Lower Bounds for the Happy Coloring Problems[*]

Ivan Bliznets[1,2] and Danil Sagunov[1]

[1] St. Petersburg Department of Steklov Institute of Mathematics of the Russian
Academy of Sciences, Saint Petersburg, Russia
`iabliznets@gmail.com`, `danilka.pro@gmail.com`
[2] National Research University Higher School of Economics, Saint Petersburg, Russia

Abstract. In this paper, we study the MAXIMUM HAPPY VERTICES and the MAXIMUM HAPPY EDGES problems (MHV and MHE for short). Very recently, the problems attracted a lot of attention and were studied in Agrawal '17, Aravind et al. '16, Choudhari and Reddy '18, Misra and Reddy '17. Main focus of our work is lower bounds on the computational complexity of these problems. Established lower bounds can be divided into the following groups: NP-hardness of the above guarantee parameterization, kernelization lower bounds (answering questions of Misra and Reddy '17), exponential lower bounds under the SET COVER CONJECTURE and the EXPONENTIAL TIME HYPOTHESIS, and inapproximability results. Moreover, we present an $\mathcal{O}^*(\ell^k)$ randomized algorithm for MHV and an $\mathcal{O}^*(2^k)$ algorithm for MHE, where ℓ is the number of colors used and k is the number of required happy vertices or edges. These algorithms cannot be improved to subexponential taking proved lower bounds into account.

1 Introduction

In this paper, we study MAXIMUM HAPPY VERTICES and MAXIMUM HAPPY EDGES. The problems are motivated by a study of algorithmic aspects of homophyly law in large networks and were introduced by Zhang and Li in 2015 [27]. The law states that in social networks people are more likely to connect with people they like. Social network is represented by a graph, where each vertex corresponds to a person of the network, and an edge between two vertices denotes that the corresponding persons are connected within the network. Furthermore, we let vertices have a color assigned. The color of a vertex indicates type, character or affiliation of the corresponding person in the network. An edge is called *happy* if its endpoints are colored with the same color. A vertex is called *happy* if all its neighbours are colored with the same color as the vertex itself. Equivalently, a vertex is happy if all edges incident to it are happy. Formal definition of MAXIMUM HAPPY VERTICES and MAXIMUM HAPPY EDGES is the following.

MAXIMUM HAPPY VERTICES (MHV)
Input: A graph G, a partial coloring of vertices $p : S \to [\ell]$ for some
               $S \subseteq V(G)$ and an integer k.
Question: Is there a coloring $c : V(G) \to [\ell]$ extending partial coloring p such
               that the number of happy vertices with respect to c is at least k?

MAXIMUM HAPPY EDGES (MHE)
Input: A graph G, a partial coloring of vertices $p : S \to [\ell]$ for some
               $S \subseteq V(G)$ and an integer k.
Question: Is there a coloring $c : V(G) \to [\ell]$ extending partial coloring p such
               that the number of happy edges with respect to c is at least k?

Recently, MHV and MHE have attracted a lot of attention and were studied
from parameterized [1, 2, 3, 6, 23] and approximation [27, 28, 26, 25] points of
view as well as from experimental perspective [21].

NP-hardness of MHVand MHE was proved by Zhang and Li even in case
when only three colors are used. Later, Misra and Reddy [23] proved NP-
hardness of both MHV and MHE on split and on bipartite graphs. However,
MHV is polynomially time solvable on cographs and trees [23, 2]. Approxi-
mation results for MHV are presented in Zhang et al. [28]. They showed that
MHV can be approximated within $\frac{1}{\Delta+1}$, where Δ is the maximum degree of
the input graph, and MHE can be approximated within $\frac{1}{2} + \frac{\sqrt{2}}{4} f(\ell)$, where
$f(\ell) = \frac{(1-1/\ell)\sqrt{\ell(\ell-1)}+1/\sqrt{2}}{\ell-1+1/2\ell}$. From parameterized point of view the following pa-
rameters were studied: pathwidth [1, 3], treewidth [1, 3], neighbourhood diver-
sity [3], vertex cover [23], distance to clique [23], distance to threshold graphs [6].
Kernelization questions were studied in works [1, 17]. Agrawal [1] provided a
$\mathcal{O}(k^2\ell^2)$ kernel for MHV where ℓ is the number of colors used and k is the
number of desired happy vertices. Independently, Gao and Gao [17] present
$2^{k\ell+k} + k\ell + k + \ell$ kernel for general case and $7(k\ell + k) + \ell - 10$ in case of
planar graphs.

Short summary of our results can be found below.

No polynomial kernels: If NP $\not\subseteq$ coNP/poly then there are no polynomial
  kernels for MHV parameterized by vertex cover, and no polynomial kernels
  for MHE under the following parametrizations: number of uncolored ver-
  tices, number of happy edges, and distance to almost any reasonable graph
  class. Moreover, under NP $\not\subseteq$ coNP/poly, there is no $\mathcal{O}((k^d\ell)^{2-\epsilon})$ and no
  $\mathcal{O}((k^d h)^{2-\epsilon})$ bitsize kernel for MHV. Note that these results answer ques-
  tion from [23]: "Do the MAXIMUM HAPPY VERTICES and MAXIMUM HAPPY
  EDGES problems admit polynomial kernels when parameterized by either the
  vertex cover or the distance to clique parameters?"
Above guarantee: Above-greedy versions of MHV and MHE are NP-complete
  even for budget equal 1.
Exponential lower bounds: Assuming the Set Cover Conjecture, MHV and
  MHE do not admit $\mathcal{O}^*((2 - \epsilon)^{n'})$ algorithms, where n' is the number of

uncolored vertices in the input graph. Even with $\ell = 3$, there is no $2^{o(n+m)}$ algorithm for MHV and MHE, unless ETH fails.

Innaproximability: Unless P $=$ NP, MHV does not admit approximation algorithm with factors $\mathcal{O}(n^{\frac{1}{2}-\epsilon})$, $\mathcal{O}(m^{\frac{1}{2}-\epsilon})$, $\mathcal{O}(h^{1-\epsilon})$, $\mathcal{O}(\ell^{1-\epsilon})$, for any $\epsilon > 0$.

Algorithms: We present $\mathcal{O}^*(\ell^k)$ randomized algorithm for MHV and $\mathcal{O}^*(2^k)$ algorithm for MHE. Running time of this algorithms match with the corresponding lower bounds. We should note that an algorithm with the running time of $\mathcal{O}^*(2^k)$ for MHE was also presented by Aravind et al. in [3].

2   Preliminaries

Basic notation. We denote the set of positive integer numbers by $\mathbb{N}$. For each positive integer k, by $[k]$ we denote the set of all positive integers not exceeding k, $\{1, 2, \ldots, k\}$. We use $\sqcup$ for the disjoint union operator, i.e. $A \sqcup B$ equals $A \cup B$, with an additional constraint that A and B are disjoint.

We use traditional $\mathcal{O}$-notation for asymptotical upper bounds. We additionally use $\mathcal{O}^*$-notation that hides polynomial factors. Many of our results concern the parameterized complexity of the problems, including fixed-parameter tractable algorithms, kernelization algorithms, and some hardness results for certain parameters. For detailed survey in parameterized algorithms we refer to the book of Cygan et al. [10].

Throughout the paper, we use standard graph notation and terminology, following the book of Diestel [15]. All graphs in our work are undirected simple graphs. We may refer to the distance to $\mathcal{G}$ parameter, where $\mathcal{G}$ is an arbitrary graph class. For a graph G, we say that a vertex subset $S \subseteq V(G)$ is a $\mathcal{G}$ *modulator* of G, if G becomes a member of $\mathcal{G}$ after deletion of S, i.e. $G \setminus S \in \mathcal{G}$. Then, the *distance to $\mathcal{G}$* parameter of G is defined as the size of its smallest $\mathcal{G}$ modulator.

Graph colorings. When dealing with instances of MAXIMUM HAPPY VERTICES or MAXIMUM HAPPY EDGES, we use a notion of colorings. A *coloring* of a graph G is a function that maps vertices of the graph to the set of colors. If this function is partial, we call such coloring *partial*. If not stated otherwise, we use ℓ for the number of distinct colors, and assume that colors are integers in $[\ell]$. A partial coloring p is always given as a part of the input for both problems, along with graph G. We also call p a *precoloring* of the graph G, and use (G, p) to denote the graph along with the precoloring. The goal of both problems is to extend this partial coloring to a specific coloring c that maps each vertex to a color. We call c a *full coloring* (or simply, a coloring) of G that extends p. We may also say that c is a coloring of (G, p). For convenience, introduce the notion of potentially happy vertices, both for full and partial colorings.

Definition 1. We call a vertex v of (G, p) *potentially happy*, if there exists a coloring c of (G, p) such that v is happy with respect to c. In other words, if u and w are precolored neighbours of v, then $p(u) = p(w)$. We denote the set of all potentially happy vertices in (G, p) by $\mathcal{H}(G, p)$.

By $\mathcal{H}_i(G, p)$ we denote the set of all potentially happy vertices in (G, p) such that they are either precolored with color i or have a neighbour precolored with color i:

$$\mathcal{H}_i(G, p) = \{v \in \mathcal{H}(G, p) \mid N[v] \cap p^{-1}(i) \neq \emptyset\}.$$

In other words, if a vertex $v \in \mathcal{H}_i(G, p)$ is happy with respect to some coloring c of (G, p), then necessarily $c(v) = i$.

For a graph with precoloring (G, p), by $h = |\mathcal{H}(G, p)|$ we denote the number of potentially happy vertices in (G, p). Note that if c is a full coloring of a graph G, then $|\mathcal{H}(G, c)|$ is equal to the number of vertices in G that are happy with respect to c.

3   Polynomial kernels for structural graph parameters

In this section, we study existence of polynomial kernels for MHV or MHE under several parameterizations. We start with proving lower bounds for structural graph parameters. We provide reductions to both MHV and MHE from the following problem.

BOUNDED RANK DISJOINT SETS [16]

Input: A set family $\mathcal{F}$ over a universe U with every set $S \in \mathcal{F}$ having size at most d, and a positive integer k.

Question: Is there a subfamily $\mathcal{F}'$ of $\mathcal{F}$ of size at least k such that every pair of sets S_1, S_2 in $\mathcal{F}'$ we have $S_1 \cap S_2 = \emptyset$?

Theorem 1 ([16]). BOUNDED RANK DISJOINT SETS *parameterized by kd does not admit a polynomial compression even if every set $S \in \mathcal{F}$ consists of exactly d elements and $|U| = kd$, unless* NP $\subseteq$ coNP/poly.

The following two theorems answer open questions posed in [23].

Theorem 2. MAXIMUM HAPPY VERTICES *parameterized by the vertex cover number does not admit a polynomial compression, unless* NP $\subseteq$ coNP/poly.

Proof. We give a polynomial reduction from the SET PACKING problem, such that the vertex cover number of the constructed instance of MHV is at most the size of the universe of the initial instance of SET PACKING plus one. Since BOUNDED RANK DISJOINT SETS is a special case of SET PACKING, from Theorem 1 the theorem statement will then follow. The reduction is as follows.

Given an instance $(U = [n], \mathcal{F} = \{S_1, S_2, \ldots, S_m\}, k)$ of SET PACKING, construct an instance (G, p, k) of MHV. For each $i \in U$, introduce vertex u_i in G and left it uncolored. For each set $S_j \in \mathcal{F}$, introduce a vertex s_j in G and precolor it with color j, i.e. $p(s_j) = j$. Thus, the set of colors used in precoloring p is exactly $[m]$. Then, for each $i \in [n]$ and $j \in [m]$ such that $i \in S_j$, introduce an edge between u_i and s_j in G. Additionally, introduce two vertices t_1 and t_2 to G and precolor them with colors 1 and 2 respectively. Then, introduce an edge

(t_1, t_2) to G and for every $i \in [n]$ and $j \in [2]$, introduce an edge (u_i, t_j) in G. Thus, vertices t_1 and t_2 never become happy and ensure that u_i never become happy for any $i \in [n]$. Finally, set the number of required happy vertices to k. Observe that $\{u_1, \ldots, u_n\} \cup \{t_1\}$ forms a vertex cover of G, hence the vertex cover number of G is at most $n + 1$.

We now claim that $(U, \mathcal{F}, k)$ is a yes-instance of SET PACKING if and only if (G, p, k) is a yes-instance of MHV. Let $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$ be the answer to $(U, \mathcal{F}, k)$, i.e. $S_{i_p} \cap S_{i_q} = \emptyset$ for every distinct $p, q \in [k]$. Since $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$ are disjoint, $s_{i_1}, s_{i_2}, \ldots, s_{i_k}$ do not have any common neighbours in G. Hence, we can extend coloring p to coloring c in a way that $s_{i_1}, s_{i_2}, \ldots, s_{i_k}$ are happy with respect to c ($c(u_i)$ is then, in fact, the index of the set containing u_i, i.e. $u_i \in S_{c(u_i)}$). At least k vertices become happy in G, hence (G, p, k) is a yes-instance of MHV.

In the other direction, let c be a coloring of G extending p so that at least k vertices in G are happy with respect to c. Only vertices that can be happy in (G, p) are vertices of type s_i, hence there are vertices $s_{i_1}, s_{i_2}, s_{i_3}, \ldots, s_{i_k}$ that are happy in G with respect to c. Since these vertices are precolored with pairwise distinct colors and are simultaneously happy, they may have no common neighbours in G. This implies that the corresponding sets of the initial instance $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$ are pairwise disjoint. Hence, they form an answer to the initial instance $(U, \mathcal{F}, k)$ of SET PACKING. This completes the proof. □

Theorem 3. MAXIMUM HAPPY EDGES *parameterized by the number of un-colored vertices or by the number of happy edges does not admit a polynomial compression, unless* $\mathrm{NP} \subseteq \mathrm{coNP/poly}$.

Proof. As in the proof of Theorem 2, we again provide a polynomial reduction from BOUNDED RANK DISJOINT SETS and then use Theorem 1. In this proof though, we will use the restricted version BOUNDED RANK DISJOINTS SETS problem itself (and not the SET PACKING problem), formulated in Theorem 1. That is, we will use the constraint that all sets in the given instance are of the same size d, and the size of the universe $|U|$ is equal to kd. We note that the following reduction has very much in common with the reduction described in the proof of Theorem 2.

Given an instance $([n], \mathcal{F} = \{S_1, S_2, \ldots, S_m\}, k)$ of BOUNDED RANK DISJOINT SETS with $n = kd$ and $|S_i| = d$ for every $i \in [m]$, we construct an instance (G, p, k') of MHE. We assume that each element of the universe $[n]$ is contained in at least one set, otherwise the given instance is a no-instance. Firstly, as in the proof of Theorem 2, for each element of the universe $i \in [n]$, introduce a corresponding vertex u_i in G. For each set S_j, $j \in [m]$, introduce not just one, but n corresponding vertices $s_{j,1}, s_{j,2}, \ldots, s_{j,n}$. Then again, similarly to the proof of Theorem 1, for each i, j such that $i \in S_j$, introduce edges between u_i and *each* vertex $s_{j,t}$ corresponding to the set S_j, i.e. n edges in total. To finish the construction of G, introduce every possible edge (u_i, u_j) in G.

Thus, $V(G) = \{u_i \mid i \in [n]\} \cup \{s_{j,t} \mid j \in [m], t \in [n]\}$ and $E(G) = \{(u_i, s_{j,t}) \mid i \in S_j, t \in [n]\} \cup \{(u_i, u_j) \mid i, j \in [n], i \neq j\}$. Then, precolor the vertices of

G in the usual way, i.e. set $p(s_{j,t}) = j$ for every $j \in [m]$ and $t \in [n]$, and leave each vertex u_i uncolored. Finally, we set the number of required happy edges to $k' = n^2 + k\binom{d}{2} = (kd)^2 + k\binom{d}{2}$. Construction of (G, p, k') is done in polynomial time. Observe that the number of uncolored vertices in (G, p, k) equals the size of the universe n, and the number of required happy edges is polynomial of n. Hence, existence of a polynomial kernel respectively to any of these two parameters for MHE contradicts the statement of Theorem 1. We argue that the initial instance is a yes-instance if and only if (G, p, k') is a yes-instance of MHE.

We prove first that if $([n], \mathcal{F}, k)$ is a yes-instance, then (G, p, k') is a yes-instance. Let $([n], \mathcal{F}, k)$ be a yes-instance of the restricted version of BOUNDED RANK DISJOINT SETS, and let $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$ be the instance solution. As usual, extend p to a coloring c of G by setting $c(u_i)$ to the index of the set in the solution containing u_i, i.e. $c(u_i) = i_t$ for some $t \in k$ and $i \in S_{c(u_i)}$. Since $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$ are disjoint, and their total size equals the size of the universe, such coloring c always exists uniquely for a fixed solution of $([n], \mathcal{F}, k)$. We claim that there are exactly k' happy edges in G with respect to c.

All edges in G are either of type $(u_i, s_{j,t})$ or of type (u_i, u_j). Consider edges of type $(u_i, s_{j,t})$ for a fixed $i \in [n]$. Happy edges among them are those with $c(s_{j,t}) = c(u_i)$. Since $c(s_{j,t}) = p(s_{j,t}) = j$ and $i \in S_{c(u_i)}$, these edges are exacly $(u_i, s_{c(u_i),t})$. Hence, there are n happy edges of this type for a fixed $i \in [n]$ and n^2 happy edges of this type in total. It is left to count the number of happy edges of the clique, i.e. edges of type (u_i, u_j). Observe that each u_i is colored with a color corresponding to a containing set of the answer. Since each set is of size d, the vertices u_i are split by color into k groups of size d. Each group contributes exactly $\binom{d}{2}$ happy edges, and no edge connecting vertices from different groups is happy. Thus, there are exactly $k\binom{d}{2}$ happy edges of type (u_i, u_j) in G with respect to c. We get that exactly $n^2 + k\binom{d}{2}$ edges of G are happy with respect to c, hence (G, p, k') is a yes-instance of MHE.

In the other direction, let (G, p, k') be a yes-instance of MHE, and let c be an optimal coloring of G extending p. At least k' edges are happy in G with respect to c. Let us show that *exactly* k' edges are happy in G with respect to c.

Claim 1. In any optimal coloring c of G extending p, $i \in S_{c(u_i)}$ for each $i \in [n]$.

Proof of Claim 1. Suppose it is not true, and c is an optimal coloring of (G, p) and $i \notin S_{c(u_i)}$ for some $i \in [n]$. For each j with $i \in S_j$, u_i is adjacent to n vertices $s_{j,t}$, which are precolored with color j. None of edges $(u_i, s_{j,t})$ are happy with respect to c, since $j \neq c(u_i)$. The only other edges incident to u_i are $n-1$ edges of the clique. Thus, u_i is incident to at most $n-1$ happy edges.

Choose arbitrary j with $i \in S_j$, and put $c(u_i) = j$. u_i becomes incident to at least n happy edges. Happiness of edges not incident with u_i has not changed. Thus, the change yields at least one more happy edge. A contradiction with the optimality of c. ∎

Claim 2. In any optimal coloring c of G extending p, there are at most $k\binom{d}{2}$ happy edges of type (u_i, u_j) in G with respect to c.

Proof of Claim 2. The vertices u_i are split into groups containing vertices of the same color by c, so the happy edges of type (u_i, u_j) are exactly the edges inside the groups. By Claim 1, each u_i is colored with a color corresponding to a set containing i in c. Hence, each group contains vertices corresponding to elements of the same set, and thus contains at most d vertices. So each u_i is incident to at most $d - 1$ happy edges of type (u_i, u_j), and in total there are at most $n \cdot (d - 1)/2 = k\binom{d}{2}$ such happy edges in G with respect to c. ∎

From Claims 1 and 2 follows that at most $n^2 + k\binom{d}{2} = k'$ edges are happy in G with respect to c. And as seen in the proof of Claim 2, the only way that yields exactly k' happy edges is when u_i are split by color into disjoint groups of size d, each containing vertices corresponding to a set of the initial instance. Hence, if c yields k' happy edges in G, $\{S_{c(u_i)} \mid i \in [n]\}$ is a solution to $([n], \mathcal{F}, k)$. Thus, $([n], \mathcal{F}, k)$ is a yes-instance of BOUNDED RANK DISJOINT SETS. This finishes the whole proof. □

Definition 2. We call a graph family $\mathcal{G}$ *uniformly polynomially instantiable*, if there is an algorithm that, given positive integer n as input, outputs a graph G, such that $|V(G)| \geq n$ and $G \in \mathcal{G}$, in poly(n) time.

Corollary 1. *For any uniformly polynomially instantiable graph family $\mathcal{G}$, MAX-IMUM HAPPY EDGES, parameterized by the distance to graphs in $\mathcal{G}$, does not admit a polynomial compression, unless* NP $\subseteq$ coNP/poly.

Proof. Suppose it is not true and there is a uniformly polynomially instantiable graph family $\mathcal{G}$, such that MHE parameterized by the distance to graphs in $\mathcal{G}$ admits a polynomial compression. We show how to reduce an instance of MHE with d uncolored vertices to an instance of MHE with the distance to graph in $\mathcal{G}$ being at most d, and then get a contradiction with Theorem 3.

Let (G, p, k) be an instance of MHE with d uncolored vertices. Denote the set of all uncolored vertices in (G, p) by U and the set of all precolored vertices by P, so $U \sqcup P = V(G)$. Assume that G has no edge between vertices in P, otherwise delete it and decrease k by one if its endpoints are of the same color in (G, p). Construct an instance (G', p', k') as follows. Use the algorithm that output instances of $\mathcal{G}$, with $|P|$ as input. The algorithm gives a graph G'', such that $|V(G'')| \geq |P|$ and $G'' \in \mathcal{G}$. Take an arbitrary subset $P' \subseteq V(G'')$ of size $|P|$, and identify its vertices with vertices in P. Construct G' by introducing $|U|$ new vertices to G'', that are identified with the vertices of U. Denote the set of these vertices by U'. Then, add an edge between vertices in U' or between vertices in U' and P' if there is an edge between corresponding vertices in G. Finally, construct p' by precoloring vertices in P' with the color of their corresponding vertices in (G, p), leave the vertices of U' uncolored, and precolor all remaining vertices arbitrarily. There may be some happy edges between precolored vertices in (G', p'), let their number be h. Set $k' = k + h$.

One may easily show that the constructed instance (G', p', k') is a yes-instanse of MHE if and only if the initial instance (G, p, k) is a yes-instance of MHE. Moreover, deletion of U' from G' yields $G'' \in \mathcal{G}$. Hence, G' has the distance to

graphs in $\mathcal{G}$ being at most $|U'| = |U| = d$. We therefore obtain the required polynomial reduction that leads to the desired contradiction. $\square$

In the rest of the section we study kernel bitsize lower bounds for MHV, parameterized by either $k + \ell$ or $k + h$, where h is the number of potentially happy vertices. This relates to the result of Agrawal in [1], where the author showed that MHV admits a polynomial kernel with $\mathcal{O}(k^2\ell^2)$ vertices. We show that, for any $d > 0$ and any $\epsilon > 0$, there is no kernel of bitsize $\mathcal{O}(k^d \cdot \ell^{2-\epsilon})$ for MHV. Similarly, we show that there is no kernel of bitsize $\mathcal{O}(k^d \cdot h^{2-\epsilon})$ for MHV. To prove these lower bounds, we refer to the framework of weak cross-compositions, that originates from works of Dell and van Mekelbeek [14], Dell and Marx [13] and Hermelin and Wu [19]. These results are finely summarized by Cygan et al. in the chapter on lower bounds for kernelization [9]. We recall the notion of weak cross-compositions.

Definition 3 ([9, 13, 19]). Let $L \subseteq \Sigma^*$ be a language and $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized language. We say that L *weakly-cross-composes* into Q if there exists a real constant $d \geq 1$, called the dimension, a polynomial equivalence relation $\mathcal{R}$, and an algorithm $\mathcal{A}$, called the *weak cross-composition*, satisfying the following conditions. The algorithm $\mathcal{A}$ takes as input a sequence of $x_1, x_2, \ldots, x_t \in \Sigma^*$ that are equivalent with respect to $\mathcal{R}$, runs in time polynomial in $\sum_{i=1}^{t} |x_i|$, and outputs one instance $(y, k) \in \Sigma^* \times \mathbb{N}$ such that:

(a) for every $\delta > 0$ there exists a polynomial $p(\cdot)$ such that for every choice of t and input strings $x_1, x_2, \ldots, x_t$ it holds that $k \leq p(\max_{i=1}^{t} |x_i|) \cdot t^{\frac{1}{d}+\delta}$, and
(b) $(y, k) \in Q$ if and only if there exists at least one index i such that $x_i \in L$.

The framework of weak cross-compositions is used for proving conditional lower bounds on polynomial compression bitsize. This is formulated in the following theorem.

Theorem 4 ([9, 13, 19]). *If an* NP-*hard language L admits a weak cross-composition of dimension d into a parameterized language Q. Then for any $\epsilon > 0$, Q does not admit a polynomial compression with bitsize $\mathcal{O}(k^{d-\epsilon})$, unless* NP $\subseteq$ coNP/poly.

Dell and Marx [13] use this framework to show that the VERTEX COVER problem parameterized by the solution size does not admit a kernel with sub-quadratic bitsize. Their result is the following.

Lemma 1 ([13, 9]). *There exists a weak cross composition of dimension 2 from an* NP-*hard problem* MULTICOLORED BICLIQUE *into the* VERTEX COVER *problem parameterized by the solution size. In fact, this weak cross-composition $\mathcal{A}$, given instances $x_1, x_2, \ldots, x_t$ of* MULTICOLORED BICLIQUE *as input, outputs an instance (G, k') of* VERTEX COVER *satisfying*

 − $|V(G)| \leq p(\max_{i=1}^{t} |x_i|) \cdot \sqrt{t}$, *and*
 − $|V(G)| - k' \leq q(\max_{i=1}^{t} |x_i|)$,

for some polynomials p and q.

The bound for $|V(G)| - k'$ is given because one can look at an instance (G, k') of VERTEX COVER as at an instance $(G, |V(G)| - k')$ of INDEPENDENT SET. Then, the solution parameter of INDEPENDENT SET is bounded with polynomial of the maximum input size, independently of the number of instances t. We are ready to prove the theorem.

Theorem 5. *For any fixed constant d and any $\epsilon > 0$, MAXIMUM HAPPY VER-TICES does not admit polynomial compressions with bitsizes $\mathcal{O}((k^d \cdot \ell)^{2-\epsilon})$ and $\mathcal{O}((k^d \cdot h)^{2-\epsilon})$, where h is the number of potentially happy vertices, unless* NP $\subseteq$ coNP/poly.

Proof. Let d be an arbitrary fixed constant. We show that MULTICOLORED BI-CLIQUE admits a polynomial compression into MHV parameterized either by $k^d \cdot \ell$ or by $k^d \cdot h$. By Theorem 4, it is sufficient for proving the theorem.

We extend the weak cross-composition $\mathcal{A}$ into VERTEX COVER from Lemma 1. Thus, we obtain the desired weak cross-composition $\mathcal{A}'$ into MHV. Firstly, $\mathcal{A}'$ runs $\mathcal{A}$ to obtain an instance (G, k') of VERTEX COVER. Equivalently, $(G, |V(G)| - k')$ is an instance of INDEPENDENT SET. Let $k = |V(G)| - k'$. Transform the instance (G, k) of INDEPENDENT SET into an equivalent instance (G', p, k), where G' is a graph obtained from G by subdivision of each edge of G. The precolor-ing p colors each vertex of G in G' with an unique color corresponding to this vertex, i.e. $p(v) = v$ for each $v \in V(G)$. The vertices of G' that are introduced because of the subdivision are left uncolored. Thus, the number of colors used equals $\ell = |V(G)|$. Note that all vertices of G are potentially happy in (G', p), so the number of potentially happy vertices in (G', p) also equals $h = \ell = |V(G)|$. The following claim shows that the constructed instance (G', p, k) is equivalent to the instance (G, k) of INDEPENDENT SET.

Claim 3. For any $S \subseteq V(G)$, S is an independent set in G if and only if all vertices in S can be simultaneously happy in (G', p).

Proof of Claim 3. Let S be an independent set in G. We construct a coloring c extending p as follows. For each uncolored vertex e_{uv} of G', that corresponds to the subdivision of the edge $uv \in E(G)$, put

$$c(e_{uv}) = \begin{cases} p(u), & \text{if } u \in S, \\ p(v), & \text{if } v \in S, \\ \text{any,} & \text{otherwise.} \end{cases}$$

Since S is an independent set, $u \in S$ and $v \in S$ never hold simultaneously. Thus, for each $v \in S$ and each $e_{uv} \in N_{G'}(v)$, $c(e_{uv}) = p(v) = c(v)$. Therefore, all vertices in S are happy with respect to c.

In the other direction, let c be a coloring of G' extending p. Firstly, note that no newly-introduced vertex $e_{uv} \in V(G') \setminus V(G)$ can be happy. e_{uv} is adjacent to vertices u and v in G', but $p(u) \neq p(v)$, as uv is an edge of G and p corresponds to a proper coloring of G. Hence, if S is a subset of vertices that are happy

with respect to c, then $S \subseteq V(G)$ necessarily. Suppose now that S is not an independent set in G, i.e. $u, v \in S$, but $uv \in E(G)$. Consider the vertex e_{uv} in G'. Since both u and v are happy with respect to c, $c(u) = c(e_{uv})$ and $c(v) = c(e_{uv})$. But $p(u) \neq p(v)$, a contradiction. The proof of the claim is finished. ∎

Finally, $\mathcal{A}'$ outputs the instance (G', p, k) as an instance of a language parameterized either by $k^d \ell$ or by $k^d h$, i.e. $((G', p, k), k^d \ell)$ or $((G', p, k), k^d h)$. Since $h = \ell$, these parameters are equal. We now show that $\mathcal{A}'$ is a weak cross-composition of dimension 2. We already proved that the instance output by $\mathcal{A}'$ is equivalent to the instance output by $\mathcal{A}$, so the condition (b) of weak cross-compositions is satisfied. It suffices to prove that the condition (a) is satisfied as well.

Note that

$$k^d h = k^d \ell = (|V(G)| - k')^d \cdot |V(G)| \leq (q(\max_{i=1}^{t} |x_i|))^d \cdot p(\max_{i=1}^{t} |x_i|) \cdot \sqrt{t}$$
$$= (q^d p)(\max_{i=1}^{t} |x_i|) \cdot \sqrt{t}, \tag{1}$$

where p and q are polynomials from Lemma 1. Thus, $\mathcal{A}'$ satisfies the condition (a) of weak cross-compositions. Hence, $\mathcal{A}'$ is the desired weak cross-composition from Multicolored Biclique to MHV, and theorem follows. □

4   Parameterization above guarantee

This section concerns the above guarantee parameter for MHV and MHE. By *guarantee* we mean the number of happy vertices or edges that can be obtained with a trivial extension of the precoloring given in input. The definition of trivial extensions follows.

Definition 4. For a graph with precoloring (G, p), we call a full coloring c a *trivial extension* of p, if p can be extended to c by choosing a single color i and assigning color i to every uncolored vertex. In other words, $p(v) = c(v)$ for every $v \in p^{-1}([\ell])$, and $c(u) = c(v)$ for every $u, v \notin p^{-1}([\ell])$.

We formulate the version of MHV where the above guarantee parameter equals one.

Above Guarantee Happy Vertices
Input: A graph G, a partial coloring $p : S \to [\ell]$ for some $S \subseteq V(G)$ and integer k, such that there is a trivial extension of p that yields exactly k happy vertices in G.
Question: Is $(G, p, k+1)$ a yes-instance of MHV?

The Above Guarantee Happy Edges is formulated analogously. We show that both these problems cannot be solved in polynomial time, unless P = NP. We start with Above Guarantee Happy Vertices. To prove that it is computationally hard, we provide a chain of polynomial reductions. An intermediate problem in this chain is the Weighted MAX-2-SAT problem.

WEIGHTED MAX-2-SAT

Input: A boolean formula in 2-CNF with integer weights assigned to its
                clauses, an integer w.

Question: Is there an assignment of the variables of ϕ satisfying clauses of
                total weight at least w in ϕ?

Lemma 2. WEIGHTED MAX-2-SAT *is* NP-*complete even when the inputs ϕ
and w satisfy

1. *The total weight of all positive clauses (i.e., clauses containing at least one positive literal) of ϕ equals $w - 1$;*
2. *Each clause of ϕ is assigned either weight 1 or weight 13;*
3. *Each variable appears exactly three times in ϕ, at least once positively in a clause containing also a negative literal, and at least once negatively in a clause containing also a positive literal.*

Proof. The proof is by a chain of technical polynomial reductions from 3-SAT, that is a classical NP-complete problem. Most of the reductions below are classical, but we carefully follow them to ensure that intermediate formulas have certain important properties.

Let $\phi_0 = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ be a formula in CNF on n variables consisting of m clauses, and each clause consists of no more than three literals. Transform the formula to $\phi_1 = (C_1 \vee x) \wedge (C_2 \vee x) \wedge \ldots \wedge (C_m \vee x) \wedge \overline{x}$, where x is a variable new to the formula. We obtain an equivalent 4-SAT input formula on $n + 1$ variables and $m + 1$ clauses, where each clause except one contains a positive literal. We can trivially satisfy m clauses of the formula, but if all $m + 1$ clauses of ϕ_1 can be satisfied, then all clauses of the initial formula ϕ_0 can be satisfied as well.

Now transform the 4-CNF formula ϕ_1 to a 3-CNF formula ϕ_2, so it contains just one negative clause as well as ϕ_1. Do it as follows. Leave clauses of ϕ_1, that consist of at most three literals, as is, and introduce them to ϕ_2. For each clause of ϕ_1 consisting of four literals, replace it with two clauses, introducing a new variable specific to this clause. That is, take a clause of ϕ_1 of length four, say $a \vee b \vee c \vee x$, where a, b, c are some literals (may be positive as well as negative), but x is necessarily a positive literal. Then, take a new variable z_i, and introduce clauses $(a \vee b \vee z_i) \wedge (\overline{z_i} \vee x)$ to ϕ_2. Note that these two clauses are both positive. Hence, ϕ_2 is a 3-CNF formula equivalent to ϕ_1, and the only negative clause in ϕ_2 is $\overline{x}$.

We then need each variable in ϕ_2 to appear at most three times. We use a standard technique to achieve that. If a variable y appears $k > 3$ times in ϕ_2, introduce k new variables $y_1, y_2, \ldots, y_k$ to ϕ_2, replace i^{th} occurence of y with a variable y_i, and introduce k clauses $(\overline{y_1} \vee y_2) \wedge (\overline{y_2} \vee y_3) \wedge \ldots \wedge (\overline{y_k} \vee y_1)$, so that newly-introduced variables are equal in any satisfying assignment. Note that no new negative clause is introduced, so the obtained formula ϕ_3 is a formula in 3-CNF equivalent to ϕ_2, with only one negative clause, and each variable appears in ϕ_3 at most three times. It is possible to satisfy all except one clause of ϕ_3 simultaneously, but it is NP-complete to decide whether one can satisfy

the entire formula. The only negative clause (that is, a clause consisting only of negative literals) in ϕ_3 consists of a single literal.

We now transform our special instance of 3-SAT into a special instance of MAX-2-SAT. In our special case, all clauses of length three are positive. We use the classical reduction from 3-SAT to MAX-2-SAT [18]: given an initial formula in 3-CNF, we replace each clause of the formula consisting of exactly three literals, say $a \vee b \vee c$, with ten clauses, introducing a new variable z_i:

$$z_i \wedge a \wedge b \wedge c \wedge (\overline{a} \vee \overline{b}) \wedge (\overline{b} \vee \overline{c}) \wedge (\overline{a} \vee \overline{c}) \wedge (\overline{z_i} \vee a) \wedge (\overline{z_i} \vee b) \wedge (\overline{z_i} \vee c).$$

These ten clauses has a property that, an assignment of the variables of a, b, c satisfies $a \vee b \vee c$ if and only if the same assignment satisfies exactly seven clauses out of these ten clauses, with at least one of the two possible assignments of z_i. Also, no more than seven clauses can be satisfied simultaneously among these ten clauses.

In our case, $a \vee b \vee c$ is a positive clause. If there are three or two positive literals among a, b, c, replace $a \vee b \vee c$ with the ten clauses above, introducing a new variable z_i. If there is only one positive literal among a, b, c, replace $a \vee b \vee c$ instead with the following ten clauses, where the literals of z_i are negated:

$$\overline{z_i} \wedge a \wedge b \wedge c \wedge (\overline{a} \vee \overline{b}) \wedge (\overline{b} \vee \overline{c}) \wedge (\overline{a} \vee \overline{c}) \wedge (z_i \vee a) \wedge (z_i \vee b) \wedge (z_i \vee c).$$

Note that after such replacement, the all-true assignment of the variables of a, b, c and z_i satisfies exactly seven out of the ten clauses above.

We do not change any clause consisting of less than three literals. Let the initial formula consist of $m = m_1 + m_2 + m_3$ clauses, where m_i is the number of clauses consisting of exactly i literals in the initial formula. Then, after the transformation, we obtain $m_1 + m_2 + 10m_3$ clauses consisting of at most two literals, and we ask to satisfy at least $m_1 + m_2 + 7m_3$ of them simultaneously.

Now continue our chain of reductions and apply the described reduction to ϕ_3. We obtain a 2-CNF formula ϕ_4 that consists of $m_1 + m_2 + 10m_3$ clauses of length at most two, where m_i is the number of clauses of length i in ϕ_3. Consider the all-true assignment in ϕ_3. It satisfies all clauses of ϕ_3, except the single clause consisting of one negative literal. Then, the all-true assignment in ϕ_4 (including the variables that are newly-introduced in ϕ_4) satisfies exactly $(m_1 - 1) + m_2 + 7m_3$ clauses of ϕ_4. Thus, we again obtain an NP-hard problem of satisfying one more clause of the formula than in the all-true assignment. And now, our formula is in 2-CNF. Moreover, each variable appears at most three times in ϕ_3, and it gets copied at most four times in ϕ_4. Therefore, our formula ϕ_4 is a 2-CNF formula that also has a property that each variable appears at most twelve times in it.

Then, we again reduce the number of occurences of a variable in our formula. We again do that in the standard way: for each variable y that occurs k times (even for $k < 3$), we introduce new variables $y_1, y_2, \ldots, y_k$, replace i^{th} occurence with y_i for each $i \in [k]$, and introduce new clauses $(\overline{y_1} \vee y_2) \wedge (\overline{y_2} \vee y_3) \wedge \ldots \wedge (\overline{y_k} \vee y_1)$. The obtained formula ϕ_5 is in 2-CNF and each variable appears in ϕ_5 at most three times, but we can not just ask to satisfy one more clause in ϕ_5 than

in the all-true assignment. The reason behind this is that now we do not ask to satisfy the whole formula, so the clauses of type $(\overline{y_1} \vee y_2) \wedge (\overline{y_2} \vee y_3) \wedge \ldots \wedge (\overline{y_k} \vee y_1)$ might not be satisfied completely in an appropriate assignment. Thus, it might be the case that in an assignment σ that satisfies a sufficient number of clauses of ϕ_5, $\sigma(y_i) \neq \sigma(y_j)$ for some pair of variables corresponding to the same variable y of the initial formula ϕ_4. So the obtained problem of satisfying clauses of ϕ_5 is not equivalent to the initial one.

To overcome this difficulty, we assign weights to the clauses of ϕ_5, making the newly-introduced clauses of type $(\overline{y_i} \vee y_{i+1})$ weigh more than the regular clauses that come from ϕ_4. Specifically, we assign weight 13 to each newly-introduced clause of ϕ_5, and we assign weight 1 to each remaining clause of ϕ_5. ϕ_4 consists of $m_1 + m_2 + 10m_3$ clauses, and $(m_1 - 1) + m_2 + 7m_3$ clauses are satisfied in ϕ_4 with the all-true assignment. Let t be the number of newly-introduced clauses in ϕ_5. Then, ϕ_5 consists of $m_1 + m_2 + 10m_3 + t$ clauses, and the total weight of clauses that are satisfied in ϕ_5 with the all-true assignment is $(m_1 - 1) + m_2 + 7m_3 + 13t$. We ask whether it is possible to satisfy clauses of total weight at least $m_1 + m_2 + 7m_3 + 13t$ in ϕ_5. We claim that this problem is equivalent to the problem of satisfying clauses of ϕ_4.

Claim 4. There is an assignment satisfying at least s clauses of ϕ_4 if and only if there is an assignment satisfying clauses of total weight at least $s + 13t$ in ϕ_5.

Proof of Claim 4. The proof in one direction is trivial. Given an appropriate assignment σ of the variables of ϕ_4, it is easy to construct an appropriate assignment σ' for ϕ_5. Just put $\sigma'(y_i) = \sigma(y)$ for i^{th} occurence of a variable y in ϕ_4. σ' satisfies all s clauses in ϕ_5 corresponding to the clauses satisfied by σ in ϕ_4, and satisfies all t newly-introduced clauses of weight 13 in ϕ_5.

In the other direction, take an assignment σ' that satisfies the maximum possible number of clauses in ϕ_5 simultaneously. We argue that σ' satisfies all clauses of weight 13 in ϕ_5. Suppose it's not true and for some variable y appearing k times in ϕ_4, at least one clause in $(\overline{y_1} \vee y_2) \wedge (\overline{y_2} \vee y_3) \wedge \ldots \wedge (\overline{y_k} \vee y_1)$ in ϕ_5 is not satisfied by σ'. Apart from two of these k clauses of weight 13, each y_i appears in exactly one clause of weight 1 in ϕ_5, that comes from ϕ_4 initially. Recall that each variable of ϕ_4 appears at most 12 times in ϕ_4. Hence, the variables $y_1, y_2, \ldots, y_k$ together touch at most 12 clauses of weight 1 in ϕ_5. Change σ' by setting $\sigma'(y_i) = 0$ for each $i \in [k]$. Some clauses of weight 1 may become unsatisfied, but there are at most 12 of them. Thus, at most 12 weight is lost with the change. At the other hand, all clauses of weight 13 become satisfied, and at least 13 weight is gained with the change. At least 1 weight is gained with the change of σ' in total — a contradiction with the optimality of σ'.

Thus, if σ' satisfies clauses of total weight at least $s + 13t$ in ϕ_5, we may be sure that all t clauses of weight 13 are satisfied by σ'. That is, $\sigma'(y_i) = \sigma'(y_j)$ holds for each pair y_i, y_j of variables corresponding to occurences of the same variable y in ϕ_4. We get that an assignment σ, constructed by $\sigma(y) = \sigma'(y_1)$, satisfies at least s clauses of ϕ_4. The claim statement follows. ∎

We conclude that obtained problem of satisfying clauses of ϕ_5 is equivalent to the initial problem of satisfying ϕ_0. Moreover, ϕ_5 satisfies the lemma conditions. All reductions presented are polynomial, and the lemma statement follows. $\square$

The chain continues with the following version of the INDEPENDENT SET problem.

INDEPENDENT SET ABOVE COLORING

Input: A graph G, properly colored with ℓ colors: $V(G) = V_1 \sqcup V_2 \sqcup \ldots \sqcup V_\ell$.

Question: Is there an independent set of size at least $\max\limits_{i=1}^{\ell} |V_i| + 1$ in G?

Lemma 3. INDEPENDENT SET ABOVE COLORING *is* NP-*complete for $\ell = 3$.*

Proof. We reduce from the special case of WEIGHTED MAX-2-SAT from the statement of Lemma 2. We reduce to the weighted version of INDEPENDENT SET first, and then show how to get rid of the weights.

Let be given a formula ϕ and an integer w as an instance of WEIGHTED MAX-2-SAT satisfying the conditions of Lemma 2. We construct a graph G with weights assigned to its vertices in the same way as in the classical reduction from SATISFIABILITY to CLIQUE (as a complement of INDEPENDENT SET) by Cook [7] or Karp [20]. That is, for each literal in ϕ, we introduce a new vertex in G. Since the clauses in ϕ are weighted, we assign each vertex a weight equal to the weight of the clause of the corresponding literal. Then, for each clause of length two in ϕ, we connect the vertices corresponding to its literals by an edge in G. Finally, we connect each pair of vertices in G that correspond to opposite literals of the same variable in ϕ by an edge. A claim follows.

Claim 5. ϕ has an assignment satisfying clauses of total weight at least w if and only if G has an independent set of total weight at least w.

We now show how to color G with three colors properly. Firstly, for each positive clause of ϕ, take an arbitrary positive literal of this clause and color the corresponding vertex in G with color 1. Thus the set V_1 of the vertices colored with color 1 in G becomes constructed. Note that by condition 1 of Lemma 2, the total weight of the vertices in V_1 equals $w-1$. Moreover, V_1 forms an independent set in G, since its vertices correspond to positive literals from pairwise different clauses.

Then, take each positive clause of ϕ that contains a negative literal. That is, a clause consisting of one negative and one positive literal. Color the negative literal of the clause with color 2. Note that for now vertices of color 2 form an independent set in G, since they correspond to negative literals from pairwise different clauses.

By condition 3 of Lemma 2, each variable of ϕ has three corresponding vertices in G. Moreover, at least two corresponding vertices are already colored in G: at least one corresponding to a positive occurence in a negative clause is colored with color 1 and at least one corresponding to a negative occurence in a positive clause is colored with color 2. Hence, for each variable of ϕ, at most one vertex corresponding to this variable is uncolored.

Take any uncolored vertex v in G. It corresponds either to a positive literal, say x, or a negative literal, say $\overline{x}$. If it corresponds to a positive literal x, then this literal is a literal from a clause consisting of two positive literals, otherwise v would be colored with color 1. Let this clause be $x \vee y$. Since the clause is positive and the vertex v corresponding to x is uncolored, then the vertex corresponding to y is colored with color 1. This vertex is connected to v by an edge, so color v with color 3.

If v corresponds to a literal $\overline{x}$, then this literal is necessarily from a negative clause. If this clause consists just of $\overline{x}$, color v with color 2 or color 3 arbitrarily. Otherwise, the clause is of length two, say $\overline{x} \vee \overline{y}$. The vertex corresponding to $\overline{y}$ is not colored, since it is not from a positive clause. Both vertices corresponding to $\overline{x}$ and $\overline{y}$ are uncolored, so color one of them with color 2 and the other with color 3 arbitrarily. Note that the vertices colored with color 2 still correspond to negative literals from pairwise different clauses, hence they still form an independent set in G. The vertices of color 3 correspond to pairwise different variables from pairwise different clauses, so they form an independent set in G as well.

We colored G with three colors, that is, partitioned $V(G)$ into $V(G) = V_1 \sqcup V_2 \sqcup V_3$, where each of V_1, V_2, V_3 forms an independent set in G. Though the constructed graph G has weights assigned to its vertices. Note that positive integer clauses weights can be avoided in WEIGHTED MAX-2-SAT just by replacing clause each clause of weight t with t copies of this clause in the formula. By condition 2 of Lemma 2, some clauses get copied 13 times, and the others remain appearing just once in the formula. Then, the reduction from MAX-2-SAT to INDEPENDENT SET is the same as for the weighted versions of the problems.

The thing why we needed weights is to simiplify the coloring of G with three colors. It is easy to see that, after replacing weights with copies in ϕ, G remains the same, just vertex weights become replaced with vertex copies. More importantly, no edge is added between different copies of the same vertex. Thus, the new, unweighted graph G' obtained by a reduction from the new, unweighted formula ϕ', can be colored with three colors in the same way as G can. It's just that all vertices in G' that are copies of the same vertex in G receive the same color as their original vertex in G. Vertices of G' becomes partitioned in three independent sets $V(G') = V_1' \sqcup V_2' \sqcup V_3'$, and $|V_1'| = w - 1$. Moreover, by Lemma 2 and Claim 5, finding an independent set of size at least w in G' is an NP-complete problem. All reductions and algorithms provided in the proof are polynomial, and the lemma follows. □

Theorem 6. ABOVE GUARANTEE HAPPY VERTICES *is* NP-*complete even when* $\ell = 3$.

Proof. We reduce from INDEPENDENT SET ABOVE COLORING with $\ell = 3$, that is NP-complete by Lemma 3. Let G be an instance of INDEPENDENT SET ABOVE COLORING. That is, G is colored properly with three colors: $V(G) = V_1 \sqcup V_2 \sqcup V_3$; and it is asked to find an independent set of size at least $\max\limits_{i=1}^{3} |V_i| + 1$ in G.

We construct an instance (G', p, k) of ABOVE GUARANTEE HAPPY VERTICES as follows. Obtain G' as a subdivision of G. Construct the partial coloring p as follows. Left all new vertices appeared after subdivision uncolored. For each other vertex, that is, for each vertex in $V(G)$, precolor it with the same color as it is colored in G. That is, for each $i \in [3]$, and for each $v \in V_i$, put $p(v) = i$. Finally, put $k = \max_{i=1}^{3} |V_i|$. Note that the reduction is done in polynomial time. We formulate that the constructed instance of ABOVE GUARANTEE HAPPY VERTICES is equivalent to the initial instance of INDEPENDENT SET ABOVE COLORING in the following claim.

Claim 6. For any $S \subseteq V(G)$, S is an independent set in G if and only if all vertices in S can be simultaneously happy in (G', p).

Proof of Claim 6. Let S be an independent set in G. We construct a coloring c extending p as follows. For each uncolored vertex e_{uv} of G', that corresponds to the subdivision of the edge $uv \in E(G)$, put

$$c(e_{uv}) = \begin{cases} p(u), & \text{if } u \in S, \\ p(v), & \text{if } v \in S, \\ \text{any}, & \text{otherwise.} \end{cases}$$

Since S is an independent set, $u \in S$ and $v \in S$ never hold simultaneously. Thus, for each $v \in S$ and each $e_{uv} \in N_{G'}(v)$, $c(e_{uv}) = p(v) = c(v)$. Therefore, all vertices in S are happy with respect to c.

In the other direction, let c be a coloring of G' extending p. Firstly, note that no newly-introduced vertex $e_{uv} \in V(G') \setminus V(G)$ can be happy. e_{uv} is adjacent to vertices u and v in G', but $p(u) \neq p(v)$, as uv is an edge of G and p corresponds to a proper coloring of G. Hence, if S is a subset of vertices that are happy with respect to c, then $S \subseteq V(G)$ necessarily. Suppose now that S is not an independent set in G, i.e. $u, v \in S$, but $uv \in E(G)$. Consider the vertex e_{uv} in G'. Since both u and v are happy with respect to c, $c(u) = c(e_{uv})$ and $c(v) = c(e_{uv})$. But $p(u) \neq p(v)$, a contradiction. The proof of the claim is finished. ∎

It is then easy to see that with a trivial extension of p with color i one can obtain exactly $|V_i|$ happy vertices in G'. Hence, k is indeed a number of happy vertices that can be obtained with a trivial extension of p. Finally, finding a coloring yielding at least $k + 1$ happy vertices in (G', p) is equivalent to finding an independent set of size $\max_{i=1}^{3} |V_i| + 1$. Thus, ABOVE GUARANTEE HAPPY VERTICES is NP-complete for $\ell = 3$. □

We now turn onto ABOVE GUARANTEE HAPPY EDGES. We provide a reduction from the following well-known NP-complete problem.

EXACT 3-COVER (X3C) [18, 20]

Input: An integer n, a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ of three-element subsets of $[3n]$.

Question: Is there an exact cover of $[3n]$ with elements of $\mathcal{S}$, i.e. is there a sequence $i_1, i_2, \ldots, i_n$, such that $S_{i_1} \cup S_{i_2} \cup \ldots \cup S_{i_n} = [3n]$?

Theorem 7. ABOVE GUARANTEE HAPPY EDGES *is* NP-*complete.*

Proof. We reduce from the EXACT 3-COVER problem. Let $(n, \mathcal{S} = (S_1, S_2, \ldots, S_m))$ be an instance of X3C. In our reduction, we need n to be an odd number. If n is even, we can always increase n by one and add the set $\{3n + 1, 3n+2, 3n+3\}$ to $\mathcal{S}$ and obtain an equivalent instance of X3C. We also assume that each element of $[3n]$ is contained in at least one set in $\mathcal{S}$. Now we construct an instance (G, p, k) of MHE as follows.

For each integer $i \in [3n]$, introduce a new uncolored vertex u_i to G. Then, for each $i, j \in [3n]$, $i \neq j$, introduce an edge between u_i and u_j in G, so $u_1, u_2, \ldots, u_{3n}$ form a clique in G. These vertices correspond to the elements in $[3n]$.

For each set $S_i \in \mathcal{S}$, introduce $3n$ new vertices in G, namely $v_{i,1}, v_{i,2}, \ldots, v_{i,3n}$. Each of these vertices we precolor with color i, i.e. $p(v_{i,j}) = i$ for each $j \in [3n]$. We also connect each of these vertices to all vertices corresponding to the elements of S_i, i.e. introduce an edge between $v_{i,j}$ and u_t for each $j \in [3n]$ and each $t \in S_i$.

Finally, we introduce a group of new $3(n+1)/2$ vertices $w_1, w_2, \ldots, w_{3(n+1)/2}$ in G. Each of them we precolor with color $m + 1$. We also introduce an edge (w_i, u_j) to G for every $i \in [3(n + 1)/2]$ and every $j \in [3n]$, except for the edge (w_1, u_1). Hence, we introduce $3(n + 1)/2 \cdot 3n - 1$ such edges.

We then set $k = 9n^2 + 3n - 1$ and say that this number of happy edges in G can be obtained by coloring every uncolored vertex with color $m + 1$. Indeed, say that u_j is colored with color $m + 1$ for every $j \in [3n]$. These vertices form clique in G, hence all $\binom{3n}{2}$ edges of the clique are happy. Only edges left that are happy are edges of type (w_i, u_j). Recall that we introduced $3(n + 1)/2 \cdot 3n - 1$ such edges, hence we get that

$$\binom{3n}{2} + 3(n+1)/2 \cdot 3n - 1 = 3n \cdot ((3n - 1)/2 + 3(n + 1)/2) - 1 = 9n^2 + 3n - 1$$

edges are happy in G with respect to such trivial extension of p.

We now argue that $(n, \mathcal{S})$ is a yes-instance of X3C if and only if $(G, p, k+1)$ is a yes-instance of MHE. Note that we use $m + 1$ colors, i.e. $\ell = m + 1$.

Let prove that if $(n, \mathcal{S})$ is a yes-instance of X3C, then $(G, p, k + 1)$ is a yes-instance of MHE. Let $i_1, i_2, \ldots, i_n$ be an answer to $(n, \mathcal{S})$. Then for each $j \in [3n]$, there is a unique $t(j) \in [n]$, such that $t(j) \in \{i_1, i_2, \ldots, i_n\}$ and $j \in S_{t(j)}$. Extend p with a coloring c such that $c(u_j) = t(j)$ for each $j \in [3n]$. We claim that there are exactly $k + 1$ happy edges in G with respect to c.

G consists of edges that have exactly one endpoint of type u_j and edges of type (u_{j_1}, u_{j_2}) for $j_1 \neq j_2$. Each u_j is connected to exactly $3n$ vertices $v_{t(j),1}, v_{t(j),2}, \ldots, v_{t(j),3n}$ of color $t(j)$, since $u_j \in S_{t(j)}$. Hence there are exactly $9n^2$ happy edges in G that have exactly one endpoint of type u_j, with respect to c. Observe that an edge (u_{j_1}, u_{j_2}) is happy with respect to c if and only if $t(j_1) = t(j_2)$, i.e. j_1 and j_2 are covered by the same set $S_{t(j_1)}$. Since the size of every set in $\mathcal{S}$ is exactly three, each u_j is adjacent to exactly two vertices of type $u_{j'}$ of the same color. Thus, there are exactly $3n$ happy edges of type (u_{j_1}, u_{j_2})

in total in G, with respect to c. Happy edges of both types sum up to a total of $9n^2 + 3n$ happy edges.

Let now prove in the other direction. Let c be an optimal coloring of G extending p such that at least $k + 1$ edges are happy in G with respect to c.

Claim 7. For any optimal coloring c of G and any $j \in [3n]$, either $c(u_j) = m+1$ or $u_j \in S_{c(u_j)}$.

Proof of Claim 7. Suppose it's not true, and c is an optimal coloring of G and $c(u_j) \neq m + 1$ and $u_j \notin S_{c(u_j)}$ for some $j \in [3n]$. Then only happy edges that are incident to u_j are edges of the clique, since any other edge incident to u_j has the other endpoint precolored with either color $m + 1$ or color i for any set S_i containing j. Hence, u_j is incident to at most $3n - 1$ happy edges with respect to c.

Now change the color of u_j to any color i such that $j \in S_i$. Such color exists, since we assumed that any element in $[3n]$ is contained in at least one set. u_j is adjacent to $3n$ vertices of color i, hence u_j now is incident to at least $3n$ happy edges. Thus, such change in $c(u_j)$ allows to win at least one happy edge. A contradiction with optimality of c.   ∎

Let s be the number of vertices among $u_1, u_2, \ldots, u_{3n}$ colored with color $m + 1$ in c, i.e. $s = |c^{-1}(m + 1) \cap \{u_1, u_2, \ldots, u_{3n}\}|$. The only other vertices colored with color $m + 1$ are vertices $w_1, w_2, \ldots, w_{3(n+1)/2}$, hence there are at most $\binom{s}{2} + s \cdot 3(n + 1)/2$ happy edges incident to vertices of color $m + 1$.

By Claim 7, every other vertex of type u_j (there are $3n - s$ of them) is colored with a color corresponding to a set containing j. Thus, each such vertex is adjacent to $3n$ precolored vertices of the same color and to at most two vertices of the same color in the clique. Hence, there are at most $(3n - s) \cdot 3n + (3n - s)$ happy edges not incident to vertices of color $m + 1$ in G with respect to c.

In total, we get that at most

$$\binom{s}{2} + s \cdot 3(n+1)/2 + (3n-s) \cdot (3n+1) = 9n^2 + 3n + \binom{s}{2} + s \cdot 3(n+1)/2 - s \cdot (3n+1)$$

edges are happy in G with respect to c. Recall that c is a coloring yielding at least $9n^2 + 3n$ happy edges in G, hence

$$\binom{s}{2} + s \cdot 3(n + 1)/2 - s \cdot (3n + 1) \geq 0,$$

$$s \cdot ((s - 1)/2 + 3(n + 1)/2 - (3n + 1)) \geq 0,$$

$$s \cdot ((s - 1) + 3(n + 1) - (6n + 2)) \geq 0,$$

$$s \cdot (s - 3n) \geq 0.$$

The inequality above holds only when $s \leq 0$ or $s \geq 3n$, but since $s \in \{0, 1, \ldots, 3n\}$, it is either $s = 0$ or $s = 3n$. From the construction of (G, p) we already know that when $s = 3n$, only $k = 9n^2 + 3n - 1$ edges are happy in G,

hence the only option left is $s = 0$. Thus, each vertex of type u_j is colored with a color corresponding to a set containing j, but not with color $m + 1$.

As we observed earlier, each u_j is adjacent to $3n$ precolored vertices of the same color and to at most two vertices of the same color among vertices of the clique. Hence, the only way to obtain $9n^2 + 3n$ happy edges is when each u_j has *exactly* two neighbours of the same color in the clique. Thus, for each color presented among $c(u_j)$, there are exactly three vertices of such color in the clique. This yields a solution of the initial instance of X3C: it is sufficient to take sets with indices in the set $\{c(u_1), c(u_2), \ldots, c(u_{3n})\}$. This finishes the proof of the theorem. □

5   ETH and Set Cover Conjecture based lower bounds

In this section, we show lower bounds for exact algorithms for MHV and MHE, based on the popular Exponential Time Hypothesis and the Set Cover Conjecture. We start with the Set Cover Conjecture and the following problem.

SET PARTITIONING
Input: An integer n, a set family $\mathcal{F} = \{S_1, S_2, \ldots, S_m\}$ over a universe U with $|U| = n$.
Question: Is there a sequence of pairwise disjoint sets $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$ in $\mathcal{F}$, such that $\bigsqcup_{j=1}^{k} S_{i_j} = U$?

Theorem 8 ([8]). *For any $\epsilon > 0$, SET PARTITIONING cannot be solved in time $\mathcal{O}^*((2 - \epsilon)^n)$, unless the Set Cover Conjecture fails.*

Theorem 9. *For any $\varepsilon > 0$, MAXIMUM HAPPY VERTICES cannot be solved in time $\mathcal{O}^*((2 - \varepsilon)^{n'})$, where n' is the number of uncolored vertices, unless the Set Cover Conjecture fails.*

Proof. The proof is by polynomial reduction preserving the size of the universe of the input instance of SET PARTITIONING in the number of uncolored vertices of the resulting instance of MAXIMUM HAPPY VERTICES.

Observe that SET PARTITIONING is a special case of the weighted version of the SET PACKING problem: for each $j \in [m]$, assign weight $|S_j|$ to the set S_j and ask to find a sequence of disjoint sets of summary weight at least n. With this observation, adjust the reduction in the proof of Theorem 2 for this special weighted version of SET PACKING by introducing $|S_j|$ copies of vertex s_j to G instead of just one. This yields the required reduction from SET PARTITIONING to MHV. □

Theorem 10. *For any $\varepsilon > 0$, MAXIMUM HAPPY EDGES cannot be solved in time $\mathcal{O}^*((2 - \varepsilon)^{n'})$, where n' is the number of uncolored vertices, unless the Set Cover Conjecture fails.*

Proof. To prove this theorem we would also like to slightly adjust the reduction used to prove the lack of polynomial kernels (Theorem 3), but there we exploit the restricted version of the problem. The reduction appears possible though, but it is quite more sophisticated than the reduction in the proof of Theorem 3. We now describe this reduction from SET PARTITIONING preserving the size of the universe in the number of uncolored vertices in the resulting instance of MHE. Throughout the proof, we refer to Theorem 3 by saying that we do as usual.

Firstly, we need to get rid of sets consisting of exactly two elements in the initial instance $(U = [n], \mathcal{F} = \{S_1, S_2, \ldots, S_m\})$ of SET PARTITIONING. We assume that $n > 2$. Let $\mathcal{F}_{=2}$ be the subfamily of $\mathcal{F}$ of sets consisting of exactly two elements, i.e. $\mathcal{F}_{=2} = \{S_i \mid S_i \in \mathcal{F}, |S_i| = 2\}$. We want that $\mathcal{F}_{=2} = \emptyset$.

Let it be the other case, $\mathcal{F}_{=2} \neq \emptyset$. Obtain an equivalent instance $(U, \mathcal{F}')$ with $\mathcal{F}'_{=2} = \emptyset$ as follows. Start with removing all sets in $\mathcal{F}_{=2}$ from $\mathcal{F}$. Then, for every pair of sets $S_i \in \mathcal{F}$ and $S_j \in \mathcal{F}_{=2}$, such that $S_i \cap S_j = \emptyset$, add $S_i \sqcup S_j$ back in $\mathcal{F}$. That is, return each set of $\mathcal{F}_{=2}$ back in $\mathcal{F}$, but as a union with some disjoint set in $\mathcal{F}$, for each such possible set. Also, for each triple of pairwise disjoint sets in $\mathcal{F}_{=2}$, add their union in $\mathcal{F}$. Formally,

$$\mathcal{F}' = \mathcal{F} \setminus \mathcal{F}_{=2} \cup \{S_i \sqcup S_j \mid S_i \in \mathcal{F}, S_j \in \mathcal{F}_{=2}, S_i \cap S_j = \emptyset\}$$
$$\cup \{S_i \sqcup S_j \sqcup S_k \mid S_i, S_j, S_k \in \mathcal{F}_{=2}\}. \tag{2}$$

Note that $\mathcal{F}'$ does not contain any set of size two and is constructed in polynomial time.

Claim 8. $(U, \mathcal{F})$ and $(U, \mathcal{F}')$ are equivalent instances of SET PARTITIONING.

Proof of Claim 8. Let $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$ be the answer to $(U, \mathcal{F})$. If the sequence does not contain sets of size two, then all these sets are contained in $\mathcal{F}'$, hence it is also an answer to $(U, \mathcal{F}')$. Otherwise, there is at least one set of size two in the sequence. Let $S_{i_1}, \ldots, S_{i_{k'}}$ be all sets of size two in the sequence, and $S_{i_{k'+1}}, \ldots, S_{i_k}$ be all other sets in the sequence, $k' \geq 1$. If k' is even, then $S_{i_1} \sqcup S_{i_2}, \ldots, S_{i_{k'-1}} \sqcup S_{i_{k'}}, S_{i_{k'+1}}, \ldots, S_{i_k}$ is an answer to $(U, \mathcal{F}')$. If $k' = 1$, then $k \geq 2$, and $S_{i_1} \sqcup S_{i_2}, S_{i_3}, \ldots, S_{i_k}$ is the answer. Otherwise, k' is odd and $k' \geq 3$, and $S_{i_1} \sqcup S_{i_2} \sqcup S_{i_3}, S_{i_4} \sqcup S_{i_5}, \ldots, S_{i_{k'-1}} \sqcup S_{i_{k'}}, S_{i_{k'+1}}, \ldots, S_{i_k}$ is an answer to $(U, \mathcal{F}')$. That is, if $(U, \mathcal{F})$ is a yes-instance, then $(U, \mathcal{F}')$ is a yes-instance.

Proof in the other direction is trivial, since each set of $\mathcal{F}'$ is a disjoint union of a number of sets in $\mathcal{F}$. Thus, the instances are equivalent. ∎

We now assume that $\mathcal{F}$ no longer contains any set of size two. We show how to reduce the instance $(U, \mathcal{F})$ of SET PARTITIONING to an equivalent instance (G, p, k') of MHE. As usual, introduce a clique on n vertices $u_1, u_2, \ldots, u_n$ in G, which vertices correspond to the elements of U. For each set $S_j \in \mathcal{F}$, do the following. Let $d = |S_j|$. Introduce $n^2 - \lfloor \frac{d-1}{2} \rfloor$ copies of a vertex s_j to G. If d is odd, for each $i \in S_j$, add edges between u_i and all $n^2 - \frac{d-1}{2}$ copies of s_j. If d is even, then divide S_j into two equal parts arbitrarily, say, $S_j = S_j^1 \sqcup S_j^2$, $|S_j^1| = |S_j^2| = \frac{d}{2}$. For each $i \in S_j^1$, add edges between u_i and all $n^2 - \frac{d}{2} + 1$ copies

of s_j. But for each $i \in S_j^2$, add edges between u_i and all except one copy of s_j, that is, connect u_i to only $n^2 - \frac{d}{2}$ copies of s_j.

As usual, precolor all copies of s_j with color j for each $j \in [m]$, and leave every vertex u_i uncolored. Finally, set $k' = n^3$. We argue that the constructed instance (G, p, k') is a yes-instance if and only if $(U, \mathcal{F})$ is a yes-instance.

To prove in one direction, let $(U, \mathcal{F})$ be a yes-instance and $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$ be the answer sequence. Construct coloring c extending p as usual, by setting $c(u_i)$ equal to the index of the unique set of the answer that contains i, i.e. $i \in S_{c(u_i)}$, for each $i \in [n]$. Observe that c yields exactly n^3 happy edges in G. c splits the clique of G into k groups $V_{i_1}, V_{i_2}, \ldots, V_{i_k}$, and each group contains vertices corresponding to the elements of the respective set, i.e. $V_{i_t} = \{u_i \mid i \in S_{i_t}\}$ for each $t \in [k]$. Take any group V_{i_t}, and let $d = |V_{i_t}| = |S_{i_t}|$. If d is odd, then exactly $d \cdot (n^2 + \frac{d-1}{2}) = dn^2 + \binom{d}{2}$ edges that have exactly one endpoint in V_{i_t} are happy. If d is even, the number of such happy edges equals $\frac{d}{2} \cdot (n^2 - \frac{d}{2} + 1) + \frac{d}{2} \cdot (n^2 - \frac{d}{2}) = dn^2 - \binom{d}{2}$ as well. The only other edges are $\binom{d}{2}$ edges inside V_{i_t}. All vertices in V_{i_t} are of color i_t, hence all these $\binom{d}{2}$ edges are happy. Thus, vertices in V_{i_t} are incident to exactly $|V_{i_t}| \cdot n^2$ happy edges. Since no two groups can share endpoints of the same happy edge, we get a total of $\sum |V_{i_t}| \cdot n^2 = n^3 = k'$ happy edges in G with respect to c. Hence, (G, p, k') is a yes-instance of MHE.

We start the proof in the other direction with a claim identical to Claim 1.

Claim 9. In any optimal coloring c of G extending p, $i \in S_{c(u_i)}$ for each $i \in [n]$.

Proof of Claim 9. Note that $n^2 - \lfloor \frac{n-1}{2} \rfloor > n - 1$ for $n > 2$, then as usual. ∎

The following claim bounds the number of happy edges incident to vertices in each color group of the clique in G.

Claim 10. Take any optimal coloring c of (G, p). Let $V_t = c^{-1}(t) \cap \{u_1, u_2, \ldots, u_n\}$ be the set of vertices of the universe clique of G that are colored with color t. Let $d = |V_t|$. Then vertices in V_t correspond to elements in S_t, are incident to at most dn^2 happy edges, and the bound of dn^2 is reached if and only if $d = |S_t|$ or $d = 0$.

Proof of Claim 10. Take any optimal coloring c of (G, p) and $t \in [\ell]$. Let $d = |V_t|$. If $d = 0$, then the claim statement holds true. Assume now that $d > 0$ and $|V_t|$ is not empty.

By Claim 9, all vertices in V_t correspond to elements in S_t. Let $D = |S_t|$. Suppose D is odd. Then each vertex in V_t is connected to $n^2 - \frac{D-1}{2}$ copies of s_t. Hence, exactly $dn^2 - d \cdot (D-1)/2$ edges going outside of V_t are happy with respect to c. All edges inside V_t are happy, so V_t is incident to $dn^2 - d \cdot (D-1)/2 + \binom{d}{2}$ happy edges. Since $d \leq D$, this number is not greater than dn^2 and equals dn^2 if and only if $d = D = |S_t|$. Thus, the claim statement is true for odd D.

Otherwise, D is even and $D \geq 4$, since we removed all sets of size two from the initial instance. Recall that for sets of even size, we split them into two halves and connected them to a different number of copies of s_t. Let $d = d_1 + d_2$

($d_1, d_2 \leq \frac{D}{2}$), so d_1 vertices in V_t are connected to $n^2 - \frac{D}{2} + 1$ copies of s_t, and the other d_2 vertices in V_t are connected to $n^2 - \frac{D}{2}$ copies of s_t. Then, V_t is incident to exactly $n^2 - d \cdot D/2 + d_1$ happy edges going outside of V_t. Only happy edges left are the $\binom{d}{2}$ edges inside V_t.

Suppose that vertices in V_t are incident to at least dn^2 happy edges in total, so $d \cdot D/2 - d_1 \leq \binom{d}{2}$. Equivalently, $d \cdot (D - (d-1)) \leq 2d_1$. Since $2d_1 \leq D$, we get that $d \cdot (D - d + 1) \leq D$, or $d^2 - Dd + (D - d) \geq 0$. The quadratic polynomial has roots in $d = 1$ and $d = D$, and the inequality holds when $d \leq 1$ or $d \geq D$. If $d \leq 1$, then $d = 1$. But then $d \cdot D/2 - d_1 = D/2 - d_1 \geq 1 > \binom{d}{2} = 0$. Thus, $d = D = |S_t|$.

The proof is finished. Note that if we would have $D = 2$, V_t could consist of one vertex connected to $n^2 - \frac{D}{2} + 1 = n^2$ copies of s_t, and the claim statement would fail. ∎

Let now (G, p, k') be a yes-instance, and c be an optimal coloring of (G, p). At least n^3 edges are happy with respect to c in G. By Claim 10, it follows that exactly n^3 edges are happy in G with respect to c, and each color group V_t provides exactly $|V_t| \cdot n^2$ happy edges. Hence, $|V_t| = |S_t|$ for each non-empty group V_t. Thus, the color partition of the vertices of the universe clique of G corresponds to a partition of the universe U into sets in $\mathcal{F}$. So $(U, \mathcal{F})$ is a yes-instance of SET PARTITIONING. We obtained the desired reduction from SET PARTITIONING to MHE. This finishes the whole proof. □

We now turn onto ETH-based lower bounds.

Theorem 11. MAXIMUM HAPPY VERTICES *with $\ell = 3$ cannot be solved in time $2^{o(n+m)}$, unless ETH fails.*

Proof. We reuse reductions discussed above in the proofs in Section 4. In the proofs of Lemma 2 and Lemma 3, we obtained a chain of linear reductions from 3-SAT to INDEPENDENT SET ABOVE COLORING. Then, in the proof of Theorem 6 we showed that INDEPENDENT SET ABOVE COLORING can be reduced linearly to ABOVE GUARANTEE HAPPY VERTICES. Since this problem is a special case of MHV, it follows that MHV cannot be solved in $2^{o(n+m)}$ time under ETH.□

We now prove another computational lower bound for MHV that is based on the reduction from INDEPENDENT SET to MHV discussed above in the proofs of Theorem 5 and Theorem 6. This reduction also implies some approximation lower bounds.

Theorem 12. MAXIMUM HAPPY VERTICES *cannot be solved in $\mathcal{O}(n^{o(k)})$ time, unless ETH fails. Also, for any $\epsilon > 0$, MAXIMUM HAPPY VERTICES cannot be approximated within $\mathcal{O}(n^{\frac{1}{2}-\epsilon})$, $\mathcal{O}(m^{\frac{1}{2}-\epsilon})$, $\mathcal{O}(h^{1-\epsilon})$ or $\mathcal{O}(\ell^{1-\epsilon})$ in polynomial time, unless $P = NP$.*

Proof. It is a well-known result that, assuming ETH, both CLIQUE and INDEPENDENT SET cannot be solved in $n^{o(k)}$ time [4, 5, 22]. As discussed in the proofs

of Theorem 5 and Theorem 6, there is a polynomial reduction from INDEPEN-
DENT SET to MHV.

Given an instance (G, k) of INDEPENDENT SET, it is enough to precolor
each vertex of G with an unique color and to subdivide each edge of G, thus
obtaining an equivalent instance (G', p, k) of MHV. G' is a subdivision of G,
so $|V(G')| = \mathcal{O}(n^2)$, where $n = |V(G)|$. Thus, an algorithm with running time
$n^{o(k)}$ for MHV would imply an algorithm with running time $n^{2o(k)} = n^{o(k)}$ for
INDEPENDENT SET. This proves that MHV cannot be solved in $n^{o(k)}$ time under
ETH.

The approximation guarantee lower bounds for MHV follow from the in-
approximability of CLIQUE and INDEPENDENT SET, as both of these problems
cannot be approximated within $\mathcal{O}(n^{1-\epsilon})$, unless $P \neq NP$ [29]. We use the same
reduction from INDEPENDENT SET to MHV, and note that $|E(G')| = \mathcal{O}(n^2)$
and $h = \ell = |V(G)| = n$. This finishes the proof. $\square$

Theorem 13. MAXIMUM HAPPY EDGES *with $\ell = 3$ cannot be solved in time*
$2^{o(n+m)}$, *unless ETH fails.*

Proof. In their work on multiterminal cuts [11], Dahlhahus et al. showed NP-
completeness of 3-TERMINAL CUT (equivalently, MAXIMUM HAPPY EDGES with
$\ell = 3$) by a linear reduction from the MAX CUT problem. The MAX CUT problem
definition is given below.

MAX CUT
Input: A graph G and an integer k.
Question: Can vertices of G be partitioned into two sets $V(G) = V_1 \sqcup V_2$, so
that the number of edges between V_1 and V_2 in G is at least k, i.e.
$|E(V_1, V_2)| \geq k$?

The reduction they give is linear, so it is sufficient to prove that MAX CUT
cannot be solved in $2^{o(n+m)}$ time, unless ETH fails. Although this result may
be well-known, we have not found any explicit statement about that MAX CUT
cannot be solved in subexponential time. For completeness, we state it here.

Lemma 4. MAX CUT *cannot be solved in time $2^{o(n+m)}$, unless ETH fails.*

Proof. This result can be obtained by following the series of classical reductions
by Papadimitriou and Yannakakis in [24]. They reduce an instance of MAX 3-
SAT to an instance of MAX 3-SAT that contains at most three occurences of
each variable, then to INDEPENDENT SET, MAX 2-SAT, then to MAX NOT-
ALL-EQUAL 3-SAT and finally to MAX CUT. All reductions they provide are
linear, so no problem in this chain can be solved in subexponential time. $\square$

We finish the proof by combining Lemma 4 and the linear reduction in [11]
from MAX CUT to 3-TERMINAL CUT. $\square$

6   Algorithms

In this section, we present two algorithms solving MHV or MHE. We start with
a randomized algorithm for MHV that runs in $\mathcal{O}^*(\ell^k)$ time and recognizes a

yes-instance and finds the required coloring with a constant probability. The algorithm is based on the following lemma.

Lemma 5. *Let (G, p) be a graph with precoloring, and $P = \bigcup\limits_{i=1}^{\ell} \mathcal{H}_i(G, p)$. Let c be a coloring that yields the maximum possible number of happy vertices in (G, p), and let $H = \mathcal{H}(G, c)$ be the set of these vertices. Then $|H \cap P| \geq \frac{1}{\ell} \cdot |P|$.*

Proof. Let k be the maximum possible number of vertices that can be happy simultaneously in (G, p), so (G, p, k) is a yes-instance of MHV, and $(G, p, k + 1)$ is not. Let i be such that $|\mathcal{H}_t(G, p)|$ is maximum possible. In particular, $|\mathcal{H}_t(G, p)| \geq \frac{1}{\ell} \cdot |P|$. Construct a coloring c' of (G, p) by trivially extending p with the color t. Note that all vertices that are not happy in G with respect to c' are contained in $\bigcup\limits_{i \in [\ell] \setminus \{t\}} \mathcal{H}_i(G, p) = P \setminus \mathcal{H}_t(G, p)$.

Let $U = \mathcal{H}(G, p) \setminus P$. All vertices in U and all vertices in $\mathcal{H}_t(G, p)$ are happy in G with respect to c'. Let $H' = \mathcal{H}(G, c')$ be the set of all vertices that are happy in G with respect to c'. Then $U \cap \mathcal{H}_t(G, p) \subseteq H'$, and $|H'| = |H' \cap U| + |H' \cap P| \geq |U| + \frac{1}{\ell} \cdot |P|$.

Take now a coloring c that yields the maximum possible number of happy vertices in (G, p), and let $H = \mathcal{H}(G, c)$ be the set of these vertices. In particular, $|H| \geq |H'|$. Suppose $|H \cap P| < \frac{1}{\ell} \cdot |P|$. But $|H| = |H \cap U| + |H \cap P| < |U| + \frac{1}{\ell} \cdot |P| \leq |H'|$. This contradiction finishes the proof. □

Theorem 14. *There is a $\mathcal{O}^*(\ell^k)$ running time randomized algorithm for MAXIMUM HAPPY VERTICES.*

Proof. Firstly, we provide a procedure that finds an answer for a given instance (G, p, k) of ℓ-MHV with a probability of at least ℓ^{-k}, if (G, p, k) is a yes-instance. The procedure is given in Fig. 1.

Claim 11. `guess_answer`(G, p, k) always outputs a set of vertices that can be happy simultaneously in (G, p).

Proof of Claim 11. We prove that by induction on k. For $k = 0$, the procedure returns U in line 4, and all vertices in this set can be happy simultaneously in (G, p) (for example, with any trivial extension of p). Let now $k > 0$ and the claim statement hold for $k - 1$. Consider in which line the procedure returns for (G, p, k). If it is line 4 or line 6, then the claim statement holds. Consider the only case left, when procedure returns in line 12.

Let H' be the set returned by `guess_answer`$(G \setminus \{v\}, p', k - 1)$. By induction, all vertices in H' can be happy simultaneously in $(G \setminus \{v\}, p')$. Note that all vertices in $H' \cup \{v\}$ can be happy simultaneously in (G, p) as well, since p' is just a restriction of p that ensures that all neighbours of v are colored with the same color as v itself. The claim statement follows immediately. ∎

Claim 12. *For any yes-instance (G, p, k) of ℓ-MHV, the `guess_answer` procedure outputs a set H with $|H| \geq k$ with a probability of at least ℓ^{-k}.*

Algorithm: guess_answer(G, p, k)

Input: An instance (G, p, k) of ℓ-MHV.
Output: A set $H \subseteq \mathcal{H}(G, p)$ such that all vertices in H can be happy in (G, p)
          simultaneously.

1   $P \longleftarrow \bigcup\limits_{i=1}^{\ell} \mathcal{H}_i(G, p)$
2   $U \longleftarrow \mathcal{H}(G, p) \setminus P$
3   **if** $k \leq |U|$ **then**
4 | **return** U
5   **if** $P = \emptyset$ **then**
6 | **return** $\emptyset$
7   $v \longleftarrow$ random vertex in P, each with equal probability
8   $p' \longleftarrow p|_{V(G) \setminus \{v\}}$
9   $i \longleftarrow$ the color such that $v \in \mathcal{H}_i(G, p)$
10 **foreach** $u \in N(v)$ **do**
11 | $p'(u) \longleftarrow i$
12 **return** guess_answer$(G \setminus \{v\}, p', k - 1) \cup \{v\}$

Fig. 1. A randomized procedure finding a set of vertices that can be happy simultaneously.

Proof of Claim 12. The proof is by induction on k. For $k = 0$, the claim statement holds, since any output suffices. Let now $k > 0$ and let the claim statement hold true for $k - 1$. Finally, let (G, p, k) be a yes-instance of MHV. Consider how the procedure processes the instance. If it returns in line 4, having that $|U| \geq k$, it outputs a required set with the probability of 1. Note that the procedure can't return in line 6, since (G, p, k) is a yes-instance. Hence, the only case left is that the procedure returns in line 12. We now consider this case.

Fix any optimal coloring c of (G, p). In particular, c yields at least k happy vertices in (G, p), so $|\mathcal{H}(G, c)| \geq k$. By Lemma 5, $|\mathcal{H}(G, c) \cap P| \geq \frac{1}{\ell} \cdot |P|$. Hence, in line 7, the procedure chooses v such that $v \in \mathcal{H}(G, c)$, with a probability of at least ℓ^{-1}. Consider the case that the procedure indeed chooses such v, so $v \in \mathcal{H}(G, c)$. Since v is precolored and is happy with respect to c, for any neighbour u of v holds $c(u) = c(v) = p(v)$. Thus, $c|_{V(G) \setminus \{v\}}$ is a coloring extending p' in $G \setminus \{v\}$. Moreover, $\mathcal{H}(G \setminus \{v\}, c|_{V(G) \setminus \{v\}}) = \mathcal{H}(G, c) \setminus \{v\}$. Therefore, $(G \setminus \{v\}, p', k - 1)$ is a yes-instance. By induction, guess_answer$(G \setminus \{v\}, p', k - 1)$ returns a set H' with $|H'| \geq k - 1$ and all vertices in H' can be happy simultaneously in $(G \setminus \{v\}, p')$, with a probability of at least ℓ^{-k+1}. By Claim 11, all vertices in $H' \cup \{v\}$ can be happy in (G, p) simultaneously. Recall that $v \in \mathcal{H}(G, c)$ with a probability of at least ℓ^{-1}, and obtain the total probability of at least ℓ^{-k}. $\blacksquare$

From the claims above immediately follows that a single launch of the procedure finds that the given instance is a yes-instance with a probability of at least ℓ^{-k}, and never finds that if the given instance is a no-instance. We finish

the construction of the randomized algorithm by saying that it repeats the procedure for ℓ^k times for the given instance, so it recognizes a yes-instance with a constant probability of at least e^{-1}, and never recognizes a no-instance as a yes-instance.   □

Note that by Theorem 12, no algorithm with running time $\mathcal{O}(\ell^{o(k)})$ exists for MHV, unless ETH fails. Similarly, no $\mathcal{O}(\ell^{o(k)})$ running time *randomized* algorithm exists for MHV under the *randomized* ETH [12]. The algorithm given above is optimal in that sence.

We now turn onto MHE and give an exact algorithm with $\mathcal{O}^*(2^k)$ running time for this problem. In its turn, this algorithm optimal in a sence that no $2^{o(k)}$ running time algorithm exists for MHE under ETH (see Theorem 13). The algorithm relies on the following kernelization result. We note that this kernelization result and an algorithm with the running time of $\mathcal{O}^*(2^k)$ was already presented by Aravind et al. in [3]. We believe that our kernelization algorithm is short and somewhat simpler, since it relies on a single reduction rule.

Theorem 15 ([3]). MAXIMUM HAPPY EDGES *admits a kernel with at most k uncolored vertices.*

Proof. Let (G, p, k) be an instance of MHE. We show how to obtain an equivalent instance (G', p', k') of MHE, where the number of uncolored vertices in (G', p') is at most k' and $k' \leq k$.

The kernelization algorithm consists just of applying the following reduction rule exhaustively to (G, p, k).

Reduction rule 1 *If there is a connected component C consisting only of uncolored vertices in (G, p), remove it from G and reduce k by the number of edges in C. That is, replace instance (G, p, k) with an instance $(G \setminus C, p|_{V(G) \setminus C}, k - |E(G[C])|)$.*

The correctness of the reduction rule follows from the fact that one can color a connected component of uncolored vertices with the same single color and make all edges in the component happy.

Claim 13. After the exhaustive application of Reduction rule 1, if G' contains at least k' uncolored vertices, then (G', p', k') is a yes-instance.

Proof of Claim 13. If $k' \leq 0$, (G', p', k') is trivially a yes-instance. Suppose now $k' \geq 1$ and G' contains at least k' uncolored vertices. We construct a coloring of (G', p') that yields at least k' happy edges.

Take any uncolored vertex in (G', p'), say v, such that v has at least one precolored neighbour in (G', p'). Note that such choice of v always exists, otherwise Reduction rule 1 would be applied.

Let u be a precolored neighbour of v. Then, set the color of v to $p(u)$. Edge uv becomes happy, and still no connected component in G' consists only of uncolored vertices. Thus, we can take an uncolored vertex with a precolored neighbour again. Repeat this procedure until no uncolored vertex remains in

G'. The procedure is repeated at least k' times, and each time a happy edge is obtained, so (G', p', k') is a yes-instance. ∎

The statement of the theorem follows directly from the claim. □

Theorem 16 ([3]). *There is a $\mathcal{O}^*(2^k)$ running time algorithm for* MAXIMUM HAPPY EDGES.

Proof. As shown by Aravind et al. in [3], MHE can be solved in time $\mathcal{O}^*(2^{n'})$, where n' is the number of uncolored vertices. By Theorem 15, we can assume that $n' \leq k$, and the statement follows. □

References

1. Agrawal, A.: On the parameterized complexity of happy vertex coloring. In: International Workshop on Combinatorial Algorithms. pp. 103–115. Springer (2017)
2. Aravind, N., Kalyanasundaram, S., Kare, A.S.: Linear time algorithms for happy vertex coloring problems for trees. In: International Workshop on Combinatorial Algorithms. pp. 281–292. Springer (2016)
3. Aravind, N., Kalyanasundaram, S., Kare, A.S., Lauri, J.: Algorithms and hardness results for happy coloring problems. arXiv preprint arXiv:1705.08282 (2017)
4. Chen, J., Chor, B., Fellows, M., Huang, X., Juedes, D., Kanj, I.A., Xia, G.: Tight lower bounds for certain parameterized NP-hard problems. Information and Computation **201**(2), 216–231 (2005)
5. Chen, J., Huang, X., Kanj, I.A., Xia, G.: Strong computational lower bounds via parameterized complexity. Journal of Computer and System Sciences **72**(8), 1346–1367 (2006)
6. Choudhari, J., Reddy, I.V.: On structural parameterizations of happy coloring, empire coloring and boxicity. In: WALCOM: Algorithms and Computation, pp. 228–239. Springer International Publishing (2018)
7. Cook, S.A.: The complexity of theorem-proving procedures. In: Proceedings of the third annual ACM symposium on Theory of computing - STOC '71. ACM Press (1971)
8. Cygan, M., Dell, H., Lokshtanov, D., Marx, D., Nederlof, J., Okamoto, Y., Paturi, R., Saurabh, S., Wahlström, M.: On problems as hard as CNF-SAT. ACM Transactions on Algorithms **12**(3), 1–24 (2016)
9. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Lower bounds for kernelization. In: Parameterized Algorithms, pp. 523–555. Springer International Publishing (2015)
10. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized algorithms, vol. 3. Springer (2015)
11. Dahlhaus, E., Johnson, D.S., Papadimitriou, C.H., Seymour, P.D., Yannakakis, M.: The complexity of multiterminal cuts. SIAM Journal on Computing **23**(4), 864–894 (1994)
12. Dell, H., Husfeldt, T., Marx, D., Taslaman, N., Wahlén, M.: Exponential time complexity of the permanent and the tutte polynomial. ACM Transactions on Algorithms **10**(4), 1–32 (2014)
13. Dell, H., Marx, D.: Kernelization of packing problems. In: Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics (2012)

14. Dell, H., Melkebeek, D.V.: Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. Journal of the ACM **61**(4), 1–27 (2014)
15. Diestel, R.: Graph theory. Springer Publishing Company, Incorporated (2018)
16. Dom, M., Lokshtanov, D., Saurabh, S.: Kernelization lower bounds through colors and IDs. ACM Transactions on Algorithms **11**(2), 1–20 (2014)
17. Gao, H., Gao, W.: Kernelization for maximum happy vertices problem. In: Latin American Symposium on Theoretical Informatics. pp. 504–514. Springer (2018)
18. Garey, M., Johnson, D., Stockmeyer, L.: Some simplified NP-complete graph problems. Theoretical Computer Science **1**(3), 237–267 (1976)
19. Hermelin, D., Wu, X.: Weak compositions and their applications to polynomial lower bounds for kernelization. In: Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics (2012)
20. Karp, R.M.: Reducibility among combinatorial problems. In: Complexity of Computer Computations, pp. 85–103. Springer US (1972)
21. Lewis, R., Thiruvady, D., Morgan, K.: Finding happiness: An analysis of the maximum happy vertices problem. Computers & Operations Research **103**, 265–276 (2019)
22. Lokshtanov, D., Marx, D., Saurabh, S., et al.: Lower bounds based on the exponential time hypothesis. Bulletin of EATCS **3**(105) (2013)
23. Misra, N., Reddy, I.V.: The parameterized complexity of happy colorings. In: International Workshop on Combinatorial Algorithms. pp. 142–153. Springer (2017)
24. Papadimitriou, C.H., Yannakakis, M.: Optimization, approximation, and complexity classes. Journal of Computer and System Sciences **43**(3), 425–440 (1991)
25. Xu, Y., Goebel, R., Lin, G.: Submodular and supermodular multi-labeling, and vertex happiness. CoRR (2016)
26. Zhang, P., Jiang, T., Li, A.: Improved approximation algorithms for the maximum happy vertices and edges problems. In: International Computing and Combinatorics Conference. pp. 159–170. Springer (2015)
27. Zhang, P., Li, A.: Algorithmic aspects of homophyly of networks. Theoretical Computer Science **593**, 117–131 (2015)
28. Zhang, P., Xu, Y., Jiang, T., Li, A., Lin, G., Miyano, E.: Improved approximation algorithms for the maximum happy vertices and edges problems. Algorithmica **80**(5), 1412–1438 (2018)
29. Zuckerman, D.: Linear degree extractors and the inapproximability of max clique and chromatic number. In: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing - STOC '06. ACM Press (2006)