\|^a$ is not bounded by a polynomial in $\|\mathcal{D}\|$ and $\|\varphi\|$, since a is unbounded. □

REFERENCES

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. 1995. *Foundations of Databases.* Addison-Wesley. http://webdam.inria.fr/Alice/

[2] Isolde Adler. 2006. *Width functions for hypertree decompositions.* Ph.D. Dissertation. Albert-Ludwigs-Universität Freiburg. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.95.2257&rep=rep1&type=pdf

[3] Noga Alon, Phuong Dao, Iman Hajirasouliha, Fereydoun Hormozdiari, and Süleyman Cenk Sahinalp. 2008. Biomolecular network motif counting and discovery by color coding. In *Proceedings 16th International Conference on Intelligent Systems for Molecular Biology (ISMB), Toronto, Canada, July 19-23, 2008.* 241–249. https://doi.org/10.1093/bioinformatics/btn163

[4] Marcelo Arenas, Pablo Barceló, and Juan L. Reutter. 2011. Query Languages for Data Exchange: Beyond Unions of Conjunctive Queries. *Theory Comput. Syst.* 49, 2 (2011), 489–564. https://doi.org/10.1007/s00224-010-9259-6

[5] Marcelo Arenas, Luis Alberto Croquevielle, Rajesh Jayaram, and Cristian Riveros. 2021. When is approximate counting for conjunctive queries tractable?. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, Samir Khuller and Virginia Vassilevska Williams (Eds.). ACM, 1015–1027. https://doi.org/10.1145/3406325.3451014

[6] Vikraman Arvind and Venkatesh Raman. 2002. Approximation Algorithms for Some Parameterized Counting Problems. In *Algorithms and Computation, 13th International Symposium, ISAAC 2002 Vancouver, BC, Canada, November 21-23, 2002, Proceedings (Lecture Notes in Computer Science, Vol. 2518)*, Prosenjit Bose and Pat Morin (Eds.). Springer, 453–464. https://doi.org/10.1007/3-540-36136-7_40

[7] Stefan Bard, Thomas Bellitto, Christopher Duffy, Gary MacGillivray, and Feiran Yang. 2018. Complexity of locally-injective homomorphisms to tournaments. *Discret. Math. Theor. Comput. Sci.* 20, 2 (2018). http://dmtcs.episciences.org/4999

[8] Andrei A. Bulatov and Stanislav Živný. 2020. Approximate Counting CSP Seen from the Other Side. *ACM Trans. Comput. Theory* 12, 2 (2020), 11:1–11:19. https://doi.org/10.1145/3389390

[9] Ashok K. Chandra and Philip M. Merlin. 1977. Optimal Implementation of Conjunctive Queries in Relational Data Bases. In *Proceedings of the 9th Annual ACM Symposium on Theory of Computing, May 4-6, 1977, Boulder, Colorado, USA*, John E. Hopcroft, Emily P. Friedman, and Michael A. Harrison (Eds.). ACM, 77–90. https://doi.org/10.1145/800105.803397

[10] Hubie Chen and Stefan Mengel. 2016. Counting Answers to Existential Positive Queries: A Complexity Classification. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, Tova Milo and Wang-Chiew Tan (Eds.). ACM, 315–326. https://doi.org/10.1145/2902251.2902279

[11] Gianluca Cima, Maurizio Lenzerini, and Antonella Poggi. 2020. Answering Conjunctive Queries with Inequalities in *DL-Lite*$_\mathcal{R}$. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020.* AAAI Press, 2782–2789. https://aaai.org/ojs/index.php/AAAI/article/view/5666

[12] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms* (1st ed.). Springer Publishing Company, Incorporated.

[13] Víctor Dalmau and Peter Jonsson. 2004. The complexity of counting homomorphisms seen from the other side. *Theor. Comput. Sci.* 329, 1-3 (2004), 315–323. https://doi.org/10.1016/j.tcs.2004.08.008

[14] Víctor Dalmau, Phokion G. Kolaitis, and Moshe Y. Vardi. 2002. Constraint Satisfaction, Bounded Treewidth, and Finite-Variable Logics. In *Principles and Practice of Constraint Programming - CP 2002, 8th International Conference, CP 2002, Ithaca, NY, USA, September 9-13, 2002, Proceedings (Lecture Notes in Computer Science, Vol. 2470)*, Pascal Van Hentenryck (Ed.). Springer, 310–326. https://doi.org/10.1007/3-540-46135-3_21

[15] Holger Dell, John Lapinskas, and Kitty Meeks. 2020. Approximately counting and sampling small witnesses using a colourful decision oracle. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, Shuchi Chawla (Ed.). SIAM, 2201–2211. https://doi.org/10.1137/1.9781611975994.135

[16] Holger Dell, Marc Roth, and Philip Wellnitz. 2019. Counting Answers to Existential Questions. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece (LIPIcs, Vol. 132)*, Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 113:1–113:15. https://doi.org/10.4230/LIPIcs.ICALP.2019.113

[17] Arnaud Durand and Stefan Mengel. 2015. Structural Tractability of Counting of Solutions to Conjunctive Queries. *Theory Comput. Syst.* 57, 4 (2015), 1202–1249. https://doi.org/10.1007/s00224-014-9543-y

[18] Martin Dyer and Catherine Greenhill. 1999. Random walks on combinatorial objects. *London Mathematical Society Lecture Note Series* (1999), 101–136.

[19] Martin E. Dyer, Leslie Ann Goldberg, Catherine S. Greenhill, and Mark Jerrum. 2004. The Relative Complexity of Approximate Counting Problems. *Algorithmica* 38, 3 (2004), 471–500. https://doi.org/10.1007/s00453-003-1073-y

[20] Jiří Fiala, Ton Kloks, and Jan Kratochvíl. 2001. Fixed-parameter complexity of lambda-labelings. *Discret. Appl. Math.* 113, 1 (2001), 59–72. https://doi.org/10.1016/S0166-218X(00)00387-5

[21] Jiří Fiala and Jan Kratochvíl. 2008. Locally constrained graph homomorphisms - structure, complexity, and applications. *Comput. Sci. Rev.* 2, 2 (2008), 97–111. https://doi.org/10.1016/j.cosrev.2008.06.001

[22] Jacob Focke, Leslie Ann Goldberg, Marc Roth, and Stanislav Živný. 2021. Approximately Counting Answers to Conjunctive Queries with Disequalities and Negations. *CoRR* abs/2103.12468 (2021). arXiv:2103.12468 https://arxiv.org/abs/2103.12468

[23] Georg Gottlob, Nicola Leone, and Francesco Scarcello. 2001. The complexity of acyclic conjunctive queries. *J. ACM* 48, 3 (2001), 431–498. https://doi.org/10.1145/382780.382783

[24] Georg Gottlob, Nicola Leone, and Francesco Scarcello. 2002. Hypertree decomposition and tractable queries. *J. Comput. System Sci.* 64, 3 (2002), 579–627. https://doi.org/10.1006/jcss.2001.1809

[25] Martin Grohe. 2007. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM* 54, 1 (2007), 1:1–1:24. https://doi.org/10.1145/1206035.1206036

[26] Martin Grohe and Dániel Marx. 2014. Constraint Solving via Fractional Edge Covers. *ACM Transactions on Algorithms* 11, 1 (2014), 4:1–4:20. https://doi.org/10.1145/2636918

[27] Víctor Gutiérrez-Basulto, Yazmín Angélica Ibáñez-García, Roman Kontchakov, and Egor V. Kostylev. 2015. Queries with negation and inequalities over lightweight ontologies. *J. Web Semant.* 35 (2015), 184–202. https://doi.org/10.1016/j.websem.2015.06.002

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

[29] Mark Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. 1986. Random Generation of Combinatorial Structures from a Uniform Distribution. *Theor. Comput. Sci.* 43 (1986), 169–188. https://doi.org/10.1016/0304-3975(86)90174-X

[30] Richard M. Karp and Michael Luby. 1983. Monte-Carlo Algorithms for Enumeration and Reliability Problems. In *24th Annual Symposium on Foundations of Computer Science, Tucson, Arizona, USA, 7-9 November 1983*. IEEE Computer Society, 56–64. https://doi.org/10.1109/SFCS.1983.35

[31] Paraschos Koutris, Tova Milo, Sudeepa Roy, and Dan Suciu. 2017. Answering Conjunctive Queries with Inequalities. *Theory Comput. Syst.* 61, 1 (2017), 2–30. https://doi.org/10.1007/s00224-016-9684-2

[32] Paraschos Koutris and Jef Wijsen. 2018. Consistent Query Answering for Primary Keys and Conjunctive Queries with Negated Atoms. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Houston, TX, USA, June 10-15, 2018*, Jan Van den Bussche and Marcelo Arenas (Eds.). ACM, 209–224. https://doi.org/10.1145/3196959.3196982

[33] Dániel Marx. 2010. Approximating fractional hypertree width. *ACM Trans. Algorithms* 6, 2 (2010), 29:1–29:17. https://doi.org/10.1145/1721837.1721845

[34] Dániel Marx. 2010. Can You Beat Treewidth? *Theory Comput.* 6, 1 (2010), 85–112. https://doi.org/10.4086/toc.2010.v006a005

[35] Dániel Marx. 2011. Tractable Structures for Constraint Satisfaction with Truth Tables. *Theory of Computing Systems* 48, 3 (2011), 444–464. https://doi.org/10.1007/s00224-009-9248-9

[36] Dániel Marx. 2013. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. *J. ACM* 60, 6 (2013). https://doi.org/10.1145/2535926 Article No. 42.

[37] Kitty Meeks. 2016. The challenges of unbounded treewidth in parameterised subgraph counting problems. *Discret. Appl. Math.* 198 (2016), 170–194. https://doi.org/10.1016/j.dam.2015.06.019

[38] Christos H. Papadimitriou and Mihalis Yannakakis. 1999. On the Complexity of Database Queries. *J. Comput. Syst. Sci.* 58, 3 (1999), 407–427. https://doi.org/10.1006/jcss.1999.1626

[39] Reinhard Pichler and Sebastian Skritek. 2013. Tractable counting of the answers to conjunctive queries. *J. Comput. Syst. Sci.* 79, 6 (2013), 984–1001. https://doi.org/10.1016/j.jcss.2013.01.012

[40] Neil Robertson and Paul D. Seymour. 1984. Graph minors. III. Planar tree-width. *Journal of Combinatorial Theory, Series B* 36, 1 (1984), 49–64. https://doi.org/10.1016/0095-8956(84)90013-3

[41] Marc Roth. 2021. Paramaterized Counting of Partially Injective Homomorphisms. *Algorithmica* (2021). https://doi.org/10.1007/s00453-021-00805-y

[42] Paweł Rzazewski. 2014. Exact algorithm for graph homomorphism and locally injective graph homomorphism. *Inf. Process. Lett.* 114, 7 (2014), 387–391. https://doi.org/10.1016/j.ipl.2014.02.012

[43] Claus-Peter Schnorr. 1976. Optimal Algorithms for Self-Reducible Problems. In *Third International Colloquium on Automata, Languages and Programming, University of Edinburgh, UK, July 20-23, 1976*, S. Michaelson and Robin Milner (Eds.). Edinburgh University Press, 322–337.

[44] Mihalis Yannakakis. 1981. Algorithms for Acyclic Database Schemes. In *Very Large Data Bases, 7th International Conference, September 9-11, 1981, Cannes, France, Proceedings*. IEEE Computer Society, 82–94.