

The Complexity of Conjunctive Queries with Degree 2

Matthias Lanzinger

Department of Computer Science, University of Oxford

United Kingdom

matthias.lanzinger@cs.ox.ac.uk

ABSTRACT

It is well known that the tractability of conjunctive query answering can be characterised in terms of treewidth when the problem is restricted to queries of bounded arity. We show that a similar characterisation also exists for classes of queries with unbounded arity and degree 2. To do so we introduce hypergraph dilutions as an alternative method to primal graph minors for studying substructures of hypergraphs. Using dilutions we observe an analogue to the Excluded Grid Theorem for degree 2 hypergraphs. In consequence, we show that the tractability of conjunctive query answering can be characterised in terms of generalised hypertree width. A similar characterisation is also shown for the corresponding counting problem. We also generalise our main structural result to arbitrary bounded degree and discuss possible paths towards a characterisation of tractable conjunctive query answering for the bounded degree case.

CCS CONCEPTS

• **Mathematics of computing** → **Hypergraphs**; • **Theory of computation** → **Problems, reductions and completeness**; • **Information systems** → **Relational database query languages**.

KEYWORDS

hypergraph, hypergraph dilation, conjunctive query, complexity of reasoning

ACM Reference Format:

Matthias Lanzinger. 2022. The Complexity of Conjunctive Queries with Degree 2. 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, 12 pages. <https://doi.org/10.1145/3517804.3524152>

1 INTRODUCTION

The complexity of answering conjunctive queries (CQs) has been a classic topic of study in database theory. CQs make up the core of many common query languages, such as SQL, SPARQL, or Datalog, and the algorithmic properties of CQs are therefore also critical to query answering in these languages. Beyond query answering, the complexity of CQs is of interest throughout theoretical computer

science where it is studied extensively under the equivalent frameworks of Constraint Satisfaction Problems or homomorphisms between relational structures.

When we speak of the complexity of answering CQs, we generally refer to the decision problem BCQ, where a CQ q and a database D are given, and the task is to decide whether q has a non-empty set of results when evaluated over the database D . In general, BCQ is NP-complete [8], but extensive research in the area has yielded large tractable¹ fragments of the problem by restricting the structure of queries [19, 22]. This line of study has also produced two important characterisations (in terms of query structure) of tractable CQ answering. Grohe [21] showed that BCQ restricted to bounded arity CQs is tractable exactly for query classes of bounded *treewidth modulo homomorphism*, i.e., only if there exists some constant c such that every query in the class is equivalent to a query with treewidth at most c (Proposition 2.1). Analogously, Marx [26] showed that the fixed-parameter tractability of BCQ parameterised by the query's hypergraph structure can be characterised in terms of *submodular width*.

Despite the wide-reaching consequences of these two results, the case of plain tractability for unbounded arity queries is still not well understood. While a number of parameters that induce tractable classes of the problem in the unbounded arity have been identified – e.g., hypertree width [19] and its generalisations [20, 22] – there is little evidence to suggest whether these parameters are even close to the limits of tractability, or whether there exists a *natural* characterisation for the unbounded arity case at all.

What makes the problem challenging is that very little is known of the hypergraph structure of queries with unbounded hypertree width (or any related parameters). Grohe's lower bound critically relies on the Excluded Grid Theorem by Robertson and Seymour [29]. Roughly speaking, in this setting the theorem states that if a query has large treewidth, then its primal graph will contain a large grid as a graph minor. Intractability of BCQ can then be shown by reduction of other problems into large enough grids. However, any minor of the primal graph lacks crucial information from the query. In particular, it is possible that large parts of the grid are covered in a single atom and thus the high connectivity of the grid is not reflected in the actual query. In particular, a reduction following Grohe's technique will produce exponentially large relations in such cases and hence not be efficient enough for the hardness results that we are aiming for.

Marx' characterisation in [26] addresses this issue through the more abstract notion of *embedding power*. Rather than relying on the existence of arbitrarily large grid minors, it is shown that in

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
PODS '22, June 12–17, 2022, Philadelphia, PA, USA

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9260-0/22/06...\$15.00
<https://doi.org/10.1145/3517804.3524152>

¹When not stated otherwise we use tractability to mean polynomial-time decidability.

classes of unbounded submodular width, there always exist instances with arbitrarily high embedding power, which in turn allows for “compact” embedding of certain other queries. While high embedding power allows for effective reductions into queries of unbounded arity, it is not known (nor suspected) that bounded embedding power or submodular width are sufficient conditions for non-parameterised tractability of BCQ in the usual setting².

These observations reveal two important questions in the search for the limits of tractability for BCQ when there is no bound on the arity.

- (1) *Are there appropriate notions of forbidden substructures in hypergraphs of unbounded rank?*
- (2) *Can we relate such forbidden substructures to any common width parameters for hypergraphs?*

Contributions. In this paper we attempt to answer these questions for hypergraphs with degree 2. We show that in this setting, large enough generalised hypertree width (ghw) always implies the existence of certain highly-connected substructures. This substructure relation, which we call *hypergraph dilation*, is also connected to the complexity of p-BCQ, the parameterisation of BCQ by the query. These observations allow us to follow a similar path as Grohe in the proof of the characterisation for bounded arity in [21] and obtain a first characterisation result for the complexity of unbounded arity CQ answering.

Assume $W[1] \neq \text{FPT}$. Let Q be a class of queries with degree 2 hypergraphs. Then $\text{BCQ}(Q)$ is tractable if and only if Q has bounded semantic generalised hypertree width.

The main contributions in this paper are summarised as follows.

- (1) To capture a type of relevant substructures of hypergraphs, we introduce *hypergraph dilutions* as a possible alternatives to primal graph minors. We show that CQ answering over a hypergraph class \mathcal{M} is fpt-reducible to CQ answering over hypergraphs \mathcal{H} , if all hypergraphs in \mathcal{M} are dilutions of hypergraphs in \mathcal{H} .
- (2) We show an analogue of the Excluded Grid Theorem for degree 2 hypergraphs. In particular, there exists a function f such that for any integer $n > 0$, any hypergraph H with $\text{ghw}(H) \geq f(n)$, contains a *jigsaw* hypergraph (the hypergraph dual of a grid) as a hypergraph dilation. This result may also be of independent interest.
- (3) In consequence, we show that BCQ over a class of hypergraphs \mathcal{H} is tractable if and only if \mathcal{H} has bounded generalised hypertree width. We extend the result to classes of queries with bounded semantic generalised hypertree width [4] and to the corresponding counting problem of counting answers of CQs.

It remains open whether this result can be extended to classes of arbitrary bounded degree. We propose possible paths to build on the results presented in this paper to proceed towards this goal. Moreover, we give a generalisation of the key structural result to the bounded degree case.

²The situation is different when truth-table representation is considered rather than standard “compact” representations via lists of tuples. See the discussion of related work on adaptive width below.

Related Work. To the best of our knowledge, there exists little related previous work on the complexity of BCQ for unbounded arity or even the structure of hypergraphs of unbounded *rank* (the maximum edge cardinality) beyond the two previously mentioned characterisation results. One important exception is work by Marx [25] which shows that BCQ is tractable only for classes of bounded adaptive width if the problem is given in truth table encoding (assuming a nonstandard conjecture). Note however that truth-table representation is generally exponentially larger than the standard succinct representation in terms of lists of tuples that we study.

We study the effect of restricting the query in this paper. This should not be confused with another prominent line of research on tractable fragments arising from restrictions to the structure of the database. There, a full dichotomy theorem is known due to Bulatov and Zhuk [6, 31]. However, the two sides of the problem are completely independent of each other and results for restrictions to the database do not affect the problem discussed here.

It is tempting to ask whether unbounded ghw also implies NP-hardness of BCQ in our setting, i.e., whether our main result can be strengthened to a dichotomy. Bodirsky and Grohe [5] have shown that, in general, no dichotomy for BCQ exists. That is, there are polynomially constructible classes of CQs for which BCQ is neither polynomial nor in NP (unless $P = NP$). Moreover, their argument is very flexible and suggests that their result may be extended to hold even under certain structural restrictions to the class of queries (e.g., classes of bounded degree).

Structure. We continue with preliminary notation and terminology in Section 2. We introduce hypergraph dilutions and show the fpt-reducibility of CQ answering along hypergraph dilutions in Section 3. We show the main structural results, and in consequence the complexity lower bounds, for hypergraphs with unbounded ghw in Section 4. In Section 5, we discuss challenges and possible paths for a characterisation of the bounded degree case. Concluding remarks and directions for further research are discussed in Section 6. Proof details that are skipped in the main body are presented in the appendix.

2 PRELIMINARIES

For positive integers n we will use $[n]$ as a shorthand for the set $\{1, 2, \dots, n\}$. When X is a set of sets we sometimes write $\bigcup X$ for $\bigcup_{x \in X} x$. We assume the reader to be familiar with standard notions of (parameterised) complexity theory. We refer to [27] and [16] for comprehensive overviews of computational complexity and parameterised complexity, respectively. As usual, we refer to a problem as *tractable* to say that it is in the complexity class P.

Graphs & Hypergraphs. A *hypergraph* H is a pair $(V(H), E(H))$ where $V(H)$ is the set of *vertices* and $E(H) \subseteq 2^{V(H)}$ is the set of (*hyper*)*edges*. We say that an edge e is *incident* to a vertex v if $v \in e$ and refer to the set of all edges incident to v by I_v . We treat *graphs* as hypergraphs where every edge has size 2, i.e., 2-uniform hypergraphs. The *degree* of a vertex v is defined as $\text{degree}(v) := |I_v|$. The degree of a hypergraph is the maximum degree over all its vertices. The *rank* of a hypergraph is $\text{rank}(H) := \max_{e \in E(H)} |e|$. The *primal graph* (or *Gaifman graph*) of a hypergraph H is the

graph G with $V(G) = V(H)$ and $\{x, y\} \in E(G)$ if and only if there is some edge in H that contains both x and y .

The *dual* H^d of H is the hypergraph with $V(H^d) = E(H)$ and $E(H^d) = \{I_v \mid v \in V(H)\}$. We say that a hypergraph H is *reduced* if (1) every vertex has at least degree 1, (2) H does not contain an empty edge, (3) and no two vertices have the same *vertex type*, i.e., for any two distinct vertices v, w , we have $I_v \neq I_w$. If a hypergraph is not reduced, we can easily make it reduced by deleting vertices with degree 1, empty edges and all but one vertex for every vertex type. Applying this process to some H yields a *reduced hypergraph* for H . The definition of reduced hypergraphs historically sometimes includes the condition that no two edges are the same. We consider this constraint implicitly always satisfied by our definition of $E(H)$ as a set. Importantly, if H is a reduced hypergraph, then $(H^d)^d = H$.

A *path* between two distinct vertices v_0, v_ℓ in H is a sequence $(v_0, e_0, v_1, e_1, \dots, e_{\ell-1}, v_\ell)$ alternating between vertices v_i and edges e_j such that $\{v_i, v_{i+1}\} \subseteq e_i$ for all $0 \leq i < \ell$. Furthermore, no edge or vertex occurs twice in a path.

Graph minors will play an important role in this paper. We say that a graph G is a minor of graph F if there exists a function $\mu: V(G) \rightarrow 2^{V(F)}$ (the *minor map*) such that

- (1) for every $v \in V(G)$, $\mu(v)$ is connected in F ,
- (2) for any two distinct $v, u \in V(G)$, $\mu(v) \cap \mu(u) = \emptyset$,
- (3) and if v and u are adjacent in G , then there is an edge in F that connects $\mu(v)$ and $\mu(u)$.

For connected graphs we can assume, w.l.o.g., that a minor map is *onto*, i.e., $V(F) = \bigcup_{v \in V(G)} \mu(v)$. Alternatively, graph minors are also commonly defined constructively in the following way. An *edge contraction* in a graph G removes an edge $\{v, w\}$ from G and merges the two vertices v, w into one new vertex which is adjacent to exactly the edges adjacent to v or w , except for the removed $\{v, w\}$. A graph G is a minor of graph F , if G can be reached from F by a sequence of vertex deletions, edge deletions, and edge contractions.

Width Parameters. We will be interested in the structure of hypergraphs in the case where certain parameters are large. We follow Adler [1] in the following definitions. A tuple $\langle T, (B_u)_{u \in T} \rangle$ is a *tree decomposition* of a hypergraph H if T is a tree, every B_u is a subset of $V(H)$ and the following two conditions are satisfied: (1) For every $e \in E(H)$ there is a node $u \in T$ s.t. $e \subseteq B_u$, and (2) for every vertex $v \in V(H)$, $\{u \in T \mid v \in B_u\}$ is connected in T . For functions $f: 2^{V(H)} \rightarrow \mathbb{R}^+$, the *f-width* of a tree decomposition is defined as $\sup\{f(B_u) \mid u \in T\}$ and the *f-width* of a hypergraph is the minimal *f-width* over all its tree decompositions. The *treewidth* $\text{tw}(H)$ of a hypergraph H is the *w-width*, where $w(B) = |B| - 1$. An *fractional edge cover* of vertex set $V' \subseteq V(H)$ is a set of mapping $\gamma: E(H) \rightarrow [0, 1]$ such that $\sum_{e \in I_v} \gamma(e) \geq 1$ for all $v \in V'$, i.e., γ assigns weights to all edges such that every vertex in V' has at least 1 total weight on its incident edges. The weight of a fractional edge cover γ is $\sum_{e \in E(H)} \gamma(e)$. The *fractional edge cover number* of $V' \subseteq V(H)$ is the minimum weight of a fractional edge cover of V' . An *(integral) edge cover* is a fractional edge cover where every edge is assigned either 0 or 1. Let ρ be the function associating sets of vertices with their integral edge cover number in H . The *generalised hypertree width* $\text{ghw}(H)$ of H is the ρ -width. Analogously, one can

define *fractional hypertree width* [22] as the ρ^* -width where ρ^* is the fractional edge cover number. We say that a class of hypergraphs has *bounded ghw* if there exists a constant c such that for every H in the class, $\text{ghw}(H) \leq c$. We use the same convention also for other numeric properties of hypergraphs of queries such as degree or treewidth.

The statement of our main result in terms of bounded ghw may be a source of confusion since there exist hypergraph classes with bounded fhw but unbounded ghw and bounded fhw is a sufficient condition for tractability [22]. For hypergraphs with bounded degree the two notions are equivalent up to some fixed function, i.e., every class has bounded fhw if and only if it has bounded ghw [18]. Thus, in the setting considered in this paper we can use the two notions interchangeably.

Conjunctive Queries. A *conjunctive query* (CQ) q is a function-free conjunction of relational atoms. Commonly, the definition of CQs also allows for (top-level) existential quantification of variables. In the context of this paper, and the decision problem BCQ as defined below, such quantification is of no consequence and all results for BCQ and p-BCQ hold also with existential quantification. This is not true for the counting problem where we explicitly consider only *full CQs*, i.e., CQs with no existential quantification. This is discussed further in the respective Section 4.4.

A *database* is a set of ground relational atoms. We say an assignment of v of variables in q to constants is a *solution* of q for D if every atom in q with variables replaced according to v is in D . We denote the set of all solutions of q for D as $q(D)$. The *arity* of a CQ is the maximal arity of its individual atoms. If no relation symbol occurs twice in q we say there are no *self-joins*. If q has no self-joins and no repeated variables in any atom we sometimes implicitly treat q as a join-query in relational algebra where the attributes for each relation are simply the lists of variables in the corresponding atoms of q .

The hypergraph of q is the hypergraph H such that $V(H) = \text{vars}(q)$ and for every atom $R(x_1, \dots, x_n)$, there exists an edge of the form $\{x_1, \dots, x_n\}$ in H (and no other edges). We transparently refer to properties of the hypergraph of q also as properties as q , e.g., by ghw or degree of q we refer to the ghw or degree of the hypergraph of q . Throughout this paper we are primarily interested in the following decision problem over some class of CQs \mathcal{Q} known as *Boolean Conjunctive Query Answering*.

BCQ(\mathcal{Q})

Instance: A CQ q in \mathcal{Q} and a database D
Question: $q(D) \neq \emptyset$?

We refer to BCQ parameterised by the hypergraph of the input query q as p-BCQ. For a hypergraph class \mathcal{H} we write $\text{BCQ}(\mathcal{H})$ to mean BCQ over the class of all CQs whose hypergraph is in \mathcal{H} . The same applies to other decision problems defined over classes of queries.

We say that two CQs q_1, q_2 are *equivalent* if $q_1(D) = q_2(D)$ for every database D . Every CQ q has a minimal (with respect to the number of atoms) equivalent query which is called the *core* of q , we write $\text{core}(q)$. For CQ q , we will also be interested in the minimal ghw over all equivalent queries. Let $\text{Eq}(q)$ be the equivalence

classes of all queries equivalent to q . The *semantic generalised hypertree width* of q ($\text{sem-ghw}(q)$) is $\min\{\text{ghw}(q') \mid q' \in \text{Eq}(q)\}$, i.e., the minimum ghw in the equivalence class of q . Analogously, we use *semantic treewidth* to refer to the minimum treewidth in the respective equivalence class. Note that semantic width is also commonly referred to as *width modulo homomorphism* in the literature since CQ equivalence coincides with homomorphic equivalence of queries. For full details and related definitions see [4].

The following two statements for BCQ will be of particular importance here. The first is what we informally refer to as Grohe’s characterisation throughout the paper. The second is a straightforward combination of two standard results of the field, one showing that ghw is equivalent to the more restricted notion of *hypertree width* (see [19]) up to a constant factor [3], and the other showing tractability of BCQ under bounded hypertree width [19].

PROPOSITION 2.1 (THEOREM 1.1, GROHE [21]). *Assume $\text{FPT} \neq \text{W}[1]$. Let \mathcal{Q} be a class of bounded arity CQs. The following three statements are equivalent:*

- (1) $\text{BCQ}(\mathcal{Q})$ is tractable;
- (2) $\text{p-BCQ}(\mathcal{Q})$ is fixed-parameter tractable;
- (3) \mathcal{Q} has bounded semantic treewidth.

If either statement is false, then $\text{p-BCQ}(\mathcal{Q})$ is $\text{W}[1]$ -hard.

PROPOSITION 2.2 (ADLER ET AL. [3], GOTTLÖB ET AL. [19]). *Let \mathcal{Q} be a class of CQs with bounded ghw. Then $\text{BCQ}(\mathcal{Q})$ is tractable.*

3 HYPERGRAPH DILUTIONS

In this section, we introduce hypergraph dilutions as a possible approach to identify relevant substructures of hypergraphs. As with graph minors, the goal of this notion is intuitively to induce an order of structural simplicity in the sense that if H is a hypergraph dilution of H' , then H should be “simpler” than H' . The difficulty of course lies in the question of what makes one hypergraph simpler than another. We do not claim to have an answer to this question and, moreover, do not propose that there is a single “correct” kind of simplicity. Rather, the generality of hypergraphs suggests that competing notions will be of interest in different settings.

In the context of our goal of identifying forbidden substructures for tractable CQ answering, the desired notion of simplicity is one that captures a kind of structural abstraction that adheres to a type of monotonicity of complexity, meaning that BCQ should not increase in complexity for simpler, more abstract, structures. Theorem 3.4 at the end of this section demonstrates that hypergraph dilutions capture this high-level idea of structural simplicity and abstraction in a meaningful way. In the following section we present further motivation for the notion, especially for hypergraphs of bounded degree.

Definition 3.1. For hypergraph H' , we say that H is a *hypergraph dilution* of H' if it is isomorphic to a hypergraph that can be reached from H' by a sequence of the following operations:

- (1) deleting a vertex (from the vertex set and all edges),
- (2) deleting an edge that is a proper subset of another edge,
- (3) *merging* on v : replacing all of the incident edges I_v of vertex v , by a new edge $(\bigcup I_v) \setminus \{v\}$.

We also say that H' *dilutes to* H and refer to the associated sequence of operations as a *dilution sequence* from H' to H .

Importantly, hypergraph dilutions do not allow deletion of arbitrary edges. This is motivated by our interest in the complexity of CQs. Hypergraph parameters that induce tractable CQ answering usually generalise the notion of hypergraph α -acyclicity [13]. An important observation there is that if there is some complex substructure (say a clique C_n) that is fully contained in a single separate hyperedge, then the complex interactions of the substructure C_n can roughly speaking be ignored when solving the associated query. Hence, removing arbitrary edges can “activate” arbitrarily complex subproblems.

Thus, deleting an edge e is only possible by deleting vertices such that e becomes a subedge of another (or equal, thus implicitly disappearing in the other edge). One special case where such deletion can be convenient is in hypergraphs that are not connected. Having multiple connected components is technically inconvenient and of little algorithmic importance – each component is essentially an independent instance – and it is common to assume connected instances. In the study of hypergraph dilutions this assumption is not necessary as we can always delete superfluous maximally connected components by deleting all vertices, leaving only a single empty edge, which is naturally a proper subset of any other edge.

The following observations on hypergraph dilutions are important in our further studies. The first two statements of Lemma 3.2 are straightforward to verify but of technical importance. In particular, the second statement also implies that every hypergraph has only a finite number of dilutions. The third statement is less simple. Deleting a vertex can possibly reduce ghw while deleting (or adding) a subedge cannot change the width at all. However, the effect of the merging operation of hypergraph dilutions is less clear since a new large edge is introduced, forcing vertices to occur in a bag of a decomposition for H that may not occur together in any optimal decomposition of H' . A proof of the third statement is given in the appendix.

LEMMA 3.2. *For hypergraphs H and H' such that H' dilutes to H , the following statements hold:*

- (1) $\text{degree}(H) \leq \text{degree}(H')$;
- (2) $|V(H)| + |E(H)| < |V(H')| + |E(H')|$;
- (3) $\text{ghw}(H) \leq \text{ghw}(H')$.

Our definition of hypergraph dilutions is of course inspired by graph minors. Previously, Adler et al. [2] introduced the notion of *hypergraph minors* as an analogue of graph minors for hypergraphs. There are some important parallels and differences between hypergraph minors and hypergraph dilutions that merit discussion. An important concept in hypergraph minors is the contraction of (the primal edge between) two vertices. Informally, contracting two vertices x, y means to replace them by a new vertex $v_{x,y}$ in the vertex set and in all edges that contain either x or y .

Definition 3.3 (Adler et al. [2]). For hypergraph H' , we say that H is a *hypergraph minor* of H' if H can be obtained from H' by a sequence of the following operations:

- (1) deleting a vertex,
- (2) deleting an edge that is a proper subset of another edge,
- (3) contraction of two vertices that are contained in a common hyperedge,

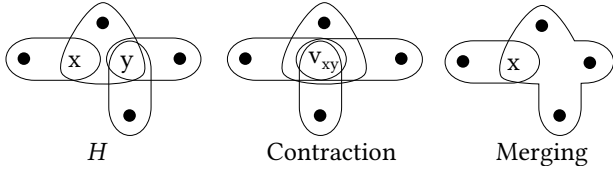


Figure 1: Example: Contraction vs Merging

- (4) or adding a hyperedge e , if the vertices of e already induce a clique in the primal graph before adding e .

The main difference between hypergraph minors and dilutions is the difference between contractions and mergings. Figure 1 provides a small illustration of this difference (the merging is on vertex y). Not only is the operation different but this simple example already illustrates how dilutions cannot be simulated by hypergraph minors or vice versa. In particular, the contraction in the example creates a vertex with degree 4, increasing the degree of the original graph. Since dilutions can not increase the degree the result of the contraction can not be a dilution of H . On the other hand, the merging creates an edge with 4 vertices. The only way this can be achieved using hypergraph minor operations is through adding a new edge over an existing 4-clique in the primal graph. However, there is no way to form a 4-clique in the primal graph, even with contraction. Furthermore, the last operation in Definition 3.3 would be problematic for our reduction in Theorem 3.4. For this reason, we consider only hypergraph dilutions for our structural results in later sections. It remains open whether similar results can be obtained for hypergraph minors. Note also that, in a sense, the contraction operation of hypergraph minors is a dual operation to the edge merging in dilutions. This relationship to contractions in the dual will play an important role in later sections.

Our main results ultimately hinge on two observations. The first is that for hypergraph classes \mathcal{H} and \mathcal{M} , if every hypergraph of \mathcal{M} is a dilution of a hypergraph in \mathcal{H} , then we can reduce from $\text{p-BCQ}(\mathcal{M})$ to $\text{p-BCQ}(\mathcal{H})$, formally stated in Theorem 3.4 below. The second key observation is that under bounded degree, high ghw guarantees the existence of certain dilutions. In combination, these two observations will then yield the lower bounds for our main results.

THEOREM 3.4. *Let \mathcal{H} be a recursively enumerable class of hypergraphs and let \mathcal{M} be a class such that any member is a hypergraph dilution of a hypergraph in \mathcal{H} . Then $\text{p-BCQ}(\mathcal{M})$ is fpt-reducible to $\text{p-BCQ}(\mathcal{H})$.*

PROOF IDEA. For some instance q, D_q with hypergraph M_q , we find by enumeration of \mathcal{H} a hypergraph H that dilutes to M_q and the corresponding dilution sequence $W = (w_1, \dots, w_\ell)$. For each dilution operation w_i – that produces hypergraph H_i from H_{i-1} – we can show how the query q_{i-1} and database D_{i-1} for H_{i-1} can be transformed into an equivalent instance q_i, D_i for H_i with $\pi_{\text{vars}(q_{i-1})}(q_i(D_i)) = q_{i-1}(D_{i-1})$, where $\pi_{\text{vars}(q_{i-1})}$ is the projection of solutions to the variables of q_{i-1} . Thus by traversing W in reverse, we arrive at an instance p, D_p with hypergraph $H_0 = H$ such $\pi_{\text{vars}(p)}(D_p) = q(D_q)$. Intuitively, this can be done by introducing keys in the database for the new positions introduced when

reversing a merging on a vertex v , and by extending all tuples by the same constant to reverse the deletion of a vertex.

For each operation, only linear time in size of the (step i) instance is required, and the total size of query and database increases at most in proportion to $\text{degree}(H)$ in each step. Hence, we observe $\|D_p\| = O(\text{degree}(H))^\ell \|D_q\|$ and analogous time bounds for the reduction, where H and ℓ both depend only on the parameter M_q . \square

It may seem natural to extend Definition 3.1 to CQs and consider reductions from classes of CQ dilutions instead of operating on hypergraph level. However, it is not clear how the operations from Definition 3.1 should be adapted to operate directly on queries. Consider the following example query $R(x, y, z) \wedge R(x, u, v) \wedge S(u, z)$ and consider the case analogous to deleting vertex v in the corresponding hypergraph. The atom $R(x, y, z)$ should not be changed but $R(x, u, v)$ would have to become a $R'(x, u)$ where R' is necessarily a new relation symbol since it has different arity than R . This change in relation symbol removes the implicit equality between variables u and y . It is unclear how the reduction in Theorem 3.4 can remain polynomial in the size of D if such situations occurred. Similar issues can arise when two edges in the hypergraph are merged into one. Note however that these problems only arise in the presence of self-joins and that Theorem 3.4 can be adapted to hold for classes of self-join free queries. In Section 4.3 we discuss how we can still derive our lower bounds for classes of queries through combination with previous results relating the complexity of all queries over a class of hypergraphs to specific classes of queries.

The complexity of deciding hypergraph dilutions is of little consequence to the contents of this paper. As the complexity may be of independent interest we state it here. An argument is given in the appendix.

THEOREM 3.5. *It is NP-complete to decide for input hypergraphs H and H' , whether H is a hypergraph dilution of H' .*

It is often technically convenient to consider the analogue of reduced hypergraphs for CQs. That is, we want to assume that no variables occur only in one atom, no atom's variables are a subset of some other atom's variables, and so on. These assumptions on CQs are usually motivated by the fact that they have no significant effect on the upper bounds of the problem and can be avoided via straightforward preprocessing. In conjunction with Theorem 3.4, the complexity implications of simplifying CQs in this way can be seen via the following Lemma 3.6, which will also be of technical importance in the following section.

LEMMA 3.6. *Let H be a reduced hypergraph for H' . Then H' dilutes to H , and a corresponding dilution sequence can be computed in polynomial time.*

4 FORBIDDEN DILUTIONS FOR DEGREE 2 CQS

In this section we show that degree 2 hypergraphs with high ghw always dilute to certain simple but highly connected structures. In particular, we obtain an analogue to the Excluded Grid Theorem for degree 2 hypergraphs. We show that p-BCQ over these contained structures is hard and thus putting everything together yields the base version of our main result.

THEOREM 4.1. *Assume $\text{FPT} \neq \text{W}[1]$. Let \mathcal{H} be a class of hypergraphs with degree 2. The following three statements are equivalent:*

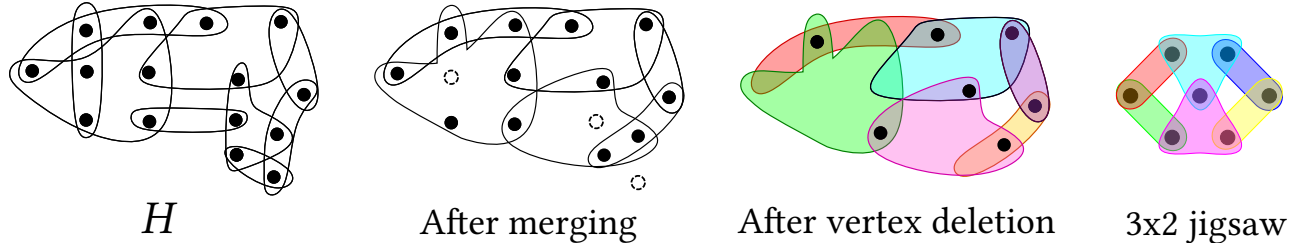


Figure 2: Example Dilution from H to the 3×2 -jigsaw.

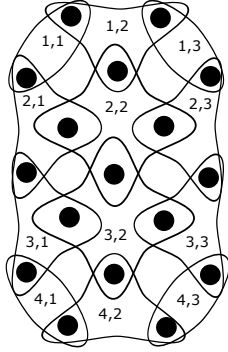


Figure 3: The 3×4 -jigsaw hypergraph

- (1) $\text{BCQ}(\mathcal{H})$ is tractable;
- (2) $\text{p-BCQ}(\mathcal{H})$ is fixed-parameter tractable;
- (3) \mathcal{H} has bounded generalised hypertree width.

If either statement is false, then $\text{p-BCQ}(\mathcal{H})$ is $\text{W}[1]$ -hard.

In general, there are tractable classes of BCQ that have bounded fractional hypertree width but unbounded generalised hypertree width. In this light, the characterisation in terms of generalised hypertree width may seem unintuitive. However, for bounded degree \mathcal{H} (and actually even more general restrictions) it is known that \mathcal{H} has bounded flw if and only if \mathcal{H} has bounded ghw [18]. Theorem 4.1 can therefore equivalently be stated in terms of fractional hypertree width (or just hypertree width).

4.1 The Structure of Hypergraphs with Degree 2 and Unbounded Generalised Hypertree Width

We will show that degree 2 hypergraphs always dilute to the hypergraph dual of a grid graph, which we will call a *jigsaw hypergraph*.

Definition 4.2 (Jigsaw Hypergraphs). An $n \times m$ -jigsaw is a hypergraph H with edges $\{e_{i,j} \mid i, j \in [n] \times [m]\}$ where every vertex has degree 2 and $|e_{i,j} \cap e_{i+1,j}| = 1$ and $|e_{i,j} \cap e_{i,j+1}| = 1$ for $i < n, j < m$ and no other pair of edges has a non-empty intersection.

The $n \times m$ -jigsaw is uniquely determined up to isomorphism. Figure 3 illustrates a 3×4 -jigsaw hypergraph. We call $n \times m$ the *dimension* of the jigsaw and we say that a class of jigsaws has unbounded dimension if there is no constant bound on either parameter. Note that the $n \times m$ -jigsaw dilutes to the $n \times (m-1)$ jigsaw (and analogously in the other axis).

Example 4.3. Figure 2 illustrates an example dilution of a hypergraph with degree 2 to a 3×2 -jigsaw. In the first step in the figure, three merging operations are performed. The vertices which we merge on are drawn as dashed empty circles. In a second step we delete superfluous vertices. The colours of the edges represent the correspondence to edges in the final jigsaw.

Our first goal in this section will be to show that it is always possible to dilute a degree 2 hypergraph H to an $n \times n$ -jigsaw where n depends on $\text{ghw}(H)$. We will first observe that graph minors and hypergraph dilutions are tightly connected in degree 2 hypergraphs. From there we then derive our main structural result (Theorem 4.7).

LEMMA 4.4. Let G be a connected graph and let H be a degree 2 hypergraph. If G is a minor of H^d , then G^d is a hypergraph dilution of H .

PROOF. We assume that H is a reduced hypergraph. Isolated vertices, empty edges and duplicate vertex types do not materially affect minor maps from G into H^d . By Lemma 3.6, there is always a dilution sequence from any hypergraph to its respective reduced version. Hence, the assumption can be made without loss of generality.

Let $\phi: E(H) \rightarrow V(H^d)$ be the bijection from edges in H to their corresponding vertex in the dual, and let $\mu: V(G) \rightarrow 2^{V(H^d)}$ be a minor map from G onto H^d . For every $v \in V(G)$, let $\delta(v) = \phi^{-1}(\mu(v))$ and observe that $\delta(v)$ is a connected set of edges in H .

For any two adjacent vertices u, v in G , there is an edge in H^d that connects $\mu(u)$ and $\mu(v)$. Hence, there is also a vertex $c_{u,v}$ that is both in an edge in $\delta(u)$ and an edge in $\delta(v)$. Since $c_{u,v}$ has degree 2, it is therefore connected to only one edge in $\delta(u)$. For each v adjacent to u in G fix such a $c_{u,v}$ and let us refer to the set of these fixed vertices for u as C_u . Let τ_u be the vertices that are incident only to edges in $\delta(u)$. Observe that either $\delta(u)$ contains one edge, or every edge in $\delta(u)$ is incident to at least one vertex τ_u . Suppose towards a contradiction that $\delta(u)$ consists of more than one edge and that there is an edge $e \in \delta(u)$ such that all vertices in e are incident to some other edge not in $\delta(u)$. Since all vertices have degree at most 2, that would imply that e is not incident to any other edge in $\delta(u)$, thus contradicting the connectedness of $\delta(u)$. Furthermore, note that τ_u and C_u are disjoint by definition.

Let H_1 be the hypergraph obtained by merging, for every $u \in V(G)$, all vertices in τ_u . By the above observations, either $\delta(u)$ was already a singleton, or the merging produced a single new merged edge e_u from all of the edges of $\delta(u)$, since every such edge was incident to some vertex in τ_u . By construction, H_1 is

clearly a dilution of H . Let $C = \bigcup_{u \in V(G)} C_u$ and observe that since $\tau_u \cap C_u = \emptyset$ for all $u \in V(G)$, no vertices in a C have been removed by the merging process.

Finally, let H_2 be the induced subhypergraph $H_1[C]$, i.e., the hypergraph obtained from H_1 by deleting all vertices not in C . Observe that for every edge $u \in E(G^d)$, there is a vertex $u \in V(G)$ and exactly one edge $e_u \cap C$ in H_2 . For every edge $\{u, v\}$ in G (or vertex $g_{u,v}$ in G^d), there is a vertex $c_{u,v}$ in C and thus in H_2 , such that $c_{u,v}$ is contained only in edges e_u and e_v . Since this correspondence from edges and vertices of G^d to vertices and edges in H_2 is one-to-one and H_2 contains only these edges and vertices by construction, the implications hold also in the other direction. Hence, H_2 is isomorphic to G^d and a hypergraph dilution of H . \square

This observed duality of graph minors and dilutions in degree 2 hypergraphs also illustrates a conceptual switch. Intuitively, high treewidth expresses large sets of highly connected vertices, while high ghw can be seen as a sign of large sets of highly connected edges. See also the discussion accompanying the definition of embedding power in [26] for further intuition.

PROPOSITION 4.5 (ROBERTSON AND SEYMOUR [29]). *There exists a function $f: \mathbb{N} \rightarrow \mathbb{N}$ with the following property: for every $n \geq 1$, every graph G with $\text{tw}(G) > f(n)$ contains an $n \times n$ -grid as a minor.*

As a final piece of the puzzle we observe that high ghw always implies high treewidth in the dual. This observation has been informally mentioned previously, but we are not aware of any formal statement or proof in the literature. Since it is key to our main theorem we provide our own proof in the appendix.

LEMMA 4.6. *Let H be a reduced hypergraph. Then $\text{ghw}(H) \leq \text{tw}(H^d) + 1$.*

THEOREM 4.7. *There exists a function f with the following property: for every $n \geq 1$, every degree 2 hypergraph H with $\text{ghw}(H) > f(n)$ dilutes to the $n \times n$ -jigsaw.³*

PROOF. Let $r: \mathbb{N} \rightarrow \mathbb{N}$ be the function from Theorem 4.5. For the function of the statement it suffices to consider $f: n \mapsto r(n) + 1$. Let H' be a hypergraph with $\text{ghw}(H') > f(n)$, let H be the reduced hypergraph for H' and recall that $\text{ghw}(H) = \text{ghw}(H')$. By Lemma 4.6, we have that $\text{tw}(H^d) > f(n) - 1 = r(n)$ and thus H^d contains a $n \times n$ -grid G_n as minor. By Lemma 4.4, G_n^d is a hypergraph dilution of H and by Lemma 3.6 also of H' . By definition G_n^d is the $n \times n$ -jigsaw J_n and thus J_n is a hypergraph dilution of H' . \square

4.2 From Jigsaw Dilutions to Lower Bounds

It is not difficult to observe that the $n \times n$ -jigsaw has ghw of at least n . This can be seen by observing that since the jigsaw can not be separated by less than n edges it can not be separated into balanced components (that is, components at most half the size of the original hypergraph) by less than n edges. It is known that such balanced separation of a hypergraph H can always be achieved with $\text{ghw}(H)$ edges [3] and hence ghw of the $n \times n$ -jigsaw must be at least n . Moreover, from Lemma 3.2 we can also observe the

³The upper bound for f is inherited from the Grid Exclusion Theorem. The best known bound currently is $f(n) = O(n^9 \text{poly log } n)$ due to Chuzhoy and Tan [11]

opposite direction: a hypergraph H has high ghw if it dilutes to a jigsaw with high dimension (regardless of the degree of H).

We are now ready to combine our main structural results with our reduction for dilutions to derive our lower bound for degree 2 CQ answering.

THEOREM 4.8. *Let \mathcal{H} be a recursively enumerable class of degree 2 hypergraphs with unbounded ghw. Then $\text{p-BCQ}(\mathcal{H})$ is $\text{W}[1]$ -hard under fpt-reductions.*

PROOF. First we observe that if \mathcal{J} is a recursively enumerable class of jigsaws with unbounded dimension, then $\text{p-BCQ}(\mathcal{J})$ is $\text{W}[1]$ -hard. From the above discussion \mathcal{J} has unbounded ghw and thus also unbounded treewidth. Let $\mathcal{Q}_{\mathcal{J}}$ be the class of all self-join free queries with no repeat variables in any atom and hypergraphs in \mathcal{J} . $\mathcal{Q}_{\mathcal{J}}$ then has arity 4 and unbounded semantic treewidth and $\text{p-BCQ}(\mathcal{Q}_{\mathcal{J}})$ is $\text{W}[1]$ -hard under fpt-reductions by Proposition 2.1. Then, by inclusion so is $\text{p-BCQ}(\mathcal{J})$.

Let \mathcal{H}^* be the class of all dilutions of \mathcal{H} . Note that \mathcal{H}^* can still be recursively enumerated. By Theorem 4.7, and the previous observation that an $n \times n$ -jigsaw dilutes to all (modulo isomorphism) jigsaws of lower dimension, \mathcal{H}^* contains the class of all jigsaws and thus $\text{p-BCQ}(\mathcal{H}^*)$ is $\text{W}[1]$ -hard by the argument above. Then by Theorem 3.4 so is $\text{p-BCQ}(\mathcal{H})$. \square

PROOF OF THEOREM 4.1. The implication $3 \Rightarrow 1$ follows directly from Proposition 2.2. The implication $1 \Rightarrow 2$ is immediate. If \mathcal{H} has unbounded ghw, then $\text{p-BCQ}(\mathcal{H})$ is $\text{W}[1]$ -hard by Theorem 4.8. Since we assume that $\text{FPT} \neq \text{W}[1]$, the implication $2 \Rightarrow 3$ follows by contraposition. \square

Theorem 4.1 also has interesting structural consequences. According to Marx [26], $\text{p-BCQ}(\mathcal{H})$ is fixed-parameter tractable if and only if \mathcal{H} has bounded submodular width (subw), assuming the Exponential Time Hypothesis (ETH) [23]. Recall, the ETH is a stronger assumption than $\text{FPT} \neq \text{W}[1]$ in the sense that, if the ETH holds, so does $\text{FPT} \neq \text{W}[1]$. It holds for any hypergraph H that $\text{subw}(H) \leq \text{ghw}(H)$, but the two complexity results imply a previously unknown, and somewhat surprising, equivalence of the two width parameters for degree 2 hypergraphs.

COROLLARY 4.9. *Assume the Exponential Time Hypothesis. Let \mathcal{H} be a class of degree 2 hypergraphs. Then \mathcal{H} has bounded submodular width if and only if it has bounded generalised hypertree width.*

Finding a constructive argument for Corollary 4.9 is an interesting open question in the search for further lower bounds beyond degree 2. We refer to Section 5 for further discussion.

4.3 To Classes of Queries

We can make Theorem 4.1 more fine-grained. Instead of all queries for a class of hypergraphs we can also consider just classes of queries as in Proposition 2.1. See also [10] for the respective extension to Marx' characterisation of fixed-parameter tractability and further discussion of the differences.

As discussed above, it is not clear how to handle hypergraph dilutions on a query level. Consequently, it is also difficult to state an analogue to the reduction in Theorem 3.4 for classes of queries. Instead, we can make use of a more general result by Chen et al.

[10] that relates the complexity of CQ answering over classes of hypergraphs to the complexity of query classes.

PROPOSITION 4.10 (CHEN ET AL. [10]). *Let Q be a class of CQs, let $\text{core}(Q)$ be the class of cores of Q and let $\mathcal{H}^{\text{core}(Q)}$ be the class of hypergraphs of the queries in $\text{core}(Q)$. Then $\text{p-BCQ}(\mathcal{H}^{\text{core}(Q)})$ is fpt-reducible to $\text{p-BCQ}(Q)$.*

There is some ambiguity in what can be considered a degree 2 CQ. The hypergraph of a query can have degree 2 even if variables occur in more than 2 atoms of a query. For example, in the query $R(x, y) \wedge S(x, y) \wedge T(x, z)$, x is in 3 atoms but only in two edges of the hypergraph since the R and S atoms become the same edge. The following results hold also for the more expansive reading, that is, we say that a CQ has degree 2 if its hypergraph has degree 2.

THEOREM 4.11. *Assume $\text{FPT} \neq \text{W}[1]$. Let Q be a recursively enumerable class of degree 2 CQs that does not have bounded semantic generalised hypertree width. Then $\text{p-BCQ}(Q)$ is $\text{W}[1]$ -hard.*

PROOF. Let $\mathcal{H}^{\text{core}(Q)}$ be the class of all hypergraphs of the cores of the queries in Q . It is known that the semantic generalised hypertree width $\text{sem-ghw}(q)$ of a CQ q is precisely $\text{ghw}(\text{core}(q))$ [4]. Thus, if Q has unbounded sem-ghw , $\mathcal{H}^{\text{core}(Q)}$ has unbounded ghw . Recall that the hypergraph of $\text{core}(q)$ is a subhypergraph of the hypergraph of q and thus will also have degree 2. Thus, we can apply Theorem 4.1 and see that $\text{p-BCQ}(\mathcal{H}^{\text{core}(Q)})$ is $\text{W}[1]$ -hard. By Proposition 4.10 the same also holds for $\text{p-BCQ}(Q)$. \square

Note that semantic fractional hypertree width is also equal to the fhw of the core [10] and thus again bounded if and only if sem-ghw is bounded, assuming bounded degree.

The tractability of $\text{BCQ}(Q)$ where Q has bounded sem-ghw is known due to Chen and Dalmau [9]. Thus by analogous argument to Theorem 4.1 we also observe the following extension.

THEOREM 4.12. *Assume $\text{FPT} \neq \text{W}[1]$. Let Q be a class of degree 2 CQs. The following three statements are equivalent:*

- (1) $\text{BCQ}(Q)$ is tractable;
- (2) $\text{p-BCQ}(Q)$ is fixed-parameter tractable;
- (3) Q has bounded semantic generalised hypertree width.

4.4 Counting

Dalmau and Jonsson [12] showed a matching result to Proposition 2.1 for to the corresponding counting problem $\#CQ$. To be precise, by $\#CQ$ we consider the problem of computing $|q(D)|$ for given full CQ q and database D . We also again consider the parameterisation by the query hypergraph $\text{p-}\#CQ$. In this setting. The main result of [12] then reads as follows.

PROPOSITION 4.13 (DALMAU AND JONSSON [12]). *Assume $\text{FPT} \neq \#W[1]^4$. Then for every recursively enumerable class Q bounded arity CQs the following three statements are equivalent.*

- (1) $\#CQ(Q)$ is in FP ;
- (2) $\text{p-}\#CQ(Q)$ is in FPT ;
- (3) Q has bounded treewidth.

⁴By slight abuse of notation we also refer to the class of fixed-parameter polynomial counting problems as FPT when speaking of counting problems

Recall that we only consider full CQs, i.e., queries with no existential quantification. For counting this is an important restriction since Pichler and Skritek [28] show that even for acyclic CQs the problem is $\#P$ -complete in the presence of even a single existentially quantified variable. This restriction also aligns our problem $\#CQ$ with the popular problem of counting homomorphisms when viewing q and D as relational structures. Pichler and Skritek [28] also establish the following upper bound.

PROPOSITION 4.14 (PICHLER AND SKRITEK [28]). *Let Q be a class of CQs with no existential quantification and bounded ghw . Then $\#CQ(Q)$ is in FP .*

Recall the reduction from Theorem 3.4. In the full proof we show that, modulo projection, the result of the reduction produces the exact same results as the original query. Through further inspection of the full proof it is not difficult to verify that even without projection the number of solutions stays the exact same after the reduction, i.e., the reduction is *parsimonious* (cf., [15]).

THEOREM 4.15. *Let \mathcal{H} be a recursively enumerable class of hypergraphs and let \mathcal{M} be a class such that any member is a hypergraph dilution of a hypergraph in \mathcal{H} . Then $\text{p-}\#CQ(\mathcal{M})$ is fixed-parameter parsimonious reducible to $\text{p-}\#CQ(\mathcal{H})$.*

From Proposition 4.13 it is straightforward to derive an analogue of Theorem 4.8 for $\text{p-}\#CQ$. Combining this observation with Theorem 4.15 and Proposition 4.14 we can then also obtain the matching result for the counting problem for full CQs with degree 2 and unbounded arity. In the following we write $\#CQ(\mathcal{H})$ and $\text{p-}\#CQ(\mathcal{H})$, where \mathcal{H} is a class of hypergraphs, for the problems $\#CQ$ and $\text{p-}\#CQ$, respectively, restricted to all full CQs with hypergraph in \mathcal{H} .

THEOREM 4.16. *Assume $\text{FPT} \neq \#W[1]$. Then for every recursively enumerable class \mathcal{H} of degree 2 hypergraphs following three statements are equivalent.*

- (1) $\#CQ(\mathcal{H})$ is in FP ;
- (2) $\text{p-}\#CQ(\mathcal{H})$ is in FPT ;
- (3) \mathcal{H} has bounded generalised hypertree width.

5 ON ARBITRARY BOUNDED DEGREE

The results of the previous section ask a natural next question: what about arbitrary bounded degree? In this section we briefly discuss possible paths towards this goal and give a generalisation of our main structural result to arbitrary fixed degrees.

It is an open question whether Theorem 4.7 holds also under the presence of bounded degree above 2. We can however state the analogous theorem for a generalisation of jigsaw hypergraphs that we will call pre-jigsaws.

Definition 5.1. Let J be a $n \times m$ -jigsaw and H a hypergraph. We say H is a $n \times m$ -pre-jigsaw if there is an mapping $\pi: V(J) \rightarrow V(H)$ and a mapping $o: E(J) \rightarrow 2^{E(H)}$ such that:

- (1) for every two edges $e, f \in E(J)$, $o(e) \cap o(f) = \emptyset$,
- (2) every edge in H is in one image $o(e)$ for some $e \in E(J)$,
- (3) and for two vertices u, v in the same edge e of J , there is a path $P_{u,v}$ from $\pi(u)$ to $\pi(v)$ using only edges in $o(e)$ and no vertices in the image of π other than $\pi(u)$ and $\pi(v)$.

Pre-jigsaws generalise jigsaws in the sense that each single edge e of a jigsaw is replaced by paths between the four vertices in e . Moreover, this “internal” connection of vertices by a jigsaw edge e is replaced only by paths using the edges in $o(e)$. Note also that a jigsaw is also a pre-jigsaw and every degree $2 \times n \times m$ -pre-jigsaw dilutes to a $n \times m$ jigsaw by merging on the vertices in connecting paths from point 3 of Definition 5.1.

However, to obtain Theorem 5.2, our definition of pre-jigsaws makes an important compromise. While the path $P_{u,v}$ for u, v in e from the definition uses only edges in $o(e)$, it is still possible that an edge $f \in E(H)$ with $f \not\subseteq o(e)$ contains a vertex w that is used in the path $P_{u,v}$. This possibility of edges touching other paths is the key technical differences between jigsaws and pre-jigsaws. The merging along the connecting paths to obtain a $n \times m$ -jigsaw from a degree $2 \times n \times m$ -pre-jigsaw noted above is not always possible when the pre-jigsaw has degree greater than 2. Merging on the vertex w in path $P_{u,v}$ and edge f from above would merge edges in $o(e)$ with the edge $f \notin o(e)$, and the resulting hypergraph after merging along paths will not be a jigsaw. Moreover, such edges that touch other paths can also be a source of unbounded arity, which in turn makes it unlikely that we can use Proposition 2.1 directly to derive hardness for important classes of pre-jigsaws. However, even in extreme cases, the structure of pre-jigsaws is not trivial and the fact that certain hypergraphs always dilute to large pre-jigsaws is still significant.

The critical Lemma 4.4 from the degree 2 case does not hold for higher degrees. Through a similar, but much more involved, argument over the dual hypergraph one can still show that high treewidth in the dual hypergraph implies the existence of a large pre-jigsaw. A full proof and further details are available in the extended version of this paper [24].

THEOREM 5.2. *For every $d \geq 1$, there exists a function $f_d: \mathbb{N} \rightarrow \mathbb{N}$ with the following property: for every $n \geq 1$, every hypergraph H with degree d and $\text{ghw}(H) > f_d(n)$ dilutes to an $n \times n$ -pre-jigsaw.*

With respect to finding a characterisation of tractability for the bounded degree case, Theorem 5.2 is only a first step. In general, a hypergraph class \mathcal{H} with unbounded ghw and bounded degree may not contain all pre-jigsaws as dilutions of its members, but only some pre-jigsaws (cf. the proof of Theorem 4.8). Recall that the $n \times n$ -jigsaw dilutes to all lower dimension jigsaws, and therefore a class with degree 2 and unbounded ghw will contain all jigsaws as its dilutions. The same does not hold for pre-jigsaws, introducing further complexity to the bounded degree case. It is therefore of interest whether Theorem 5.2 can be made more precise in terms of showing that specific kinds of pre-jigsaws always exist as dilutions of hypergraphs with high ghw.

Further exploration of Corollary 4.9 may offer an alternative path to the desired result. While the corollary states that submodular width and generalised hypertree width are equivalent under degree 2, the result is observed as a consequence of our complexity results and it remains unclear how to show the equivalence from a structural perspective. A structural argument would likely provide important further insight in the interaction between the two width parameters and may be amenable to a generalisation to bounded degree.

6 CONCLUSION & OUTLOOK

We have proposed hypergraph dilutions as an alternative to graph minors in the study of structural properties of hypergraphs. While the two notions are connected technically, dilutions operate on the hypergraph level and therefore avoid critical issues with graph minors in the presence of arbitrarily large hyperedges. Our study of dilutions yields analogues of the Excluded Grid Theorem and Grohe’s characterisation of tractability for bounded arity CQ answering, for degree 2 hypergraphs. To the best of our knowledge these are the first such results for hypergraphs of unbounded rank.

It remains open whether such a neat delineation of tractable CQ answering even exists under more general circumstances such as bounded degree. In support of this natural next step, we show a generalisation of our main structural result for fixed degree and discuss possible paths to extend the presented results to bounded degree. As an immediate next goal we hope to find a more informative proof of Corollary 4.9, with the eventual goal of better understanding the submodular width of unbounded pre-jigsaws.

Dilutions are closely related to graph minors and our results here rely on key results for graph minors. However, recent thought in graph theory has identified *tangles* as possibly even more fundamental notion of what it means for a graph to be highly connected (e.g., see the discussion in [30]). Adler et al. [3] have previously generalised tangles to *hypertangles* and showed their connection to other hypergraph notions (such as ghw). The further study of tangles in hypergraphs thus presents an interesting alternative direction towards further understanding substructures in hypergraphs.

Finally, we are not aware of a version of Proposition 4.10 for counting, and it is not immediate whether the arguments apply also for counting problems. Extending Theorem 4.16 to classes of queries is left as an open problem. Recently, it has been shown that #CQ is also difficult to approximate [7] under certain conditions. Whether the more elaborate machinery for the approximation case also translates to our setting is a further interesting open question.

ACKNOWLEDGEMENTS

Matthias Lanzinger acknowledges support by the Royal Society project “RAISON DATA” (Project reference: RP\R1201074). The author is grateful to the detailed feedback by anonymous referees which has greatly improved this manuscript. The author would also like to thank Marco Bressan for reporting an error in an earlier draft.

REFERENCES

- [1] Isolde Adler. 2006. *Width functions for hypertree decompositions*. Ph. D. Dissertation. Albert-Ludwigs-Universität Freiburg.
- [2] Isolde Adler, Tomas Gavenciak, and Tereza Klimosová. 2012. Hypertree-depth and minors in hypergraphs. *Theor. Comput. Sci.* 463 (2012), 84–95. <https://doi.org/10.1016/j.tcs.2012.09.007>
- [3] Isolde Adler, Georg Gottlob, and Martin Grohe. 2007. Hypertree width and related hypergraph invariants. *Eur. J. Comb.* 28, 8 (2007), 2167–2181. <https://doi.org/10.1016/j.ejc.2007.04.013>
- [4] Pablo Barceló, Diego Figueira, Georg Gottlob, and Andreas Pieris. 2020. Semantic Optimization of Conjunctive Queries. *J. ACM* 67, 6 (2020), 34:1–34:60. <https://doi.org/10.1145/3424908>
- [5] Manuel Bodirsky and Martin Grohe. 2008. Non-dichotomies in Constraint Satisfaction Complexity. In *Proc. IICALP, Part II (Lecture Notes in Computer Science, Vol. 5126)*. Springer, 184–196. https://doi.org/10.1007/978-3-540-70583-3_16
- [6] Andrei A. Bulatov. 2017. A Dichotomy Theorem for Nonuniform CSPs. In *Proc. FOCS*. IEEE Computer Society, 319–330. <https://doi.org/10.1109/FOCS.2017.37>

- [7] Andrei A. Bulatov and Stanislav Zivný. 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>
- [8] Ashok K. Chandra and Philip M. Merlin. 1977. Optimal Implementation of Conjunctive Queries in Relational Data Bases. In *Proc. ACM Symposium on Theory of Computing*. ACM, 77–90. <https://doi.org/10.1145/800105.803397>
- [9] Hubie Chen and Victor Dalmau. 2005. Beyond Hypertree Width: Decomposition Methods Without Decompositions. In *Proc. CP (Lecture Notes in Computer Science, Vol. 3709)*. Springer, 167–181. https://doi.org/10.1007/11564751_15
- [10] Hubie Chen, Georg Gottlob, Matthias Lanzinger, and Reinhard Pichler. 2020. Semantic Width and the Fixed-Parameter Tractability of Constraint Satisfaction Problems. In *Proc. IJCAI 2020*. ijcai.org, 1726–1733. <https://doi.org/10.24963/ijcai.2020/239>
- [11] Julia Chuzhoy and Zihan Tan. 2021. Towards tight(er) bounds for the Excluded Grid Theorem. *J. Comb. Theory, Ser. B* 146 (2021), 219–265. <https://doi.org/10.1016/j.jctb.2020.09.010>
- [12] Victor 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>
- [13] Ronald Fagin. 1983. Degrees of Acyclicity for Hypergraphs and Relational Database Schemes. *J. ACM* 30, 3 (1983), 514–550. <https://doi.org/10.1145/2402.322390>
- [14] Wolfgang Fischl, Georg Gottlob, Davide Mario Longo, and Reinhard Pichler. 2021. HyperBench: A Benchmark and Tool for Hypergraphs and Empirical Findings. *ACM J. Exp. Algorithmics* 26, Article 1.6 (jul 2021), 40 pages. <https://doi.org/10.1145/3440015>
- [15] Jörg Flum and Martin Grohe. 2004. The Parameterized Complexity of Counting Problems. *SIAM J. Comput.* 33, 4 (2004), 892–922. <https://doi.org/10.1137/S0097539703427203>
- [16] Jörg Flum and Martin Grohe. 2006. *Parameterized Complexity Theory*. Springer. <https://doi.org/10.1007/3-540-29953-X>
- [17] M. R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.
- [18] Georg Gottlob, Matthias Lanzinger, Reinhard Pichler, and Igor Razgon. 2021. Complexity Analysis of Generalized and Fractional Hypertree Decompositions. *J. ACM* 68, 1 (2021).
- [19] Georg Gottlob, Nicola Leone, and Francesco Scarcello. 2002. Hypertree Decompositions and Tractable Queries. *J. Comput. Syst. Sci.* 64, 3 (2002), 579–627. <https://doi.org/10.1006/jcss.2001.1809>
- [20] Georg Gottlob, Zoltán Miklós, and Thomas Schwentick. 2009. Generalized hypertree decompositions: NP-hardness and tractable variants. *J. ACM* 56, 6 (2009), 30:1–30:32. <https://doi.org/10.1145/1568318.1568320>
- [21] 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>
- [22] Martin Grohe and Daniel Marx. 2014. Constraint Solving via Fractional Edge Covers. *ACM Trans. Algorithms* 11, 1 (2014), 4:1–4:20. <https://doi.org/10.1145/2636918>
- [23] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. 2001. Which Problems Have Strongly Exponential Complexity? *J. Comput. Syst. Sci.* 63, 4 (2001), 512–530. <https://doi.org/10.1006/jcss.2001.1774>
- [24] Matthias Lanzinger. 2021. Forbidden Substructures for Tractable Conjunctive Query Answering with Degree 2. *CoRR* abs/2111.11532 (2021). [arXiv:2111.11532](https://arxiv.org/abs/2111.11532)
- [25] Daniel Marx. 2011. Tractable Structures for Constraint Satisfaction with Truth Tables. *Theory Comput. Syst.* 48, 3 (2011), 444–464. <https://doi.org/10.1007/s00224-009-9248-9>
- [26] Daniel Marx. 2013. Tractable Hypergraph Properties for Constraint Satisfaction and Conjunctive Queries. *J. ACM* 60, 6 (2013), 42:1–42:51. <https://doi.org/10.1145/2535926>
- [27] Christos H. Papadimitriou. 2007. *Computational complexity*. Academic Internet Publ.
- [28] 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>
- [29] Neil Robertson and Paul D. Seymour. 1986. Graph minors. V. Excluding a planar graph. *J. Comb. Theory, Ser. B* 41, 1 (1986), 92–114. [https://doi.org/10.1016/0095-8956\(86\)90030-4](https://doi.org/10.1016/0095-8956(86)90030-4)
- [30] Neil Robertson and Paul D. Seymour. 2003. Graph Minors. XVI. Excluding a non-planar graph. *J. Comb. Theory, Ser. B* 89, 1 (2003), 43–76. [https://doi.org/10.1016/S0095-8956\(03\)00042-X](https://doi.org/10.1016/S0095-8956(03)00042-X)
- [31] Dmitry Zhuk. 2020. A Proof of the CSP Dichotomy Conjecture. *J. ACM* 67, 5 (2020), 30:1–30:78. <https://doi.org/10.1145/3402029>

A DEGREE 2 IN PRACTICE

While the contributions of this paper are primarily theoretical and the degree 2 case is viewed as a first step towards possible broader

characterisations, it may be of interest how relevant the degree 2 case is in practice. This is of particular relevance as degree 2 in graphs is highly restrictive, with only line graphs and cycles satisfying the condition. In hypergraphs the situation is different, and much more complex structures can be constructed with degree 2 as was already shown through $n \times n$ -jigsaws or the example in Figure 2.

To offer some further perspective on this question we present some statistics from the HyperBench [14] benchmark. HyperBench consists of collection of hypergraphs from synthetic and real-world CQs and Constraint Satisfaction Problems.

Of the 3649 hypergraphs in HyperBench, 932 have degree 2. Out of these 932 only 16 are obtained from synthetic queries. Furthermore, these hypergraphs are not necessarily simple and a significant number of them have high ghw. Table 1 shows the number of degree 2 hypergraphs with $\text{ghw} > k$ in detail. We see that of the 932 degree 2 hypergraphs, 649 are acyclic ($\text{ghw} > 1$) and almost 400 have ghw even higher than 5.

In summary, this suggests that degree 2 hypergraphs with non-trivial ghw occur naturally in a variety of applications. This may also motivate the study of dilutions to jigsaws as a tool for determining ghw or as a factor in solving degree 2 queries with high width.

Table 1: Number of Degree 2 Hypergraphs in HyperBench with $\text{ghw} > k$

k	amount
1	649
2	575
3	506
4	452
5	389

B ADDITIONAL DETAILS FOR SECTION 3

PROOF OF STATEMENT (3), LEMMA 3.2. We will only argue that for any hypergraph H , merging all incident edges I_v for a vertex v by replacing all edges I_v by a single new edge $e_v = \bigcup I_v \setminus \{v\}$ can not increase ghw. Let us refer to the new hypergraph after the merging as H' . For the other operations the fact that ghw only decreases is well known (see e.g., [18]). The full statement thus follows from proving this case.

Let $\langle T, (B_u)_{u \in T} \rangle$ be a tree decomposition with minimal ghw k for H and associate a λ_u to every $u \in T$ that describes a minimal edge cover in H of each bag. We will now derive new labels $(\lambda'_u)_{u \in T}$ such that for every $u \in T$ we have $|\lambda'_u| \leq |\lambda_u|$ and λ'_u is a set cover of $B_u \setminus \{v\}$ in H' . We will then adapt the bags, such that at least one of them also covers the new edge e_v .

For the appropriate new covers it is enough to set

$$\lambda'_u = \begin{cases} (\lambda_u \setminus I_v) \cup \{e_v\} & \text{if } I_v \cap \lambda_u \neq \emptyset \\ \lambda_u & \text{otherwise} \end{cases}$$

By definition e_v covers the same vertices as all edges in I_v together (in H').

We now move on to defining the bags B'_u of the new decomposition for H' . Let T_v be the subtree $\{u \in T \mid v \in B_u\}$. Then our new bags are defined as follows for every $u \in T$.

$$B'_u = \begin{cases} B_u \cup e_v & \text{if } u \in T_v \\ B_u & \text{otherwise} \end{cases}$$

It is easy to see that all edges of H' are now contained in some bag B'_u . The unchanged edges are still present in the same bag as before, and e_v is in at least one bag, since T_v can not be empty.

To verify connectedness of the newly constructed decomposition it is enough to observe that every edge in I_v will occur fully in some bag in T_v . This is because every must be fully covered by at least one bag. Since all edges in I_v contain the vertex v , this must happen somewhere in T_v . With the updated bags, every vertex $w \in e_v$ now occurs in the union of T_w and T_v . By the observation on I_v above, $T_w \cap T_v \neq \emptyset$ and thus their union is again connected.

What is left is to observe that $B'_u \subseteq \bigcup \lambda'_u$ for every node u . Observe that $v \in B_u$ only if $I_v \cap \lambda_u \neq \emptyset$ since the edges in I_v are the only ones that contain v . Thus, we also have $e_v \subseteq B'_u$ only if $e_v \in \lambda'_u$. All unchanged edges are clearly still covered the same as in the original decomposition, as noted above.

Hence, we see that $\langle T, B'_u \rangle$ is a tree decomposition with ghw at most k , since λ'_u is a witness of set covers with at most k elements for each bag. \square

PROOF OF THEOREM 3.4. Let q be a CQ with hypergraph M_q in \mathcal{M} and D_q a database with the same schema as q . In particular, we assume w.l.o.g. that q has no self-joins. If it did we could reduce to a self-join free q' with database D' in polynomial time by splitting duplicate relation names in q into new individual relation names where the relations in D' are direct copies of the respective original relation in D_q . The hypergraph of such a q' would be the same as the hypergraph of q .

By enumeration of \mathcal{H} , find a hypergraph H such that H dilutes to M_q and let $W = (w_1, \dots, w_\ell)$ be a dilution sequence from H to M_q . Note that H and W depend on M_q , i.e., the parameter of the problem. We will now show that by traversing W in reverse, we can construct (in fixed-parameter polynomial time) a query p such that $\pi_{\text{vars } q}(p(D_p)) = q(D_q)$, and the hypergraph of p is H .

To do so, we will show for each dilution operation w_i – that produces hypergraph H_i from H_{i-1} – how the query q_{i-1} and database D_{i-1} for H_{i-1} can be transformed into an equivalent instance q_i, D_i for H_i with $\pi_{\text{vars}(q_{i-1})}(q_i(D_i)) = q_{i-1}(D_{i-1})$. Thus, ultimately we can reduce from a query for $H_\ell = M_q$ to a query for $H_0 = H$. We also argue for each operation that $\|D_{i-1}\| \leq f(M_q) \|D_i\|$, from which it will become apparent that this is indeed an fpt-reduction.

It will be convenient to observe that the degree never increases along a dilution sequence, i.e., for all $1 \leq i \leq \ell$ it holds that $\text{degree}(H_i) \leq \text{degree}(H_{i-1})$. The observation is easy to verify directly from Definition 3.1. The reduction introduces new constants that will serve to link relations via functional dependence on the new constant. For this purpose, consider the new constants $(\star_i)_{i \geq 0}$ that do not occur in D_q . The final reduction will at most as many constants as the maximum number of tuples in a relation in D_q .

w_i deletes a vertex v from H_{i-1} . While the basic principle of this direction is simple, there are some technicalities that require a certain amount of care. In particular, deleting vertex v can make two edges the same. Hence, reversing the operation is not as straightforward as the deletion. Fortunately, even a very direct approach will be enough for our purposes.

Let E_v be the edges in H_{i-1} that are incident to v . For every edge $e \in E_v$, fix a $\text{pre}(e) \in E(H_i)$ such that $\text{pre}(e) \cup \{v\} = e$. Let $R_{\text{pre}(e)}(\bar{x})$ be the atom in q_i that corresponds to edge $\text{pre}(e)$ in the hypergraph. Then, for each edge e in E_v , create a new atom $S_e(\bar{x}, v)$ in q_i where \bar{x} are the arguments of atom $R_{\text{pre}(e)}(\bar{x})$ and let

$$S_e^{D_{i-1}} = R_{\text{pre}(e)}^{D_i} \times \{(\star_0)\}$$

where the product is interpreted as in relational algebra. The rest of q_i and D_i is made up of direct copies of those atoms/relations that correspond to edges that are in both H_{i-1} and H_i . Since all values in all tuples in the position of the newly introduced joins over v are the same, it is straightforward to observe that $\pi_{\text{vars}(q_{i-1})}(q_i(D_i)) = q_{i-1}(D_{i-1})$.

Let us consider how the size of D_{i-1} is related to the size of D_i . We create at most $\text{degree}(v)$ new relations, where each relation is a relation from D_{i-1} with each tuple extended by a constant. Thus, the representation of such a new relation of S_e increases over the corresponding $R_{\text{pre}(e)}$ only by some constant factor. Hence, overall at most $O(\text{degree}(v) \|D_i\|)$ space (and time) is required to create the new relations. At most the whole previous database is kept, adding at most $\|D_i\|$ size to the new D_{i-1} . Since degree never increases along dilution sequences, $\text{degree}(v) \leq \text{degree}(H)$ and we arrive at our bound of

$$\|D_{i-1}\| = O(\text{degree}(H) \cdot \|D_i\|)$$

w_i replaces the incident edges E of vertex v in H_{i-1} , by a new edge $e = \bigcup E \setminus \{v\}$ in H_i . Let e_1, \dots, e_n be the edges that make up the set E . Let $R(\bar{e})$ be the atom corresponding to edge e in H_i . In q_{i-1} we replace $R(\bar{e})$ by new atoms $R_j(\bar{e}_j)$ for every $j \in [n]$. Let v always be in the last position of the new atoms. To define the new relations, suppose R^{D_i} . Let R' be R^{D_i} extended by a new attribute v , with every tuple extended by a distinct \star_i ($i \leq |R^{D_i}|$) in the new position. Let the new relations in D_{i-1} for the new atoms R_j in q_{i-1} be $R_j^{D_{i-1}} = \pi_{e_j}(R')$. Again everything except R_e is copied directly from q_i, D_i . Since every tuple in R' has a distinct \star_i value for attribute v , R' and, in consequence, every R_j is functionally dependant on v . Since everything else in q_i and D_i remains unchanged we again have $\pi_{\text{vars}(q_{i-1})}(q_i(D_i)) = q_{i-1}(D_{i-1})$.

Clearly, the database can increase in size by no more than if we just copied R' fully n times. Again we see that $n \leq \text{degree}(H_{i-1}) \leq \text{degree}(H)$ and the detailed argument follows the same steps as in the vertex deletion case above.

$$\|D_{i-1}\| \leq c \text{degree}(H) \|D_i\|$$

w_i deletes a subedges $f \subset e$ from H_{i-1} . In this case it is enough to add a new $R_f(\bar{f})$ to q_i to obtain q_{i-1} . The relation is naturally $R_f^{D_{i-1}} = \pi_f(R_e^{D_i})$ and we have the following bound on size of the new database $\|D_{i-1}\| \leq 2 \|D_i\|$. It is straightforward to verify that $q_i(D_i) = q_{i-1}(D_{i-1})$.

Putting it all together. We have shown how to reduce $q_\ell = q$ to $q_0 = p$. The computational effort in each step from i to $i - 1$ consists only of extending relations by one attribute, copying a single relation, or projection, and is feasible in $O(\text{degree}(H)(\|q_i\| + \|D_i\|))$ time. From the bounds on the database size derived for each operation we can deduce the following bound for the final database $D_p = D_0$

$$\|D_p\| = c \text{ degree}(H)^\ell \|D_q\|$$

Since we introduce no self-joins or duplicate variables in the same atom, the size of the final query depends only on the size of H . Recall, H and W , and thus also ℓ , depend only on the parameter M_q . The described process thus reduces q, D_q to p, D_p in $f(M_q)(\|D_q\|)$ time, such that $\pi_{\text{vars}(q)}(p(D_p)) = q(D_q)$. \square

Before we show the NP-completeness, we first show the opposite of Lemma 4.4 as its own statement, and then observe NP-hardness of deciding hypergraph dilutions as consequence of the two lemmas put together.

LEMMA B.1. *Let G be a connected graph and let H a degree 2 hypergraph. If G^d is a hypergraph dilation of H , then G is a minor of H^d .*

PROOF. Suppose G^d is a dilation of H . We will construct an appropriate minor map $\mu: V(G) \rightarrow 2^{E(H)}$ from G into H^d .

For this purpose, suppose we keep track of labels $L(e)$ for the edges of the hypergraphs the dilation process. We set $L(e) = \{e\}$ initially and the labels are then updated as follows, depending on operation. When deleting a vertex collapses multiple edges e_1, \dots, e_ℓ into one edge e_0 we set $L(e_0) = \bigcup_{i=1}^\ell L(e_i)$ and copy the other labels unchanged. When deleting a subedge $e_1 \subset e_0$ we set $L(e_0) = L(e_1) \cup L(e_0)$ and copy any other labels unchanged. Finally, when merging edges I_v over a vertex v , we set the label of the new edge e_v as $L(e_v) = \bigcup_{e \in I_v} L(e)$.

After dilation from H to G^d , we then every edge of G_v associated with a label which is a set of edges of H . Since $E(G^d) = V(G)$, L is thus a function $V(G) \rightarrow 2^{E(H)}$. We claim that L is a minor map, i.e., that every image set $L(e)$ is connected in H^d and any two $L(e_1), L(e_2)$ are disjoint if $e_1 \neq e_2$.

We first observe the disjointness of any two labels in G^d . Note that by construction, all labels are trivially disjoint in H . In every step, every label is either copied unchanged, or multiple labels are combined into a single label. Since the individual parts of this combined label are disjoint with all unchanged labels, so is the combined label.

Connectedness of a set of edges implies connectedness of the respective vertices in the dual hypergraph. Hence, connectedness of a image of the minor map in H^d follows directly from the connectedness of any $L(e)$ in H in G^d . For the merging and subedge deletion operations it is straightforward to see that connectedness is preserved in the construction of the labels. When deleting a vertex, observe that multiple edges collapse into one only if their only difference was vertex v and they are the same otherwise (hence actually $\ell \leq 1$ in the case above). Since they are the same otherwise they are connected via at least one vertex that is not v (they can not both contain only v). Hence, L is a minor map from G into (and actually onto) H^d . \square

PROOF OF THEOREM 3.5. Recall, it is known to be NP-complete to decide whether a graph G is a minor of graph F [17]. We prove NP-hardness of our problem by reduction from graph minor checking. By Lemmas 4.4 and B.1 we have that G is a minor of hypergraph H^d if and only if G^d is a hypergraph dilation of H . The desired reduction then follows from setting $H = F^d$ and observing that the dual of graph F always has degree at most 2.

NP-membership follows from the observation that hypergraph dilutions are, in a sense, *monotonically decreasing*. That is, if H' dilutes to H , then $|V(H)| \leq |V(H')|$, $|E(H)| \leq |E(H')|$, and at least one of the inequalities is strict. Hence, if H is a hypergraph dilation of H' , then there is a linear length dilation sequence from H' to H . Hence, a linear size guess of a dilation sequence leads to an NP algorithm for the problem. \square

C ADDITIONAL DETAILS FOR SECTION 4

PROOF OF LEMMA 4.6. Let $\langle T, (D_u)_{u \in T} \rangle$ be a tree decomposition of H^d with width k . We construct a generalised hypertree decomposition (GHD) $\langle T, (B_u)_{u \in T}, (\lambda_u)_{u \in T} \rangle$ for H by taking for every node u in T , $\lambda_u = D_u$ and $B_u = \bigcup \lambda_u$ (note that the elements of D_u are edges in H). Recall, a GHD is a tree decomposition with an additional labelling $(\lambda_u)_{u \in T}$ that describes an explicit edge cover for each bag.

It is not difficult to verify that this is indeed a GHD of width $k + 1$ of H . To do so we have to argue two properties (the width is trivial). First, that for every $e \in E(H)$, there is a node u such that $e \subseteq B_u$, and second that the connectedness condition holds.

For the first property, consider an arbitrary $e \in E(H)$ as a vertex in H^d . Then, there is some node u such that $e \in D_u$, since we have a tree decomposition of H^d . Then, also $e \in \lambda_u$, and in consequence $e \subseteq B_u = \bigcup \lambda_u$.

For connectedness, consider an arbitrary vertex $v \in V(H)$. Let $f_v \in E(H^d)$ be the edge corresponding to v in the dual. Recall, the elements of $f_v = \{e_1, \dots, e_n\}$ correspond to the edges incident to v in H . Let u be a node in T such that $f_v \subseteq D_u$. Then, by connectedness of the TD, the subtrees $T_{e_i} = \{e_i \in D_u \mid u \in T\}$ for $e_i \in f_v$ are each connected and all contain the node u . Hence, also $T_v = \bigcup_{e_i \in f_v} T_{e_i}$ is connected. By our definition of the bags in the GHD, v occurs exactly in nodes that have an $e_i \in f_v$ in their λ label, i.e., in the nodes of T_v . Thus, we see that connectedness holds for every vertex in the constructed GHD. \square