Symmetric Parameterised Holants on Hypergraphs: Towards a Classification for Parameterised VCSPs

Panagiotis Aivasiliotis
Hasso Plattner Institute, University of Potsdam, Germany

Andreas Göbel
Hasso Plattner Institute, University of Potsdam, Germany

Marc Roth
School of Electronic Engineering and Computer Science, Queen Mary University of London, UK

Abstract

We study the complexity of the parameterised counting constraint satisfaction problem: given a set of constraints over a set of variables and a positive integer k, how many ways are there to assign k variables to 1 (and the others to 0) such that all constraints are satisfied. While this problem, and its decision version, received significant attention during the last two decades, existing work has so far exclusively focused on restricted settings such as finding and counting homomorphisms between relational structures due to Grohe (JACM 2007) and Dalmau and Jonsson (TCS 2004), or the case of finite constraint languages due to Creignou and Vollmer (SAT 2012), and Bulatov and Marx (SICOMP 2014).

In this work, we tackle a more general setting of parameterised (counting) valued constraint satisfaction problems (VCSPs) with infinite constraint languages: we allow our constraints to be chosen from an infinite set of permitted constraints and we allow our constraints to map an assignment of its variables not only to True or False, but to arbitrary values. In this setting we are able to model and classify significantly more general problems such as (weighted) parameterised factor problems on hypergraphs and counting weight-k solutions of systems of linear equations, none of which are captured by existing complexity classifications of parameterised constraint satisfaction problems.

On a formal level, we express parameterised VCSPs as parameterised *holant problems* on uniform hypergraphs, and we establish complete and explicit complexity dichotomy theorems for this family of problems both w.r.t. classical complexity theory (P vs. #P) and parameterised complexity (FPT vs. #W[1]). For resolving the P vs. #P question, we mainly rely on the use of hypergraph gadgets, the existence of which we prove using properties of degree sequences necessary for realisability in uniform hypergraphs. As a technical highlight, we also employ Curticapean's "CFI Filters" (SODA 2024) – named after the Cai-Fürer-Immermann construction for bounding the expressiveness of the Weisfeiler-Leman heuristic – to establish polynomial-time algorithms for isolating vectors in the homomorphism basis of some of our holant problems. For the FPT vs. #W[1] question, we build upon the recently established combinatorial toolkit for parameterised holants on the special case of graphs by Aivasiliotis et al. (ICALP 2025) and also rely on an extension of the framework of the homomorphism basis due to Curticapean, Dell and Marx (STOC 17) to uniform hypergraphs.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases Parameterised Complexity, Counting Problems, Constraint Satisfaction Problems, Holant Problems

Digital Object Identifier 10.4230/LIPIcs.ICALP.2026.9

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2508.19794 [2]

Funding *Andreas Göbel*: Funded by the Postdoc Network Brandenburg.

53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026).
Editors: Sayan Bhattacharya, Danupon Nanongkai, Michael Benedikt, and Gabriele Puppis;
Article No. 9; pp. 9:1–9:15

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Extended Abstract

Constraint satisfaction problems ("CSPs") and their counting analogues ("#CSPs") belong to the most well-studied computational problems in algorithms and complexity theory: given a set of constraints over a set of variables, find one of (or count) the assignments of variables to elements of a finite set D such that all constraints are satisfied. CSPs and #CSPs are extremely rich in expressibility, allowing us not only to model a multitude of fundamental problems such as the Boolean satisfiability problem, the graph homomorphism problem, and the computation of partition functions, but they also provide a uniform language for studying and analysing seemingly very different problems (see e.g. Bulatov's survey on CSPs [6]). At the same time, families of (#)CSPs are robust enough in structure to often allow for a *precise and exhaustive* complexity classification of their members, which usually comes in form of a *complexity dichotomy theorem* stating that each member of a family of (#)CSPs is either solvable in polynomial time, or NP-hard.[1]

One of the earliest examples of a comprehensive analysis of a CSP is Schaefer's Dichotomy [41], stating that each member of the generalised satisfiability problem is either solvable in polynomial time or NP-complete. Over the course of the following 20 years a significant amount of research had been undertaken with the goal of lifting Schaefer's result from the Boolean domain to arbitrary finite domains, culminating in the formulation of the Feder-Vardi-Conjecture in 1998 [27]. It then took almost another 20 years until the conjecture was proved independently by Bulatov [5] and Zhuk [44] in 2017.

The quest for counting constraint satisfaction problems (#CSPs) has seen similar success: Creignou and Hermann [16] established a counting version of Schaefer's dichotomy in 1996, and the classification for #CSPs over arbitrary domains was established again by Bulatov [4] whose result was then simplified and its tractability criterion shown to be decidable by Dyer and Richerby [26]. Moreover, the classification was eventually extended and strenghthened to #CSPs with complex weights by Cai and Chen [8]. In the present work, we study the version of #CSP in which we only count "weight-k-solutions", that is assignments that set precisely k variables to 1; this version is also known as *parameterised* #CSP and we focus on the Boolean domain, i.e., #CSPs defined over $D = \{0, 1\}$.

1.1 Parameterising CSPs by Solution Size

The *weighted satisfiability problem* WSAT asks, given a Boolean formula F and a positive integer k, whether F contains a satisfying assignment of Hamming weight k, that is, whether it is possible to satisfy the formula by setting precisely k variables to 1. Similarly, its counting version #WSAT asks to compute the number of such assignments. WSAT and #WSAT are the foundational problems for the complexity theory of parameterised decision and counting problems: depending on the allowed structure of the input formula[2], they define, respectively, the W and the #W-hierarchies, including the classes W[1] and #W[1], which can be considered the parameterised decision and counting equivalents of NP. We refer the reader to the standard textbook of Flum and Grohe [29] for a comprehensive introduction to structural parameterised complexity theory.

Notably, WSAT and #WSAT constitute the most rudimentary instances of parameterised CSP and #CSP, and it comes to no surprise that analogues of Schaefer's dichotomy, both for decision and for counting, have already been established for those problems [38, 7, 17].

[1] This property is rather surprising, given that Ladner's Theorem states that, assuming P $\neq$ NP, the class NP contains an infinite hierarchy of complexities between P and NPC [35].

[2] The "W" in the W-hierarchy stands for "weft", which describes a structural property of boolean circuits.

The key difference in the parameterised setting is that the notion of tractability is relaxed from polynomial-time algorithms to *fixed-parameter tractable* "FPT" algorithms, the requirement for which is a running time bound of $f(k) \cdot n^{O(1)}$, where n is the input size and k is the problem parameter – in this work, k will always be the Hamming weight of the sought solutions. The function f can be any computable function; in other words, we allow for a super-polynomial overhead only in the problem parameter, reflecting the assumption that the parameter is significantly smaller than the input size in problem instances of interest (see [7] for a discussion on this assumption in the context of parameterised CSPs).

When studying families of parameterised CSPs, the goal is, thus, to classify the problems into instances that are FPT and instances that are hard for a class in the W (or #W) hierarchy. In particular, it is well-known that under the *exponential time hypothesis* (ETH) [31, 32], which asserts that 3-SAT cannot be solved in time $\mathsf{exp}(o(n))$ (where n is the number of variables of the input formula), (#)W[1]-hard problems are not in FPT [14, 15, 22]. In addition to the aforementioned analogues of Schaefer's dichotomy, such classifications have also been achieved for the problems of finding and counting homomorphisms between relational structures, which can both be expressed as parameterised (#)CSPs (see [30, 23]).

However, all existing results are limited by either focusing on a special case (such as simple, undirected graph homomorphisms) or by the restriction to finite constraint languages. For example, the following natural counting problems cannot be expressed in existing frameworks for parameterised #CSPs:[3]

- Let p be a prime. The problem #CodeWord$_p$ gets as input a matrix A over $\mathbb{Z}/p\mathbb{Z}$ and a positive integer k, and the goal is to compute the number of solutions to the system of linear equations $A\vec{x} = \vec{0}$ (also over $\mathbb{Z}/p\mathbb{Z}$) that set precisely k variables to 1 (and the other variables to 0). Note that #CodeWord$_2$ is identical to the counting version of Exact-Even-Set (see e.g. [25]).
- Let $S \subseteq \mathbb{N}$. The problem #Factor(S) gets as input a hypergraph H and a positive integer k, and the goal is to compute the number of k-edge-subsets of H that induce a hypergraph in which the degree of every vertex is in S. Given $d \in \mathbb{N}$, we denote #Factor$_d(S)$ for the restriction to d-uniform hypergraphs. For example, #Factor$_d(\{0,1\})$ is identical to the problem of counting k-matchings in d-uniform hypergraphs, where a matching in a hypergraph is a set of pairwise disjoint hyperedges.

In this work, we introduce and study a version of parameterised #CSP which is powerful enough in expressibility to model the previous problems – in fact, we will consider a much more general model which can also describe constraints that map assignments to arbitrary numbers (rather than to just True or False). Moreover, we will be able to provide complete complexity classifications for our model which reveal the complexity of the aforementioned examples as straightforward special cases.

Parameterised Symmetric Valued #CSPs and Hypergraph Holants

For reasons of notational consistency with existing work on (parameterised) CSPs (see, e.g., [33]) we will first introduce our setting of interest via so-called *Valued* CSPs ("VCSPs").[4]

[3] We note that special cases of #CodeWord$_p$ and of #Factor(S), such as restrictions to certain input matrices and to graphs, are expressible in existing frameworks such as in [1]. However, the complexity of both problems has not been fully resolved yet.

[4] Being aware of another work (see [34]) studying a problem under the name Valued CSPs, we would like to point out that their problem is incomparable to ours and that our definition complies with [33].

Let $\mathcal{F}$ denote a finite set of symmetric functions $f : \{0,1\}^* \to \mathbb{C}$; by symmetric we mean that f is invariant under permutation of entries of input vectors.

▶ Remark 1. Note that, a symmetric function that is not associated with a specific arity, i.e., the length of its input is arbitrary, depends only on the Hamming weight of its input. Hence, such functions can be equivalently defined over natural numbers.

An instance $\mathcal{I}$ of parameterised symmetric P-VCSP($\mathcal{F}$) consists of a positive integer k, a set X of n *variables* $x_1, \ldots, x_n$ and a set C of m *constraints* of the form $\langle f, (x_{i_1}, ..., x_{i_r})\rangle$ for some $r \in \mathbb{N}$, where $f \in \mathcal{F}$. The vector $(x_{i_1}, \ldots, x_{i_r})$ is known as the *scope* of the constraint. Given an instance $\mathcal{I} = (X, C, k)$, the task is to compute

$$Z(\mathcal{I}) = \sum_{\substack{\alpha: X \to \{0,1\} \\ \sum_{x \in X} \alpha(x) = k}} \prod_{\langle f, (x_{i_1}, \ldots, x_{i_r})\rangle \in C} f(\alpha(x_{i_1}), \ldots, \alpha(x_{i_r}))\,.$$

For example, if $\mathcal{F}$ contains precisely the function $\vec{x} \mapsto \sum_{x \in \vec{x}} x \mod p$, then P-VCSP($\mathcal{F}$) is identical to #CodeWord$_p$. If $\mathcal{F}$ contains precisely the function $\mathsf{hw}_{\leq 1}$ which maps a vector $\vec{x} \in \{0,1\}^*$ to 1 if it has at most one 1 entry, and to 0 otherwise, then P-VCSP($\mathcal{F}$) is identical to #Factor($\{0,1\}$).

Our goal is to understand the complexity of #P-VCSP($\mathcal{F}$) for any set of symmetric constraint functions $\mathcal{F}$ - recall that symmetric constraints only depend on the Hamming weight of their assignment. To this end, we introduce and rely on a generalisation of the parameterised holant framework [19, 1] from graphs to hypergraphs, which enables us to naturally model VCSPs in a way that allows us to leverage a variety of powerful tools that have been established in parameterised counting complexity theory over the past years. We proceed by formally introducing the holant framework, after which we provide details on the equivalence between VCSPs and parameterised holants on hypergraphs.

Holant problems, implicitly defined by Valiant in his seminal paper on *holographic reductions* [43] and formally introduced by Cai, Lu and Xia [13], yield an avenue for modeling various counting problems, including problems that cannot be formulated as a #CSP problem, such as the generating function of perfect matchings in a graph (see [10, 33]). Similarly to VCSPs, holant problems assume oracle access to a finite collection of functions, which by convention we call *signatures* and denote them with s instead of f.

An instance of a holant problem with respect to a finite set $\mathcal{S}$ of signatures is a *signature grid* that consists of a (simple) undirected graph G and an assignment of signatures from $\mathcal{S}$ to the vertices of G. For a vertex $v \in V(G)$, we denote by s_v, the signature that is assigned to v. The objective is to compute the holant value (or simply, the *holant*) of the signature grid $(G, \{s_v\}_{v \in V(G)})$ which is given as follows

$$\mathsf{Holant}(G, \{s_v\}_{v \in V(G)}) = \sum_{\alpha: E(G) \to \{0,1\}} \prod_{v \in V(G)} s_v(\alpha|_{E_v(G)})\,, \tag{1}$$

where $\alpha|_{E_v(G)}$ is the restriction of α to the edges that are incident to v. The focus of this article is on the following holant variant.

The *parameterised holant problem* on a finite set $\mathcal{S}$ of symmetric signatures $s : \{0,1\}^* \to \mathbb{C}$, takes as input a signature grid $\Omega = (G, \{s_v\}_{v \in V(G)})$, and a number $k \in \mathbb{Z}_{\geq 0}$ and computes

$$\mathsf{Holant}(\Omega, k) = \sum_{\substack{\alpha: E(G) \to \{0,1\} \\ \sum_e \alpha(e) = k}} \prod_{v \in V(G)} s_v(\alpha|_{E_G(v)})\,,$$

▶ Remark 2. Following the equivalent definition of symmetric signatures $s : \{0,1\}^* \to \mathbb{C}$ as $s : \mathbb{N} \to \mathbb{C}$, we note that $\mathsf{Holant}(\Omega, k)$ can be written in a simplified form by observing that for any assignment $\alpha : E(G) \to \{0,1\}$ s.t. $\sum_e \alpha(e) = k$ and any vertex $v \in V(G)$, by letting $A = \{e \in E(G) \mid \alpha(e) = 1\}$, we have that the number of 1s in $\alpha|_{E_G(v)}$ is precisely equal to $|A \cap E_G(v)|$. Hence,

$$\mathsf{Holant}(\Omega, k) = \sum_{\substack{A \subseteq E(G) \\ |A|=k}} \prod_{v \in V(G)} s_v(|A \cap E_G(v)|)\,.$$

The integer k is called the *parameter* of the problem instance, and, in addition to classifying the polynomial-time tractable cases, we are interested in *fixed-parameter tractable* (FPT) algorithms. Holant instances can naturally be extended to signature grids over *hypergraphs* (V, E), where V is the set of vertices and $E \subseteq 2^V$ is the *multiset* of *hyperedges*[5], that generalise graphs. The holant (value) of a signature grid over an underlying hypergraph is defined identically to (1), where a hyperedge e is incident to a vertex v if $v \in e$.

Thus far, parameterised holant problems may take as input a signature grid with an underlying hypergraph of arbitrarily large rank[6]. Naturally, we are also interested in the case in which the rank of the input hypergraph is *bounded*. There are two ways by which we may *bound* the rank of the instance's underlying hypergraph, namely, either to include the rank in the parametrisation along with k, or to allow only for hypergraphs of fixed rank, that is, their rank is at most some *constant* d (the latter of which is in fact the *slice* of the former). In particular, the most interesting variant considering fixed rank is the one that is restricted to *uniform* hypergraphs. The three aforementioned variants are formally stated below[7] – we emphasize that we assume signature grids over hypergraphs that feature *no* multiple hyperedges for everything that follows.

▶ **Definition 3** (Parameterised Holant Problems on Hypergraphs). *Let $\mathcal{S}$ be a finite set of signatures.*

- *The problem* p-$\textsc{Holant}^{\mathsf{UBD}}(\mathcal{S})$ *expects as input a positive integer k and a signature grid $\Omega = (G, \{s_v\}_{v \in V(G)})$ over $\mathcal{S}$. The output is $\mathsf{Holant}(\Omega, k)$ and the problem parameter is k.*
- *The problem* p-$\textsc{Holant}(\mathcal{S})$ *expects as input a positive integer k and a signature grid $\Omega = (G, \{s_v\}_{v \in V(G)})$ over $\mathcal{S}$. The output is $\mathsf{Holant}(\Omega, k)$ and the problem parameters are k and the rank of G.*
- *For any constant $d \in \mathbb{Z}_{\geq 2}$, the problem* p-$\textsc{Holant}^{d}_{\mathsf{uni}}(\mathcal{S})$ *expects as input a positive integer k and a signature grid $\Omega = (G, \{s_v\}_{v \in V(G)})$ over $\mathcal{S}$, such that G is d-uniform, that is, each hyperedge has cardinality d. The output is $\mathsf{Holant}(\Omega, k)$ and the problem parameter is k.*

▶ Remark 4 (On infinite domains). We emphasise that our definition of parameterised holants on hypergraphs is, up to the extension from graphs to hypergraphs, *identical* to the definition in existing literature [1, 19]. As discussed in [1], computable signatures with infinite domains – though unusual in classical holant theory – yield a non-trivial well-defined framework

[5] Note that a hyperedge is a set and *not* a multiset, but multiple copies of the same hyperedge are allowed.
[6] Recall that the rank of a hypergraph G is given by $\max_{e \in E(G)} |e|$.
[7] We point out that parameterised holant problems p-$\textsc{Holant}$ have been traditionally considered in a *colourful* setting in which edges are coloured, while uncoloured holant problems are typically denoted as p-$\textsc{UnColHolant}$ (see e.g., [19], [1]). In this work, we only consider the uncoloured setting, thus we refrain from indicating in the notation of our holant problems that are uncoloured.

that is able to capture problems that would otherwise require an infinite number of (finite-arity) signatures (see, e.g., #CodeWord$_p$ and #Factor(S) discussed earlier). Even more importantly, finite-arity signatures can only capture those instances for which the underlying hypergraph is of bounded vertex-degree. In the parameterised setting, such a restriction above is known to admit trivial FPT-algorithms (a detailed discussion about the latter point is formally provided in [1] as well as briefly in Section 1.7). Thereby a study on infinite domains is crucial for establishing the complexity of important families of problems that remained uncaptured by existing frameworks.

▶ Remark 5 (On Boolean domain and symmetric signatures). In general, holant problems allow for general domains (see, e.g., [9]) and signatures that do not need to be symmetric [36]. However, even in the classical setting, only partial results are known for symmetric functions on general domains [12] or for asymmetric functions on Boolean domains [39, 40]. Therefore, this work focusses on symmetric Boolean signatures not because the asymmetric case is less interesting, but as a natural starting point for investigating parameterised holants on hypergraphs.

1.2 Equivalence of VCSPs and Holant Problems on Hypergraphs

We show in a very simple reduction, that holant problems on hypergraphs are VCSPs (and vice versa). In particular, here we show the equivalence for the respective parameterised problems. We stress that for the aforementioned equivalence to hold, we need to modify our definition of holant problems such that the underlying hypergraphs allow for multi-hyperedges, i.e., multiple copies of the same hyperedge may appear. However, this is only a technical subtlety that we do not need to consider in the rest of the paper. The reason is that all of our hardness results for the *restricted* holant problems transfer trivially to the more general holant problems that allows for multi-hyperedges and as we show in the full version (see [2]) our tractability results still apply as well.

We let $\mathcal{F}$ be a finite set of functions $f : \{0,1\}^r \to \mathbb{C}$ and recall that, given a set X of n variables $x_1, \ldots, x_n$, a set C of m constraints of the form $\langle f, (x_{i_1}, \ldots, x_{i_r})\rangle$, where $f \in \mathcal{F}$, and a positive integer k, the problem p-VCSP($\mathcal{F}$) computes

$$Z(X, C, k) = \sum_{\substack{\alpha: X \to \{0,1\} \\ \sum_e \alpha(e) = k}} \prod_{\langle f, (x_{i_1}, \ldots, x_{i_r})\rangle \in C} f(\alpha(x_{i_1}), \ldots, \alpha(x_{i_r}))\,.$$

Given $\mathcal{I} = (X, C, k)$, we construct a hypergraph $G(\mathcal{I})$, that will be used as the underlying hypergraph of the intended holant instance, as follows. Each constraint $c \in C$ corresponds to a (unique) vertex $v_c \in G(\mathcal{I})$ and each variable $x \in X$ corresponds to a (unique) hyperedge $e_x \in E(G(\mathcal{I}))$ given as $\{v_c \mid x \text{ is in the scope of } c\}$. Note that, there may be variables $x \neq x'$ such that $e_x = e_{x'}$. We allow $G(\mathcal{I})$ to contain multiple copies of the same hyperedge, such that each variable x corresponds uniquely to (some copy of) e_x.

Next, we let $\Omega(\mathcal{I})$ denote the signature grid obtained from $G(\mathcal{I})$ by equipping, for each $c = (f, (x_{i_1}, \ldots, x_{i_r})) \in C$, the variable v_c with the signature $s_{v_c} = f$. We also assume that the order of the arguments of f is preserved, that is, if x is the j-th argument of f, then e_x is the j-th argument of s_{v_c}. It can be readily verified that $Z(\mathcal{I}) = \mathsf{Holant}(\Omega(\mathcal{I}), k)$.

▶ Remark 6. For $x \in X$, we write $\mathsf{occ}_{\mathcal{I}}(x)$ for the *occurence* of x in $\mathcal{I}$, that is, the number of constraints $\langle f, (x_{i_1}, \ldots, x_{i_r})\rangle \in C$ that contain x in their scope, that is, $x = x_{i_j}$, for some $1 \leq j \leq r$. For any variable $x \in X$, we have that $\mathsf{occ}_{\mathcal{I}}(x)$ is equal to the size of e_x. Furthermore, for any constraint $c \in C$, we have that $|c|$, that is, the number of variables in the scope of c, is equal to the number of hyperedges in $G(\mathcal{I})$ that are incident to v_c.

Furthermore, the reverse direction, that is, formulating a (parameterised) generalised holant problem as a VSCP can be achieved in a similar fashion.

1.3 Existing Results for Parameterised Holants on Graphs

The parameterised complexity of the problem p-Holant$^2_{\text{uni}}(\mathcal{S})$ was recently classified by Aivasiliotis et al. [1]. The classification criterion depends on the *type* of the set of allowed signatures, as defined in [1], which we also define here momentarily, for reasons of self-containment. In particular, only signature sets that exclude signatures s with $s(0) = 0$ are assigned a type in [1] and the reason is that – roughly speaking – signatures s with $s(0) = 0$ do not affect the *parameterised* complexity of the problem, as it was shown in [1]. In other words, any fixed-parameter tractable finite set of signatures remains fixed-parameter tractable even if we add to it any number of signatures s with $s(0) = 0$. However, we will see that this is not necessarily true for polynomial-time tractability, which is why we explicitly include the case of $s(0) = 0$ in the present paper.

The type of a signature set is decided by the *signature fingerprint* function, which is a combinatorial sum that is reminiscent of the Möbius function of partition lattices, given as follows.

▶ **Definition 7** (Signature Fingerprints [1]). *Let a be a positive integer and let s be a signature. If $s(0) \neq 0$ then the* fingerprint *of a and s is defined as*

$$\chi(a, s) := \sum_{\sigma} (-1)^{|\sigma|-1}(|\sigma| - 1)! \cdot \prod_{B \in \sigma} \frac{s(|B|)}{s(0)},$$

where the sum is over all partitions σ of $[a]$ and $|\sigma|$ denotes the number of blocks $B \in \sigma$. For technical reasons, if $s(x) = 0$ for all x, we set $\chi(a, s) = 0$. In all remaining cases, i.e., $s(0) = 0$ but $s(x) \neq 0$ for some $x > 0$, the fingerprint is undefined and we set $\chi(a, s) = \perp$.

▶ **Definition 8** (Types of Signature Sets [1]). *Let $\mathcal{S}$ be a finite set of signatures. We say that $\mathcal{S}$ is of type*[8]

1. $\mathbb{T}[1]$ *if $\chi(a, s) = 0$ for all $s \in \mathcal{S}, a \geq 2$,*
2. $\mathbb{T}[2]$ *if $\chi(a, s) = 0$ for all $s \in \mathcal{S}, a \geq 3$, but there exists $s \in \mathcal{S}$ with $\chi(2, s) \neq 0$, and*
3. $\mathbb{T}[\infty]$ *otherwise, i.e., there exists $s \in \mathcal{S}$ and $a \geq 3$ such that $\chi(a, s) \neq 0$.*

It was shown in [1] that there are infinitely many signature sets of each type. We are now ready to state the known complexity dichotomy for p-Holant$^2_{\text{uni}}(\mathcal{S})$. In particular, it is only the signature sets of the first type that render the problem tractable. However, the third type of signature sets allow for tighter conditional lower bounds.

▶ **Theorem 9** (Complexity Dichotomy for p-Holant$^2_{\text{uni}}(\mathcal{S})$ [1]). *Let $\mathcal{S}$ be a finite set of signatures. We set $\mathcal{S}_0 = \{s \in \mathcal{S} \mid s(0) = 0\}$.*

1. *If $\mathcal{S} \setminus \mathcal{S}_0$ is of type $\mathbb{T}[1]$, then p-Holant$^2_{\text{uni}}(\mathcal{S})$ can be solved in FPT-near-linear time.*
2. *Otherwise p-Holant$^2_{\text{uni}}(\mathcal{S})$ is #W[1]-complete. If, additionally, $\mathcal{S} \setminus \mathcal{S}_0$ is of type $\mathbb{T}[\infty]$, then p-Holant$^2_{\text{uni}}(\mathcal{S})$ cannot be solved in time $f(k) \cdot |V(\Omega)|^{o(k/\log k)}$, unless ETH fails.* ⌟

Finally, the tractability criteria above are in fact explicit, in the sense that for a set $\mathcal{S}$ of signatures of type $\mathbb{T}[1]$ and a signature $s \in \mathcal{S}$, we know that s has the following explicit representation: $s(n) = \alpha^n$, for some $\alpha \in \mathbb{C}$. The above implies that $s(0) = 1$ must be assumed, but in the context of holant problems it is well-known that scaling a signature by some constant, does not affect the complexity of the problem.

[8] We point out that we have renamed the types of signatures to avoid introducing further concepts.

1.4 Further Related Work on Holants

In this part, we explore some well established approaches that appear, at first glance, to allow for deriving complexity classifications for holants on hypergraphs; however, as we argue below, they turn out to be insufficient for our (parameterised and classical) settings.

Firstly, as it has already been pointed out, holant problems on hypergraphs are equivalent to a natural class of symmetric counting constraint satisfaction problems (see Section 1.2). Furthermore, readers familiar with classical holant theory might be aware that the extension of signature grids from graphs to hypergraphs has only limited benefits since classifications for higher arity #CSPs had already been known when the holant framework was introduced – e.g. the dichotomy for #CSP [4] first appeared in ICALP'08 while for holants [11] in STOC'13. However, in the parameterised world where we only consider assignments that map precisely k variables/edges to 1, the situation is flipped: not much is known about the complexity of #CSP for higher arities and the framework of parameterised holants on hypergraphs allows us to tackle such problems.

Secondly, holants on hypergraphs are also known to reduce, in the classical sense, to holants on graphs via incidence graphs; we provide the details for the sake of self-containment: To this end, we let $\mathcal{S}$ be a finite set of signatures. We consider a hypergraph G and an assignment $\{s_v\}_{v\in V(G)}$ of signatures from $\mathcal{S}$ to its vertices. It is folklore that any hypergraph can be represented as a bipartite incidence graph B as follows. We let $V(B) = L \dot\cup R$ and consider two arbitrary bijections $\pi_L : V(G) \to L$ and $\pi_R : E(G) \to R$. Furthermore, for each $v \in V(G)$ and each $e \in E(G)$ such that $v \in e$, we add $\{\pi_L(v), \pi_R(e)\}$ to $E(B)$. Then, we consider the following assignment $\{s'_v\}_{v\in V(B)}$ of signatures to $V(B)$: for each $v \in L$ we assign to v the signature of $\pi_L^{-1}(v)$ and for each $u \in R$, we assign to u the signature $=_\ell$: $\{0,1\}^\ell \to \{0,1\}$ where $\ell = |\pi_R^{-1}(e)|$ and $=_\ell$ evaluates to one if and only if all of its arguments are equal. It can be readily verified that

$$\mathsf{Holant}(G, \{s_v\}_{v\in V(G)}) = \mathsf{Holant}(B, \{s'_v\}_{v\in V(B)}), \tag{2}$$

where s'_v is either in $\mathcal{S}$ or is an equality signature $=_\ell$ for some $\ell \in \mathbb{N}$. In other words, holants on hypergraphs reduce to holants on (bipartite) graphs assuming – besides $\mathcal{S}$ – freely available equality signatures.

It is known that signatures for equality (i.e., $=_\ell$) are part of the tractable family of affine signatures, implying the easiness of holant problems on hypergraphs. Moreover, hardness of holants on graphs transfer directly to holant problems on hypergraphs since the former is a special case of the latter.

However, we point out that the aforementioned trivial equivalence already becomes incomplete when introducing even mild natural restrictions to the problem, e.g., restricting holants to uniform hypergraphs of fixed rank (i.e., of upper-bounded maximum hyperedge-size), or, more importantly, by not allowing freely available equality signatures, which cases are considered and remedied in this work. Note that the latter case is realised when we only consider signatures of infinite domain which are our main consideration. It is easy to see that such signatures cannot model equality signatures.

1.5 Our Contributions

We provide exhaustive complexity dichotomy results both with respect to classical complexity theory (P vs. #P) and with respect to parameterised complexity theory (FPT vs. #W[1]).

We first state our main results in the parameterised setting. As it has been already pointed out, the problem p-$\textsc{Holant}^d_{\mathsf{uni}}(\mathcal{S})$ is a "slice" of p-$\textsc{Holant}(\mathcal{S})$. The above implies that we need not state two complexity dichotomies, since an FPT algorithm for the latter

problem implies an FPT algorithm for the former (w.r.t. the respective parametrisation) and if any slice of p-Holant($\mathcal{S}$) is hard, then p-Holant($\mathcal{S}$) must already be hard. Hence, our main result – stated formally below – consists of

1. an FPT algorithm for p-Holant($\mathcal{S}$), for finite sets of signatures $\mathcal{S}$ of type $\mathbb{T}[1]$, which also addresses the tractable case in which $\mathcal{S}$ contains signatures s with $s(0) = 0$.
2. #W[1]-hardness results for the problem p-Holant$^d_{\mathsf{uni}}(\mathcal{S})$, for any $d \geq 2$.

We recall that the parameterised complexity in the special case of p-Holant$^2_{\mathsf{uni}}(\mathcal{S})$, that is, the case of graphs, has already been established in [1].

▶ **Theorem 10** (Parameterised Complexity Dichotomy). *Let $\mathcal{S}$ be a finite set of signatures. Let furthermore $\mathcal{S}_0 = \{s \in \mathcal{S} \mid s(0) = 0\}$.*

1. *If $\mathcal{S} \setminus \mathcal{S}_0$ is of type $\mathbb{T}[1]$, then p-Holant($\mathcal{S}$) can be solved in FPT-(near)-linear time, that is, in time $f(k, \mathsf{r}) \cdot \tilde{O}(|\Omega|)$, where r is the rank of the underlying hypergraph of Ω.*
2. *If $\mathcal{S} \setminus \mathcal{S}_0$ is not of type $\mathbb{T}[1]$, then p-Holant($\mathcal{S}$) is #W[1]-hard.*
   In particular, for any $d \in \mathbb{Z}_{\geq 2}$, the problem p-Holant$^d_{\mathsf{uni}}(\mathcal{S})$ is #W[1]-hard. Furthermore, if $\mathcal{S} \setminus \mathcal{S}_0$ is of type $\mathbb{T}[\infty]$, then p-Holant$^d_{\mathsf{uni}}(\mathcal{S})$ cannot be solved in time $f(k) \cdot |V(\Omega)|^{o(k/\log(k))}$, for any computable function f, unless ETH fails. ⌟

Note that the previous result implies that all FPT cases are in fact FPT-near-linear time solvable. We also classify the aforementioned problems in the non-parameterised setting. We point out that such a classification is novel *even* for the case of graphs (that is, $d = 2$).

▶ **Theorem 11** (Classical Complexity Dichotomy). *Let $\mathcal{S}$ be a finite set of signatures and let $d \in \mathbb{Z}_{\geq 2}$.*

1. *If $\mathcal{S}$ is of type $\mathbb{T}[1]$ (implying that all signatures $s \in \mathcal{S}$ satisfy $s(0) \neq 0$), then p-Holant$^d_{\mathsf{uni}}(\mathcal{S})$ can be solved in polynomial time.*
2. *Otherwise, p-Holant$^d_{\mathsf{uni}}(\mathcal{S})$ is #P-hard.* ⌟

Note that the second part (2.) in the previous result implies that p-Holant$^d_{\mathsf{uni}}(\mathcal{S})$ is #P-hard whenever $\mathcal{S}$ contains a signature s with $s(0) = 0$ but $s(x) \neq 0$ for some $x > 0$.

▶ Remark 12 (Finite signature sets vs. finite constraint languages). Finite signature sets are not the same as finite constraint languages, since our signatures have domain $\{0,1\}^*$, but constraints come with a fixed arity.

As we explain in Section 1.2 our theorems immediately imply the analogous classifications for (the classical and parameterised) complexity of p-VCSP. We now give two applications of our main results:

▶ **Corollary 13.** *For each prime p, the problem #Codeword$_p$ is #P-hard and #W[1]-hard. Moreover, assuming ETH, #Codeword$_p$ cannot be solved in time $f(k) \cdot |A|^{o(k/\log k)}$ for any function f, where A is the input matrix and $k \in \mathbb{N}$ is the problem parameter.*

Proof. The problem #Codeword$_p$ reduces from p-Holant($\{s_p\}$) where $s_p(x) = 1$ if $x \equiv 0 \mod p$ and $s_p(x) = 0$ otherwise. To verify the existence of this reduction, observe that for each vertex v of the signature grid and its incident edges $e_1, \ldots, e_\ell$, we just need to add the equation $\sum_{i=1}^{\ell} e_i \equiv 0 \mod p$; in fact, this reduction applies in both directions if we allow multiple copies of the same hyperedge in the signature grid (this happens if the matrix of the system has two identical columns).

Clearly, $s_p(0) \neq 0$. Now consider $\chi(p, s_p)$. There is only one partition of $[p]$ that contains a block of size p: this is the coarsest partition $\top$ containing only one block $[p]$. Clearly, $s_p(|[p]|) = s_p(p) = 1$. All further partitions of $[p]$ contain a block B with $1 \leq |B| \leq p-1$, hence $s(|B|) = 0$. Consequently,

$$\chi(p, s_p) = (-1)^{|\top|-1}(|\top| - 1)! \cdot 1 \neq 0\,.$$

For $p > 2$, this shows that $\{s_p\}$ is of type $\mathbb{T}[\infty]$. For $p = 2$, it is easy to verify that $\chi(3, s_p) \neq 0$, hence the type is also $\mathbb{T}[\infty]$. Intractability then immediately follows from Theorems 10 and 11, and from the fact that p-Holant$^d_{\mathsf{uni}}(\mathcal{S})$ reduces to p-Holant$(\mathcal{S})$ for any d and $\mathcal{S}$. ◀

For the next application, we consider the problem of counting k-matchings in d-uniform hypergraphs. It is well-known that this problem is #P-hard since it subsumes as a special case the problem of counting perfect matchings in d-uniform hypergraphs for which #P-hardness is known (in particular, the case $d = 2$ was shown by Valiant [42], while the case $d > 2$ was shown later by Creignou [16]). For $d = 2$, it took almost a decade to strengthen #P-hardness to #W[1]-hardness.[9] We obtain #W[1]-hardness for all $d \geq 2$ as an easy application of our main result.

▶ **Corollary 14.** *For each $d \in \mathbb{Z}_{\geq 2}$, the problem of counting k-matchings in a d-uniform hypergraph G is #W[1]-hard and cannot be solved in time $f(k) \cdot |G|^{o(k/\log k)}$ for any function f, unless ETH fails.*

Proof. We just need to show that the signature set $\{\mathsf{hw}_{\leq 1}\}$ is of type $\mathbb{T}[\infty]$. Here, $\mathsf{hw}_{\leq 1}(x) = 1$ if $x \leq 1$ and $\mathsf{hw}_{\leq 1}(x) = 0$ otherwise. The claim then follows immediately from Theorem 10, since the problem is identical to p-Holant$^d_{\mathsf{uni}}(\{\mathsf{hw}_{\leq 1}\})$. It is easy to verify that $\chi(3, \mathsf{hw}_{\leq 1}) \neq 0$. Moreover, $\mathsf{hw}_{\leq 1}(0) \neq 0$. Hence $\{\mathsf{hw}_{\leq 1}\}$ is of type $\mathbb{T}[\infty]$. ◀

In particular, Corollary 14 also follows from the novel complete complexity classification of #Factor$_d(S)$ which we establish below.

▶ **Corollary 15.** *Let $S \subseteq \mathbb{Z}_{\geq 0}$ be a finite set. For any $d \in \mathbb{Z}_{\geq 2}$,*

1. #Factor$_d(S)$ *is* #P*-hard.*
2. *If $0 \notin S$,* #Factor$_d(S)$ *is solvable in FPT-(near)-linear time, that is, in time $f(k) \cdot \tilde{O}(|G|)$, for some computable function f.*
3. *If $0 \in S$,* #Factor$_d(S)$ *is* #W[1]*-hard and cannot be solved in time $f(k) \cdot |V(G)|^{o(k/\log(k))}$, for any computable function f, unless ETH fails.*

Proof. Let (G, k) be an instance of #Factor$_d(S)$. Consider the signature $t_S : \mathbb{Z}_{\geq 0} \to \{0, 1\}$ where $t_S(x) = 1$ if and only if $x \in S$. It is easy to verify that #Factor$_d(S)$ is equivalent[10] to p-Holant$^d_{\mathsf{uni}}(\{t_S\})$ via a reduction that associates to (G, k) a holant instance (Ω, k), where the underlying hypergraph of Ω is G and each vertex of Ω is equipped with signature t_S.

First, note that if $0 \notin S$ (implying that $t_S(0) = 0$), it follows from Theorem 10 that #Factor$_d(S)$ is solvable in FPT-(near)-linear time. Furthermore, Theorem 11 implies that for any $d \in \mathbb{Z}_{\geq 2}$, #Factor$_d(S)$ is #P-hard.

Now, assume that $0 \in S$. It has been shown in (the full version of) [1, Corollary 6.22] that t_S is of type $\mathbb{T}[\infty]$. Hence, Theorem 10 implies that for any $d \in \mathbb{Z}_{\geq 2}$, #Factor$_d(S)$ is #W[1]-hard and cannot be solved in time $f(k) \cdot |V(G)|^{o(k/\log(k))}$, for any computable function f, unless ETH fails. Furthermore, Theorem 11 implies that for any $d \in \mathbb{Z}_{\geq 2}$, #Factor$_d(S)$ is #P-hard. ◀

[9] The #W[1]-hardness of counting k-matchings in graphs was first conjectured by Flum and Grohe in 2004 [28], and proved by Curticapean in 2013 [18].

[10] Clearly, with respect to parameterised as well as non-parameterised polynomial-time Turing-reductions.

1.6 Our Techniques

Before we conclude our introduction, we give a brief exposition of the machinery we are employing for deriving our results.

1.6.1 Gadgets

One of the tools that we employ are *gadgets*. Gadgets have mainly been used in counting problems outside of the holant framework. We point out that our gadgets will be much more general than what the familiar reader might know as *matchgates* which have been extensively used within the holant framework, the details of which are beyond the scope of this paper.

Roughly speaking, gadgets are signature grids Ω_F that are *embedded* on other signature grids Ω_G, typically by identifying vertices of (a copy of) Ω_F with vertices of Ω_G and possibly forming new hyperedges (e.g., between different copies of gadgets). The above implies that gadgets induce a mapping between signature grids (possibly allowing different signatures). In this work, the construction of gadgets is based on the realisability of *degree sequences* of certain suitable hypergraphs that also meet some structural properties. For example, some of our gadgets rely on the existence of infinitely many d-uniform b-regular hypergraphs for $d, b \geq 2$ that are also connected.

For our parameterised holant problems, we will consider a more general mapping f that maps a parameterised holant instance (Ω, k) (instead of just a signature grid) into another $(\Omega', k') = f(\Omega, k)$, that can be computed efficiently with respect to parameterised or classical complexity. The choice of the new parameter k' is based on the following observation: the most useful gadgets are naturally those that preserve the holant value, that is, $\mathsf{Holant}(\Omega', k') = \mathsf{Holant}(\Omega, k)$, hence the need for possibly choosing a different parameter. However, the previous requirement is rather strict, and we would also be content if the holant values of the respective holant instances are *efficiently comparable*, in the sense that there is an efficient way to compute $\mathsf{Holant}(\Omega, k)$ by computing only $\mathsf{Holant}(\Omega', k')$. For our problems, we devise gadgets such that

$$\mathsf{Holant}(\Omega', k') = \alpha(\Omega, k) \cdot \mathsf{Holant}(\Omega, k) + \beta(\Omega, k)\,,$$

where α, β are efficiently computable functions, which implies that we can efficiently compute $\mathsf{Holant}(\Omega, k)$ via computing $\mathsf{Holant}(\Omega', k')$. More formally, we are interested in those gadgets that induce a mapping f which will imply a polynomial time (or FPT) Turing-reduction between two holant problems (that may not necessarily allow for the same signatures).

1.6.2 Graph Motif Parameters

Another tool in our machinery is the framework of *graph motif parameters* and the property of *complexity monotonicity* that they enjoy, introduced by Curticapean, Dell and Marx [21].

A *graph parameter* is any function $f : \mathcal{G} \to \mathbb{Q}$ over graphs that is invariant under isomorphisms. A *graph motif parameter* is a graph parameter f that admits the following expansion: there exist pair-wise non-isomorphic *pattern* graphs $H_1, \ldots, H_t$ and *coefficients* $\zeta_1, \ldots, \zeta_t \in \mathbb{Q} \setminus \{0\}$ such that for any graph G, $f(G) = \sum_{i=1}^{t} \zeta_i \cdot \#\mathsf{Hom}(H_i \to G)$, where $\#\mathsf{Hom}(H \to G)$ counts the number of homomorphisms, i.e., edge-preserving mappings from $V(H)$ to $V(G)$. Is it known that such an expansion is unique [37]. We will sometimes refer to it as the expansion of f into the *homomorphism basis* implying that homomorphism counts span the vector space of graph motif parameters, the details of which are beyond the scope of this introduction.

A well-known result, first established in [24] (see also [21]), states that evaluating the number $\#\mathsf{Hom}(H \to G)$ can be done in time $\mathsf{poly}(|V(H)|) \cdot n^{\mathsf{tw}(H)+1}$, where $\mathsf{tw}(H)$ is the treewidth of H and $n = |V(G)|$. Hence, any fixed graph motif parameter can be evaluated in polynomial time, since the pattern graphs are fixed. It is therefore more interesting to investigate the complexity of evaluating a graph motif parameter f from an infinite – and assumed recursively enumerable – family $\mathcal{F}$ of graph motif parameters.

From our discussion above, it follows that if we can bound the treewidth of the pattern graphs featured in *any* graph motif parameter $f \in \mathcal{F}$, then the problem of evaluating any $f \in \mathcal{F}$ is computable in polynomial time. From the perspective of parameterised complexity, the result above further implies quite straightforwardly that the problem of evaluating $f \in \mathcal{F}$, when parameterised by the maximum number of vertices of the pattern graphs in f, is fixed-parameter tractable when the treewidth of pattern graphs in any $f \in \mathcal{F}$ is bounded. In a remarkable result due to Curticapean, Dell and Marx [21] it was shown that the boundedness of the treewidth is the right dichotomy criterion for the aforementioned parameterised problem, in the sense that, if $\mathcal{F}$ contains graph motif parameters with pattern graphs of unbounded treewidth, then the problem is #W[1]-hard. Specifically, it is shown in [21] that, if we can evaluate a graph motif parameter f, then we can also evaluate the mapping $\#\mathsf{Hom}(H \to \star) : G \mapsto \#\mathsf{Hom}(H \to G)$, via FPT Turing-reductions, for any pattern graph H of f as long as its corresponding coefficient is non-zero. This property was coined as *complexity monotonicity* and since its inception, it has yielded an avenue for charting the complexity of many counting problems that can be formulated within the framework of graph motif parameters. In particular, the problem of evaluating a graph motif parameter $f \in \mathcal{F}$ then becomes a combinatorial problem – arguably of very challenging nature – , since one has to investigate which graphs are supported in the expansion of f in the homomorphism basis, i.e., their coefficient is non-zero.

Interestingly, Curticapean showed very recently that – under some technical restrictions – a polynomial-time analogue of complexity monotonicity is feasible, that is, extracting homomorphism counts from the homomorphism expansion of a graph motif parameter can be done in polynomial time [20]. If the technical conditions apply, this result can be used to infer #P-hardness for the problem of evaluating graph motif parameters.

The property of complexity monotonicity also applies to graph motif parameters over more general structures, e.g., hypergraphs that may also feature vertex-colourings and it is otherwise known as *Dedekind interpolation* (see e.g., the full version of [3, Theorem 18]).

In particular, signature grids can be equivalently seen as vertex-coloured hypergraphs where the colours are given by the signatures. Letting $\mathcal{G}(\mathcal{S})$ denote the set of all signature grids over $\mathcal{S}$, we can show that for any $k \in \mathbb{Z}_{\geq 0}$, the mapping $f : \Omega \in \mathcal{G}(\mathcal{S}) \to \mathsf{Holant}(\Omega, k)$ is a graph motif parameter (see [2] for a proof) and thus we can formulate and study holant problems within the framework of graph motif parameters.

1.7 Conclusion and Future Work

Following the recent advancements of Aivasiliotis et al. [1], in the line of work for charting the complexity of parameterised holant problems introduced by Curticapean [19], we introduce generalised parameterised holant problems that can model parameterised VCSPs. The latter can be seen as the counting counterpart (that assumes complex values) of the parameterised satisfiability problem introduced and classified by Marx in [38], the parametrisation of which is given by the solution-size which is restricted to some $k \in \mathbb{Z}_{\geq 0}$.

We fully classify the complexity of parameterised VCSPs (as well as the complexity of the non-parameterised equivalent problems) for unbounded symmetric constraints. In particular, several natural variants are also considered subject to additional restrictions, e.g., when no two variables can appear in the same set of constraints and/or variables appear in exactly d constraints, that is the *occurrence* of every variable is precisely d.

As mentioned above, in our setting we assumed unbounded arity (of constraints) and in particular, we further assumed bounded occurrence of (variables). For the rest of this section, we would like to elaborate on the other possible configurations of the problem with respect to the (un)boundedness of the arity and the occurrence.

First, in the context of unbounded arity, we show that trivially tractable instances of the problem already become W[2]-hard if we assume unbounded occurrences (see [2]). In this context, an FPT Turing-reduction from parameterised VCSPs to the problem of evaluating graph motif parameters would imply the collapse of W-hierarchy, since the expansion of instances of the latter problem into the homomorphism basis includes patterns, the size of which is not bounded by the (only) parameter k. This shows that a novel framework becomes necessary for tackling those instances of parameterised #CSPs, which is an exciting avenue for future work.

Next, in the context of bounded arity, if we further assume that occurrence is bounded then we can infer trivial fixed-parameter tractability results (for all instances) which follow from standard algorithmic techniques in parameterised complexity such as the bounded search-tree paradigm.

The last case, that is the combination of bounded arity and unbounded occurrence, is not remedied in this work as it is orthogonal to our main setting of interest. Hence we leave this family of parameterised #CSPs open for investigation in future work.

References

1 Panagiotis Aivasiliotis, Andreas Göbel, Marc Roth, and Johannes Schmitt. Parameterised Holant Problems. In *ICALP 2025*, volume 334, pages 7:1–7:14, 2025. Full version : https://arxiv.org/abs/2409.13579. doi:10.4230/LIPIcs.ICALP.2025.7.

2 Panagiotis Aivasiliotis, Andreas Göbel, and Marc Roth. Symmetric parameterised holants on hypergraphs: Towards a classification for parameterised vcsps, 2026. arXiv:2508.19794.

3 Marco Bressan, Matthias Lanzinger, and Marc Roth. The complexity of pattern counting in directed graphs, parameterised by the outdegree. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pages 542–552, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3564246.3585204.

4 Andrei A. Bulatov. The complexity of the counting constraint satisfaction problem. *J. ACM*, 60(5):34:1–34:41, 2013. doi:10.1145/2528400.

5 Andrei A. Bulatov. A Dichotomy Theorem for Nonuniform CSPs. In *Proc. FOCS 2017*, pages 319–330, 2017. doi:10.1109/FOCS.2017.37.

6 Andrei A. Bulatov. Constraint satisfaction problems: complexity and algorithms. *ACM SIGLOG News*, 5(4):4–24, November 2018. doi:10.1145/3292048.3292050.

7 Andrei A Bulatov and Dániel Marx. Constraint satisfaction parameterized by solution size. *SIAM Journal on Computing*, 43(2):573–616, 2014. doi:10.1137/120882160.

8 Jin-Yi Cai and Xi Chen. Complexity of counting CSP with complex weights. *J. ACM*, 64(3):19:1–19:39, 2017. doi:10.1145/2822891.

9 Jin-Yi Cai and Artem Govorov. The complexity of counting edge colorings for simple graphs. *Theor. Comput. Sci.*, 889:14–24, 2021. doi:10.1016/J.TCS.2021.07.033.

10 Jin-Yi Cai and Artem Govorov. Perfect matchings, rank of connection tensors and graph homomorphisms. *Comb. Probab. Comput.*, 31(2):268–303, 2022. doi:10.1017/S0963548321000286.

11 Jin-Yi Cai, Heng Guo, and Tyson Williams. A complete dichotomy rises from the capture of vanishing signatures. *SIAM J. Comput.*, 45(5):1671–1728, 2016. doi:10.1137/15M1049798.

12 Jin-Yi Cai and Jin Soo Ihm. Holant* Dichotomy on Domain Size 3: A Geometric Perspective. In *52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025)*, volume 334 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 148:1–148:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.ICALP.2025.148.

13 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holant problems and counting csp. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 715–724, 2009. doi:10.1145/1536414.1536511.

14 Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David W. Juedes, Iyad A. Kanj, and Ge Xia. Tight lower bounds for certain parameterized NP-hard problems. *Inf. Comput.*, 201(2):216–231, 2005. doi:10.1016/j.ic.2005.05.001.

15 Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.*, 72(8):1346–1367, 2006. doi:10.1016/j.jcss.2006.04.007.

16 Nadia Creignou and Miki Hermann. Complexity of generalized satisfiability counting problems. *Information and computation*, 125(1):1–12, 1996. doi:10.1006/INCO.1996.0016.

17 Nadia Creignou and Heribert Vollmer. Parameterized complexity of weighted satisfiability problems: Decision, enumeration, counting. *Fundam. Informaticae*, 136(4):297–316, 2015. doi:10.3233/FI-2015-1159.

18 Radu Curticapean. Counting matchings of size k is #W[1]-hard. In *Proc. of ICALP*, volume 7965, pages 352–363, 2013. doi:10.1007/978-3-642-39206-1_30.

19 Radu Curticapean. *The simple, little and slow things count: On parameterized counting complexity.* PhD thesis, Saarland University, 2015. URL: http://scidok.sulb.uni-saarland.de/volltexte/2015/6217/.

20 Radu Curticapean. Count on CFI graphs for #P-hardness. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1854–1871. SIAM, 2024. doi:10.1137/1.9781611977912.74.

21 Radu Curticapean, Holger Dell, and Dániel Marx. Homomorphisms are a good basis for counting small subgraphs. In *Proc. of ACM STOC*, pages 210–223, 2017. doi:10.1145/3055399.3055502.

22 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

23 Víctor Dalmau and Peter Jonsson. The complexity of counting homomorphisms seen from the other side. *Theoretical Computer Science*, 329(1-3):315–323, 2004. doi:10.1016/J.TCS.2004.08.008.

24 Josep Díaz, Maria J. Serna, and Dimitrios M. Thilikos. Counting H-colorings of partial k-trees. *Theor. Comput. Sci.*, 281(1-2):291–309, 2002. doi:10.1016/S0304-3975(02)00017-8.

25 Rodney G. Downey, Michael R. Fellows, Alexander Vardy, and Geoff Whittle. The parametrized complexity of some fundamental problems in coding theory. *SIAM J. Comput.*, 29(2):545–570, 1999. doi:10.1137/S0097539797323571.

26 Martin E. Dyer and David Richerby. An effective dichotomy for the counting constraint satisfaction problem. *SIAM J. Comput.*, 42(3):1245–1274, 2013. doi:10.1137/100811258.

27 Tomás Feder and Moshe Y. Vardi. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM J. Comput.*, 28(1):57–104, 1998. doi:10.1137/S0097539794266766.

28 Jörg Flum and Martin Grohe. The Parameterized Complexity of Counting Problems. *SIAM J. Comput.*, 33(4):892–922, 2004. doi:10.1137/S0097539703427203.

29 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer, 2006. doi:10.1007/3-540-29953-X.

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

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

32 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which Problems Have Strongly Exponential Complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

33 Mark Jerrum. Counting constraint satisfaction problems. *Dagstuhl Follow-Ups*, 2017. doi:10.4230/DFU.VOL7.15301.8.

34 Vladimir Kolmogorov, Andrei Krokhin, and Michal Rolínek. The complexity of general-valued csps. *SIAM Journal on Computing*, 46(3):1087–1110, 2017. doi:10.1137/16M1091836.

35 Richard E. Ladner. On the structure of polynomial time reducibility. *J. ACM*, 22(1):155–171, 1975. doi:10.1145/321864.321877.

36 Jiabao Lin and Hanpin Wang. The complexity of holant problems over boolean domain with non-negative weights. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 29:1–29:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.29.

37 László Lovász. *Large Networks and Graph Limits*, volume 60 of *Colloquium Publications*. American Mathematical Society, 2012. URL: http://www.ams.org/bookstore-getitem/item=COLL-60.

38 Dániel Marx. Parameterized complexity of constraint satisfaction problems. *Comput. Complex.*, 14(2):153–183, 2005. doi:10.1007/S00037-005-0195-9.

39 Boning Meng, Juqiu Wang, and Mingji Xia. The $\mathrm{FP}^{\mathrm{NP}}$ versus #p dichotomy for #eo. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 1795–1806, 2025.

40 Boning Meng, Juqiu Wang, Mingji Xia, and Jiayi Zheng. From an odd arity signature to a holant dichotomy. In Srikanth Srinivasan, editor, *40th Computational Complexity Conference, CCC 2025, Toronto, Canada, August 5-8, 2025*, volume 339 of *LIPIcs*, pages 23:1–23:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.CCC.2025.23.

41 Thomas J Schaefer. The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 216–226, 1978. doi:10.1145/800133.804350.

42 Leslie G Valiant. The complexity of enumeration and reliability problems. *siam Journal on Computing*, 8(3):410–421, 1979. doi:10.1137/0208032.

43 Leslie G. Valiant. Holographic Algorithms. *SIAM J. Comput.*, 37(5):1565–1594, 2008. doi:10.1137/070682575.

44 Dmitriy Zhuk. A Proof of CSP Dichotomy Conjecture. In *Proc. FOCS 2017*, pages 331–342, 2017. doi:10.1109/FOCS.2017.38.