A Complexity-Theoretic Analysis of Majority Illusion in Social Networks

UMBERTO GRANDI, Université Toulouse Capitole, France
LAWQUEEN KANESH, Indian Institute of Technology Indore, India
GRZEGORZ LISOWSKI, AGH University, Poland
M.S. RAMANUJAN, University of Warwick, United Kingdom
PAOLO TURRINI, University of Warwick, United Kingdom

Majority illusion occurs in a social network when the majority of the network vertices belong to a certain type but the majority of each vertex's neighbours belong to a different type, therefore creating the wrong perception, that is, the illusion, that the majority type is different from the actual one. From a system engineering point of view, this motivates the search for algorithms to detect and, where possible, correct this often undesirable phenomenon. In this we provide a computational study of majority illusion in social networks, paying particular attention to the problem of its verification, that is, whether majority illusion can occur on social networks, and elimination, that is, how can we eliminate majority illusion by social network rewiring. While we show that the problems we consider are generally **NP**-complete, we also provide a parameterised complexity analysis, showing FPT-algorithms for the detection problem and **W**[1]-hardness for the elimination problem, using natural graph-theoretic parameters.

JAIR Associate Editor: Nisarg Shah

JAIR Reference Format:
Umberto Grandi, Lawqueen Kanesh, Grzegorz Lisowski, M.S. Ramanujan, and Paolo Turrini. 2025. A Complexity-Theoretic Analysis of Majority Illusion in Social Networks. *Journal of Artificial Intelligence Research* 83, Article 26 (August 2025), 37 pages.
DOI: 10.1613/jair.1.17741

1 Introduction

The impact of peer pressure in people's beliefs is well documented. In a famous experiment, Solomon [3] showed how a person's perception of the length of a stick converges towards the aggregate opinion of the person's "neighbours". The effects of such distortions can critically be felt in the context of collective decision-making, where debates, resolutions and even elections can be decided by individuals' misperception of what others think. In social networks in particular, the tendency of individuals to conform to their immediate connections can be exploited by strategic positioning, and a well-placed minority view can quickly become what most people come to believe [40]. In other words, when individuals use their social network as a source of information, it may be the case that minority groups are more "visible" as a result of being better placed. This makes them over-represented, and even appear to be majorities in many friendships' groups – a phenomenon known as *majority illusion*.

Authors' Contact Information: Umberto Grandi, ORCID: 0000-0002-1908-5142, umberto.grandi@irit.fr, Université Toulouse Capitole, Toulouse, France; Lawqueen Kanesh, ORCID: 0000-0001-9274-4119, lawqueenkanesh091@gmail.com, Indian Institute of Technology Indore, Indore, India; Grzegorz Lisowski, ORCID: 0009-0007-1604-8999, grzegorz.adam.lisowski@gmail.com, AGH University, Kraków, Poland; M.S. Ramanujan, ORCID: 0000-0002-2116-6048, R.Maadapuzhi-Sridharan@warwick.ac.uk, University of Warwick, Coventry, United Kingdom; Paolo Turrini, ORCID: 0000-0003-4609-1051, p.turrini@warwick.ac.uk, University of Warwick, Coventry, United Kingdom.

DOI: 10.1613/jair.1.17741

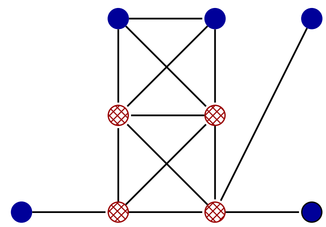

Fig. 1. An instance in which all agents are under majority illusion. The well-placed red (shaded) minority is perceived as majority by everyone.

Majority illusion was originally introduced by Lerman et al. [36], who studied the existence of social networks in which most agents belong to a certain binary type, but most of their peers belong to a different one. Thus, they acquire the wrong perception, that is, the illusion, that the majority type is different from the actual one. Figure 1 shows an example of this.

Majority illusion has important consequences when paired with opinion formation. If, for example, individuals change their mind based on what their friends say, for instance following a threshold model [26], then majority illusion means that strategically placed minorities may well become stable majorities. As such it is important to predict its occurrence in a network and, crucially, to eliminate it when undesirable.

The graph structure of majority illusion was analysed by Lerman et al. [36], who studied network features that correlate with having many individuals under illusion. They demonstrated how disassortative networks, that is, those in which highly connected agents tend to link with lowly connected ones, increase the chances of majority illusion. However, no algorithms have yet been provided to check whether majority illusion can occur in a social network. This is especially important for checking if the structure of a network is protected from the possibility of majority illusion due to its topology.

Likewise, the approach of eliminating undesirable properties by network transformation is not new, and extensively pursued in the context of election manipulation [12], influence maximisation [44], anonymisation [31] and of k-core maximisation [14, 45]). Looking at it from an online social network perspective, adding or removing edges can be a way of determining which posts to show to a user, presenting them with the views held by others. However, such natural operations have yet to be studied in the context of eliminating majority illusion.

All in all, the computational questions of checking whether a network admits majority illusion and how this can be eliminated are still unexplored.

Our Contribution. In this paper we initiate the algorithmic analysis of majority illusion in social networks, focusing on two computational questions. We first consider the problem of verifying illusion, that is, deciding whether there is a labelling of the vertices such that a majoritarian fraction of agents are under illusion, and we prove it to be **NP**-complete[1]. In fact, we show that checking the existence of a labelling in which a fraction q of agents is under illusion is computationally hard for every rational $q \in (\frac{1}{2}, 1]$. Our **NP**-hardness proof techniques also imply **NP**-hardness on bipartite networks, planar networks, networks of constant maximum degree and networks of constant c-closure.

In light of these negative results, we aim to identify tractable restrictions of the problem by carrying out a parameterised complexity analysis involving well-established graph width measures and their variants. First, we turn our attention to tree width, which measures how "close" a graph is to a tree (we refer to [16] for an overview of the applicability of tree width). We obtain a fixed-parameter algorithm (FPT algorithm) for verifying majority illusions,parameterised by the maximum degree of the network plus its tree width, as well as by the size of the

[1]We note that the labels we consider can be seen, for example, as agents' *opinions* or *preferences*.

minimum vertex cover. Along the way, we show that for every constant value of the network's tree width, the problem can be solved in polynomial time (i.e., an XP algorithm parameterised by the tree width). These two results are of specific interest to sparse networks , in particular due to the bounded number of edges in trees. We then also consider dense networks by parameterising by the neighbourhood diversity of the input network. This parameter, for which we provide an FPT algorithm, measures what is the minimum number of groups of vertices being either cliques or independent sets, into which one can decompose a graph. We note that this parameter is especially relevant for dense networks , as networks with a small neighbourhood diversity can still have a high density of edges.

Finally we move to the problem of eliminating majority illusions, which we model as edge transformation by bounded Hamming distance. We show this problem to be **NP**-complete in general and **W**[1]-hard when parameterised by the number of modified edges. Table 1 shows an overview of our parameterised complexity results.

Related Work. Our results are grounded in a number of research lines in artificial intelligence, notably those dealing with the computational analysis of agent interaction and collective decision-making.

Opinion Manipulation. Our work is directly related to computational models of social influence. The closest work is that of Auletta et al. [5], who identify networks and initial distributions of opinions such that an opinion can become a consensus opinion following local majority updates. Observe in this respect that when all vertices are under majority illusion, a synchronous majoritarian update causes an initial minority to evolve into a consensus in just one step. Other notable models include the work of Doucette et al. [19] who studied the propagation of possibly incorrect opinions with an objective truth value in a social network, and the stream of papers studying the computational aspects of exploiting (majoritarian) social influence via opinion transformation (see [32, 10, 5, 11]). It is also worth mentioning that our research is related to preventing the negative effects of information spread [2, 7].

Election Manipulation on Networks. In recent literature growing attention has been devoted to the problem of manipulating the results of voting using voters' connections in social networks. As noted in [17], the growing role of technology poses new abilities for political campaigns. In the multi-agent systems literature, for instance, in [15] it has been studied how to select a subset of targeted voters whose change of opinion might make a certain candidate win after communication. Similarly, for example Wilder and Vorobeychik [43] studied how an external manipulator having a limited budget can select a set of agents to directly influence, to obtain a desired outcome of elections. In a similar setting, Faliszewski et al. [20] studied "bribes" of voters' clusters. In Section 4 we take a similar approach, with the specific objective of eliminating a majority illusion.

There are also various other accounts of paradoxical effects in social networks which are related to our work, such as the *friendship paradox*, according to which, on average, individuals are less well-connected than their friends (see [28, 1]).

Another interesting line of research involves the connections between distortion in social networks and *social good games*. In [24] it is investigated how agents' connections in a network might lead to otherwise impossible non-trivial equilibria in such games. We note that majority illusion is especially relevant in this context, as agents that are under illusion have a reversed view on which action is the most profitable for them in such games. Furthermore, exploiting social networks paradox's, Santos et al. [39] recently showed how false consensus leads to the lack of participation in team efforts.

Paper Structure. Section 2 provides the basic setup and definitions. Section 3 focuses on checking whether illusion can occur in a network while Section 4 studies illusion elimination. Then, Section 5 provides a discussion of the impact of results shown in this paper. Section 6 concludes the paper presenting various potential future directions.

Table 1. Summary of the main parameterised complexity results. Here, ND denotes neighbourhood diversity, Δ is the maximum degree, tw is the tree width, cw is the clique width and VC denotes vertex cover number.

	Parameters
Majority Illusion	
FPT	Δ + tw Thm. 3, Δ + cw Cor. 2, ND Thm. 4, VC Cor. 3
XP	tw Cor. 1
Para-NP-Hard	Δ Obs. 2, c-closure Obs. 1, max-clique-size Obs. 3
Illusion Removal	
W[1] – Hard	Changed Edges Thm. 5, Thm. 6

2 Preliminaries

Let us define the key notions that we use in the paper. First, in Section 2.1, we will describe the framework. Then, in Section 2.2 we provide crucial notions in parameterised complexity.

2.1 The Model

Our model features a set N of agents, connected in a graph (N, E), with $E \subseteq N^2$. For convenience, we also denote $|N|$ as n. Throughout the paper we will consider *undirected graphs*, that is, we require E to be symmetric. Furthermore, we assume that E is *irreflexive*, that is, that E does not include self-loops. We call such a graph a *social network*. For $i \in N$ we denote as $N(i) = \{j \in N : (i, j) \in E\}$ the set of agents that i is following. We also say that a network (N', E') is a *subnetwork* of (N, E), if $N' \subseteq N$ and $E' \subseteq E$. We assume that each of the agents on the network has an opinion, which we model as a labelling. Throughout the paper we assume a binary set of labels $\{b, r\}$ (*blue* and *red*, respectively).

Definition 1 (Labelled Social Network). *A* labelled social network *is a tuple (N, E, f), where (N, E) is a social network and $f : N \longrightarrow \{b, r\}$ is a* labelling *which assigns one of the two values to each agent.*

In a network where every vertex is labeled blue or red, the *blue surplus* of a vertex is the number of its blue neighbours minus the number of its red neighbours , and the blue surplus of a network is the difference between the total number of agents labelled blue and the total number of agents labelled red . The red surplus is defined analogously. For a vertex set X and a labelling $f : X \longrightarrow \{b, r\}$, we define the *red neighbourhood of a vertex i under f* as the set of neighbours of i in X that are assigned the label r by f, and this set is denoted by $N^X_{f,r}(i)$. We drop the explicit reference to X or f in this notation if clear from the context. The analogue of this definition for blue neighbourhood is symmetric. Furthermore, given a labelling f of a social network (N, E), we denote the set of red vertices $\{i \in N : f(i) = r\}$ as R_f and the set of blue vertices $\{i \in N : f(i) = b\}$ as B_f. Moreover, for a set $S \subseteq N$, R^S_f is the set of red vertices in S, while B^S_f is the set of blue vertices in S. We omit f if clear from the context.

Majority Illusion. A label $l \in \{b, r\}$ is a (strict) majority winner in a labelled social network (N, E, f) if there are strictly more vertices labelled with l than with $\{b, r\} \setminus \{l\}$ We use $W_{(N,E,f)}$ to denote such a winner whenever it exists. Similarly, a label l is a (strict) majority winner in i's neighbourhood if i's l surplus is strictly positive. We use $W^i_{(N,E,f)}$ to denote such a winner, whenever it exists.

We say that an agent $i \in N$ is *under illusion* if they have a wrong perception of the majority winner. In other words, for agent i to be under illusion in a social network (N, E) with labelling f, we must have that: $W_{(N,E,f)}$ and $W^i_{(N,E,f)}$ exist, and that $W^i_{(N,E,f)} \neq W_{(N,E,f)}$.

In this paper we are concerned with the *proportion* of agents in a network that are under illusion. For that we define the concept of q-majority illusion.

Definition 2 (q-majority illusion). *For a given social network (N, E), fraction $q \in \mathbb{Q} \cap [0, 1]$, and labelling $f : N \to \{b, r\}$, we say that f* induces *q-majority illusion, if at least $q{\cdot}n$ agents are under illusion in (N, E, f).*

If there is a labelling of a network (N, E) that induces q-majority illusion, then we say that (N, E) *admits* q-majority illusion. Henceforth, we assume that the majority label is blue, whenever one exists. Also, for a network (N, E) and agents $i, i' \in N$ such that $N(i) = \{i'\}$, we say that i is a *dependent* of i'. Let us further observe that if a labelling f induces 1-majority illusion for a network (N, E) and i is a dependent of i', then $f(i') = r$.

Example 1. *Take the labelled social network shown in Figure 1. Observe that there every vertex has the strict majority of red vertices in their neighbourhood. Hence, the depicted labelling induces 1-majority illusion. In particular, the bottom-left blue vertex is a dependent of a vertex that is labelled red.*

2.2 Parameterised Complexity

We say that a problem with an input I is *fixed-parameter tractable* (FPT), or that it is in the class **FPT** for a parameter k, if it is solvable in time $O(f(k) \cdot |I|^c)$ for some computable function f and constant c independent of k. Moreover, a problem is in **XP** (*slice-wise polynomial*) for a parameter k if there exists an algorithm solving this problem that runs in time $|I|^{f(k)}$ (called an XP-algorithm), where f is some computable function. Note that **FPT**⊆**XP**. Further, the **W**-hierachy defines a series of complexity classes extending **XP** and showing that a problem is hard for any class in this hierarchy is evidence that the problem is unlikely to be in **FPT**. In the context of our paper, we say that a problem P is **W**[1]-hard parameterised by r if there is a many-one reduction to it from the classic k-clique problem [16] in time $f(k) \cdot |I|^{O(1)}$ (where I is the instance of k-clique), with $r \leq g(k)$ for some computable function g. In the k-clique problem, given a graph G and an integer k, the objective is to check whether G contains a clique of size k as a subgraph.

Tree Decomposition. Tree width is a fundamental graph parameter, useful for the design of parameterised algorithms, which will play an important role in our analysis. Intuitively, this measurement indicates how "close" a graph is to a tree. Then, an FPT (or even XP) algorithm for a problem parameterised by the tree width implies a polynomial-time algorithm on "tree-like" graphs. Given a graph G, let $V(G)$ and $E(G)$ denote the vertex and edge set of G, respectively. For a rooted tree T and a non-root vertex $t \in V(T)$, by $parent(t)$ we denote the parent of t in the tree T. For vertices $u, t \in T$, we say that u is a *descendant* of t, denoted $u \preceq t$, if t lies on the unique path from u to the root. Note that every vertex is its own descendant. If $u \preceq t$ and $u \neq t$, then we write $u \prec t$.

Definition 3. *A* tree decomposition *of a graph G is a pair (T, β) of a tree T (whose vertices are called* nodes*) and a function $\beta : V(T) \to 2^{V(G)}$, such that:*

(1) $\bigcup_{t \in V(T)} \beta(t) = V(G)$.
(2) For every edge $e \in E(G)$, there exists a node $t \in V(T)$ such that both endpoints of e belong to $\beta(t)$.
(3) For every vertex $v \in V(G)$, the subgraph of T induced by the set $T_v = \{t \in V(T) : v \in \beta(t)\}$ is a connected tree.

The width *of (T, β) is $\max_{v \in V(T)}\{|\beta(v)|\} - 1$. The* tree width *of G, which we also refer to as $tw(G)$, is the minimum width of a tree decomposition of G.*

Let (T, β) be a tree decomposition of a graph G. We always assume that T is a rooted tree and so we have a natural parent-child and ancestor-descendant relationship among vertices in T. The set $\beta(t)$ is the *bag* at node t. For a node $t \in V(T)$, we denote by V_t the set $\bigcup_{t' \preceq t} \beta(t')$, that is, the set of all the vertices in the bags in the subtree of T rooted at t.

When designing algorithms using tree decompositions, it is generally helpful to work with a special kind of tree decomposition, that is, a *nice tree decomposition.*

Definition 4. *Let (T, β) be a tree decomposition of a graph G, where r is the root of T. The tree decomposition (T, β) is called a* nice tree decomposition *if the following conditions are satisfied.*

(1) $\beta(r) = \emptyset$ and $\beta(\ell) = \emptyset$ for every leaf node ℓ of T.
(2) Every non-leaf node (including the root node) t of T is of one of the following types:
- **Introduce node:** *The node t has exactly one child t' in T and $\beta(t) = \beta(t') \cup \{v\}$, where $v \notin \beta(t')$.*
- **Forget node:** *The node t has exactly one child t' in T and $\beta(t) = \beta(t') \setminus \{v\}$, where $v \in \beta(t')$.*
- **Join node:** *The node t has exactly two children t_1 and t_2 in T and $\beta(t) = \beta(t_1) = \beta(t_2)$.*

We note that, using a well-known, polynomial-time algorithm, we can convert any given tree decomposition to a nice tree decomposition of the same width [16].

2.3 Further Graph Parameters

Another graph parameter we consider is the *neighbourhood diversity* [35], which captures the number of "twin classes" in the graph. We say vertices u and v are *twins* if they have the same neighbours, that is $N(u) \setminus \{v\} = N(v) \setminus \{u\}$.

Definition 5. *The* neighbourhood diversity *(ND) of a graph G ($nd(G)$), is the minimum w such that $V(G)$ can be partitioned into w sets of* twin vertices.

Observe that each set of twins, which we call a *module*, is either a clique or an independent set. Hence, we call these *clique modules* and *independent modules,* respectively.

Note that graphs of bounded tree width are sparse. That is, the number of edges in a graph of tree width k is $O(kn)$. On the other hand, graphs of bounded ND can be dense. For instance, a complete graph has a ND of 1, but has $\Omega(n^2)$ edges. Moreover, note that ND is "incomparable" with tree width. That is, there are graphs of constant ND with unbounded tree width (e.g., a clique) and graphs of constant tree width with unbounded ND (e.g., a path).

We will further consider a property of a social networks that has gained importance in recent years, that is, the *c-closure* [22, 33]. For a natural number c, we say that a network is *c-closed* if every pair of vertices in this network that have at least c neighbours in common is adjacent. This concept was introduced in an attempt to capture the spirit of "social-network-like" graphs without relying on probabilistic models. Note that c-closure generalises one of the most agreed-upon properties of social networks – *triadic closure*, the property that when two members of a social network have a friend in common, they are likely to be friends themselves. [22][Table A.1], and later [33][Table 1], showed that several social networks and biological networks are indeed c-closed for rather small values of c.

Example 2. *The labelled network shown in Figure 2 consists of three clique modules, as well as two independent modules. We note that the depicted partition has the minimum number of twin classes for this network. Hence, it has the neighbourhood diversity 5. We further note that each pair of vertices i, j that has at least two neighbours in common is adjacent. Hence, the network is 2-closed.*

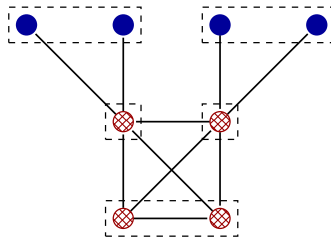

Fig. 2. A 2-closed labelled social network with ND=5.

3 Verifying Illusion

We are interested in the computational problem of checking, for a rational q, if a given network admits q-majority illusion. Formally, we will study the following problem.

q-MAJORITY ILLUSION:
Input: Social network (N, E).
Question: Is there a labelling $f : N \longrightarrow \{b, r\}$, such that f induces a q-majority illusion?

First, in Section 3.1 we show that q-MAJORITY ILLUSION is **NP**-complete even for bipartite (Theorem 1) networks. Then, in Section 3.2, we show that q-MAJORITY ILLUSION is also hard for planar (Theorem 2) networks. Subsequently, we explore the parameterised complexity of this problem. In particular, in Section 3.3 (Theorem 3) we demonstrate that q-MAJORITY ILLUSION is in **FPT** for tree width combined with maximum degree. Furthermore, in Section 3.4 (Theorem 4) we show that q-MAJORITY ILLUSION is also in **FPT** for neighbourhood diversity. Following these results we also get that this problem is **FPT** for vertex cover and clique width combined with maximum degree, and that an XP-algorithm exists for tree width alone. Nevertheless, our constructions used in proofs of Theorems 1 and 2 entail that q-MAJORITY ILLUSION is para-**NP**-hard for maximum degree alone, as well as for minimum c-closure and for maximum clique size.

We note that similar parameterised complexity landscapes have been observed for other problems in the literature. For instance, in the work by Ordyinak and Szeider [38] on Bayesian Network Learning parameterised by graph parameters of the super-structure, they obtain para-**NP**-hardness parameterised by the maximum degree Δ, an XP-algorithm parameterised by tree width (tw) and an FPT-algorithm parameterised by $\Delta + tw$.

3.1 NP-Completeness of Majority Illusion for Bipartite Networks

Observe that q-MAJORITY ILLUSIONis in **NP** for every q, since verifying if a labelling induces a q-majority illusion can be done by checking, for every vertex i, if i is under illusion. We now prove that q-MAJORITY ILLUSION is **NP**-hard for every rational $q \in (\frac{1}{2}, 1]$ by providing a reduction from the **NP**-hard problem 2P2N-3-SAT for every such q. In 2P2N-3-SAT we check the satisfiability of a given CNF formula in which all clauses have exactly three literals, and in which every variable appears exactly twice in the positive form, and twice in the negative form (see [8]). We say that such a formula is in 2P2N-3-CNF.

Let φ be a formula in 2P2N-3-CNF. We will construct an instance of q-MAJORITY ILLUSION, for which the answer is positive if and only if φ is satisfiable. We commence with constructing a social network, which we call the *encoding* of φ, or $E_\varphi = (N, E)$. This is provided in Section 3.1.1. We will further show that it admits 1-majority illusion if and only if φ is satisfiable, entailing the **NP**-hardness of 1-MAJORITY ILLUSION, which we show in Lemma 3. Finally, for each $q \in (\frac{1}{2}, 1]$, we construct a network E^q_φ, which we obtain by appending a non-trivial network construction to E_φ. We then conclude the proof by showing in Theorem 1 that E^q_φ admits a q-majority

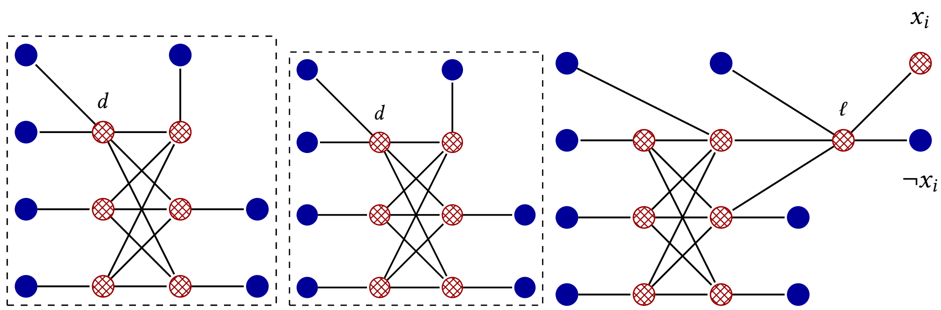

Fig. 3. Variable gadget for a variable x_i. The structures in dashed boxes show filling structures, with a unique labelling such that all members are under illusion. The labelling of the gadget is of type A on the right. The labelling of type B differs only by the labels of the literal vertices.

illusion if and only if φ is satisfiable, using the fact that increasing the blue surplus in E_φ results in an increased number of vertices under illusion (see Lemma 4), and a technical Lemma 5.

3.1.1 *Variable, Clause and Balance Gadgets.*

For a formula φ in 2P2N-3-CNF, we denote the set of variables in φ as $X = \{x_1, \ldots, x_m\}$, and the set of clauses in φ as $C = \{C_1, \ldots, C_n\}$. Let us first encode propositional variables. For a variable x_i, we define a subnetwork, which we call a *variable gadget*, as depicted in Figure 3. There, the subfigures in dashed rectangles present what we call *filling structures*. Each filling structure consists of a complete bipartite network $K_{3,3}$ and seven additional vertices. Of these seven additional vertices, at least six are a dependent of some vertex in the corresponding $K_{3,3}$. If this is true for all seven, then we assume that a designated d vertex has two dependents. We further assume that in a filling structure each of the $K_{3,3}$ vertices has at least one dependent.

Then, the variable gadget contains three copies of the filling structure. In two of them, seven vertices are a dependent of a vertex in the same filling structure. Furthermore, a variable gadget contains three additional vertices, connected as shown on the right side of Figure 3. We refer to two vertices in the right side of Figure 3 marked as x_i and $\neg x_i$, as *literal vertices*. They are adjacent to what we call a *linking vertex*, marked as ℓ in Figure 3. Furthermore, we say that upper literal vertex corresponds to x_i, and the lower literal vertex corresponds to $\neg x_i$. Finally, in the third filling structure, one of the vertices is a dependent of the linking vertex.

The following lemma shows that it is necessary for exactly one of the literal vertices in a variable gadget to be labelled r in a labelling of this structure, which induces 1-majority illusion. This observation will be crucial in demonstrating that a labelling of the encoding of φ, in which all vertices in variable gadgets are under illusion, corresponds to a valuation over X.

Lemma 1. *A labelling of a variable gadget (considered as a separate network) induces a 1-majority illusion only if at most one of the literal vertices is labelled r.*

Proof. Take a variable gadget V_i, as defined above. Also, suppose that there is a labelling f of V_i that induces 1-majority illusion. Let us begin by observing that all vertices in the $K_{3,3}$s in filling structures are labelled r in f. This observation holds as each of them has a dependent. Notice now that as the vertex adjacent to literal vertices has five neighbours, at least three of them are labelled r in f. We will now show that it cannot be the case that both literal vertices are labelled r in f.

Let us first observe that the linking vertex needs to be labelled r in f, as it has a dependent. Then, notice that there are forty-two vertices in a variable gadget. Thus, at most twenty vertices can be labelled r in f, as we

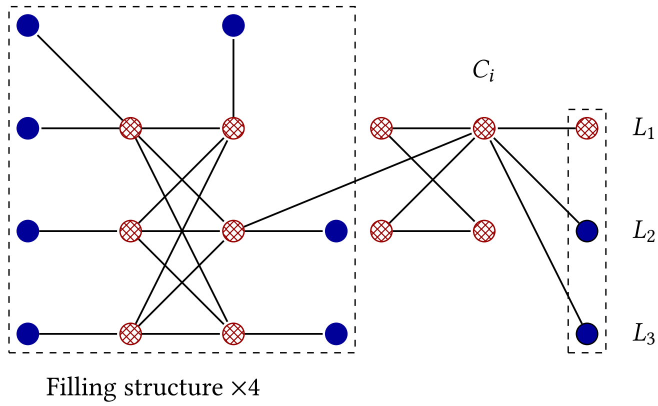

Fig. 4. Clause gadget with a unique labelling such that all members are under illusion, including five filling structures. Note that each of the vertices in the gadget outside of filling structures have a dependent, while the verifier vertex is also adjacent to one of the vertices in a $K_{3,3}$. Furthermore, each of the vertices in the four-clique have a dependent in one of the filling structures (which is omitted for clarity).

assume that the strict majority label in f is b. As we observed, at least eighteen of the vertices in filling structures need to be labelled r in such a labelling of this gadget, as they have dependents. Hence, at most one of the literal vertices can be labelled r in this labelling. It follows that f induces a 1-majority illusion only if at most one of the literal vertices is labelled r.

□

We distinguish two labellings of a variable gadget (as a separate network) that admit 1-majority illusion. In one of them, the vertex corresponding to x_i is labelled r (we say that such a labelling is of *type A*). In the second, the vertex corresponding to $\neg x_i$ is labelled r (we say that such a labelling is of *type B*). Consider now a labelled network (N_1, E_1, f), where a variable gadget V_i is a subnetwork. We note that if the blue surplus is 1 in $(N_1 \setminus V(V_i), E \setminus E(V_i), f')$, where f' is such that, for every $i \in N \setminus V(V_i)$, we have that $f(i) = f'(i)$, then f induces 1-majority illusion if V_i is labelled in type A or type B, but not if both literal vertices are labelled r.

We further introduce, for each clause $C_i \in C$, what we call a *clause gadget* corresponding to C_i, as depicted in Figure 4. The three vertices in the right side of this figure do not belong to the clause gadget. Instead, they correspond to the literals in C_i. Then, the gadget includes five filling structures. Furthermore, there are four vertices outside of filling structures in this gadget, such that each of them has a dependent in some filling structure in the same gadget. Then, we call one of them, the top vertex in the middle of the Figure 4, the *verifier vertex* for C_i. Also, the verifier vertex is adjacent to one vertex in a $K_{3,3}$ of some filling structure. So, a clause gadget has sixty-nine vertices in total: thirteen vertices in each of five filling structures, as well as four additional vertices. Observe that as all of the vertices in $K_{3,3}$s in filling structures have a dependent, it holds that in a labelling of a clause gadget that induces 1-majority illusion, they are all labelled r.

We will now show that a labelling of a clause gadget can only induce 1-majority illusion if at least one of the adjacent literal vertices is labelled r. This fact will later allow us to show that the fact that all vertices in a clause gadget for C_i are under illusion for some labelling of an encoding means that C_i is satisfied in a valuation.

Lemma 2. *There exists a labelling f of an encoding of φ that (1) induces 1-majority illusion in a clause gadget G for C_i, and (2) blue is the majority winner in G, if and only if at least one vertex in G is adjacent to a literal vertex labelled r in f.*

Proof. Take some clause gadget, and call it G. Let us first observe that there are sixty-nine vertices in G. Let us further observe that, for every filling structure F, it holds that in every labelling f of F that induces 1-majority illusion at least six members which have dependents, are labelled r. Further, by previous observations, it holds that all vertices outside of filling structures are labelled r in f.

Let us now show f exists, if at least one literal vertex, adjacent to the verifier vertex in G, is labelled r. Suppose that this is the case. Then, let us construct such a labelling f. First, let us label r all members of the $K_{3,3}$s in filling structures, as well as all of the vertices in G, which are not in filling structures. Further, let all other vertices in the gadget be labelled b. Observe that in this labelling, thirty-four vertices are labelled r, and thirty-five are labelled b. Observe further that this implies that all nodes that are a dependent are labelled b in f. Notice that then all of the vertices in G, other than the verifier vertex, are under majority illusion. Notice now that, by assumption, one of the literal vertices adjacent to the verifier vertex is labelled r. Hence, the verifier vertex is under illusion, given the proposed labelling, and so the claim holds.

Suppose, towards a contradiction, that all literal vertices adjacent to the verifier vertex are labelled b, but there is a labelling f of G that induces 1-majority illusion. It follows by previous reasoning that the vertex which is a dependent of the verifier vertex is labelled b in f. Notice, however, that then the verifier vertex is not under illusion, as four out of seven of its neighbours are labelled b, which contradicts the assumptions.

□

Finally, for an integer $s \geq 2$, we define what we call a *balance gadget,* in order to ensure that, in the encoding E_φ, q-majority illusion can be induced only by a labelling corresponding to a valuation in which all of the clauses are satisfied. If s is even, then the balance gadget is the collection of $\frac{s}{2}$ pairs of vertices, which are disconnected from the rest of the vertices in the encoding. Otherwise, we construct five vertices, such that four of them form a bipartite complete graph $K_{2,2}$, while the fifth is a dependent of one of the other vertices, as well as the balance gadget for $s-3$, if $s \geq 5$. Observe that the balance gadget is bipartite, and that, for every labelling of this gadget, which induces 1-majority illusion (not as a separate network), at most one vertex in this structure is labelled b (this is the case when s is odd). Note that otherwise some vertex in one of the pairs would only not be adjacent to the strict majority of red vertices and hence it would not be under illusion.

3.1.2 *Encoding of a 2P2N-3-CNF Formula.*

We are now ready to construct a social network E_φ, which *encodes* φ. First, for every variable $x \in X$, let us construct a variable gadget, as depicted in Figure 3. Further, for every clause $C_i \in C$, that is, $\{L_i^1, L_i^2, L_i^3\}$, let us create a clause gadget, as shown in Figure 4, with literal vertices corresponding to L_i^1, L_i^2, and L_i^3 being adjacent to the verifier vertex in the clause gadget corresponding to C_i. As a final step, let us construct a balance gadget for $s = 2m + n - 1$, which by construction is always greater than or equal to two (this is because there is at least one variable and one clause in φ).

Observe that, since there are $2m + n - 1$ vertices in the balance gadget if s is even, and $2m + n + 1$ vertices otherwise. Recall that we have $42m$ vertices in variable gadgets, and $69n$ vertices in clause gadgets. Hence, there are $44m + 70n - 1$, or $44m + 70n + 1$, vertices in E_φ. Let us further notice that, following previous observations, for every labelling of E_φ, which induces 1-majority illusion, and for every variable gadget (consisting of forty-two vertices), at least twenty of its members are labelled r. Similarly, in such a labelling, for every clause gadget, we have that at least thirty-four out of sixty-nine members of the gadget are labelled r. Finally, by previous observations, all vertices in the balance gadget are labelled r, if s is even, and $2m + n + 1$ vertices otherwise. We note that these observations imply that in every labelling f of E_φ that induces 1-majority illusion it holds that $|B_f| - |R_f| = 1$.

Lemma 3. *Let φ be a formula in 2P2N-3-CNF. Then, φ is satisfiable if and only if E_φ admits 1-majority illusion.*

Proof. Let us consider a formula φ in 2P2N-3-CNF, with the set of variables $X = \{x_1, \ldots, m\}$, and the set of clauses $C_\varphi = \{C_1, \ldots, C_n\}$. Then, we will construct the encoding E_φ of φ, and show that it admits 1-majority illusion if and only if φ is satisfiable.

Let us first suppose that it is. Then, take a model M of φ and label E_φ as follows. First, label variable gadgets, so that, for every such gadget corresponding to a variable x_i, it is of type A if x_i is true in M, and of type B otherwise. Then, observe that, by previous observations on the construction of E_φ, we have that every verifier vertex in E_φ is adjacent to some literal vertex which is labelled r, as all clauses are satisfied under M. Hence, following previous observations, E_φ admits 1-majority illusion.

Suppose now that φ is not satisfiable. Then, observe that every labelling of E_φ that admits a 1-majority illusion requires variable gadgets not to have both literal vertices labelled r. This follows from Lemma 1, as in such a labelling it holds that $|B_f| - |R_f| = 1$. Further, as φ is not satisfiable, it holds that at least one verifier vertex would need to be adjacent to three literal vertices labelled b. But then, it would not be under majority illusion, which contradicts the assumptions. Hence, E_φ admits 1-majority illusion if and only if φ is satisfiable.

□

We now show some further properties of E_φ. We will henceforth assume, for simplicity, that $s = 2m + n - 1$ is even. The subsequent claims can be shown for odd s similarly, as the construction of the balance gadget for odd integers allows for analogous reasoning. Given a 2P2N-3-CNF formula φ, let $I_\varphi = 22m + 35n - 1$, where m is the number of variables and n the number of clauses in φ. Observe that this is the maximum number of vertices which can be labelled red in E_φ, if blue is the strict majority label in this network.

Lemma 4. *For every 2P2N-3-CNF formula φ, natural $s \leq I_\varphi$ and any labelling f of $E_\varphi = (N, E)$, such that $|R_f| = I_\varphi - s$, the number of vertices under illusion in E_φ under f is at most $|N| - s$.*

Proof. Consider a formula φ in 2P2N-3-CNF and a natural $s < I_\varphi$, as well as a labelling f of $E_\varphi = (N, E)$, such that $|R_f| = I_\varphi - s$. We will show that the number of vertices in E_φ, which are not under illusion given f, is at most $|N_\varphi| - s$.

Let us denote as A the set of all vertices in $K_{3,3}$s in filling structures, and all of the other vertices in variable gadgets that are not literal vertices. Then, let B' be the set of all vertices in clause gadgets that are not in filling structures. Further, let C be the set of literal vertices, and D be the set of vertices in the balance gadget. Finally, let E' be the set of all remaining vertices in N. Observe now that $A \cup B' \cup C \cup D \cup E' = N$. Moreover, by construction, we have that $|A| + |B'| + \frac{|C|}{2} + |D| = I_\varphi$.

We further show some crucial properties of A, B', C, and D. Observe now that each vertex in A has a dependent. Hence, there exists a set $N_A \subseteq E'$, such that for every $i \in N_A$, we have that i is not under illusion, while $|N_A| = |B^A|$. Similarly, for every $i \in B'$, we have that i has a dependent. Hence, there is a set $N_{B'} \subseteq E'$, such that, for each $j \in N_{B'}$, we have that j is not under illusion, while $|N_{B'}| = |B^{B'}|$. Let further $M_C = \frac{|C|}{2} - |B^C|$ if $\frac{|C|}{2} - |B^C| > 0$, and 0 otherwise. Notice that, by construction, there is a set $N_C \subseteq A$, such that, for every $i \in N_C$, i is not under illusion, while $|N_C| = M_C$. Finally, notice that, by construction of a balance gadget, there is a set $N_D \subseteq D$, such that, for every $i \in D$, we have that i is not under illusion, while $|N_D| \geq |B^D|$. Let us also observe that N_A, $N_{B'}$, N_C, and N_D are pairwise disjoint. This implies that the number of vertices not under illusion is at least the sum of cardinalities of these sets.

We can now show that at least k vertices are not under illusion under f. We note that $|A| + \frac{|C|}{2} + \frac{|B'|}{2} + |D| = I_\varphi$, and that at most I_φ vertices are labelled r in f, as otherwise b would not be the strict majority label. But then, at least s vertices are labelled b in $A \cup B' \cup C \cup D$. This implies, however, that $|N_A| + |N_{B'}| + |N_C| + |N_D| \geq s$, and hence at least s vertices are not under illusion under f.

□

Observe also that, by the reasoning similar to the proof of Lemma 4, we also get that, for a labelling f of E_φ that maximises the number of vertices under illusion (which we call M), $s \leq I_\varphi$ and any labelling f' of $E_\varphi = (N, E)$, such that $R_{f'} = I_\varphi - s$, the number of vertices under illusion in E_φ under f' is at most $M - s$.

We further need the following technical lemma.

Lemma 5. *Let q be a rational number in $(\frac{1}{2}, 1]$, and $s > 0$ be a polynomial-time computable natural number. Then, there exists a natural number h^* such that $\frac{s+h^*}{s+2h^*} \geq q$, but $\frac{s+h^*-1}{s+2h^*} < q$.*

Proof. Take a $s > 0$, and a fraction $\frac{a}{b} \in (\frac{1}{2}, 1]$. Observe that if $\frac{a}{b} = 1$, then the claim holds immediately. So, we will only consider fractions such that $\frac{a}{b} < 1$. Then, we define a function $f : \mathbb{N} \longrightarrow \mathbb{Q}$ such that, for a natural h, we have that $f(h) = \frac{s+h}{s+2h}$. Observe first that $f(0) = 1$. Also, observe that f is strictly downwards monotone, and is bounded by, and tends to $\frac{1}{2}$. But then, as $q \in (\frac{1}{2}, 1]$, there needs to exist a maximal h such that $f(h) \geq q$, and as f is strictly downwards monotone, $f(h+1) < q$. We denote such a number as h^*. Note that we can efficiently compute such a number by first computing a real number h such that $\frac{a}{b} = \frac{s+h}{s+2h}$ and then taking its ceiling. Then, $f(h^*+1) = \frac{s+h^*+1}{s+2h^*+2}$. Let us notice that $\frac{s+h^*-1}{s+2h^*} \leq \frac{s+h^*+1}{s+2h^*+2}$. This is because $\frac{s+h^*+1}{s+2h^*+2} - \frac{s+h^*-1}{s+2h^*} = \frac{2(h^*+1)}{(s+2h^*)(s+2h^*+2)} \geq 0$. But then, since $f(h+1) < q$ and $f(h) \geq q$, the claim follows.

□

We refer to such a number as $h^*_{s,q}$. Note that the $h^*_{s,q}$ being tractably computable is crucial to ensure that the intended reduction is constructable in polynomial time.

3.1.3 NP-Completeness.

We are now ready to prove the **NP**-hardness of q-majority illusion, for each $q \in (\frac{1}{2}, 1]$. Towards this end, we construct a network E^q_φ, for every formula φ in 2P2N-3-CNF and a such a fraction q. We start with constructing E_φ, and a set of $h^*_{|V(E_\varphi|),q}$ pairs of vertices. Then, it follows from Lemma 3, as well as Lemmata 4 and 5, that E^q_φ admits q-majority illusion if and only if φ is satisfiable. Below we include the full proof of this claim. Observe further that q-majority illusion is in **NP**, as one can easily check the number of vertices under illusion in a labelled network.

Theorem 1. *q-majority illusion is **NP**-complete for every rational q in $(\frac{1}{2}, 1]$, even for bipartite networks.*

Proof. Take any rational q in $(\frac{1}{2}, 1]$. We will now show that q-majority illusion is **NP**-hard by reduction from 2P2N-3-SAT.

Consider a 2P2N-3-CNF formula φ with the set $X = \{x_1, \dots, x_m\}$ of variables, and the set $C = \{C_1, \dots, C_n\}$ of clauses. Let us construct what we call a *q-encoding* of φ. First, let E_φ be a subnetwork of the q-encoding of φ. Moreover, construct $h^*_{|V(E_\varphi)|,q}$ pairs of vertices, such that vertices in each such pair are connected to each other, but not to any other vertex in the network. We call this set of pairs H. Observe further that, following previous observations, the q-encoding of φ can be constructed in polynomial time. Also, by Lemma 3 and Lemma 5, the q-encoding of φ admits q-majority illusion, if at least $|V(E_\varphi)| + h^*_{|V(E_\varphi)|,q}$ vertices are under illusion in f.

Let us now show that the q-encoding of φ admits q-majority illusion if and only if φ is satisfiable. First, suppose that φ is satisfiable. Observe further that, as φ is satisfiable, by Lemma 3, we have that E_φ admits 1-majority illusion as a separate network. Hence, there is a labelling of the q-encoding of φ, such that exactly I_φ vertices in E_φ, as well as one of vertices in each pairs in H, are labelled red, while $|V(E_\varphi)| + h^*_{|V(E_\varphi)|,q}$ vertices are under illusion. Hence, the q-encoding of φ admits q-majority illusion.

Suppose now that φ is not satisfiable. Then, suppose towards a contradiction that there is a labelling f of the q-encoding of φ that induces q-majority illusion. Let us first observe that if less than $h^*_{|V(E_\varphi)|,q}$ are labelled red in H, then f does not induce q-majority illusion. Indeed, if it was the case, then less than $h^*_{|V(E_\varphi)|,q}$ vertices in H

would be under illusion, and hence the number of vertices under illusion in the q-encoding of φ would be strictly smaller than $|V(E_\varphi)| + h^*_{|V(E_\varphi)|,q}$. But then, as f induces q-majority illusion, at least $h^*_{|V(E_\varphi)|,q}$ are labelled red in H. So, the number of vertices labelled red in E_φ is smaller or equal to I_φ. If it is equal to I_φ, then the number of vertices under illusion in H is $h^*_{|V(E_\varphi)|,q}$. But, as φ is not satisfiable, not all members of E_φ are under illusion, and hence f does not induce q-majority illusion. Now, suppose that less than I_φ vertices are labelled red in E_φ. Let $k = I_\varphi - |R^{V(E_\varphi)}|$. Further, let us denote as M the maximum number of vertices under illusion in E_φ, if I_φ vertices are labelled red in this subnetwork.

Now, by Lemma 4, we have that the number of vertices under illusion in E_φ is at most $I_\varphi - k$. But then, the number of vertices labelled red in H is at most $h^*_{|V(E_\varphi)|,q} + k$, and hence the number of vertices under illusion in the q-encoding of φ is at most $M - k + h^*_{|V(E_\varphi)|,q}$, which is smaller than $|V(E_\varphi)| + h^*_{|V(E_\varphi)|,q}$, since $M < |V(E_\varphi)|$. □

Moreover, by inspecting all pairs of vertices in the construction in the proof of Theorem 1, we get that q-MAJORITY ILLUSION is **NP**-complete also for networks in which minimum c-closure is bounded by a constant.

OBSERVATION 1. *q-MAJORITY ILLUSION is* **NP**-*complete for every rational q in $(\frac{1}{2}, 1]$, even for networks with minimum c-closure bounded by 3.*

PROOF. Let us show that the claim holds by demonstrating that minimum c-closure of E_φ is at most 3. We will show that, for every pair of vertices i, j in E_φ, we have that if $|N(i) \cap N(j)| \geq 3$, then i and j are adjacent. We consider the following, exhaustive cases. (1) i and j are both in the same variable gadget, (2) i and j are both in the same clause gadget, (3) i is in some variable gadget while j is in some clause gadget, (4) i and j belong to distinct variable gadgets, (5) i and j belong to distinct clause gadgets, and (6) i and j are in the balance gadget. Observe that in all other cases i and j do not have neighbours in common.

If (1) is the case, observe that if i and j are in the filling structure, then either they are adjacent or $|N(i) \cap N(j)| \leq 2$. Hence, the claim holds. Similarly, if they both belong to additional three vertices in the gadget, then $|N(i) \cap N(j)| \leq 2$, since we assume that a literal does not appear more than twice in a formula. If (2) is the case, then the claim holds by similar reasoning. Further, if (3) is the case, then notice that i and j have at most two neighbours in common. Also, if (4) is the case, then the only neighbours that i and j can have in common are verifier vertices. Notice further that $|N(i) \cap N(j)| \leq 2$ as φ is in 2P2N-3-CNF. Moreover, if (5) holds, then the only vertices that i and j have in common are literal vertices. But then, we have that $|N(i) \cap N(j)| \leq 3$, as the size of the clauses in φ is limited by three. Finally, if (6) is the case, then i and j have at most two neighbours in common, so the claim follows as well.

□

Furthermore, again by examining the reduction used in the proof of Theorem 1, we get that q-MAJORITY ILLUSION is **NP**-complete even if the maximum degree of a vertex in a network is bounded by 6.

OBSERVATION 2. *q-MAJORITY ILLUSION is* **NP**-*complete for every rational q in $(\frac{1}{2}, 1]$, even for networks with maximum degree bounded by 6.*

PROOF. Let us show that the claim holds by demonstrating that, for a formula φ in 2P2N-3-CNF, no vertex in E_φ has the degree greater than 6. To see that, take any vertex i in E_φ. Let us examine the following exhaustive cases. (1) i is in a filling structure, (2) i is in a variable gadget, but is not a literal vertex and is not in a filling structure, (3) i is a literal vertex, (4) i is in the balance gadget, (5) i is in a clause gadget, but is not a verifier vertex and is not in a filling structure, (6) i is a verifier vertex.

Let us then notice that if (1) is the case, then by construction i has the degree of at most 5. Similarly, if (2) holds, then the degree of i is bounded by 5. Further, if (3) is the case, then as each literal appears exactly twice in φ, we have that i is adjacent to at most two verifier vertices. Hence, the degree of i is at most 3. Also, if (4) holds, then by construction we have that i has the degree of at most 2. Moreover, if the (5) is the case, then we get that the degree of i is bounded by 3. Finally, if (6) holds, then i is adjacent to at most three literal vertices, as the clause that its gadget corresponds to is limited to three literals. Hence, i is adjacent to at most 6 vertices. Then, the claim follows.

□

It is important to note that in order to obtain Observations 1 and 2, we crucially use the fact that the formulas we encode are 2P2N-3-CNF.

3.2 NP-Completeness of Majority Illusion for Planar Networks

We now show that q-majority illusion is **NP**-complete also for *planar* networks. Observe that this result rules out using generalisations of planarity as possible structural restrictions to get polynomial-time algorithms. We prove it by reduction from Planar 3-SAT, where one is given a formula φ in 3-CNF such that the incidence graph of φ is planar, and the goal is to decide whether φ is satisfiable. The reduction that we use to show this result follows a similar structure to the one we construct in the proof of Theorem 1. Nevertheless, to show that the hardness of q-majority illusion also holds for planar networks, we reformulate the reduction using planar graphs.

So, we first construct a network E_φ, a planar encoding of φ, which is an input of Planar 3-SAT. We show, in Lemma 8, that it admits 1-majority illusion if and only if φ is satisfiable. Then, in Theorem 2, we show the **NP**-hardness of q-majority illusionon planar networks similarly to the proof of Theorem 1.

3.2.1 *Variable, Clause, and Balance Gadgets.*

For a formula φ in CNF with a planar incidence graph, we denote the set of variables in φ as $X = \{x_1, \dots, x_m\}$ and the set of clauses in φ as $C = \{C_1, \dots, C_n\}$. We assume, without loss of generality, that there are at least two clauses in φ. Let us first encode the propositional variables.

For each variable $x_i \in X$, we construct a subnetwork, which we call a *variable gadget*, as depicted in Figure 5. Further, one of the vertices in this structure corresponds to x_i, and one to $\neg x_i$, as shown in Figure 5. We call them *literal vertices*. Let us first observe that there are seventeen vertices in this gadget. Notice that seven of them have dependents, which implies that in every labelling of a variable gadget, which induces 1-majority illusion, they are labelled r. We also observe that this gadget is planar.

Similarly to the encoding used in the previous part of this section, we show that in a labelling of a variable gadget, which induces 1-majority illusion, exactly one of the literal vertices is labelled r. We will later use this fact to demonstrate that every labelling of the encoding of φ, which induces 1-majority illusion, corresponds to some valuation over X.

Lemma 6. *A labelling of a variable gadget (considered as a separate network) induces a 1-majority illusion only if exactly one of the literal vertices is labelled r.*

Proof. Observe first that in every labelling f of a variable gadget V_i for x_i that induces 1-majority illusion, seven of the members of V_i need to be labelled r, as they have dependents. Then observe that as there are seventeen vertices V_i, at most eight of them are labelled r in f. Let us now suppose that exactly one of literal vertices is labelled r and, without loss of generality, let this vertex correspond to x_i. Then it is enough to consider a labelling as presented in Figure 5, and to observe that then, there exactly eight vertices are labelled r, while all of the members of the gadget are under illusion.

Assume now towards a contradiction that both of the literal vertices are labelled b in f. Then, by previous observations, it holds that seven vertices with dependents are labelled r, and hence at most one node which is a

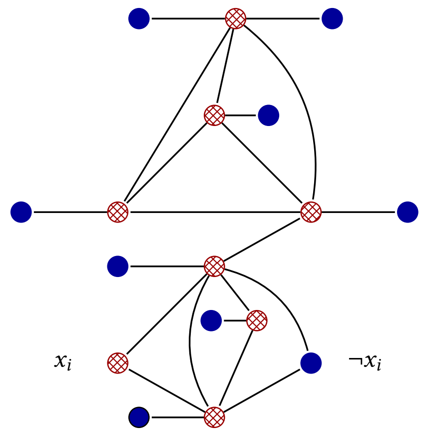

Fig. 5. Variable gadget for the variable x_i, with a labelling such that all members are under illusion. Two of the vertices in the gadget correspond to literals, x_i and $\neg x_i$. Observe that this structure is planar.

dependent is labelled r. Note that this implies that at least a half of neighbours of one of the vertices adjacent to the vertex corresponding to x_i is labelled b, and thus it is not under illusion, which contradicts the assumptions. □

We note that there are exactly two labellings of a variable gadget for x_i that induce 1-majority illusion. We say that such a labelling is of type A, if the literal vertex corresponding to x_i is labelled r, and of type B, if the literal vertex corresponding to $\neg x_i$ is labelled r. Intuitively, if this gadget is labelled in type A, then x_i is set to true, while if it is labelled in type B, it is set to false.

Furthermore, let us define what we call a *clause gadget*, which corresponds to a clause $C_i \in C$. We begin with introducing what we call a *planar filling structure* (PFS), as depicted in Figure 6. It consists of nine vertices, four of which have dependents. We distinguish one of those with one dependent only as the *connector* vertex PFS_c. Observe that in every labelling of this structure, which induces 1-majority illusion, at least four vertices are labelled r. Notice that this gadget is planar.

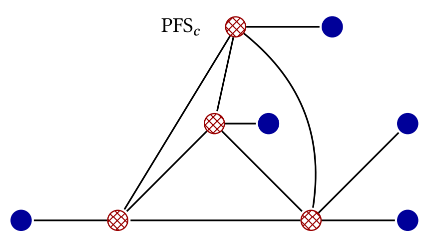

Fig. 6. Planar filling structure (PFS) with a labelling, such that all members are under illusion. The top-centre vertex is the connector vertex.

Then, the clause gadget, as shown in Figure 7, for a clause C_i, consists of three copies of PFS, as well as four additional vertices. The central vertex, which we call a *verifier vertex*, is adjacent to literal vertices corresponding to the members of C_i. Further, it is adjacent to the PFS_c of each PFS in the gadget. Finally, it has one dependent, and the remaining two vertices form an isolated pair. Observe that this structure is planar.

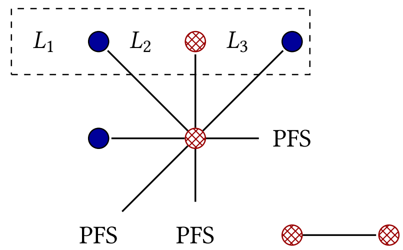

Fig. 7. Clause gadget, with a labelling, such that all members are under illusion. There, the vertices in the dashed box are literal vertices corresponding to the literals in the clause.

We further show that as in the case of the encoding used earlier in this section a clause gadget admits 1-majority illusion only when the verifier vertex is adjacent to some literal vertex that is labelled r. We will later associate the fact that it is the case for a clause gadget corresponding to a clause C_i with C_i being satisfied in a valuation over X.

Lemma 7. *There exists a labelling of a clause gadget for C_i (not as a separate network), which induces 1-majority illusion, if and only if at least one of literal vertices corresponding to a member of C_i is labelled r.*

Proof. Let us first observe that there are thirty-one vertices in the clause gadget corresponding to the clause C_i. Thus, in a labelling f inducing 1-majority illusion, at most fifteen of them are labelled r. Further, as observed before, we have that twelve members of PFS subnetworks in the gadget are labeled r in f, as they have dependents. Moreover, in such a labelling, the isolated pair of vertices also needs to be labelled r, as otherwise one of them would not be under illusion. Furthermore, the verifier vertex needs to be labelled r as well, since it has a dependent. Then, it follows that all other vertices in the gadget, including one of verifier vertex's neighbours, are labelled b in f, as otherwise r would not be the strict minority label.

Let us then assume that at least one of the literal vertices adjacent to the verifier vertex is labelled r. Without loss of generality, let it be L_i^2. Then, it is enough to construct a labelling, as shown in Figure 7 and Figure 6, and notice that then all vertices in the gadget are under illusion. Now, suppose towards a contradiction that f induces 1-majority illusion, but all literal vertices corresponding to literals in C_i are labelled b. Then notice that, by previous observations, one of the neighbours of the verifier vertex within the gadget is labelled b. But then, four out of seven vertices adjacent to the verifier vertex are labelled b, and thus it is not under majority illusion, which contradicts the assumptions.

□

Finally, for $s \geq 2$, we define what we call a *balance gadget*, similar to the construction for bipartite graphs. If s is even, then the balance gadget is the collection of $\frac{s}{2}$ pairs vertices, which are disconnected from all other vertices in the encoding. Otherwise, we construct a triple of vertices, which forms a clique disconnected from the rest of the network, as well as the balance gadget for $s-3$, if $s \geq 5$. Observe that the balance gadget is planar, and that for every labelling of this gadget, which induces 1-majority illusion (not as a separate network), all of its members are labelled r.

3.2.2 *Encoding of a 3-CNF Formula.*

We can now construct a social network $E_\varphi = (N, E)$ that *encodes* a 3-CNF φ, in which the incidence graph between variables and clauses is planar. First, for every variable $x_j \in X$, let us construct a variable gadget, as depicted in Figure 5. Further, for every clause $C_i \in C$, which we denote as $\{L_i^1, L_i^2, L_i^3\}$, let us create a clause gadget, as shown in Figure 7, with literal vertices corresponding to L_i^1, L_i^2, and L_i^3 being adjacent to the verifier vertex in the clause gadget that corresponds to C_i. As a final step, let us construct a balance gadget with $s = m + n - 1$, which by construction is always at least two, since we assumed that there are at least two clauses in φ. Observe how since the incidence graph of φ is planar, we can obtain that so is E_φ.

Notice further that since there are $m + n - 1$ vertices in the balance gadget, there are $18m + 32n - 1$ vertices in E_φ. Let us further notice that following previous observations, for every labelling of E_φ, which induces 1-majority illusion, all vertices in the balance gadget are labelled r. Also, then, for every variable gadget, which consists of seventeen vertices, at least eight of its members are labelled r. Similarly, in such a labelling, for every clause gadget, we have that exactly fifteen out of thirty one-members of the gadget are labelled r. This implies that in such a labelling each variable gadget is either labelled in type A, or in type B, as otherwise b would not be a strict majority label.

We can now show that the encoding of φ admits 1-majority illusion exactly when φ is satisfiable.

LEMMA 8. *Let φ be a formula in 3-CNF. Then, φ is satisfiable if and only if E_φ admits 1-majority illusion.*

PROOF. Take a formula φ in 3-CNF, with the set of variables $X = \{x_1, \ldots, x_m\}$ and the set of clauses $C = \{C_1, \ldots, C_n\}$. Then, we show that the encoding E_φ of φ admits 1-majority illusion if and only if φ is satisfiable.

Let us first suppose that it is and show that E_φ admits 1-majority illusion. Take a model M of φ and label E_φ as follows. First, label variable gadgets, so that every such gadget corresponding to some variable $x_i \in X$ is of type A, if x_i is true in M, and of type B, if x_i is false in M. Then observe that by the construction of E_φ, every verifier vertex in the construction is adjacent to some literal vertex, which is labelled r, as all clauses are satisfied under M. Hence, following previous observations, E_φ admits 1-majority illusion.

Otherwise, suppose that φ is not satisfiable. Then, observe that since every labelling of E_φ that admits a 1-majority illusion requires variable gadgets to be labelled in type A or type B, and φ is not satisfiable, it holds that at least one verifier vertex would need to be adjacent to three literal vertices labelled b. But then, it would not be under the majority illusion, which contradicts the assumptions. Hence, E_φ admits 1-majority illusion if and only if φ is satisfiable.

□

We now show some further properties of E_φ, which will be crucial towards showing that q-MAJORITY ILLUSION is **NP**-hard for every $q \in (\frac{1}{2}, 1]$, even for planar network. Given a 3-CNF formula φ, let $I_\varphi = 9m + 16n - 1$, where m is the number of variables and n the number of clauses in φ. Observe that this is the maximum number of vertices, which can be labelled red in E_φ if blue is the strict majority label in this network. Also notice that following previous observations it is the minimum number of vertices, which need to be labelled red in a labelling that induces 1-majority illusion in E_φ.

We can now show that if less than I_φ vertices are labelled r in E_φ, then the number of vertices under illusion is limited.

LEMMA 9. *For every 3-CNF formula φ, $s \leq I_\varphi$, and labelling f of $E_\varphi = (N, E)$, such that $|R_f| = I_\varphi - s$, the number of vertices under illusion in E_φ under f is at most $|N| - s$.*

PROOF. We now show several properties of A, B', and C. First, observe that all vertices in A have dependents. This implies that there exists a set $N_A \subseteq D$, such that for every $i \in N_A$, i is not under illusion, while $|N_A| = |B^A|$. Let now $M_{B'} = \frac{|B'|}{2} - |B^{B'}|$, if $\frac{|B'|}{2} - |B^{B'}| > 0$, and 0 otherwise. Observe that, by construction, there is a set $N_{B'} \subseteq D$, such that for every $i \in N_{B'}$, i is not under illusion, while $|N_{B'}| = M_C$. Finally, we note that there is a set $N_C \subseteq C$, such that for every $i \in C$, i is not under illusion, while $|N_C| \geq |B^C|$. Let us also observe that N_A, $N_{B'}$, and N_C, are pairwise disjoint. This implies that the number of vertices not under illusion is at least the sum of cardinalities of these sets.

We are now ready to show, that at least k of them are not under illusion under f. Notice, that $|A|+|B'|+\frac{|C|}{2} = I_\varphi$, and that at most I_φ vertices are labelled r in f, as otherwise b would not be the strict majority label. But then, at least s vertices are labelled b in $A \cup B' \cup C \cup D$. This implies, however, that $|N_A| + |N_{B'}| + |N_C| + |D| \geq s$, and hence at least s vertices are not under illusion under f.

□

Observe also that, by the reasoning similar to the proof of Lemma 9, we also get that, for a labelling f of E_φ that maximises the number of vertices under illusion (which we call M), $s \leq I_\varphi$ and any labelling f' of $E_\varphi = (N, E)$, such that $R_{f'} = I_\varphi - s$, the number of vertices under illusion in E_φ under f' is at most $M - s$.

3.2.3 **NP**-*Completeness.*

We are now ready to show **NP**-completness of q-majority illusionfor planar graphs. The proof is similar to the proof of Theorem 1.

Theorem 2. *q-majority illusion is* **NP**-*complete for every rational q in $(\frac{1}{2}, 1]$, even for planar networks.*

Proof. Take any rational q in $(\frac{1}{2}, 1]$. First, notice that as observed before, q-majority illusion is in **NP**. We will now show that it is **NP**-hard by reduction from *Planar 3-SAT*.

Consider a 3-CNF formula φ with the set $X = \{x_1, \ldots, x_m\}$ of variables, and the set $C = \{C_1, \ldots, C_n\}$ of clauses, with a planar incidence graph. Let us construct what we call a *q-encoding* E^q_φ of φ. First, let E_φ be a subnetwork of the q-encoding of φ. Moreover, we construct $h^*_{|V(E_\varphi)|,q}$ pairs of vertices,[2] such that vertices in each such pair are connected to each other, but not to any other vertex in the network. We call this set of pairs H. Observe further that the q-encoding of φ can be constructed in polynomial time. Also, by Lemma 8 and Lemma 5, the q-encoding of φ admits q-majority illusion if at least $|V(E_\varphi)| + h^*_{|V(E_\varphi)|,q}$ vertices are under illusion in f.

Let us now show that the q-encoding of φ admits q-majority illusion if and only if φ is satisfiable. First, suppose that φ is satisfiable. Then observe that as φ is satisfiable, by Lemma 8, it holds that E_φ admits 1-majority illusion as a separate network. Hence, there is a labelling of the q-encoding of φ, such that exactly I_φ vertices in E_φ, as well as one of vertices in each additional pairs, are labelled red, and $|V(E_\varphi)| + h^*_{|V(E_\varphi)|,q}$ vertices are under illusion. Hence, the q-encoding of φ admits q-majority illusion.

Suppose now that φ is not satisfiable. Then, suppose that there is a labelling f of the q-encoding of φ that induces q-majority illusion. Let us first observe that if less than $h^*_{|V(E_\varphi)|,q}$ are labelled red in H, then f does not induce q-majority illusion. Indeed, if it was the case, then less than $h^*_{|V(E_\varphi)|,q}$ vertices in H would be under illusion, and hence the number of vertices under illusion in the q-encoding of φ would be strictly smaller than $|V(E_\varphi)| + h^*_{|V(E_\varphi)|,q}$. But then, as f induces q-majority illusion, we have that, following Lemma 5, at least $h^*_{|V(E_\varphi)|,q}$ vertices are labelled red in H. So, the number of vertices labelled red in E_φ is smaller or equal to I_φ. If it is equal to I_φ, then the number of vertices under illusion in H is $h^*_{|V(E_\varphi)|,q}$, but as φ is not satisfiable, not all members of E_φ are under illusion, and hence f does not induce q-majority illusion.

Now, suppose that less than I_φ vertices are labelled red in E_φ. Let $s = I_\varphi - |R^{V(E_\varphi)}|$. Further, let us denote as M the maximum number of vertices under illusion in E_φ, if I_φ vertices are labelled red in this subnetwork. Now, by Lemma 9, we have that the number of vertices under illusion is at most $M - s$. But then, the number of vertices labelled red in H is at most $h^*_{|V(E_\varphi)|,q} + s$, and hence the number of vertices under illusion in the q-encoding of φ is at most $M - s + h^*_{|V(E_\varphi)|,q}$, which is smaller than $|V(E_\varphi)| + h^*_{|V(E_\varphi)|,q}$, since $M < |V(E_\varphi)|$. It follows, by Lemma 9, that less than $q \cdot |V(E^q_\varphi)|$ vertices are under illusion, which contradicts the assumptions.

□

[2]See Lemma 5 for the definition of this number.

Then, from the fact that networks with clique size greater than 4 are not planar, we obtain by Theorem 2 that the problem we consider is hard even when the network has a constant maximum clique-size. We formulate this observation as follows.

Observation 3. *q-majority illusion is* **NP***-complete, even for networks with maximum clique-size bounded by some constant greater than 4, for every rational* $q \in (\frac{1}{2}, 1]$.

3.3 FPT Algorithm for Tree Width and Maximum Degree

Our **NP**-completeness results for q-majority illusionmotivate the study of this problem from the perspective of parameterised complexity, with the aim of identifying various restrictions on its input, which allow for tractability. Note that our result that q-majority illusionis **NP**-complete on networks of constant max-degree implies that, unless **P**=**NP**, q-majority illusiondoes not have an algorithm deciding it, with a running time $|N|^{f(\Delta)}$, for any computable function f, where Δ is the max-degree. In other words, q-majority illusionis para-**NP**-hard when parameterised by Δ. Hence, we extend this parameterisation, using other structural properties of the graph.

Our first fixed-parameter tractability result, that is, Theorem 3, states that if we parameterise q-majority illusion by the max-degree *and* tree width of the input network, then we can obtain a FPT-algorithm. The idea behind our proof is that we can use dynamic programming over a nice tree decomposition of a network to check if it admits q-majority illusion, assuming that the maximum degree of vertices in this network is bounded.

We next prove our first parameterised tractability result.

Theorem 3. *q-majority illusion can be solved in time* $\Delta^{O(k)}|N|^{O(1)}$ *on networks of tree width k and max-degree* Δ.

Proof. Let the input social network be *SN*, with tree width k. We first run the $2^{O(k)}|N|^{O(1)}$-time 2-approximation algorithm of [34], in order to compute a tree decomposition of width at most $2k+1$, and then use the well-known polynomial-time algorithm to convert any given tree decomposition to a nice tree decomposition of the same width (see [16]). We now design a dynamic programming algorithm over this nice tree decomposition (T, β), of width at most $2k+1$.

We define a boolean function H (i.e., to the set $\{0,1\}$), whose domain is the set of all tuples, where each tuple comprises a vertex $t \in V(T)$, a labelling $\mathsf{col} : \beta(t) \to \{r, b\}$ of vertices in the bag $\beta(t)$, a function $\mathsf{esurp} : V_t \to \{-\Delta, \dots, \Delta\}$, where $\mathsf{esurp}(v) = 0$ for all vertices $i \notin \beta(t)$, a function $\mathsf{isurp} : \beta(t) \to \{-\Delta, \dots, \Delta\}$, some $\alpha \in [0, |N|]$, and some $\ell_r \in [0, |N|]$. If $\beta(t) = \emptyset$, then we have that $\mathsf{col} = \mathsf{esurp} = \mathsf{isurp} = \emptyset$. We further define $H(t, \mathsf{col}, \mathsf{esurp}, \mathsf{isurp}, \alpha, \ell_r) = 1$ if and only if there exists a labelling $\rho : V_t \to \{r, b\}$, such that the following hold:

(1) For every $i \in \beta(t)$, we have that $\rho(i) = \mathsf{col}(i)$.
(2) The size of the set $R_\rho^{V_t} = \{i \in V_t : \rho(i) = r\}$ is ℓ_r.
(3) α is the size of the set:

$$\{i \in V_t : |N_{\rho,r}^{V_t}(i)| > |N_{\rho,b}^{t}(i)| + \mathsf{esurp}(i)\}$$

(4) For every $i \in \beta(t)$, we have that $\mathsf{isurp}(i) = |N_{\rho,b}^{V_t}(i)| - |N_{\rho,r}^{V_t}(i)|$ captures the internal blue surplus of every vertex in $\beta(t)$ under ρ , that is, how many more blue than red vertices are in i's neighbourhood within V_t.

The intuition behind the description of the function H is the following. Consider a hypothetical labelling f for the social network $SN = (N, E)$ that witnesses q-majority illusion. Then, fix a bag $\beta(t)$, and let δ be the restriction of f to the set V_t. Subsequently, we have that:

(1) col is the restriction of δ to the vertices of the bag $\beta(t)$.

(2) The function esurp (read *external surplus*) describes the blue surplus for the vertices in V_t, that is, provided by the vertices outside of the set V_t. Note that then only vertices of the bag $\beta(t)$ get non-zero blue surplus from outside of V_t, since only these vertices (among those in V_t) have any neighbours outside of V_t, by the definition of a tree decomposition. Hence, we may assume a value of 0 "external" blue surplus, for all vertices in V_t, which are not in $\beta(t)$. On the other hand, since the max-degree of the graph is Δ, the "external" blue surplus of any vertex in $\beta(t)$ is at least $-\Delta$ and at most Δ.
(3) The value of ℓ_r is the number of vertices of V_t that are assigned r by f, and hence also by δ.
(4) The number α is the number of vertices of V_t which are under illusion with respect to f. This includes all vertices in $V_t \setminus \beta(t)$ that have more red neighbours than blue neighbours under δ, and all vertices in $\beta(t)$, for which, if we add the blue surplus given by vertices in V_t (which can be deduced from δ) and the blue surplus from outside V_t (which is given by the function esurp), we get at most -1.
(5) Finally, the function isurp (read *internal surplus*) describes the blue surplus for the vertices in $\beta(t)$, which is provided by the vertices within V_t. As for esurp, since the max-degree is Δ, we have that the range of the function lies in $\{-\Delta, \dots, \Delta\}$.

The crux of the correctness of the procedure which we will define is that if we could find a labelling, say ρ, for V_t, which is not necessarily in accordance with δ, but has the same "signature" of δ in terms of col, ℓ_r, α, isurp, then, given the same esurp, then we can "cut" δ from f and replace it with ρ. This allows us to obtain another labelling of *SN*, which has exactly the same number of vertices under illusion as γ. This gives us the so-called *optimal substructure property*, which is crucial for our dynamic programming algorithm.

Notice that there are only $2^{2k+2} \cdot (2\Delta + 1)^{2(2k+2)} |N|^{O(1)} = \Delta^{O(k)} |N|^{O(1)}$ possible tuples. This is because each bag contains at most $2k + 2$ vertices, implying at most 2^{2k+2} possibilities for col at any bag and since, for every bag, we have that esurp can only have non-zero values for vertices in the bag (and at most $2\Delta + 1$ possible values at that), we infer that there are at most $(2\Delta + 1)^{2k+2}$ possibilities for esurp at any bag. The same bound extends to isurp as well. The remaining elements of the tuple, that is, α and ℓ_r, are both bounded by $|N|$, and hence there are at most $|N|^2$ possibilities for them at any bag.

Now, suppose that we have computed $H(t, \mathsf{col}, \mathsf{esurp}, \mathsf{isurp}, \alpha, \ell_r)$ for all possible valid values of the arguments. Notice that if this is achieved, then we can answer whether *SN* admits q-majority illusion by examining the table entries corresponding to the root bag $\beta(t^\star)$. Observe that, by the definition of a nice tree decomposition, this bag is empty. Then, we have that *SN* admits q-majority illusion if and only if there exist values $l_r \in [0, |N|]$ and $\alpha \in [0, |N|]$, such that $\alpha \geq \lceil q \cdot |N| \rceil$, $l_r < \frac{|N|}{2}$ and $H(\emptyset, \emptyset, \emptyset, \alpha, \ell_r) = 1$.

We next describe, how to compute the table entries at each bag by going over the following, exhaustive, cases and assuming that all the table entries at all descendant bags have been computed correctly.

Leaf Node. Let t be a leaf node. This is our base case. By the definition of a nice tree decomposition, we have that $\beta(t) = \emptyset$. Then, we set $H[t, \emptyset, \emptyset, \emptyset, 0, 0] = 1$. For all other values of α and ℓ_r, we set $H[t, \emptyset, \emptyset, \emptyset, \alpha, \ell_r] = 0$.

Introduce Node. Let t be an introduce node and t' be its child in T, such that $\beta(t) \setminus \beta(t') = \{u\}$. Then, consider the tuple $(t, \mathsf{col}, \mathsf{esurp}, \mathsf{isurp}, \alpha, \ell_r)$, for which we want to fill the table entry.

We next define the tuple $(t', \mathsf{col}', \mathsf{esurp}', \mathsf{isurp}', \alpha', \ell_r')$. Let col$'$ denote the restriction of col to $\beta(t')$. If $\mathsf{col}(u) = r$, then we set $\ell_r' := \ell_r - 1$, and otherwise we set $\ell_r' = \ell_r$. Let $\mathsf{esurp}' : V_{t'} \to \{-\Delta, \dots, \Delta\}$, and $\mathsf{isurp}' : \beta(t) \to \{-\Delta, \dots, \Delta\}$, be defined as follows. For every vertex v in $V_{t'} \setminus \beta(t')$, set $\mathsf{esurp}'(v) = 0$. Also, for every vertex v in $\beta(t)$, which is a neighbour of u, if $\mathsf{col}(u) = r$, then we set $\mathsf{esurp}'(v) = \mathsf{esurp}(v) - 1$, and set $\mathsf{isurp}'(v) = \mathsf{isurp}(v) + 1$. Further, for every vertex v in $\beta(t)$, which is a neighbour of u, if $\mathsf{col}(u) = b$, then we set $\mathsf{esurp}'(v) = \mathsf{esurp}(v) + 1$, and we set $\mathsf{isurp}'(v) := \mathsf{isurp}(v) - 1$. Finally, we define α' as follows. If $\mathsf{esurp}(u)$ plus the number of neighbours of u in $\beta(t')$, which are labelled blue under col, minus the number of neighbours of u in $\beta(t')$, which are labelled red under col is at most -1, then we set $\alpha' = \alpha - 1$. Otherwise, we set $\alpha' = \alpha$. Now, we set:

$$H[t, \text{col}, \text{esurp}, \text{isurp}, \alpha, \ell_r] := H[t', \text{col}', \text{esurp}', \text{isurp}', \alpha', \ell_r'].$$

Forget Node. Let t be a forget node and t' be its child in T, such that $\beta(t') \setminus \beta(t) = \{u\}$. Consider the tuple $(t, \text{col}, \text{esurp}, \text{isurp}, \alpha, \ell_r)$, for which we want to fill the table entry. We set $H[t, \text{col}, \text{esurp}, \text{isurp}, \alpha, \ell_r] = 1$ if and only if there exists $\text{col}', \text{esurp}', \text{isurp}'$, such that (1) $H[t', \text{col}', \text{esurp}', \text{isurp}', \alpha, \ell_r] = 1$, (2) col is the restriction of col' to $\beta(t)$, and (3) esurp (isurp) is the restriction of esurp' (respectively, isurp') to $\beta(t)$.

Join Node. Let t be a join node and t_1, t_2 be its children in T. Then, by the definition of a nice tree decomposition, we have that $\beta(t) = \beta(t_1) = \beta(t_2)$. Consider the tuple $(t, \text{col}, \text{esurp}, \text{isurp}, \alpha, \ell_r)$, for which we want to fill the table entry. We set $H[t, \text{col}, \text{esurp}, \text{isurp}, \alpha, \ell_r] = 1$ if and only if for some pair of tuples $(t_1, \text{col}, \text{esurp}_1, \text{isurp}_1, \alpha_1, \ell_{r,1})$ and $(t_2, \text{col}, \text{esurp}_2, \text{isurp}_2, \alpha_2, \ell_{r,2})$, it holds that:

(1) Entries $H[t_1, \text{col}, \text{esurp}_1, \text{isurp}_1, \alpha_1, \ell_{r,1}]$ and $H[t_2, \text{col}, \text{esurp}_2, \text{isurp}_2, \alpha_2, \ell_{r,2}]$ are both 1.
(2) $\alpha = \alpha_1 + \alpha_2 - x$, where x is the number of vertices of $\beta(t)$ forced to be under illusion by the combination of esurp and isurp. That is, x is the size of the set $\{v \in \beta(t) : \text{esurp}(v) + \text{isurp}(v) \leq -1\}$. We are subtracting x from $\alpha_1 + \alpha_2$, because these vertices are counted in both α_1 and α_2.
(3) $\ell_r = \ell_{r,1} + \ell_{r,2} - y$, where y is the number of vertices of $\beta(t)$ labelled red by col. That is, y is the size of the set $\{v \in \beta(t) : \text{col}(v) = r\}$. We are subtracting y from $\ell_{r,1} + \ell_{r,2}$, because the vertices of $\beta(t)$ labelled red by col is counted once in $\ell_{r,1}$, and once in $\ell_{r,2}$.
(4) For every $v \in \beta(t_1)$, we have that $\text{esurp}_1(v) = \text{esurp}(v) + \text{isurp}_2(v) - |N|^{\beta(t_1)}_{\text{col},b}(v) + |N|^{\beta(t_1)}_{\text{col},r}(v)$. Here, we are saying that the blue surplus of a vertex v, external to the set V_{t_1}, should be obtained by taking the blue surplus of v, external to both V_{t_1} and V_{t_2} (which is given by $\text{esurp}(v)$), and then adding to it the blue surplus of v *internal* to V_{t_2} (while accounting for edges between vertices in $\beta(t_1)$). Precisely, we subtract $|N|^{\beta(t_1)}_{\text{col},b}(v) - |N|^{\beta(t_1)}_{\text{col},r}(v)$, because these quantity deals with blue surplus, which is given by the edges between vertices of $\beta(t_1)$, and these should not be counted in the external surplus of v with respect to the bag $\beta(t_1)$.
(5) For every $v \in \beta(t_2)$, we have that $\text{esurp}_2(v) = \text{esurp}(v) + \text{isurp}_1(v) - |N|^{\beta(t_2)}_{\text{col},b}(v) + |N|^{\beta(t_2)}_{\text{col},r}(v)$. The reasoning behind this constraint is symmetrical to the previous one.
(6) For every $v \in \beta(t)$, we have that $\text{isurp}(v) = \text{isurp}_1(v) + \text{isurp}_2(v) - |N|^{\beta(t)}_{\text{col},b}(v) + |N|^{\beta(t)}_{\text{col},r}(v)$. Here, we are saying that the surplus of v internal to V_t should be obtained by taking the surplus of v internal to V_{t_1} and to V_{t_2}, and adding them, while accounting for the fact, that we are double-counting the contribution of edges within $\beta(t)$. This motivates the subtraction of $|N|^{\beta(t)}_{\text{col},b}(v) - |N|^{\beta(t)}_{\text{col},r}(v)$.

Notice that filling all the table entries corresponding to any specific bag is dominated the time taken for the join nodes, which in turn is dominated by $|N|^{O(1)}$ times the number of possible tuples to consider, from each of the two children bags. Hence, the time taken to fill the entries for any one bag is bounded by $\Delta^{O(k)}|N|^{O(1)}$, and as we have argued earlier, there are at most $\Delta^{O(k)}|N|^{O(1)}$ possible tuples corresponding to each bag. The stated running time then follows. This completes the proof of the theorem.

□

We next discuss some immediate implications of the above result. First of all, notice that Δ, that is, the max-degree, is at most $|N| - 1$. Hence, our FPT-algorithm, parameterised by Δ and the tree width, is in fact an XP-algorithm, parameterised by the tree width alone.

Corollary 1. *q-*majority illusion *can be solved in time $|N|^{O(k)}$ on networks of tree width k.*

Secondly, consider the following relation between tree width and another well-studied graph width parameter, that is, clique width [27], which we denote by $cw(G)$, on bounded-degree graphs. We make use of the following result.

Proposition 1. [27] *Let G be a graph that does not contain the complete bipartite graph $K_{d,d}$ as a subgraph. If $cw(G) \leq k$, then it holds that $tw(G) \leq 3k(d-1) - 1$.*

As graphs with maximum degree Δ exclude $K_{\Delta+1,\Delta+1}$ as a subgraph, Proposition 1, along with Theorem 3, implies that q-majority illusion is in **FPT** when parameterised by the maximum degree and clique width of the input graph.

Corollary 2. majority illusion *can be solved in time $\Delta^{O(\Delta \cdot k)}|N|^{O(1)}$ on networks of max-degree Δ and clique width k.*

3.4 FPT Algorithm for Neighbourhood Diversity

Here, we provide an FPT algorithm for q-majority illusion parameterised by neighbourhood diversity. The following properties of labellings of social networks form the crux of our algorithm.

Lemma 10. *Let (N, E) be a social network, and let $C = \{T_1, \ldots, T_k\}$ denote a partition of N into k modules. Further, let $f : N \rightarrow \{r, b\}$ be a labelling, where b is the majority label. Then, the following hold.*

(1) If one vertex of an independent module is under illusion under f, then every vertex of this module is under illusion.
(2) If a blue vertex of a clique module is under illusion under f, then all blue vertices in this module are also under illusion.
(3) If a red vertex of a clique module is under illusion under f, then every vertex in this module is also under illusion.

Proof. Let us first of all recall that, by the definition of neighbourhood diversity, we have that every pair of vertices in each module have exactly the same neighbourhood outside the module. Hence, the first statement immediately follows since, in an independent module, there are no edges within the module.

Now, consider the second statement, and fix a clique module C. Let further $u, v \in C$ be a pair of blue vertices. Then, the number of red (blue) neighbours of u within C is exactly the same, as the number of red (respectively, blue) neighbours of v within C. This implies that if u is under illusion, then so is v.

Now, consider the third statement. By an argument identical to that for the second statement, we can conclude that if a red vertex is under illusion, then all red vertices are under illusion. It is now sufficient to argue that if a red vertex is under illusion, then at least one blue vertex (assuming it exists) is under illusion. Let u be a red vertex in the module that is under illusion, and let v be a blue vertex in the module. Recall that u and v have the same neighbourhood outside C. Moreover, the number of blue neighbours of u within C is strictly greater than the number of blue neighbours of v in C. Consequently, u has strictly fewer red neighbours within C than v. This implies that if u is under illusion, then so is v. This completes the proof of the lemma.

□

We will use as a subroutine the well-known FPT algorithm for ILP-Feasibility. The ILP-Feasibility problem is defined as follows. The input is a matrix $A \in \mathbb{Z}^{m \times p}$ and a vector $b \in \mathbb{Z}^{m \times 1}$ and the objective is to find a vector $\bar{x} \in \mathbb{Z}^{p \times 1}$ satisfying the m inequalities given by A, that is, $A \cdot \bar{x} \leq b$, or decide that such a vector does not exist.

Proposition 2. [29, 30, 23] ILP-Feasibility *can be solved using $O(p^{2.5p+o(p)} \cdot L)$ arithmetic operations and space polynomial in L, where L is the number of bits in the input and p is the number of variables.*

Intuitively, we will make use of Lemma 10 to construct a number of ILP-Feasibility instances, sufficient to solve q-majority illusion. Towards this end, following Claim 1, we show that if the proposed constraints are satisfied, then we can conclude that the network admits q-majority illusion. This will allow us to subsequently solve q-majority illusionefficiently, for small value of neighbourhood diversity.

Theorem 4. *q-majority illusioncan be solved in time $2^{O(k \log k)}|N|^{O(1)}$ on networks of neighbourhood diversity k.*

Proof. Let $SN = (N, E)$ be a given input social network, and let $\mathcal{T} = \{T_1, \ldots, T_k\}$ denote the partition of N into k modules, each of which is a clique or an independent set. Observe that the set $\mathcal{T}$ can be computed in polynomial time (see [35]). Then, for every $i \in [1, k]$, let $adj(i)$ denote the set $\{j \in [1, k] : j \neq i$ and $\exists u \in T_i, v \in T_j : (u, v) \in E\}$. That is, $adj(i)$ comprises the indices of all those modules T_j, in which at least one vertex (and hence all vertices) is adjacent to a vertex of T_i (and hence to all vertices of T_i). Let further $\chi = \lceil q \cdot |N| \rceil$ denote the required number of vertices to be under illusion, in order for q-majority illusion to hold. The main intuition behind our algorithm is to construct $2^{O(k)}$ instances of ILP-Feasibility, each with $O(k)$ variables, such that if there is a labelling of SN, which induces q-majority illusion, then the solution to one of these ILP-Feasibility instances can be used to obtain a solution to the given instance of q-majority illusion.

Let now C denote the set of all clique modules in $\mathcal{T}$, and let $\mathcal{I}$ denote the set of all independent modules in $\mathcal{T}$. We are now ready to start describing the design of the ILP-Feasibility instances. For every function:

- Clique-col: $C \longrightarrow \{r, b, \text{both}\}$
- Clique-maj: $C \longrightarrow \{b, \text{all}, \text{none}\}$
- Ind-maj: $\mathcal{I} \longrightarrow \{\text{all}, \text{none}\}$

we construct one ILP-feasibility instance, for which the set of variables and constraints will be discussed later in this proof. We first sketch the intuition behind these functions. Let $f : N \longrightarrow \{r, b\}$ be a labelling, which places at least χ vertices under illusion (if one exists). Then, the function Clique-col expresses, for every clique module, whether it contains both red and blue vertices according to f. Note that if this is case, then this module is mapped to both. Further, if it contains only red vertices, then this module is mapped to r. Finally, if it contains only blue vertices, then this module is mapped to b.

Furthermore, the function Clique-maj expresses, for every clique module, whether no vertices are under illusion (mapped to none), or only blue vertices are under illusion (mapped to b), or all vertices are under illusion (mapped to all) under f. Recall that from the second and third statements of Lemma 10, we have that these are the only three possibilities.

The function Ind-maj expresses, for every independent module, whether all vertices in the module are under illusion (mapped to all) in the optimal labelling, or none of them are under illusion (mapped to none). Recall, that from the first statement of Lemma 10, we have that these are the only two possibilities. If f exists, then a "correct" triple of these functions exist. Notice that there are at most 3^k possibilities for Clique-col and Clique-maj, and at most 2^k possibilities for Ind-maj. Hence, we may iterate over all possible at most 18^k triples of functions, and we know that at least one of these triples is the "correct" one if ρ exists.

Now, let us fix the functions Clique-col, Clique-maj, Ind-maj and describe the ILP-Feasibility instance corresponding to it. To better understand the constraints we will design, we consider the three selected functions to be the "correct" ones that correspond to f. We will also assume that these functions are consistent with each other. That is, if Clique-col(T_i) is r (respectively, b), then it cannot be the case that Clique-maj(T_i) is b (respectively, r). In other words, if we guess that every vertex of T_i is labelled red, then we will not guess that all of the blue vertices of T_i will be under illusion. Moreover, we have a convention that in Clique-maj, the value all takes "priority" over r or b. That is, if Clique-col(T_i) is b, then Clique-maj(T_i) is either none or all, and never b. This is because setting it to all achieves the same effect as setting it to b, since all vertices in T_i are blue. Any triple of functions where these conditions are not satisfied are not considered further.

We now proceed to describe the ILP-Feasibility instance. For every $i \in [1, k]$, let s_i denote the size of $V(T_i)$. We know the value of each s_i, since we have we know $\mathcal{T}$. The set of variables in this instance is $\bigcup_{i \in [1,k]} \{r_i, b_i, p_i\}$. The intuitive meaning of these variables is the following. Recall that $f : V(SN) \longrightarrow \{r, b\}$ is a hypothetical optimal labelling that places at least χ vertices under illusion. Then, for every module T_i, we have that r_i represents the number of vertices of T_i labelled red in f. Similarly, b_i is the number of vertices in T_i labelled blue in f, while p_i

will be used to represent the number of vertices of T_i, which are under illusion following f. Notice that we have $3k$ variables in total. We are now ready to present our constraints.

(1) For every $i \in [1, k]$, we have the constraint $r_i + b_i = s_i$[3]. This constraint says that the numbers of red and blue vertices of each module should add up to the total number of vertices in that module.
(2) $-\sum_{i \in [1,k]} b_i + \sum_{i \in [0,k]} r_i \leq -1$. This constraints says that the total number of vertices labelled blue must exceed the total number of vertices labelled red. So, as a consequence, we get that blue would be the majority label.
(3) $-\sum_{i \in [1,k]} p_i \leq -\chi$. This says that as long as we ensure that the number of vertices under illusion from each T_i is at least p_i, then the total number of vertices under illusion is at least χ.
(4) For every $i \in [1, k]$, such that T_i is an independent module, if $\mathsf{Ind\text{-}maj}(T_i) = \mathsf{all}$, then we add the constraint $\sum_{j \in adj(i)} b_j - \sum_{j \in adj(i)} r_j \leq -1$. That is, for every vertex in T_i, the number of red neighbours must exceed the number of blue neighbours, that is, they are all under illusion.
(5) For every $i \in [1, k]$, such that T_i is an independent module, if $\mathsf{Ind\text{-}maj}(T_i) = \mathsf{all}$, then we add the constraint $p_i = s_i$, and otherwise (i.e., when $\mathsf{Ind\text{-}maj}(T_i) = \mathsf{none}$), we add the constraint $p_i = 0$.
(6) For every $i \in [1, k]$, such that T_i is a clique module, we add a constraint as follows:
   - If $\mathsf{Clique\text{-}maj}(T_i) = \mathsf{none}$, then $p_i = 0$.
   - If $\mathsf{Clique\text{-}maj}(T_i) = b$, then $p_i = b_i$.
   - If $\mathsf{Clique\text{-}maj}(T_i) = \mathsf{all}$, then $p_i = s_i$.

   These constraints ensure that the number of vertices of T_i that are supposed to be under illusion match with the information provided by the function Clique-maj, that is, whether the set of vertices under illusion is empty, or is equal to the set of all blue vertices, or to all of the vertices in the module.
(7) For every $i \in [1, k]$, such that T_i is a clique module, we do the following:
   - If $\mathsf{Clique\text{-}col}(T_i) = r$, then we add the constraint $s_i = r_i$. Further, if it also holds that $\mathsf{Clique\text{-}maj}(T_i) = \mathsf{all}$, then we add the constraint:
   $$-(r_i - 1) + \sum_{j \in adj(i)} (b_j - r_j) \leq -1$$
   - If $\mathsf{Clique\text{-}col}(T_i) = b$, then add the constraint $s_i = b_i$. Further, if it also holds, that $\mathsf{Clique\text{-}maj}(T_i) = \mathsf{all}$, then we add the constraint:
   $$b_i - 1 + \sum_{j \in adj(i)} (b_j - r_j) \leq -1$$

   These constraints say that if every vertex in T_i is labelled red (blue), then the number of red vertices (respectively, blue vertices) is the total number of vertices in T_i. Moreover, if every vertex is required to be under illusion according to the function Clique-maj, then the blue surplus of any vertex in T_i is at most -1.
(8) For every $i \in [1, k]$, such that T_i is a clique module with $\mathsf{Clique\text{-}col}(T_i) = \mathsf{both}$, we do the following:
   - If $\mathsf{Clique\text{-}maj}(T_i) = \mathsf{all}$, then we add the constraint:
   $$b_i - r_i + 1 + \sum_{j \in adj(i)} (b_j - r_j) \leq -1$$
   This constraint says that if we take a red vertex in T_i, and compute its blue surplus, then it is at most -1. That is, it is under illusion. This in turn implies that every vertex is under illusion, as required by $\mathsf{Clique\text{-}maj}(T_i)$.

[3]Notice that equality constraints can always be expressed as two inequalities, as required in the definition of ILP-Feasibility. For instance, in this case, we have $r_i + b_i \leq s_i$ and $-r_i - b_i \leq -s_i$.

- If $\mathsf{Clique\text{-}maj}(T_i) = b$, then we add the constraint

$$b_i - r_i - 1 + \sum_{j \in adj(i)} (b_j - r_j) \leq -1$$

This constraint says that if we take a blue vertex in T_i, and compute its blue surplus, then it is at most -1.

(9) For every variable $x \in \bigcup_{i \in [1,k]} \{r_i, b_i, p_i\}$, we have a constraint $-x \leq 0$, which imposes a non-negativity constraint on every variable. This will allow us to treat r_i, b_i, p_i as sizes of vertex sets.

(10) Finally, for every $i \in [1, k]$, we add the constraint $p_i \leq s_i$, in order to indicate that the number of vertices of T_i, which are under illusion, can never be more than the total number of vertices in T_i.

This completes the description of the ILP-Feasibility instance. We refer to the previously defined constraints as C1-C10. Observe that the ILP-Feasibility instance can be computed in polynomial time, given the three functions which we consider in such an instance.

One can further observe that for the optimal labelling (in terms of the number of vertices under illusion) ρ, as well as the corresponding three functions, these constraints are satisfied. This implies that if ρ exists, then at least for one of the triples, the corresponding ILP-Feasibility instance can be solved – for each $i \in [1, k]$, set r_i (b_i) to be the number of vertices of T_i, which are labelled red (blue) by f, and set p_i to be the number of vertices of T_i under illusion.

We next show that if we solve the ILP-Feasibility instance for some triple, then we can recover a labelling inducing the required q-majority illusion (which may not be the same as f). Let $\bigcup_{i \in [1,k]} \{r_i^*, b_i^*, p_i^*\}$ be a solution for the ILP-Feasibility instance. Observe that due to C9, we have that all variables get non-negative values.

We now define a labelling f^* as follows. For every $i \in [1, k]$, we select an arbitrary set of r_i^* vertices from T_i, and label them red. Furthermore, we label the remaining vertices of each T_i (of which must there must be exactly b_i^*, due to C1), with blue. Since C2 is satisfied, it follows that blue is the majority label. We next prove the following claim, which, along with C3, would then imply that at least χ vertices in total are under illusion, as required.

Claim 1. *For every $i \in [1, k]$, the number of vertices of T_i under illusion, with respect to f^*, is at least p_i^*.*

Proof. Consider an independent module T_i. Suppose that $\mathsf{Ind\text{-}maj}(T_i) = \mathsf{all}$. Since C4 is satisfied, it follows that all s_i vertices of T_i are under illusion. Moreover, C5 implies that $p_i^* = s_i$, hence validating our claim that the number of vertices of T_i under illusion with respect to f^* is at least p_i^*. On the other hand, if $\mathsf{Ind\text{-}maj}(T_i) = \mathsf{none}$, then $p_i^* = 0$, and the claim is trivially true, because the number of vertices of T_i under illusion is always at least 0. The same argument applies if we consider a clique module T_i, such that $\mathsf{Clique\text{-}maj}(T_i) = \mathsf{none}$. That is, $p_i^* = 0$, and the claim is trivially true, because the number of vertices of T_i under illusion is always at least 0. Hence, we assume that we are only left with clique modules T_i, such that $\mathsf{Clique\text{-}maj}(T_i) \neq \mathsf{none}$. Now, we have the following, exhaustive, subcases.

Case 1: $\mathsf{Clique\text{-}col}(T_i) = \boldsymbol{r}$. Since we have assumed that $\mathsf{Clique\text{-}maj}(T_i) \neq \mathsf{none}$, it must be the case that $\mathsf{Clique\text{-}maj}(T_i) = \mathsf{all}$. Then, C7 guarantees that $-(r_i^* - 1) + \sum_{j \in adj(i)} (b_j^* - r_j^*) \leq -1$. But notice then that we have labelled exactly r_j^* vertices of each T_j red, and the remaining vertices blue. Hence, it must be the case that the blue surplus of every vertex in T_i (as expressed in C7) is at most -1, and so all of the s_i vertices of T_i are under illusion. This satisfies the claim, since p_i^* is always at most s_i (due to C10).

Case 2: $\mathsf{Clique\text{-}col}(T_i) = \boldsymbol{b}$. Again, it must be the case that $\mathsf{Clique\text{-}maj}(T_i) = \mathsf{all}$. Then, C7 guarantees that, since it holds that $b_i^* - 1 + \sum_{j \in adj(i)} (b_j^* - r_j^*) \leq -1$, and we have labelled exactly r_j^* vertices of each T_j red and the remaining blue, it follows that the blue surplus of every vertex in T_i is at most -1. So, every vertex of T_i is under illusion. As before, this satisfies the claim, since $p_i^* \leq s_i$ (C10).

Case 3: Clique-col(T_i) = both. In this case, Clique-maj(T_i) could be all or b. In the former case, C8 guarantees that $b_i^* - r_i^* + 1 + \sum_{j \in adj(i)} (b_j^* - r_j^*) \leq -1$, implying that at least one red vertex is under illusion. So, by Lemma 10(3), we have that every vertex in T_i is under illusion. In the latter case, C8 guarantees that $b_i^* - r_i^* - 1 + \sum_{j \in adj(i)} (b_j^* - r_j^*) \leq -1$, implying that at least one blue vertex is under illusion, and so, by Lemma 10(2), we have that every blue vertex in T_i is under illusion. Hence, the number of vertices of T_i under illusion is at least the number of blue vertices, that is, b_i^*. However, in this case C6 guarantees that $p_i^* = b_i^*$ and hence, the number of vertices under illusion in T_i is again at least p_i^*, as required. This completes the proof of the claim.

□

We have argued the correctness of the algorithm. Notice further that the running time is bounded by the time required to compute $\mathcal{T}$, which is polynomial, plus 18^k, multiplied by the time required to construct an ILP-Feasibility instance and to execute the algorithm from Proposition 2, which is bounded by $2^{O(k \log k)} |N|^{O(1)}$. This gives an overall bound of $2^{O(k \log k)} |N|^{O(1)}$ on our algorithm. This completes the proof of the theorem.

□

Since graphs with vertex cover number at most k have neighbourhood diversity at most $k + 2^k$ (see [35]), Theorem 4 has the following corollary.

Corollary 3. *q-majority illusioncan be solved in time $2^{2^{O(k)}} |N|^{O(1)}$, on networks with vertex cover number k.*

4 Eliminating Illusion

We now turn to the problem of reducing the number of vertices under illusion in a given labelled network by modifying the connections between them. Namely, we consider the problem of checking if it is possible to ensure that q-majority illusion does not hold in a labelled network, by altering only a bounded number of edges.[4]

q-Illusion Elimination:
Input: $SN = (N, E, f)$, such that f induces q-majority illusion in (N, E, f), $k \in \mathbb{N}$, such that $k \leq |E|$.
Question: Is there a $SN' = (N, E', f)$, such that $|\{e \in N^2 : e \in E \textit{ iff } e \notin E'\}| \leq k$, while f does not induce q-majority illusion in SN'?

We also consider two refinements of this problem. First, let us introduce Addition q-Illusion Elimination, in which we restrict the possible actions to *adding edges*.

Addition q-Illusion Elimination:
Input: $SN = (N, E, f)$, such that f induces q-majority illusion in SN, $k \in \mathbb{N}$, such that $k \leq |E|$.
Question: Is there a $SN' = (N, E', f)$, such that $E \subseteq E'$, $|E'| - |E| \leq k$, and f does not induce q-majority illusion in SN'?

We also define Removal q-Illusion Elimination, for *removing edges*.

Removal q-Illusion Elimination:
Input: $SN = (N, E, f)$ such that f induces q-majority illusion in SN, $k \in \mathbb{N}$ such that $k \leq |E|$.
Question: Is there a $SN' = (N, E', f)$ such that $E' \subseteq E$, $|E| - |E'| \leq k$ and f does not induce q-majority illusion in SN'?

First, in Section 4.1 we demonstrate computational hardness of q-Illusion Elimination. Then, in Section 4.2 we provide analogous results for Addition q-Illusion Elimination and Removal q-Illusion Elimination.

[4]Note that the condition specified in the question corresponds exactly to the number of edges present in the original network but not in the altered one, and the number of those only present in the original network. This captures the number of operations of rewiring of the edges.

4.1 NP-Completeness of Illusion Elimination

In this section, we show that q-Illusion Elimination is **NP**-complete. In fact, our construction implies that this problem is also $\mathbf{W}[1]$-hard when parameterised by the number of changed edges in a social network. We obtain this by providing the required reduction from k-clique. In the following, we give a sketch of our reduction to convey the necessary intuition behind our proof, which is followed by a formal proof.

Let us first introduce some useful notation. For a pair of labelled social networks (N, E, f), (N, E', f), and $i \in N$, we say that i is *pushed into illusion* in (N, E', f), if i is under illusion in (N, E', f), but not in (N, E, f). Symmetrically, we say that illusion is *eliminated* from i in (N, E', f) if i is under illusion in (N, E, f) but not in (N, E', f).

Consider an instance (G, k) of k-Clique, where $G = (V_G, E_G)$. We now construct a social network (N, E, f) as follows. First, we add the vertex set V_G to N, and the edge set E_G to E. We further assign each vertex i in V_G the label red, that is $f(i) = r$. Next, for each vertex $i \in V_G$, we add a set r_i of red labelled vertices, and a set b_i of blue labelled vertices, while ensuring that number of red neighbours of i is exactly $k-1$ more than the number of blue neighbours of i. So, for a vertex i in V_G for which $N^{V_g}(i) \geq k$, we set $|b_i| = |N^{V_g}(i)| - k + 1$ and $|r_i| = 0$. Also, if $|N^{V_g}(i)| < k$, we set $|r_i| = k - |N^{V_g}(i)| - 1$ and $|b_i| = 0$.

The idea behind adding these vertices is to make sure that each vertex in V_G has a red surplus of exactly $k-1$. Then, the vertices in V_G are under illusion. Now, we impose the condition that only the vertices in V_G remain under illusion by adding, for each vertex j in the sets r_i and b_i, two blue labelled vertices j_1, j_2 and adding edges $(j_1, j), (j_2, j), (j_1, j_2)$. Then, j is not under illusion, as it has two blue labelled neighbours, as well as one red labelled neighbour. Moreover, j_1, j_2 are not under illusion, as they have one red labelled and one blue labelled neighbour. We show, in Lemma 13, that it is possible to eliminate illusion from k vertices in this structure by altering at most $\binom{k}{2}$ edges exactly when there exists a k-clique in G. Next, we add some extra red and blue labelled vertices, which are not under illusion, to guarantee that blue is the majority label, and the ratio of vertices under illusion minus k, to the total number of vertices, is at most q. We set our budget (of edge modifications in the network) to be $\binom{k}{2}$, that is, $\frac{k^2-k}{2}$. This completes our reduction.

To argue the correctness of our reduction, we show that in order to remove q-majority illusion from the constructed network, we must make sure that at least k more vertices are not under illusion, and that these must come from V_G, as only vertices in V_G are under illusion. To achieve this, at least $k-1$ edges on each such vertex must be removed. Achieving this goal by deleting at most $\binom{k}{2}$ edges is only possible if there is a clique on k vertices in G. Conversely, if we can delete any $\binom{k}{2}$ edges to make k vertices under illusion in (N, E, f), then there must be a k-clique in G. We show that this is the case in the proof of Lemma 14, which implies the $\mathbf{W}[1]$ hardness of the problem we consider.

We start with two additional structures, which we call *m-pump-up* and *m-pump-down* gadgets. These gadgets will help us to ensure the correctness of the construction which we will provide for a chosen q. We note that adding them to a network in which r is the minority label does not affect the fact that b is the majority label.

m-*Pump-Up Gadget.* Let us construct what we call an *m-pump-up gadget*. For a natural number $m \geq 1$, we create $m+4$ blue vertices, which are not connected to each other. In addition, we construct 4 red vertices, which are also not connected to each other. Furthermore, let each red vertex in the gadget be connected to all blue vertices in this structure. Note that this forms $K_{m+4,4}$. Observe that if a m-pump-up gadget is embedded in a network in which blue is the majority label, then $m+4$ vertices are under illusion in this structure, while 4 are not. Also, for every blue vertex i in the gadget, the blue surplus of i is -4. Figure 8 depicts a 2-pump-up gadget.

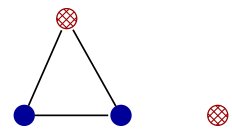

Fig. 9. m-pump-down gadget for $m = 4$.

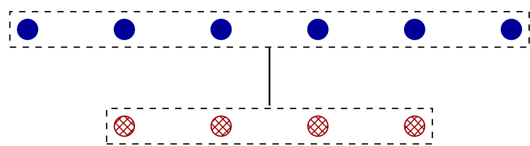

Fig. 8. m-pump-up gadget for $m = 2$.

m-*Pump-Down Gadget.* Let us further construct what we call an *m-pump-down gadget.* For an odd, natural number $m \geq 1$, the m-pump-down gadget is a m-clique, in which blue has the majority of 1. Also, for an even, natural number $m \geq 2$, we construct the pump-down gadget for $m - 1$, and a disjoint red vertex. Observe that if an m-pump-down gadget is embedded in a network in which blue is the majority winner, then all m members of the structure are not under illusion. Moreover, if a blue vertex in the gadget would be adjacent to an additional red vertex, then it would be pushed into illusion. Figure 9 depicts a 4-pump-down gadget.

The following technical lemmas will help us decide what is the number m for which we are required to add either an m-pump-up or an m-pump-down gadget. The first of them is related to the pump-up gadget.

Lemma 11. *For every pair of natural numbers $m, k > 0$ and every rational number q in $(0, 1)$, such that $\frac{m}{k} < q$ there exists a polynomially-computable h, such that $\frac{m+h}{k+h+4} < q$, but $\frac{m+h+1}{k+h+4} \geq q$.*

Proof. Take any such k, m and $q = \frac{a}{b}$, such that $\frac{m}{k} < q$. We define a function $f_{m,k} : \mathbb{N} \rightarrow \mathbb{Q}$ such that for a natural number h, $f_{k,m}(h) = \frac{m+h}{k+h+4}$. First, observe that as $\frac{m}{k} < q$ it holds that $f_{m,k}(0) < q$. Moreover, observe that $f_{m,k}$ is strictly increasing, and that it is bounded by 1. Therefore, there exists an h, such that $f_{m,k}(h) < q$, while $f_{m,k}(h + 1) \geq q$. We call such a number $h^{\#}$. Note that we can efficiently compute such a number by first computing a real number h such that $\frac{a}{b} = \frac{m+h}{k+h+4}$ and then taking its floor.

Suppose now, towards a contradiction, that $\frac{m+h^{\#}+1}{k+h^{\#}+4} < \frac{a}{b}$. Then, we have that $b(m+h^{\#}+1) < a(k+h^{\#}+4)$, which is equivalent to $a(k + h^{\#}) + 4a > b(m + h^{\#}) + b$. We denote this inequality as α. Additionally, as $f_{m,k}(h^{\#} + 1) \geq q$, we know that $\frac{m+h^{\#}+1}{k+h^{\#}+5} \geq \frac{a}{b}$. So, $a(k + h^{\#} + 5) \leq b(m + h^{\#} + 1)$, and thus $-a(k + h^{\#} + 5) \geq -b(m + h^{\#} + 1)$. This is equivalent to $-a(k + h^{\#}) - 5a \geq -b(m + h^{\#} + 1)$. We denote this inequality as β. By adding α and β we get that $-a \geq 0$, so $a \leq 0$. But this is impossible, since $\frac{a}{b} > 0$.

□

We will further denote such a number as $h^{\#}_{m,k,q}$, or $h^{\#}$, if m, k and q are clear from the context. The following lemma will be relevant for the placement of a pump-down gadget.

Lemma 12. *For every rational number $q \in (0, 1)$ and $m, k \in \mathbb{N}$, such that $\frac{m}{k} \geq q$ there is a natural, polynomially-computable h, such that $\frac{m}{k+h} < q$, but $\frac{m+1}{k+h} \geq q$.*

Proof. Take any such m, k and $q = \frac{a}{b}$. We first define a function $g_{m,k} : \mathbb{N} \rightarrow \mathbb{Q}$, such that, for each natural number h, we have that $g_{k,m}(h) = \frac{m}{k+h}$. Observe that $g_{m,k}(0) = \frac{m}{k}$ and that $g_{m,k}$ is strictly decreasing, while it is bounded by 0. So, there exists a natural h, such that $g_{m,k}(h) < q$, but $g_{m,k}(h - 1) \geq q$, as $q > 0$. We will further

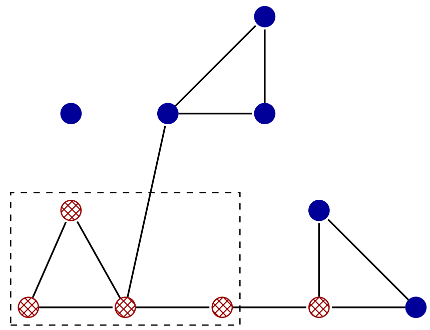

Fig. 10. Example of an encoding EN_G, for a graph G with four vertices, such that three of them form a clique, and one of them is a dependent of a member of this clique, and $k = 3$.

call such a number h^+. Note that we can efficiently compute such a number by first computing a real number h such that $\frac{a}{b} = \frac{m}{k+h}$ and then taking its floor.

Then, suppose towards a contradiction that $\frac{m+1}{k+h^+} < \frac{a}{b}$.Then, we have that $bm + b < ak + ah^+$, and so $-bm - b > -ak - ah^+$. We denote this inequality as α. Also, notice that by definition of h^+ we get that $\frac{m}{k+h^+-1} \geq \frac{a}{b}$. So, $bm \geq ak + ah^+ - a$. We denote this inequality as β. By adding α and β, we get that $-b \geq -a$, and so $a \geq b$ which is impossible, since $\frac{a}{b} < 1$.

□

We denote such a number as $h^+_{m,k,q}$, or h^+, if m, k and q are clear from the context.

We now construct, for a graph $G = (V_G, E_G)$, a labelled social network $EN_G = (N, E, f)$, which we call an *encoding* of G. Let us first describe the subnetwork of EN_G, which we call a G-*gadget*. For every vertex in V_G, we create a vertex in the G-gadget, which is labelled r, with the relation between them being identical to E_G. Further, for every vertex i in the G-gadget, we create a set of vertices labelled r, which we denote as r_i, and a set of vertices labelled b, which we call b_i. We require all members of r_i and of b_i to be adjacent to i. Furthermore, we set the cardinalities of r_i and b_i to be smallest, such that $|N(i)| + |r_i| - |b_i| = k - 1$. Further, for every vertex j in $r_i \cup b_i$, we create two vertices labelled b, adjacent to j, and to each other. Finally, we construct the minimum number of isolated vertices labelled b, satisfactory for b to be the strict majority label in EN_G. Observe now that the only vertices under illusion in this encoding are those in the G-gadget. Moreover, all of the members of this gadget are under illusion. Figure 10 depicts an example of EN_G.

We further call $|V_G| - k$ the *requirement*, or r_G. Also, we call $\binom{k}{2}$ the *budget*, or b_G. We say that network $EN'_G = (N, E', f)$ satisfies the requirement r_G and the budget if $|\{e \in N^2 : e \in E \text{ iff } e \notin E'\}| \leq b_G$, while at most r_G vertices are under illusion in EN'_G.

Lemma 13. *For every graph G, there is a network $EN'_G = (N, E', f)$, which satisfies the requirement and the budget if and only if there exists a k-clique in G.*

Proof. Take a graph $G = (V_G, E_G)$. First, suppose that there exists a k-clique in G. Then, take such a clique, and call the corresponding set of vertices in the G-gadget C. Observe that since, following previous observations, all of the vertices that are under illusion in the encoding $EN_G = (N, E, f)$ are in the G-gadget, it holds that the network (N, E', f), with $E' = E \setminus \{(i, j) \mid i, j \in C\}$, satisfies the budget, since $|\{(i, j) : i, j \in C\}| = \binom{k}{2}$. Observe that it also satisfies the requirement, as $|C| = k$, and we have that for every $i \in C$, it holds that the blue surplus in i's neighbourhood amounts to $|N_r^{V_G}(i)| + |r_i| - |b_i| - k - 1$, which by construction is equal to 0, and hence i is not under illusion.

Suppose now that there is no k-clique in G. Further, suppose towards a contradiction that there is a network $EN'_G = (N, E', f)$, which satisfies the requirement and the budget. Then, there is a set of vertices $S \subseteq V(C)$, with $|S| = k$, such that, for every $i \in S$, we have that illusion is eliminated from i in EN'_G. Further, by assumption, we have that S is not a clique. Notice, however, that then, as $b_G = \binom{k}{2}$, at least one member of S is under illusion in EN'_G, which contradicts the assumptions.

□

This observation allows us to demonstrate **NP**-completeness of q-Illusion Elimination.

Lemma 14. *q-Illusion Elimination is* **NP***-complete for every* $q \in (0, 1)$.

Proof. Consider any rational $q \in (0, 1)$. First, observe that q-Illusion Elimination is in **NP**, as verifying if a labelling induces a q-majority illusion is possible in polynomial time. Let us further construct a network E^q_G and a number m for graph G, such that the answer to q-Illusion Elimination for E^q_G and m is positive if and only if G contains a k-clique.

In the instance we consider, we will check whether we can find a subnetwork of E^q_G, in which connections between at most b_G pairs of vertices can be changed, and in which q-majority illusion does not hold. The first component of E^q_φ is E_G. If $\frac{|V(G)|-k}{V(E_G)} < q$, then we construct a l-pump up gadget for $l = h^{\#}_{|V(E_G)|-k,k,q}$. Otherwise, we construct a l'-pump down gadget, for $l' = h^{+}_{|V(E_G)|-k,k,q}$. Let us now show that the answer to q-Illusion Elimination for E^q_G and $\binom{k}{2}$ is positive if and only if G contains a k-clique.

First, suppose that G contains a k-clique. We will show that the answer to q-Illusion Elimination for E^q_G and k is positive. Let us first consider the case, in which $\frac{V(G)-k}{|V(E_G)}| < q$. Then observe that, as G is contains a k-clique, by Lemma 13 we have that it is possible to find a subnetwork E'_G of E_G, in which $\binom{k}{2}$ edges are altered, and where illusion was eliminated from k vertices. Then, by Lemma 11, we get that $\frac{|V(E_G)|-k+l}{|V(G)|-k+l+4} < q$. So we can construct a network of E^q_G, in which only $\binom{k}{2}$ edges are altered, but q-majority illusion does not hold. Similarly, if $\frac{|V(G)|-k}{|V(E_\varphi)|} \geq q$, we observe that, by Lemma 12, we get that $\frac{|V(E_G)|-k}{|V(E_G)|+l'} < q$. So, we get that we can eliminate illusion from k vertices in E_G by modifying $\binom{k}{2}$ edges. But then, we can construct a network E^q_G, in which only $\binom{k}{2}$ edges are removed, while q-majority illusion does not hold.

Suppose now that G does not contain a k-clique. We will show that the answer to the problem we consider for E^q_G and k is negative. Let us first consider the case in which $\frac{V(E_G)-k}{|V(E_G)|} < q$. Notice that, by reasoning in Lemma 13, we have that the minimum number of vertices from which illusion needs to be removed for q-majority illusion not to hold in E^q_G is k. Furthermore, let us notice that in the pump-up gadget, the minimum number of edges that is needed to be added to eliminate the illusion from a single vertex is greater than 4, which requires that we can only alter $\binom{k}{2}$ edges adjacent to the G-gadget to eliminate the illusion from k of its members, which is needed to eliminate q-majority illusion. Thus, we get from Lemma 13 that since G does not contain a k-clique, it is not possible to remove the illusion from at least k in E^q_φ by altering connections between at most $\binom{k}{2}$ pairs of vertices. The reasoning for the case in which $\frac{|V(E_G)|-k}{|V(E_G)|} \geq q$ is symmetric.

□

As a consequence, we get the following hardness result.

Theorem 5. *For all* $q \in (0, 1)$ *q-Illusion Elimination is* **W**[1]-hard parameterised by the number of altered edges.

Proof. Follows immediately from Lemma 14, noticing that the budget is bounded by a quadratic function of k.

□

4.2 NP-Completeness of Removal q-Illusion Elimination and Addition q-Illusion Elimination

Using reductions similar to the one provided above, we obtain **W**[1]-hardness of Removal q-Illusion Elimination and Addition q-Illusion Elimination. We note that while those problems are similar to q-Illusion Elimination, establishing their computational complexity does not follow immediately. To show hardness of Removal q-Illusion Elimination, by reduction from k-Clique, we use the same encoding, as in the proof of Theorem 5. We observe that, following the reasoning in the proof of Theorem 5, in this construction we get that for a graph G, $q \in (0,1)$, and $E^q_G = (N, E)$, there exists a network $E'^q_G = (N, E')$, such that q-majority illusion does not hold in E'^q_G, while $|\{e \in N^2 : e \in E \textit{ iff } e \notin E'\}| \leq b_G$ if and only if $|E'| - |E| \leq \binom{k}{2}$. The following result follows.

Lemma 15. *Removal q-Illusion Elimination is* **NP**-*complete for every* $q \in (0,1)$.

To show the hardness of Addition q-Illusion Elimination, we provide a reduction from k-Independent Set problem, similar to the previously shown construction. Notice that previous two reductions for illusion elimination problems involved removing edges between agents, whereas we now want to account for illusion elimination by adding edges, therefore we are going to employ a different reduction method.

We now construct, for a graph $G = (V_G, E_G)$, a labelled social network $EN_G = (N, E, f)$, which we call an *encoding* of G. Let us first describe the subnetwork of EN_G, which we call a G-*gadget*. For every vertex in V_G, we create a vertex in the G-gadget, which is labelled b, with the relation between them being identical to E_G. Further, for every vertex i in the G-gadget, we create a set of vertices labelled r, which we denote as r_i, and a set of vertices labelled b, which we call b_i. We also have that all members of r_i and of b_i are adjacent to i. We also denote the set of neighbours of i in the G-gadget as G_i. Further, we set the cardinalities of r_i and b_i to be smallest such that $|r_i| - |G_i| - |b_i| = k - 1$. Furthermore, for every vertex j in $r_i \cup b_i$, we create three vertices labelled b, adjacent to each other, and with one of them adjacent to j. Finally, we construct the minimum number of isolated vertices labelled b, satisfactory for b to be the strict majority label in EN_G. Observe now that the only vertices under illusion in this encoding are in the G-gadget. Moreover, all of the members of this gadget are under illusion. An example of an encoding EN_G is shown in Figure 11.

We further call $|V(G)| - k + 1$ the *requirement*, or r_G. Also, we call $\binom{k}{2}$ the *budget*, or b_G. Moreover, we say that a network $EN'_G = (N, E', f)$ satisfies the requirement r_G and the budget if $|E'| - |E| \leq b_G$, while less than r_G vertices are under illusion in EN'_G.

Lemma 16. *Addition q-Illusion Elimination is* **NP**-*complete for every* $q \in (0,1)$.

Proof. To show that the claim holds, we observe that, by reasoning symmetric to the proof of Lemma 14, for a given graph G, we can construct a network $EN'^q_G = (N, E', f)$ in which q-majority illusion does not hold, while $|E'| - |E| < \binom{k}{2}$ if and only if G contains a k-independent set. Then, the result follows by reasoning symmetric to the proof of Lemma 14.

□

By combining Lemma 15 and Lemma 16, and noticing that in both cases the budget is bounded by a quadratic function of k, we obtain the following theorem.

Theorem 6. *For all* $q \in (0,1)$, *Addition q-Illusion Elimination and Removal q-Illusion Elimination are* **W**[1]-hard.

5 Discussion

Our work addresses the problem of majority illusion from the computational complexity perspective. While previous work on this topic (see [36]) was mainly focused on empirical exploration of this phenomenon, our results concern the computational limitations of both detecting the possibility of majority illusion and removing

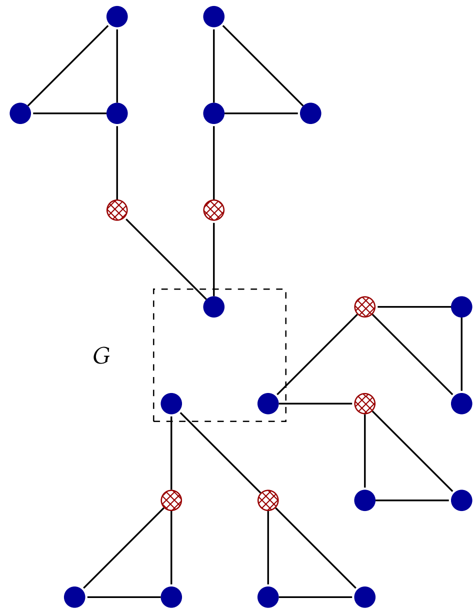

Fig. 11. An encoding EN_G, for a graph consisting of the independent set of three vertices.

it. We see our results as providing insights to a designer of a social network system aiming at reducing the impact of agents' misperception of the social system they belong to, specifically when it comes to forming beliefs about what everyone else thinks.

We note that several recent papers have, directly or indirectly, built upon our work. From the computational point of view, some edge cases and variants are being addressed. For instance, [18] have shown tractability of q-Illusion Elimination for $q = 0$, while [21] considered eliminating majority illusion by relabelling vertices and [42] studied the occurrence of majority illusion in restricted classes of graphs. Other papers have noted the importance of our contribution in the context of related problems. For instance, [37] notice the importance of majority illusion in distorting people's local perception in the context of *voting*. Also, [6] observe the impact of majority illusion in the context of *exposure bias*.

Our results have implications for the study of *voting*. Generally speaking, voters can be persuaded to change their opinion if a large number of people they communicate with through a social network disagree with them. Furthermore, the perceived distribution of opinions regarding candidates is a crucial piece of information for a strategic voter. Hence, taking majority illusion into account is important towards the analysis of voters' behaviour.

While our analysis is limited to two opinions only, impacting therefore only two-candidate elections, further extension of our work is needed when elections have larger numbers of participants. In what follows we look at a natural definition of "plurality" illusion, where a label can be viewed as an agent's favourite candidate. Surprisingly, we show that this is not reducible to majority illusion. Afterwards, we look at the connection between majority illusion and opinion diffusion, reflecting on the consequences of this phenomenon when decision-making takes place in a social network.

5.1 Plurality Illusion

Having studied majority illusion for networks labelled using two labels, it is natural to ask whether similar results hold for networks labelled using more labels. Towards this end we define *plurality illusion*, which happens

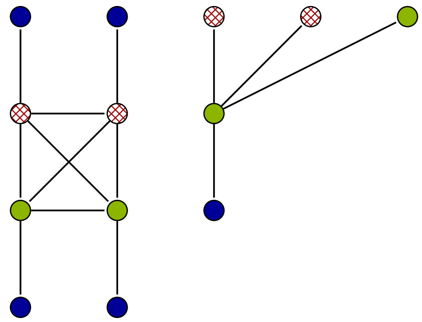

Fig. 12. Example of a social network admitting a 1-plurality illusion with three labels, but not admitting a 1-majority illusion.

when (all) agents see a label which is not the plurality winner in the network as the plurality winner in their neighbourhood. We now show that there are networks that admit plurality illusion for all agents, but not a 1-majority illusion. This observation motivates the fact that plurality illusion is interesting in its own right and deserves further investigation.

Formally, define *plurality illusion* as follows. Let C be a finite set of labels. Given a labelled social network SN=(N, E, f), where $f : N \to C$ is a labelling, we denote the set of *most popular labels* in SN as Pl_{SN}. So, $Pl_{SN} = \arg\max_{c \in C} |\{i \in N : f(i) = c\}|$. If the most popular label is unique, we will call it the *plurality winner*. Similarly, for an agent $i \in N$, we say that Pl^i_{SN} is the set of most popular options in i's neighbourhood. Formally, $Pl^i_{SN} = \arg\max_{c \in C} |\{i \in N(i) : f(i) = c\}|$. If $Pl^i_{SN} = \{c\}$, for some $c \in C$, we say that c is the plurality winner in i's neighbourhood. Then, we say that an agent $i \in N$ is under plurality illusion if the plurality winner in i's neighbourhood is different from the plurality winner (provided both exist). Further, we say that f *induces plurality illusion* if all agents in N are under plurality illusion in (N, E, f). Also, we say that (N, E) *admits* plurality illusion if some labelling $f : N \to C$ induces plurality illusion.

Observation 4. *There are networks which admit a plurality illusion, but not 1-majority illusion.*

The following example shows that Observation 4 holds.

Example 3. *Consider the social network shown in Figure 12. This network admits a plurality illusion with three labels, as per labelling in Figure 12. Notice that in this case five vertices are labelled blue, four are labelled red, and four are labelled green. Thus, blue is the plurality winner. However, one can verify that there is a plurality winner other than blue in the neighbourhood of every vertex in the network.*

The social network in Figure 12 does not however admit 1-majority illusion. Suppose towards a contradiction that there is a labelling f of this network that induces a 1-majority illusion. Then, observe that there are thirteen vertices in the network, and hence at most six vertices are labelled red in f. Moreover, all vertices in the clique in the left subnetwork, as well as the central vertex in the right subnetwork, have dependents. Hence, they are labelled red in f. Furthermore, the central vertex in the right subnetwork has four neighbours, and hence, by assumption that it is under illusion, at least three of the vertices adjacent to it are labelled red in f. But then, at least eight vertices are labelled red in f, which contradicts the assumptions. So, the network in Figure 12 does not admit a 1-majority illusion.

5.2 Opinion Diffusion

As argued, majority illusion can affect the way in which agents change their mind regarding a specific issue, that is, a well-placed minority can gain an advantage by communicating their opinion. This is especially visible in

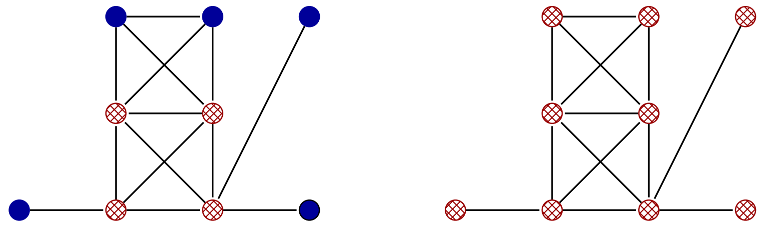

Fig. 13. An instance in which 1-majority illusion holds and the minority view becomes a consensus after one opinion diffusion step.

the context of the threshold-based opinion diffusion model (see [26]), where each agent modifies their opinion looking at a set threshold of their incoming connections. For example, they all come to think what the majority of their influencers say. In this model 1-majority illusion means that the minority view becomes unanimous consensus at the next step.

Let us look at the case of majority thresholds with synchronous updates, widely studied in the distributed AI literature [32, 10, 13]. Formally, for a labelled social network (N, E, f) and an agent i, we denote as $A(i) = \{j \in N(i) : f(i) = f(j)\}$ the set of individuals in i's neighbourhood who agree with them. Similarly, let $D(i) = N(i) \setminus A(i)$ be the set of those who disagree with i.

We can now define how agents change their label following their neighbours' views. Let $SN = (N, E, f)$ be a labelled social network, and let $i \in N$ be an agent. Then, the *opinion diffusion step* is the function $OD : N \longrightarrow \{b, r\}$, such that $OD(SN, i) = \mathit{flip}(f(i))$, if $|D(i)| > |A(i)|$, and $OD(SN, i) = f(i)$ otherwise, where $\mathit{flip}(k) = \{b, r\} \setminus f(k)$ denotes the change from an original label to its opposite value. Using the notion of opinion change we can define the *synchronous update* protocol. There, $SU(SN) = (N, E, f')$, where for every $i \in N$ we have that $f'(i) = OD(SN, i)$.

We can observe how, in networks in which 1-majority illusion holds, a minority view becomes a consensus just in one opinion diffusion step. Figure 13 exemplifies this phenomenon. Similarly, we can observe that in networks in which q-majority illusion holds, with $q > \frac{1}{2}$, the minority view is adopted by the majority of agents immediately under the protocol defined above. Let us emphasise that this observation showcases the importance of majority illusion with respect to existing work on consensus development through opinion diffusion as in, for example, [4, 5]. In fact, it constitutes a compelling case of instances in which the consensus (or a strong majority) is achieved immediately.

We conclude by observing how majority illusion might impact the results of elections when voters are connected by a social network. To see that, let us take the opinion diffusion protocol defined above, as well as an electoral competition between two candidates. Following previous observations we see that if q-majority illusion holds in a network, with $q > \frac{1}{2}$, then the losing candidate becomes a winner even if the voters are allowed to communicate only once. Further connections between majority or plurality illusion and voting, especially regarding changing the structure of a network, remain a compelling research avenue.

6 Conclusion and Open Problems

In this paper we provided a computational analysis of majority illusion in a social network, focusing on detecting its occurrence and transforming the network structure to see to its elimination. Here we summarise our contributions.

Summary of Contributions. We showed the algorithmic hardness of verifying (Theorems 1 and 2) and eliminating (Theorems 5 and 6) q-majority illusion. Moreover, we provided a number of parameterised algorithms for the verification problem (see Table 1). Also, we demonstrated $\mathbf{W}[1]$-hardness for the elimination of majority illusion

(Theorems 5 and 6). Informally, we have shown that even if illusion identification is a hard problem in general, there are various natural constraints that make it feasible.

Future Research. Our research in this paper leaves a vast number of avenues for further study. We identify a few potential problems and directions to accompany what we laid out in our discussion section.

Extension of Computational Complexity Results.

- Our results on illusion elimination did not show natural parameters which would make the problem easy to resolve. It thus remains open whether we can find another good parametrisation.
- Further, in this paper we have only shown the hardness of q-MAJORITY ILLUSION for $q > \frac{1}{2}$. Although we conjecture that the problem remains **NP**-complete for smaller values of q, we leave establishing the complexity of q-MAJORITY ILLUSIONfor such q as an avenue for further research.
- From a complexity-theoretic point of view, it remains open to check the existence of a single-exponential time algorithm with respect to neighbourhood diversity.
- We note that, in addition to our parameterised complexity results, designing an efficient algorithm that approximates the maximal value of q in a given instance of q-MAJORITY ILLUSION would be a natural direction for further study.
- While the results shown in this paper establish computational complexity of problems related to majority illusion, identifying networks for which they are tractable, or where illusion is not possible, is also left as an avenue for further study. Some of such cases might include grids, possibly interpreted as housing neighbourhoods. Another interesting restricted case is the one of homophilic agents, that is, of labellings in which individuals with the same label tend to be grouped together.

Model Extensions.

- An open challenge is to explore the setting in which the assumption of binary labelling is lifted. As we have shown in Section 5.1, surprisingly, there are social networks that do not admit a majority illusion but do admit a "plurality" illusion, that is, agents have a wrong perception of the plurality winner when more than two labels are allowed. This is particularly relevant for voting contexts such as elections with multiple candidates.
- As pointed out from the discussion around Figure 13, exploring the connections between majority illusion and opinion diffusion is a natural and important follow up. One can observe that in a labelling, which induces 1-majority illusion, all agents adopt the minority opinion after just one opinion diffusion step, given the synchronous, majority-based threshold protocol. This observation motivates further connections between the majority illusion and the spread of opinions.

It is fair to say that majority illusion may not always be an undesirable phenomenon. There are countless situations where positive change can be induced by misperception of what the others think [41]. Here we have taken the perspective of a system designer that does not what to use any form of distortion to achieve socially desirable objectives [9]. In other words, we have endorsed the point of view that misperception is an undesirable social property in itself.

Acknowledgments

This work significantly extends our previous AAAI'23 manuscript [25], incorporating the full proofs of all the results, accompanied by the needed illustrations. We also provide all the technical background needed to understand the constructions and reasoning used in the proofs. We also provide a broad discussion of potential generalisations of the problem and their impact in related research questions such as voting and the convergence of opinion dynamics.

This work was supported by the Engineering and Physical Sciences Research Council (grant numbers EP/V007793/1 and EP/V044621/1). Umberto Grandi acknowledges the support of the ANR JCJC project SCONE (ANR 18-CE23-0009-01). Paolo Turrini acknowledges the support of the Leverhulme Trust for the Research Project Grant RPG-2023-050 and the EU TAILOR network for the Connectivity Fund 29/2023. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 101002854). During work on this paper Grzegorz Lisowski and Lawqueen Kanesh were affiliated with the University of Warwick.

References

[1] N. Alipourfard, B. Nettasinghe, A. Abeliuk, V. Krishnamurthy, and K. Lerman. 2020. Friendship paradox biases perceptions in directed networks. *Nature communications*, 11, 1, 1–9.

[2] M. Amoruso, D. Anello, V. Auletta, R. Cerulli, D. Ferraioli, and A. Raiconi. 2020. Contrasting the spread of misinformation in online social networks. *Journal of Artificial Intelligence Research*, 69, 847–879.

[3] S. Asch. 1951. Effects of group pressure on the modification and distortion of judgments. In *Groups, Leadership and Men: Research in Human Relations*. Carnegie Press, 177–190.

[4] V. Auletta, I. Caragiannis, D. Ferraioli, C. Galdi, and G. Persiano. 2015. Minority becomes majority in social networks. In *International conference on web and internet economics*. Springer, 74–88.

[5] V. Auletta, D. Ferraioli, and G. Greco. 2020. On the complexity of reasoning about opinion diffusion under majority dynamics. *Artificial Intelligence*, 284, 103–288.

[6] N. Bartley, K. Burghardt, and K. Lerman. 2024. Evaluating content exposure bias in social networks. In *Proceedings of the 2023 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (ASONAM '23). Association for Computing Machinery, Kusadasi, Turkiye, 379–383. ISBN: 9798400704093. DOI: 10.1145/3625007.3627724.

[7] R. Becker, G. D'angelo, S. Ghobadi, and H. Gilbert. 2022. Fairness in influence maximization through randomization. *Journal of Artificial Intelligence Research*, 73, 1251–1283.

[8] P. Berman, M. Karpinski, and A. Scott. 2004. Approximation hardness of short symmetric instances of MAX-3SAT. Tech. rep. Weizmann Institute of Science.

[9] T. Grüne-Yanoff and S. O. Hansson, (Eds.) 2009. *The ethics of nudge. Preference Change: Approaches from Philosophy, Economics and Psychology*. Springer Netherlands, Dordrecht, 207–219. ISBN: 978-90-481-2593-7. DOI: 10.1007/978-90-481-2593-7_10.

[10] R. Bredereck and E. Elkind. 2017. Manipulating opinion diffusion in social networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*.

[11] M. Castiglioni, D. Ferraioli, and N. Gatti. 2020. Election control in social networks via edge addition or removal. In *The 34th AAAI Conference on Artificial Intelligence (AAAI)*.

[12] M. Castiglioni, D. Ferraioli, N. Gatti, and G. Landriani. 2021. Election manipulation on social networks: seeding, edge removal, edge addition. *Journal of Artificial Intelligence Research*, 71, 1049–1090.

[13] D. Chistikov, G. Lisowski, M. Paterson, and P. Turrini. 2020. Convergence of opinion diffusion is pspace-complete. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 7103–7110.

[14] R. Chitnis and N. Talmon. 2018. Can we create large k-cores by adding few edges? In *Computer Science – Theory and Applications*. F. V. Fomin and V. V. Podolskii, (Eds.) Springer International Publishing, Cham, 78–89.

[15] F. Corò, E. Cruciani, G. D'Angelo, and S. Ponziani. 2019. Vote for me!: election control via social influence in arbitrary scoring rule voting systems. In *International Conference on Autonomous Agents and Multi Agent Systems, (AAMAS)*, 1895–1897.

[16] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. 2015. *Parameterized Algorithms*. Springer.

[17] R. Davis, J. C. Baumgartner, P. L. Francia, and J. S. Morris. 2009. The internet in us election campaigns. *Routledge handbook of Internet politics*, 1, 13–25.

[18] J. Dippel, M. D. la Tour, A. Niu, S. Roy, and A. Vetta. 2025. Eliminating majority illusion is easy. In *Proceedings of the AAAI Conference on Artificial Intelligence* number 13. Vol. 39, 13763–13770.

[19] J. A. Doucette, A. Tsang, H. Hosseini, K. Larson, and R. Cohen. 2019. Inferring true voting outcomes in homophilic social networks. *Autonomous Agents and Multi-Agent Systems*, 33, 3, 298–329.

[20] P. Faliszewski, R. Gonen, M. Koutecký, and N. Talmon. 2018. Opinion diffusion and campaigning on society graphs. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*.

[21] F. Fioravantes, A. Lahiri, A. Lauerbach, M. D. Sieper, and S. Wolf. Eliminating Majority Illusion. working paper or preprint, (May 2024). https://hal.science/hal-04574647.

[22] J. Fox, T. Roughgarden, C. Seshadhri, F. Wei, and N. Wein. 2020. Finding cliques in social networks: A new distribution-free model. *SIAM J. Comput.*, 49, 2, 448–464.

[23] A. Frank and É. Tardos. 1987. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7, 1, 49–65.

[24] M. Gilboa and N. Nisan. 2022. Complexity of public goods games on graphs. In *International Symposium on Algorithmic Game Theory*. Springer, 151–168.

[25] U. Grandi, L. Kanesh, G. Lisowski, R. Sridharan, and P. Turrini. 2023. Identifying and eliminating majority illusion in social networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37, 5062–5069.

[26] M. Granovetter. 1978. Threshold models of collective behavior. *American journal of sociology*, 83, 6, 1420–1443.

[27] F. Gurski and E. Wanke. 2000. The tree-width of clique-width bounded graphs without $K_{n,n}$. In *26th International Workshop in Graph-Theoretic Concepts in Computer Science* (Lecture Notes in Computer Science). Vol. 1928. Springer, 196–205.

[28] N. O. Hodas, F. Kooti, and K. Lerman. 2013. Friendship paradox redux: your friends are more interesting than you. In *Seventh International AAAI Conference on Weblogs and Social Media*.

[29] H. W. L. Jr. 1983. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8, 4, 538–548.

[30] R. Kannan. 1987. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12, 3, 415–440.

[31] B. Kapron, G. Srivastava, and S. Venkatesh. 2011. Social network anonymization via edge addition. In *2011 International Conference on Advances in Social Networks Analysis and Mining*. IEEE.

[32] D. Kempe, J. M. Kleinberg, and É. Tardos. 2015. Maximizing the spread of influence through a social network. *Theory of Computing*, 11, 105–147.

[33] T. Koana, C. Komusiewicz, and F. Sommer. 2022. Exploiting c-closure in kernelization algorithms for graph problems. *SIAM Journal on Discrete Mathematics*, 36, 4, 2798–2821.

[34] T. Korhonen. 2021. A single-exponential time 2-approximation algorithm for treewidth. In *Symposium on Foundations of Computer Science (FOCS)*, 184–192.

[35] M. Lampis. 2012. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64, 1, 19–37.

[36] K. Lerman, X. Yan, and X.-Z. Wu. 2016. The "majority illusion" in social networks. *PloS one*, 11, 2.

[37] X. Liu, F. Ren, G. Su, M. Zhang, W. Gu, and S. Kato. 2024. Predicting voting outcomes for multi-alternative elections in social networks. *IEEE Access*, 12, 98960–98970. DOI: 10.1109/ACCESS.2024.3425160.

[38] S. Ordyniak and S. Szeider. 2013. Parameterized complexity results for exact bayesian network structure learning. *J. Artif. Intell. Res.*, 46, 263–302.

[39] F. P. Santos, S. A. Levin, and V. V. Vasconcelos. 2021. Biased perceptions explain collective action deadlocks and suggest new mechanisms to prompt cooperation. *Iscience*, 24, 4, 102375.

[40] A. J. Stewart, M. Mosleh, M. Diakonova, A. A. Arechar, D. G. Rand, and J. B. Plotkin. 2019. Information gerrymandering and undemocratic decisions. *Nature*, 573, 7772, 117–121.

[41] R. H. Thaler and C. R. Sunstein. 2008. *Nudge:Nudge: Improving decisions about health, wealth, and happiness*. John Wiley and Sons Ltd.

[42] M. Venema-Los, Z. Christoff, and D. Grossi. 2023. On the graph theory of majority illusions. In *European Conference on Multi-Agent Systems (EUMAS)*.

[43] B. Wilder and Y. Vorobeychik. 2018. Controlling elections through social influence. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

[44] 2021. *Maximizing influence of leaders in social networks. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, New York, NY, USA, 2400–2408.

[45] Z. Zhou, F. Zhang, X. Lin, W. Zhang, and C. Chen. 2019. K-core maximization: an edge addition approach. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*.

Received December 16, 2024; revised June 18, 2025; accepted June 25, 2025