Forbidden Substructures for Tractable Conjunctive Query Answering 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 the queries’ treewidth when the problem is restricted to queries of bounded arity. We extend this result to cases with unbounded arity and degree 2. To do so, we introduce hypergraph dilutions as an alternative method to prism 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 the bounded degree case.

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 $D$ is 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 tractable fragments of the problem by restricting the structure of queries [18,21]. This line of study has also produced two important characterisations (in terms of query structure) of tractable CQ answering. Grohe [20] showed that BCQ restricted to bounded arity CQs is tractable exactly for queries of bounded tree width. Analogously, Marx [24] 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 [18] and its generalisations [19,21] - there is very 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 [27]. 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. Reduction of a problem into a grid minor will therefore generally lead to relation sizes that are exponential in the arity using Grohe’s technique.

Marx’ characterisation in [24] 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 classes of unbounded submodular width, there are always 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 setting.

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 arity?
(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 always implies the existence of certain highly-connected substructures. This substructure relation, which we call hypergraph dilution, is also connected to the complexity of $p$-BCQ. These observations allow us to follow a similar path as Grohe in the proof of the characterisation for bounded arity in [20] and obtain a first characterisation result for the complexity of unbounded arity CQ answering.

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

\footnote{The 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.}

1
The main contributions in this paper can be 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 all queries of hypergraph dilutions of a hypergraph class \( \mathcal{H} \) is fpt-reducible to CQ answering over all queries of the class \( \mathcal{H} \) itself.

(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 \( \text{glw}(H) \geq f(n) \), contains a jigsaw hypergraph (the hypergraph dual of a grid) as a hypergraph dilution. 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 reach approach this goal. Moreover, we give a generalisation of the key structural result to the bounded degree case.

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 beyond the two previously mentioned characterisation results. One important exception is work by Marx [23] 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, 29]. 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 glw 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 glw 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 \( \{1, 2, \ldots, n\} \). 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 [25] and [15] 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 \( \text{P} \).

Hypergraphs. A hypergraph \( H \) is a pair \( (V(H), E(H)) \) where \( V(H) \) is a set of vertices and \( E(H) \subseteq 2^{V(H)} \) is a set of hyperedges. 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 \( H \) as 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 \( |H^d| = |H| \).

A path between two distinct vertices \( v_0, v_\ell \) in \( H \) is a sequence \((v_0, v_1, \ldots, v_\ell)\) alternating between vertices \( v_i \) and edges \( e_i \) 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 is 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 \( u, v \in V(G) \), \( \mu(u) \cap \mu(v) = \emptyset \), (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 if a minor map is onto, i.e., \( V(F) = \bigcup_{v \in V(G)} \mu(v) \).

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 \((T, (B_u)_{u \in T})\) 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
We refer to this width as the dilution sequence of operations as a \( q \) (\text{sem-ghw}(q)) function, we use semantic treewidth to refer to the minimal treewidth in the respective equivalence class. Note that semantic width is also commonly referred to as \textit{width modulo homomorphism} in the literature since CQ equivalence coincides with homomorphic equivalence of queries. For full details and related definitions see [4].

Commonly, the definition of CQs also allows for (top-level) existential quantification of variables, effectively restricting the solutions only to assignments over the free variables. In the context of this paper such quantification is of little consequence and all results for BCQ and p-BCQ hold also with existential quantification. Some care is required in the results for the counting problem and we discuss the special situation there directly in the respective Section 4.4.

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 glw is equivalent to hypertree width up to a constant factor [3], and the other showing tractability of BCQ under bounded hypertree width [18].

**Proposition 2.1 (Theorem 1.1, Grohe [20]).** Assume \( \text{FPT} \neq W[1] \). Let \( Q \) be a class of bounded arity 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 treewidth.

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

**Proposition 2.2 (Adler et al. [3], Gottlob et al. [18]).** Let \( Q \) be a class of CQs with bounded \( \text{ghw} \). Then \( \text{BCQ}(Q) \) 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 \( H' \), then \( H \) should be “simpler” than \( H' \). The difficulty of course lies in the question of what makes a hypergraph simpler than another. We do not claim to have an answer to this question and, moreover, do not propose that there is a singular “correct” kind of simplicity. Rather, the generality of hypergraphs suggests that competing notions will be of interest in different settings. Later in this section we present some evidence that in the context of CQs, hypergraph dilutions capture such simplicity 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. a merging on \( v \) replacing all of the incident edges \( E_v \) of vertex \( v \), by a new edge \( \{ \cup E_v \} \setminus \{v\} \).

We also say that \( H' \) dilutes to \( H \) and refer to the associated sequence of operations as a \textit{dilution sequence} from \( H \) to \( H \).
Important, 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, in the inverse, 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 will be important in our further studies. The first two statements in 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 glhw 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 full proof of the third statement is given in the appendix.

**Lemma 3.2.** For hypergraphs \(H\) and \(H'\) such that \(H'\) dilutes to \(H\). Then the following statements hold:

1. \(\deg(H) \leq \deg(H')\)
2. \(|V(H)| + |E(H)| < |V(H')| + |E(H')|\)
3. \(\text{glhw}(H) \leq \text{glhw}(H')\)

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,
4. or adding a hyperedge \(e\) such that the vertices of \(e\) induce a clique in the primal graph.

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 can not be simulated by hypergraph minors or vice versa. 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, roughly speaking, CQ answering can only become easier along the order of hypergraph dilutions. This is intuition is stated formally in Theorem 3.4 below. The second key observation is that under bounded degree, high glhw 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 \(\text{fpt}\)-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 \(W = (w_1, \ldots, w_t)\). 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 \(\pi_{\text{vars}(q_{i-1})}(q_i(D_i)) = q_{i-1}(D_{i-1})\). Thus by traversing \(W\) in reverse, we arrive at an instance \(p, D_p\) with hypergraph \(H_0 = H\) such \(\pi_p(D_p) = q(D_q)\). Intuitively, this can be done by introducing keys in the database for the new positions introduced when reversing a merging on a vertex \(x\), 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 instance size increases only by a factor of at most \(\deg(H)\) in each step. Hence, we observe \(\|D_p\| = O(\deg(H))\|D_q\|\) and analogous time bounds for the reduction, where \(H\) and \(t\) both depend on the parameter \(M_q\).

It may seem natural to extend Definition 3.3 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...
Defining 3.3 should be adapted to operate directly on queries. Consider the following example query $R(x, y, z) \land R(x, u, v) \land S(u, z)$. Consider the case analogous to deleting vertex $v$ in the corresponding hypergraph. The atom $R(x, y, z)$ should remain unchanged 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 dilation 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 only 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 dilation 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 FPT ≠ W[1]. Let $\mathcal{H}$ be a class of hypergraphs with degree 2. The following three statements are equivalent:

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 $W[1]$-hard.

In general, there are tractable classes of BCQ that have 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 fhw if and only if $\mathcal{H}$ has bounded ghw [17]. Theorem 4.1 can therefore equivalently also be stated in terms of fractional hypertree width (or just hypertree width).

Strengthening Theorem 4.1 to the main result stated in the introduction is subject of Section 4.3. The analogous characterisation result for the problem of counting the number of solutions of a CQ is discussed in Section 4.4.

### 4.1 The Structure of Hypergraphs with Degree 2 and Unbounded ghw

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 $|e_{i,j} \cap e_{i,j+1}| = 1$ and $|e_{i,j} \cap e_{i+1,j}| = 1$ for $i < n, j < m$ and no other pair of edges has a non-empty intersection. Equivalently, the $n \times m$ jigsaw is the dual hypergraph of the $n \times m$-grid graph.

The $n \times m$-jigsaw is uniquely determined up to isomorphism. Figure 2 illustrates a $3 \times 3$-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).

The following Example 4.3 illustrates a dilation of a hypergraph to the $3 \times 2$-jigsaw. Our first goal in this section will be to show that it is always possible to dilute a degree 2 hypergraph $H$ to a $n \times n$-jigsaw where $n$ depends on ghw$(H)$.

**Example 4.3.** Figure 3 illustrates an example dilation of a hypergraph with degree 2 to a 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.

We will first observe that grid 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 $H$ 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 $\phi: E(H) \to V(H^d)$ be the bijection from edges in $H$ to their corresponding vertex in the dual, and let $\mu: V(G) \to 2^V(H^d)$ 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 $\delta(u)$ and $\delta(v)$. Since $c_{u,v}$ has degree 2, it is therefore connected to only one edge in $\delta(u)$. Let $C_u$ be the set of all vertices $c_{u,v}$ for all $v$ adjacent to $u$ in $G$. Since $\delta(u)$ is connected and all vertices in $C_u$ are elements of only one edge in $\delta(u)$, it follows that there is a minimal hitting set $\tau_u$ of $\delta(u)$ such that either $\tau_u \cap C_u = \emptyset$, or $\delta(u)$ consists only of one edge that contains exactly the vertices $C_u$. Since $\tau_u$ is a minimal hitting set, the edge merging operation from Definition 3.3 can be applied along $\tau_u$ vertices to merge all edges in $\delta(u)$ into a single edge. Note that in the case where there is no $\tau_u$, $\delta(u)$ already contains only a single edge and nothing has to be done.

Let $H_1$ be the hypergraph obtained by merging, for every $u \in V(H)$, all $\delta(u)$ in this way and generating a single new merged edge $e_u$ from $\delta(u)$ (where necessary). By construction $H_1$ is 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 $C$ have been removed by the merging process.

Finally, let $H_2$ be the subhypergraph $H_1[C]$, i.e., the hypergraph obtained from $H_1$ by deleting all vertices not in $C$. Observe that for every edge $e \in E(G^d)$, 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 $v_{u,v}$ 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 edges and vertices 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, we see that $H_2$ is isomorphic to $G^d$ and a hypergraph dilution of $H$.

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 [24] for further intuition.

PROPOSITION 4.5 (ROBERTSON AND SEYMOUR [27]). There exists a function $f: \mathbb{N} \to \mathbb{N}$ 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 ghw($H$) $\leq$ 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 ghw($H$) $>$ $f(n)$ dilutes to the $n \times n$-jigsaw.

Proof. Let $r: \mathbb{N} \to \mathbb{N}$ 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 ghw($H'$) $>$ $f(n)$, let $H$ be the reduced hypergraph for $H'$ and recall that ghw($H$) = ghw($H'$). By Lemma 4.6 we have that 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 a $n \times n$-jigsaw $J_n$ and thus $J_n$ is a hypergraph dilution of $H'$.

4.2 From Jigsaw Dilutions to Lower Bounds

It is not difficult to observe that the $n \times n$-jigsaw has ghw at least $n$. This can be seen by observing that since the jigsaw cannot 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 ghw($H$) edges [5] and hence ghw of the $n \times n$-jigsaw must be at least $n$. Moreover, from Lemma 3.2 it then follows that a hypergraph $H$ has high ghw if it dilutes to a large jigsaw (regardless of the degree of $H$).

\[\text{The upper bound for } f \text{ is inherited from the Grid Exclusion Theorem. The best known bound currently is } f(n) = O(n^9 \text{ poly log } n) \text{ due to Chuzhoy and Tan [11].} \]
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 \( \text{ghw} \). Then \( p\text{-BCQ}(\mathcal{H}) \) is \( W[1] \)-hard under \( \text{fpt} \)-reductions.

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

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, \( \mathcal{H}' \) contains the class of all jigsaws and thus \( p\text{-BCQ}(\mathcal{H}') \) is \( W[1] \)-hard by the argument above. Then by Theorem 3.4 so is \( p\text{-BCQ}(\mathcal{H}) \).

**Proof of Theorem 4.1.** The implication \( 3 \Rightarrow 1 \) follows from Proposition 2.2 and \( 1 \Rightarrow 2 \) is trivial. If \( \mathcal{H} \) has unbounded \( \text{ghw} \), then \( p\text{-BCQ}(\mathcal{H}) \) is \( W[1] \)-hard by Theorem 4.8. Since we assume that \( \text{FPT} \neq W[1] \), the implication \( 2 \Rightarrow 3 \) follows by contraposition.

Theorem 4.1 also has interesting structural consequences. According to Marx [24], \( p\text{-BCQ}(\mathcal{H}) \) is fixed-parameter tractable if and only if \( \mathcal{H} \) has bounded submodular width (subw), assuming the Exponential Time Hypothesis (ETH) [22]. Recall, the ETH is a stronger assumption than \( \text{FPT} \neq W[1] \) in the sense that, if the ETH holds, so does \( \text{FPT} \neq W[1] \). 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 strengthen Theorem 4.1 to be 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. In consequence 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 \( p\text{-BCQ}(\mathcal{H}^{\text{core}(Q)}) \) is \( \text{fpt-reducible to} \) \( p\text{-BCQ}(Q) \).

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 \( R(x, y) \land S(x, y) \land T(x, z) \), \( x \) 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 \( \text{FPT} \neq W[1] \). Let \( Q \) be a recursively enumerable class of degree 2 CQs that does not have bounded semantic generalised hypertree width. Then \( p\text{-BCQ}(Q) \) is \( W[1] \)-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 \( \text{sem-ghw} \), \( \mathcal{H}^{\text{core}(Q)} \) has unbounded \( \text{ghw} \). Recall that the hypergraph of \( \text{core}(q) \) is a subhypergraph of the hypergraph of \( q \) and thus will also have degree 2. Thus, by Theorem 4.1 \( p\text{-BCQ}(\mathcal{H}^{\text{core}(Q)}) \) is \( W[1] \)-hard, and by Proposition 4.10 so is \( p\text{-BCQ}(Q) \).

Note that semantic fractional hypertree width is also equal to the flow of the core [10] and thus again bounded if and only if \( \text{sem-ghw} \) is bounded, assuming bounded degree.

The tractability of \( \text{BCQ}(Q) \) where \( Q \) has bounded \( \text{sem-ghw} \) 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 \( \text{FPT} \neq W[1] \). Let \( Q \) be a class of degree 2 CQs. The following three statements are equivalent:

1. \( \text{BCQ}(Q) \) is tractable.
2. \( p\text{-BCQ}(Q) \) 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 the corresponding counting problem \( \#Q \). To be precise, by \( \#Q \) we consider the problem of computing \( |q(D)| \) for given CQ \( q \) and database \( D \). We also again consider the parameterisation by the query hypergraph \( p\#Q \). In this setting, the main result of [12] then reads as follows.

**Proposition 4.13 (Dalmau and Jonsson [12]).** Assume \( \text{FPT} \neq \#W[1]^3 \). Then for every recursively enumerable class \( Q \) bounded arity \( CQ \)s the following three statements are equivalent:

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

Recall that we only consider CQs with no existential quantification. For counting this is an important restriction since Pichler and Skritek [26] 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 Skrîtek [26] also establish the following upper bound.

**Proposition 4.14 (PicHLer and Skrîtek [26]).** Let $Q$ be a class of CQs with no existential quantification and bounded glw. Then $\#CQ(\mathcal{Q})$ is in FP.

Recall the reduction from Theorem 3.4. In the argument we showed 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., [14]).

**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 $p\cdot\#CQ(\mathcal{M})$ is fixed-parameter parsimonious reducible to $p\cdot\#CQ(\mathcal{H})$.

From Proposition 4.13 it is straightforward to derive an analogue of Theorem 4.8 for $p\cdot\#CQ$. Combining this with Theorem 4.15 and Proposition 4.14 we can then also obtain the matching result for the counting problem in our setting.

**Theorem 4.16.** Assume 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. $p\cdot\#CQ(\mathcal{H})$ is in FPT.
3. $\mathcal{H}$ has bounded generalised hypertree width.

Where $\#CQ(\mathcal{H})$ and $p\cdot\#CQ(\mathcal{H})$ are the respective problems restricted to the class of all CQs with hypergraph in $\mathcal{H}$ and no existential quantification.

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 $\pi : V(J) \to V(H)$ and a mapping $\sigma : E(J) \to 2^{E(H)}$ such that:

1. for every two edges $e, f \in E(J)$, $\sigma(e) \cap \sigma(f) = \emptyset$,
2. every edge in $H$ is in one image $\sigma(e)$ for some $e \in E(J)$,
3. and for two vertices $u, v$ in the same edge $e$ of $J$ there is a path $p_{uv}$ from $\pi(u)$ to $\pi(v)$ using only edges in $\sigma(e)$ and no vertices in the image of $\pi$ 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 disjoint paths between the four vertices in $e$. That is, the “internal” connection of vertices by a jigsaw edge $e$ is replaced by the edges in $\sigma(e)$. Furthermore, every vertex in a pre-jigsaw contributes either directly to this internal connection, or corresponds to a vertex of the jigsaw. 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 the sets $\sigma(e)$ for every edge. In general, while paths are unique in pre-jigsaws, it is still possible that an edge in a path contains a vertex that is used in some other path. Dilution along the paths would thus merge the two paths and therefore decrease the dimension of the resulting jigsaw. Touching of other paths is the key technical issue in transferring results for jigsaws to pre-jigsaws. We argue that there study may still have merit as a first step to deeper insight into the structure of hypergraphs with high glw and bounded degree.

In particular, 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. The full proof is given in the appendix.

**Theorem 5.2.** For every $d \geq 1$, there exists a function $f_d : \mathbb{N}^2 \to \mathbb{N}$ with the following property. For every $n \geq 1$, every hypergraph $H$ with degree $d$ and $\text{glw}(H) > f_d(n)$ dilutes to a $n \times n$-pre-jigsaw.

Hence, showing that $p$-BCQ($\text{J}_{\text{pre}}$) is not fixed-parameter tractable when $\text{J}_{\text{pre}}$ is a class of bounded degree pre-jigsaws of unbounded dimension would be enough to extend our main result to bounded degree. However, pre-jigsaws no longer have bounded arity, hence W[1]-hardness no longer follows from Proposition 2.1.

Instead of explicitly constructing a reduction, it may be feasible to apply Marx’ characterisation for the bounded arity case to pre-jigsaws. In particular, a lower bound for $p$-BCQ($\text{J}_{\text{pre}}$) can not be in FPT if the class $\text{J}_{\text{pre}}$ has unbounded submodular width (assuming the ETH). By Corollary 4.9 we know that submodular width and generalised hypertree width are equivalent under degree 2. A more structural proof of the statement may be open to a generalisation to certain bounded degree pre-jigsaws, or even to bounded degree hypergraphs in general.

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 results presented here 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 [28]). Adler et al. [3] have previously generalized 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, through related methods, 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.

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 Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008. Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations (Lecture Notes in Computer Science, Vol. 5126), Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz (Eds.). 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 50th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017, Chris Umans (Ed.). IEEE Computer Society, 319–330. https://doi.org/10.1109/FOCS.2017.37
[7] 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/10.1145/3383930
[8] Ashok K. Chandra and Philip M. Merlin. 1977. Optimal Implementation of Functional Dependencies. J. ACM 24, 6 (1977), 22–38. https://doi.org/10.1145/360967.360983
[9] Hubie Chen. 2020. A Proof of the CSP Dichotomy Conjecture. J. ACM 67, 5 (2020), 30:1–30:78. https://doi.org/10.1145/3402029
[10] Dániel 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
[11] Dmitriy Zhuk. 2020. A Further Details for Section 3 Proof of Statement (3), Lemma 3.2. We will only argue that for any hypergraph H, merging all incident edges I₀ of a vertex v by replacing all edges I₀ with a single new edge e_v = I₀ \ {v} cannot 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., [17]). The full statement thus follows from proving this case.

Let (T, (Bᵥ)ᵥ∈T) be a tree decomposition with minimal ghw k for H and associate a λ_u to every u ∈ T that describes a minimal edge cover in H of each bag. We will now derive new labels (λ_u')ᵥ∈T such that for every u ∈ T we have λ_u' ≤ λ_u and λ_u' is a set cover of B_u \ {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

\[ λ_u' = \begin{cases} λ_u \setminus I₀ \cup \{e_v\} & \text{if } I₀ \cap λ_u \neq \emptyset \\ λ_u & \text{otherwise} \end{cases} \]

By definition e_v covers the same vertices as all edges in I₀ together (in H').

We now move on to defining the bags B_u' of the new decomposition for H'. Let T₀ be the subtree \{u ∈ T \mid v ∈ B_u\}. Then our new bags are defined as follows for every u ∈ T.

B_u' = \begin{cases} B_u \cup e_v & \text{if } u \in T₀ \\ 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, and $e_{0}$ is in at least one bag, since $T_{0}$ can not be empty.

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 $e_{0} \subseteq B'_{u}$ only if $e_{0} \in \lambda'_{u}$. All unchanged edges are clearly still covered as noted above.

Hence, we see that $(T, B'_{u})$ is a tree decomposition with $ghw$ at most $k$, since $\lambda'_{u}$ is a witness of set covers with at most $k$ elements for each bag.

**Proof of Theorem 3.4.** Let $q$ be a CQ with hypergraph $M_{q}$ in $M$ 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 $W = (w_{1}, \ldots, w_{p})$ 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 $\pi_{vars}(q)(p(D_{p})) = q(D_{p})$, and the hypergraph of $p$ is $H$.

To do so, we will show for each dilution operation $w_{j}$ that produces hypergraph $H_{j}$ from $H_{j-1}$ - how the query $q_{j-1}$ and database $D_{j-1}$ for $H_{j-1}$ can be transformed into an equivalent instance $q_{j}, D_{j}$ for $H_{j}$ with $\pi_{vars}(q_{j-1})(q_{j}(D_{j})) = q_{j-1}(D_{j-1})$. Thus, ultimately we can reduce from a query for $H_{H} = M_{q}$ to a query for $H_{0} = H$. We also argue for each operation that $\|D_{j-1}\| \leq f(M_{q}) \|D_{j}\|$, 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 $t \in [\ell]$ it holds that degree($H_{t}$) $\leq$ degree($H_{t-1}$). 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 $(\star)_{i \geq 0}$ 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 $D_{q}$.

$w_{1}$ deletes a vertex $v$ from $H_{1-1}$. 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 $H_{1-1}$ that are incident to $v$. For every edge $e \in E_{v}$, fix a pre$(e) \in E(H_{1})$ such that pre$(e) \cup \{v\} = e$. Let pre$(e)(\bar{x})$ be the atom in $q_{t}$ that corresponds to edge pre$(e)$ in the hypergraph. Then, for each edge $e \in E_{v}$, create a new atom $S_{e}(\bar{x}, \bar{v})$ in $q_{t}$ where $\bar{x}$ are the arguments of atom pre$(e)(\bar{x})$ and let

$$S_{e}^{D_{j-1}} = R^{D_{j}}_{pre(e)} \times \{\star\}$$

where the product is interpreted as in relational algebra. The rest of $q_{t}$ and $D_{t}$ is made up of direct copies of those atoms/relations that correspond to edges that are in both $H_{t-1}$ and $H_{t}$. 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 $\pi_{vars}(q_{t-1})(q_{t}(D_{t})) = q_{t-1}(D_{t-1})$.

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

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

$w_{1}$ replaces the incident edges $E$ of vertex $v$ in $H_{1-1}$, by a new edge $e = \bigcup E \setminus \{v\}$ in $H_{1}$. Let $e_{1}, \ldots, e_{n}$ be the edges that make up the set $E$. Let $R(e)$ be the atom corresponding to edge $e$ in $H_{1}$. In $q_{t-1}$ we replace $R(e)$ by new relations $R_{i}(e_{i})$ for each $i \in [n]$. Let $v$ always be in the last position of the new atoms. To define the new relations, suppose $R^{D_{j}}$. Let $R'$ be $R^{D_{j}}$ extended by a new attribute $w$, with every tuple extended by a distinct $\star_{i} (i \leq [R^{D_{j}}])$ in the new position. Let the new relations in $D_{j-1}$ for the new atoms $R_{i}$ in $q_{t-1}$ be $R_{i}^{D_{j-1}} = R_{i}(R')$. Again everything except $R_{i}$ is copied directly from $q_{t-1}, D_{t-1}$. Since every tuple in $R'$ has a distinct $\star_{i}$ value for attribute $v$, $R'$ and, in consequence, every $R_{i}$ is functionally dependant on $v$. Since everything else in $q_{t-1}$ and $D_{t-1}$ remains unchanged we again have $\pi_{vars}(q_{t-1})(q_{t}(D_{t})) = q_{t-1}(D_{t-1})$.

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

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

$w_{1}$ deletes a subedges $f \subset e$ from $H_{1-1}$. In this case it is enough to add a new $R_{f}(f)$ to $q_{t}$ to obtain $q_{t-1}$. The relation is naturally $R_{f}^{D_{j-1}} = R_{f}(R^{D_{j}})$ and we have the following bound on size of the new database $\|D_{j-1}\| \leq 2 \|D_{j}\|$. It is straightforward that $q_{t}(D_{t}) = q_{t-1}(D_{t-1})$.

**Putting it all together.** We have shown how to reduce $q_{t} = q$ to $q_{0} = p$. The computational effort in each step from $t$ to $t-1$ consists only of extending or copying relations by one attribute and projection, and is feasible in $O(\text{degree}(H)(\|q_{t}\| + \|D_{t}\|))$ time. From the bounds on the database size derived for each operation we can deduce the following bound for the final database $D_{p} = D_{0}$

$$\|D_{p}\| = c \text{degree}(H)^{f} \|D_{q}\|$$

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 $\ell$, depend only on the parameter $M_{q}$. The described process thus reduces $q, D_{0}$ to $p, D_{p}$ in $f(M_{q})(\|D_{q}\|)$ time, such that $\pi_{vars}(q)(p(D_{p})) = q(D_{p})$.

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 A.1.** Let $G$ be a connected graph and let $H$ a degree 2 hypergraph. If $G^d$ is a hypergraph dilution of $H$, then $G$ is a minor of $H^d$.

**Proof.** Suppose $G^d$ is a dilution of $H$. We will construct an appropriate minor map $μ: V(G) → 2^{E(H)}$ from $G$ into $H^d$.

For this purpose, suppose we keep track of labels $L(e)$ for the edges of the hypergraphs the dilution process. At the start set $L(e) = \{e\}$, and the labels are then updated as follows, depending on operation. When deleting a vertex collapses multiple edges $e_1, \ldots, e_t$ into one edge $e_0$ we set $L(e_0) = \bigcup_{i=1}^{t} L(e_i)$ and copy the other labels unchanged. When deleting a subedge $e_1 \subseteq e_0$ we set $L(e_1) = L(e_1) \cup L(e_0)$ and copy any other labels unchanged. Finally, when merging edges $e_0$ over a vertex $v$, we set the label of the new edge $e_0$ as $L(e_0) = \bigcup_{e \subseteq e_0} L(e)$.

After dilution from $H$ to $G^d$, we then every edge of $G^d$ associated with a label which is a set of vertices in $H$. Since $E(G^d) = V(G)$, $L$ is thus a function $V(G) → 2^{E(H)}$. 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 $H$. 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 $t \leq 1$ 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) $H^d$. $\Box$

**Proof of Theorem 3.5.** Recall, it is known to be NP-complete to decide whether a graph $G$ is a minor of graph $F$ [16]. We prove NP-hardness of our problem by reduction from graph minor checking. By Lemmas 4.4 and A.1 we have that $G$ is a minor of hypergraph $H^d$ if and only if if $G^d$ is a hypergraph dilution 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 $|V(H')| ≤ |V(H)|$, $|E(H')| ≤ |E(H)|$, and at least one of the inequalities is strict. Hence, if $H$ is a hypergraph dilution of $H'$, then there is a linear length dilution sequence from $H'$ to $H$. Hence, a linear size guess of a dilution sequence leads to a NP algorithm for the problem. $\Box$

**B FURTHER DETAILS FOR SECTION 4**

**Proof of Lemma 4.6.** Let $(T, (D_v)_{v \in T})$ be a tree decomposition of width $k$ of $H^d$. We then construct a generalised hypertree decomposition (GHD) $(T, (B_v)_{v \in T}, (\lambda_u)_{u \in T})$ 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 $e \subseteq B_u$, 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 $f_v = \{e_1, \ldots, e_t\}$ 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 $T_v = \{e_i \in D_u \mid u \in T\}$ for $e_i \in f_v$ are each connected and all contain the node $u$. Hence, also $T_v = \bigcup e_i f_v T_v$ is connected. By our definition of the bags in the GHD, $v$ occurs exactly in the nodes that have an $e_i \in f_v$ in their $\lambda$ label, i.e., in the nodes of $T_v$. Thus, we see that for every vertex, connectedness holds in the constructed GHD. $\Box$

**C PROOFS FOR SECTION 5**

We will again show the property via observations of graph minors in the dual. In particular, we will extend the notion of a graph minor to hypergraphs in a way that takes hyperedges into account. We will call such minors expressive minors. We show how such expressive grid minors are related to normal grid minors via the rank of the hypergraph. From there we will then see that for classes of bounded degree, the dual hypergraphs also always contain large expressive grid minors if their treewidth is unbounded.

**Definition C.1 (Expressive Minor Map).** Let $G$ be a graph and $H$ a hypergraph. We say that a mapping $μ: V(G) → 2^{V(H)}$ is an expressive minor map from $G$ into $H$ if $μ$ is a minor map from $G$ onto $H$ and there exists a mapping $ρ: E(G) → E(H)$ such that:

1. $ρ$ is injective, i.e., no two edges in $G$ are mapped to the same edge in $H$.
2. For any edge $e = \{u, v\}$ in $G$, $ρ(e)$ connects $μ(u)$ and $μ(v)$ in $H$.
3. For any two incident edges $e_1, e_2$ in $G$, there is a path from $ρ(e_1)$ to $ρ(e_2)$ that uses no edge in $ρ(E(G))$ (except for $ρ(e_1)$ and $ρ(e_2)$ as the start/end).

We say $G$ is an expressive minor of $H$ if there exists an expressive minor map from $G$ into $H$.

**Note** that if $H$ is a simple graph, i.e., if the rank of all edges is 2, then every minor is an expressive minor.

**Lemma C.2.** Let $H$ be a reduced hypergraph and let $m = 4 \text{rank}(H)^6 n^5$. If the $n \times m$ grid is a minor of the primal graph of $H$, then the $n \times n$-grid is an expressive minor of $H$. 
We start from the observation that in $G$, we consider blocks $B_{k,l}$ which contain the vertices $v_{k,l}$ of the grid where $(k-1) \cdot 4^a n^4 \leq i \leq k \cdot 4^a n^4$ and $(\ell - 1) \cdot 4^a n^4 \leq j \leq \ell \cdot 4^a n^4$.

Let $G_n$ now be a $n \times n$-grid and define $\mu_n : V(G) \to 2^{V(H)}$ as $\mu_n(v_{k,l}) = \bigcup_{u \in B_{k,l}} \mu_m(u)$ for all $k \in [n], \ell \in [n]$. It is straightforward to observe that $G_n$ is a minor of $G_m$ and therefore $G_n$ is also a minor of $H$. Since vertices in $G_n$ correspond one-to-one to blocks in $G_m$, we also write $B_n$ for the block in $G_n$ corresponding to vertex $v$ in $G$. We now construct an appropriate mapping $\rho : E(G) \to E(H)$ to show that $\rho_n$ is indeed an expressive minor. To simplify further discussion, we refer to the edges in $E(H)$ that are mapped to by $\rho$ as marked edges.

The most challenging part on constructing $\rho$ correctly is to respect the third condition of Definition C.1. From the grid minor we have knowledge of paths between different edges, in particular if there is a path from edges $u \to v \in G$, then there is a path from any edge touching $\mu(u)$ to any edge touching $\mu(v)$ in $H$ since the images of a minor map are always connected and adjacency in $G$ implies adjacency of the respective images in $H$. However, avoiding marked edges complicates the situation as not every path in the grid necessarily has an analogue in $H$ without using marked edges. We say that paths in the grid are restricted by marked edges if they do not imply a corresponding path in $H$ that uses no restricted edges.

There are two ways in which marked edges can block paths in the grid. If some edge $(u, v)$ of the grid is satisfied by some marked $e \in E(H)$ that touches both $\mu_m(u)$ and $\mu_m(v)$, then this edge in the grid cannot be used to construct paths between edges in $H$, since it is not guaranteed that this connection exists without the marked edge $e$. Alternatively, a problem can arise if two vertices in a marked $e$ are in the same image $\mu_m(v)$ for some $v \in G$. Then it is not clear whether the connectedness of $\mu_m(v)$ relies on $e$ and thus no paths using such vertices can be used to reconstruct paths in $H$.

The strategy of our construction will now be as follows. For each two adjacent blocks in $G_m$, we will choose one of the edges that connect the two blocks in $G_m$ as marked. Suppose we make a choice for horizontally adjacent blocks. We want to chose the edge in the $i$-th row of the block only if no edge or vertex in the $i$-th row of the block is restricted by some other marked edges. For vertical connections our goal is analogue for columns instead of rows. Since all pairs of rows and columns in a block intersect, this property then implies the existence of non-restricted paths between any pair of marked edges of a block.

Now, to see that such choices of connecting edges always exist we start from the observation that in $G_m$, each block has $4^a n^4$ edges that connect to each adjacent block. Every edge of $H$ contains at most $a^2$ vertices and can therefore also block at most $\binom{4^a n^4}{2} a^2$ edges or $a^2$ vertices in $G_m$. Note that if a vertex is restricted by an edge, then two vertices in the edge are in image of the same $v \in V(G_m)$. In this situation the number of edges in the grid that can be restricted is also reduced and we can thus use the simpler bound of $a^2$ objects in $G_m$ per marked edge.

It follows that there are at least $4^a n^4$ distinct edges $E(H)$ that connect (the image of) each block to (the image of) an adjacent block. That is, for each edge in $G_m$, there are at least $4^a n^4$ possible distinct choices of map to in $H$. For simplicity, we associate each edge in $E(H)$ with just one row/column in each block even though it possible contains covers multiple edges in the transition.

As discussed above, every marked edge blocks at most $a^2$ edges or vertices. Thus each marked edge can destroy our desired property, that no edges or vertices of a row are restricted, in at most $a^2$ rows. Since we can make no statement on which blocks the rows are restricted in we overestimate and say that $a^2$ rows and columns are restricted in every block, by every marked edge.

We can now argue that the $4^a n^4$ possible choices per edge in $G_n$ are enough to establish our $\rho$ as intended. For our argument we give a procedure that constructs such a $\rho$. Fix some ordering $O = (e_1, e_2, \ldots, e_\ell)$ on the edges of $G_m$. In a first pass, in order of $O$, we select a set of $\beta_i = 2a^2 n^2$ edges (we call this a bundle for the $i$-th edge in $O$) for each edge in $G_m$. For every choice made in step $i$, block all the rows/columns where some edge or vertex is restricted by some edge in the chosen bundle $\beta_i$. In every future step $j > i$, corresponding to edge $e_j = (u, v)$, the bundle $\beta_j$ can contain only edges associated with rows/columns in blocks $B_{u,2}$ and $B_{2,v}$ that have not been restricted yet. Thus, when choosing bundle $\beta_j$, there are at most $2a^2 n^2$ restricted rows and columns in each block.

In the last step, since $\ell \leq n^2$, there are at most $2a^2 n^2 (n^2 - 1)$ restricted rows and columns in each block. And since we consider only the two blocks corresponding to the two vertices in the final edge $e_\ell$, only $4a^4 n^4 - 2a^2 n^2$ choices are restricted for $\beta_\ell$ and thus clearly a choice of $2a^2 n^2$ edges is still available from the initial $4a^4 n^4$ choices. Clearly, then for all previous edges in $O$ it was also still possible to make a choice satisfying the requirements.

This first phase establishes "forward" consistency with restricted rows and columns. What is left, is now filter down the bundles to single edges such that the choices are consistent overall. This can be achieved by repeating the same idea in reverse direction of $O$. Note that in step $i$ of the reverse order corresponds to step $\ell - i + 1$ in the forward iteration over $O$ in the first phase. We iterate in reverse order of $O$. In the step $i$, we select a $p(e_{\ell-i+1}) \in betu_{\ell-i+1}$ that is associated to rows/columns that have not been marked as restricted in the second phase, and after the choice, we mark all rows/columns as restricted where some vertex or edge is restricted by $p(e_{\ell-i+1})$. In this phase, after step $i$ we have fixed $i$ choices and thus restricted at most $ia^2$ rows and columns. Thus, when arriving at the last step we again restricted no more than $(n^2 - 1)a^2$ rows/columns. Since there are $2a^2 n^2$ choices in every bundle we see by the same argument as above that choices corresponding to unrestricted rows/columns can be made at every step of the second phase.

What is left to do is to show that for every edge $e = (u, v)$ in $G$, our final $p(e)$ now indeed is always associated only to rows or columns that are not restricted by any other edge fixed in $\rho$. Let $I$ be the index in $O$ of the edge $e$ and suppose it was associated with a row restricted by $p(e_j)$ for some $e_j \in O$ with $j < i$. Since $p(e) \in \beta_i$, it follows that an edge of the bundle $\beta_i$ was chosen despite an associated row being marked as restricted by the previous bundle $\beta_j$ that contains $p(e_j)$ in the first phase, a contradiction.
if \( j > i \) then \( \rho(e_j) \) would have restricted an associated row in \( B_u \) or \( B_v \) in the second phase, before the choice for \( \rho(e_i) \) was made. \( \square \)

In the following Theorem C.3 we also show a further property that may be of interest and can be required from pre-jigsaws without sacrificing the statement of the theorem. Namely, for a pre-jigsaw \( H \) and underlying jigsaw \( J \), we can fix one path \( P_{u, v} \) for every pair \( u, v \) of vertices occurring together in an edge of \( J \) such that every vertex in \( V(H) \) is either in the image of \( \pi \) or occurs in one of the fixed paths.

**Theorem C.3.** There exists a function \( f : \mathbb{N}^2 \rightarrow \mathbb{N} \) with the following property. For every \( n \geq 1 \), every bounded rank hypergraph \( H \) with \( tw(H) > f(n, rank(H)) \) contains the \( n \times n \)-grid as an expressive minor.

**Lemma C.4.** If the \( n \times n \)-grid is an expressive minor of \( H^d \), then there exists an \( n \times n \)-pre-jigsaw \( P \) such that \( H \) dilutes to \( P \).

**Proof.** First, we assume w.l.o.g. that \( H \) no isolated vertices, no empty edges, and no duplicate vertex types. It is straightforward to observe that such vertices and edges can not contribute to the minor mapping of \( G \) into \( H^d \) in any meaningful way. Note that these assumptions are only made in the argument for sake of simplicity and the statement of the lemma still holds in full generality since the assumed properties can always be enforced through a simple dilution sequence (cf. Lemma 3.6).

Let \( G \) be the \( n \times n \)-grid and let \( J \) be the corresponding \( n \times n \)-jigsaw dual to \( G \). Consider an expressive minor map \( \mu : V(G) \rightarrow 2^{V(H^d)} \), which we assume w.l.o.g. to be onto. Let \( \rho : E(G) \rightarrow E(H^d) \) be the mapping as in Definition C.1.

We will now consider the dualisations of mappings \( \mu \) and \( \rho \), that is we consider domains and co-domains with the respective dual mappings applied. In particular, let \( \pi : V(G^d) \rightarrow V(H) \) such that \( \pi(x) = \rho(x) \), and analogously let \( o \) the respective dualisation \( E(G^d) \rightarrow 2^{E(H)} \) of \( \mu \). Alternatively, we see that \( \pi \) is a mapping \( V(J) \rightarrow V(H) \) and \( o \) is \( E(J) \rightarrow 2^{E(H)} \).

For the dilution to \( P \), first fix the disjoint paths (as in Definition C.1) for every pair \( u, v \) of vertices that occur together in an edge in \( J \). Let \( C \) be the set of all vertices that (explicitly) occur in any fixed path. To obtain \( P \) we delete all vertices that are neither in \( C \) nor \( \pi(V(J)) \) and delete any empty edges that are created in the process. Clearly, all necessary paths for the pre-jigsaw conditions are preserved in \( P \). To adapt \( o \) to edges of \( P \) we take
\[
o' : e \mapsto \{ f \cap V(P) \mid f \in o(e) \} \setminus \emptyset
\]

It is straightforward to see that any \( o'(e) \) is still connected by the paths connecting the pairs of vertices in \( e \). Since \( o \) is onto, so is \( o' \) and in extension \( o' \), i.e., every edge of \( P \) is in some image of \( o' \). It is then no difficult to verify the other properties of a pre-jigsaw are as they correspond to dualised properties of expressive minors. \( \square \)

**Proof of Theorem 5.2.** Let \( H \) be a hypergraph with degree \( d \). Let \( f_d \) be the function from Theorem C.3 with the second parameter fixed to \( d \). Note that \( H^d \) has rank \( d \) and thus by Theorem C.3 contains a \( n \times n \)-grid minor. By Lemma C.4 \( H \) then also dilutes to a pre-jigsaw. \( \square \)