# On the Hardness of Compressing Weights

**Bart M. P. Jansen** ✉ 🏠 ⓘ
Eindhoven University of Technology, The Netherlands

**Shivesh K. Roy** ✉ 🏠 ⓘ
Eindhoven University of Technology, The Netherlands

**Michał Włodarczyk** ✉ 🏠 ⓘ
Eindhoven University of Technology, The Netherlands

──── **Abstract** ────

We investigate computational problems involving large weights through the lens of kernelization, which is a framework of polynomial-time preprocessing aimed at compressing the instance size. Our main focus is the weighted CLIQUE problem, where we are given an edge-weighted graph and the goal is to detect a clique of total weight equal to a prescribed value. We show that the weighted variant, parameterized by the number of vertices $n$, is significantly harder than the unweighted problem by presenting an $\mathcal{O}(n^{3-\varepsilon})$ lower bound on the size of the kernel, under the assumption that NP $\not\subseteq$ coNP/poly. This lower bound is essentially tight: we show that we can reduce the problem to the case with weights bounded by $2^{\mathcal{O}(n)}$, which yields a randomized kernel of $\mathcal{O}(n^3)$ bits.

We generalize these results to the weighted $d$-UNIFORM HYPERCLIQUE problem, SUBSET SUM, and weighted variants of Boolean CONSTRAINT SATISFACTION PROBLEMS (CSPs). We also study weighted minimization problems and show that weight compression is easier when we only want to preserve the collection of optimal solutions. Namely, we show that for node-weighted VERTEX COVER on bipartite graphs it is possible to maintain the set of optimal solutions using integer weights from the range $[1, n]$, but if we want to maintain the ordering of the weights of all inclusion-minimal solutions, then weights as large as $2^{\Omega(n)}$ are necessary.

## 1 Introduction

A prominent class of problems in algorithmic graph theory consist of finding a subgraph with certain properties in an input graph $G$, if one exists. Some variations of this problem can be solved in polynomial time (detecting a triangle), while the general problem is NP-complete since it generalizes the CLIQUE problem. In recent years, there has been an increasing interest in understanding the complexity of such subgraph detection problems in *weighted* graphs, where either the vertices or the edges are assigned integral weight values, and the goal is either to find a subgraph of a given form which optimizes the total weight of its elements, or alternatively, to find a subgraph whose total weight matches a prescribed value.

Incorporating weights in the problem definition can have a significant effect on computational complexity. For example, determining whether an unweighted $n$-vertex graph has a triangle can be done in time $\mathcal{O}(n^\omega)$ (where $\omega < 2.373$ is the exponent of matrix

46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021).
Editors: Filippo Bonchi and Simon J. Puglisi; Article No. 64; pp. 64:1–64:21
Leibniz International Proceedings in Informatics
LIPIcs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

multiplication) [14], while for the analogous weighted problem of finding a triangle of minimum edge-weight, no algorithm of running time $\mathcal{O}(n^{3-\varepsilon})$ is known for any $\varepsilon > 0$. Some popular conjectures in fine-grained complexity theory even postulate that no such algorithms exist [27]. Weights also have an effect on the best-possible exponential running times of algorithms solving NP-hard problems: the current-fastest algorithm for the NP-complete HAMILTONIAN CYCLE problem in undirected graphs runs in time $\mathcal{O}(1.66^n)$ [3], while for its weighted analogue, TRAVELING SALESPERSON, no algorithm with running time $\mathcal{O}((2-\varepsilon)^n)$ is known for general undirected graphs (cf. [23]).

In this work we investigate how the presence of weights in a problem formulation affects the *compressibility* and *kernelization complexity* of NP-hard problems. Kernelization is a subfield of parameterized complexity [6, 9] that investigates how much a *polynomial-time* preprocessing algorithm can compress an instance of an NP-hard problem, without changing its answer, in terms of a chosen complexity parameter.

For a motivating example of kernelization, we consider the VERTEX COVER problem. For the unweighted variant, a kernelization algorithm based on the Nemhauser-Trotter theorem [25] can efficiently reduce an instance $(G, k)$ of the decision problem, asking whether $G$ has a vertex cover of size at most $k$, to an equivalent one $(G', k')$ consisting of at most $2k$ vertices, which can therefore be encoded in $\mathcal{O}(k^2)$ bits via its adjacency matrix. In the language of parameterized complexity, the unweighted VERTEX COVER problem parameterized by the solution size $k$ admits a kernelization (self-reduction) to an equivalent instance on $\mathcal{O}(k^2)$ bits. For the *weighted* variant of the problem, where an input additionally specifies a weight threshold $t \in \mathbb{N}_+$ and a weight function $w \colon V(G) \to \mathbb{N}_+$ on the vertices, and the question is whether there is a vertex cover of size at most $k$ and weight at most $t$, the guarantee on the encoding size of the reduced instance is weaker. Etscheid et al. [10, Thm. 5] applied a powerful theorem of Frank and Tardös [12] to develop a polynomial-time algorithm to reduce any instance $(G, w, k, t)$ of WEIGHTED VERTEX COVER to an equivalent one with $\mathcal{O}(k^2)$ edges, which nevertheless needs $\mathcal{O}(k^8)$ bits to encode due to potentially large numbers occurring as vertex weights. The WEIGHTED VERTEX COVER problem, parameterized by solution size $k$, therefore has a kernel of $\mathcal{O}(k^8)$ bits.

The overhead in the kernel size for the weighted problem is purely due to potentially large weights. This led Etscheid et al. [10] to ask in their conclusion whether this overhead in the kernelization sizes of weighted problems is necessary, or whether it can be avoided. As one of the main results of this paper, we will prove a lower bound showing that the kernelization complexity of some weighted problems is strictly larger than their unweighted counterparts.

**Our results.** We consider an edge-weighted variation of the CLIQUE problem, parameterized by the number of vertices $n$:

---

EXACT-EDGE-WEIGHT CLIQUE (EEWC)
**Input:** An undirected graph $G$, a weight function $w \colon E(G) \to \mathbb{N}_0$, and a target $t \in \mathbb{N}_0$.
**Question:** Does $G$ have a clique of total edge-weight exactly $t$, i.e., a vertex set $S \subseteq V(G)$ such that $\{x, y\} \in E(G)$ for all distinct $x, y \in S$ and such that $\sum_{\{x,y\} \subseteq S} w(\{x, y\}) = t$?

---

Our formulation of EEWC does not constrain the cardinality of the clique. This formulation will be convenient for our purposes, but we remark that by adjusting the weight function it is possible to enforce that any solution clique $S$ has a prescribed cardinality. Through such a cardinality restriction we can obtain a simple reduction from the problem with potentially negative weights to equivalent instances with weights from $\mathbb{N}_0$, by increasing all weights by a suitably large value and adjusting $t$ according to the prescribed cardinality.

Note that an instance of EEWC can be reduced to an equivalent one where $G$ has all possible edges, by simply inserting each non-edge with a weight of $t + 1$. Hence the difficulty of the problem stems from achieving the given target weight $t$ as the total weight of the edges spanned by $S$, not from the requirement that $G[S]$ must be a clique.

EEWC is a natural extension of ZERO-WEIGHT TRIANGLE [1], which has been studied because it inherits fine-grained hardness from both 3-SUM [29] and ALL PAIRS SHORTEST PATHS [28, Footnote 3]. EEWC has previously been considered by Abboud et al. [2] as an intermediate problem in their W[1]-membership reduction from $k$-SUM to $k$-CLIQUE. Vassilevska-Williams and Williams [29] considered a variation of this problem with weights drawn from a finite field. The related problem of detecting a triangle of negative edge weight is central in the field of fine-grained complexity for its subcubic equivalence [30] to ALL PAIRS SHORTEST PATHS. Another example of an edge-weighted subgraph detection problem with an exact requirement on the weight of the target subgraph is EXACT-EDGE-WEIGHT PERFECT MATCHING, which can be solved using algebraic techniques [22, §6] and has been used as a subroutine in subgraph isomorphism algorithms [21, Proposition 3.1].

The unweighted version of EEWC, obtained by setting all edge weights to 1, is NP-complete because it is equivalent to the CLIQUE problem. When using the number of vertices $n$ as the complexity parameter, the problem admits a kernelization of size $\mathcal{O}(n^2)$ obtained by simply encoding the instance via its adjacency matrix. We prove the following lower bound, showing that the kernelization complexity of the edge-weighted version is a factor $n$ larger. The lower bound even holds against *generalized* kernelizations (Definition 4).

▶ **Theorem 1.** *The* EXACT-EDGE-WEIGHT CLIQUE *problem parameterized by the number of vertices $n$ does not admit a generalized kernelization of $\mathcal{O}(n^{3-\varepsilon})$ bits for any $\varepsilon > 0$, unless* NP $\subseteq$ coNP/poly.

Intuitively, the lower bound exploits the fact that the weight value of each of the $\Theta(n^2)$ edges in the instance may be a large integer requiring $\Omega(n)$ bits to encode. We also provide a randomized kernelization which matches this lower bound.

▶ **Theorem 2.** *There is a randomized polynomial-time algorithm that, given an $n$-vertex instance $(G, w, t)$ of* EXACT-EDGE-WEIGHT CLIQUE, *outputs an instance $(G', w', t')$ of bitsize $\mathcal{O}(n^3)$, in which each number is bounded by $2^{\mathcal{O}(n)}$, that is equivalent to $(G, w, t)$ with probability at least $1 - 2^{-n}$. Moreover, if the input is a YES-instance, then the output is always a YES-instance.*

The proof is based on the idea that taking the weight function modulo a random prime preserves the answer to the instance with high probability. We adapt the argument by Harnik and Naor [13] that it suffices to pick a prime of magnitude $2^{\mathcal{O}(n)}$. As a result, each weight can be encoded with just $\mathcal{O}(n)$ bits.

It is noteworthy that the algorithm above can produce only false positives, therefore instead of using randomization we can turn it into a co-nondeterministic algorithm which guesses the correct values of the random bits. The framework of cross-composition excludes not only deterministic kernelization, but also co-nondeterministic [8], thus the lower bound from Theorem 1 indeed makes the presented algorithm tight.

Together, Theorems 1 and 2 pin down the kernelization complexity of EXACT-EDGE-WEIGHT CLIQUE, and prove it to be a factor $n$ larger than for the unit-weight case. For CLIQUE, the kernelization of $\mathcal{O}(n^2)$ bits due to adjacency-matrix encoding cannot be improved to $\mathcal{O}(n^{2-\varepsilon})$ for any $\varepsilon > 0$, as was shown by Dell and van Melkebeek [8].

We extend our results to the hypergraph setting, which is defined as follows: given a $d$-regular hypergraph ($d \geq 3$) with non-negative integer weights on the hyperedges, and a target value $t$, test if there is a vertex set $S$ for which each size-$d$ subset is a hyperedge (so

that $S$ is a hyperclique) such that the sum of the weights of the hyperedges contained in $S$ is exactly $t$. By a bootstrapping reduction using Theorem 1, we prove that EXACT-EDGE-WEIGHT $d$-UNIFORM HYPERCLIQUE does not admit a generalized kernel of size $\mathcal{O}(n^{d+1-\varepsilon})$ for any $\varepsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$, while the randomized hashing technique yields a randomized kernelization of size $\mathcal{O}(n^{d+1})$.

We can view the edge-weighted ($d$-hyper)clique problem on $(G, k, w, t)$ as a weighted constraint satisfaction problem (CSP) with weights from $\mathbb{Z}$, by introducing a binary variable for each vertex, and a weighted constraint for each subset $S'$ of $d$ vertices, which is satisfied precisely when all variables for $S'$ are set to true. If $S'$ is a (hyper)edge $e \in E(G)$ then the weight of the constraint on $S'$ equals the weight of $e$; if $S'$ is not a hyperedge of $G$, then the weight of the constraint on $S'$ is set to $-\infty$ to prevent all its vertices from being simultaneously chosen. Under this definition, $G$ has a (hyper)clique of edge-weight $t$ if and only if there is an assignment to the variables for which the total weight of satisfied constraints is $t$. Via this interpretation, the lower bounds for EEWC yield lower bounds on the kernelization complexity of weighted variants of CSP. We employ a recently introduced framework [17] of reductions among different CSPs whose constraint languages have the same maximum degree $d$ of their characteristic polynomials, to transfer these lower bounds to other CSPs (see Section 3.3 for definitions). We obtain tight kernel bounds when parameterizing the exact-satisfaction-weight version of CSP by the number of variables, again using random prime numbers to obtain upper bounds. Our lower bounds for EXACT-EDGE-WEIGHT $d$-UNIFORM HYPERCLIQUE transfer to all CSPs with degree $d \geq 2$. In degree-1 CSP each constraint depends on exactly one variable, therefore its exact-weighted variant is equivalent to the SUBSET SUM problem, for which we also provide a tight lower bound.

▶ **Theorem 3.** *SUBSET SUM parameterized by the number of items $n$ does not admit a generalized kernelization of size $\mathcal{O}(n^{2-\varepsilon})$ for any $\varepsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.*

Theorem 3 tightens a result of Etscheid et al. [10, Theorem 14], who ruled out (standard) kernelizations for SUBSET SUM of size $\mathcal{O}(n^{2-\varepsilon})$ assuming the Exponential Time Hypothesis. Our reduction, conditioned on the incomparable assumption $\mathsf{NP} \nsubseteq \mathsf{coNP/poly}$, additionally rules out generalized kernelizations that compress into an instance of a potentially different problem. Note that the new lower bound implies that the input data in SUBSET SUM cannot be efficiently encoded in a more compact way, whereas the previous lower bound relies on the particular way the input is encoded in the natural formulation of the problem. On the other hand, a randomized kernel of size $\mathcal{O}(n^2)$ is known [13].

The results described so far characterize the kernelization complexity of broad classes of weighted constraint satisfaction problems in which the goal is to find a solution for which the total weight of satisfied constraints is exactly equal to a prescribed value. We also broaden our scope and investigate the maximization or minimization setting, in which the question is whether there is a solution whose cost is at least, or at most, a prescribed value. Some of our upper-bound techniques can be adapted to this setting: using a procedure by Nederlof, van Leeuwen and de Zwaan [24] a maximization problem can be reduced to a polynomial number of exact queries. This leads, for example, to a *Turing* kernelization (cf. [11]) for the weight-maximization version of $d$-UNIFORM HYPERCLIQUE which decides an instance in randomized polynomial time using queries of size $\mathcal{O}(n^{d+1})$ to an oracle for an auxiliary problem. We do not have lower bounds in the maximization regime.

In an attempt to understand the relative difficulty of obtaining an exact target weight versus maximizing the target weight, we finally investigate different models of weight reduction for the WEIGHTED VERTEX COVER problem studied extensively in earlier works [5, 10, 24]. We consider the problem on *bipartite* graphs, where an optimal solution can be found in

polynomial time, but we investigate whether a weight function can be efficiently compressed while either preserving (a) the collection of minimum-weight vertex covers, or (b) the relative ordering of total weight for all *inclusion-minimal* vertex covers. We give a polynomial-time algorithm for case (a) which reduces to a weight function with range $\{1, \ldots, n\}$ using a relation to $b$-matchings, but show that in general it is impossible to achieve (b) with a weight function with range $\{1, \ldots, 2^{o(n)}\}$, by utilizing lower bounds on the number of different threshold functions.

**Organization.** We begin with short preliminaries with the crucial definitions. We prove our main Theorem 1 in Section 3 by presenting a cross-composition of degree 3 into EXACT-EDGE-WEIGHT CLIQUE and employing it to obtain kernelization lower bounds for $d$-uniform hypergraphs for $d \geq 2$. This section also contains the kernelization lower bound for SUBSET SUM as well as the generalization of these results to Boolean CSPs. Next, in Section 4 we focus on bipartite WEIGHTED VERTEX COVER and the difficulty of compressing weight functions. The proofs of statements marked with ($\bigstar$) are located in the appendix. The proofs of statements marked with ($\spadesuit$) can be found in the full version [16]. The proof of Theorem 2, together with Turing kernelization for maximization problems, is given in Appendix B. The kernel upper bounds for Boolean CSPs can be found in the full version [16].

## 2 Preliminaries

We denote the set of natural numbers including zero by $\mathbb{N}_0$, and the set of positive natural numbers by $\mathbb{N}_+$. For positive integers $n$ we define $[n] := \{1, \ldots, n\}$. For a set $U$ and integer $d \geq 1$ we denote by $\binom{U}{d}$ the collection of all size-$d$ subsets of $U$. All logarithms we employ have base 2. Given a set $U$ and a weight function $w \colon U \to \mathbb{N}_0$, for a subset $S \subseteq U$ we denote $w(S) := \sum_{v \in S} w(v)$.

A graph $G$ has a vertex set $V(G)$ and an edge set $E(G) \subseteq \binom{V(G)}{2}$. For $d \geq 2$, a $d$-uniform hypergraph $G$ consists of a vertex set $V(G)$ and a set of hyperedges $E(G) \subseteq \binom{V(G)}{d}$, that is, each hyperedge is a set of exactly $d$ vertices. Hence a 2-uniform hypergraph is equivalent to a standard graph. A *clique* in a $d$-uniform hypergraph $G$ is a vertex set $S \subseteq V(G)$ such that for each $X \in \binom{S}{d}$ we have $X \in E(G)$: each possible hyperedge among the vertices of $S$ is present. A *vertex cover* for a graph $G$ is a vertex set $S \subseteq V(G)$ containing at least one endpoint of each edge. A vertex cover is *inclusion-minimal* if no proper subset is a vertex cover.

**Parameterized complexity.** A *parameterized problem* $Q$ is a subset of $\Sigma^* \times \mathbb{N}_+$, where $\Sigma$ is a finite alphabet.

▶ **Definition 4.** *Let* $Q, Q' \subseteq \Sigma^* \times \mathbb{N}_+$ *be parameterized problems and let* $h \colon \mathbb{N}_+ \to \mathbb{N}_+$ *be a computable function. A* generalized kernel *for* $Q$ *into* $Q'$ *of size* $h(k)$ *is an algorithm that, on input* $(x, k) \in \Sigma^* \times \mathbb{N}_+$, *takes time polynomial in* $|x| + k$ *and outputs an instance* $(x', k')$ *such that:*
1. $|x'|$ *and* $k'$ *are bounded by* $h(k)$, *and*
2. $(x', k') \in Q'$ *if and only if* $(x, k) \in Q$.
*The algorithm is a* kernel *for* $Q$ *if* $Q = Q'$. *It is a* polynomial (generalized) kernel *if* $h(k)$ *is a polynomial.*

▶ **Definition 5** (Linear-parameter transformations). *Let* $P$ *and* $Q$ *be parameterized problems. We say that* $P$ *is* linear-parameter transformable *to* $Q$, *if there exists a polynomial-time computable function* $f \colon \Sigma^* \times \mathbb{N}_+ \to \Sigma^* \times \mathbb{N}_+$, *such that for all* $(x, k) \in \Sigma^* \times \mathbb{N}_+$, *(a)* $(x, k) \in P$ *if and only if* $(x', k') = f(x, k) \in Q$ *and (b)* $k' \leq \mathcal{O}(k)$. *The function* $f$ *is called a* linear-parameter transformation.

We employ a linear-parameter transformation for proving the lower bound for SUBSET SUM. For other lower bounds we use the framework of cross-composition [4] directly.

▶ **Definition 6** (Polynomial equivalence relation, [4, Def. 3.1]). *Given an alphabet $\Sigma$, an equivalence relation $\mathcal{R}$ on $\Sigma^\star$ is called a polynomial equivalence relation if the following conditions hold.*

**(i)** *There is an algorithm that, given two strings $x, y \in \Sigma^\star$, decides whether $x$ and $y$ belong to the same equivalence class in time polynomial in $|x| + |y|$.*

**(ii)** *For any finite set $S \subseteq \Sigma^\star$ the equivalence relation $\mathcal{R}$ partitions the elements of $S$ into a number of classes that is polynomially bounded in the size of the largest element of $S$.*

▶ **Definition 7** (Degree-$d$ cross-composition). *Let $L \subseteq \Sigma^\star$ be a language, let $\mathcal{R}$ be a polynomial equivalence relation on $\Sigma^\star$, and let $Q \subseteq \Sigma^\star \times \mathbb{N}_+$ be a parameterized problem. A degree-$d$ OR-cross-composition of $L$ into $Q$ with respect to $\mathcal{R}$ is an algorithm that, given $z$ instances $x_1, x_2, \ldots, x_z \in \Sigma^\star$ of $L$ belonging to the same equivalence class of $\mathcal{R}$, takes time polynomial in $\sum_{i=1}^{z} |x_i|$ and outputs an instance $(x', k') \in \Sigma^\star \times \mathbb{N}_+$ such that:*

**(i)** *the parameter $k'$ is bounded by $\mathcal{O}(z^{1/d} \cdot (\max_i |x_i|)^c)$, where $c$ is some constant independent of $z$, and*

**(ii)** *$(x', k') \in Q$ if and only if there is an $i \in [z]$ such that $x_i \in L$.*

▶ **Theorem 8** ([4, Theorem 3.8]). *Let $L \subseteq \Sigma^\star$ be a language that is NP-hard under Karp reductions, let $Q \subseteq \Sigma^\star \times \mathbb{N}_+$ be a parameterized problem, and let $\varepsilon > 0$ be a real number. If $L$ has a degree-$d$ OR-cross-composition into $Q$ and $Q$ parameterized by $k$ has a polynomial (generalized) kernelization of bitsize $\mathcal{O}(k^{d-\varepsilon})$, then $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.*

## 3 Kernel lower bounds

### 3.1 Exact-Edge-Weight Clique

In this section we show that EXACT-EDGE-WEIGHT CLIQUE parameterized by the number of vertices in the given graph $n$ does not admit a generalized kernel of size $\mathcal{O}(n^{3-\varepsilon})$, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. We use the framework of cross-composition to establish a kernelization lower bound [4]. We will use the NP-hard RED-BLUE DOMINATING SET (RBDS) as a starting problem for the cross-composition. Observe that RBDS is NP-hard because it is equivalent to SET COVER and HITTING SET [19].

---

RED-BLUE DOMINATING SET (RBDS)
**Input:** A bipartite graph $G$ with a bipartition of $V(G)$ into sets $R$ (red vertices) and $B$ (blue vertices), and a positive integer $d \leq |R|$.
**Question:** Does there exist a set $D \subseteq R$ with $|D| \leq d$ such that every vertex in $B$ has at least one neighbor in $D$?

---

The following lemma forms the heart of the lower bound. It shows that an instance of EEWC on $z \cdot N^{\mathcal{O}(1)}$ vertices can encode the logical OR of a sequence of $z^3$ instances of size $N$ each. Roughly speaking, this should be interpreted as follows: when $z \gg N$, each of the roughly $z^2$ edge weights of the constructed graph encodes $z$ useful bits of information, in order to allow the instance on $\approx z^2$ edges to represent all $z^3$ inputs.

▶ **Lemma 9.** *There is a polynomial-time algorithm that, given integers $z, d, n, m$ and a set of $z^3$ instances $\{(G_{i,j,k}, R_{i,j,k}, B_{i,j,k}, d) \mid i, j, k \in [z]\}$ of RBDS such that $|R_{i,j,k}| = m$ and $|B_{i,j,k}| = n$ for each $i, j, k \in [z]$, constructs an undirected graph $G'$, integer $t > 0$, and weight function $w \colon E(G') \to \mathbb{N}_0$ such that:*

1. *the graph $G'$ contains a clique of total edge-weight exactly $t$ if and only if there exist $i^*, j^*, k^* \in [z]$ such that $G_{i^*,j^*,k^*}$ has a red-blue dominating set of size at most $d$,*
2. *the number of vertices in $G'$ is $\mathcal{O}(z(m+nd))$, and*
3. *the values of $t$ and $|V(G')|$ depend only on $z, d, n,$ and $m$.*

**Proof.** We describe the construction of $(G', w, t)$; it will be easy to see that it can be carried out in polynomial time. Label the vertices in each set $R_{i,j,k}$ arbitrarily as $r_1, \ldots, r_m$, and similarly label the vertices in each set $B_{i,j,k}$ as $b_1, \ldots, b_n$. We construct a graph $G'$ with edge-weight function $w$ and integer $t$ such that $G'$ has a clique of total edge weight exactly $t$ if and only if some $G_{i,j,k}$ is a YES-instance of RBDS. In the following construction we interpret edge weights as vectors of length $nz + 1$ written in base $(m + d + 2)$, which will be converted to integers later. Starting from an empty graph, we construct $G'$ as follows; see Figure 1.

1. For each $i \in [z]$, create a vertex $b_i$. The vertices $b_i$ form an independent set, so that any clique in $G'$ contains at most one vertex $b_i$.
2. For each $j \in [z]$, create a vertex set $R_j = \{r_1^j, r_2^j, \cdots, r_m^j\}$ and insert edges of weight $\vec{0}$ between all possible pairs of $R_j$.
3. For each $k \in [z]$, create a vertex $s_k$. The vertices $s_k$ form an independent set, so that any clique in $G'$ contains at most one vertex $s_k$.
4. For each $j, k \in [z]$, for each $x \in [m]$, insert an edge between $s_k$ and $r_x^j$ of weight $\vec{0}$.

The next step is to ensure that the neighborhood of a vertex $r_x$ in $G_{i,j,k}$ is captured in the weights of the edges which are incident on $r_x^j$ in $G'$.

5. For each $i, j \in [z]$, for each $x \in [m]$, insert an edge between $b_i$ and $r_x^j$.
6. The weight of each edge $\{b_i, r_x^j\}$ is a vector of length $nz + 1$, out of which the least significant $nz$ positions are divided into $z$ blocks of length $n$ each, and the most significant position is 1. The numbering of blocks as well as positions within a given block start with the least significant position.
   For each $i, j \in [z]$, for each $x \in [m]$, the weight of edge $\{b_i, r_x^j\}$ is defined as follows. For each $k \in [z]$, for each $q \in [n]$, the value $v_{k,q}(b_i, r_x^j)$ represents the value of the $q^{th}$ position of the $k^{th}$ block of the weight of $\{b_i, r_x^j\}$. The value is defined based on the neighborhood of vertex $r_x$ in $G_{i,j,k}$ as follows:

$$v_{k,q}(b_i, r_x^j) = \begin{cases} 1 & \text{if } \{b_q, r_x\} \in E(G_{i,j,k}) \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

   Intuitively, the vector representing the weight of edge $\{b_i, r_x^j\}$ is formed by a 1 followed by the concatenation of $z$ blocks of length $n$, such that the $k^{th}$ block is the 0/1-incidence vector describing which of the $n$ blue vertices of instance $G_{i,j,k}$ are adjacent to $r_x$.

Note that the $n$ blue vertices of an input instance $G_{i,j,k}$ are represented by a single blue vertex $b_i$ in $G'$. The difference between distinct blue vertices is encoded via different positions of the weight vectors. The most significant position of the weight vectors, which is always set to 1 for edges of the form $\{b_i, r_x^j\}$, will be used to keep track of the number of red vertices in a solution to RBDS.

The graph constructed so far has a mechanism to select the first index $i$ of an instance $G_{i,j,k}$ (by choosing a vertex $b_i$), to select the second index $j$ (by choosing vertices $r_x^j$), and to select the third index $k$ (by choosing a vertex $s_k$). The next step in the construction adds weighted
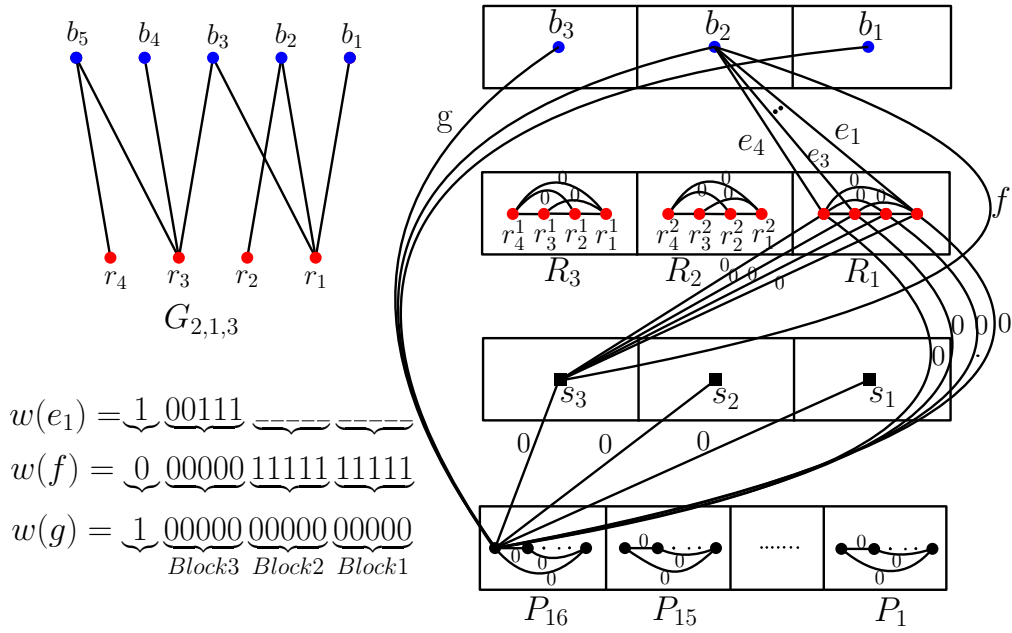
**Figure 1** Top-left: An instance $(G_{2,1,3}, R_{2,1,3}, B_{2,1,3}, 2)$ of RBDS with $m = 4, n = 5$, and $d = 4$. Right: Illustration of the EEWC instance created for a sequence of $3^3$ inputs including the one on the left. For readability, only a subset of the edges is drawn. Bottom-left: For each type edge with non-zero weight, an example weight is shown in vector form.

edges $\{b_i, s_k\}$, of which a solution clique in $G'$ will contain exactly one. The weight vector for this edge is chosen so that the domination requirements from all RBDS instances whose third index differs from $k$ (and which are therefore not selected) can be satisfied "for free".

**7.** For each $i, k \in [z]$, insert an edge between $b_i$ and $s_k$.

**8.** As in Step 6, the weight of the edge $\{b_i, s_k\}$ is a $(1 + nz)$-tuple consisting of the most significant position followed by $z$ blocks of length $n$. There is a 0 at the most significant position, block $k$ consists of $n$ zeros, and the other blocks are filled with ones. Hence the weight of the edge $\{b_i, s_k\}$ is independent of $i$.

To be able to ensure that $G'$ has a clique of exactly weight $t$ if some input instance $G_{i,j,k}$ has a solution, we need to introduce padding numbers which may be used as part of the solution to EEWC.

**9.** For each position $v \in [nz+1]$ of a weight vector, add a vertex set $P_v = \{p_1^v, p_2^v, \cdots, p_{d-1}^v\}$ to $G'$. Recall that $d$ is the upper bound on the solution size for RBDS.

**10.** For each $i \in [z]$, for each $v \in [nz + 1]$, for each $y \in [d - 1]$, add an edge $\{b_i, p_y^v\}$. The weight of edge $\{b_i, p_y^v\}$ has value 1 at the $v^{th}$ position and zeros elsewhere.

**11.** For each $v \in [nz + 1]$, for each $y \in [d - 1]$, add an edge $\{p_y^v, u\}$ of weight $\vec{0}$ for all $u \in V(G') \setminus (\{b_i \mid i \in [z]\} \cup \{p_y^v\})$, i.e., for all vertices $u \neq p_y^v$ which were not already adjacent to $p_y^v$.

We define the target weight $t$ to be the $(nz + 1)$-length vector with value $d$ at each position, which satisfies Condition 3. Observe that $G'$ has $\mathcal{O}(z(m + nd))$ vertices: Steps 1 and 3 contribute $\mathcal{O}(z)$ vertices, Step 2 contributes $\mathcal{O}(zm)$, and Step 9 contributes $\mathcal{O}(d(nz))$. Hence Condition 2 is satisfied. It remains to verify that $G'$ has a clique of total edge weight

exactly $t$ if and only if some input instance $G_{i,j,k}$ has a solution of RED-BLUE DOMINATING SET of size at most $d$. Before proving this property, we show the following claim which implies that no carries occur when summing up the weights of the edges of a clique in $G'$.

▷ **Claim 10.** For any clique $S \subseteq V(G')$, for any position $v \in [nz+1]$ of a weight vector, there are at most $d + m + 1$ edges of the clique $G'[S]$ whose weight vector has a 1 at position $v$, and all other weight vectors are 0 at position $v$.

Proof. By construction, the entries of the vector encoding an edge weight are either 0 or 1.

By Steps 1 and 3, a clique $S$ in $G'$ contains at most one vertex $b_i$ and one vertex $s_k$. Since $G'$ does not have edges between vertices in distinct sets $R_j$ and $R_{j'}$ by Step 2, any clique in $G'$ consists of at most one vertex $b_i$, one vertex $s_k$, a subset of one set $R_j$, and a subset of $\bigcup_{v \in [nz+1]} P_v$. For any fixed position $v \in [nz + 1]$, the only edge-weight vectors which can have a 1 at position $v$ are the $d - 1$ edges from $P_v$ to $b_i$, the edge $\{b_i, s_k\}$, and the $m$ edges between $R_j$ and $b_i$. As this yields $(d - 1) + 1 + m$ edges that possibly have a 1 at position $v$, the claim follows.                                                                  ◁

The preceding claim shows that when we convert each edge-weight vector to an integer by interpreting the vector as its base-$(m + d + 2)$-representation, then no carries occur when computing the sum of the edge-weights of a clique. Hence the integer edge-weights of a clique $S \subseteq V(G')$ sum to the integer represented by vector $t$, if and only if the edge-weight vectors of the edges in $S$ sum to the vector $t$. In the remainder, it therefore suffices to prove that there is a YES-instance $G_{i^*,j^*,k^*}$ of RBDS among the inputs if and only if $G'$ has a clique whose edge-weight vectors sum to the vector $t$. We prove these two implications.

▷ **Claim 11.** If some input graph $G_{i^*,j^*,k^*}$ has a red-blue dominating set of size at most $d$, then $G'$ has a clique of edge-weight exactly $t$.

Proof. Let $S \subseteq R_{i^*,j^*,k^*}$ of size at most $d$ be a dominating set of $B_{i^*,j^*,k^*}$. We define a vertex set $S' \subseteq V(G')$ as follows. Initialize $S' := \{b_{i^*}, s_{k^*}\}$, and for each vertex $r_x \in S$, add the corresponding vertex $r_x^{j^*} \in R_{j^*}$ to $S'$.

We claim that $S'$ is a clique in $G'$. To see this, note that $R_{j^*}$ is a clique by Step 2. Vertex $s_{k^*}$ is adjacent to all vertices of $R_{j^*}$ by Step 4. Vertex $b_{i^*}$ is adjacent to all vertices of $R_{j^*}$ by Step 5. By Step 8 there is an edge between $b_{i^*}$ and $s_{k^*}$.

Let us consider the weight of clique $S'$. Since $S$ is a dominating set of $B_{i^*,j^*,k^*}$, if we sum up the weight vectors of the edges $\{b_{i^*}, r_x^{j^*}\}$ for $r_x \in S$, then by Step 6 we get a value of at least one at each position of block $k^*$. The most significant position of the resulting sum vector has value $|S| \leq d$. By Step 8 the weight vector of the edge $\{b_{i^*}, s_{k^*}\}$ consists of all ones, except for block $k^*$ and the most significant position, where the value is zero. Thus adding the edge weight of $\{b_{i^*}, s_{k^*}\}$ to the previous sum ensures that each block has value at least 1 everywhere, whereas the most significant position has value $|S|$. All other edges spanned by $S$ have weight $\vec{0}$. Letting $t'$ denote the vector obtained by summing the weights of the edges of clique $S'$, we therefore find that $t'$ has value $|S|$ as its most significant position and value at least 1 everywhere else.

Next we add some additional vertices to the set $S'$ to get a clique of weight exactly $t$. By Step 11, vertices from the sets $P_v$ for $v \in [nz + 1]$ are adjacent to all other vertices in the graph and can be added to any clique. All edges incident on a vertex $p_y^v \in P_v$ have weight $\vec{0}$, except the edges to vertices of the form $b_i$ whose weight vector has a 1 at the $v^{th}$ position and 0 elsewhere. Since $S'$ contains exactly one such vertex $b_{i^*}$, for any $v \in [nz + 1]$ we can add up to $d - 1$ vertices from $P_v$ to increase the weight sum at position $v$ from its value of at least 1 in $t'$, to a value of exactly $d$. Hence $G'$ has a clique of edge-weight exactly $t$.                                   ◁

▷ **Claim 12.** If $G'$ has a clique of edge-weight exactly $t$, then some input graph $G_{i^*,j^*,k^*}$ has a red-blue dominating set of size at most $d$.

Proof. Suppose $G'[S']$ is a clique whose total edge weight is exactly $t$. Note that only edges for which one of the endpoints is of the form $b_i$ for $i \in [z]$ have positive edge weights. The remaining edges all have weight $\vec{0}$. Also, by Step 1 there is at most one $b$-vertex in $S'$. Hence since $t \neq \vec{0}$ there is exactly one vertex $b_{i^*}$ in $S'$. By Step 9 and 10, the edges of type $\{b_{i^*}, p_y^v\}$ for $p_y^v \in P_v$ contribute at most $d-1$ to the value of each position $v \in [nz+1]$ of the sum. Hence for each position $v \in [nz+1]$ there is an edge in clique $S'$ of the form $\{b_{i^*}, r_x^j\}$ or $\{b_{i^*}, s_k\}$ which has a 1 at position $v$. We use this to show there is an input instance with a red-blue dominating set of size at most $d$.

By Step 3, there is at most one $s$-vertex in $S'$. Let $k^* := 1$ if $S \cap \{s_1, \ldots, s_z\} = \emptyset$, and otherwise let $s_{k^*}$ be the unique $s$-vertex in $S'$. Since the weight of the edge $\{b_{i^*}, s_{k^*}\}$ has zeros in block $k^*$ by Step 8, our previous argument implies that for each of the $n$ positions of block $k^*$, there is an edge in clique $S'$ of the form $\{b_{i^*}, r_x^j\}$ whose weight has a 1 at that position. Hence $S'$ contains at least one $r$-vertex, and by Step 2 all $r$-vertices in the clique $S'$ are contained in a single set $R_{j^*}$. We show that $G_{i^*,j^*,k^*}$ has a red-blue dominating set of size at most $d$. Let $S := \{r_x \mid r_x^{j^*} \in S'\}$. Since for each of the $n$ positions of block $k^*$ there is an edge $\{b_{i^*}, r_x^j\}$ in $S'$ with a 1 at that position, by Step 5 each blue vertex of $B_{i^*,j^*,k^*}$ has a neighbor in $S$. Hence $S$ is a red-blue dominating set. By Step 5, the most significant position of each edge between $b_{i^*}$ and $R_{j^*}$ has value 1. As the most significant position of the target $t$ is set to $d$, it follows that $|S| \leq d$, which proves that $G_{i^*,j^*,k^*}$ has a red-blue dominating set of size at most $d$. ◁

This completes the proof of Lemma 9. ◀

Lemma 9 forms the main ingredient in a cross-composition that proves kernelization lower bounds for EXACT-EDGE-WEIGHT CLIQUE and its generalization to hypergraphs. For completeness, we formally define the hypergraph version as follows.

---

EXACT-EDGE-WEIGHT $d$-UNIFORM HYPERCLIQUE (EEW-$d$-HC)
**Input:** A $d$-uniform hypergraph $G$, weight function $w \colon E(G) \to \mathbb{N}_0$, and a positive integer $t$.
**Question:** Does $G$ have a hyperclique of total edge-weight exactly $t$?

---

The following theorem generalizes Theorem 1. The case $d = 2$ of the theorem follows almost directly from Lemma 9 and Theorem 8, as the construction in the lemma gives the crucial ingredient for a degree-3 cross-composition. For larger $d$, we essentially exploit the fact that increasing the size of hyperedges by one allows one additional dimension of freedom, as has previously been exploited for other kernelization lower bounds for $d$-HITTING SET and $d$-SET COVER [7, 8]. The proof is given in Appendix A.1.

▶ **Theorem 13.** (★) *For each fixed $d \geq 2$, EXACT-EDGE-WEIGHT $d$-UNIFORM HYPERCLIQUE parameterized by the number of vertices $n$ does not admit a generalized kernel of size $\mathcal{O}(n^{d+1-\varepsilon})$ for any $\varepsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.*

## 3.2   Subset Sum

We show that SUBSET SUM parameterized by the number of items $n$ does not have generalized kernel of bitsize $\mathcal{O}(n^{2-\varepsilon})$ for any $\varepsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. We prove the lower bound by giving a linear-parameter transformation from EXACT RED-BLUE DOMINATING SET. We

use EXACT RED-BLUE DOMINATING SET rather than RED-BLUE DOMINATING SET as our starting problem for this lower bound because it will simplify the construction: it will avoid the need for "padding" to cope with the fact that vertices are dominated multiple times.

The SUBSET SUM problem is formally defined as follows.

---

SUBSET SUM (SS) **Parameter:** $n$
**Input:** A multiset $X$ of $n$ positive integers and a positive integer $t$.
**Question:** Does there exist a subset $S \subseteq X$ with $\sum_{x \in S} x = t$?

---

We use the following problem as the starting point of the reduction.

---

EXACT RED-BLUE DOMINATING SET (ERBDS) **Parameter:** $n := |V(G)|$
**Input:** A bipartite graph $G$ with a bipartition of $V(G)$ into sets $R$ (red vertices) and $B$ (blue vertices), and a positive integer $d \leq |R|$.
**Question:** Does there exist a set $D \subseteq R$ of size *exactly* $d$ such that every vertex in $B$ has exactly one neighbor in $D$?

---

Jansen and Pieterse proved the following lower bound for ERBDS.

▶ **Theorem 14** ([15, Thm. 4.9]). *EXACT RED-BLUE DOMINATING SET parameterized by the number of vertices $n$ does not admit a generalized kernel of size $\mathcal{O}(n^{2-\varepsilon})$ unless* NP $\subseteq$ coNP/poly.

Actually, the lower bound they proved is for a slightly different variant of ERBDS where the solution $D$ is required to have size *at most d*, instead of exactly $d$. Observe that the variant where we demand a solution of size *exactly d* is at least as hard as the *at most d* version: the latter reduces to the former by inserting $d$ isolated red vertices. Therefore the lower bound by Jansen and Pieterse also works for the version we use here, which will simplify the presentation.

▶ **Theorem 3.** *SUBSET SUM parameterized by the number of items $n$ does not admit a generalized kernelization of size $\mathcal{O}(n^{2-\varepsilon})$ for any $\varepsilon > 0$, unless* NP $\subseteq$ coNP/poly.

**Proof.** Given a graph $G$ with a bipartition of $V(G)$ into $R$ and $B$ with $R = \{r_1, r_2, \ldots, r_{n_R}\}$, $B = \{b_1, b_2, \ldots, b_{n_B}\}$, and target value $d$ for ERBDS, we transform it to an equivalent instance $(X, t)$ of SS such that $|X| = n_R$. We start by defining $n_R$ numbers $N_1, N_2, \ldots, N_{n_R}$ in base $(n_R + 1)$. For each $i \in [n_R]$, the number $N_i$ consists of $(n_B + 1)$ digits. We denote the digits of the number $N_i$ by $N_i[1], \ldots, N_i[n_B + 1]$, where $N_i[1]$ is the least significant and $N_i[n_B + 1]$ is the most significant digit. Intuitively, the number $N_i$ corresponds to the red vertex $r_i$. See Figure 2 for an illustration.

For each $i \in [n_R]$, for each $j \in [n_B + 1]$, digit $N_i[j]$ of number $N_i$ is defined as follows:

$$N_i[j] = \begin{cases} 1 & \text{if } j = n_B + 1 \\ 1 & \text{if } j \in [n_B] \text{ and } \{r_i, b_j\} \in E(G) \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Hence the most significant digit of each number is 1, and the remaining digits of number $N_i$ form the 0/1-vector indicating to which of the $n_B$ blue vertices $r_i$ is adjacent in $G$.

To complete the construction we set $X = \{N_1, N_2, \ldots, N_{n_R}\}$ and we define $t$ as follows:

$$t = d \underbrace{11 \ldots 1}_{n_B \text{ times}} \tag{3}$$

| | $ct$ | $B_5$ | $B_4$ | $B_3$ | $B_2$ | $B_1$ |
|---|---|---|---|---|---|---|
| $N_1$ | 1 | 0 | 0 | 0 | 1 | 1 |
| $N_2$ | 1 | 0 | 1 | 1 | 1 | 0 |
| $N_3$ | 1 | 1 | 1 | 1 | 0 | 0 |
| $N_4$ | 1 | 1 | 0 | 0 | 0 | 1 |
| $t$ | 2 | 1 | 1 | 1 | 1 | 1 |
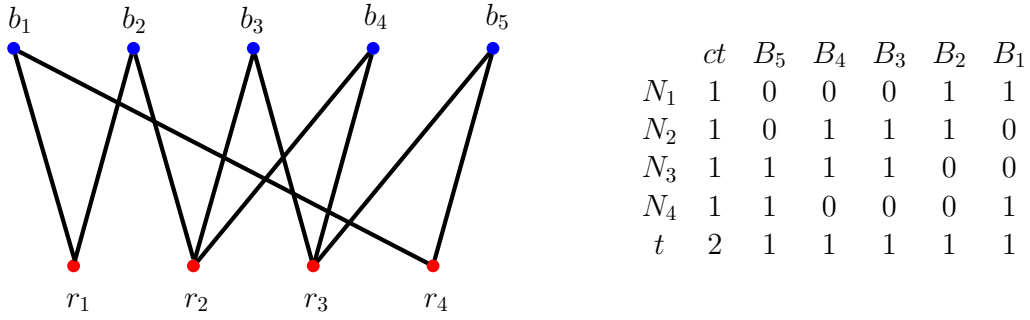
■ **Figure 2** Left: An instance of ERBDS with $n_R = 4$, $n_B = 5$, and $d = 2$. Right: Illustration of the SS instance created for the given input. Note that $\{r_2, r_4\}$ and the numbers $\{N_2, N_4\}$ form a solution for ERBDS and SS, respectively. The leftmost column corresponds to the total count ($ct$) of the number of elements; the remaining columns correspond to blue vertices.

Observe that under these definitions, there are no carries when adding up a subset of the numbers in $X$, as each digit of each of the $n_R$ numbers is either 0 or 1 and we work in base $n_R + 1$.

The number of items $|X|$ in the constructed instance of SS is $n_R$, upper bounded by the parameter $|V(G)|$ of ERBDS. It is easy to see that the construction can be carried out in polynomial time. To complete the linear-parameter transformation from ERBDS to SS, it remains to prove that $G$ has a set $D \subseteq R$ of size exactly $d$ such that every vertex in $B$ has exactly one neighbor in $D$, if and only if there exist a set $S \subseteq X$ with $\sum_{x \in S} x = t$.

In the forward direction, suppose that there exists a set $D \subseteq R$ of size exactly $d$ such that every vertex in $B$ has exactly one neighbor in $D$. We claim that $\{N_i \mid r_i \in D\}$ is a solution to SS. The resulting sum has value $d$ at the most significant digit since $|D| = d$. All other digits correspond to vertices in $B$. Since each blue vertex is adjacent to exactly one vertex from $D$ it is easy to verify that all remaining digits of the sum are exactly one, implying that the numbers sum to exactly $t$.

For the reverse direction, suppose there is a set $S \subseteq X$ with $\sum_{x \in S} x = t$. Since the most significant digit of $t$ is set to $d$ and each number in $X$ has a 1 as most significant digit, we have $|S| = d$ since there are no carries during addition. Define $D := \{r_i \mid N_i \in S\}$ as the set of the red vertices corresponding to the numbers in $S$. As $\sum_{x \in S} x = t$ and no carries occur in the summation, we have $\sum_{x \in S} x[j] = t[j] = 1$ for each $j \in [n_B]$. As the $j$-th digit of all numbers is either 0 or 1 by definition, there is a unique $N_i \in S$ with $N_i[j] = 1$, so that $r_i \in D$ is the unique neighbor of $b_j$ in $D$. This shows that $D$ is an exact red-blue dominating set of size $d$, concluding the linear-parameter transformation.

If there was a generalized kernelization for SS of size $\mathcal{O}(n^{2-\varepsilon})$, then we would obtain a generalized kernelization for ERBDS of size $\mathcal{O}(n^{2-\varepsilon})$ by first transforming it to SS, and then applying the generalized kernelization for the latter. Hence by contraposition and Theorem 14, the claim follows. ◀

## 3.3 Constraint Satisfaction Problems

In this section we extend our lower bounds to cover Boolean Constraint Satisfaction Problems (CSPs). We employ the recently introduced framework [17] of reductions among different CSPs to make a connection with EEW-$d$-HC. We start with introducing terminology necessary to identify crucial properties of CSPs.

**Preliminaries on CSPs.**   A $k$-ary constraint is a function $f \colon \{0,1\}^k \to \{0,1\}$. We refer to $k$ as the arity of $f$, denoted $\text{AR}(f)$. We always assume that the domain is Boolean. A constraint $f$ is satisfied by an input $s \in \{0,1\}^k$ if $f(s) = 1$. A constraint language $\Gamma$ is a finite collection of constraints $\{f_1, f_2, \ldots, f_\ell\}$, potentially with different arities. A *constraint application*, of a $k$-ary constraint $f$ to a set of $n$ Boolean variables, is a triple $\langle f, (i_1, i_2, \ldots i_k), w \rangle$, where the indices $i_j \in [n]$ select $k$ of the $n$ Boolean variables to whom the constraint is applied, and $w$ is an integer weight. The variables can repeat in a single application.

A formula $\Phi$ of $\text{CSP}(\Gamma)$ is a set of constraint applications from $\Gamma$ over a common set of variables. For an assignment $x$, that is, a mapping from the set of variables to $\{0,1\}$, the integer $\Phi(x)$ is the sum of weights of the constraint applications satisfied by $x$. The considered decision problems are defined as follows.

---

EXACT-WEIGHT $\text{CSP}(\Gamma)$                                                         **Parameter:** $n$
**Input:** A formula $\Phi$ of $\text{CSP}(\Gamma)$ over $n$ variables, an integer $t \in \mathbb{Z}$.
**Question:** Is there an assignment $x$ for which $\Phi(x) = t$?

---

MAX-WEIGHT $\text{CSP}(\Gamma)$                                                           **Parameter:** $n$
**Input:** A formula $\Phi$ of $\text{CSP}(\Gamma)$ over $n$ variables, an integer $t \in \mathbb{Z}$.
**Question:** Is there an assignment $x$ for which $\Phi(x) \geq t$?

---

The compressibility of MAX-WEIGHT $\text{CSP}(\Gamma)$ has been studied by Jansen and Włodarczyk [17], who obtained essentially optimal kernel sizes for every $\Gamma$ in the case where the weights are polynomial with respect to $n$. Even though the upper and lower bounds in [17] are formulated for MAX-WEIGHT $\text{CSP}(\Gamma)$, they could be adapted to work with EXACT-WEIGHT $\text{CSP}(\Gamma)$. The crucial idea which allows to determine compressibility of $\Gamma$ is the representation of constraints via multilinear polynomials.

▶ **Definition 15.** *For a $k$-ary constraint $f \colon \{0,1\}^k \to \{0,1\}$ its characteristic polynomial $P_f$ is the unique $k$-ary multilinear polynomial over $\mathbb{R}$ satisfying $f(x) = P_f(x)$ for any $x \in \{0,1\}^k$.*

It is known that such a polynomial always exists and it is unique [26].

▶ **Definition 16.** *The degree of constraint language $\Gamma$, denoted $\deg(\Gamma)$, is the maximal degree of a characteristic polynomial $P_f$ over all $f \in \Gamma$.*

The main result of Jansen and Włodarczyk [17] states that MAX-WEIGHT $\text{CSP}(\Gamma)$ with polynomial weights admits a kernel of $\mathcal{O}(n^{\deg(\Gamma)} \log n)$ bits and, as long as the problem is NP-hard, it does not admit a kernel of size $\mathcal{O}(n^{\deg(\Gamma) - \varepsilon})$, for any $\varepsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. It turns out that in the variant when we allow both positive and negative weights the problem is NP-hard whenever $\deg(\Gamma) \geq 2$ [18]. The lower bounds are obtained via linear-parameter transformations, where the parameter is the number of variables $n$. We shall take advantage of the fact that these transformations still work for an unbounded range of weights.

▶ **Lemma 17** ([17], Lemma 5.4). *For constraint languages $\Gamma_1, \Gamma_2$ such that $2 \leq \deg(\Gamma_1) \leq \deg(\Gamma_2)$, there is a polynomial-time algorithm that, given a formula $\Phi_1 \in CSP(\Gamma_1)$ on $n_1$ variables and integer $t_1$, returns a formula $\Phi_2 \in CSP(\Gamma_2)$ on $n_2$ variables and integer $t_2$, such that*
1. $n_2 = \mathcal{O}(n_1)$,
2. $\exists_x \Phi_1(x) = t_1 \iff \exists_y \Phi_2(y) = t_2$,
3. $\exists_x \Phi_1(x) \geq t_1 \iff \exists_y \Phi_2(y) \geq t_2$.

**Kernel lower bounds for CSP.** The lower bound of $\Omega(n^{\deg(\Gamma)-\varepsilon})$ has been obtained via a reduction from $d$-SAT (with $d = \deg(\Gamma)$) to MAX-WEIGHT CSP($\Gamma$), combined with the fact that MAX $d$-SAT does not admit a kernel of size $\mathcal{O}(n^{d-\varepsilon})$ for $d \geq 2$ [8, 17]. We are going to show that when the weights are arbitrarily large, then the optimal compression size for EXACT-WEIGHT CSP($\Gamma$) becomes essentially $\mathcal{O}(n^{\deg(\Gamma)+1})$, so the exponent is always larger by one compared to the case with polynomial weights. To this end, we are going to combine the aforementioned reduction framework with our lower bound for EXACT-EDGE-WEIGHT $d$-UNIFORM HYPERCLIQUE.

Consider a constraint language $\Gamma_{\mathrm{AND}}^d$ consisting of a single $d$-ary constraint $\mathrm{AND}_d$, which is satisfied only if all the arguments equal 1. The characteristic polynomial of $\mathrm{AND}_d$ is simply $P(x_1, \ldots, x_d) = x_1 x_2 \cdots x_d$, hence the degree of $\Gamma_{\mathrm{AND}}^d$ equals $d$. We first translate our lower bounds for the hyperclique problems into a lower bound for EXACT-WEIGHT CSP($\Gamma_{\mathrm{AND}}^d$) for all $d \geq 2$, and then extend it to other CSPs.

▶ **Lemma 18.** *For all $d \geq 2$, EXACT-WEIGHT CSP($\Gamma_{\mathrm{AND}}^d$) does not admit a generalized kernel of size $\mathcal{O}(n^{d+1-\varepsilon})$, for any $\varepsilon > 0$, unless* NP $\subseteq$ coNP/poly.

**Proof.** Consider an instance $(G, w, t)$ of EXACT-EDGE-WEIGHT $d$-UNIFORM HYPERCLIQUE. Let $W$ be the sum of all weights, which are by the definition non-negative. We can assume $t \in [0, W]$, as otherwise there is clearly no solution. We create an instance $\Phi$ of EXACT-WEIGHT CSP($\Gamma_{\mathrm{AND}}^d$) with the variable set $V(G)$ as follows. For each potential hyperedge $e = \{v_1, \ldots, v_d\}$, if $e \in E(G)$ we create a constraint application $\langle \mathrm{AND}_d, (v_1, \ldots, v_d), w(e) \rangle$ and if $e \notin E(G)$, we create a constraint application $\langle \mathrm{AND}_d, (v_1, \ldots, v_d), W + 1 \rangle$.

If $X \subseteq V(G)$ is a hyperclique with total weight $t$, then for the assignment $x(v) = [v \in X]$ it holds that $\Phi(x) = t$. In the other direction, if $\Phi(x) = t$ then $x$ cannot satisfy any constraint application with weight $W + 1$. Hence, each size-$d$ subset of 1-valued variables corresponds to a hyperedge in $G$ and $X = \{v \in V(G) \mid x(v) = 1\}$ forms a hyperclique of total weight $t$.

We have constructed a linear-parameter transformation from EEW-$d$-HC to EXACT-WEIGHT CSP($\Gamma_{\mathrm{AND}}^d$). Therefore, any generalized kernel of size $\mathcal{O}(n^{d+1-\varepsilon})$ for the latter would entail the same bound for EEW-$d$-HC. The claim follows from Theorem 13. ◀

The lower bound for EXACT-WEIGHT CSP($\Gamma_{\mathrm{AND}}^d$) given by Lemma 18 yields a lower bound for general EXACT-WEIGHT CSP($\Gamma$) using the reduction framework described above.

▶ **Theorem 19.** *For any $\Gamma$ with $\deg(\Gamma) \geq 2$, EXACT-WEIGHT CSP($\Gamma$) does not admit a generalized kernel of size $\mathcal{O}(n^{\deg(\Gamma)+1-\varepsilon})$, for any $\varepsilon > 0$, unless* NP $\subseteq$ coNP/poly.

**Proof.** Consider an $n$-variable instance $(\Phi_1, t_1)$ of EXACT-WEIGHT CSP($\Gamma_{\mathrm{AND}}^d$), where $d = \deg(\Gamma)$. It holds that $\deg(\Gamma_{\mathrm{AND}}^d) = d$. By Lemma 17, there is a linear-parameter transformation that translates $(\Phi_1, t_1)$ into an equivalent instance $(\Phi_2, t_2)$ of EXACT-WEIGHT CSP($\Gamma$). If we could compress $(\Phi_2, t_2)$ into $\mathcal{O}(n^{d+1-\varepsilon})$ bits, this would entail the same compression for $(\Phi_1, t_1)$. The claim follows from Lemma 18. ◀

This concludes the discussion of kernelization lower bounds. The kernelization upper bounds discussed in the introduction can be found in Appendix B (for hyperclique problems) and in the full version [16] (for CSPs).

## 4 Node-weighted Vertex Cover in bipartite graphs

**Preserving all minimum solutions.** For a graph $G$ with node-weight function $w \colon V(G) \to \mathbb{N}_+$, we denote by $\mathcal{C}(G, w)$ the collection of subsets of $V(G)$ which are minimum-weight vertex covers of $G$. For $n$-vertex *bipartite* graphs there exists a weight function with range $[n]$ that preserves the set of minimum-weight vertex covers, which can be computed efficiently.

▶ **Theorem 20.** (♠) *There is an algorithm that, given an n-vertex bipartite graph G and node-weight function $w\colon V(G) \to \mathbb{N}_+$, outputs a weight function $w^*\colon V(G) \to [n]$ such that $\mathcal{C}(G, w) = \mathcal{C}(G, w^*)$. The running time of the algorithm is polynomial in $|V(G)|$ and the binary encoding size of w.*

The proof of the theorem is given in the full version [16]. It relies on the fact that a maximum $b$-matching (the linear-programming dual to VERTEX COVER) can be computed in strongly polynomial time in bipartite graphs by a reduction to MAX FLOW. The structure of a maximum $b$-matching allows two weight-reduction rules to be formulated whose exhaustive application yields the desired weight function. We also prove that the bound of $n$ on the largest weight in Theorem 20 is best-possible.

**Preserving the relative weight of solutions.**    For a graph $G$, we say that two node-weight functions $w, w'$ are *vertex-cover equivalent* if the ordering of inclusion-minimal vertex covers by total weight is identical under the two weight functions, i.e., for all pairs of inclusion-minimal vertex covers $S_1, S_2 \subseteq V(G)$ we have $w(S_1) \leq w(S_2) \Leftrightarrow w'(S_1) \leq w'(S_2)$. While a minimum-weight vertex cover of a bipartite graph can be found efficiently, the following theorem shows that nevertheless weight functions with exponentially large coefficients may be needed to preserve the ordering of minimal vertex covers by weight.

▶ **Theorem 21.** (♠) *For each $n \geq 1$, there exists a node-weighted bipartite graph $G_n$ on $2(n+1)$ vertices with weight function $w\colon V(G_n) \to \mathbb{N}_+$ such that for all weight functions $w'\colon V(G) \to \mathbb{N}_+$ which are vertex-cover equivalent to w, we have:* $\max\limits_{v \in V(G_n)} w'(v) \geq 2^{\Omega(n)}$.

## 5    Conclusions

We have established kernelization lower bounds for SUBSET SUM, EXACT-EDGE-WEIGHT $d$-UNIFORM HYPERCLIQUE, and a family of EXACT-WEIGHT CSP problems, which make it unlikely that there exists an efficient algorithm to compress a single weight into $o(n)$ bits. This gives a clear separation between the setting involving arbitrarily large weights and the case with polynomially-bounded weights, which can be encoded with $\mathcal{O}(\log n)$ bits each. The matching kernel upper bounds are randomized and we leave it as an open question to derandomize them. For SUBSET SUM parameterized by the number of items $n$, a deterministic kernel of size $\mathcal{O}(n^4)$ is known [10].

Kernelization of weighted minimization/maximization problems is so far less understood. We are able to match the same kernel size as for the exact-weight problems, but only through Turing kernels. Using techniques from [10] one can obtain, e.g., a kernel of size $\mathcal{O}(n^8)$ for MAX-EDGE-WEIGHT CLIQUE. Improving upon this bound possibly requires a better understanding of the threshold functions. Our study of weighted VERTEX COVER on bipartite graphs indicates that preserving the order between all the solutions might be overly demanding and it could be easier to keep track only of the structure of the optimal solutions. Can we extend the theory of threshold functions so that better bounds are feasible when we just want to maintain a separation between optimal and non-optimal solutions?

### References

1    Amir Abboud, Shon Feller, and Oren Weimann. On the fine-grained complexity of parity problems. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 5:1–5:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPIcs.ICALP.2020.5`.

**2**    Amir Abboud, Kevin Lewi, and Ryan Williams. Losing weight by gaining edges. In Andreas S. Schulz and Dorothea Wagner, editors, *Algorithms – ESA 2014 – 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings*, volume 8737 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2014. `doi:10.1007/978-3-662-44777-2_1`.

**3**    Andreas Björklund. Determinant sums for undirected Hamiltonicity. *SIAM J. Comput.*, 43(1):280–299, 2014. `doi:10.1137/110839229`.

**4**    Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM J. Discrete Math.*, 28(1):277–305, 2014. `doi:10.1137/120880240`.

**5**    Miroslav Chlebík and Janka Chlebíková. Crown reductions for the minimum weighted vertex cover problem. *Discret. Appl. Math.*, 156(3):292–312, 2008. `doi:10.1016/j.dam.2007.03.026`.

**6**    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`.

**7**    Holger Dell and Dániel Marx. Kernelization of packing problems. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 68–81. SIAM, 2012. `doi:10.1137/1.9781611973099.6`.

**8**    Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23:1–23:27, 2014. `doi:10.1145/2629620`.

**9**    Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. `doi:10.1007/978-1-4471-5559-1`.

**10**   Michael Etscheid, Stefan Kratsch, Matthias Mnich, and Heiko Röglin. Polynomial kernels for weighted problems. *J. Comput. Syst. Sci.*, 84:1–10, 2017. `doi:10.1016/j.jcss.2016.06.004`.

**11**   Henning Fernau. Kernelization, Turing kernels. In *Encyclopedia of Algorithms*, pages 1043–1045. Springer, 2016. `doi:10.1007/978-1-4939-2864-4_528`.

**12**   András Frank and Éva Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987.

**13**   Danny Harnik and Moni Naor. On the compressibility of NP instances and cryptographic applications. *SIAM Journal on Computing*, 39(5):1667–1713, 2010. `doi:10.1137/060668092`.

**14**   Alon Itai and Michael Rodeh. Finding a minimum circuit in a graph. *SIAM J. Comput.*, 7(4):413–423, 1978. `doi:10.1137/0207033`.

**15**   Bart M. P. Jansen and Astrid Pieterse. Optimal sparsification for some binary CSPs using low-degree polynomials. *TOCT*, 11(4):28:1–28:26, 2019. `doi:10.1145/3349618`.

**16**   Bart M. P. Jansen, Shivesh K. Roy, and Michał Włodarczyk. On the hardness of compressing weights, 2021. `arXiv:2107.02554`.

**17**   Bart M. P. Jansen and Michal Wlodarczyk. Optimal polynomial-time compression for Boolean Max CSP. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 63:1–63:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPIcs.ESA.2020.63`.

**18**   Peter Jonsson and Andrei Krokhin. Maximum *H*-colourable subdigraphs and constraint optimization with arbitrary weights. *Journal of Computer and System Sciences*, 73(5):691–702, 2007. `doi:10.1016/j.jcss.2007.02.001`.

**19**   Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. `doi:10.1007/978-1-4684-2001-2_9`.

**20**   Richard M. Karp and Michael O. Rabin. Efficient randomized pattern-matching algorithms. *IBM journal of research and development*, 31(2):249–260, 1987.

**21**   Dániel Marx and Michal Pilipczuk. Everything you always wanted to know about the parameterized complexity of subgraph isomorphism (but were afraid to ask), 2013. `arXiv:1307.2187v3`.

**22**   Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Comb.*, 7(1):105–113, 1987. `doi:10.1007/BF02579206`.

**23**   Jesper Nederlof. Bipartite TSP in $o(1.9999^n)$ time, assuming quadratic time matrix multiplication. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 40–53. ACM, 2020. `doi:10.1145/3357713.3384264`.

**24**   Jesper Nederlof, Erik Jan van Leeuwen, and Ruben van der Zwaan. Reducing a target interval to a few exact queries. In Branislav Rovan, Vladimiro Sassone, and Peter Widmayer, editors, *Mathematical Foundations of Computer Science 2012 – 37th International Symposium, MFCS 2012, Bratislava, Slovakia, August 27-31, 2012. Proceedings*, volume 7464 of *Lecture Notes in Computer Science*, pages 718–727. Springer, 2012. `doi:10.1007/978-3-642-32589-2_62`.

**25**   G.L. Nemhauser and L.E.jun. Trotter. Vertex packings: structural properties and algorithms. *Math. Program.*, 8:232–248, 1975. `doi:10.1007/BF01580444`.

**26**   Noam Nisan and Mario Szegedy. On the degree of Boolean functions as real polynomials. *Computational Complexity*, 4:301–313, 1994. `doi:10.1007/BF01263419`.

**27**   Virginia Vassilevska Williams. Hardness of easy problems: Basing hardness on popular conjectures such as the strong exponential time hypothesis (invited talk). In Thore Husfeldt and Iyad A. Kanj, editors, *10th International Symposium on Parameterized and Exact Computation, IPEC 2015, September 16-18, 2015, Patras, Greece*, volume 43 of *LIPIcs*, pages 17–29. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. `doi:10.4230/LIPIcs.IPEC.2015.17`.

**28**   Virginia Vassilevska Williams and Ryan Williams. Finding, minimizing, and counting weighted subgraphs. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pages 455–464. ACM, 2009. `doi:10.1145/1536414.1536477`.

**29**   Virginia Vassilevska Williams and Ryan Williams. Finding, minimizing, and counting weighted subgraphs. *SIAM J. Comput.*, 42(3):831–854, 2013. `doi:10.1137/09076619X`.

**30**   Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix, and triangle problems. *J. ACM*, 65(5):27:1–27:38, 2018. `doi:10.1145/3186893`.

## A   Kernel lower bounds

### A.1   Omitted proof for Exact-Edge-Weight Clique

▶ **Theorem 13.** (★) *For each fixed $d \geq 2$, EXACT-EDGE-WEIGHT $d$-UNIFORM HYPER-CLIQUE parameterized by the number of vertices $n$ does not admit a generalized kernel of size $\mathcal{O}(n^{d+1-\varepsilon})$ for any $\varepsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.*

**Proof.** We give a degree-$(d+1)$ OR-cross-composition (Definition 7) from RBDS to the weighted hyperclique problem using Lemma 9. We start by giving a polynomial equivalence relation $\mathcal{R}$ on inputs of RBDS. Let two instances of RBDS be equivalent under $\mathcal{R}$ if they have the same number of red vertices, the same number of blue vertices, and the same target value $d$. It is easy to check that $\mathcal{R}$ is a polynomial equivalence relation.

Consider $Z$ inputs of RBDS from the same equivalence class of $\mathcal{R}$. If $Z$ is not a $(d+1)^{th}$ power of an integer, then we duplicate one of the input instances until we reach the first number of the form $2^{(d+1)i}$, which is trivially such a power. This increases the number of instances by at most the constant factor $2^{d+1}$ and does not change whether there is a YES-instance among the instances. As all requirements on a cross-composition are oblivious to constant factors, from now on we may assume without loss of generality that $Z = z^{d+1}$ for some integer $z$. By definition of $\mathcal{R}$, all instances have the same number $m$ of red vertices, the same number $n$ of blue vertices, and have the same maximum size $d$ of a solution.

For $d = 2$, we can simply invoke Lemma 9 for the $z^{d+1} = z^3$ instances of RBDS and output the resulting instance $(G', w, t)$ of EEWC, which acts as the logical OR. Since the encoding size $N$ of an instance of RBDS with $m$ red vertices, $n$ blue vertices, and target value $d$ satisfies $N \in \Omega(n + m + d)$, Lemma 9 guarantees that $G'$ has $\mathcal{O}(z(m + nd)) \in \mathcal{O}(\sqrt[3]{Z} \cdot N^2)$ vertices, which is suitably bounded for a degree-3 cross-composition for the parameterization by the number of vertices. Hence the claimed lower bound for generalized kernelization then follows from Theorem 8.

In the remainder of the proof, we assume $d \geq 3$. Partition the $z^{d+1}$ inputs in $z^{d-2}$ groups $\{X_{i_1,\ldots,i_{d-2}} \mid i_1, \ldots, i_{d-2} \in [z]\}$ of size $z^3$ each. Apply Lemma 9 to each group $X_{i_1,\ldots,i_{d-2}}$. This results in $z^{d-2}$ instances $(G_{i_1,\ldots,i_{d-2}}, w_{i_1,\ldots,w_{d-2}}, t)$ of EEWC on a simple graph. Note that all instances share the same value of $t > 0$, as Lemma 9 ensures that $t$ only depends on $(z, d, n, m)$ which are identical for all groups. Similarly, all resulting instances have the same number of vertices. Hence we can re-label the vertices in each graph so that all graphs $G_{i_1,\ldots,i_{d-2}}$ have the same vertex set $\mathcal{V}$ of size $\mathcal{O}(z(m + nd))$. The YES/NO-answer to each composed instance is the disjunction of the answers to the RBDS instances in its corresponding group.

Build a $d$-uniform hypergraph $G^*$ with weight function $w^* \colon E(G^*) \to \mathbb{N}_0$ and target value $t^*$ as follows:

1. $V(G^*) = \mathcal{V} \cup Y_1 \cup \cdots \cup Y_{d-2}$, where $Y_\ell = \{y_{\ell,j} \mid j \in [z]\}$ for $\ell \in [d-2]$.
2. A set $S \subseteq V(G^*)$ of exactly $d$ vertices is a hyperedge of $G^*$ if there is no $\ell \in [d-2]$ for which $|S \cap Y_\ell| > 1$.
3. The weight of a hyperedge $S$ is equal to 0 if there exists $\ell \in [d-2]$ with $S \cap Y_\ell = \emptyset$. Otherwise, for each $\ell \in [d-2]$ let $i_\ell$ be the unique index $j$ such that $y_{\ell,j} \in S$.
   - If $e_S := S \cap \mathcal{V}$ is an edge in graph $G_{i_1,\ldots,i_{d-2}}$ then define $w^*(S) := w_{i_1,\ldots,i_{d-2}}(e_S)$.
   - Otherwise, let $w^*(S) := t + 1$.
4. Set $t^* = t$.

Since $d \in \mathcal{O}(1)$, hypergraph $G^*$ has $\mathcal{O}(z \cdot (m + nd)) + \mathcal{O}(z \cdot d) \in \mathcal{O}(Z^{1/(d+1)} \cdot (m + n)^{\mathcal{O}(1)})$ vertices. (We use here that $d \leq m$.) Hence the parameter value of the constructed EXACT-EDGE-WEIGHT $d$-UNIFORM HYPERCLIQUE instance is indeed bounded by the $(d + 1)$-th root of the number of input instances times a polynomial in the maximum size of an input instance, satisfying the parameter bound of a degree-$(d + 1)$ cross-composition.

It remains to verify that $G^*$ has a hyperclique of weight $t^*$ if and only if one of the input instances has a RBDS of size at most $d$. By the guarantee of Lemma 9, it suffices to show that $G^*$ has a hyperclique of weight $t^*$ if and only if one of the weighted standard graphs $(G_{i_1,\ldots,i_{d-2}}, w_{i_1,\ldots,i_{d-2}})$ obtained by applying that lemma to some group of $z^3$ inputs, has a clique of weight $t$.

First suppose there exists a weighted graph $(G_{i_1^*,\cdots,i_{d-2}^*}, w_{i_1^*,\ldots,i_{d-2}^*})$ that contains a clique $S$ of total edge weight $t$. Let $I := \{y_{\ell,i_\ell^*} \mid \ell \in [d-2]\}$. Let $S' := S \cup I$. By Step 2, the set $S'$ is a hyperclique in $G^*$. It remains to verify that its weight is $t^* = t$. By Step 3, for each edge $e$ of the clique $S$ the set $e \cup I$ is a hyperedge in $G^*$ of the same weight. Additionally, each subset of $S'$ that does not contain $I$ has weight 0. Hence the weight of hyperclique $S'$ is equal to the weight of clique $S$ and is therefore $t^* = t$.

For the other direction, suppose $G^*$ has a clique $G^*[S^*]$ of weight $t^* = t$. Since $t > 0$ and all hyperedges in $G^*$ of nonzero weight contain exactly one vertex of each set $Y_\ell$ for $\ell \in [d-2]$, there exist $i_1^*, \ldots, i_{d-2}^*$ such that $S^* \cap Y_\ell = \{i_\ell^*\}$ for each $\ell \in [d-2]$. Let $I := \{y_{\ell,i_\ell^*} \mid \ell \in [d-2]\}$. We will show that $S^* \cap \mathcal{V}$ is a clique of weight $t$ in $G_{i_1^*,\ldots,i_{d-2}^*}$. Since $t^* = t > 0$ and edge-weights are non-negative, it follows that no hyperedge in $S^*$ has weight $t + 1$. By Step 3, this implies each subset of $S^* \cap \mathcal{V}$ of size two is an edge of $G_{i_1^*,\ldots,i_{d-2}^*}$, and hence $S^* \cap \mathcal{V}$ is a clique.

For each set $e \subseteq \mathcal{V}$ of size two, the weight of the hyperedge $e \cup I$ is equal to $w_{i_1^*,\dots,i_{d-2}^*}(e)$. As all other hyperedges in $S^*$ have weight 0, it follows that the weight of the clique $S^* \cap \mathcal{V}$ equals that of hyperclique $S^*$, and is therefore equal to $t^* = t$. This implies $G_{i_1^*,\dots,i_{d-2}^*}$ has a clique of total edge weight $t = t^*$, which concludes the proof.                                                      ◄

## B    Kernel upper bounds

In this section, we present randomized kernel upper bounds for EEW-$d$-HC, which match the obtained lower bounds. For the maximization variant of EEW-$d$-HC, we present a Turing kernel with the same bounds. The results in this section follow from combining known arguments from Harnik and Naor [13] and Nederlof et al. [23] with gadgets that allow us to produce an instance of the same problem that is being compressed (so we obtain a true kernelization, not a generalized one).

Consider a family $\mathcal{F}$ of subsets of a universe $U$ and a weight function $w \colon \mathcal{F} \to [-N, N]$. For a subset $X \subseteq U$, we denote $w_{\text{SUM}}(X) = \sum_{Y \in \mathcal{F}, Y \subseteq X} w(Y)$. The following fact has been observed by Harnik and Naor [13, Claim 2.7] and for the sake of completeness we provide a proof for the formulation which is the most convenient for us.

▶ **Lemma 22.** *Let $U$ be a set of size $n$, $\mathcal{F} \subseteq 2^U$ be a family of subsets, $w \colon \mathcal{F} \to [-N, N]$ be a weight function, and $t \in [-N, N]$. There exists a randomized polynomial-time algorithm that, given a real $\varepsilon > 0$, returns a prime number $p \leq 2^n \cdot \text{poly}(n, \log N, \varepsilon^{-1})$, such that if there is no $X \subseteq U$ satisfying $w_{\text{SUM}}(X) = t$, then*

$$\mathbb{P}\Bigg( \text{there is } X \subseteq U \text{ satisfying } w_{\text{SUM}}(X) \equiv t \pmod{p} \Bigg) \leq \varepsilon.$$

**Proof.** For a fixed function $w$, we say that $p$ is *bad* if for some $X \subseteq U$ it holds that $w_{\text{SUM}}(X) \equiv t \pmod{p}$ but $w_{\text{SUM}}(X) \neq t$. This implies that $p$ divides $|w_{\text{SUM}}(X) - t|$. We argue that the number of bad primes is bounded by $2^n \cdot (n + 1 + \log(N))$. Since $|w_{\text{SUM}}(X) - t| \leq 2^{n+1} \cdot N$, this number can have at most $\log(2^{n+1} \cdot N) = n + 1 + \log N$ different prime divisors. There are at most $2^n$ choices of $X$, which proves the bound.

We sample a random prime $p$ among the set of the first $M = 2^n \cdot (n + 1 + \log(N)) \cdot \varepsilon^{-1}$ primes. It is known that the first $M$ primes lie in the interval $[2, \mathcal{O}(M \log M)]$ and we can uniformly sample a prime number from this interval in time $(\log M)^{\mathcal{O}(1)} = (n + \log \log(N) + \log(\varepsilon^{-1}))^{\mathcal{O}(1)}$ [20]. By the argument above, the probability of choosing a bad prime is bounded by $\varepsilon$.                                                                ◄

▶ **Theorem 23.** *There is a randomized polynomial-time algorithm that, given an $n$-vertex instance $(G, w, t)$ of EXACT-EDGE-WEIGHT $d$-UNIFORM HYPERCLIQUE, outputs an instance $(G', w', t')$ of bitsize $\mathcal{O}(n^{d+1})$, such that:*
1. *if $(G, w, t)$ is a YES-instance, then $(G', w', t')$ is always a YES-instance,*
2. *if $(G, w, t)$ is a NO-instance, then $(G', w', t')$ is a NO-instance with probability at least $1 - 2^{-n}$.*

*Furthermore, each number in $(G', w', t')$ is bounded by $2^{\mathcal{O}(n)}$.*

**Proof.** Let us define $N = \max(t, \max_{e \in E(G)} w_e)$. We can assume $\log N \leq 2^n$, because otherwise the input length is lower bounded by $2^n$ and the brute-force algorithm for EEW-$d$-HC becomes polynomial.

We apply Lemma 22 to the weight function $w$, target $t$, and $\varepsilon = 2^{-n}$, to compute the desired prime $p \leq 2^n \cdot \text{poly}(n, \log N, \varepsilon^{-1}) = 2^{\mathcal{O}(n)}$. If there exists a hyperclique $X \subseteq V(G)$ satisfying $w_{\text{SUM}}(X) = t$ with respect to the weighted set family $E(G) \subseteq \binom{V(G)}{d}$, then clearly $w_{\text{SUM}}(X) \equiv t \pmod{p}$. Furthermore, with probability $1 - 2^{-n}$, the implication in the other direction holds as well. In particular, in this case $p$ does not divide $t$.

Let us construct a new instance $(G', w', t')$ of Exact-Edge-Weight $d$-Uniform Hyperclique with weights bounded by $p \cdot n^d$, which is bounded by $2^{\mathcal{O}(n)}$ for constant $d$. We set $w'(v) = w(v) \pmod{p}$ and $t_p = t \pmod{p}$. The condition $w_{\text{SUM}}(X) \equiv t \pmod{p}$ is equivalent to the existence of $i \in [0, n^d)$ for which $w'_{\text{SUM}}(X) = t_p + ip$, because the sum $w'_{\text{SUM}}(X)$ comprises of at most $n^d$ summands from the range $[0, p)$.

We introduce a set $U_Z$ of $d - 1$ new vertices and for each $j \in [0, d - 1]$ we introduce a set $U_j$ of $n$ new vertices. Intuitively, the sets $U_j$ can be used to represent any number $i \in [0, n^d)$ in base $n$. For every $j$ and every $v \in U_j$ we create a hyperedge $e = U_Z \cup \{v\}$ with weight $w'_e = n^j \cdot p$. For every other size-$d$ subset containing at least one new vertex, we create a hyperedge with weight 0. Observe that for every integer $i \in [0, n^d]$, we can find a set $Y \subseteq U_Z \cup U_0 \cup \cdots \cup U_{d-1}$ such that $w'_{\text{SUM}}(Y) = ip$. Let $G'$ be the graph with the set of vertices $V(G) \cup U_Z \cup U_0 \cup \cdots \cup U_{d-1}$ and hyperedges inherited from $G$ plus these defined above. We set $t' = t_p + n^d \cdot p$.

Suppose now that $X \subseteq V(G)$ forms a hyperclique of total weight $t$ in $G$. Then $w'_{\text{SUM}}(X) = t_p + ip$ for some $i \in [0, n^d)$. By the argument above, we can find a set $Y \subseteq U_Z \cup U_0 \cup \cdots \cup U_{d-1}$ such that $w'_{\text{SUM}}(X \cup Y) = t'$ and $X \cup Y$ is a hyperclique in $G'$.

In the other direction, suppose we have successfully applied Lemma 22 and there is a hyperclique $X' \subseteq V(G')$ with total weight $t'$. Then $p$ divides $w'_{\text{SUM}}(X' \setminus V(G))$, so since all hyperedges intersecting both $V(G)$ and $X' \setminus V(G)$ have weight 0, we have $w'_{\text{SUM}}(X' \cap V(G)) \equiv t \pmod{p}$ and $w_{\text{SUM}}(X' \cap V(G)) = t$, which gives a desired hyperclique in $G$.

The new instance has $\mathcal{O}(n)$ vertices and $\mathcal{O}(n^d)$ edges. The weight range is $[0, n^d \cdot p]$ and, since $p = 2^{\mathcal{O}(n)}$, each weight can be encoded with $\mathcal{O}(n)$ bits. The claim follows.     ◀

We obtain Theorem 2 as a corollary by taking $d = 2$.

## B.1    Turing kernel for Max Weighted Hyperclique

We turn our attention to the maximization variant of the weighted hyperclique problem. We consider the problem Max-Edge-Weight $d$-Uniform Hyperclique, which takes the same input as Exact-Edge-Weight $d$-Uniform Hyperclique, but the goal is to detect a hyperclique of total weight greater or equal to the target value $t$. Even though we are not able to compress the weight function as in Theorem 23, we present a Turing kernelization with the same size. We rely on a generic technique of reducing interval queries to exact queries.

▶ **Theorem 24** ([24], Theorem 1). *Let $U$ be a set of cardinality $n$, let $w \colon U \to \mathbb{N}_0$ be a weight function, and let $l < u$ be non-negative integers with $u - l > 1$. There is a polynomial-time algorithm that returns a set of pairs $\Omega = (w_1, t_1), \ldots, (w_K, t_K)$ with $w_i \colon U \to \mathbb{N}_0$ and integers $t_1, t_2, \ldots, t_K$, such that:*
1. *$K$ is at most $(5n + 2) \cdot \log(u - l)$,*
2. *for every set $X \subseteq U$ it holds that $w(X) \in [l, u]$ if and only if there exist $i \in [1, K]$ such that $w_i(X) = t_i$.*

A polynomial Turing kernel (cf. [11]) for a parameterized problem $\mathcal{P}$ is a polynomial-time algorithm that decides any instance of $\mathcal{P}$ with access to an oracle that answers instances of size polynomial with respect to the parameter. The size of a Turing kernel is the maximal size of the instances queried to the oracle. Note that if the classic problem underlying $\mathcal{P}$ is NP-hard, then any query for a potentially different problem $\mathcal{Q}$ can be translated into a query for $\mathcal{P}$ whose size is only polynomially larger. Hence in many settings, including ours, it does not make a difference for the existence of polynomial-size Turing kernels whether the queries are for the same problem or for another problem contained in NP.

Observe that the number of calls to the oracle is not restricted, although it is polynomial in the input size since the overall procedure runs in polynomial time. The following theorem gives a one-sided error randomized Turing kernel of size $\mathcal{O}(n^{d+1})$ for MAX-EDGE-WEIGHT $d$-UNIFORM HYPERCLIQUE parameterized by the number of vertices $n$.

▶ **Theorem 25.** *There is a randomized polynomial-time algorithm that, given an $n$-vertex instance $(G, w, t)$ of MAX-EDGE-WEIGHT $d$-UNIFORM HYPERCLIQUE, returns a family of $K$ instances $(G_i, w_i, t_i)$, $i \in [K]$, of EXACT-EDGE-WEIGHT $d$-UNIFORM HYPERCLIQUE, each of bitsize $\mathcal{O}(n^{d+1})$, such that:*

1. *$K$ is polynomial with respect to the input size,*
2. *if $(G, w, t)$ is a YES-instance, then at least one instance $(G_i, w_i, t_i)$ is a YES-instance,*
3. *if $(G, w, t)$ is a NO-instance, then with probability $1 - 2^{-\Omega(n)}$ all the instances $(G_i, w_i, t_i)$ are NO-instances.*

**Proof.** We apply Theorem 24 with $U$ being the set of hyperedges in $G$, $l = t$, and $u = n^d \cdot \max_{e \in E(G)} w_e$. We can assume that $l \leq u$, as otherwise there can be no solution. We obtain $K = \log(u - l) \cdot \mathcal{O}(n^d)$ many weight functions $w^i$ and integers $t_i$, so that for each $X \subseteq U$ it holds $w_{\text{SUM}}(X) \geq t$ if and only if $w^i_{\text{SUM}}(X) = t_i$ for some $i \in [K]$. Observe that $\log(u - l)$ is upper bounded by the input size, so the condition (1) is satisfied.

The original problem thus reduces to a disjunction of polynomially many instances of EXACT-EDGE-WEIGHT $d$-UNIFORM HYPERCLIQUE. We use Theorem 23 to compress each of them to $\mathcal{O}(n^{d+1})$ bits. The probability that a single instance would be incorrectly compressed is bounded by $2^{-n}$. By the union bound, the probability that any instance would be incorrectly compressed is $n^{\mathcal{O}(1)} \cdot 2^{-n} = 2^{-\Omega(n)}$. ◀