Fast Matrix Multiplication meets the Subdmodular Width

MAHMOUD ABO-KHAMIS, RelationalAI, USA
XIAO HU, University of Waterloo, Canada
DAN SUCIU, University of Washington, USA

One fundamental question in database theory is the following: Given a Boolean conjunctive query Q, what is the best complexity for computing the answer to Q in terms of the input database size N? When restricted to the class of combinatorial algorithms, the best known complexity for any query Q is captured by the *submodular width* of Q [3, 4, 17]. However, beyond combinatorial algorithms, certain queries are known to admit faster algorithms that often involve a clever combination of fast matrix multiplication and data partitioning. Nevertheless, there is no systematic way to derive and analyze the complexity of such algorithms for arbitrary queries Q.

In this work, we introduce a general framework that captures the best complexity for answering any Boolean conjunctive query Q using matrix multiplication. Our framework unifies both combinatorial and non-combinatorial techniques under the umbrella of information theory. It generalizes the notion of submodular width to a new stronger notion called the ω-*submodular width* that naturally incorporates the power of fast matrix multiplication. We describe a matching algorithm that computes the answer to any query Q in time corresponding to the ω-submodular width of Q. We show that our framework recovers the best known complexities for Boolean queries that have been studied in the literature, to the best of our knowledge, and also discovers new algorithms for some classes of queries that improve upon the best known complexities.

CCS Concepts: • **Theory of computation → Database query processing and optimization (theory)**; • **Mathematics of computing → Information theory**; • **Computing methodologies → Linear algebra algorithms**.

Additional Key Words and Phrases: join algorithms; fast matrix multiplication; information theory; Shannon inequalities

1 INTRODUCTION

We focus on the problem of answering a *Boolean conjunctive query* Q. In particular, we have a set $\mathsf{vars}(Q)$ of variables (or attributes) and a set $\mathsf{atoms}(Q)$ of relations where each relation $R(\boldsymbol{X}) \in \mathsf{atoms}(Q)$ is over a variable set $\boldsymbol{X} \subseteq \mathsf{vars}(Q)$. In particular, each relation $R(\boldsymbol{X})$ is a list of satisfying assignments to the variables $\boldsymbol{X}$, and the query Q asks whether there exists an assignment to all variables $\mathsf{vars}(Q)$ that simultaneously satisfies all relations in $\mathsf{atoms}(Q)$:

$$Q() \text{ :- } \bigwedge_{R(\boldsymbol{X}) \in \mathsf{atoms}(Q)} R(\boldsymbol{X}) \tag{1}$$

We assume the query to be fixed, hence its size is a constant, and we measure the runtime in terms of the total size of the input relations, denoted by N, i.e., we use *data complexity*. For brevity, throughout the paper, we refer to a Boolean conjunctive query as just "query".

Using only combinatorial algorithms, the best known complexity for any query Q is given by the *submodular width* [3, 4, 17]. However, when fast matrix multiplication is allowed, some isolated queries admit faster algorithms, but there is no general framework to derive such algorithms for any query. In this paper, we introduce such a framework that naturally unifies both combinatorial and non-combinatorial techniques under the umbrella of *information theory*. In particular, we generalize the submodular width to incorporate matrix multiplication and develop a matching algorithm. We show that our general algorithm matches or improves upon the best known custom algorithms for queries that have been studied in the literature.

1.1 Background

1.1.1 Combinatorial Join Algorithms. We start with some background on combinatorial algorithms for queries. When restricted to the class of combinatorial algorithms, there are only three basic techniques that are sufficient to recover the best-known complexity over this class of algorithms for any query Q:

- *For-loops*: Worst-case optimal join (WCOJ) algorithms [18, 19], like GenericJoin [20] or LeapFrog-TrieJoin [22], can be viewed as a sequence of nested for-loops, each of which iterates over possible assignments of one variable. For example, consider the Boolean triangle query:
$$Q_\triangle() \text{ :- } R(X,Y) \wedge S(Y,Z) \wedge T(X,Z) \tag{2}$$
One possible WCOJ algorithm consists of a for-loop over the intersection of X-values from R and T, and for each such assignment, a for-loop over the intersection of Y-values from R and S, and for each such assignment, a for-loop over the intersection of Z-values from S and T. This simple algorithm gives a runtime of $O(N^{3/2})$ for this query and $O(N^{\rho^*(Q)})$ in general, where $\rho^*(Q)$ is the *fractional edge cover number* of Q, which is also an upper bound on the join size [7, 8, 15].
- *Tree Decompositions (TDs)*: Sometimes, two nested loops can be (conditionally) independent of one other. For example, consider the query:
$$Q_{\triangle\triangle}() \text{ :- } R(X,Y) \wedge S(Y,Z) \wedge T(X,Z) \wedge S'(Y,Z') \wedge T'(X,Z') \tag{3}$$
We could solve it in time $O(N^2)$ using 4 nested for-loops over X, Y, Z, and Z' in order. However, note that once we fix the values of X and Y, the two inner loops over Z and Z' become independent, hence can be unnested. One way to capture and utilize such conditional independence is using the framework of *tree decompositions*, which are a form of query plans. In this example, we could break down the query using a tree decomposition, or *TD* for short, consisting of two "bags" (i.e. two subqueries in the query plan) where one bag corresponds to a triangle query over $\{X,Y,Z\}$ while the other bag is a triangle query over $\{X,Y,Z'\}$, thus leading to a runtime of $O(N^{3/2})$. Using TDs (alongside for-loops), we can answer any query Q in time $O(N^{\mathsf{fhtw}(Q)})$ where $\mathsf{fhtw}(Q)$ is the *fractional hypertree width* of Q [2, 15, 16].
- *Data Partitioning*: For-loops and TDs alone are not sufficient to unleash the full power of combinatorial join algorithms for all queries. Consider the following *4-cycle query*:
$$Q_\square() \text{ :- } R(X,Y) \wedge S(Y,Z) \wedge T(Z,W) \wedge U(W,X) \tag{4}$$
$Q_\square$ admits two TDs: one with two bags $\{X,Y,Z\}$ and $\{Z,W,X\}$, while the other with two bags $\{Y,Z,W\}$ and $\{W,X,Y\}$. However, using either TD alone, we cannot achieve a runtime better than $O(N^2)$. On the other hand, if we partition the input relations carefully into multiple parts, and select a proper TD for each part, we could achieve a runtime of $O(N^{3/2})$ [6]. Partitioning is done based on the "degrees" of relations where we think of a (binary) relation like $R(X,Y)$ as a bipartite graph, and compute degrees of vertices accordingly. Taking the partitioning approach to the extreme[1], the PANDA algorithm [3, 4] can solve any query Q in time $\tilde{O}(N^{\mathsf{subw}(Q)})$ where $\mathsf{subw}(Q)$ is the *submodular width* of Q [17].[2]

The submodular width is a single definition that combines the above three techniques, and the corresponding PANDA algorithm achieves (up to a polylogarithmic factor) the best-known complexity for any query Q over combinatorial algorithms. The submodular width and PANDA have a deep connection to information theory. At a very high level, the submodular width of a

[1]By that, we mean partitioning not just the input relations but also intermediate relations that result from the join of (input or intermediate) relations. This "multi-level" partitioning can indeed lower the complexity further than one-level partitioning, for some classes of queries.

[2]$\tilde{O}$ hides a polylogarithmic factor in N.

given query Q can be thought of as aiming to capture the best complexity of answering Q using the following meta-algorithm: Think of (binary) input relations, like $R(X, Y)$ above, as bipartite graphs, and partition each of them into (a polylogarithmic number of) parts that are almost "uniform", i.e. where all vertices within the same part have roughly the same degree. Now each part of the data can be described by its combination of degrees, called "degree configuration". For each degree configuration, we pick the best TD, go over its bags, and solve the corresponding subqueries, using for-loops.[3] Instead of reasoning about degree configurations directly, we model them as *edge-dominated polymatroids*, or *ED-polymatroids* for short, which is an information-theoretic concept. In particular, given a degree configuration, the corresponding polymatroid is roughly the entropy of a certain probability distribution over the join of input relations having the given degree configuration. (Formal Definition will be given in Sec. 3.) Using polymatroids as a proxy to degree configurations allows us to transform a database problem into an information-theoretic problem. To sum up, the submodular width has the following skeleton (the formal definition will be given later):

$$\text{subw}(Q) \stackrel{\text{def}}{=} \underbrace{\max_{\text{ED-polymatroid } \boldsymbol{h}}}_{\substack{\text{worst part}\\ \text{of the data}}} \quad \underbrace{\min_{\text{tree decomposition } T}}_{\substack{\text{best query plan}\\ \text{for this part}}} \quad \underbrace{\max_{\text{bag } B \in T}}_{\substack{\text{worst subquery}\\ \text{in the plan}}} \quad \underbrace{h(B)}_{\substack{\text{subquery cost}\\ \text{(using for-loops)}}} \tag{5}$$

1.1.2 Beyond Combinatorial Join Algorithms. For certain queries, there are known non-combinatorial algorithms with lower complexity than the best known combinatorial algorithms. In addition to the three techniques mentioned above, these non-combinatorial algorithms typically involve a fourth technique, which is *matrix multiplication*, or MM for short. For background, given two $n \times n$ matrices, there are algebraic algorithms that can multiply them in time $o(n^3)$. The *matrix multiplication exponent* ω is the smallest exponent α where this multiplication can be done in time $O(n^{\alpha+o(1)})$. It was first discovered by Strassen [21] that $\omega < 3$, and to date, the best known upper bound for ω is 2.371552 [23]. For certain queries, incorporating MM can lead to faster algorithms by first partitioning the data based on degrees, and then for each part, we choose to either perform an MM or use a traditional combinatorial algorithm (consisting of a TD and for-loops). Which choice is better depends on the degree configuration of the part. For parts with low degrees, combinatorial algorithms are typically better, while parts with high degrees tend to benefit from MM. The complexities of such algorithms typically involve ω. For example, for the triangle query $Q_\triangle$, there is a non-combinatorial algorithm with complexity $O(N^{\frac{2\omega}{\omega+1}})$ [6].[4] However, such non-combinatorial algorithms are only known for some isolated queries; see Table 1 for a summary of known results. There is no general framework for answering any query Q using MM.

1.2 Our Contributions

In this paper, we make the following contributions: (Recall that a "query" refers to a Boolean conjunctive query.)

- We introduce a generalization of the submodular width of a query Q, called the *ω-submodular width of Q*, and denoted by ω-subw(Q), that naturally incorporates the power of matrix multiplication. The ω-submodular width is always upper bounded by the submodular width, and becomes identical when $\omega = 3$.

[3]Note that this algorithm uses the three techniques mentioned above in reverse order.

[4]Note that for $\omega = 3$, this complexity collapses back to $O(N^{3/2})$ which is the same as the combinatorial algorithm.

Query	Best Prior Algorithm	Our Algorithm
Arbitrary query Q	$\tilde{O}(N^{\mathsf{subw}(Q)})$ [3, 4]	$\tilde{O}(N^{\omega\text{-}\mathsf{subw}(Q)})$
Triangle $Q_\triangle$ (Eq. (2))	$O\left(N^{\frac{2\omega}{\omega+1}}\right)$ [5]	same
4-Clique	$O\left(N^{\frac{\omega+1}{2}}\right)$ [9]	same
5-Clique	$O\left(N^{\frac{\omega}{2}}+1\right)$ [9]	same
k-Clique ($k \geq 6$)	$O\left(N^{\overline{\omega}(\frac{1}{2}\cdot\lceil\frac{k}{3}\rceil,\frac{1}{2}\cdot\lceil\frac{k-1}{3}\rceil,\frac{1}{2}\cdot\lfloor\frac{k}{3}\rfloor)}\right)$ [13]	$\tilde{O}\left(N^{\frac{1}{2}\cdot\lceil\frac{k}{3}\rceil+\frac{1}{2}\cdot\lceil\frac{k-1}{3}\rceil+\frac{1}{2}\cdot\lfloor\frac{k}{3}\rfloor\cdot(\omega-2)}\right)$ (same for $\omega = 2$)
4-Cycle $Q_\square$ (Eq. (4))	$\tilde{O}\left(N^{\frac{4\omega-1}{2\omega+1}}\right)$ [10, 24]	same
k-Cycle	$\tilde{O}\left(N^{\bar{c}_k}\right)$ [10, 24]	$\tilde{O}\left(N^{c_k^\square}\right)$ (same for $\omega = 2$)
k-Pyramid (Eq. (31))	$\tilde{O}\left(N^{2-\frac{1}{k}}\right)$ [3, 4]	$\tilde{O}\left(N^{2-\frac{2}{\omega(k-1)-k+3}}\right)$

Table 1. A summary of prior results and the corresponding results obtained by our framework. $\overline{\omega}(a,b,c)$ is the smallest exponent for multiplying two rectangular matrices of sizes $n^a \cdot n^b$ and $n^b \cdot n^c$ within $O\left(n^{\overline{\omega}(a,b,c)}\right)$ time. In contrast, $\omega^\square(a,b,c)$ is the smallest upper bound on $\overline{\omega}(a,b,c)$ that is obtained through *square* matrix multiplication. In particular, $\omega^\square(a,b,c) \stackrel{\text{def}}{=} \max\{a+b+(\omega-2)c, a+(\omega-2)b+c, (\omega-2)a+b+c\}$; see Sec. 3. Obviously, $\overline{\omega}(a,b,c) \leq \omega^\square(a,b,c)$. Moreover, this becomes an equality when $\omega = 2$ or when $a = b = c$; see Sec. 3. The symbol $\bar{c}_k$ is the best-known exponent for detecting cycles using rectangular matrix multiplication [10, Theorem 1.3], while $c_k^\square$ is the smallest upper bound on $\bar{c}_k$ that is obtained through *square* matrix multiplication; see Eq. (43) and (45). By definition, $\bar{c}_k \leq c_k^\square$, and this becomes an equality when $\omega = 2$. Moreover, this is an equality when k is odd as well as $k = 4$ or 6; see [10].

- We introduce a general framework to compute any query Q in time $\tilde{O}(N^{\omega\text{-}\mathsf{subw}(Q)})$ for any rational[5] ω, where $\tilde{O}$ hides a polylogarithmic factor in N. Our framework unifies known combinatorial and non-combinatorial techniques under the umbrella of *information theory*.
- We show that for any query Q, our framework recovers the best known complexity for Q over *both* combinatorial *and* non-combinatorial algorithms. See Table 1.
- We show that there are classes of queries where our framework discovers *new* algorithms with strictly lower complexity than the best-known ones. See Table 1.

1.3 Paper Outline

In Section 2, we present a high-level overview of our framework and illustrate it using the triangle query $Q_\triangle$. In Section 3, we give formal definitions for background concepts. We formally define the ω-submodular width in Section 4, and show how to compute it for several classes of queries in Section 5. In Sections 6 and 7, we give an algorithm for computing the ω-submodular for any query Q and an algorithm for computing the answer to any query Q in ω-subdmodular width time, while deferring some technical details to the appendix. We conclude in Section 8.

2 OVERVIEW

We give here a simplified overview of our framework. We start with how to generalize the subdmodular width to incorporate MM, and to that end, we need to answer two basic questions:

[5]When ω is not rational, we can take any rational upper bound.

- Q1: How can we express the complexity of MM in terms of polymatroids?
- Q2: How can we develop a notion of query plans that naturally reconciles TDs with MM?

2.1 Q1: Translating MM complexity into polymatroids

Given a query Q of the form (1), a polymatroid is a function $\boldsymbol{h} : 2^{\text{vars}(Q)} \rightarrow \mathbb{R}_+$ that satisfies *Shannon inequalities*. By that, we mean $\boldsymbol{h}$ is *monotone* (i.e. $h(\boldsymbol{X}) \leq h(\boldsymbol{Y})$ for all $\boldsymbol{X} \subseteq \boldsymbol{Y} \subseteq \text{vars}(Q)$), *submodular* (i.e. $h(\boldsymbol{X}) + h(\boldsymbol{Y}) \geq h(\boldsymbol{X} \cup \boldsymbol{Y}) + h(\boldsymbol{X} \cap \boldsymbol{Y})$ for all $\boldsymbol{X}, \boldsymbol{Y} \subseteq \text{vars}(Q)$), and satisfies $h(\emptyset) = 0$. A polymatroid is *edge-dominated* if $h(\boldsymbol{X}) \leq 1$ for all $R(\boldsymbol{X}) \in \text{atoms}(Q)$. The subdmodular width, given by Eq. (5), uses edge-dominated polymatroids as a proxy for the degree configurations of different part of the data.

Throughout the paper, we assume that ω is a fixed constant within the range $[2, 3]$. Given two rectangular matrices of dimensions $n^a \times n^b$ and $n^b \times n^c$, we can multiply them by partitioning them into square blocks of dimensions $n^d \times n^d$ where $d \stackrel{\text{def}}{=} \min(a, b, c)$ and then multiplying each pair of blocks using square matrix multiplication in time $n^{d \cdot \omega}$. This leads to an overall runtime of $n^{\omega^{\square}(a,b,c)}$ where $\omega^{\square}(a, b, c)$ is defined below and $\gamma \stackrel{\text{def}}{=} \omega - 2$: (Proof is in Sec. 3.)

$$\omega^{\square}(a, b, c) \stackrel{\text{def}}{=} \max\{a + b + \gamma \cdot c, \quad a + \gamma \cdot b + c, \quad \gamma \cdot a + b + c\} \tag{6}$$

Now suppose we have two relations $R(X, Y)$ and $S(Y, Z)$, and we want to compute $P(X, Z)$:- $R(X, Y) \wedge S(Y, Z)$, by viewing R and S as two matrices of dimensions $n^a \times n^b$ and $n^b \times n^c$ respectively and multiplying them. In the polymatroid world, we can think of $h(X)$, $h(Y)$, and $h(Z)$ as representing a, b, and c, respectively. Motivated by this, we define the following new information measure:

$$\text{MM}(X; Y; Z) \stackrel{\text{def}}{=} \max(h(X) + h(Y) + \gamma h(Z), \quad h(X) + \gamma h(Y) + h(Z), \quad \gamma h(X) + h(Y) + h(Z)) \tag{7}$$

And now, we can use $\text{MM}(X; Y; Z)$ to capture the complexity of the above MM, on log-scale. We also extend the MM notation to allow treating multiple variables as a single dimension. For example, given the query $Q_{\triangle\triangle}$ from Eq. (3), we use $\text{MM}(X; Y; ZZ')$ to refer to the cost of MM where we treat Z and Z' as a single dimension. In particular, we view $R(X, Y)$ as one matrix and $S(Y, Z) \wedge S'(Y, Z')$ as another matrix, and multiply them to get $P(X, Z, Z')$.

2.2 Q2: Query plans for MM

Variable Elimination [11, 12, 25, 26] is a language for expressing query plans that is known to be equivalent to TDs [2] for combinatorial join algorithms. We show that variable elimination can be naturally extended to incorporate MM in its query plans. For background, given a query Q, *variable elimination* refers to the process of picking an order $\boldsymbol{\sigma}$ of the variables vars(Q), known as *variable elimination order*, or *VEO* for short, and then going through the variables in order and "eliminating" them one-by-one. Eliminating a variable X from a query Q means transforming Q into an equivalent query Q' that doesn't contain X, and this is done by removing all relations that contain X and creating a new relation with all variables that co-occurred with X. (Formal Definition will be given in Sec. 3.) For example, given the 4-cycle query $Q_{\square}$ from Eq. (4), we can eliminate Y by computing a new relation $P(X, Z)$:- $R(X, Y) \wedge S(Y, Z)$ and now the remaining query becomes a triangle query:

$$Q'_{\square}() \text{ :- } P(X, Z) \wedge T(Z, W) \wedge U(W, X)$$

We use $U_Y^{\boldsymbol{\sigma}}$ to refer to the set of variables involved in the subquery that eliminates Y, which is $\{X, Y, Z\}$ in this example. Every VEO is equivalent to a TD, and vice versa [2]. In this example, any VEO that eliminates either Y or W first is equivalent to a TD with two bags $\{X, Y, Z\}$ and $\{Z, W, X\}$, whereas all remaining VEOs are equivalent to the other TD described before.

In the presence of MM, VEOs become more expressive. For example, the triangle query $Q_\triangle$ has only one trivial TD with a single bag $\{X, Y, Z\}$. Now, suppose we have a VEO that eliminates Y first. There are two different ways to eliminate Y:

- Either compute the full join combinatorially using for-loops, and then project Y away. This computation costs $h(XYZ)$ on log-scale.
- Or view $R(X, Y)$ and $S(Y, Z)$ as matrices and multiply them to get $P(X, Z)$. This costs $\mathsf{MM}(X; Y; Z)$.

Alternatively, we could have eliminated either X or Z first. In this simple example, there is only a single way to eliminate a variable, say Y, using MM, but in general, there could be several, and we can choose the best of them. For example, in the query $Q_{\triangle\triangle}$ from Eq. (3), Y occurs in three relations, and we can arrange them into two matrices in different ways. One way is to join $S(Y, Z)$ and $S'(Y, Z')$ into a single matrix $S''(Y, ZZ')$ and multiply it with the matrix $R(X, Y)$ leading to a cost of $\mathsf{MM}(X; Y; ZZ')$. Alternatively, we could have obtained costs $\mathsf{MM}(XZ; Y; Z')$ or $\mathsf{MM}(XZ'; Y; Z)$, and we will see later that there are even more options! [6] Given a VEO $\boldsymbol{\sigma}$ and a variable Y, we use $\mathsf{EMM}^{\boldsymbol{\sigma}}_Y$ to denote the minimum cost of eliminating Y using MM. In contrast, $h(U^{\boldsymbol{\sigma}}_Y)$ is the cost of eliminating Y using for-loops. For example, in $Q_\triangle$, $\mathsf{EMM}^{\boldsymbol{\sigma}}_Y = \mathsf{MM}(X; Y; Z)$ and $h(U^{\boldsymbol{\sigma}}_Y) = h(XYZ)$, whereas in $Q_{\triangle\triangle}$, $\mathsf{EMM}^{\boldsymbol{\sigma}}_Y = \min(\mathsf{MM}(X; Y; ZZ'), \mathsf{MM}(XZ; Y; Z'), \mathsf{MM}(XZ'; Y; Z), \ldots)$ and $h(U^{\boldsymbol{\sigma}}_Y) = h(XYZZ')$.

2.3 Defining the ω-submodular width

Putting pieces together, we are now ready to define our notion of ω-subdmodular width of a query Q. We take the maximum over all ED-polymatroids $\boldsymbol{h}$, and for each polymatroid, we take the minimum over all VEOs $\boldsymbol{\sigma}$. For each $\boldsymbol{\sigma}$, we take the maximum elimination cost over all variables X, where the elimination cost of X is the minimum over all possible ways to eliminate X using either for-loops or MM:

$$\omega\text{-subw}(Q) \stackrel{\text{def}}{=} \underbrace{\max_{\text{ED-polymatroid } \boldsymbol{h}}}_{\substack{\text{worst part}\\ \text{of the data}}} \underbrace{\min_{\text{VEO } \boldsymbol{\sigma}}}_{\substack{\text{best query plan}\\ \text{for this part}}} \underbrace{\max_{\text{variable } X}}_{\substack{\text{worst variable}\\ \text{elimination cost}}} \min\Big(\underbrace{h(U^{\boldsymbol{\sigma}}_X)}_{\substack{\text{cost of eliminating}\\ X \text{ using for-loops}}}, \underbrace{\mathsf{EMM}^{\boldsymbol{\sigma}}_X}_{\substack{\text{cost of eliminating}\\ X \text{ using MM}}} \Big) \tag{8}$$

To compare the above to the submodular width, we include below an alternative definition of the submodular width that is equivalent to Eq. (5). The equivalence follows from the equivalence of VEOs and TDs [2]:

$$\text{subw}(Q) \stackrel{\text{def}}{=} \underbrace{\max_{\text{ED-polymatroid } \boldsymbol{h}}}_{\substack{\text{worst part}\\ \text{of the data}}} \underbrace{\min_{\text{VEO } \boldsymbol{\sigma}}}_{\substack{\text{best query plan}\\ \text{for this part}}} \underbrace{\max_{\text{variable } X}}_{\substack{\text{worst variable}\\ \text{elimination cost}}} \underbrace{h(U^{\boldsymbol{\sigma}}_X)}_{\substack{\text{cost of eliminating}\\ X \text{ using for-loops}}} \tag{9}$$

The only difference between Eq. (8) and Eq. (9) is the inclusion of $\mathsf{EMM}^{\boldsymbol{\sigma}}_X$ in the former. This shows that ω-subw(Q) is always upper bounded by subw(Q). We show later that they become identical when $\omega = 3$.

For example, $Q_\triangle$ has 6 different VEOs. For a fixed VEO, the cost of eliminating the first variable dominates the other two, thus we ignore the two. We have seen before that the cost of eliminating

[6] In particular, later we will extend the notion of MM to allow for *group-by variables*, and we will also generalize the concept of VEOs to allow eliminating *multiple variables* at once.

Y is $\min(h(XYZ), \mathsf{MM}(X;Y;Z))$, which also happens to be the cost of eliminating either X or Z.[7] Hence, the ω-subdmodular width becomes:

$$\omega\text{-subw}(Q_\triangle) = \max_{\text{ED-polymatroid } \boldsymbol{h}} \min\big(h(XYZ), \mathsf{MM}(X;Y;Z)\big) \tag{10}$$

2.4 Computing the ω-submodular width

Now that we have defined the ω-submodular width, our next concern is how to compute it for a given query Q. The ω-submodular width is a deeply nested expression of min and max. (Recall that $\mathsf{EMM}^{\sigma}_{X}$ is a minimum of potentially many terms of the form $\mathsf{MM}(X;Y;Z)$, each of which is a maximum of three terms in Eq. (7).) To compute ω-subw(Q), we first pull all max operators outside by distributing min over max[8], and then swap the order of the max operators so that the max over $\boldsymbol{h}$ is the inner most max. Applying this to Eq. (10), we get:

$$\begin{aligned}\omega\text{-subw}(Q_\triangle) = \max\Big(&\max_{\text{ED-polymatroid } \boldsymbol{h}} \min(h(XYZ), h(X) + h(Y) + \gamma h(Z)),\\ &\max_{\text{ED-polymatroid } \boldsymbol{h}} \min(h(XYZ), h(X) + \gamma h(Y) + h(Z)),\\ &\max_{\text{ED-polymatroid } \boldsymbol{h}} \min(h(XYZ), \gamma h(X) + h(Y) + h(Z))\Big)\end{aligned} \tag{11}$$

Assume that $\gamma \stackrel{\text{def}}{=} \omega - 2$ is fixed, and consider the first term inside the outermost max above. We can turn this term into a linear program (LP) by introducing a new variable t and replacing the min operator with a max of t subject to some upper bounds on t:

$$\max_{\substack{t\in\mathbb{R}\\ \text{ED-polymatroid } \boldsymbol{h}}} \{t \quad | \quad t \leq h(XYZ), \quad t \leq h(X) + h(Y) + \gamma h(Z)\} \tag{12}$$

Let opt denote the optimal objective value of the above LP. We will show that opt $= \frac{2\omega}{\omega+1}$. Since the other two LPs are similar, it follows that ω-subw$(Q_\triangle) = \frac{2\omega}{\omega+1}$.

First, we show that opt $\geq \frac{2\omega}{\omega+1}$. To that end, consider the following polymatroid (where $h(\emptyset) = 0$):

$$h(X) = h(Y) = h(Z) = \frac{2}{\omega+1}, \qquad h(XY) = h(YZ) = h(XZ) = 1, \qquad h(XYZ) = \frac{2\omega}{\omega+1}.$$

It can be verified that this is a valid polymatroid (for any $\omega \in [2,3]$), it is edge-dominated, and forms a feasible (primal) solution to the LP (alongside $t = \frac{2\omega}{\omega+1}$), thus proving that opt $\geq \frac{2\omega}{\omega+1}$. In the next section, we will show a feasible dual solution that proves opt $\leq \frac{2\omega}{\omega+1}$.

2.5 Computing query answers in ω-subdmodular width time

We now give a simplified overview of our algorithm for computing the answer to a query Q in time $\tilde{O}(N^{\omega\text{-subw}(Q)})$ for any rational ω.[9] We will use the triangle query $Q_\triangle$ as an example. It should be noted however that this simple example is not sufficient to reveal the major technical challenges of designing the general algorithm. Many of these challenges are unique to matrix multiplication, and are not encountered in the original PANDA algorithm [3, 4].

[7]Note that $\mathsf{MM}(X;Y;Z)$ is symmetric.
[8]Note that $\min(a, \max(b,c)) = \max(\min(a,b), \min(a,c))$
[9]If ω is not rational, then any rational upper bound works.

A *Shannon inequality* is an inequality that holds over all polymatroids $\boldsymbol{h}$. The following Shannon inequality corresponds to a feasible dual solution to the LP from Eq. (12):

$$\omega \underbrace{h(XYZ)}_{\substack{\geq \\ t \\ \text{(for-loop cost)}}} + \underbrace{h(X) + h(Y) + \gamma h(Z)}_{\substack{\geq \\ t \\ \text{(one term of MM cost)}}} \leq 2 \underbrace{h(XY)}_{\substack{\leq \\ 1 \\ (R(X,Y))}} + (\omega - 1) \underbrace{h(YZ)}_{\substack{\leq \\ 1 \\ (S(Y,Z))}} + (\omega - 1) \underbrace{h(XZ)}_{\substack{\leq \\ 1 \\ (T(X,Z))}} \tag{13}$$

In particular, the above is a Shannon inequality because it is a sum of the following submodularities:

$$h(XYZ) + h(X) \leq h(XY) + h(XZ)$$
$$h(XYZ) + h(Y) \leq h(XY) + h(YZ)$$
$$\gamma h(XYZ) + \gamma h(Z) \leq \gamma h(XZ) + \gamma h(YZ)$$

Since $\boldsymbol{h}$ is edge-dominated, each term on the RHS of Eq. (13) is upper bounded by 1, hence the RHS is $\leq 2\omega$. On the other hand, by Eq. (12), the LHS is at least $(\omega + 1)t$. Therefore, Inequality (13) implies $t \leq \frac{2\omega}{\omega+1}$, hence opt $\leq \frac{2\omega}{\omega+1}$. Each term on the RHS of Eq. (13) corresponds to one input relation, whereas each *group of terms* on the LHS corresponds to the cost of solving a subquery in the plan. In particular, the group $h(XYZ)$ corresponds to the cost of solving the query using for-loops, whereas the group $h(X) + h(Y) + \gamma h(Z)$ corresponds to one of three terms that capture the cost of solving the query using MM.

First, the algorithm constructs a *proof sequence* of the Shannon inequality (13), which is a step-by-step proof of the inequality that transforms the RHS into the LHS. Figure 1 (left) shows the proof sequence for Eq. (13). Then, the algorithm translates each proof step into a corresponding database operation. In particular, initially each term on the RHS of Eq. (13) corresponds to an input relation. Each time we apply a proof step replacing some terms on the RHS with some other terms, we simultaneously apply a database operation replacing the corresponding relations with some new relations. Figure 1 (right) shows the corresponding database operations, which together make the algorithm for answering $Q_\triangle$. In this example, there are only two types of proof steps:

- *Decomposition Step* of the form $h(XY) \to h(X) + h(Y|X)$. Let $R(X, Y)$ be the relation corresponding to $h(XY)$. The corresponding database operation is to *partition* $R(X, Y)$ into two parts based on the *degree* of X, i.e. the number of matching Y-values for a given X. In particular, X-values with degree $> \Delta \stackrel{\text{def}}{=} N^{\frac{\omega-1}{\omega+1}}$ go into in the "heavy" part $R_h(X)$, whereas the remaining X-values (along with their matching Y-values) go into the "light" part $R_\ell(X, Y)$. Note that $|R_h|$ cannot exceed $N/\Delta = N^{\frac{2}{\omega+1}}$.
- *Submodularity Step*[10] of the form $h(XZ) + h(Y|X) \to h(XYZ)$. The corresponding database operation is to *join* the two corresponding relations, $T(X, Z) \bowtie R_\ell(X, Y)$. Since R_ℓ is the light part, this join takes time $N \cdot N^{\frac{\omega-1}{\omega+1}} = N^{\frac{2\omega}{\omega+1}}$, as desired. The same goes for the other submodularity steps.

The three submodularity steps compute three relations $Q_{\ell,1}, Q_{\ell,2}, Q_{\ell,3}$ covering triangles (X, Y, Z) where either X, Y, or Z is light. We still need to account for triangles where all three are heavy. For those, we use $R_h(X)$, $S_h(Y)$, and $T_h(Z)$ to from two matrices $M_1(X, Y)$ and $M_2(Y, Z)$, and then multiply them to get $M(X, Z)$. Since $|R_h|, |S_h|, |T_h| \leq N^{\frac{2}{\omega+1}}$, this multiplication takes time $N^{\frac{2\omega}{\omega+1}}$, as desired. Finally, we join $M(X, Z)$ with $T(X, Z)$ to get $Q_h(X, Z)$. There exists a triangle if and only if either one of $Q_{\ell,1}, Q_{\ell,2}, Q_{\ell,3}$ or Q_h is non-empty.[11]

[10] Note that $h(XZ) + h(Y|X) \geq h(XYZ)$ is just another form of the submodularity $h(XZ) + h(XY) \geq h(X) + h(XYZ)$, hence the name.

[11] Recall that inequality (13) comes from the optimal dual solution of the LP in Eq. (12), which comes from the first term (out of three) inside the outer max in Eq. (11). There are two other Shannon inequalities that come from the other two terms. In

Proof Sequence			Algorithm		
$h(XY)$	$\to$	$h(X) + h(Y\mid X)$	$R(X,Y)$	$\xrightarrow{\text{partition}}$	$R_h(X), R_\ell(X,Y)$
$h(XZ) + h(Y\mid X)$	$\to$	$h(XYZ)$	$T(X,Z) \bowtie R_\ell(X,Y)$	$\to$	$Q_{\ell,1}(X,Y,Z)$
$h(YZ)$	$\to$	$h(Y) + h(Z\mid Y)$	$S(Y,Z)$	$\xrightarrow{\text{partition}}$	$S_h(Y), S_\ell(Y,Z)$
$h(XY) + h(Z\mid Y)$	$\to$	$h(XYZ)$	$R(X,Y) \bowtie S_\ell(Y,Z)$	$\to$	$Q_{\ell,2}(X,Y,Z)$
$\gamma h(XZ)$	$\to$	$\gamma h(Z) + \gamma h(X\mid Z)$	$T(X,Z)$	$\xrightarrow{\text{partition}}$	$T_h(Z), T_\ell(Z,X)$
$\gamma h(YZ) + \gamma h(X\mid Z)$	$\to$	$\gamma h(XYZ)$	$S(Y,Z) \bowtie T_\ell(Z,X)$	$\to$	$Q_{\ell,3}(X,Y,Z)$
			$R_h(X) \bowtie S_h(Y) \bowtie R(X,Y) \to M_1(X,Y)$		
			$S_h(Y) \bowtie T_h(Z) \bowtie S(Y,Z) \to M_2(Y,Z)$		
			$M_1(X,Y) \times M_2(Y,Z) \xrightarrow{\text{MM}} M(X,Z)$		
			$M(X,Z) \bowtie T(X,Z) \to Q_h(X,Z)$		

Fig. 1. The proof sequence for the Shannon inequality (13) along with the corresponding algorithm for $Q_\triangle$.

3 PRELIMINARIES

In this section, we present the formal definitions and notations for various background concepts that were introduced informally in the introduction.

Hypergraphs. A hypergraph $\mathcal{H}$ is a pair $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of vertices, and $\mathcal{E} \subseteq 2^{\mathcal{V}}$ is a set of hyperedges. Each hyperedge $Z \in \mathcal{E}$ is a subset of $\mathcal{V}$. We typically use k to denote the number of vertices in $\mathcal{V}$. Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ and a vertex $X \in \mathcal{V}$, we define:

- $\partial_{\mathcal{H}}(X)$ is the set of hyperedges that contain X, i.e. $\partial_{\mathcal{H}}(X) \stackrel{\text{def}}{=} \{Z \in \mathcal{E} \mid X \in Z\}$.
- $U_{\mathcal{H}}(X)$ is the union of all hyperedges that contain X, i.e. $U_{\mathcal{H}}(X) \stackrel{\text{def}}{=} \bigcup_{Z \in \partial_{\mathcal{H}}(X)} Z$.
- $N_{\mathcal{H}}(X)$ is the set of neighbors of X (excluding X), i.e. $N_{\mathcal{H}}(X) \stackrel{\text{def}}{=} U_{\mathcal{H}}(X) \setminus \{X\}$.

When $\mathcal{H}$ is clear from the context, we drop the subscript and simply write $\partial(X)$, $U(X)$, and $N(X)$. Given a query Q of the form (1), the *hypergraph of* Q is a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} \stackrel{\text{def}}{=} \text{vars}(Q)$ and $\mathcal{E} \stackrel{\text{def}}{=} \{Z \mid R(Z) \in \text{atoms}(Q)\}$. We often use a query Q and its hypergraph $\mathcal{H}$ interchangeably, e.g. in the contexts of tree decompositions, submodular width, etc.

Tree Decompositions. Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ (or a query Q whose hypergraph is $\mathcal{H}$), a *tree decomposition*, or *TD* for short, is a pair (T, χ), where T is a tree, and $\chi : \text{nodes}(T) \to 2^{\mathcal{V}}$ is a map from the nodes of T to subsets of $\mathcal{V}$, that satisfies the following properties:

- For every hyperedge $Z \in \mathcal{E}$, there is a node $t \in \text{nodes}(T)$ such that $Z \subseteq \chi(t)$.
- For every vertex $X \in \mathcal{V}$, the set $\{t \in \text{nodes}(T) \mid X \in \chi(t)\}$ forms a connected sub-tree of T.

Each set $\chi(t)$ is called a *bag* of the tree decomposition. We use $\mathcal{T}(\mathcal{H})$ to denote the set of all tree decompositions of $\mathcal{H}$. A tree decomposition is called *trivial* if it consists of a single bag containing all vertices. A tree decomposition (T, χ) is called *redundant* if it contains two different bags $t_1 \neq t_2 \in \text{nodes}(T)$ where $\chi(t_1) \subseteq \chi(t_2)$. It is well-known that every redundant tree decomposition can be converted into a non-redundant one by removing bags that are contained in other bags.

Variable Elimination. Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be a hypergraph, $k \stackrel{\text{def}}{=} |\mathcal{V}|$, and $\pi(\mathcal{V})$ denote the set of all permutations of $\mathcal{V}$. Given a fixed permutation $\boldsymbol{\sigma} = (X_1, \ldots, X_k) \in \pi(\mathcal{V})$, we define a sequence of

this simple example, we are lucky enough since they lead to the same algorithm described above. In general, there is a fairly complicated combinatorial argument to reason about many Shannon inequalities at once. See Sections E.5 and E.6.

hypergraphs $\mathcal{H}_1^\sigma, \ldots, \mathcal{H}_k^\sigma$, called an *elimination hypergraph sequence*, as follows: $\mathcal{H}_1^\sigma \stackrel{\text{def}}{=} \mathcal{H}$, and for $i = 1, ..., k-1$, the hypergraph $\mathcal{H}_{i+1}^\sigma = (\mathcal{V}_{i+1}^\sigma, \mathcal{E}_{i+1}^\sigma)$ is defined recursively in terms of the previous hypergraph $\mathcal{H}_i^\sigma = (\mathcal{V}_i^\sigma, \mathcal{E}_i^\sigma)$ using:

$$\mathcal{V}_{i+1}^\sigma \stackrel{\text{def}}{=} \mathcal{V}_i^\sigma \setminus \{X_i\}, \qquad \mathcal{E}_{i+1}^\sigma \stackrel{\text{def}}{=} \mathcal{E}_i^\sigma \setminus \partial_{\mathcal{H}_i^\sigma}(X_i) \cup \{N_{\mathcal{H}_i^\sigma}(X_i)\}.$$

In words, $\mathcal{H}_{i+1}^\sigma$ results from $\mathcal{H}_i^\sigma$ by removing the vertex X_i and replacing all hyperedges that contain X_i with a single hyperedge which is their union *minus* X_i. We refer to the permutation $\boldsymbol{\sigma}$ as a *variable elimination order*, or *VEO* for short. For convenience, we define $\partial_i^\sigma \stackrel{\text{def}}{=} \partial_{\mathcal{H}_i^\sigma}(X_i)$ and also define U_i^σ, and N_i^σ analogously. Given a number k, we use $[k]$ to denote the set $\{1, \ldots, k\}$.

Proposition 3.1 (Equivalence of TDs and VEOs [2]). *Given $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ where $k \stackrel{\text{def}}{=} |\mathcal{V}|$:*

(1) For every TD $(T, \chi) \in \mathcal{T}(\mathcal{H})$, there exists a VEO $\boldsymbol{\sigma} \in \pi(\mathcal{V})$ that satisfies: for every $i \in [k]$, there exists a node $t \in \text{nodes}(T)$ such that $U_i^\sigma \subseteq \chi(t)$.

(2) For every VEO $\boldsymbol{\sigma} \in \pi(\mathcal{V})$, there exists a TD $(T, \chi) \in \mathcal{T}(\mathcal{H})$ that satisfies: For every node $t \in \text{nodes}(T)$, there exists $i \in [k]$ such that $\chi(t) \subseteq U_i^\sigma$.

The Submodular Width. Given a set $\mathcal{V}$, a function $\boldsymbol{h} : 2^{\mathcal{V}} \to \mathbb{R}_+$ is called a *polymatroid* if it satisfies the following properties:

$$h(\boldsymbol{X}) + h(\boldsymbol{Y}) \geq h(\boldsymbol{X} \cup \boldsymbol{Y}) + h(\boldsymbol{X} \cap \boldsymbol{Y}) \qquad \forall \boldsymbol{X}, \boldsymbol{Y} \subseteq \mathcal{V} \quad \text{(submodularity)} \tag{14}$$

$$h(\boldsymbol{X}) \leq h(\boldsymbol{Y}) \qquad \forall \boldsymbol{X} \subset \boldsymbol{Y} \subseteq \mathcal{V} \quad \text{(monotonicity)} \tag{15}$$

$$h(\emptyset) = 0 \qquad \text{(strictness)} \tag{16}$$

The above properties are also known as *Shannon inequalities*. We use $\Gamma_{\mathcal{V}}$ to denote the set of all polymatroids over $\mathcal{V}$. When $\mathcal{V}$ is clear from the context, we drop $\mathcal{V}$ and simply write Γ. Given a polymatroid $\boldsymbol{h}$ and sets $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \subseteq \mathcal{V}$, we use $\boldsymbol{XY}$ as a shorthand for $\boldsymbol{X} \cup \boldsymbol{Y}$, and we define:

$$h(\boldsymbol{Y}|\boldsymbol{X}) \stackrel{\text{def}}{=} h(\boldsymbol{XY}) - h(\boldsymbol{X}) \tag{17}$$

$$h(\boldsymbol{Y}; \boldsymbol{Z}|\boldsymbol{X}) \stackrel{\text{def}}{=} h(\boldsymbol{XY}) + h(\boldsymbol{XZ}) - h(\boldsymbol{X}) - h(\boldsymbol{XYZ}) \tag{18}$$

Using the above notation, we can rewrite Eq. (15) as $h(\boldsymbol{Y}|\boldsymbol{X}) \geq 0$, and Eq. (14) as $h(\boldsymbol{X}; \boldsymbol{Y}|\boldsymbol{X} \cap \boldsymbol{Y}) \geq 0$.

Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, a function $\boldsymbol{h} : 2^{\mathcal{V}} \to \mathbb{R}_+$ is called *edge-dominated* if it satisfies $h(\boldsymbol{X}) \leq 1$ for all $\boldsymbol{X} \in \mathcal{E}$. We use $\text{ED}_{\mathcal{H}}$ to denote the set of all edge-dominated functions over $\mathcal{H}$. When $\mathcal{H}$ is clear from the context, we drop $\mathcal{H}$ and simply write ED.

Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, the *submodular width* [17] of $\mathcal{H}$ is defined as follows:

$$\text{subw}(\mathcal{H}) \stackrel{\text{def}}{=} \max_{\boldsymbol{h} \in \Gamma \cap \text{ED}} \min_{(T,\chi) \in \mathcal{T}(\mathcal{H})} \max_{t \in \text{nodes}(T)} h(\chi(t)) \tag{19}$$

Based on Proposition 3.1 along with the fact that a polymatroid $\boldsymbol{h} : 2^{\mathcal{V}} \to \mathbb{R}_+$ is monotone (Eq. (15)), we can equivalently define the submodular width using VEOs as follows:

$$\text{subw}(\mathcal{H}) \stackrel{\text{def}}{=} \max_{\boldsymbol{h} \in \Gamma \cap \text{ED}} \min_{\boldsymbol{\sigma} \in \pi(\mathcal{V})} \max_{i \in [k]} h(U_i^\sigma) \tag{20}$$

Fast Matrix Multiplication. We show here how to multiply two rectangular matrices A and B of dimensions $n^a \times n^b$ and $n^b \times n^c$ in the runtime $n^{\omega^\square(a,b,c)}$, as defined by Eq. (6). Let $d = \min(a, b, c)$. We partition A into $\frac{n^a}{n^d} \times \frac{n^b}{n^d}$ blocks and B into $\frac{n^b}{n^d} \times \frac{n^c}{n^d}$ blocks. Therefore, we have to perform $n^{a+b+c-3d}$ block multiplications, each of which takes time $O(n^{d \cdot \omega})$. The overall complexity, on $(\log n)$-scale, is $a + b + c - (3 - \omega) \min(a, b, c)$, which gives Eq. (6). When $\omega = 2$, this algorithm for rectangular matrix multiplication is already optimal because the complexity from Eq. (6) becomes linear in the sizes of the input and output matrices. However, when $\omega > 2$, there could be faster algorithms that are not based on square matrix multiplication; see e.g. [14]. We define $\overline{\omega}(a, b, c)$ as the smallest exponent for multiplying two rectangular matrices of sizes $n^a \times n^b$ and $n^b \times n^c$ within $O(n^{\overline{\omega}(a,b,c)})$ time. Based on the above discussion, we have $\overline{\omega}(a, b, c) \leq \omega^{\square}(a, b, c)$ and this becomes an equality when $\omega = 2$ or when $a = b = c$.

4 THE ω-SUBMODULAR WIDTH: FORMAL DEFINITION

While Eq. (8) gives a high-level sketch of our definition of the ω-submodular width, we aim here to provide the full formal definition. To that end, we start with some auxiliary definitions.

4.1 Generalizing variable elimination orders

In the traditional submodular width (Eq. (9)), it was sufficient to consider variable elimination orders that eliminate only one variable at a time. However once we allow using MM to eliminate variables, there could be situations where eliminating multiple variables at once using a single MM might be cheaper than eliminating the same set of variables one at a time using a sequence of MMs. Motivated by this observation, our first task is to generalize the notion of a VEO and an elimination hypergraph sequence to allow eliminating multiple variables at once.

First, we lift the definitions of $\partial_{\mathcal{H}}(X)$, $U_{\mathcal{H}}(X)$, and $N_{\mathcal{H}}(X)$ from a single variable X to a set of variables $\boldsymbol{X}$. Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ and a non-empty set of variables $\boldsymbol{X} \subseteq \mathcal{V}$, we define $\partial_{\mathcal{H}}(\boldsymbol{X})$ as the set of hyperedges in $\mathcal{H}$ that overlap with $\boldsymbol{X}$:

$$\partial_{\mathcal{H}}(\boldsymbol{X}) \stackrel{\text{def}}{=} \{\boldsymbol{Z} \in \mathcal{E} \mid \boldsymbol{X} \cap \boldsymbol{Z} \neq \emptyset\}, \qquad U_{\mathcal{H}}(\boldsymbol{X}) \stackrel{\text{def}}{=} \bigcup_{\boldsymbol{Z} \in \partial_{\mathcal{H}}(\boldsymbol{X})} \boldsymbol{Z}, \qquad N_{\mathcal{H}}(\boldsymbol{X}) \stackrel{\text{def}}{=} U_{\mathcal{H}}(\boldsymbol{X}) \setminus \boldsymbol{X}.$$

Definition 4.1 (Generalized Variable Elimination Order (GVEO)). Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, let $\overline{\pi}(\mathcal{V})$ denote the set of all *ordered partitions* of $\mathcal{V}$. Namely, each element $\overline{\boldsymbol{\sigma}} \in \overline{\pi}(\mathcal{V})$ is a tuple $(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{|\overline{\boldsymbol{\sigma}}|})$ of non-empty and pairwise disjoint sets $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{|\overline{\boldsymbol{\sigma}}|}$ whose union is $\mathcal{V}$. We refer to each $\overline{\boldsymbol{\sigma}} \in \overline{\pi}(\mathcal{V})$ as a *generalized variable elimination order*, or *GVEO* for short. Given $\overline{\boldsymbol{\sigma}} \in \overline{\pi}(\mathcal{V})$, we define a *generalized elimination hypergraph sequence* $\mathcal{H}_1^{\overline{\boldsymbol{\sigma}}}, \ldots, \mathcal{H}_{|\overline{\boldsymbol{\sigma}}|}^{\overline{\boldsymbol{\sigma}}}$ as follows: $\mathcal{H}_1^{\overline{\boldsymbol{\sigma}}} \stackrel{\text{def}}{=} \mathcal{H}$, and for $i = 1, \ldots, |\overline{\boldsymbol{\sigma}}| - 1$:

$$\mathcal{V}_{i+1}^{\overline{\boldsymbol{\sigma}}} \stackrel{\text{def}}{=} \mathcal{V}_i^{\overline{\boldsymbol{\sigma}}} \setminus \boldsymbol{X}_i, \qquad \mathcal{E}_{i+1}^{\overline{\boldsymbol{\sigma}}} \stackrel{\text{def}}{=} \mathcal{E}_i^{\overline{\boldsymbol{\sigma}}} \setminus \partial_{\mathcal{H}_i^{\overline{\boldsymbol{\sigma}}}}(\boldsymbol{X}_i) \cup \{N_{\mathcal{H}_i^{\overline{\boldsymbol{\sigma}}}}(\boldsymbol{X}_i)\}.$$

We define $\partial_i^{\overline{\boldsymbol{\sigma}}} \stackrel{\text{def}}{=} \partial_{\mathcal{H}_i^{\overline{\boldsymbol{\sigma}}}}(\boldsymbol{X}_i)$, just like before, and the same goes for $U_i^{\overline{\boldsymbol{\sigma}}}$ and $N_i^{\overline{\boldsymbol{\sigma}}}$.

4.2 Expressing MM runtime using polymatroids

We now give the formal definition of the MM expression that generalizes the special case $\mathsf{MM}(X; Y; Z)$ that was given earlier in Eq. (6). Recall that ω is a constant in the range $[2, 3]$ and $\gamma \stackrel{\text{def}}{=} \omega - 2$.

Definition 4.2 (Matrix multiplication expression, MM). Let $\boldsymbol{h} : 2^{\mathcal{V}} \to \mathbb{R}_+$ be a polymatroid. Given four pairwise disjoint subsets, $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{G} \subseteq \mathcal{V}$, we define the *matrix multiplication expression* $\mathsf{MM}(\boldsymbol{X}; \boldsymbol{Y}; \boldsymbol{Z}|\boldsymbol{G})$ as follows:

$$\begin{aligned}\mathsf{MM}(\boldsymbol{X}; \boldsymbol{Y}; \boldsymbol{Z}|\boldsymbol{G}) \stackrel{\text{def}}{=} \max\big(&h(\boldsymbol{X}|\boldsymbol{G}) + h(\boldsymbol{Y}|\boldsymbol{G}) + \gamma h(\boldsymbol{Z}|\boldsymbol{G}) + h(\boldsymbol{G}),\\ &h(\boldsymbol{X}|\boldsymbol{G}) + \gamma h(\boldsymbol{Y}|\boldsymbol{G}) + h(\boldsymbol{Z}|\boldsymbol{G}) + h(\boldsymbol{G}),\\ &\gamma h(\boldsymbol{X}|\boldsymbol{G}) + h(\boldsymbol{Y}|\boldsymbol{G}) + h(\boldsymbol{Z}|\boldsymbol{G}) + h(\boldsymbol{G})\big)\end{aligned} \tag{21}$$

When $\boldsymbol{G}$ is empty, we write $\mathsf{MM}(\boldsymbol{X}; \boldsymbol{Y}; \boldsymbol{Z})$ as a shorthand for $\mathsf{MM}(\boldsymbol{X}; \boldsymbol{Y}; \boldsymbol{Z}|\emptyset)$.

The following proposition intuitively says that the MM runtime is at least linear in the sizes of the two input matrices and the output matrix.

Proposition 4.3. $\max(h(\boldsymbol{XYG}), h(\boldsymbol{YZG}), h(\boldsymbol{XZG})) \quad \leq \quad \mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G})$

Proof. $h(\boldsymbol{XYG}) \leq h(\boldsymbol{X}|\boldsymbol{G}) + h(\boldsymbol{Y}|\boldsymbol{G}) + h(\boldsymbol{G}) \leq h(\boldsymbol{X}|\boldsymbol{G}) + h(\boldsymbol{Y}|\boldsymbol{G}) + \gamma h(\boldsymbol{Z}|\boldsymbol{G}) + h(\boldsymbol{G})$. □

Proposition 4.4. *If $\omega = 3$, then $h(\boldsymbol{XYZG}) \leq \mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G})$.*

Proof. When $\omega = 3$, $h(\boldsymbol{XYZG}) \leq h(\boldsymbol{X}|\boldsymbol{G}) + h(\boldsymbol{Y}|\boldsymbol{G}) + h(\boldsymbol{Z}|\boldsymbol{G}) + h(\boldsymbol{G}) = \mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G})$. □

4.3 Variable elimination via matrix multiplication

Here we present the formal definition of the quantity EMM introduced earlier in Section 2.2, which captures the cost of eliminating a variable (or set of variables) using matrix multiplication. Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ and a set of vertices $\boldsymbol{X} \subseteq \mathcal{V}$, we take the neighboring hyperedges $\partial_{\mathcal{H}}(\boldsymbol{X})$ of $\boldsymbol{X}$ and assign them to two (potentially overlapping) sets of hyperedges $\mathcal{A}$ and $\mathcal{B}$, each of which will form a matrix in a matrix multiplication expression. Finally, we take the minimum expression over all such assignments.

Definition 4.5 (Variable elimination expression via matrix multiplication, EMM*).* Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be a hypergraph and $\boldsymbol{h} : 2^{\mathcal{V}} \to \mathbb{R}_+$ be a polymatroid. Given a non-empty set of vertices $\boldsymbol{X} \in \mathcal{V}$, we define the *variable elimination expression via matrix multiplication*, denoted by $\mathsf{EMM}_{\mathcal{H}}(\boldsymbol{X})$, as:

$$\begin{aligned}\mathsf{EMM}_{\mathcal{H}}(\boldsymbol{X}) \stackrel{\text{def}}{=} \min\Big\{ &\mathsf{MM}\big((\boldsymbol{A} \setminus \boldsymbol{B}) \setminus \boldsymbol{G} \quad ; \quad (\boldsymbol{B} \setminus \boldsymbol{A}) \setminus \boldsymbol{G} \quad ; \quad \boldsymbol{X} \quad | \quad \boldsymbol{G}\big) \quad | \\ &\exists \mathcal{A}, \mathcal{B} \subseteq \partial_{\mathcal{H}}(\boldsymbol{X}), \quad \mathcal{A} \cup \mathcal{B} = \partial_{\mathcal{H}}(\boldsymbol{X}), \quad \boldsymbol{A} = \cup \mathcal{A}, \quad \boldsymbol{B} = \cup \mathcal{B}, \\ &\boldsymbol{X} \subseteq \boldsymbol{A} \cap \boldsymbol{B}, \quad (\boldsymbol{A} \cap \boldsymbol{B}) \setminus \boldsymbol{X} \quad \subseteq \quad \boldsymbol{G} \quad \subseteq \quad (\boldsymbol{A} \cup \boldsymbol{B}) \setminus \boldsymbol{X} \Big\}\end{aligned} \tag{22}$$

We can simplify the above expression further by excluding trivial combinations where either one of the two sets $(\boldsymbol{A} \setminus \boldsymbol{B}) \setminus \boldsymbol{G}$, or $(\boldsymbol{B} \setminus \boldsymbol{A}) \setminus \boldsymbol{G}$ is empty.

Example 4.6. Consider the following hypergraph representing a 4-clique:

$$\mathcal{H} = (\{X, Y, Z, W\}, \quad \{\{X, Y\}, \{X, Z\}, \{X, W\}, \{Y, Z\}, \{Y, W\}, \{Z, W\}\}) \tag{23}$$

Here we have $\partial_{\mathcal{H}}(X) = \{\{X, Y\}, \{X, Z\}, \{X, W\}\}$, and there are 6 different and non-trivial ways to assign them to $\mathcal{A}$ and $\mathcal{B}$, resulting in 6 different MM expressions:

$$\begin{aligned}\mathsf{EMM}_{\mathcal{H}}(X) = \min\big(&\mathsf{MM}(YZ; W; X), \quad \mathsf{MM}(YW; Z; X), \quad \mathsf{MM}(ZW; Y; X), \\ &\mathsf{MM}(Y; Z; X|W), \quad \mathsf{MM}(Y; W; X|Z), \quad \mathsf{MM}(Z; W; X|Y)\big)\end{aligned} \tag{24}$$

The first three expressions above correspond to cases where $\mathcal{A}$ and $\mathcal{B}$ are disjoint, while the last three correspond to cases where $\mathcal{A}$ and $\mathcal{B}$ overlap. For example, the fourth expression $\mathsf{MM}(Y; Z; X|W)$ results from taking $\mathcal{A} = \{\{X, Y\}, \{X, W\}\}$, $\mathcal{B} = \{\{X, Z\}, \{X, W\}\}$, and $\boldsymbol{G} = \{W\}$.

Let $\overline{\boldsymbol{\sigma}} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{|\overline{\sigma}|})$ be a generalized variable elimination order and $\mathcal{H}_1^{\overline{\sigma}}, \ldots, \mathcal{H}_{|\overline{\sigma}|}^{\overline{\sigma}}$ be the corresponding generalized hypergraph sequence (Definition 4.1). We use $\mathsf{EMM}_i^{\overline{\sigma}}$ to denote $\mathsf{EMM}_{\mathcal{H}_i^{\overline{\sigma}}}(\boldsymbol{X}_i)$.

4.4 Putting pieces together

We now have all the components in place to formally define the ω-submodular width.

Definition 4.7 (ω-submodular width). Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, the *$\omega$-submodular width* of $\mathcal{H}$, denoted by ω-subw$(\mathcal{H})$, is defined as follows:

$$\omega\text{-subw}(\mathcal{H}) \quad \stackrel{\text{def}}{=} \quad \max_{\boldsymbol{h}\in\Gamma\cap\mathsf{ED}} \quad \min_{\overline{\sigma}\in\overline{\pi}(\mathcal{V})} \quad \max_{i\in[|\overline{\sigma}|]} \quad \min(h(U_i^{\overline{\sigma}}), \mathsf{EMM}_i^{\overline{\sigma}}) \tag{25}$$

To compare the ω-submodular width with the traditional submodular width, we include below a definition of the submodular width that is equivalent to Eq. (20), except that it uses GVEOs:

Proposition 4.8. *The following is an equivalent definition of the submodular width of $\mathcal{H}$:*

$$\mathit{subw}(\mathcal{H}) \quad \stackrel{\text{def}}{=} \quad \max_{\boldsymbol{h}\in\Gamma\cap\mathsf{ED}} \quad \min_{\overline{\sigma}\in\overline{\pi}(\mathcal{V})} \quad \max_{i\in[|\overline{\sigma}|]} \quad h(U_i^{\overline{\sigma}}) \tag{26}$$

Proposition 4.9. *For any hypergraph $\mathcal{H}$, ω-subw$(\mathcal{H}) \leq$ subw$(\mathcal{H})$.*

Proposition 4.10. *If $\omega = 3$, then for any hypergraph $\mathcal{H}$, ω-subw$(\mathcal{H}) =$ subw$(\mathcal{H})$.*

Proposition 4.9 holds by comparing Eq. (25) to (26). Proposition 4.10 holds because Proposition 4.4 implies that when $\omega = 3$, $h(U_i^{\overline{\sigma}}) \leq \mathsf{EMM}_i^{\overline{\sigma}}$.

For the purpose of computing the ω-submodular width, we propose an equivalent form of Eq. (25) that is easier to compute.

Proposition 4.11. *The ω-submodular width of a hypergraph can be equivalently defined as follows:*

$$\omega\text{-}\mathit{subw}(\mathcal{H}) \quad \stackrel{\text{def}}{=} \quad \max_{\boldsymbol{h}\in\Gamma\cap\mathsf{ED}} \quad \min_{\overline{\sigma}\in\overline{\pi}(\mathcal{V})} \quad \max_{\substack{i\in[|\overline{\sigma}|] \\ \forall j<i, U_i^{\overline{\sigma}} \not\subseteq U_j^{\overline{\sigma}}}} \quad \min(h(U_i^{\overline{\sigma}}), \mathsf{EMM}_i^{\overline{\sigma}}) \tag{27}$$

The only difference between Eq. (27) and Eq. (25) is that $\max_{i\in[|\overline{\sigma}|]}$ is more restricted in Eq. (27). In particular, we only need to consider i where $U_i^{\overline{\sigma}}$ is not contained in any previous $U_j^{\overline{\sigma}}$. For example, given the hypergraph of the triangle query $Q_\triangle$ from Eq. (2), Eq. (27) allows us to only consider the first variable elimination step, thus simplifying the ω-submodular width to become Eq. (10).

Example 4.12. Let's take the 4-clique hypergraph from Eq. (23). By Proposition 4.11, for any $\overline{\sigma} \in \overline{\pi}(\mathcal{V})$, we only need to consider $U_1^{\overline{\sigma}}$. Hence, the ω-submodular width becomes:

$$\begin{aligned}\omega\text{-subw}(\mathcal{H}) = \max_{\boldsymbol{h}\in\Gamma\cap\mathsf{ED}} \min\big(&h(XYZW), \quad \mathsf{MM}(XY;Z;W), \quad \mathsf{MM}(XZ;Y;W), \\ &\mathsf{MM}(XW;Y;Z), \quad \mathsf{MM}(YZ;X;W), \quad \mathsf{MM}(YW;X;Z), \\ &\mathsf{MM}(ZW;X;Y), \quad \mathsf{MM}(Y;Z;W|X), \quad \mathsf{MM}(X;Z;W|Y), \\ &\mathsf{MM}(X;Y;W|Z), \quad \mathsf{MM}(X;Y;Z|W)\big)\end{aligned} \tag{28}$$

5 THE ω-SUBMODULAR WIDTH FOR EXAMPLE QUERIES.

We show in Table 2 the ω-submodular width for several classes of queries and compare it with the submodular width for each of them. Proofs can be found in Appendix C. They use a variety of techniques to obtain both upper and lower bounds on the ω-submodular width. Below are the hypergraphs for these query classes:

$$k\text{-Clique:} \quad \mathcal{H} = \big(\{X_1, X_2, \cdots, X_k\},\ \{\{X_i, X_j\} : i, j \in [k], i \neq j\}\big) \tag{29}$$

$$k\text{-Cycle:} \quad \mathcal{H} = (\{X_1, X_2, \cdots, X_k\},\ \{\{X_i, X_{i+1}\} : i \in [k-1]\} \cup \{\{X_k, X_1\}\}) \tag{30}$$

$$k\text{-Pyramid:} \quad \mathcal{H} = (\{Y, X_1, X_2, \cdots, X_k\},\ \{\{Y, X_1, \}, \{Y, X_2\}, \cdots, \{Y, X_k\}, \{X_1, X_2, \cdots, X_k\}\}) \tag{31}$$

Queries	Submodular Width	ω-Submodular Width
Triangle $Q_\triangle$ (Eq.(2))	1.5	$\frac{2\omega}{\omega+1}$ (Lemma C.5)
4-Clique	2	$\frac{\omega+1}{2}$ (Lemma C.6)
5-Clique	2.5	$\frac{\omega}{2}+1$ (Lemma C.7)
k-Clique (Eq. (29))	$\frac{k}{2}$	$\frac{1}{2}\cdot\lceil\frac{k}{3}\rceil+\frac{1}{2}\cdot\lceil\frac{k-1}{3}\rceil+\frac{1}{2}\cdot\lfloor\frac{k}{3}\rfloor\cdot(\omega-2)$ (Lemma C.8)
4-Cycle $Q_\square$ (Eq.(4))	1.5	$2-\frac{3}{2\cdot\min\{\omega,\frac{5}{2}\}+1}$ (Lemma C.9)
k-Cycle (Eq. (30))	$2-\frac{1}{\lceil k/2\rceil}$	$\leq c_k^\square$ (Lemma C.10)
3-Pyramid	$\frac{5}{3}$	$2-\frac{1}{\omega}$ (Lemma C.13)
k-Pyramid (Eq. (31))	$2-\frac{1}{k}$	$\leq 2-\frac{2}{\omega\cdot(k-1)-k+3}$ (Lemma C.14)

Table 2. Comparison of the submodular with and ω-submodular width on example queries. Entries marked with "$\leq$" are upper bounds, while the rest are exact values. The symbol $\overline{c}_k$ is the best-known exponent for detecting cycles using rectangular matrix multiplication [10, Theorem 1.3], while $c_k^\square$ is the smallest upper bound on $\overline{c}_k$ that is obtained through *square* matrix multiplication; see Eq. (43) and (45).

6 ALGORITHM FOR COMPUTING THE ω-SUBMODULAR WIDTH

We present in this section a mechanical algorithm for computing the value of the ω-submodular width for any given hypergraph. This algorithm is a crucial component in our algorithm for computing the answer to a query Q in ω-submodular width time, as we will see in Section 7. Using Eq. (27) alone, it is not immediately clear how to compute the ω-submodular width of a hypergraph $\mathcal{H}$ because the outer $\max_{\boldsymbol{h}\in\Gamma\cap\mathsf{ED}}$ ranges over an infinite set. Nevertheless, every other max and min in Eq. (27) ranges over a finite set whose cardinality only depends on $\mathcal{H}$. Note that by Eq. (22), $\mathsf{EMM}_i^{\overline{\sigma}}$ is a minimum of a collection of terms, each of which has the from $\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G})$. Note also that each term $\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G})$ is by itself a maximum of 3 terms, as described by Eq. (21).

In order to compute the ω-submodular width, the first step is to distribute every min over max in Eq. (27), thus pulling all max operators to the top level. To formally describe this distribution, we need some notation. Given an expression e which is either a minimum or a maximum of a collection of terms, we use $\text{args}(e)$ to denote the collection of terms over which the minimum or maximum is taken. Let k be the number of vertices in $\mathcal{H}$. Let $f:\overline{\pi}(\mathcal{V})\to[k]$ be a function that maps every generalized variable elimination order $\overline{\boldsymbol{\sigma}}\in\overline{\pi}(\mathcal{V})$ to a number i between 1 and $|\overline{\boldsymbol{\sigma}}|$ that satisfies $U_i^{\overline{\sigma}}\not\subseteq U_j^{\overline{\sigma}}$ for all $j\in[i-1]$. (Note that $|\overline{\boldsymbol{\sigma}}|\leq k$ for every $\overline{\boldsymbol{\sigma}}$.) Let $\mathcal{F}$ be the set of all such functions f. For a fixed function $f\in\mathcal{F}$, let g_f be a function that maps every pair $(\overline{\boldsymbol{\sigma}},\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G}))$ where $\overline{\boldsymbol{\sigma}}\in\overline{\pi}(\mathcal{V})$ and $\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G})$ is a term in $\text{args}(\mathsf{EMM}_{f(\overline{\sigma})}^{\overline{\sigma}})$ to one of the three terms in $\text{args}(\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G}))$ from Eq. (21). Let $\mathcal{G}_f$ be the set of all such functions g_f for a fixed f. Then, by distributing the min over the max in Eq. (27), we get:

ω-subw$(\mathcal{H})$

$$\begin{aligned}
&\stackrel{\text{def}}{=} \max_{\boldsymbol{h}\in\Gamma\cap\mathsf{ED}} \quad \min_{\overline{\boldsymbol{\sigma}}\in\overline{\pi}(\mathcal{V})} \quad \max_{\substack{i\in[|\overline{\boldsymbol{\sigma}}|] \\ \forall j<i, U_i^{\overline{\boldsymbol{\sigma}}}\not\subseteq U_j^{\overline{\boldsymbol{\sigma}}}}} \quad \min(h(U_i^{\overline{\boldsymbol{\sigma}}}), \mathsf{EMM}_i^{\overline{\boldsymbol{\sigma}}}) \\
&= \max_{\boldsymbol{h}\in\Gamma\cap\mathsf{ED}} \quad \max_{f\in\mathcal{F}} \quad \min_{\overline{\boldsymbol{\sigma}}\in\overline{\pi}(\mathcal{V})} \quad \min\left(h(U_{f(\overline{\boldsymbol{\sigma}})}^{\overline{\boldsymbol{\sigma}}}), \mathsf{EMM}_{f(\overline{\boldsymbol{\sigma}})}^{\overline{\boldsymbol{\sigma}}}\right) \\
&= \max_{\boldsymbol{h}\in\Gamma\cap\mathsf{ED}} \quad \max_{f\in\mathcal{F}} \quad \min_{\overline{\boldsymbol{\sigma}}\in\overline{\pi}(\mathcal{V})} \quad \min\left(h(U_{f(\overline{\boldsymbol{\sigma}})}^{\overline{\boldsymbol{\sigma}}}), \min_{\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G})\in\mathsf{args}\left(\mathsf{EMM}_{f(\overline{\boldsymbol{\sigma}})}^{\overline{\boldsymbol{\sigma}}}\right)} \mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G})\right) \\
&= \max_{\boldsymbol{h}\in\Gamma\cap\mathsf{ED}} \quad \max_{f\in\mathcal{F}} \quad \min_{\overline{\boldsymbol{\sigma}}\in\overline{\pi}(\mathcal{V})} \quad \min\left(h(U_{f(\overline{\boldsymbol{\sigma}})}^{\overline{\boldsymbol{\sigma}}}), \min_{\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G})\in\mathsf{args}\left(\mathsf{EMM}_{f(\overline{\boldsymbol{\sigma}})}^{\overline{\boldsymbol{\sigma}}}\right)} \quad \max_{e\in\mathsf{args}(\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G}))} e\right) \\
&= \max_{\boldsymbol{h}\in\Gamma\cap\mathsf{ED}} \quad \max_{f\in\mathcal{F}} \quad \max_{g_f\in\mathcal{G}_f} \quad \min_{\overline{\boldsymbol{\sigma}}\in\overline{\pi}(\mathcal{V})} \min\left(h(U_{f(\overline{\boldsymbol{\sigma}})}^{\overline{\boldsymbol{\sigma}}}), \min_{\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G})\in\mathsf{args}\left(\mathsf{EMM}_{f(\overline{\boldsymbol{\sigma}})}^{\overline{\boldsymbol{\sigma}}}\right)} g_f(\overline{\boldsymbol{\sigma}}, \mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G}))\right) \\
&= \max_{f\in\mathcal{F}} \quad \max_{g_f\in\mathcal{G}_f} \\
&\qquad \underbrace{\max_{\boldsymbol{h}\in\Gamma\cap\mathsf{ED}} \quad \min_{\overline{\boldsymbol{\sigma}}\in\overline{\pi}(\mathcal{V})} \quad \min\left(h(U_{f(\overline{\boldsymbol{\sigma}})}^{\overline{\boldsymbol{\sigma}}}), \min_{\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G})\in\mathsf{args}\left(\mathsf{EMM}_{f(\overline{\boldsymbol{\sigma}})}^{\overline{\boldsymbol{\sigma}}}\right)} g_f(\overline{\boldsymbol{\sigma}}, \mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G}))\right)}_{\text{Equivalent to an LP}} \qquad (32)
\end{aligned}$$

The above expression can be rewritten into the following form:

$$\begin{aligned}
&\omega\text{-subw}(\mathcal{H}) = \max_{i\in[\mathcal{I}]} \\
&\qquad \underbrace{\max_{\boldsymbol{h}\in\Gamma\cap\mathsf{ED}} \quad \min\big(\min_{\ell\in[L_i]} h(\boldsymbol{U}_{i\ell}), \quad \min_{j\in[J_i]} h(\boldsymbol{X}_{ij}|\boldsymbol{G}_{ij}) + h(\boldsymbol{Y}_{ij}|\boldsymbol{G}_{ij}) + \gamma h(\boldsymbol{Z}_{ij}|\boldsymbol{G}_{ij}) + h(\boldsymbol{G}_{ij})\big)}_{\text{Equivalent to LP (49) \textit{assuming} } \gamma \text{ is fixed}} \qquad (33)
\end{aligned}$$

For a fixed $i \in [\mathcal{I}]$, the inner expression is equivalent to an LP. See Appendix D for more details.

7 ALGORITHM FOR ANSWERING QUERIES IN ω-SUBMODULAR WIDTH TIME

We show below our main result about evaluating a Boolean conjunctive query of the form (1) in ω-submodular width time. The theorem assumes that ω is a rational number, but can still be applied to the case where ω is irrational by taking any rational upper bound on ω. The runtime is measured in terms of N, which is the size of the input database instance, i.e., $N \stackrel{\text{def}}{=} \sum_{R(\boldsymbol{Z})\in\mathsf{atoms}(Q)} |R|$. The $\tilde{O}$-notation hides a polylogarithmic factor in N.

Theorem 7.1. *Assuming that ω is a rational number, given a Boolean conjunctive query Q and a corresponding input database instance D, there is an algorithm that computes the answer to Q in time $\tilde{O}(N^{\omega\text{-subw}(Q)})$, where N is the size of D.*

The proof is quite involved and requires developing a series of technical tools. It also relies on the algorithm for computing the ω-submodular width of a hypergraph, which was given in Section 6. The full proof can be found in Appendix E.

8 CONCLUSION

We proposed a general framework for evaluating Boolean conjunctive queries that naturally subsumes both combinatorial and non-combinatorial techniques under the umbrella of information theory. Our framework generalizes the notion of submodular width by incorporating matrix multiplication, and provides a matching algorithm for evaluating queries in the corresponding time complexity. Using this framework, we show how to recover the best known complexity for various classes of queries as well as improve the complexity for some of them.

Our framework can be straightforwardly extended from Boolean conjunctive queries to count queries and more generally to aggregate queries over any semiring, albeit *without* free variables. In particular, in the combinatorial world, the transition from Boolean to aggregate queries is done by adapting the submodular width to become the *Sharp-submodular width* [1]. The latter is a variant of the submodular width that is obtained by relaxing the notion of polymatroids. In the presence of matrix multiplication, a similar transition from Boolean to aggregate queries is possible by adapting the ω-submodular width to become the *Sharp-ω-submodular width.*

REFERENCES

[1] Mahmoud Abo Khamis, Ryan R. Curtin, Benjamin Moseley, Hung Q. Ngo, Xuanlong Nguyen, Dan Olteanu, and Maximilian Schleich. 2020. Functional Aggregate Queries with Additive Inequalities. *ACM Trans. Database Syst.* 45, 4, Article 17 (dec 2020), 41 pages. https://doi.org/10.1145/3426865

[2] Mahmoud Abo Khamis, Hung Q. Ngo, and Atri Rudra. 2016. FAQ: Questions Asked Frequently. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, Tova Milo and Wang-Chiew Tan (Eds.). ACM, 13–28. https://doi.org/10.1145/2902251.2902280

[3] Mahmoud Abo Khamis, Hung Q. Ngo, and Dan Suciu. 2017. What Do Shannon-type Inequalities, Submodular Width, and Disjunctive Datalog Have to Do with One Another?. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017*, Emanuel Sallinger, Jan Van den Bussche, and Floris Geerts (Eds.). ACM, 429–444. https://doi.org/10.1145/3034786.3056105 Extended version available at http://arxiv.org/abs/1612.02503.

[4] Mahmoud Abo Khamis, Hung Q. Ngo, and Dan Suciu. 2024. PANDA: Query Evaluation in Submodular Width. *arXiv e-prints*, Article arXiv:2402.02001 (Feb. 2024), arXiv:2402.02001 pages. https://doi.org/10.48550/arXiv.2402.02001 arXiv:2402.02001 [cs.DB]

[5] Noga Alon, Raphael Yuster, and Uri Zwick. 1995. Color-coding. *Journal of the ACM (JACM)* 42, 4 (1995), 844–856.

[6] Noga Alon, Raphael Yuster, and Uri Zwick. 1997. Finding and Counting Given Length Cycles. *Algorithmica* 17, 3 (1997), 209–223. https://doi.org/10.1007/BF02523189

[7] Albert Atserias, Martin Grohe, and Dániel Marx. 2008. Size Bounds and Query Plans for Relational Joins. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*. IEEE Computer Society, 739–748. https://doi.org/10.1109/FOCS.2008.43

[8] Albert Atserias, Martin Grohe, and Dániel Marx. 2013. Size Bounds and Query Plans for Relational Joins. *SIAM J. Comput.* 42, 4 (2013), 1737–1767. https://doi.org/10.1137/110859440

[9] Mina Dalirrooyfard, Surya Mathialagan, Virginia Vassilevska Williams, and Yinzhan Xu. 2024. Towards Optimal Output-Sensitive Clique Listing or: Listing Cliques from Smaller Cliques. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. 923–934.

[10] Mina Dalirrooyfard, Thuy Duong Vuong, and Virginia Vassilevska Williams. 2019. Graph pattern detection: Hardness for all induced patterns and faster non-induced cycles. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. 1167–1178.

[11] Rina Dechter. 1999. Bucket elimination: a unifying framework for reasoning. *Artif. Intell.* 113, 1–2 (Sept. 1999), 41–85. https://doi.org/10.1016/S0004-3702(99)00059-4

[12] Rina Dechter. 2019. *Reasoning with Probabilistic and Deterministic Graphical Models.* Springer Cham. https://doi.org/10.1007/978-3-031-01583-0

[13] Friedrich Eisenbrand and Fabrizio Grandoni. 2004. On the complexity of fixed parameter clique and dominating set. *Theoretical Computer Science* 326, 1-3 (2004), 57–67.

[14] Francois Le Gall and Florent Urrutia. [n. d.]. *Improved Rectangular Matrix Multiplication using Powers of the Coppersmith-Winograd Tensor.* 1029–1046. https://doi.org/10.1137/1.9781611975031.67 arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611975031.67

[15] Martin Grohe and Dániel Marx. 2006. Constraint solving via fractional edge covers. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006.* ACM Press, 289–298. http://dl.acm.org/citation.cfm?id=1109557.1109590

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

[17] 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

[18] Hung Q. Ngo. 2018. Worst-Case Optimal Join Algorithms: Techniques, Results, and Open Problems. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems* (Houston, TX, USA) *(PODS '18)*. Association for Computing Machinery, New York, NY, USA, 111–124. https://doi.org/10.1145/3196959.3196990

[19] Hung Q. Ngo, Ely Porat, Christopher Ré, and Atri Rudra. 2018. Worst-case Optimal Join Algorithms. *J. ACM* 65, 3, Article 16 (March 2018), 40 pages. https://doi.org/10.1145/3180143

[20] Hung Q. Ngo, Christopher Ré, and Atri Rudra. 2013. Skew strikes back: new developments in the theory of join algorithms. *SIGMOD Rec.* 42, 4 (2013), 5–16. https://doi.org/10.1145/2590989.2590991

[21] Volker Strassen. 1969. Gaussian elimination is not optimal. *Numer. Math.* 13 (1969), 354–356. https://api.semanticscholar.org/CorpusID:121656251

[22] Todd L. Veldhuizen. 2014. Triejoin: A Simple, Worst-Case Optimal Join Algorithm. In *Proc. 17th International Conference on Database Theory (ICDT), Athens, Greece, March 24-28, 2014*, Nicole Schweikardt, Vassilis Christophides, and Vincent Leroy (Eds.). OpenProceedings.org, 96–106. https://doi.org/10.5441/002/icdt.2014.13

[23] Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. [n. d.]. *New Bounds for Matrix Multiplication: from Alpha to Omega.* 3792–3835. https://doi.org/10.1137/1.9781611977912.134 arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611977912.134

[24] Raphael Yuster and Uri Zwick. 2004. Detecting short directed cycles using rectangular matrix multiplication and dynamic programming.. In *SODA*, Vol. 4. 254–260.

[25] Nevin Lianwen Zhang and David Poole. 1996. Exploiting causal independence in Bayesian network inference. *J. Artif. Int. Res.* 5, 1 (Dec. 1996), 301–328.

[26] Nevin Lianwen Zhang and David L. Poole. 1994. A simple approach to Bayesian network computations. https://api.semanticscholar.org/CorpusID:2978086

A MISSING DETAILS FROM SECTION 3

In this appendix, we provide further examples and details that are related to Section 3.

A.1 Hypergraphs

Example A.1 ($\partial_{\mathcal{H}}(X)$, $U_{\mathcal{H}}(X)$ and $N_{\mathcal{H}}(X)$). Consider a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ with vertices $\mathcal{V} = \{A, B, C, D, E\}$ and hyperedges $\mathcal{E} = \{\{A, B, C\}, \{A, B, D\}, \{C, D, E\}\}$. Then:

$$\partial(A) = \{\{A, B, C\}, \{A, B, D\}\}, \qquad U(A) = \{A, B, C, D\}, \qquad N(A) = \{B, C, D\}.$$

A.2 Tree Decompositions

Example A.2 (Tree Decompositions of a Hypergraph). Consider the following hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ that represents a 4-cycle:

$$\mathcal{V} = \{A, B, C, D\}, \qquad \mathcal{E} = \{\{A, B\}, \{B, C\}, \{C, D\}, \{D, A\}\} \tag{34}$$

This hypergraph has the following two (non-trivial and non-redundant) tree decompositions:

- A tree decomposition (T_1, χ_1) with two nodes t_{11} and t_{12} corresponding to two bags, $\chi_1(t_{11}) = \{A, B, C\}$ and $\chi_1(t_{12}) = \{C, D, A\}$.
- A tree decomposition (T_2, χ_2) with two nodes t_{21} and t_{22} corresponding to two bags, $\chi_2(t_{21}) = \{B, C, D\}$ and $\chi_2(t_{22}) = \{D, A, B\}$.

A.3 Variable Elimination

Example A.3 (Elimination Hypergraph Sequence). Consider the 4-cycle hypergraph $\mathcal{H}$ from Example A.2 and the variable elimination order $\boldsymbol{\sigma}_1 = (B, C, D, A)$. This elimination order results in the following sequence of hypergraphs:

$$\begin{aligned}
\mathcal{H}_1^{\boldsymbol{\sigma}_1} &= (\{A, B, C, D\}, \{\{A, B\}, \{B, C\}, \{C, D\}, \{D, A\}\}) \\
\mathcal{H}_2^{\boldsymbol{\sigma}_1} &= (\{A, C, D\}, \{\{A, C\}, \{C, D\}, \{D, A\}\}) \\
\mathcal{H}_3^{\boldsymbol{\sigma}_1} &= (\{A, D\}, \{\{D, A\}\})
\end{aligned}$$

$$\mathcal{H}_4^{\boldsymbol{\sigma}_1} = (\{A\}, \{\{A\}\})$$

In contrast, the order $\boldsymbol{\sigma}_2 = (A, B, C, D)$ results in the following sequence:

$$\begin{aligned}
\mathcal{H}_1^{\boldsymbol{\sigma}_2} &= (\{A, B, C, D\}, \{\{A, B\}, \{B, C\}, \{C, D\}, \{D, A\}\}) \\
\mathcal{H}_2^{\boldsymbol{\sigma}_2} &= (\{B, C, D\}, \{\{D, B\}, \{B, C\}, \{C, D\}\}) \\
\mathcal{H}_3^{\boldsymbol{\sigma}_2} &= (\{C, D\}, \{\{C, D\}\}) \\
\mathcal{H}_4^{\boldsymbol{\sigma}_2} &= (\{D\}, \{\{D\}\})
\end{aligned}$$

Example A.4 (Equivalence of Tree Decompositions and Variable Elimination Orders). Continuing with Example A.3, consider the 4-cycle hypergraph in Eq. (34). The variable elimination order $\boldsymbol{\sigma}_1 = (B, C, D, A)$ results in:

$$U_1^{\boldsymbol{\sigma}_1} = \{A, B, C\}, \quad U_2^{\boldsymbol{\sigma}_1} = \{C, D, A\}, \quad U_3^{\boldsymbol{\sigma}_1} = \{D, A\}, \quad U_4^{\boldsymbol{\sigma}_1} = \{A\}.$$

Note that $\boldsymbol{\sigma}_1$ "subsumes" the tree decomposition (T_1, χ_1) from Example A.2, in the sense that the pair $((T_1, \chi_1), \boldsymbol{\sigma}_1)$ satisfies part 1 of Proposition 3.1. Also, (T_1, χ_1) "subsumes" $\boldsymbol{\sigma}_1$ in the sense that the pair $((T_1, \chi_1), \boldsymbol{\sigma}_1)$ satisfies part 2 of Proposition 3.1.

In contrast, the variable order $\boldsymbol{\sigma}_2 = (A, B, C, D)$ results in:

$$U_1^{\boldsymbol{\sigma}_2} = \{D, A, B\}, \quad U_2^{\boldsymbol{\sigma}_2} = \{B, C, D\}, \quad U_3^{\boldsymbol{\sigma}_2} = \{C, D\}, \quad U_4^{\boldsymbol{\sigma}_2} = \{D\}.$$

In particular, $\boldsymbol{\sigma}_2$ above subsumes and is subsumed by the tree decomposition (T_2, χ_2).

A.4 The Submodular Width

We summarize here how to compute the submodular width of a given hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, as defined in Eq. (19). To that end, we need to apply a couple of tricks. Let $\mathcal{T}(\mathcal{H}) = \{(T_1, \chi_1), (T_2, \chi_2), \ldots, (T_K, \chi_K)\}$ be the list of all tree decompositions of $\mathcal{H}$. Let $\boldsymbol{t} = (t_1, t_2, \ldots, t_K) \in \mathsf{nodes}(T_1) \times \ldots \times \mathsf{nodes}(T_K)$ denote a tuple of nodes, one from each tree decomposition. (Note that $t_1, \ldots, t_K$ are *not* nodes from the same tree decomposition.) By distributing[12] the min over the max in Eq. (19), we can equivalently write (where $[K]$ denotes $\{1, \ldots, K\}$):

$$\begin{aligned}
\mathsf{subw}(\mathcal{H}) &\stackrel{\text{def}}{=} \max_{\boldsymbol{h} \in \Gamma \cap \mathsf{ED}} \min_{(T, \chi) \in \mathcal{T}(\mathcal{H})} \max_{t \in \mathsf{nodes}(T)} h(\chi(t)) \\
&= \max_{\boldsymbol{h} \in \Gamma \cap \mathsf{ED}} \max_{\substack{(t_1, \ldots, t_K) \in \\ \mathsf{nodes}(T_1) \times \ldots \times \mathsf{nodes}(T_K)}} \min_{i \in [K]} h(\chi_i(t_i)) && (35) \\
&= \max_{\substack{(t_1, \ldots, t_K) \in \\ \mathsf{nodes}(T_1) \times \ldots \times \mathsf{nodes}(T_K)}} \underbrace{\max_{\boldsymbol{h} \in \Gamma \cap \mathsf{ED}} \min_{i \in [K]} h(\chi_i(t_i))}_{\text{Equivalent to LP (38)}} && (36)
\end{aligned}$$

Eq. (36) follows by swapping the two max operators in Eq. (35). Now for a *fixed* tuple $(t_1, \ldots, t_K)$, consider the inner optimization problem:

$$\max_{\boldsymbol{h} \in \Gamma \cap \mathsf{ED}} \min_{i \in [K]} h(\chi_i(t_i)) \tag{37}$$

Although the constraints $\boldsymbol{h} \in \Gamma \cap \mathsf{ED}$ are linear, Eq. (37) is still not a linear program (LP) because the objective is a minimum of linear functions. However, we can convert (37) into an LP by introducing a new variable w that is upper bounded by each term in the minimum:

$$\max_{w, \boldsymbol{h} \in \Gamma \cap \mathsf{ED}} \{w \mid w \leq h(\chi_1(t_1)), \ldots, w \leq h(\chi_K(t_K))\} \tag{38}$$

[12]Note that $(\mathbb{R}, \max, \min)$ is a *commutative semiring*.

As a result, computing the submodular width can be reduced to taking the maximum solution of a finite number of LPs, one for each tuple $(t_1, \dots, t_K) \in \text{nodes}(T_1) \times \dots \times \text{nodes}(T_K)$.

Example A.5. Consider the 4-cycle hypergraph from Example A.2. The submodular width of this hypergraph can be computed by considering the two tree decompositions (T_1, χ_1) and (T_2, χ_2) from Example A.2. Applying Eq. (19), we get:

$$\text{subw}(\mathcal{H}) = \max_{\boldsymbol{h} \in \Gamma \cap \text{ED}} \min(\max(h(ABC), h(CDA)), \max(h(BCD), h(DAB)))$$

By distributing the min over the inner max and then swapping the two max operators, we get:

$$\text{subw}(\mathcal{H}) = \max\Bigg(\max_{\boldsymbol{h} \in \Gamma \cap \text{ED}} \min(h(ABC), h(BCD)), \quad \max_{\boldsymbol{h} \in \Gamma \cap \text{ED}} \min(h(ABC), h(DAB)),$$
$$\max_{\boldsymbol{h} \in \Gamma \cap \text{ED}} \min(h(CDA), h(BCD)), \quad \max_{\boldsymbol{h} \in \Gamma \cap \text{ED}} \min(h(CDA), h(DAB)) \Bigg)$$

Let's take the first term inside the max as an example $\max_{\boldsymbol{h} \in \Gamma \cap \text{ED}} \min(h(ABC), h(BCD))$. This term is equivalent to the following LP:

$$\max_{w, \boldsymbol{h} \in \Gamma \cap \text{ED}} \{ w \mid w \leq h(ABC), w \leq h(BCD) \}$$

The above LP has an optimal objective value of $3/2$. The other three terms inside the max are similar. Therefore, the submodular width of the 4-cycle hypergraph is $3/2$.

B MISSING DETAILS FROM SECTION 4

This appendix presents missing proofs from Section 4.

Proposition 4.8. *The following is an equivalent definition of the submodular width of $\mathcal{H}$:*

$$\textit{subw}(\mathcal{H}) \quad \stackrel{\text{def}}{=} \quad \max_{\boldsymbol{h} \in \Gamma \cap \text{ED}} \quad \min_{\overline{\boldsymbol{\sigma}} \in \overline{\pi}(\mathcal{V})} \quad \max_{i \in [|\overline{\boldsymbol{\sigma}}|]} \quad h(U_i^{\overline{\boldsymbol{\sigma}}})$$

Proof. The above definition is obviously upper bounded by Eq. (20) since $\overline{\pi}(\mathcal{V})$ is a superset of $\pi(\mathcal{V})$. In order to show the opposite direction, consider an arbitrary $\overline{\boldsymbol{\sigma}} = (\boldsymbol{X}_1, \dots, \boldsymbol{X}_{|\overline{\boldsymbol{\sigma}}|}) \in \overline{\pi}(\mathcal{V})$. Consider the non-negative quantity $\varphi(|\overline{\boldsymbol{\sigma}}|) \stackrel{\text{def}}{=} \sum_{i \in [|\overline{\boldsymbol{\sigma}}|]} (|\boldsymbol{X}_i| - 1)$. As long as $\overline{\boldsymbol{\sigma}}$ contains some $\boldsymbol{X}_i$ whose size it at least 2, this quantity will be positive, and we will construct another generalized variable elimination order $\overline{\boldsymbol{\sigma}'}$ that reduces this quantity and where every $U_j^{\overline{\boldsymbol{\sigma}'}}$ is a subset of some $U_i^{\overline{\boldsymbol{\sigma}}}$. WLOG assume that $|\boldsymbol{X}_1| > 1$. Let Y be an arbitrary element of $\boldsymbol{X}_1$, and define $\overline{\boldsymbol{\sigma}'} = (\{Y\}, \boldsymbol{X} \setminus \{Y\}, \boldsymbol{X}_2, \dots, \boldsymbol{X}_{|\overline{\boldsymbol{\sigma}}|})$. Clearly, both $U_1^{\overline{\boldsymbol{\sigma}'}}$ and $U_2^{\overline{\boldsymbol{\sigma}'}}$ are subsets of $U_1^{\overline{\boldsymbol{\sigma}}}$. Moreover, $\mathcal{H}_3^{\overline{\boldsymbol{\sigma}'}}$ is identical to $\mathcal{H}_2^{\overline{\boldsymbol{\sigma}}}$, hence by induction, every $U_j^{\overline{\boldsymbol{\sigma}'}}$ is a subset of some $U_i^{\overline{\boldsymbol{\sigma}}}$. □

Proposition 4.11. *The ω-submodular width of a hypergraph can be equivalently defined as follows:*

$$\omega\textit{-subw}(\mathcal{H}) \quad \stackrel{\text{def}}{=} \quad \max_{\boldsymbol{h} \in \Gamma \cap \text{ED}} \quad \min_{\overline{\boldsymbol{\sigma}} \in \overline{\pi}(\mathcal{V})} \quad \max_{\substack{i \in [|\overline{\boldsymbol{\sigma}}|] \\ \forall j < i, U_i^{\overline{\boldsymbol{\sigma}}} \not\subseteq U_j^{\overline{\boldsymbol{\sigma}}}}} \quad \min(h(U_i^{\overline{\boldsymbol{\sigma}}}), \text{EMM}_i^{\overline{\boldsymbol{\sigma}}})$$

Proof. Consider an arbitrary polymatroid $\boldsymbol{h} \in \Gamma \cap \text{ED}$ and a generalized elimination order $\overline{\boldsymbol{\sigma}} \in \overline{\pi}(\mathcal{V})$. It suffices to show that for any pair (j, i) where $j < i$ and $U_i^{\overline{\boldsymbol{\sigma}}} \subseteq U_j^{\overline{\boldsymbol{\sigma}}}$, we must have $h(U_i^{\overline{\boldsymbol{\sigma}}}) \leq \min \left\{ h(U_j^{\overline{\boldsymbol{\sigma}}}), \text{EMM}_j^{\overline{\boldsymbol{\sigma}}} \right\}$. In this case, we further notice that $\boldsymbol{X}_j \cap U_i^{\overline{\boldsymbol{\sigma}}} = \emptyset$ since $j < i$ hence the variables $\boldsymbol{X}_j$ must have been eliminated before $U_i^{\overline{\boldsymbol{\sigma}}}$ was created. This implies that $U_i^{\overline{\boldsymbol{\sigma}}} \subseteq U_j^{\overline{\boldsymbol{\sigma}}} \setminus \boldsymbol{X}_j$.

By the monotonicity of $\boldsymbol{h}$ (Eq. (15)), $h(U_i^{\overline{\sigma}}) \le h(U_j^{\overline{\sigma}} \setminus \boldsymbol{X}_j) \le h(U_j^{\overline{\sigma}})$. Moreover, by Proposition 4.3, we have $h(U_j^{\overline{\sigma}} \setminus \boldsymbol{X}_j) \le \mathsf{EMM}_j^{\overline{\sigma}}$. □

C MISSING DETAILS FROM SECTION 5

We show here how to compute the ω-submodular width for the classes of queries that are given by Table 2 in Section 5. We start with some preliminaries:

Definition C.1 (Fractional Edge Covering Number). For a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, a fractional edge covering is a function $W : \mathcal{E} \to [0, 1]$ such that $\sum_{e \in \mathcal{E}: X \in e} W(e) \ge 1$ holds for each vertex $X \in \mathcal{V}$. The fractional edge covering number of $\mathcal{H}$, denoted as $\rho^*(\mathcal{H})$, is defined as the minimum weight assigned to hyperedges over all possible fractional edge covering, i.e., $\rho^*(\mathcal{H}) = \min_W \sum_{e \in \mathcal{E}} W(e)$.

Proposition C.2. *In any hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, for any polymatroid $\boldsymbol{h} \in \Gamma \cap \mathsf{ED}$, $h(\mathcal{V}) \le \rho^*(\mathcal{H})$.*

C.1 Clique Hypergraphs

Proposition C.3. *In a k-clique hypergraph $\mathcal{H} = (\{X_1, X_2, \cdots, X_k\}, \{\{X_i, X_j\} : i, j \in [k], i \ne j\})$, for any generalized elimination ordering $\overline{\sigma} \in \pi(\mathcal{V})$ and any $i \in [\overline{\sigma}]$, $U_i^{\overline{\sigma}} \subseteq \mathcal{V} = U_1^{\overline{\sigma}}$.*

From Proposition C.3 and Definition 4.7, we can simplify ω-subw for clique hypergraphs $\mathcal{H}$ as below:

$$\omega\text{-subw}(\mathcal{H}) \quad = \quad \max_{\boldsymbol{h} \in \Gamma \cap \mathsf{ED}} \quad \min\left(h(\mathcal{V}), \min_{\overline{\sigma} \in \overline{\pi}(\mathcal{V})} \mathsf{EMM}_1^{\overline{\sigma}}\right) \tag{39}$$

Proposition C.4. *In a clique hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, with a polymatroid $\boldsymbol{h} \in \Gamma \cap \mathsf{ED}$ such that $h(\boldsymbol{A}|\boldsymbol{B}) = h(\boldsymbol{A})$ for arbitrary $\boldsymbol{A}, \boldsymbol{B} \subseteq \mathcal{V}$ with $\boldsymbol{A} \cap \boldsymbol{B} = \emptyset$, for any generalized elimination ordering $\overline{\sigma} \in \pi(\mathcal{V})$,*

$$\mathsf{EMM}_1^{\overline{\sigma}} = \min_{(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}) \text{ is a partition of } \mathcal{V}} \mathsf{MM}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$$

Proof. The direction $\mathsf{EMM}_1^{\overline{\sigma}} \ge \min_{(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}) \text{ is a partition of } \mathcal{V}} \mathsf{MM}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}; \emptyset)$ follows (4.5). Below, we focus on:

$$\begin{aligned}
\mathsf{EMM}_1^{\overline{\sigma}} &= \min_{(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}) \text{ is a partition of } \mathcal{V}} \mathsf{MM}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}|\boldsymbol{D}) \\
&= \min_{(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}) \text{ is a partition of } \mathcal{V}} \min \begin{cases} \gamma h(\boldsymbol{A}|\boldsymbol{D}) + h(\boldsymbol{B}|\boldsymbol{D}) + h(\boldsymbol{C}|\boldsymbol{D}) + h(\boldsymbol{D}) \\ h(\boldsymbol{A}|\boldsymbol{D}) + \gamma h(\boldsymbol{B}|\boldsymbol{D}) + h(\boldsymbol{C}|\boldsymbol{D}) + h(\boldsymbol{D}) \\ h(\boldsymbol{A}|\boldsymbol{D}) + h(\boldsymbol{B}|\boldsymbol{D}) + \gamma h(\boldsymbol{C}|\boldsymbol{D}) + h(\boldsymbol{D}) \end{cases} \\
&= \min_{(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}) \text{ is a partition of } \mathcal{V}} \min \begin{cases} \gamma h(\boldsymbol{A}) + h(\boldsymbol{B}) + h(\boldsymbol{C}) + h(\boldsymbol{D}) \\ h(\boldsymbol{A}) + \gamma h(\boldsymbol{B}) + h(\boldsymbol{C}) + h(\boldsymbol{D}) \\ h(\boldsymbol{A}) + h(\boldsymbol{B}) + \gamma h(\boldsymbol{C}) + h(\boldsymbol{D}) \end{cases} \\
&\le \min_{(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}) \text{ is a partition of } \mathcal{V}} \min \begin{cases} \gamma h(\boldsymbol{A}) + h(\boldsymbol{B}) + h(\boldsymbol{C}\boldsymbol{D}) \\ h(\boldsymbol{A}) + \gamma h(\boldsymbol{B}) + h(\boldsymbol{C}\boldsymbol{D}) \\ h(\boldsymbol{A}) + h(\boldsymbol{B}) + \gamma h(\boldsymbol{C}\boldsymbol{D}) \end{cases} \\
&= \min_{(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}) \text{ is a partition of } \mathcal{V}} \mathsf{MM}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})
\end{aligned}$$

Together, we have completed the proof. □

Lemma C.5. *For the 3-clique hypergraph as defined in (2), ω-subw$(\mathcal{H}) = \frac{2\omega}{\omega+1}$.*

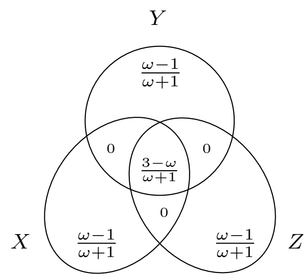

Fig. 2. Diagram of the polymatroid for the clique-3 hypergraph.

Proof. <u>*Direction ω-subw$(\mathcal{H}) \leq \frac{2\omega}{\omega+1}$*</u>. We partition all polymatroids $h \in \Gamma \cap \mathsf{ED}$ into the following two cases:

- Case 1: there exists a pair of vertices with their conditional entropy is smaller than $\frac{\omega-1}{\omega+1}$, say $h(Y|X) \leq \frac{\omega-1}{\omega+1}$. In this case, we have $h(XYZ) = h(ZX)+h(Y|ZX) \leq h(ZX)+h(Y|X) \leq 1+\frac{\omega-1}{\omega+1} = \frac{2\omega}{\omega+1}$, where $h(Y|ZX) \leq h(Y|X)$ follows the sub-modularity of h and $h(ZX) \leq 1$ follows the edge-domination of h.
- Case 2: None of the above, i.e., for each pair of vertices, their conditional entropy is larger than $\frac{\omega-1}{\omega+1}$. As $h(Y|X) > \frac{\omega-1}{\omega+1}$ and $h(XY) \leq 1$, we have $h(X) = h(XY) - h(Y|X) \leq 1 - \frac{\omega-1}{\omega+1} = \frac{2}{\omega+1}$. Similarly, we have $h(Y), h(Z) \leq \frac{2}{\omega+1}$. Consider a generalized elimination ordering $\overline{\boldsymbol{\sigma}} = (\{X\},\{Y\},\{Z\})$. We have $\mathsf{EMM}_1^{\overline{\sigma}} \leq \mathsf{MM}(Y;Z;X) = \frac{2\omega}{\omega+1}$ by choosing $\mathcal{A} = \{\{X,Y\}\}$ and $\mathcal{B} = \{\{X,Z\}\}$.

<u>*Direction ω-subw$(\mathcal{H}) \geq \frac{2\omega}{\omega+1}$*</u>. We identify a polymatroid $\boldsymbol{h}$ below. Let a, b, c, d be independently random variables with $h(a) = h(b) = h(c) = \frac{\omega-1}{\omega+1}$ and $h(d) = \frac{3-\omega}{\omega+1}$. Let $X = (ad), Y = (bd), Z = (cd)$.

- $h(X) = h(Y) = h(Z) = \frac{2}{\omega+1}$;
- $h(XY) = h(XZ) = h(YZ) = 1$;
- $h(XYZ) = \frac{2\omega}{\omega+1}$.

Consider a generalized elimination ordering $\overline{\boldsymbol{\sigma}} = (\{X\},\{Y\},\{Z\})$, $\mathsf{EMM}_1^{\overline{\sigma}} = \mathsf{MM}(Y,Z,X) = \frac{2\omega}{\omega+1}$. Hence, $\min\left(h(\mathcal{V}), \min_{\overline{\boldsymbol{\sigma}} \in \overline{\pi}(\mathcal{V})} \mathsf{EMM}_1^{\overline{\sigma}}\right) = \frac{2\omega}{\omega+1}$ holds for $\boldsymbol{h}$. From (39), we conclude ω-subw$(\mathcal{H}) \geq \frac{2\omega}{\omega+1}$. The other generalized elimination orderings are the same. □

Lemma C.6. *For the 4-clique hypergraph as defined in (23), ω-subw$(\mathcal{H}) = \frac{\omega+1}{2}$.*

Proof. <u>*Direction ω-subw$(\mathcal{H}) \leq \frac{\omega+1}{2}$*</u>. We partition all polymatroids $\boldsymbol{h} \in \Gamma \cap \mathsf{ED}$ into the following two cases:

Case 1: there exists a vertex such that its all conditional entropies are smaller than $\frac{1}{2}$, say, W is such a vertex with $h(X|W), h(Y|W), h(Z|W) \leq \frac{1}{2}$. Consider a generalized elimination ordering $\overline{\boldsymbol{\sigma}} = (\{X\},\{Y\},\{Z\},\{W\})$. By choosing $\mathcal{A} = \{\{X,W\},\{Y,W\}\}$ and $\mathcal{B} = \{\{X,W\},\{Z,W\}\}$,

$$\mathsf{EMM}_1^{\overline{\sigma}} \leq \mathsf{MM}(Y;Z;X|W) = \max\begin{cases} \gamma h(Y|W) + h(X|W) + h(Z|W) + h(W) \\ h(Y|W) + \gamma h(Z|W) + h(X|W) + h(W) \\ h(Y|W) + h(Z|W) + \gamma h(X|W) + h(W) \end{cases} \leq \frac{\gamma}{2} + \frac{1}{2} + 1 = \frac{\omega+1}{2}.$$

Case 2: None of the above, i.e., each vertex has at least one conditional entropy larger than $\frac{1}{2}$. Wlog, suppose $h(Y|X) > \frac{1}{2}$. As $h(XY) \leq 1$, we have $h(X) = h(XY) - h(Y|X) < 1 - \frac{1}{2} = \frac{1}{2}$. Similarly, we have $h(Y), h(Z), h(W) < \frac{1}{2}$. Consider a generalized elimination ordering $\overline{\boldsymbol{\sigma}} = (\{X\}, \{Y\}, \{Z\}, \{W\})$. We have $\mathsf{EMM}_1^{\overline{\sigma}} \leq \mathsf{MM}(Y; Z; X|W) = \frac{\omega+1}{2}$.

Direction ω-subw$(\mathcal{H}) \geq \frac{\omega+1}{2}$. We identify a polymatroid $\boldsymbol{h} \in \Gamma \cap \mathsf{ED}$ as follows: let X, Y, Z, W be independently random variables with $h(X) = h(Y) = h(Z) = h(W) = \frac{1}{2}$. Then, we have $h(\mathcal{V}) = 2$. From Proposition C.4, for an arbitrary generalized elimination ordering $\overline{\boldsymbol{\sigma}} \in \pi(\mathcal{V})$, $\mathsf{EMM}_1^{\overline{\sigma}} = \frac{\omega+1}{2}$. From (39), we conclude ω-subw$(\mathcal{H}) \geq \frac{\omega+1}{2}$. □

Lemma C.7. *For the following 5-clique hypergraph $\mathcal{H}$, ω-subw$(\mathcal{H}) = \frac{\omega}{2} + 1$.*

$$\mathcal{H} = (\{X, Y, Z, W, L\}, \{\{X, Y\}, \{X, Z\}, \{X, W\}, \{X, L\}, \{Y, Z\}, \{Y, W\}, \{Y, L\}, \{Z, W\}, \{Z, L\}, \{W, L\}\}) \tag{40}$$

Proof. *Direction ω-subw$(\mathcal{H}) \leq \frac{\omega}{2} + 1$.* We partition all polymatroids $\boldsymbol{h} \in \Gamma \cap \mathsf{ED}$ into the following five cases:

- Case 1: there exists a vertex such that all its conditional entropies are smaller than $\frac{1}{2}$, say, L is such a vertex with $h(X|L), h(Y|L), h(Z|L), h(W|L) \leq \frac{1}{2}$. Consider a generalized elimination ordering $\overline{\boldsymbol{\sigma}} = (\{X\}, \{YZ\}, \{W\}, \{L\})$. By choosing $\mathcal{A} = \{\{X, W\}, \{Y, W\}, \{X, L\}\}$ and $\mathcal{B} = \{\{X, W\}, \{Z, W\}\}$,

$$\begin{aligned}\mathsf{EMM}_1^{\overline{\sigma}} \leq \mathsf{MM}(YZ, W, X|L) &= h(YZ) + \gamma \cdot \min\{h(W|L), h(X|L)\} + \max\{h(W|L), h(X|L)\} + h(L) \\ &\leq 1 + \frac{\gamma}{2} + 1 = \frac{\omega}{2} + 1.\end{aligned}$$

- Case 2: None of the above, i.e., each vertex has at least one conditional entropy larger than $\frac{1}{2}$. Suppose $h(Y|X) > \frac{1}{2}$ for vertex x. As $h(XY) \leq 1$, we have $h(X) = h(XY) - h(Y|X) < 1 - \frac{1}{2} = \frac{1}{2}$. Similarly, we have $h(Y), h(Z), h(W), h(L) < \frac{1}{2}$. Consider a generalized elimination ordering $\overline{\boldsymbol{\sigma}} = (\{X\}, \{Y\}, \{Z\}, \{W\}, \{L\})$. We have $\mathsf{EMM}_1^{\overline{\sigma}} \leq \mathsf{MM}(YZ; WL; X) \leq \frac{\omega}{2} + 1$.

Direction ω-subw$(\mathcal{H}) \geq \frac{\omega}{2} + 1$. We identify a polymatroid $\boldsymbol{h} \in \Gamma \cap \mathsf{ED}$ as follows: let X, Y, Z, W, L be independently random variables with $h(X) = h(Y) = h(Z) = h(W) = h(L) = \frac{1}{2}$. Then, we have $h(\mathcal{V}) = \frac{5}{2}$. From Proposition C.4, for an arbitrary generalized elimination ordering $\overline{\boldsymbol{\sigma}} \in \pi(\mathcal{V})$, $\mathsf{EMM}_1^{\overline{\sigma}} = \frac{\omega}{2} + 1$. □

Lemma C.8. *For a k-clique hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ for $k \geq 6$, where $\mathcal{V} = \{X_1, X_2, \cdots, X_k\}$ and $\mathcal{E} = \{\{X_i, X_j\} : i, j \in [k], i \neq j\}$, ω-subw$(\mathcal{H}) = \frac{1}{2} \cdot \lceil\frac{k}{3}\rceil + \frac{1}{2} \cdot \lceil\frac{k-1}{3}\rceil + \frac{1}{2} \cdot \lfloor\frac{k}{3}\rfloor \cdot (\omega - 2)$.*

Proof. *Direction ω-subw$(\mathcal{H}) \leq \frac{1}{2} \cdot \lceil\frac{k}{3}\rceil + \frac{1}{2} \cdot \lceil\frac{k-1}{3}\rceil + \frac{1}{2} \cdot \lfloor\frac{k}{3}\rfloor \cdot (\omega - 2)$.* Let $i = \lfloor\frac{k}{3}\rfloor$ and $j = \lceil\frac{k-1}{3}\rceil$. Let $\boldsymbol{X} = \{X_1, \cdots, X_i\}$, $\boldsymbol{Y} = \{X_{i+1}, \cdots, X_{i+j}\}$ and $\boldsymbol{Z} = \{X_{i+j+1}, \cdots, X_k\}$. we consider a generalized elimination ordering $\overline{\boldsymbol{\sigma}} = (\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})$. Implied by Proposition C.2, $h(\boldsymbol{X}) \leq \frac{1}{2} \cdot \lfloor\frac{k}{3}\rfloor$, $h(\boldsymbol{Y}) \leq \frac{1}{2} \cdot \lceil\frac{k-1}{3}\rceil$, and $h(\boldsymbol{Z}) \leq \frac{1}{2} \cdot \lceil\frac{k}{3}\rceil$.

$$\mathsf{EMM}_1^{\overline{\sigma}} \leq \mathsf{MM}(\boldsymbol{Y}; \boldsymbol{Z}; \boldsymbol{X}) = h(\boldsymbol{Y}) + h(\boldsymbol{Z}) + \gamma h(\boldsymbol{X}) = \frac{1}{2} \cdot \lceil\frac{k}{3}\rceil + \frac{1}{2} \cdot \lceil\frac{k-1}{3}\rceil + \frac{1}{2} \cdot \lfloor\frac{k}{3}\rfloor \cdot (\omega - 2)$$

Direction ω-subw$(\mathcal{H}) \geq \frac{1}{2} \cdot \lceil\frac{k}{3}\rceil + \frac{1}{2} \cdot \lceil\frac{k-1}{3}\rceil + \frac{1}{2} \cdot \lfloor\frac{k}{3}\rfloor \cdot (\omega - 2)$. We identify a polymatroid $\boldsymbol{h}$ as follows: let $X_1, X_2, \cdots, X_k$ be independently random variables with $h(X_i) = \frac{1}{2}$ for $i \in [k]$. Then, we have $h(\mathcal{V}) = \frac{k}{2}$. From Proposition C.4, for an arbitrary generalized elimination ordering $\overline{\boldsymbol{\sigma}} \in \pi(\mathcal{V})$, $\mathsf{EMM}_1^{\overline{\sigma}} = \frac{1}{2} \cdot \lceil\frac{k}{3}\rceil + \frac{1}{2} \cdot \lceil\frac{k-1}{3}\rceil + \frac{1}{2} \cdot \lfloor\frac{k}{3}\rfloor \cdot (\omega - 2)$. From (39), we conclude ω-subw$(\mathcal{H}) \geq \frac{\omega}{2} + 1$. □

C.2 Cycle Hypergraphs

Lemma C.9. *For the following hypergraph representing a* 4*-cycle* $\mathcal{H}$*,* ω*-subw*$(\mathcal{H}) = 2 - \frac{3}{2\cdot\min\{\omega,\frac{5}{2}\}+1}$.

$$\mathcal{H} = (\{X, Y, Z, W\}, \quad \{\{X, Y\}, \{Y, Z\}, \{Z, W\}, \{W, X\}\}) \tag{41}$$

Proof. *Direction* ω*-subw*$(\mathcal{H}) \leq 2 - \frac{3}{2\cdot\min\{\omega,\frac{5}{2}\}+1}$. For simplicity, for vertex X, we denote $h(Y|X)$, $h(W|X)$ as the *neighboring conditional entropy* of X. Similar denotations apply to Y, Z, W. We partition all polymatroids $\boldsymbol{h} \in \Gamma \cap \mathsf{ED}$ into the following four cases:

- Case 1: each vertex has some neighboring conditional entropy larger than Δ. Wlog, assume vertex X has $h(Y|X) > \Delta$. As $h(XY) \leq 1$, we have $h(X) = h(XY) - h(Y|X) \leq 1 - \Delta$. Similarly, we can show $h(Y), h(Z), h(W) < 1 - \Delta$. Consider a generalized elimination ordering $\overline{\boldsymbol{\sigma}} = (\{X\}, \{Z\}, \{Y\}, \{W\})$. By choosing $\mathcal{A} = \{\{X, Y\}\}$ and $\mathcal{B} = \{\{X, W\}\}$, we have $\mathsf{EMM}_1^{\overline{\sigma}} \leq \mathsf{MM}(Y; W; X) \leq \omega(1 - \Delta)$. Meanwhile, $h\left(U_1^{\overline{\sigma}}\right) = h(XYW) \leq h(XY) + h(W) \leq 2 - \Delta$. By choosing $\mathcal{A} = \{\{Z, Y\}\}$ and $\mathcal{B} = \{\{Z, W\}\}$, we have $\mathsf{EMM}_2^{\overline{\sigma}} \leq \mathsf{MM}(Y; W; Z) \leq \omega(1 - \Delta)$.Meanwhile, $h\left(U_2^{\overline{\sigma}}\right) = h(YZW) \leq h(YZ) + h(W) \leq 2 - \Delta$. As $U_3^{\overline{\sigma}}, U_4^{\overline{\sigma}} \subseteq U_1^{\overline{\sigma}}$, we have $\max_{i\in[|\overline{\sigma}|]} \min\left(h(U_i^{\overline{\sigma}}), \mathsf{EMM}_i^{\overline{\sigma}}\right) \leq \min\{\omega(1 - \Delta), 2 - \Delta\}$.
- Case 2: one vertex has all neighboring conditional entropies smaller than Δ, and the other three vertices have some neighboring conditional entropies larger than Δ. Wlog, assume Z is the vertex with $h(Y|Z), h(W|Z) \leq \Delta$. Similar to Case 1, we can have $h(X), h(Y), h(W) < 1 - \Delta$. Consider a generalized elimination ordering $\overline{\boldsymbol{\sigma}} = (\{X\}, \{Z\}, \{Y\}, \{W\})$. By choosing $\mathcal{A} = \{\{X, Y\}\}$ and $\mathcal{B} = \{\{X, W\}\}$, we have $\mathsf{EMM}_1^{\overline{\sigma}} \leq \mathsf{MM}(Y; W; X) \leq \omega(1 - \Delta)$. Meanwhile, $h\left(U_1^{\overline{\sigma}}\right) = h(XYW) \leq h(XY) + h(W) \leq 2 - \Delta$. We have $h(U_2^{\overline{\sigma}}) = h(YZW) \leq h(Y|ZW) + h(ZW) \leq h(Y|Z) + h(ZW) \leq 1 + \Delta$. As $U_3^{\overline{\sigma}}, U_4^{\overline{\sigma}} \subseteq U_1^{\overline{\sigma}}$, we have:

$$\max_{i\in[|\overline{\sigma}|]} \min\left(h(U_i^{\overline{\sigma}}), \mathsf{EMM}_i^{\overline{\sigma}}\right) \leq \max\left\{\min\{\omega(1 - \Delta), 2 - \Delta\}, 1 + \Delta\right\}.$$

- Case 3: two non-tangent vertices have all their neighboring conditional entropies smaller than Δ. Suppose X, Z are two non-tangent vertices with all their neighboring conditional entropies smaller than Δ. Consider a generalized elimination ordering $\overline{\boldsymbol{\sigma}} = (\{X\}, \{Z\}, \{Y\}, \{W\})$. We have $h(U_1^{\overline{\sigma}}) = h(XYW) \leq h(W|XY) + h(XY) \leq h(W|X) + h(XY) \leq 1 + \Delta$. Similarly, $h(U_2^{\overline{\sigma}}) = h(ZYW) \leq h(W|ZY) + h(ZY) \leq h(W|Z) + h(ZY) \leq 1 + \Delta$. As $U_3^{\overline{\sigma}}, U_4^{\overline{\sigma}} \subseteq U_1^{\overline{\sigma}}$, we have $\max_{i\in[|\overline{\sigma}|]} \min\left(h(U_i^{\overline{\sigma}}), \mathsf{EMM}_i^{\overline{\sigma}}\right) \leq 1 + \Delta$.
- Case 4: two tangent vertices have all their neighboring conditional entropies smaller than Δ, and the other two have some neighboring conditional entropy larger than Δ. Wlog, assume Z, W are two tangent vertices with all their neighboring conditional entropies smaller than Δ. Similar as above, we have $h(X), h(Y) \leq 1 - \Delta$. We have two different observations:
  - Consider a generalized elimination ordering $\overline{\boldsymbol{\sigma}} = (\{Y\}, \{W\}, \{X\}, \{Z\})$. We have $h(U_1^{\overline{\sigma}}) = h(XYZ) \leq h(X) + h(YZ) \leq 2 - \Delta$ and $h(U_2^{\overline{\sigma}}) = h(XZW) \leq h(X) + h(ZW) \leq 2 - \Delta$. As $U_3^{\overline{\sigma}}, U_4^{\overline{\sigma}} \subseteq U_1^{\overline{\sigma}}$, we have $\max_{i\in[|\overline{\sigma}|]} \min\left(h(U_i^{\overline{\sigma}}), \mathsf{EMM}_i^{\overline{\sigma}}\right) \leq 2 - \Delta$.
  - We can also further distinguish two more cases:
    * If $h(Y|Z) \leq \frac{\Delta}{2}$ and $h(X|W) \leq \frac{\Delta}{2}$, we have $h(XYZW) \leq h(X|YZW) + h(Y|ZW) + h(ZW) \leq h(X|W) + h(Y|Z) + h(ZW) \leq 1 + \Delta$.

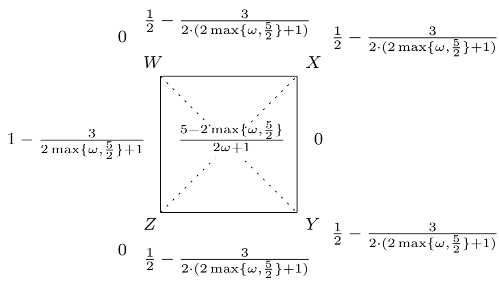

Fig. 3. Diagram of the polymatroid for cycle-4 hypergraph.

- $*$ Otherwise, either $h(Y|Z) > \frac{\Delta}{2}$ or $h(X|W) > \frac{\Delta}{2}$. Wlog, assume $h(Y|Z) > \frac{\Delta}{2}$. Similar as above, we have $h(Z) \geq 1 - \frac{\Delta}{2}$. Consider a generalized elimination ordering $\overline{\sigma} = (\{Y\}, \{W\}, \{Z\}, \{X\})$. By choosing $\mathcal{A} = \{\{X, Y\}\}$ and $\mathcal{B} = \{\{Y, Z\}\}$, we have $\mathsf{EMM}_1^{\overline{\sigma}} \leq \mathsf{MM}(X; Z; Y|\emptyset) = (\omega - 1)(1 - \Delta) + (1 - \frac{\Delta}{2})$ and $h(U_1^{\overline{\sigma}}) = h(XYZ) \leq h(X) + h(YZ) \leq 2 - \Delta$. Moreover, $h(U_2^{\overline{\sigma}}) = h(ZWX) \leq h(Z|WX) + h(WX) \leq h(Z|W) + h(WX) \leq 1 + \Delta$. As $U_3^{\overline{\sigma}}, U_4^{\overline{\sigma}} \subseteq U_1^{\overline{\sigma}}$, we have:

$$\max_{i \in [|\overline{\sigma}|]} \min\left(h(U_i^{\overline{\sigma}}), \mathsf{EMM}_i^{\overline{\sigma}}\right) \leq \max\left\{1 + \Delta, \min\left\{(\omega - 1)(1 - \Delta) + \left(1 - \frac{\Delta}{2}\right), 2 - \Delta\right\}\right\}.$$

Putting all cases together, we obtain the upper bound:

$$\max\left\{1 + \Delta, \min\left\{(\omega - 1)(1 - \Delta) + 1 - \frac{\Delta}{2}, 2 - \Delta\right\}\right\}$$

If $\frac{5}{2} \leq \omega \leq 3$, we set $\Delta = \frac{1}{2}$ to obtain the upper bound as $\frac{3}{2}$. If $2 \leq \omega \leq \frac{5}{2}$, we set $\Delta = \frac{2(\omega-1)}{2\omega+1}$ to obtain the upper bound as $\frac{4\omega-1}{2\omega+1}$. Putting two cases together, we obtain the desired upper bound.

<u>*Direction ω-subw$(\mathcal{H}) \geq 2 - \frac{3}{2\cdot\min\{\omega,\frac{5}{2}\}+1}$*</u>. Correspondingly, we identify a polymatroid $\boldsymbol{h} \in \Gamma \cap \mathsf{ED}$ by distinguishing the following two cases:

Case 1: $\frac{5}{2} \leq \omega \leq 3$. Let a, b, c, d, e be independently random variables with $h(a) = h(b) = h(c) = h(d) = \frac{1}{4}$ and $h(e) = \frac{1}{2}$. Let $h(X) = (ab)$, $h(Y) = (cd)$, $h(Z) = (de)$ and $h(W) = (ae)$. We have

- $h(X) = h(Y) = \frac{1}{2}$, $h(Z) = h(W) = \frac{3}{4}$;
- $h(XY) = h(YZ) = h(ZW) = h(WX) = 1$ and $h(XZ) = h(YW) = \frac{5}{4}$;
- $h(XZW) = h(YZW) = \frac{5}{4}$ and $h(XYZ) = h(XYW) = \frac{3}{2}$;
- $h(XYZW) = \frac{3}{2}$;

Consider any generalized elimination ordering $\overline{\sigma} \in \pi(\mathcal{V})$. We distinguish the following cases:

- $\overline{\sigma}[1] = \{X\}$. $h(U_1^{\overline{\sigma}}) = h(XYW) = \frac{3}{2}$ and $\mathsf{EMM}_1^{\overline{\sigma}} = \mathsf{MM}(Y, W, X) = \frac{\omega}{2} + \frac{1}{4} \geq \frac{3}{2}$ when $\omega \geq \frac{5}{2}$. The case with $\overline{\sigma}[1] = \{W\}$ is the same.
- $\overline{\sigma}[1] = \{W\}$. $h(U_1^{\overline{\sigma}}) = h(XZW) = \frac{5}{4}$ and $\mathsf{EMM}_1^{\overline{\sigma}} = \mathsf{MM}(X, Z, W) = \frac{\omega}{2} + \frac{1}{4} \geq \frac{3}{2}$ when $\omega \geq \frac{5}{2}$. The case with $\overline{\sigma}[1] = \{Z\}$ is the same.
- $|\sigma[1]| = 2$. $h(U_1^{\overline{\sigma}}) = h(\mathcal{V}) = \frac{3}{2}$ and $\mathsf{EMM}_1^{\overline{\sigma}} \geq \mathsf{MM}(X, W, Z) = \frac{\omega}{2} + \frac{1}{4} \geq \frac{3}{2}$.

Case 2: $\omega < \frac{5}{2}$. Let a, b, c, d, e, f be independently random variables with $h(a) = \frac{2(\omega-1)}{2\omega+1}$, $h(b) = h(c) = h(d) = h(e) = \frac{\omega-1}{2\omega+1}$ and $h(f) = \frac{5-2\omega}{2\omega+1}$. Let $X = (bcf)$, $Y = (def)$, $Z = (aef)$, $W = (abf)$. We have

- $h(W) = h(Z) = \frac{\omega+2}{2\omega+1}$ and $h(X) = h(Y) = \frac{3}{2\omega+1}$;

- $h(WX) = h(XY) = h(YZ) = h(ZW) = 1$, $h(WY) = h(XZ) = \frac{3\omega}{2\omega+1}$;
- $h(WXY) = h(XYZ) = \frac{4\omega-1}{2\omega+1}$, $h(XZW) = h(YZW) = \frac{3\omega}{2\omega+1}$;
- $h(XYZW) = \frac{4\omega-1}{2\omega+1}$;

Consider any generalized elimination ordering $\overline{\sigma} \in \pi(\mathcal{V})$. We distinguish the following cases:

- $\overline{\sigma}[1] = \{X\}$. $h(U_1^{\overline{\sigma}}) = h(XYW) = \frac{4\omega-1}{2\omega+1}$ and $\mathsf{EMM}_1^{\overline{\sigma}} = \mathsf{MM}(Y, W, X) = \frac{\omega+2}{2\omega+1} + \frac{3(\omega-1)}{2\omega+1} = \frac{4\omega-1}{2\omega+1}$. The case with $\overline{\sigma}[1] = \{Y\}$ is the same.
- $\overline{\sigma}[1] = \{W\}$. $h(U_1^{\overline{\sigma}}) = h(XZW) = \frac{3\omega}{2\omega+1}$ and $\mathsf{EMM}_1^{\overline{\sigma}} = \mathsf{MM}(X, Z, W) = \frac{2(\omega+2)}{2\omega+1} + \frac{3(\omega-2)}{2\omega+1} = \frac{5\omega-2}{2\omega+1} > \frac{4\omega-1}{2\omega+1}$. The case with $\overline{\sigma}[1] = \{Z\}$ is the same.
- $|\overline{\sigma}[1]| = 2$. $h(U_1^{\overline{\sigma}}) = h(\mathcal{V}) = \frac{4\omega-1}{2\omega+1}$ and $\mathsf{EMM}_1^{\overline{\sigma}} \geq \mathsf{MM}(Y, W, X) = \frac{4\omega-1}{2\omega+1}$.

□

[10] defines an exponent $\overline{c}_k$ for detecting a k-cycle in a graph, and the definition is based on *rectangular* matrix multiplication. We define below an upper bound, $c_k^\square$, on $\overline{c}_k$ that uses only *square* matrix multiplication and show that ω-subw$(\mathcal{H}) \leq c_k^\square$ for any k-cycle graph $\mathcal{H}$. In particular, $\overline{c}_k \leq c_k^\square$ and this becomes an equality when $\omega = 2$. Moreover, [10] shows that $\overline{c}_k = c_k^\square$ when k is odd as well as $k = 4$ or 6. More details can be found in [10].

First, we recall the definition of $\overline{c}_k$ from [10]. Recall from Table 1 that $\overline{\omega}(a, b, c)$ is the smallest exponent for multiplying two rectangular matrices of sizes $n^a \cdot n^b$ and $n^b \cdot n^c$ within $O\left(n^{\overline{\omega}(a,b,c)}\right)$ time, whereas $\omega^\square(a, b, c)$ is the smallest upper bound on $\overline{\omega}(a, b, c)$ that is obtained through *square* matrix multiplication, i.e. $\omega^\square(a, b, c) \stackrel{\text{def}}{=} \max\{a + b + (\omega - 2)c, a + (\omega - 2)b + c, (\omega - 2)a + b + c\}$. In particular, $\overline{\omega}(a, b, c) \leq \omega^\square(a, b, c)$, and this becomes an equality when $\omega = 2$ or when $a = b = c$. Given a graph of size N, let $\mathbf{D} = \{0, \log_N 2, 2\log_N 2, \cdots, 1\}^{2k}$. For each vector $\vec{d}$, [10] defines a function $\overline{P}_{i,j}^{\vec{d}}$ (for $i < j$) in a recursive way as follows:

$$\overline{P}_{i,j}^{\vec{d}} = \min\left\{\overline{P}_{i,j-1}^{\vec{d}} + d_{j-1}^+, \overline{P}_{i+1,j}^{\vec{d}} + d_{i+1}^-, \min_{i<r<j}\left\{\overline{P}_{i,r}^{\vec{d}}, \overline{P}_{r,j}^{\vec{d}}, \overline{\omega}(1 - d_i, 1 - d_r, 1 - d_j)\right\}\right\} \tag{42}$$

with $\overline{P}_{i,i+1}^{\vec{d}} = 1$. Finally, [10] defines $\overline{c}_k$ as follow:

$$\overline{c}_k = \max_{\vec{d}=\left(d_1^-, d_1^+, d_2^-, d_2^+, \cdots, d_k^-, d_k^+\right) \in \mathbf{D}} \min\left\{\min_{i \in [k]} 2 - d_i, \min_{i,j \in [k]: i<j} \max\left\{\overline{P}_{i,j}^{\vec{d}}, \overline{P}_{j,i}^{\vec{d}}\right\}\right\} \tag{43}$$

In contrast, by replacing $\overline{\omega}(1 - d_i, 1 - d_r, 1 - d_j)$ with $\omega^\square(1 - d_i, 1 - d_r, 1 - d_j)$ in Eq. (42), we obtain our variants $P_{i,j}^{\vec{d}}$ and $c_k^\square$ defined below (with $P_{i,i+1}^{\vec{d}} = 1$):

$$P_{i,j}^{\vec{d}} = \min\left\{P_{i,j-1}^{\vec{d}} + d_{j-1}^+, P_{i+1,j}^{\vec{d}} + d_{i+1}^-, \min_{i<r<j}\left\{P_{i,r}^{\vec{d}}, P_{r,j}^{\vec{d}}, \omega^\square(1 - d_i, 1 - d_r, 1 - d_j)\right\}\right\} \tag{44}$$

$$c_k^\square = \max_{\vec{d}=\left(d_1^-, d_1^+, d_2^-, d_2^+, \cdots, d_k^-, d_k^+\right) \in \mathbf{D}} \min\left\{\min_{i \in [k]} 2 - d_i, \min_{i,j \in [k]: i<j} \max\left\{P_{i,j}^{\vec{d}}, P_{j,i}^{\vec{d}}\right\}\right\} \tag{45}$$

Lemma C.10. *For the following hypergraph representing a k-cycle:*

$$\mathcal{H} = (\{X_1, X_2, \cdots, X_k\}, \quad \{\{X_1, X_2\}, \{X_2, X_3\}, \cdots, \{X_{k-1}, X_k\}, \{X_k, X_1\}\}) \tag{46}$$

ω-subw$(\mathcal{H}) \leq c_k^\square$ as defined in Eq. (45).

Proof of Lemma C.10. We partition all polymatroids $\boldsymbol{h}$ into the following $(1 + \log N)^{|\mathcal{V}|}$ cases. For simplicity, let $X_1 = X_{k+1}$. Each polymatroid $\boldsymbol{h} \in \Gamma \cap \mathsf{ED}$ is associated with a k-tuple vector

$\vec{d} = (d_1^-, d_1^+, d_2^-, d_2^+, \cdots, d_k^-, d_k^+) \in \mathbf{D}$ such that $d_i^- \le h(X_{i-1}|X_i) < d_i^- + \log_N 2$ and $d_i^+ \le h(X_{i+1}|X_i) < d_i^+ + \log_N 2$. We have the following two observations:

- For each $i \in [k]$, consider a generalized elimination ordering $\overline{\sigma} = (\{X_{i-1}\}, \cdots, \{X_1\}, \{X_k\}, \cdots, \{X_i\})$. As $h(X_{i-1}|X_i) \ge d_i^-$ and $h(X_{i-1}X_i) \le 1$, we have $h(X_i) = h(X_{i-1}X_i) - h(X_{i-1}|X_i) < 1 - d_i^-$. Similarly, we have $h(X_i) < 1 - d_i^+$. For simplicity, we define $d_i = \max\{d_i^-, d_i^+\}$. For each $i \in [k]$, we have $h\left(X_i X_j X_{j+1}\right) \le h(X_i) + h(X_j X_{j+1}) < \min\{1 - d_i^-, 1 - d_i^+\} + 1 = 2 - d_i$. So, $h\left(U_j^{\overline{\sigma}}\right) \le 2 - d_i$ for any $j \in [|\overline{\sigma}|]$, i.e., $\max_{j \in [|\overline{\sigma}|]} h\left(U_j^{\overline{\sigma}}\right) \le 2 - d_i$.
- For each pair of distinct values $i, j \in [k]$ (wlog assume $i < j$), we define $\overline{\sigma}^{\vec{d}}_{i,j}$ as an ordering of $\{X_{i+1}, X_{i+2}, \cdots, X_{j-1}\}$ as follows:
$$\overline{\sigma}^{\vec{d}}_{i,j} = \begin{cases} \left(\overline{\sigma}^{\vec{d}}_{i,j-1},\ X_{j-1}\right) & \text{if } P^{\vec{d}}_{i,j} = P^{\vec{d}}_{i,j-1} + d^+_{j-1}, \\ \left(\overline{\sigma}^{\vec{d}}_{i+1,j},\ X_{i+1}\right) & \text{if } P^{\vec{d}}_{i,j} = P^{\vec{d}}_{i+1,j} + d^-_{i+1}, \\ \left(\overline{\sigma}^{\vec{d}}_{i,r},\ \overline{\sigma}^{\vec{d}}_{r,j},\ X_r\right) & \text{if } P^{\vec{d}}_{i,j} = \min\left\{P^{\vec{d}}_{i,r}, P^{\vec{d}} f_{r,j}, \omega^{\square}(1-d_i, 1-d_r, 1-d_j)\right\}, \end{cases}$$
and $\overline{\sigma}^{\vec{d}}_{i,i+1} = \emptyset$. At last, we construct a generalized elimination order $\overline{\sigma}^{\vec{d}} = \left(\overline{\sigma}^{\vec{d}}_{i,j}, \overline{\sigma}^{\vec{d}}_{j,i}, X_i, X_j\right)$. It can be proved that $\max_{\ell \in \left[\left|\overline{\sigma}^{\vec{d}}_{i,j}\right|\right]} \min\left\{h\left(U_\ell^{\overline{\sigma}^{\vec{d}}_{i,j}}\right), \mathsf{EMM}_\ell^{\overline{\sigma}^{\vec{d}}_{i,j}}\right\} \le P^{\vec{d}}_{i,j}$. In the base case when $j = i+1$, $h(\emptyset) = 0 \le 1$ trivially holds. In general, we distinguish three cases:
  - If $\overline{\sigma}^{\vec{d}}_{i,j} = \left(\overline{\sigma}^{\vec{d}}_{i,j-1},\ X_{j-1}\right)$: by hypothesis, assume this claim holds for $\overline{\sigma}^{\vec{d}}_{i,j-1}$, i.e.,
  $$\max_{\ell \in \left[\left|\overline{\sigma}^{\vec{d}}_{i,j-1}\right|\right]} \min\left\{h\left(U_\ell^{\overline{\sigma}^{\vec{d}}_{i,j-1}}\right), \mathsf{EMM}_\ell^{\overline{\sigma}^{\vec{d}}_{i,j-1}}\right\} \le P^{\vec{d}}_{i,j-1}$$
  For any $\ell \in \left[\left|\overline{\sigma}^{\vec{d}}_{i,j}\right| - 1\right]$, $U_\ell^{\overline{\sigma}^{\vec{d}}_{i,j}} = U_\ell^{\overline{\sigma}^{\vec{d}}_{i,j-1}}$, and $\mathsf{EMM}_\ell^{\overline{\sigma}^{\vec{d}}_{i,j-1}} = \mathsf{EMM}_\ell^{\overline{\sigma}^{\vec{d}}_{i,j}}$. For $\ell = \left|\overline{\sigma}^{\vec{d}}_{i,j}\right|$, $h(U_\ell^{\overline{\sigma}^{\vec{d}}_{i,j}}) = h(X_i X_{j-1} X_j) \le h(X_i X_{j-1}) + h(X_j|X_{j_1}) \le P^{\vec{d}}_{i,j-1} + d^+_{j-1}$. Together, we have:
  $$\max_{\ell \in \left[\left|\overline{\sigma}^{\vec{d}}_{i,j}\right|\right]} \min\left\{h\left(U_\ell^{\overline{\sigma}^{\vec{d}}_{i,j}}\right), \mathsf{EMM}_\ell^{\overline{\sigma}^{\vec{d}}_{i,j}}\right\} \le P^{\vec{d}}_{i,j-1} + d^+_{j-1}$$
  - If $\overline{\sigma}^{\vec{d}}_{i,j} = \left(\overline{\sigma}^{\vec{d}}_{i+1,j},\ X_{j-1}\right)$: the case is similar as above.
  - If $\overline{\sigma}^{\vec{d}}_{i,j} = \left(\overline{\sigma}^{\vec{d}}_{i,r},\ \overline{\sigma}^{\vec{d}}_{r,j},\ X_r\right)$: by hypothesis, assume this claim holds for $\overline{\sigma}^{\vec{d}}_{i,r}$ and $\overline{\sigma}^{\vec{d}}_{r,j}$, i.e.,
  $$\max_{\ell \in \left[\left|\overline{\sigma}^{\vec{d}}_{i,r}\right|\right]} \min\left\{h\left(U_\ell^{\overline{\sigma}^{\vec{d}}_{i,r}}\right), \mathsf{EMM}_\ell^{\overline{\sigma}^{\vec{d}}_{i,r}}\right\} \le P^{\vec{d}}_{i,r}$$
  $$\max_{\ell \in \left[\left|\overline{\sigma}^{\vec{d}}_{r,j}\right|\right]} \min\left\{h\left(U_\ell^{\overline{\sigma}^{\vec{d}}_{r,j}}\right), \mathsf{EMM}_\ell^{\overline{\sigma}^{\vec{d}}_{r,j}}\right\} \le P^{\vec{d}}_{r,j}$$
  For any $\ell \in \left[\left|\overline{\sigma}^{\vec{d}}_{i,r}\right|\right]$, $U_\ell^{\overline{\sigma}^{\vec{d}}_{i,r}} = U_\ell^{\overline{\sigma}^{\vec{d}}_{i,j}}$, and $\mathsf{EMM}_\ell^{\overline{\sigma}^{\vec{d}}_{i,r}} = \mathsf{EMM}_\ell^{\overline{\sigma}^{\vec{d}}_{i,j}}$. For any $\ell \in \left[\left|\overline{\sigma}^{\vec{d}}_{r,j}\right|\right]$, $U_\ell^{\overline{\sigma}^{\vec{d}}_{r,j}} = U_{\ell+\left|\overline{\sigma}^{\vec{d}}_{i,r}\right|}^{\overline{\sigma}^{\vec{d}}_{i,j}}$, and $\mathsf{EMM}_\ell^{\overline{\sigma}^{\vec{d}}_{r,j}} = \mathsf{EMM}_{\ell+\left|\overline{\sigma}^{\vec{d}}_{i,r}\right|}^{\overline{\sigma}^{\vec{d}}_{i,j}}$. For $\ell = \left|\overline{\sigma}^{\vec{d}}_{i,j}\right|$, $\mathsf{EMM}_\ell^{\overline{\sigma}^{\vec{d}}_{i,j}} = \mathsf{MM}(X_i, X_r, X_j) \le \omega^{\square}(1 -$

$d_i, 1-d_r, 1-d_j)$, since $h(X_i) \leq 1-d_i$, $h(X_r) \leq 1-d_r$ and $h(X_j) \leq 1-d_j$. Together, we have:

$$\max_{\ell \in \left[\left|\overline{\sigma}^{\vec{d}}_{i,j}\right|\right]} \min\left\{h\left(U_\ell^{\overline{\sigma}^{\vec{d}}_{i,j}}\right), \mathsf{EMM}_\ell^{\overline{\sigma}^{\vec{d}}_{i,j}}\right\} \leq \min\left\{P^{\vec{d}}_{i,r}, P^{\vec{d}}_{r,j}, \omega^{\square}(1-d_i, 1-d_r, 1-d_j)\right\}$$

We define a generalized elimination ordering $\overline{\sigma}^{\vec{d}} = \left(\overline{\sigma}^{\vec{d}}_{i,j}, \overline{\sigma}^{\vec{d}}_{j,i}, X_i, X_j\right)$. Hence,

$$\max_{\ell \in \left[\left|\overline{\sigma}^{\vec{d}}\right|\right]} \min\left\{h\left(U_\ell^{\overline{\sigma}^{\vec{d}}}\right), \mathsf{EMM}_\ell^{\overline{\sigma}^{\vec{d}}}\right\} \leq \max\left\{P^{\vec{d}}_{i,j}, P^{\vec{d}}_{j,i}\right\}.$$

Combining these two cases and applying these arguments to all possible $\vec{d}$, we have

$$\omega\text{-subw}(\mathcal{H}) \leq \min_{\vec{d}=(d_1^-,d_1^+,d_2^-,d_2^+,\cdots,d_k^-,d_k^+)} \min\left\{\min_{i\in[k]} 2-d_i, \min_{i,j\in[k]:i<j} \max\left\{P^{\vec{d}}_{i,j}, P^{\vec{d}}_{j,i}\right\}\right\} = c_k^{\square}.$$

□

C.3 Clustered Hypergraphs

Definition C.11 (Clustered Hypergraph). A hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ is clustered if for any pair of vertices $X_i, X_j \in \mathcal{V}$, there exists some hypergraph $e \in \mathcal{E}$ such that $X_i, X_j \in e$.

Lemma C.12. *A clustered hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ satisfies the following properties:*

- *$subw(\mathcal{H}) = \rho^*(\mathcal{H})$.*
- *for any generalized elimination ordering $\overline{\sigma} \in \pi(\mathcal{V})$, $U_1^{\overline{\sigma}} = \mathcal{V}$; and therefore $U_i^{\overline{\sigma}} \subseteq U_1^{\overline{\sigma}}$ for any $i \in [|\overline{\sigma}|]$.*

Lemma C.13. *For the following hypergraph representing a 3-pyramid:*

$$\mathcal{H} = (\{Y, X_1, X_2, X_3\}, \quad \{\{Y, X_1\}, \{Y, X_2\}, \{Y, X_3\}, \{X_1, X_2, X_3\}\}) \tag{47}$$

ω-subw$(\mathcal{H}) = 2 - \frac{1}{\omega}$.

Proof of Lemma C.13. Direction ω-subw$(\mathcal{H}) \leq 2 - \frac{1}{\omega}$. We partition all polymatroids $\boldsymbol{h}$ into the following cases:

Case 1: there exists some $i \in [3]$ such that $h(Y|X_i) \leq \Delta$. In this case, $h(\mathcal{V}) \leq h(Y|X_1X_2X_3) + h(X_1X_2X_3) \leq h(Y|X_i) + h(X_1X_2X_3) \leq \Delta + 1$;

Case 2: $h(X_i|Y) \leq \frac{\Delta}{2}$ for each $i \in [3]$. In this case, $h(\mathcal{V}) \leq h(X_1|Y) + h(X_2|Y) + h(X_3Y) \leq \Delta + 1$.

Case 3: none of the cases above, i.e., $h(Y|X_i) > \Delta$ for each $i \in [3]$. As $h(X_iY) \leq 1$ and $h(Y|X_i) > \Delta$, we have $h(X_i) = h(X_iY) - h(Y|X_i) \leq 1 - \Delta$. As $h(X_iY) \leq 1$ and $h(X_i|Y) > \frac{\Delta}{k-1}$, we have $h(Y) = h(X_iY) - h(X_i|Y) \leq 1 - \frac{\Delta}{2}$. Let $\boldsymbol{A} = \{X_2\}$ and $\boldsymbol{B} = \{X_3\}$. Note that $h(\boldsymbol{A}) \leq (1-\Delta)$ and $h(\boldsymbol{B}) \leq (1-\Delta)$. We assume that Δ is a parameter such that $\Delta \geq (1-\Delta)$ holds, i.e., $\Delta \geq \frac{1}{2}$. Consider a generalized elimination ordering $\overline{\sigma} = (\{Y\}, \boldsymbol{A}, \boldsymbol{B}, \{X_1\})$. We have

$$\mathsf{EMM}_1^{\overline{\sigma}} \leq \mathsf{MM}(\boldsymbol{A}; \boldsymbol{B}; Y|X_1) \leq \gamma h(\boldsymbol{A}) + h(\boldsymbol{B}) + h(X_1Y) \leq \gamma(1-\Delta) + (1-\Delta) + 1$$

Combining all the cases, we obtain the desired upper bound $1 + \max\{\Delta, \gamma(1-\Delta) + (1-\Delta)\} = 2 - \dfrac{1}{\omega}$, by setting $\Delta = 1 - \frac{1}{\omega}$ (note that $\Delta \geq \frac{1}{2}$).

Direction ω-subw$(\mathcal{H}) \geq 2 - \frac{1}{\omega}$. We identify the following polymatroid $\boldsymbol{h} \in \Gamma \cap \mathsf{ED}$:

- $h(X_1) = h(X_2) = h(X_3) = \frac{1}{\omega}$;
- $h(Y) = 1 - \frac{1}{\omega}$;
- $h(X_1X_2) = h(X_1X_3) = h(X_2X_3) = \frac{2}{\omega}$;

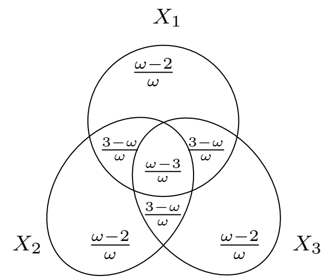

Fig. 4. Diagram of the polymatroid for 3-Pyramid hypergraph.

- $h(X_1Y) = h(X_2Y) = h(X_3Y) = 1$;
- $h(X_1X_2X_3) = 1$;
- $h(X_1X_2Y) = h(X_1X_3Y) = h(X_2X_3Y) = \frac{\omega+1}{\omega}$;
- $h(X_1X_2X_3Y) = 2 - \frac{1}{\omega}$;

  We distinguish an arbitrary generalized elimination ordering $\overline{\sigma}$ into the following cases:
- $\overline{\sigma}[1] = \{Y\}$. $h(U_1^{\overline{\sigma}}) = h(\mathcal{V}) = 2 - \frac{1}{\omega}$. $\mathsf{EMM}_1^{\overline{\sigma}} = \min\{\mathsf{MM}(X_1, X_2, Y; X_3), \mathsf{MM}(X_1, X_2X_3, Y)\} = 2 - \frac{1}{\omega}$;
- $\overline{\sigma}[1] = \{X_1\}$. $h(U_1^{\overline{\sigma}}) = h(\mathcal{V}) = 2 - \frac{1}{\omega}$. $\mathsf{EMM}_1^{\overline{\sigma}} = \min\{\mathsf{MM}(X_2, Y, X_1|X_3),\ \mathsf{MM}(X_2X_3, Y, X_1),\ \mathsf{MM}(X_2, X_3Y, X_1)\} =$

  The case with $\overline{\sigma}[1] = \{X_2\}$ or $\overline{\sigma}[1] = \{X_3\}$ is the same.
- $\overline{\sigma}[1] = \{X_1, Y\}$. $h(U_1^{\overline{\sigma}}) = h(\mathcal{V}) = 2 - \frac{1}{\omega}$. $\mathsf{EMM}_1^{\overline{\sigma}} = \mathsf{MM}(X_2, X_3, X_1Y) = 2 - \frac{1}{\omega}$. The case with $\overline{\sigma}[1] = \{X_2, Y\}$ or $\overline{\sigma}[1] = \{X_3, Y\}$ is the same.
- $\overline{\sigma}[1] = \{X_1, X_2\}$. $h(U_1^{\overline{\sigma}}) = h(\mathcal{V}) = 2 - \frac{1}{\omega}$. $\mathsf{EMM}_1^{\overline{\sigma}} = \mathsf{MM}(X_3, Y, X_1X_2) = 2 - \frac{1}{\omega}$. The case with $\overline{\sigma}[1] = \{X_1, X_3\}$ or $\overline{\sigma}[1] = \{X_2, X_3\}$ is the same.

Hence, for such $\boldsymbol{h}$, we have $\max_{i \in [|\overline{\sigma}|]} \min\left(h\left(U_i^{\overline{\sigma}}\right), \mathsf{EMM}_i^{\overline{\sigma}}\right) \geq 2 - \dfrac{1}{\omega}$. □

Lemma C.14. *For the following hypergraph representing a k-pyramid:*

$$\mathcal{H} = (\{Y, X_1, X_2, \cdots, X_k\},\ \{\{Y, X_1\}, \{Y, X_2\}, \cdots, \{Y, X_k\}, \{X_1, X_2, \cdots, X_k\}\}) \tag{48}$$

for $k \geq 3$, ω-subw($\mathcal{H}$) $\leq 2 - \dfrac{2}{\omega \cdot (k-1) - k + 3}$.

Proof of Lemma C.14. We partition all polymatroids $\boldsymbol{h}$ into the following two cases:

- Case 1: there exists some $i \in [k]$ such that $h(Y|X_i) \leq \Delta$. In this case, $h(\mathcal{V}) \leq h(Y|X_1X_2\cdots X_k) + h(X_1X_2\cdots X_k) \leq h(Y|X_i) + h(X_1X_2\cdots X_k) \leq \Delta + 1$;
- Case 2: $h(X_i|Y) \leq \frac{\Delta}{k-1}$ for each $i \in [k]$. In this case, $h(\mathcal{V}) \leq \sum_{i \in [k-1]} h(X_1|Y) + h(X_kY) \leq \Delta + 1$.
- Case 3: none of the cases above, i.e., $h(Y|X_i) > \Delta$ for each $i \in [k]$. As $h(X_iY) \leq 1$ and $h(Y|X_i) > \Delta$, we have $h(X_i) = h(X_iY) - h(Y|X_i) \leq 1 - \Delta$. As $h(X_iY) \leq 1$ and $h(X_i|Y) > \frac{\Delta}{k-1}$, we have $h(Y) = h(X_iY) - h(X_i|Y) \leq 1 - \frac{\Delta}{k-1}$.

  We further distinguish two more cases depending on whether k is even or odd.
  - **When k is odd.** Let $\boldsymbol{A} = \left\{X_2, X_3, \cdots, X_{\frac{k+1}{2}}\right\}$ and $\boldsymbol{B} = \left\{X_{\frac{k+1}{2}+1}, X_{\frac{k+1}{2}+2}, \cdots, X_k\right\}$. Note that $h(\boldsymbol{A}) \leq (1-\Delta) \cdot \frac{k-1}{2}$ and $h(\boldsymbol{B}) \leq (1-\Delta) \cdot \frac{k-1}{2}$. In this case, we assume that $\Delta > (1-\Delta) \cdot \frac{k-1}{2}$. Consider a generalized elimination ordering $\overline{\sigma} = (\{Y\}, \boldsymbol{A}, \boldsymbol{B}, \{X_1\})$. We have

$$\mathsf{EMM}_1^{\overline{\sigma}} \leq \mathsf{MM}(\boldsymbol{A}; \boldsymbol{B}; Y|X_1) \leq \gamma h(\boldsymbol{A}) + h(\boldsymbol{B}) + h(X_1Y) \leq (\gamma+1)(1-\Delta) \cdot \frac{k-1}{2} + 1$$

Combining all the cases, we obtain the desired upper bound:

$$1 + \max\left\{\Delta, (\gamma+1)(1-\Delta)\cdot\frac{k-1}{2}\right\} = 2 - \frac{2}{\omega\cdot(k-1)-k+3}.$$

- **When k is even.** We partition all polymatroids $\boldsymbol{h}$ into the following two cases: Let $\boldsymbol{A} = \left\{X_1, X_2, X_3, \cdots, X_{\frac{k}{2}}\right\}$ and $\boldsymbol{B} = \left\{X_{\frac{k}{2}+1}, X_{\frac{k}{2}+1}, \cdots, X_k\right\}$. Note that $h(\boldsymbol{A}) \le (1-\Delta)\cdot\frac{k}{2}$ and $h(\boldsymbol{B}) \le (1-\Delta)\cdot\frac{k}{2}$. We assume that Δ is a parameter such that $1-\frac{\Delta}{k-1} > (1-\Delta)\cdot\frac{k}{2}$. Consider a generalized elimination ordering $\overline{\boldsymbol{\sigma}} = (\{Y\}, \boldsymbol{A}, \boldsymbol{B})$. We have

$$\mathsf{EMM}_1^{\overline{\sigma}} \le \mathsf{MM}(\boldsymbol{A};\boldsymbol{B};Y) \le \gamma h(\boldsymbol{A}) + h(\boldsymbol{B}) + h(Y) \le \gamma(1-\Delta)\cdot\frac{k}{2} + (1-\Delta)\cdot\frac{k}{2} + 1 - \frac{\Delta}{k-1}$$

Combining all the cases, we obtain the desired upper bound:

$$1 + \max\left\{\Delta, \gamma(1-\Delta)\cdot\frac{k}{2} + (1-\Delta)\cdot\frac{k}{2} - \frac{\Delta}{k-1}\right\} \ge 2 - \frac{2}{\omega\cdot(k-1)-k+3}$$

□

Lemma C.15. *For the following hypergraph*

$$\mathcal{H} = (\{X,Y,Z,W,L\}, \quad \{\{X,Y,W\},\{X,Y,L\},\{X,Z\},\{Y,Z\},\{Z,W,L\}\})$$

ω-subw$(\mathcal{H}) \le 2 - \frac{1}{2(\omega-2)+3}$.

Remark. For the hypergraph $\mathcal{H}$ in Lemma C.15, $\mathsf{subw}(\mathcal{H}) = \frac{9}{5}$. If $\omega = 3$, ω-$\mathsf{subw}(\mathcal{H}) \le 2 - \frac{1}{2\omega-1} = \mathsf{subw}(\mathcal{H}) = \frac{9}{5}$. If $\omega < 3$, ω-$\mathsf{subw}(\mathcal{H}) < \mathsf{subw}(\mathcal{H})$.

Proof. Let $\Delta = \frac{2(\omega-1)}{2\omega-1}$. We partition all polymatroids $\boldsymbol{h}$ into the following cases:

- Case 1: $h(XY|W) \le \Delta$ or $h(ZL|W) \le \Delta$. wlog, suppose $h(XY|W) \le \Delta$. We have $h(XYZWL) \le h(XY|ZWL) + h(ZWL) \le h(XY|W) + h(ZWL) \le 1+\Delta$.
- Case 2: $h(XY|L) \le \Delta$ or $h(ZW|L) \le \Delta$. This case is similar to Case 1.
- Case 3: $h(X|Z) \le \frac{\Delta}{2}$ and $h(Y|Z) \le \frac{\Delta}{2}$. We have $h(XYZWL) \le h(X|ZWL) + h(Y|ZWL) + h(ZWL) \le h(X|Z) + h(Y|Z) + h(ZWL) \le 1+\Delta$.
- Case 4: either $h(L|XY) \le \frac{2\Delta-1}{3}$ or $h(W|XY) \le \frac{2\Delta-1}{3}$, and either $h(Z|X) \le \frac{1+\Delta}{3}$ or $h(Z|Y) \le \frac{1+\Delta}{3}$. Wlog, suppose $h(L|XY) \le \frac{2\Delta-1}{3}$ and $h(Z|X) \le \frac{1+\Delta}{3}$. In this case, we have $h(XYZWL) \le h(L|XYW) + h(Z|XYW) + h(XYW) \le 1+\Delta$.
- Case 5: none of the cases above. As $h(XY|W) > \Delta$ and $h(XYW) \le 1$, we have $h(W) = h(XYW) - h(XY|W) < 1-\Delta$. As $h(XY|L) > \Delta$ and $h(XYL) \le 1$, we have $h(L) < 1-\Delta$. As $h(X|Z) > \frac{\Delta}{2}$ and $h(XZ) \le 1$, we have $h(Z) < 1-\frac{\Delta}{2}$. If either $h(L|XY) \le \frac{2\Delta-1}{3}$ or $h(W|XY) \le \frac{2\Delta-1}{3}$, but $h(Z|X) \le \frac{1+\Delta}{3}$ and $h(Z|Y) \le \frac{1+\Delta}{3}$, we have $h(X) \le 1-\frac{1+\Delta}{3} = \frac{2-\Delta}{3}$ and $h(Y) \le 1-\frac{1+\Delta}{3} = \frac{2-\Delta}{3}$. Hence, $h(XY) \le \frac{2(2-\Delta)}{3}$. If both $h(L|XY) > \frac{2\Delta-1}{3}$ and $h(W|XY) > \frac{2\Delta-1}{3}$, we have $h(XY) \le 1-\frac{2\Delta-1}{3} = \frac{2(2-\Delta)}{3}$. Consider a generalized elimination ordering $\overline{\boldsymbol{\sigma}} = (\{XY\},\{Z\},\{W\},\{L\})$.

$$\mathsf{EMM}_1^{\overline{\sigma}} \le \mathsf{MM}(Z;W;XY|L) \le \max\begin{cases}\gamma h(Z) + h(W) + h(XYL)\\ h(Z) + \gamma h(W) + h(XYL)\\ h(Z) + h(W) + \gamma h(XYL)\end{cases} \le (\omega-2)\cdot(1-\Delta) + 1 - \frac{\Delta}{2} + 1$$

Putting everything together, we obtain the desired upper bound

$$\max\left\{1+\Delta, (\omega-2)\cdot(1-\Delta) + 2 - \frac{\Delta}{2}\right\} = 2 - \frac{1}{2(\omega-2)+3}.$$

□

D MISSING DETAILS FROM SECTION 6

In Eq. (33), $\mathcal{I}$ is the number of max terms at the top level. For every $i \in [\mathcal{I}]$, the i-th term is a maximum over $\boldsymbol{h} \in \Gamma \cap \mathsf{ED}$ of a minimum of $L_i + J_i$ terms that are divided into two sets:

- L_i terms of the form $h(\boldsymbol{U})$ corresponding to $h(U_i^{\overline{\sigma}})$ from Eq. (27).
- J_i terms of the form $h(\boldsymbol{X}|\boldsymbol{G}) + h(Y|\boldsymbol{G}) + \gamma h(\boldsymbol{Z}|\boldsymbol{G}) + h(\boldsymbol{G})$ corresponding to terms in Eq. (21).

Suppose that ω (and by extension γ) is a fixed constant. For a fixed $i \in [\mathcal{I}]$, the inner expression in Eq. (33) is equivalent to an LP, as we now show. The condition $\boldsymbol{h} \in \Gamma \cap \mathsf{ED}$ is a finite collection of linear constraints. The objective function is a minimum of $L_i + J_i$ linear functions of $\boldsymbol{h}$. To convert it to a linear objective function, we introduce a fresh variable t that is lower bounded by each of these $L_i + J_i$ linear functions, and we set the objective to maximize t as follows:

$$\begin{aligned}\max_{t,\boldsymbol{h}\in\Gamma\cap\mathsf{ED}} \Big\{t \quad | \quad &\forall \ell \in [L_i], \quad t \leq h(\boldsymbol{U}_{i\ell}),\\ &\forall j \in [J_i], \quad t \leq h(\boldsymbol{X}_{ij}|\boldsymbol{G}_{ij}) + h(Y_{ij}|\boldsymbol{G}_{ij}) + \gamma h(\boldsymbol{Z}_{ij}|\boldsymbol{G}_{ij}) + h(\boldsymbol{G}_{ij})\Big\}\end{aligned} \tag{49}$$

Hence, computing the ω-submodular width of a hypergraph $\mathcal{H}$ reduces to solving $\mathcal{I}$ linear programs of the form in Eq. (49), and taking the maximum of their optimal values.[13] Section 2.4 exemplifies the above using the hypergraph of the triangle query $Q_\triangle$ (Eq. (2)).

Example D.1. Consider the 4-clique hypergraph from Eq. (23). The ω-submodular width of this hypergraph is given by Eq. (28). Inside the min, we have 10 different terms MM, each of which is a maximum of 3 terms. By distributing the min over the max, we get $\mathcal{I} = 3^{10} = 59049$ terms. Our mechanical algorithm for computing the ω-submodular width of this hypergraph consists of exhaustively solving an LP for each one of these $\mathcal{I}$ terms and takeing their maximum optimal objective value, which turns out to be $\frac{\omega+1}{2}$ in this example. There are smarter but non-algorithmic ways to reach the same conclusion, as shown in Lemma C.6.

E MISSING DETAILS FROM SECTION 7

In this appendix, we prove Theorem 7.1 by showing our algorithm for evaluating a Boolean conjunctive query in ω-submodular width time. To that end, we need to develop a series of technical tools, which we explain in the following subsections.

E.1 Upper bound on the matrix multiplication expression

We start with introducing an upper bound on the matrix multiplication expression from Definition 4.2.

Definition E.1 (ω-dominant triple (α, β, ζ)). A triple of numbers (α, β, ζ) is called *ω-dominant* if it satisfies the following conditions:

$$\alpha, \beta \geq 1, \tag{50}$$

$$\zeta \geq 0, \tag{51}$$

$$\alpha + \beta + \zeta \geq \omega \tag{52}$$

[13] If ω is not fixed, then Eq. (49) is not an LP because the coefficient γ depends on ω. Hence, we end up having constraints that are quadratic in terms of ω and $\boldsymbol{h}$.

Proposition E.2 (Upper bound on $\mathsf{MM}(X;Y;Z|G)$). *Let the triples $(\alpha_1, \beta_1, \zeta_1)$, $(\alpha_2, \beta_2, \zeta_2)$, and $(\alpha_3, \beta_3, \zeta_3)$ be ω-dominant. For any polymatroid $\boldsymbol{h} : 2^{\mathcal{V}} \to \mathbb{R}_+$ and pairwise-disjoint subsets $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{G} \subseteq \mathcal{V}$, the following inequality holds:*

$$\begin{aligned}\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G}) \leq \max\big(&\alpha_1 h(\boldsymbol{X}|\boldsymbol{G}) + \beta_1 h(\boldsymbol{Y}|\boldsymbol{G}) + \zeta_1 h(\boldsymbol{Z}|\boldsymbol{G}) + h(\boldsymbol{G}),\\ &\alpha_2 h(\boldsymbol{X}|\boldsymbol{G}) + \zeta_2 h(\boldsymbol{Y}|\boldsymbol{G}) + \beta_2 h(\boldsymbol{Z}|\boldsymbol{G}) + h(\boldsymbol{G}),\\ &\zeta_3 h(\boldsymbol{X}|\boldsymbol{G}) + \alpha_3 h(\boldsymbol{Y}|\boldsymbol{G}) + \beta_3 h(\boldsymbol{Z}|\boldsymbol{G}) + h(\boldsymbol{G})\big)\end{aligned} \tag{53}$$

Proof. WLOG suppose that $h(\boldsymbol{Z}|\boldsymbol{G})$ is the minimum among the three terms $h(\boldsymbol{X}|\boldsymbol{G})$, $h(\boldsymbol{Y}|\boldsymbol{G})$ and $h(\boldsymbol{Z}|\boldsymbol{G})$. Then, $\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G})$ becomes identical to $h(\boldsymbol{X}|\boldsymbol{G}) + h(\boldsymbol{Y}|\boldsymbol{G}) + \gamma h(\boldsymbol{Z}|\boldsymbol{G}) + h(\boldsymbol{G})$, and the following inequality holds:

$$h(\boldsymbol{X}|\boldsymbol{G}) + h(\boldsymbol{Y}|\boldsymbol{G}) + \gamma h(\boldsymbol{Z}|\boldsymbol{G}) + h(\boldsymbol{G}) \leq \alpha_1 h(\boldsymbol{X}|\boldsymbol{G}) + \beta_1 h(\boldsymbol{Y}|\boldsymbol{G}) + \zeta_1 h(\boldsymbol{Z}|\boldsymbol{G}) + h(\boldsymbol{G})$$

□

E.2 Class of Shannon Inequalities

We now introduce the class of Shannon inequalities at the center of our algorithms.

Definition E.3 (ω-Shannon Inequality). A Shannon inequality is called an ω-Shannon inequality if it has the following form:

$$\sum_{\ell\in[L]} \lambda_\ell h(\boldsymbol{U}_\ell) + \sum_{j\in[J]} \big(\alpha_j h(\boldsymbol{X}_j|\boldsymbol{G}_j) + \beta_j h(\boldsymbol{Y}_j|\boldsymbol{G}_j) + \zeta_j h(\boldsymbol{Z}_j|\boldsymbol{G}_j) + \kappa_j h(\boldsymbol{G}_j)\big) \quad\leq\quad \sum_{i\in[I]} w_i h(\boldsymbol{Y}_i|\boldsymbol{X}_i) \tag{54}$$

where

- all coefficients $\lambda_\ell, \alpha_j, \beta_j, \zeta_j$ and w_i are non-negative,
- all coefficients κ_j are positive,
- and all triples $\left(\frac{\alpha_j}{\kappa_j}, \frac{\beta_j}{\kappa_j}, \frac{\zeta_j}{\kappa_j}\right)$ are ω-dominant.

Following [4], the following proposition follows immediately from Farkas' Lemma. Recall the notation for $h(\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{X})$ from Eq. (18).

Proposition E.4. *Given an ω-Shannon inequality of the form* (54)*, there must exist non-negative vectors $\boldsymbol{m} \stackrel{\text{def}}{=} (m_p)_{p\in[P]}$ and $\boldsymbol{s} \stackrel{\text{def}}{=} (s_q)_{q\in[Q]}$ such that the following equality is an identity over symbolic variables $h(\boldsymbol{X})$ for $\boldsymbol{X} \subseteq \mathcal{V}$ where $h(\emptyset) = 0$:*

$$\begin{aligned}\sum_{\ell\in[L]} \lambda_\ell h(\boldsymbol{U}_\ell) + \sum_{j\in[J]} \big(\alpha_j h(\boldsymbol{X}_j|\boldsymbol{G}_j) + \beta_j h(\boldsymbol{Y}_j|\boldsymbol{G}_j) + \zeta_j h(\boldsymbol{Z}_j|\boldsymbol{G}_j) + \kappa_j h(\boldsymbol{G}_j)\big) \quad=\quad &\sum_{i\in[I]} w_i h(\boldsymbol{Y}_i|\boldsymbol{X}_i)\\ &- \sum_{p\in[P]} m_p h(\boldsymbol{Y}_p|\boldsymbol{X}_p) - \sum_{q\in[Q]} s_q h(\boldsymbol{Y}_q;\boldsymbol{Z}_q|\boldsymbol{X}_q)\end{aligned} \tag{55}$$

We use the standard notation $\|\boldsymbol{\lambda}\|_1$ to denote $\sum_{\ell\in[L]} |\lambda_\ell|$.

Proposition E.5. *Given any ω-Shannon inequality of the form* (54)*, the following inequality holds:*

$$\|\boldsymbol{\lambda}\|_1 + \|\boldsymbol{\kappa}\|_1 \leq \sum_{i\in[I] | \boldsymbol{X}_i = \emptyset} w_i \tag{56}$$

Proof. Every ω-Shannon inequality (54) must hold for the following polymatroid:

$$h(\boldsymbol{W}) = \begin{cases} 0 & \text{if } \boldsymbol{W} = \emptyset \\ 1 & \text{otherwise} \end{cases}$$

When evaluated over the above polymatroid, the RHS of identity (55) is at most $\sum_{i \in [I] | X_i = \emptyset} w_i$, while the LHS is at least $\|\boldsymbol{\lambda}\|_1 + \|\boldsymbol{\kappa}\|_1$. Note that for every $j \in [J]$, we must have $\alpha_j \geq \kappa_j$ because the triple $\left(\frac{\alpha_j}{\kappa_j}, \frac{\beta_j}{\kappa_j}, \frac{\zeta_j}{\kappa_j}\right)$ is ω-dominant. Hence, when $\boldsymbol{G}_j = \emptyset$, the term $\alpha_j h(\boldsymbol{X}_j | \boldsymbol{G}_j)$ contributes at least κ_j to the LHS of (55). □

Definition E.6 (An integral *ω-Shannon inequality).* An ω-Shannon inequality of the form (54) is called *integral* if

- all coefficients $\lambda_\ell, \alpha_j, \beta_j, \zeta_j, \kappa_j$ and w_i are integers,
- and there exist non-negative *integers* $(m_p)_{p \in [P]}$ and $(s_q)_{q \in [Q]}$ such that the identity in Eq. (55) holds.

E.3 Generalizing the Reset Lemma

We present here a highly non-trivial generalization of the Reset Lemma from [3, 4]. In particular, given an integral ω-Shannon inequality (54), this lemma allows us to sacrifice any unconditional term $h(Y_i|\emptyset)$ on the RHS while losing at most one unit from the quantity $\|\boldsymbol{\lambda}\|_1 + \|\boldsymbol{\kappa}\|_1$ on the LHS. Unlike the original Reset Lemma [3, 4], this lemma has to support proper conditionals $h(\boldsymbol{X}_j|\boldsymbol{G}_j)$ on the LHS of the ω-Shannon inequality. Moreover, this lemma has to maintain the ω-dominance property of the triples $\left(\frac{\alpha_j}{\kappa_j}, \frac{\beta_j}{\kappa_j}, \frac{\zeta_j}{\kappa_j}\right)$, which will be crucial for the join algorithm introduced later.

Lemma E.7 (Generalized Reset Lemma). *Given an integral ω-Shannon inequality of the form* (54), *suppose that for some $i_0 \in [I]$, we have $\boldsymbol{X}_{i_0} = \emptyset$ and $w_{i_0} > 0$. Then, there exist non-negative integers $\lambda'_\ell, \alpha'_j, \beta'_j, \zeta'_j, \kappa'_j, w'_i$ for which the following is also an integral ω-Shannon inequality:*

$$\sum_{\ell \in [L]} \lambda'_\ell h(\boldsymbol{U}_\ell) + \sum_{j \in [J]} \left(\alpha'_j h(\boldsymbol{X}_j|\boldsymbol{G}_j) + \beta'_j h(\boldsymbol{Y}_j|\boldsymbol{G}_j) + \zeta'_j h(\boldsymbol{Z}_j|\boldsymbol{G}_j) + \kappa'_j h(\boldsymbol{G}_j)\right) \quad \leq \quad \sum_{i \in [I]} w'_i h(\boldsymbol{Y}_i|\boldsymbol{X}_i)$$

and the coefficients satisfy the following conditions:

$$\begin{aligned} w'_{i_0} &\leq w_{i_0} - 1, \\ w'_i &\leq w_i && \forall i \in [I] \setminus \{i_0\}, \\ \|\boldsymbol{\lambda}'\|_1 + \|\boldsymbol{\kappa}'\|_1 &\geq \|\boldsymbol{\lambda}\|_1 + \|\boldsymbol{\kappa}\|_1 - 1 \end{aligned} \tag{57}$$

Proof. Let $\boldsymbol{W} \stackrel{\text{def}}{=} \boldsymbol{Y}_{i_0}$. We recognize the following cases:

- If $\boldsymbol{W} = \boldsymbol{U}_\ell$ for some $\ell \in [L]$ where $\lambda_\ell > 0$, we set $\lambda'_\ell \stackrel{\text{def}}{=} \lambda_\ell - 1$ and $w'_{i_0} \stackrel{\text{def}}{=} w_{i_0} - 1$. All other coefficients $\lambda'_\ell, \alpha'_j, \beta'_j, \zeta'_j, \kappa'_j, w'_i$ remain the same as the original $\lambda_\ell, \alpha_j, \beta_j, \zeta_j, \kappa_j, w_i$. The new inequality remains a valid integral ω-Shannon inequality since we dropped $h(\boldsymbol{W}|\emptyset)$ from both sides.
- If $\boldsymbol{W} = \boldsymbol{G}_j\boldsymbol{X}_j$ for some $j \in [J]$ where $\alpha_j, \kappa_j > 0$, then we set $\alpha'_j \stackrel{\text{def}}{=} \alpha_j - 1$, $\kappa'_j \stackrel{\text{def}}{=} \kappa_j - 1$, and $w'_{i_0} \stackrel{\text{def}}{=} w_{i_0} - 1$. The inequality still holds since we dropped $h(\boldsymbol{W}|\emptyset)$ from both sides. If $\kappa'_j = 0$, then we drop j from $[J]$. Otherwise, the triple $\left(\frac{\alpha'_j}{\kappa'_j}, \frac{\beta_j}{\kappa'_j}, \frac{\zeta_j}{\kappa'_j}\right)$ is still ω-dominant (Definition E.1). In

particular, inequality (50) continues to hold because:

$$\frac{\alpha'_j}{\kappa'_j} = \frac{\alpha_j - 1}{\kappa_j - 1} \geq \frac{\kappa_j - 1}{\kappa_j - 1} = 1.$$

Moreover, inequality (52) continues to hold because of the following:

$$\begin{aligned}
\frac{\alpha'_j + \beta_j + \zeta_j}{\kappa'_j} &= \frac{\alpha_j + \beta_j + \zeta_j - 1}{\kappa_j - 1} \\
&\geq \frac{\kappa_j \omega - 1}{\kappa_j - 1} && \text{(Inductively by Eq. (52))} \\
&= \omega + \underbrace{\frac{\omega - 1}{\kappa_j - 1}}_{\geq 0} && (58)
\end{aligned}$$

- If $\boldsymbol{W} = \boldsymbol{G}_j \boldsymbol{Y}_j$ for some $j \in [J]$ where $\beta_j, \kappa_j > 0$, then this is symmetric to the previous case. Namely, we set $\beta'_j \stackrel{\text{def}}{=} \beta_j - 1$, $\kappa'_j \stackrel{\text{def}}{=} \kappa_j - 1$, and $w'_{i_0} \stackrel{\text{def}}{=} w_{i_0} - 1$.
- If $\boldsymbol{W} = \boldsymbol{G}_j \boldsymbol{Z}_j$ for some $j \in [J]$ where $\zeta_j, \kappa_j > 0$, then we set $\zeta'_j \stackrel{\text{def}}{=} \zeta_j - 1$, $\kappa'_j \stackrel{\text{def}}{=} \kappa_j - 1$, and $w'_{i_0} \stackrel{\text{def}}{=} w_{i_0} - 1$. If $\kappa'_j = 0$, then we drop j from $[J]$. Otherwise, the triple $\left(\frac{\alpha_j}{\kappa'_j}, \frac{\beta_j}{\kappa'_j}, \frac{\zeta'_j}{\kappa'_j}\right)$ is still ω-dominant. In particular, similar to above, inequality (52) continues to hold as follows:

$$\frac{\alpha_j + \beta_j + \zeta'_j}{\kappa'_j} = \frac{\alpha_j + \beta_j + \zeta_j - 1}{\kappa_j - 1} \geq \omega$$

  The last inequality follows from Eq. (58).
- If $\boldsymbol{W} = \boldsymbol{G}_j$ for some $j \in [J]$ *and* $\kappa_j > \max(\alpha_j, \beta_j, \zeta_j)$, then we set $\kappa'_j \stackrel{\text{def}}{=} \kappa_j - 1$ and $w'_{i_0} \stackrel{\text{def}}{=} w_{i_0} - 1$. The triple $\left(\frac{\alpha_j}{\kappa'_j}, \frac{\beta_j}{\kappa'_j}, \frac{\zeta_j}{\kappa'_j}\right)$ is still ω-dominant. On the other hand, if $\boldsymbol{W} = \boldsymbol{G}_j$ and $\kappa_j \leq \max(\alpha_j, \beta_j, \zeta_j)$, then $\kappa_j h(\boldsymbol{G}_j)$ already cancels out with one of the three terms $\alpha_j h(\boldsymbol{X}_j|\boldsymbol{G}_j)$, $\beta_j h(\boldsymbol{Y}_j|\boldsymbol{G}_j)$, $\zeta_j h(\boldsymbol{Z}_j|\boldsymbol{G}_j)$ on the LHS. Hence, this case falls under the next case.
- In all other cases, $h(\boldsymbol{W}|\emptyset)$ must cancel out with some term on the RHS of the identity (55), just like in the original Reset Lemma [3, 4]. Here we recognize the same three cases from the original Reset Lemma, which we repeat below for self-containment:
  - **Case 1:** $h(\boldsymbol{W})$ cancels out with some other term $h(\boldsymbol{Y}_i|\boldsymbol{X}_i)$ for some $i \in [I] \setminus \{i_0\}$ where $\boldsymbol{X}_i = \boldsymbol{W}$. In this case, we apply the following rewrite to the RHS of the identity (55):
  $$h(\boldsymbol{W}) + h(\boldsymbol{Y}_i|\boldsymbol{W}) = h(\boldsymbol{Y}_i\boldsymbol{W}) \tag{59}$$
  And now inductively, our target is to cancel out the newly added term $h(\boldsymbol{Y}_i\boldsymbol{W})$.
  - **Case 2:** $h(\boldsymbol{W})$ cancels out with some term $-h(\boldsymbol{Y}_p|\boldsymbol{X}_p)$ from some $p \in [P]$ where $\boldsymbol{W} = \boldsymbol{X}_p\boldsymbol{Y}_p$. In this case, we apply the following rewrite to the RHS of the identity (55):
  $$h(\boldsymbol{W}) - h(\boldsymbol{Y}_p|\boldsymbol{X}_p) = h(\boldsymbol{W}) - h(\boldsymbol{X}_p\boldsymbol{Y}_p) + h(\boldsymbol{X}_p) = h(\boldsymbol{X}_p) \tag{60}$$
  And now, we proceed inductively to eliminate the new term $h(\boldsymbol{X}_p)$.
  - **Case 3:** $h(\boldsymbol{W})$ cancels out with some term $-h(\boldsymbol{Y}_q; \boldsymbol{Z}_q|\boldsymbol{X}_q)$ for some $q \in [Q]$ where $\boldsymbol{W} = \boldsymbol{X}_q\boldsymbol{Y}_q$. In this case, we apply the rewrite:
  $$\begin{aligned} h(\boldsymbol{W}) - h(\boldsymbol{Y}_q; \boldsymbol{Z}_q|\boldsymbol{X}_q) &= h(\boldsymbol{W}) - h(\boldsymbol{X}_q\boldsymbol{Y}_q) - h(\boldsymbol{X}_q\boldsymbol{Z}_q) + h(\boldsymbol{X}_q) + h(\boldsymbol{X}_q\boldsymbol{Y}_q\boldsymbol{Z}_q) \\ &= h(\boldsymbol{X}_q\boldsymbol{Y}_q\boldsymbol{Z}_q) - h(\boldsymbol{Z}_q|\boldsymbol{X}_q) \end{aligned} \tag{61}$$
  And now we inductively eliminate $h(\boldsymbol{X}_q\boldsymbol{Y}_q\boldsymbol{Z}_q)$.

By applying each one of the above three cases, the following integral quantity always decreases by at least one, thus proving that this process must terminate:

$$\|\boldsymbol{w}\|_1 + \|\boldsymbol{m}\|_1 + 2\|\boldsymbol{s}\|_1 \tag{62}$$

In particular, Case 1 decreases $\|\boldsymbol{w}\|_1$ by 1, Case 2 decreases $\|\boldsymbol{m}\|_1$ by 1, and Case 3 *increases* $\|\boldsymbol{m}\|_1$ by 1 and decreases $\|\boldsymbol{s}\|_1$ by 1.

□

E.4 Generalizing the Proof Sequence Construction

The following is a generalization of the proof sequence construction from [3, 4]. In particular, the original proof sequence construction cannot be used directly because an ω-Shannon inequality (54) might contain proper conditionals $h(\boldsymbol{X}_j|\boldsymbol{G}_j)$ on the LHS. The other challenge here is maintaining the ω-dominance property of the triples $\left(\frac{\alpha_j}{\kappa_j}, \frac{\beta_j}{\kappa_j}, \frac{\zeta_j}{\kappa_j}\right)$, which is needed later to get a corresponding join algorithm that uses matrix multiplication.

Theorem E.8 (Generalized Proof Sequence Construction). *Given an integral ω-Shannon inequality of the form* (54)*, there exists a finite sequence of steps that transforms the RHS of the inequality into the LHS. Each step in the sequence replaces some terms on the RHS with smaller terms, thus the full sequence proves that the LHS is smaller than the RHS. Moreover, each step has either one of the following forms:*

- **Decomposition Step:** $h(\boldsymbol{X}\boldsymbol{Y}) \to h(\boldsymbol{X}) + h(\boldsymbol{Y}|\boldsymbol{X})$.
- **Composition Step:** $h(\boldsymbol{X}) + h(\boldsymbol{Y}|\boldsymbol{X}) \to h(\boldsymbol{X}\boldsymbol{Y})$.
- **Monotonicity Step:** $h(\boldsymbol{X}\boldsymbol{Y}) \to h(\boldsymbol{X})$.
- **Submodularity Step:** $h(\boldsymbol{Y}|\boldsymbol{X}) \to h(\boldsymbol{Y}|\boldsymbol{X}\boldsymbol{Z})$.

Note that by Shannon inequalities, each one of the above steps replaces one or two terms with *smaller* terms. In particular,

$$\begin{aligned}
h(\boldsymbol{X}\boldsymbol{Y}) &= h(\boldsymbol{X}) + h(\boldsymbol{Y}|\boldsymbol{X}) && \text{(by Eq. (17))}\\
h(\boldsymbol{X}\boldsymbol{Y}) &\geq h(\boldsymbol{X}) && \text{(by Eq. (15))}\\
h(\boldsymbol{Y}|\boldsymbol{X}) &\geq h(\boldsymbol{Y}|\boldsymbol{X}\boldsymbol{Z}) && \text{(by Eq. (14))}
\end{aligned}$$

For the purpose of proving the above theorem, we will represent an ω-Shannon inequality slightly differently from Eq. (54). In particular, some terms on the LHS of Eq. (54) might already occur on the RHS, therefore we could decompose the RHS into two parts: One part which is a subset of terms on the LHS, and another part containing the remaining RHS terms. Formally, we can represent an ω-Shannon inequality as follows:

$$\begin{aligned}
\sum_{\ell\in[L]} \lambda_\ell h(\boldsymbol{U}_\ell) + \sum_{j\in[J]} \left(\alpha_j h(\boldsymbol{X}_j|\boldsymbol{G}_j) + \beta_j h(\boldsymbol{Y}_j|\boldsymbol{G}_j) + \zeta_j h(\boldsymbol{Z}_j|\boldsymbol{G}_j) + \kappa_j h(\boldsymbol{G}_j)\right) \quad &\leq \\
\sum_{j\in[J]} \left(\hat{\alpha}_j h(\boldsymbol{X}_j|\boldsymbol{G}_j) + \hat{\beta}_j h(\boldsymbol{Y}_j|\boldsymbol{G}_j) + \hat{\zeta}_j h(\boldsymbol{Z}_j|\boldsymbol{G}_j) + \hat{\kappa}_j h(\boldsymbol{G}_j)\right) + \sum_{i\in[I]} \hat{w}_i h(\boldsymbol{Y}_i|\boldsymbol{X}_i) &
\end{aligned} \tag{63}$$

where in addition to the conditions from Definition E.3, we also require that

- For all $j \in [J]$, we have $0 \leq \hat{\alpha}_j \leq \alpha_j$, $0 \leq \hat{\beta}_j \leq \beta_j$, $0 \leq \hat{\zeta}_j \leq \zeta_j$, and $0 \leq \hat{\kappa}_j \leq \kappa_j$.
- For all $i \in [I]$, we have $0 \leq \hat{w}_i \leq w_i$.
- Furthermore, for each $j \in [J]$, we can assume that $\hat{\alpha}_j + \hat{\beta}_j + \hat{\zeta}_j + \hat{\kappa}_j < \alpha_j + \beta_j + \zeta_j + \kappa_j$. Otherwise, we could drop the term j from both sides.

- Finally, we assume, for all $j \in [J]$,

$$\kappa_j - \hat{\kappa}_j \geq \max(\alpha_j - \hat{\alpha}_j, \beta_j - \hat{\beta}_j, \zeta_j - \hat{\zeta}_j) \tag{64}$$

By Proposition E.4, for every inequality of the form (63), there exist non-negative vectors $\boldsymbol{m} \stackrel{\text{def}}{=} (m_p)_{p\in[P]}$ and $\boldsymbol{s} \stackrel{\text{def}}{=} (s_q)_{q\in[Q]}$ such that the following identity holds over symbolic variables $h(\boldsymbol{X})$ for $\boldsymbol{X} \subseteq \mathcal{V}$ where $h(\emptyset) = 0$:

$$\begin{aligned}\sum_{\ell\in[L]} \lambda_\ell h(\boldsymbol{U}_\ell) + \sum_{j\in[J]} \left(\alpha_j h(\boldsymbol{X}_j|\boldsymbol{G}_j) + \beta_j h(\boldsymbol{Y}_j|\boldsymbol{G}_j) + \zeta_j h(\boldsymbol{Z}_j|\boldsymbol{G}_j) + \kappa_j h(\boldsymbol{G}_j)\right) \quad &= \\ \sum_{j\in[J]} \left(\hat{\alpha}_j h(\boldsymbol{X}_j|\boldsymbol{G}_j) + \hat{\beta}_j h(\boldsymbol{Y}_j|\boldsymbol{G}_j) + \hat{\zeta}_j h(\boldsymbol{Z}_j|\boldsymbol{G}_j) + \hat{\kappa}_j h(\boldsymbol{G}_j)\right) + \sum_{i\in[I]} \hat{w}_i h(\boldsymbol{Y}_i|\boldsymbol{X}_i)& \\ - \sum_{p\in[P]} m_p h(\boldsymbol{Y}_p|\boldsymbol{X}_p) - \sum_{q\in[Q]} s_q h(\boldsymbol{Y}_q;\boldsymbol{Z}_q|\boldsymbol{X}_q)&\end{aligned} \tag{65}$$

Proof of Theorem E.8. An inequality of the form (54) can be converted to the from (63) by initializing $\hat{w}_i = w_i$ for all $i \in [I]$ and $\hat{\alpha}_j = \hat{\beta}_j = \hat{\zeta}_j = \hat{\kappa}_j = 0$ for all $j \in [J]$. Consider an inequality of the form (63) and the corresponding identity (65). If $\|\boldsymbol{\lambda}\|_1 + \|\boldsymbol{\kappa} - \hat{\boldsymbol{\kappa}}\|_1 = 0$, then by Condition (64), we have $\hat{\alpha}_j = \alpha_j$, $\hat{\beta}_j = \beta_j$, $\hat{\zeta}_j = \zeta_j$ and $\hat{\kappa}_j = \kappa_j$ for all j and we are done. Now assume that $\|\boldsymbol{\lambda}\|_1 + \|\boldsymbol{\kappa} - \hat{\boldsymbol{\kappa}}\|_1 > 0$. By Proposition E.5, we have

$$\sum_{i\in[I]|\boldsymbol{X}_i=\emptyset} \hat{w}_i \geq \|\boldsymbol{\lambda}\|_1 + \|\boldsymbol{\kappa} - \hat{\boldsymbol{\kappa}}\|_1 > 0.$$

Hence, there exists some $i_0 \in [I]$ such that $\boldsymbol{X}_{i_0} = \emptyset$ and $\hat{w}_{i_0} > 0$. Define $\boldsymbol{W} \stackrel{\text{def}}{=} \boldsymbol{Y}_{i_0}$. We recognize the following cases:

- If there exists $\ell \in [L]$ where $\boldsymbol{U}_\ell = \boldsymbol{W}$ and $\lambda_\ell > 0$, then we reduce both λ_ℓ and $\hat{w}_{i_0}$ by one thus canceling out the term $h(\boldsymbol{W})$ from both sides of the identity (65).
- If there exists $j \in [J]$ where $\boldsymbol{G}_j\boldsymbol{X}_j = \boldsymbol{W}$ and $\alpha_j > \hat{\alpha}_j$, then we apply the following decomposition step to the term $h(\boldsymbol{Y}_{i_0}|\boldsymbol{X}_{i_0}) = h(\boldsymbol{W}|\emptyset)$ on the RHS of (65):

$$h(\boldsymbol{W}) \to h(\boldsymbol{G}_j) + h(\boldsymbol{X}_j|\boldsymbol{G}_j)$$

  and now we have $h(\boldsymbol{X}_j|\boldsymbol{G}_j)$ on both sides of the identity (65). We increase $\hat{\alpha}_j$ by one thus pairing the two terms $h(\boldsymbol{X}_j|\boldsymbol{G}_j)$ with one another. Inequality (64) continues to hold.
- If there exists $j \in [J]$ where either $\boldsymbol{G}_j\boldsymbol{Y}_j = \boldsymbol{W}$ and $\beta_j > \hat{\beta}_j$, or $\boldsymbol{G}_j\boldsymbol{Z}_j = \boldsymbol{W}$ and $\zeta_j > \hat{\zeta}_j$, then this is similar to the previous case.
- If there exists $j \in [J]$ where $\boldsymbol{G}_j = \boldsymbol{W}$ and $\kappa_j - \hat{\kappa}_j > \max(\alpha_j - \hat{\alpha}_j, \beta_j - \hat{\beta}_j, \zeta_j - \hat{\zeta}_j)$, then we have $h(\boldsymbol{W})$ on both sides of identity (65). We increase $\hat{\kappa}_j$ by one thus pairing the two terms $h(\boldsymbol{G}_j)$ with one another. Condition (64) continues to hold. On the other hand, if $\boldsymbol{G}_j = \boldsymbol{W}$ and $\kappa_j - \hat{\kappa}_j = \max(\alpha_j - \hat{\alpha}_j, \beta_j - \hat{\beta}_j, \zeta_j - \hat{\zeta}_j)$, then $h(\boldsymbol{G}_j)$ already cancels out with one of the three terms $h(\boldsymbol{X}_j|\boldsymbol{G}_j)$, $h(\boldsymbol{Y}_j|\boldsymbol{G}_j)$, and $h(\boldsymbol{Z}_j|\boldsymbol{G}_j)$ on the LHS of (65), hence this case falls under the next case.
- In all other cases, the term $h(\boldsymbol{W})$ must cancel out with some other term on the RHS of the identity (65). We recognize three cases similar to the original proof sequence construction from [3, 4]:
  - **Case 1:** $h(\boldsymbol{W})$ cancels out with some other term $h(\boldsymbol{Y}_i|\boldsymbol{X}_i)$ for some $i \in [I]$ on the RHS where $\boldsymbol{X}_i = \boldsymbol{W}$. In this case, we can compose the two terms using the rewrite from Eq. (59), which

corresponds to a composition step:

$$h(\boldsymbol{W}) + h(Y_i|\boldsymbol{W}) \rightarrow h(Y_i\boldsymbol{W}) \qquad \text{(Composition Step)}$$

- **Case 2:** $h(\boldsymbol{W})$ cancels out with some term $-h(\boldsymbol{Y}_p|\boldsymbol{X}_p)$ for some $p \in [P]$ where $\boldsymbol{W} = \boldsymbol{X}_p\boldsymbol{Y}_p$. In this case, we can apply the rewrite from Eq. (60), which corresponds to a monotonicity step:

$$h(\boldsymbol{W}) \rightarrow h(\boldsymbol{X}_p) \qquad \text{(Monotonicity Step)}$$

- **Case 3:** $h(\boldsymbol{W})$ cancels out with some term $-h(\boldsymbol{Y}_q;\boldsymbol{Z}_q|\boldsymbol{X}_q)$ for some $q \in [Q]$ where $\boldsymbol{W} = \boldsymbol{X}_q\boldsymbol{Y}_q$. In this case, instead of applying the rewrite from Eq. (61), we apply the following rewrite:

$$\begin{aligned} h(\boldsymbol{W}) - h(\boldsymbol{Y}_q;\boldsymbol{Z}_q|\boldsymbol{X}_q) &= h(\boldsymbol{W}) - h(\boldsymbol{X}_q\boldsymbol{Y}_q) - h(\boldsymbol{X}_q\boldsymbol{Z}_q) + h(\boldsymbol{X}_q) + h(\boldsymbol{X}_q\boldsymbol{Y}_q\boldsymbol{Z}_q) \\ &= h(\boldsymbol{X}_q) + h(\boldsymbol{Y}_q|\boldsymbol{X}_q\boldsymbol{Z}_q) \end{aligned} \tag{66}$$

The above rewrite corresponds to applying a decomposition step followed by a submodularity step:

$$\begin{aligned} h(\boldsymbol{W}) &\rightarrow h(\boldsymbol{X}_q) + h(\boldsymbol{Y}_q|\boldsymbol{X}_q) && \text{(Decomposition Step)} \\ h(\boldsymbol{Y}_q|\boldsymbol{X}_q) &\rightarrow h(\boldsymbol{Y}_q|\boldsymbol{X}_q\boldsymbol{Z}_q) && \text{(Submodularity Step)} \end{aligned}$$

In each one of the above three cases, the following integral quantity always decreases by at least one, thus proving that this process must terminate:

$$\|\hat{\boldsymbol{w}}\|_1 + \|\boldsymbol{m}\|_1 + 2\|\boldsymbol{s}\|_1 \tag{67}$$

In particular, Case 1 decreases $\|\hat{\boldsymbol{w}}\|_1$ by 1, Case 2 decreases $\|\boldsymbol{m}\|_1$ by 1, and Case 3 *increases* $\|\hat{\boldsymbol{w}}\|_1$ by 1 and decreases $\|\boldsymbol{s}\|_1$ by 1.

□

E.5 Generalizing PANDA for Disjunctive Datalog Rules

Consider a Boolean conjunctive query Q of the form given by Eq. (1). Our algorithms for evaluating Q rely heavily on the concept of *degree* in a relation, that is defined below.

Definition E.9 (Degree in a relation). Let $R(\boldsymbol{Z})$ be a relation over variable set $\boldsymbol{Z}$, and let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be two sets of variables such that $Y \setminus \boldsymbol{X} \subseteq \boldsymbol{Z}$. Given a tuple $\boldsymbol{x}$ over the variable set $\boldsymbol{X}$, we define the *degree of* Y *conditioned on* $\boldsymbol{X} = \boldsymbol{x}$ *in* R as follows:

$$\deg_R(Y|\boldsymbol{X} = \boldsymbol{x}) \stackrel{\text{def}}{=} \left|\pi_{Y\setminus X}\left(\sigma_{X\cap Z=\pi_{X\cap Z}(\boldsymbol{x})}(R)\right)\right| \tag{68}$$

Moreover, we define the *degree of* Y *conditioned on* $\boldsymbol{X}$ *in* R as follows:

$$\deg_R(Y|\boldsymbol{X}) \stackrel{\text{def}}{=} \max_{\boldsymbol{x}} \deg_R(Y|\boldsymbol{X} = \boldsymbol{x}) \tag{69}$$

As a building block for the next section which presents our algorithm for evaluating a query Q in ω-submodular width time, we present in this section an algorithm for evaluating a specific type of *disjunctive Datalog rules* [3, 4]. We can think of this algorithm as being concerned with evaluating a query $Q_\vee$ that has a disjunction in the head:

$$\bigvee_{\ell\in[L]} P_\ell(\boldsymbol{U}_\ell) \vee \bigvee_{j\in[J]} \left(S_j(\boldsymbol{X}_j\boldsymbol{G}_j) \wedge T_j(\boldsymbol{Y}_j\boldsymbol{G}_j) \wedge W_j(\boldsymbol{Z}_j\boldsymbol{G}_g)\right) \quad \text{:-} \quad \bigwedge_{R(\boldsymbol{Z})\in\text{atoms}(Q)} R(\boldsymbol{Z}) \tag{70}$$

Evaluating the above query means computing output tables $P_\ell(\boldsymbol{U}_\ell)$ for $\ell \in [L]$, and $S_j(\boldsymbol{X}_j\boldsymbol{G}_j)$, $T_j(\boldsymbol{Y}_j\boldsymbol{G}_j)$, $W_j(\boldsymbol{Z}_j\boldsymbol{G}_j)$ for $j \in [J]$ so for that every tuple $\boldsymbol{t}$ in the join of all atoms in the body of the query, the projection of $\boldsymbol{t}$ must be present in either some output table P_ℓ or some triple of output tables (S_j, T_j, W_j).

Theorem E.10 (Evaluating a disjunctive rule (70)). *Suppose we are given the following:*

- *A Boolean query Q of the form* (1) *along with a corresponding input database instance D.*
- *An integral ω-Shannon inequality of the form* (54) *where for every $i \in [I]$, there exists an atom $R_i(\boldsymbol{Z}_i) \in \mathsf{atoms}(Q)$ that satisfies $\boldsymbol{Y}_i \setminus \boldsymbol{X}_i \subseteq \boldsymbol{Z}_i$.*

Define the quantity:

$$\mathsf{obj} \stackrel{\text{def}}{=} \frac{\sum_{i\in[I]} w_i \log_2 \deg_{R_i}(\boldsymbol{Y}_i|\boldsymbol{X}_i)}{\|\boldsymbol{\lambda}\|_1 + \|\boldsymbol{\kappa}\|_1} \tag{71}$$

Then, there is an algorithm that runs in time $\tilde{O}(2^{\mathsf{obj}})$ and computes tables $P_\ell(\boldsymbol{U}_\ell)$ for $\ell \in [L]$ and tables $(S_j(\boldsymbol{X}_j\boldsymbol{G}_j), T_j(\boldsymbol{Y}_j\boldsymbol{G}_j), W_j(\boldsymbol{Z}_j\boldsymbol{G}_j))$ for $j \in [J]$ that satisfy the following:

- *For each tuple $\boldsymbol{t} \in \Join_{R(\boldsymbol{Z})\in\mathsf{atoms}(Q)} R(\boldsymbol{Z})$:*
  - *either there exists $\ell \in [L]$ where $\pi_{\boldsymbol{U}_\ell}(\boldsymbol{t}) \in P_\ell$.*
  - *or there exists $j \in [J]$ where $\pi_{\boldsymbol{X}_j\boldsymbol{G}_j}(\boldsymbol{t}) \in S_j$, $\pi_{\boldsymbol{Y}_j\boldsymbol{G}_j}(\boldsymbol{t}) \in T_j$, and $\pi_{\boldsymbol{Z}_j\boldsymbol{G}_j}(\boldsymbol{t}) \in W_j$.*
- *For each $\ell \in [L]$, we have $|P_\ell| = \tilde{O}(2^{\mathsf{obj}})$.*
- *For each $j \in [J]$, we have $|S_j|, |T_j|, |W_j| = \tilde{O}(2^{\mathsf{obj}})$ and there exists an ω-dominant triple $(\alpha'_j, \beta'_j, \zeta'_j)$ such that*

$$|\pi_{\boldsymbol{G}_j}(S_j)| \cdot \left(\deg_{S_j}(\boldsymbol{X}_j|\boldsymbol{G}_j)\right)^{\alpha'_j} \cdot \left(\deg_{T_j}(\boldsymbol{Y}_j|\boldsymbol{G}_j)\right)^{\beta'_j} \cdot \left(\deg_{W_j}(\boldsymbol{Z}_j|\boldsymbol{G}_j)\right)^{\zeta'_j} = \tilde{O}(2^{\mathsf{obj}}). \tag{72}$$

Proof. The algorithm is a variant of the PANDA algorithm from [3, 4]. The quantity obj is a constant that is fixed at the beginning of the algorithm. The execution of the algorithm branches into a polylogarithmic number of branches, each of which is concerned with a different part of the data. Throughout the algorithm, and on each branch, we will modify the integral ω-Shannon inequality (54) and we will also update the tables $(R_i(\boldsymbol{Z}_i))_{i\in[I]}$ that are associated with terms $h(\boldsymbol{Y}_i|\boldsymbol{X}_i)$ on the RHS of inequality (54). However, on each branch, we will always maintain the following invariants:

(1) The current inequality is a valid integral ω-Shannon inequality (54).
(2) Every term $h(\boldsymbol{Y}_i|\boldsymbol{X}_i)$ on the RHS of the inequality (54) has an associated table $R_i(\boldsymbol{Z}_i)$ that satisfies $\boldsymbol{Y}_i \setminus \boldsymbol{X}_i \subseteq \boldsymbol{Z}_i$ and

$$|R_i| \leq 2^{\mathsf{obj}} \tag{73}$$

(3) The inequality satisfies:

$$\|\boldsymbol{\lambda}\|_1 + \|\boldsymbol{\kappa}\|_1 > 0 \tag{74}$$

(4) The inequality along with the associated tables satisfy:

$$\frac{\sum_{i\in[I]} w_i \log_2 \deg_{R_i}(\boldsymbol{Y}_i|\boldsymbol{X}_i)}{\|\boldsymbol{\lambda}\|_1 + \|\boldsymbol{\kappa}\|_1} \leq \mathsf{obj} \tag{75}$$

On every branch, the algorithm follows the proof sequence construction from Theorem E.8. For that purpose, we represent the ω-Shannon inequality (54) using the representation from Eq. (63). Every proof step replaces one or two terms $h(\boldsymbol{Y}_i|\boldsymbol{X}_i)$ on the RHS of the inequality with one or two new terms that are smaller. Simultaneously, the algorithm replaces the tables $R_i(\boldsymbol{Z}_i)$ that are associated with the old terms with new tables that are associated with the new terms, where these new tables are produced through some algorithmic operation on the old ones. WLOG we can assume that on every branch, and for every $\boldsymbol{Z} \subseteq \mathcal{V}$, there is at most one table $R(\boldsymbol{Z})$. This is because if we have multiple tables $R(\boldsymbol{Z})$ on the same branch, we can always take their intersection to obtain a single table $R(\boldsymbol{Z})$ that achieves the minimum degree $\deg_R(\boldsymbol{Y}|\boldsymbol{X})$ among all these tables for every $\boldsymbol{X}$ and $\boldsymbol{Y}$. The algorithm terminates for a specific branch if either one of the following two conditions is met:

(a) There exists some $\ell \in [L]$ and $i \in [I]$ where $\lambda_\ell, \hat{w}_i > 0$, $\boldsymbol{Y}_i = \boldsymbol{U}_\ell$, $\boldsymbol{X}_i = \emptyset$. In this case, invariant (73) implies that the associated table, denoted $P_\ell(\boldsymbol{U}_\ell) \stackrel{\text{def}}{=} R_i(\boldsymbol{U}_\ell)$, has size $|P_\ell| \leq 2^{\text{obj}}$.

(b) There exists some $j \in [J]$ where $\hat{\alpha}_j = \alpha_j$, $\hat{\beta}_j = \beta_j$, $\hat{\zeta}_j = \zeta_j$, $\hat{\kappa}_j = \kappa_j$ and the associated tables, denoted $S_j(\boldsymbol{X}_j\boldsymbol{G}_j)$, $T_j(\boldsymbol{Y}_j\boldsymbol{G})$, and $W_j(\boldsymbol{Z}_j\boldsymbol{G}_j)$, satisfy the following: (WLOG we can assume that $\pi_{G_j}(S_j) = \pi_{G_j}(T_j) = \pi_{G_j}(W_j)$ because we can always semijoin reduce the tables together.)

$$|\pi_{G_j}(S_j)| \cdot \left(\deg_{S_j}(\boldsymbol{X}_j|\boldsymbol{G}_j)\right)^{\frac{\alpha_j}{\kappa_j}} \cdot \left(\deg_{T_j}(\boldsymbol{Y}_j|\boldsymbol{G}_j)\right)^{\frac{\beta_j}{\kappa_j}} \cdot \left(\deg_{W_j}(\boldsymbol{Z}_j|\boldsymbol{G}_j)\right)^{\frac{\zeta_j}{\kappa_j}} \leq 2^{\text{obj}} \tag{76}$$

(Recall that because the inequality is an ω-Shannon inequality, the triple $\left(\frac{\alpha_j}{\kappa_j}, \frac{\beta_j}{\kappa_j}, \frac{\zeta_j}{\kappa_j}\right)$ must be ω-dominant.) The final output of the algorithm is the union of the tables P_ℓ, S_j, T_j and W_j from all branches. In contrast, if condition (b) above is met except that inequality (76) is violated, then we drop the term j from both sides of the inequality. In this case, both invariants from Eq. (74) and Eq. (75) continue to hold. In particular, Eq. (75) continues to hold because on the LHS, the denominator is reduced by κ_j while the numerator is reduced by:

$$\kappa_j \log_2 |\pi_{G_j}(S_j)| + \alpha_j \log_2 \deg_{S_j}(\boldsymbol{X}_j|\boldsymbol{G}_j) + \beta_j \log_2 \deg_{T_j}(\boldsymbol{Y}_j|\boldsymbol{G}_j) + \zeta_j \log_2 \deg_{W_j}(\boldsymbol{Z}_j|\boldsymbol{G}_j) > \kappa_j \cdot \text{obj}$$

Eq. (74) continues to hold because if the new denominator was zero, then the LHS of Eq. (75) before we dropped the j-term must have been greater than $\frac{\kappa_j \cdot \text{obj}}{\kappa_j}$, hence Eq. (75) could not have been satisfied to begin with.

Initially, all invariants are satisfied except possibly for Eq. (73). However, if Eq. (73) is violated, we can use the Reset Lemma E.7 to drop the large relation R_i along with the corresponding term $h(\boldsymbol{Y}_i|\boldsymbol{X}_i)$. Just like above, dropping R_i preserves all invariants including Eq. (75) and Eq. (74).

On every branch, the algorithm translates each proof step from Theorem E.8 into an algorithmic operation, as described in the original PANDA algorithm. We summarize these operations below for self-containment:

- **Composition Step** $h(\boldsymbol{X}) + h(\boldsymbol{Y}|\boldsymbol{X}) \to h(\boldsymbol{XY})$**:** Let R and S be the two tables associated with $h(\boldsymbol{X})$ and $h(\boldsymbol{Y}|\boldsymbol{X})$ respectively. We recognize two cases:
  - If $|R| \cdot \deg_S(\boldsymbol{Y}|\boldsymbol{X}) \leq 2^{\text{obj}}$, then we compute the join $T \stackrel{\text{def}}{=} R \bowtie S$. The new table T is associated with the new term $h(\boldsymbol{XY})$, and it satisfies Eq. (73).
  - If $|R| \cdot \deg_S(\boldsymbol{Y}|\boldsymbol{X}) > 2^{\text{obj}}$, then we replace the two terms $h(\boldsymbol{X}) + h(\boldsymbol{Y}|\boldsymbol{X})$ with $h(\boldsymbol{XY})$ and then drop the term $h(\boldsymbol{XY})$ using the Reset Lemma (Lemma E.7). Invariant (75) continues to hold because on the LHS of (75), the denominator is reduced by at most one (thanks to Eq. (57)) while the numerator is reduced by $\log_2 |R| + \log_2 \deg_S(\boldsymbol{Y}|\boldsymbol{X}) > \text{obj}$. Invariant (74) continues to hold because if the new denominator was zero, then the LHS of Eq. (75) before the reset must have been greater than obj, hence invariant (75) could not have been satisfied to begin with.
- **Monotonicity Step** $h(\boldsymbol{XY}) \to h(\boldsymbol{X})$**:** Let R be the table associated with $h(\boldsymbol{XY})$. We associate R with the new term $h(\boldsymbol{X})$. Because $\deg_R(\boldsymbol{X}|\emptyset) \leq \deg_R(\boldsymbol{XY}|\emptyset)$, invariant (75) continues to hold.
- **Submodularity Step** $h(\boldsymbol{Y}|\boldsymbol{X}) \to h(\boldsymbol{Y}|\boldsymbol{XZ})$**:** Let R be the table associated with $h(\boldsymbol{Y}|\boldsymbol{X})$. We associate R with the new term $h(\boldsymbol{Y}|\boldsymbol{XZ})$. Invariant (75) holds because $\deg_R(\boldsymbol{Y}|\boldsymbol{XZ}) \leq \deg_R(\boldsymbol{Y}|\boldsymbol{X})$.
- **Decomposition Step** $h(\boldsymbol{XY}) \to h(\boldsymbol{X}) + h(\boldsymbol{Y}|\boldsymbol{X})$**:** Let R be the table associated with $h(\boldsymbol{XY})$. We partition R into $k \stackrel{\text{def}}{=} O(\log_2 |R|)$ buckets $R^1, \ldots, R^k$ based on the degree of $\boldsymbol{Y}$ conditioned on $\boldsymbol{X}$. In particular, bucket R^i contains all tuples $\boldsymbol{t} \in R$ such that $\deg_R(\boldsymbol{Y}|\boldsymbol{X} = \pi_{\boldsymbol{X}}(\boldsymbol{t})) \in [2^i, 2^{i+1})$. Now,

within each bucket R^i, we have

$$|\pi_X(R^i)| \cdot \deg_{R^i}(Y|X) \leq 2 \cdot |\pi_{XY}(R)| \tag{77}$$

If we partition each bucket R^i into two sub-buckets, we can drop the factor of two from the above inequality. We create a new branch of the algorithm for each bucket R^i. Within the i-th branch, we associate the two new terms $h(X)$ and $h(Y|X)$ with R^i. Invariant (75) continues to hold because inequality (77) implies (after dropping the factor of two):

$$\log_2 \deg_{R_i}(X|\emptyset) + \log_2 \deg_{R_i}(Y|X) \leq \log_2 \deg_R(XY|\emptyset)$$

□

E.6 Proof of Theorem 7.1

We are now ready to prove Theorem 7.1 about answering Boolean conjunctive queries in ω-submodular width time.

Theorem 7.1. *Assuming that ω is a rational number, given a Boolean conjunctive query Q and a corresponding input database instance D, there is an algorithm that computes the answer to Q in time $\tilde{O}(N^{\omega\text{-subw}(Q)})$, where N is the size of D.*

In order to prove Theorem 7.1, we will need some further preliminaries. The following lemma is very similar to a lemma in [3, 4]. Its proof relies on the observation that every feasible dual solution to the linear program (49) corresponds to an ω-Shannon inequality (54) of a certain structure. By strong duality, if we pick an optimal dual solution, we can match the optimal objective value of the (primal) LP (49).

Lemma E.11 (From LP (49) to an ω-Shannon Inequality (54)). *Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, let $\mathsf{ED} \stackrel{\text{def}}{=} \mathsf{ED}_{\mathcal{H}}$, and consider a linear program of the following form (which is the same as LP (49) but where we drop the index $i \in [\mathcal{I}]$ to reduce clutter):*

$$\begin{aligned}\max_{t, \boldsymbol{h} \in \Gamma \cap \mathsf{ED}} \big\{ t \quad | \quad & \forall \ell \in [L], \quad t \leq h(\boldsymbol{U}_\ell), \\ & \forall j \in [J], \quad t \leq h(\boldsymbol{X}_j|\boldsymbol{G}_j) + h(\boldsymbol{Y}_j|\boldsymbol{G}_j) + \gamma h(\boldsymbol{Z}_j|\boldsymbol{G}_j) + h(\boldsymbol{G}_j) \big\}\end{aligned} \tag{78}$$

Let opt *be the optimal objective value of the above LP. Then, there must exist an ω-Shannon inequality of the form* (54) *that satisfies the following:*

- *For each $i \in [I]$, we have $\boldsymbol{X}_i = \emptyset$ and $\boldsymbol{Y}_i \in \mathcal{E}$.*
- *For each $j \in [J]$, we have $\alpha_j = \beta_j = \kappa_j$ and $\zeta_j = \kappa_j \cdot \gamma$.*
- *The coefficients of the inequality satisfy:*

$$\frac{\|\boldsymbol{w}\|_1}{\|\boldsymbol{\lambda}\|_1 + \|\boldsymbol{\kappa}\|_1} = \mathsf{opt}. \tag{79}$$

Moreover, if ω is rational, then the above ω-Shannon inequality can be chosen to be integral (Definition E.6).

Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ and a generalized variable elimination order $\overline{\boldsymbol{\sigma}} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{|\overline{\boldsymbol{\sigma}}|}) \in \overline{\pi}(\mathcal{V})$, each set of vertices $\boldsymbol{X}_i$ can be eliminated by either a join algorithm or some matrix multiplication. The concept of an ω-*query plan*, that we define below, amends a generalized variable elimination order with a mapping that specifies how to eliminate each variable.

Definition E.12 (ω-Query Plan). Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, an ω-*query plan* is a pair $(\overline{\boldsymbol{\sigma}}, e)$ where:

- $\overline{\boldsymbol{\sigma}} \in \overline{\pi}(\mathcal{V})$ is a generalized variable elimination order.

- e is a function that maps every index $i \in [|\overline{\sigma}|]$ (that satisfies $U_i^{\overline{\sigma}} \not\subseteq U_j^{\overline{\sigma}}, \forall j \in [i-1]$) to a term $e(i) \in \{h(U_i^{\overline{\sigma}})\} \cup \text{args}(\mathsf{EMM}_i^{\overline{\sigma}})$.

Note that, for a given hypergraph $\mathcal{H}$, the number of ω-query plans is finite.

Proof of Theorem 7.1. Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be the query hypergraph. Consider the ω-submodular width of $\mathcal{H}$ written in the form of Eq. (33). For each $i \in [\mathcal{I}]$, the corresponding LP (49) has an optimal objective value that is upper bounded by $\omega\text{-subw}(\mathcal{H})$. Consider a fixed LP (49), written in the form of Eq. (78), and let opt be its optimal objective value. By Lemma E.11, there exists an ω-Shannon inequality of the form (54) that satisfies the following (in addition to other properties stated in the lemma):

$$\frac{\|\boldsymbol{w}\|_1}{\|\boldsymbol{\lambda}\|_1 + \|\boldsymbol{\kappa}\|_1} = \text{opt} \le \omega\text{-subw}(\mathcal{H}).$$

Note that for every atom $R(\boldsymbol{Z})$, we have $\deg_R(\boldsymbol{Z}|\emptyset) \le N$. Define obj $\stackrel{\text{def}}{=}$ opt $\cdot \log_2 N$. Then by Theorem E.10, we can, in time $\tilde{O}(2^{\text{obj}}) = \tilde{O}(N^{\omega\text{-subw}(\mathcal{H})})$, compute tables $P_\ell(\boldsymbol{U}_\ell)$ for $\ell \in [L]$ and tables $(S_j(\boldsymbol{X}_j\boldsymbol{G}_j), T_j(\boldsymbol{Y}_j\boldsymbol{G}_j), W_j(\boldsymbol{Z}_j\boldsymbol{G}_j))$ for $j \in [J]$ that satisfy the conditions stated in the theorem. We apply this for every LP (49) that we obtain from Eq. (33).

The following claim basically says that at the end of the above process, we would have computed all inputs to a number of ω-query plans such that every tuple $\boldsymbol{t}$ in the join $\Join_{R(\boldsymbol{Z}) \in \text{atoms}(Q)} R(\boldsymbol{Z})$ is accounted for by at least one query plan. The number of ω-query plans only depends on $\mathcal{H}$, hence, it is a constant in data complexity.

Claim E.1. *At the end of the above process, for each tuple $\boldsymbol{t} \in \Join_{R(\boldsymbol{Z}) \in \text{atoms}(Q)} R(\boldsymbol{Z})$, there must exist a generalized variable elimination order $\overline{\sigma} \in \overline{\pi}(\mathcal{V})$, such that for every $i \in [|\overline{\sigma}|]$ that satisfies $U_i^{\overline{\sigma}} \not\subseteq U_j^{\overline{\sigma}}, \forall j \in [i-1]$:*

(a) either there exists a table $P(U_i^{\overline{\sigma}})$ of size $\tilde{O}(N^{\omega\text{-subw}(\mathcal{H})})$ where $\pi_{U_i^{\overline{\sigma}}}(\boldsymbol{t}) \in P$,

(b) or there exists a term $\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G}) \in \text{args}(\mathsf{EMM}_i^{\overline{\sigma}})$ along with all the following:

(b1) Tables $S_1(\boldsymbol{XG}), T_1(\boldsymbol{YG}), W_1(\boldsymbol{ZG})$ of size $\tilde{O}(N^{\omega\text{-subw}(\mathcal{H})})$ such that $\pi_{\boldsymbol{XG}}(\boldsymbol{t}) \in S_1$, $\pi_{\boldsymbol{YG}}(\boldsymbol{t}) \in T_1$ and $\pi_{\boldsymbol{ZG}}(\boldsymbol{t}) \in W_1$ and an ω-dominant triple $(\alpha_1, \beta_1, \zeta_1)$ satisfying:

$$|\pi_{\boldsymbol{G}}(S_1)| \cdot \left(\deg_{S_1}(\boldsymbol{X}|\boldsymbol{G})\right)^{\alpha_1} \cdot \left(\deg_{T_1}(\boldsymbol{Y}|\boldsymbol{G})\right)^{\beta_1} \cdot \left(\deg_{W_1}(\boldsymbol{Z}|\boldsymbol{G})\right)^{\zeta_1} = \tilde{O}(N^{\omega\text{-subw}(\mathcal{H})}) \tag{80}$$

(b2) And tables $S_2(\boldsymbol{XG}), T_2(\boldsymbol{YG}), W_2(\boldsymbol{ZG})$ of size $\tilde{O}(N^{\omega\text{-subw}(\mathcal{H})})$ such that $\pi_{\boldsymbol{XG}}(\boldsymbol{t}) \in S_2$, $\pi_{\boldsymbol{YG}}(\boldsymbol{t}) \in T_2$ and $\pi_{\boldsymbol{ZG}}(\boldsymbol{t}) \in W_2$ and an ω-dominant triple $(\alpha_2, \beta_2, \zeta_2)$ satisfying:

$$|\pi_{\boldsymbol{G}}(S_2)| \cdot \left(\deg_{S_2}(\boldsymbol{X}|\boldsymbol{G})\right)^{\alpha_2} \cdot \left(\deg_{T_2}(\boldsymbol{Y}|\boldsymbol{G})\right)^{\zeta_2} \cdot \left(\deg_{W_2}(\boldsymbol{Z}|\boldsymbol{G})\right)^{\beta_2} = \tilde{O}(N^{\omega\text{-subw}(\mathcal{H})}) \tag{81}$$

(b3) And tables $S_3(\boldsymbol{XG}), T_3(\boldsymbol{YG}), W_3(\boldsymbol{ZG})$ of size $\tilde{O}(N^{\omega\text{-subw}(\mathcal{H})})$ such that $\pi_{\boldsymbol{XG}}(\boldsymbol{t}) \in S_3$, $\pi_{\boldsymbol{YG}}(\boldsymbol{t}) \in T_3$ and $\pi_{\boldsymbol{ZG}}(\boldsymbol{t}) \in W_3$ and an ω-dominant triple $(\alpha_3, \beta_3, \zeta_3)$ satisfying:

$$|\pi_{\boldsymbol{G}}(S_3)| \cdot \left(\deg_{S_3}(\boldsymbol{X}|\boldsymbol{G})\right)^{\zeta_3} \cdot \left(\deg_{T_3}(\boldsymbol{Y}|\boldsymbol{G})\right)^{\alpha_3} \cdot \left(\deg_{W_3}(\boldsymbol{Z}|\boldsymbol{G})\right)^{\beta_3} = \tilde{O}(N^{\omega\text{-subw}(\mathcal{H})}) \tag{82}$$

Note that Case (b) in the above claim implies that the matrix multiplication corresponding to the term $\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G}) \in \text{args}(\mathsf{EMM}_i^{\overline{\sigma}})$ can be done in time $\tilde{O}(N^{\omega\text{-subw}(\mathcal{H})})$. In particular, if we take the tables S_i, T_i, W_i for $i \in [3]$ from the claim, and define

$$S \stackrel{\text{def}}{=} S_1 \cap S_2 \cap S_3, \qquad T \stackrel{\text{def}}{=} T_1 \cap T_2 \cap T_3, \qquad W \stackrel{\text{def}}{=} W_1 \cap W_2 \cap W_3, \tag{83}$$

then, Eq. (80), (81), and (82) imply that

$$\begin{aligned}|\pi_G(S)| \cdot \max\Big(&\big(\deg_S(X|G)\big)^{\alpha_1} \cdot \big(\deg_T(Y|G)\big)^{\beta_1} \cdot \big(\deg_W(Z|G)\big)^{\zeta_1}, \\ &\big(\deg_S(X|G)\big)^{\alpha_2} \cdot \big(\deg_T(Y|G)\big)^{\zeta_2} \cdot \big(\deg_W(Z|G)\big)^{\beta_2}, \\ &\big(\deg_S(X|G)\big)^{\zeta_3} \cdot \big(\deg_T(Y|G)\big)^{\alpha_3} \cdot \big(\deg_W(Z|G)\big)^{\beta_3}\Big) \qquad =\tilde{O}(N^{\omega\text{-subw}(\mathcal{H})})\end{aligned}$$

However, using the same reasoning from the proof of Proposition E.2, the LHS above is lower bounded by the LHS below:

$$\begin{aligned}|\pi_G(S)| \cdot \max\Big(&\deg_S(X|G) \cdot \deg_T(Y|G) \cdot \big(\deg_W(Z|G)\big)^{\gamma}, \\ &\deg_S(X|G) \cdot \big(\deg_T(Y|G)\big)^{\gamma} \cdot \deg_W(Z|G), \\ &\big(\deg_S(X|G)\big)^{\gamma} \cdot \deg_T(Y|G) \cdot \deg_W(Z|G)\Big) \qquad =\tilde{O}(N^{\omega\text{-subw}(\mathcal{H})}) \qquad (84)\end{aligned}$$

In particular, the claim implies that we can, in time $\tilde{O}(N^{\omega\text{-subw}(\mathcal{H})})$, go over all tuples $\boldsymbol{g} \in \pi_G(S)$ and for each $\boldsymbol{g}$, perform a matrix multiplication over two matrices whose three dimensions are $\deg_S(X|G = \boldsymbol{g})$, $\deg_T(Y|G = \boldsymbol{g})$, and $\deg_W(Z|G = \boldsymbol{g})$.

Before proving the above claim, we show how to use it to prove Theorem 7.1. The remaining steps of the algorithm are as follows: We go over all ω-query plans. Consider a fixed ω-query plan $(\overline{\boldsymbol{\sigma}}, e)$, where $\overline{\boldsymbol{\sigma}} = (X_1, \ldots, X_{|\overline{\boldsymbol{\sigma}}|})$. Initially, each hyperedge $Z \in \mathcal{E}$ has a corresponding atom $R(Z) \in \text{atoms}(Q)$. We use the generalized variable elimination order $\overline{\boldsymbol{\sigma}}$ to eliminate the variable sets $X_1, \ldots, X_{|\overline{\boldsymbol{\sigma}}|}$ and generate the corresponding hypergraph sequence $\mathcal{H}_1^{\overline{\boldsymbol{\sigma}}}, \ldots, \mathcal{H}_{|\overline{\boldsymbol{\sigma}}|}^{\overline{\boldsymbol{\sigma}}}$, as described in Definition 4.1. Each time we eliminate a variable set X_i, we remove adjacent hyperedges $\partial_i^{\overline{\boldsymbol{\sigma}}}$ and add a new hyperedge $U_i^{\overline{\boldsymbol{\sigma}}} \setminus X_i$. Our target below is to create a new corresponding atom $R(U_i^{\overline{\boldsymbol{\sigma}}} \setminus X_i)$ in time $\tilde{O}(N^{\omega\text{-subw}(\mathcal{H})})$. To that end, for each $i \in [|\overline{\boldsymbol{\sigma}}|]$ in order, we do the following:

- If $U_i^{\overline{\boldsymbol{\sigma}}} \not\subseteq U_j^{\overline{\boldsymbol{\sigma}}}, \forall j \in [i-1]$, we recognize two cases:
  - If $e(i) = h(U_i^{\overline{\boldsymbol{\sigma}}})$, we compute the new atom $R(U_i^{\overline{\boldsymbol{\sigma}}} \setminus X_i)$ by semi-join reducing the table $P(U_i^{\overline{\boldsymbol{\sigma}}})$ from Case (a) of Claim E.1 with atoms $R(Z)$ corresponding to the hyperedges $Z \in \partial_i^{\overline{\boldsymbol{\sigma}}}$ (Definition 4.1), and then projecting X_i away:

    $$R(U_i^{\overline{\boldsymbol{\sigma}}} \setminus X_i) \coloneq P(U_i^{\overline{\boldsymbol{\sigma}}}) \wedge \bigwedge_{Z \in \partial_i^{\overline{\boldsymbol{\sigma}}}} R(Z)$$

    Computing $R(U_i^{\overline{\boldsymbol{\sigma}}} \setminus X_i)$ above takes time proportional to the size of $P(U_i^{\overline{\boldsymbol{\sigma}}})$, which is $\tilde{O}(N^{\omega\text{-subw}(\mathcal{H})})$. By induction, the query Q before eliminating X_i has the same answer as the following query resulting from eliminating X_i:

    $$Q'() \coloneq R(U_i^{\overline{\boldsymbol{\sigma}}} \setminus X_i) \wedge \bigwedge_{\substack{R(Z) \in \text{atoms}(Q) \\ Z \notin \partial_i^{\overline{\boldsymbol{\sigma}}}}} R(Z) \qquad (85)$$

  - If $e(i) = \mathsf{MM}(Y; Z; X_i|G)$ where $\mathsf{MM}(Y; Z; X_i|G) \in \text{args}(\mathsf{EMM}_i^{\overline{\boldsymbol{\sigma}}})$, then by Eq. (22), there must exist $\mathcal{A}, \mathcal{B} \subseteq \partial_i^{\overline{\boldsymbol{\sigma}}}$ such that $\mathcal{A} \cup \mathcal{B} = \partial_i^{\overline{\boldsymbol{\sigma}}}$ where $A \overset{\text{def}}{=} \cup\mathcal{A}$ and $B \overset{\text{def}}{=} \cup\mathcal{B}$ satisfy the following:

    $$\begin{aligned} X_i &\subseteq A \cap B, & (A \cap B) \setminus X_i \subseteq G &\subseteq (A \cup B) \setminus X_i \\ Y &= (A \setminus B) \setminus G, & Z &= (B \setminus A) \setminus G \end{aligned}$$

Consider the tables $S(\boldsymbol{X}_i\boldsymbol{G}), T(\boldsymbol{Y}\boldsymbol{G})$ and $W(\boldsymbol{Z}\boldsymbol{G})$ defined by Eq. (83) from Claim E.1. We compute two relations:

$$M_1(\boldsymbol{G}\boldsymbol{Y}\boldsymbol{X}_i) \coloneq S(\boldsymbol{X}_i\boldsymbol{G}) \wedge T(\boldsymbol{Y}\boldsymbol{G}) \wedge \bigwedge_{\boldsymbol{Z}' \in \mathcal{A}} R(\boldsymbol{Z}'),$$
$$M_2(\boldsymbol{G}\boldsymbol{X}_i\boldsymbol{Z}) \coloneq S(\boldsymbol{X}_i\boldsymbol{G}) \wedge W(\boldsymbol{Z}\boldsymbol{G}) \wedge \bigwedge_{\boldsymbol{Z}' \in \mathcal{B}} R(\boldsymbol{Z}').$$

Now for each $\boldsymbol{g} \in \pi_{\boldsymbol{G}}(S)$, we view $\sigma_{\boldsymbol{G}=\boldsymbol{g}}(M_1)$ and $\sigma_{\boldsymbol{G}=\boldsymbol{g}}(M_2)$ as two matrices of dimensions $\boldsymbol{Y} \times \boldsymbol{X}_i$ and $\boldsymbol{X}_i \times \boldsymbol{Z}$ respectively and perform a fast matrix multiplication. By combining the resulting matrices for all $\boldsymbol{g}$, we obtain the new atom $R(U_i^{\overline{\boldsymbol{\sigma}}} \setminus \boldsymbol{X}_i) = R(\boldsymbol{G}\boldsymbol{Y}\boldsymbol{Z})$. By Eq. (84), this takes time $\tilde{O}(N^{\omega\text{-subw}(\mathcal{H})})$. Moreover, by induction, the query Q before eliminating $\boldsymbol{X}_i$ has the same answer as the query Q' from Eq. (85) that results from eliminating $\boldsymbol{X}_i$.

- If $U_i^{\overline{\boldsymbol{\sigma}}} \subseteq U_j^{\overline{\boldsymbol{\sigma}}}$ for some $j \in [i-1]$, then $U_i^{\overline{\boldsymbol{\sigma}}} \subseteq U_j^{\overline{\boldsymbol{\sigma}}} \setminus \boldsymbol{X}_j$. Hence, we can obtain $R(U_i^{\overline{\boldsymbol{\sigma}}} \setminus \boldsymbol{X}_i)$ by projecting $R(U_j^{\overline{\boldsymbol{\sigma}}} \setminus \boldsymbol{X}_j)$.

Finally, we prove Claim E.1.

Proof of Claim E.1. For sake of contradiction, suppose that there exists a tuple $\boldsymbol{t} \in \Join_{R(\boldsymbol{Z}) \in \text{atoms}(Q)} R(\boldsymbol{Z})$ where for every generalized elimination order $\overline{\boldsymbol{\sigma}}$, there exists an index in $[|\overline{\boldsymbol{\sigma}}|]$, let's call this index $f^*(\overline{\boldsymbol{\sigma}})$, such that:

- We don't have a table $P(U_{f^*(\overline{\boldsymbol{\sigma}})}^{\overline{\boldsymbol{\sigma}}})$ that contains the projection of $\boldsymbol{t}$.
- And, for every term $\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G}) \in \text{args}(\mathsf{EMM}_{f^*(\overline{\boldsymbol{\sigma}})}^{\overline{\boldsymbol{\sigma}}})$, either one of conditions (b1), (b2), and (b3) of the claim is violated. We map each one of the three conditions (b1), (b2), and (b3) to one of the three terms in $\text{args}(\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G}))$ from Eq. (21) in order. Let $g^*(\overline{\boldsymbol{\sigma}}, \mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G}))$ be one term in $\text{args}(\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G}))$ corresponding to one condition from (b1), (b2), or (b3) that is violated. (If more than one condition is violated, we pick any one arbitrarily.)

But now consider Eq. (32). Specifically, fix $f \stackrel{\text{def}}{=} f^*$ and $g_f \stackrel{\text{def}}{=} g^*$ and consider the corresponding inner LP. By Lemma E.11 and Theorem E.10, there must exist some $\overline{\boldsymbol{\sigma}}$ such that:

- Either there exists a table $P(U_{f^*(\overline{\boldsymbol{\sigma}})}^{\overline{\boldsymbol{\sigma}}})$ containing the projection of $\boldsymbol{t}$.
- Or there exists a term $\mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G}) \in \text{args}(\mathsf{EMM}_{f^*(\overline{\boldsymbol{\sigma}})}^{\overline{\boldsymbol{\sigma}}})$ and three tables $S_i(\boldsymbol{X}\boldsymbol{G})$, $T_i(\boldsymbol{Y}\boldsymbol{G})$, $W_i(\boldsymbol{Z}\boldsymbol{G})$ containing the projection of $\boldsymbol{t}$ and satisfying one of the three conditions (b1), (b2), or (b3), specifically the one corresponding to the term $g^*(\overline{\boldsymbol{\sigma}}, \mathsf{MM}(\boldsymbol{X};\boldsymbol{Y};\boldsymbol{Z}|\boldsymbol{G}))$, which we had assumed to be violated.

This is a contradiction. ◻

◻