Identifying and Eliminating Majority Illusion in Social Networks

Umberto Grandi[1], Lawqueen Kanesh[2], Grzegorz Lisowski[3], Ramanujan Sridharan[3], Paolo Turrini[3]

[1] University of Toulouse, France
[2] IIT Jodhpur, India
[3] University of Warwick, UK
umberto.grandi@irit.fr, lawqueen@itj.ac.in, {grzegorz.lisowski, r.maadapuzhi-sridharan,p.turrini}@warwick.ac.uk

Abstract

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, i.e., 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 undesirable phenomenon. In this paper we initiate the computational study of majority illusion in social networks, providing NP-hardness and parametrised complexity results for its occurrence and elimination.

1 Introduction

Social networks shape the way people think. Individuals' private opinions can change as a result of social influence and a well-placed minority view can become what most people come to believe (Stewart et al. 2019). The COVID-19 vaccination debate has brought to the fore the dramatic effects that misperception can have (Johnson et al. 2020), highlighting the importance of social networks where participants receive the most unbiased information possible.

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*. Majority illusion was originally introduced by Lerman, Yan, and Wu (2016), 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, i.e., 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, e.g., they follow a threshold model (Granovetter 1978), 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 see to it that this undesirable phenomenon is eliminated.

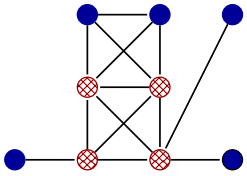

Figure 1: An instance of majority illusion. The well-placed red (shaded) minority is perceived as majority by everyone.

The graph structure of majority illusion was analysed by Lerman, Yan, and Wu (2016), who studied network features that correlate with having many individuals under illusion. They demonstrated how disassortative networks, i.e. 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.

Likewise, the approach of eliminating undesirable properties by network transformation is not new, and extensively pursued in the context of election manipulation (see, e.g., Castiglioni et al. (2021)), influence maximisation (Zhou and Zhang 2021), anonymisation (see, e.g., Kapron, Srivastava, and Venkatesh (2011)) and of k-core maximisation (see, e.g., Chitnis and Talmon (2018) and Zhou et al. (2019)). 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 the possibility of illusion, i.e., deciding whether there is a labelling of the vertices such that a set majoritarian fraction of agents are under illusion, and we prove it to be NP-complete. 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 parametrised complexity analysis involving well-established graph width measures and their variants. In par-

ticular, we obtain a fixed-parameter algorithm (FPT algorithm) for this problem parametrised by the maximum degree of the network plus its tree width, as well as by the size of the 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 parametrised by the tree width). These two results are of specific interest to sparse networks. We then also consider dense networks by parameterising by the neighborhood diversity of the input network and obtain an FPT algorithm. Finally we move to the problem of eliminating illusion, 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 parametrised by arguably the most natural parameter, i.e., the number of modified edges.

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, Ferraioli, and Greco (2020), who identify networks and initial distributions of opinions such that an opinion can become a consensus 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. However, checking for the possibility of majority illusion does not correspond to checking whether a minority colour can be adopted by the majority of agents, as studied by Auletta et al. (2015, 2017). Note also that the minority colour can be adopted by the majority in one step even if only a small fraction of agents is under majority illusion.

Other notable models include the work of Doucette et al. (2019) 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 (Kempe, Kleinberg, and Tardos 2015; Bredereck and Elkind 2017; Auletta, Ferraioli, and Greco 2020; Castiglioni, Ferraioli, and Gatti 2020).

Network Manipulation. An important research line has looked at how to transform a social network structure with applications in the voting domain. Wilder and Vorobeychik (2018), e.g., 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. (2018) 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, e.g. Hodas, Kooti, and Lerman (2013); Alipourfard et al. (2020)). Exploiting a similar paradox, Santos, Levin, and Vasconcelos (2021) recently showed how false consensus leads to the lack of participation in team efforts. Parametrised complexity of problems related to social networks is an established research direction (see, e.g., Bredereck and Elkind (2017)).

Paper Structure. Section 2 provides the basic setup and definitions. Section 3 focuses on checking if illusion can occur in a network while Section 4 studies illusion elimination. Section 5 concludes the paper presenting potential future directions. In the interest of space some proofs are omitted and can be found in the extended version of this paper[1].

2 Preliminaries

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*, requiring E to be symmetric. Furthermore, we assume that E is *irreflexive*, i.e., 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 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 colours $\{b, r\}$ (*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 $f : N \to \{b, r\}$ is a* labelling *which assigns an alternative to each agent.*

In a graph 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. For a vertex set X and a labelling $f : X \to \{b, r\}$, 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 $N^X_{f,r}(i)$. The analogue of this definition for blue neighbourhood is symmetric. 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 drop the explicit reference to X or f in this notation if clear from the context.

Majority Illusion. A colour $k \in \{b, r\}$ is a strict majority winner in a labelled social network (N, E, f) if there are strictly more vertices coloured with k than with k' such that $k' = \{b, r\} \setminus k$. We use $W_{(N,E,f)}$ to denote such a winner if it exists. Similarly, a colour k is a strict majority winner in i's neighbourhood if i's k surplus is strictly positive. We use $W^i_{(N,E,f)}$ to denote such a winner if it exists. We say that an agent $i \in N$ is *under illusion* if they have a wrong perception of the majority winner. So, for agent i to be under illusion in a (N, E, 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 and labelling $f : N \to \{b, r\}$, we*

[1] shorturl.at/aisxB

say that f induces *a 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) which induces a q-majority illusion, then we say that (N, E) *admits* a q-majority illusion. Henceforth, we assume that the majority colour is blue, whenever one exists.

Parametrised 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 for a parameter k, if there exists an algorithm solving this problem running in time $|I|^{f(k)}$ (called an XP-algorithm), where f is some computable function. Note that FPT $\subseteq$ 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 parametrised by r, if there is a many-one reduction to it from the classic k-CLIQUE problem (Cygan et al. 2015) in time $f(k) \cdot |I|^{O(1)}$ (where I is the instance of k-CLIQUE), with $r \leq g(k)$ some computable function g.

Tree Decomposition. Tree width is a fundamental graph parameter, useful for the design of parametrised algorithms, which is crucial 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 parametrised 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: (i) $\bigcup_{t \in V(T)} \beta(t) = V(G)$, (ii) for every edge $e \in E(G)$, there exists a node $t \in V(T)$ such that both endpoints of e belong to $\beta(t)$, and (iii) 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 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 $t \in V(T)$, by V_t, we denote the set $\bigcup_{t' \preceq t} \beta(t')$, i.e., 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 helpful to work with 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 (Cygan et al. 2015).

Further Graph Parameters. Another graph parameter we consider is the *neighbourhood diversity* (Lampis 2012) which captures the number of "twin classes" in the graph. We say vertices u and v are *twins* if they have the same neighbours, i.e. $N(u) \setminus \{v\} = N(v) \setminus \{u\}$.

Definition 5. *The* neighbourhood diversity *(ND) of a graph G (which we also denote as $nd(G)$), is the minimum w such that $V(G)$ can be partitioned into w sets of twin vertices. Each set of twins, called a* module*, is either a clique or an independent set. 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, i.e., the *c-closure* (Fox et al. 2018, 2020; Koana, Komusiewicz, and Sommer 2022). 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 are adjacent. This concept was introduced 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 agents in social network have a friend in common, they are likely to be friends. Fox et al. (Fox et al. 2020, Table A.1), and later Koana et al. (Koana, Komusiewicz, and Sommer 2022, Table 1), showed that several social and biological networks are c-closed for small values of c.

3 Verifying the Possibility of Illusion

We are interested in the problem of checking, for a specific q, if a given network admits q-majority illusion. Formally:

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

3.1 Hardness

We now prove that q-MAJORITY ILLUSION is NP-hard for every rational $q \in (\frac{1}{2}, 1]$, by 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, and twice in the negative form, i.e., it is in 2P2N-3-CNF (see (Berman, Karpinski, and Scott 2004)). In our reduction, which can be found in the extended version of this paper, we construct an *encoding* of a formula φ in 2P2N-3-CNF, which is a social network admitting 1-majority illusion iff φ is satisfiable.

To this end, we construct what we call a *variable gadget*, for every variable p_i in φ. Within these structures we distinguish two *literal vertices*, corresponding to p_i and to $\neg p_i$ respectively. Also, for every clause C_j in φ, we create what we call a *clause gadget*. There, one of the vertices is adjacent to all vertices corresponding to the literals in C_j. We also use what we call a *balance gadget*, a structure whose vertices are labelled r, if a labelling induces a 1-majority illusion in the encoding of φ. We obtain that a labelling of the encoding which we construct induces 1-majority illusion only if exactly one of the literal vertices in each variable gadgets is labelled r, and a vertex in every clause gadget is adjacent to some literal vertex labelled r. It follows that the encoding of φ admits 1-majority illusion if and only if φ is satisfiable.

We then obtain NP-hardness of q-MAJORITY ILLUSION for every rational $q \in (\frac{1}{2}, 1]$ using the fact that, for every such q, we can extend the encoding E_φ of φ by a number of pairs of vertices such that a labelling of this extended network induces q-majority illusion if and only if all of the vertices in E_φ, and half of the vertices in the additional pairs, are under illusion. We also obtain that the problem is NP-hard for bipartite networks by ensuring that all of the components of the construction are bipartite. We defer the full proof to the extended version of this paper.

Theorem 1. *q-MAJORITY ILLUSION is NP-complete for every rational q in $(\frac{1}{2}, 1]$, even on bipartite graphs.*

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.*

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 a constant.

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.*

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.

We further show that the q-MAJORITY ILLUSION problem is NP-complete also for *planar* networks. This result rules out using the genus of the graph or other generalisations of planarity as possible structural restrictions to get polynomial-time algorithms. We show 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 prove this result is similar to the one used in the proof of Theorem 1. However, it is non-trivial to design appropriate planar gadgets to ensure that the reduced instance is planar. The proof can be found in the extended version of this paper.

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

Then, from the fact that networks with clique size greater than 4 are not planar, we get the following observation.

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 $q \in (\frac{1}{2}, 1]$.*

3.2 Parametrised Complexity

The NP-completeness results for q-MAJORITY ILLUSION motivate the study of this problem from the perspective of parametrised complexity, aiming at identifying restrictions on the input that allow for tractability. Note that our result that q-MAJORITY ILLUSION is NP-hard on networks of constant max-degree implies that, unless P=NP, q-MAJORITY ILLUSION does not even have an algorithm with running time $n^{f(\Delta)}$ for any computable function f, where Δ is the max-degree. So, q-MAJORITY ILLUSION is Para-NP-hard parametrised by Δ. Hence, we must extend this parameterisation using other structural properties of the graph. Our first FPT result (Theorem 3) states that there exists a FPT algorithm for q-MAJORITY ILLUSION parametrised by the max-degree *and* tree width of the input network.

We next sketch this algorithm. Suppose that the input network has tree width k and we have a nice tree decomposition (T, β) for it of width at most $w = O(k)$. This can be computed using known results in time $2^{O(k)}n^{O(1)}$. We define a boolean function H whose domain is the set of all tuples where each tuple comprises a node $t \in V(T)$, a labelling $\mathsf{col} : \beta(t) \to \{\mathsf{red}, \mathsf{blue}\}$ of vertices in the bag $\beta(t)$, a function $\mathsf{esurp} : V_t \to \{-\Delta, \ldots, \Delta\}$ where $\mathsf{esurp}(v) = 0$ for all $v \notin \beta(t)$, a function $\mathsf{isurp} : \beta(t) \to \{-\Delta, \ldots, \Delta\}$, some $\alpha \in [n]$, and some $\ell_r \in [n]$. If $\beta(t) = \emptyset$, then $\mathsf{col}, \mathsf{esurp}, \mathsf{isurp} = \emptyset$. We define $H(t, \mathsf{col}, \mathsf{esurp}, \mathsf{isurp}, \alpha, \ell_r) = 1$ if and only if there is a labelling $\rho : V_t \to \{\mathsf{red}, \mathsf{blue}\}$ such that the following hold:

1. for every $i \in \beta(t)$, $\rho(i) = \mathsf{col}(i)$,
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 $\{i \in V_t : |N_{\rho,r}^{V_t}(v)| > |N_{\rho,b}^{V_t}(i)| + \mathsf{esurp}(i)\}$,
4. for every $i \in \beta(t)$, $\mathsf{isurp}(i) = |N_{\rho,b}^{V_t}(i)| - |N_{\rho,r}^{V_t}(i)|$ is the internal blue surplus of every vertex in $\beta(t)$ under ρ.

The intuition behind the description of the function is the following. Take a hypothetical labelling γ for the (N, E) that witnesses that (N, E) admits q-majority illusion, fix a bag $\beta(t)$ and let δ be the restriction of γ to the set V_t. Then,

- col is the restriction of δ to the vertices of the bag $\beta(t)$.
- The function esurp (read *external surplus*) describes the blue surplus for the vertices in V_t that is provided by the vertices *outside* the set V_t. Note that only vertices of the bag $\beta(t)$ get non-zero blue surplus from outside V_t (since only these vertices have any neighbours at all outside V_t by the definition of a tree decomposition) hence, we may assume a value of 0 "external" blue surplus for all vertices 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.
- The value of ℓ_r is simply the number of vertices of V_t that are assigned red by γ, and hence also by δ.
- The number α is the number of vertices of V_t that are under illusion under γ. 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 up the blue surplus given by vertices in V_t (which can be deduced from col) and the blue surplus from outside V_t (which is given by the function esurp), we get at most -1.
- The function isurp (i.e., *internal surplus*) describes the blue surplus for the vertices in $\beta(t)$ provided by the vertices *within* V_t. As for esurp, since the max-degree is Δ, the range of the function lies in $\{-\Delta, \ldots, \Delta\}$.

The crux of our correctness is that, if we could find a labelling ρ for V_t that is not necessarily δ, but has the same "signature" of δ in terms of $\mathsf{col}, \ell_r, \alpha, \mathsf{isurp}$ given the same esurp, then we can "cut" δ from γ and replace it with ρ to get another labelling of G with exactly the same number of vertices under illusion as γ. This gives us the so-called optimal substructure property, crucial for dynamic programming.

Note that there are $2^{w+1} \cdot (2\Delta + 1)^{2(w+1)} n^{O(1)} = \Delta^{O(w)} n^{O(1)}$ possible tuples. This is because each bag contains at most $w + 1$ vertices, implying at most 2^{w+1} possibilities for col at any bag. Since, for any bag, esurp can only have at most $2\Delta + 1$ possible values for any vertex in a bag, we get that there are at most $(2\Delta + 1)^{w+1}$ possibilities for esurp at any bag. The same bound extends to isurp. The remaining elements of the tuple, α and ℓ_r, are bounded by n. So, 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 G admits q-majority illusion by examining the table entries corresponding to the root bag $\beta(t^\star)$, which by the definition of a nice tree decomposition, is empty. Then, we have that (N, E) admits q-majority illusion if and only if there exists values $\ell_r \in [n]$ and $\alpha \in [n]$ such that $\alpha \geq \lceil qn \rceil$, $l_r < \frac{n}{2}$, and $H(\emptyset, \emptyset, \emptyset, \alpha, \ell_r) = 1$.

Our algorithm relies on the fact that we can compute the table entries at each bag assuming that all the table entries at all descendant bags have been computed correctly (the base case is leaf bags, which are by definition empty). The details of this procedure can be found in the extended version of this paper. We note that time taken to fill the entries for any one bag is bounded by $\Delta^{O(w)} n^{O(1)}$ and we have already argued that there is a total of $\Delta^{O(w)} n^{O(1)}$ possible tuples corresponding to each bag. The stated running time follows.

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.

We next discuss some immediate implications of the above result. First of all, notice that $\Delta \leq n$. Hence, our FPT algorithm parametrised by Δ and the tree width is in fact an XP algorithm parametrised by the tree width alone.

Corollary 1. q-MAJORITY ILLUSION *can be solved in time* $n^{O(k)}$ *on networks of tree width* k.

Then, consider the following relation between tree width and the parameter cliquewidth (Gurski and Wanke 2000) (denoted by $cw(G)$) on bounded-degree graphs.

Proposition 1. *(Gurski and Wanke 2000) 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* $tw(G) \leq 3k(d-1) - 1$.

Since graphs with max-degree Δ exclude $K_{\Delta+1,\Delta+1}$ as a subgraph, Proposition 1 along with Theorem 3 implies that q-MAJORITY ILLUSION is FPT parametrised by the maximum degree and cliquewidth of the input graph.

Corollary 2. q-MAJORITY ILLUSION *can be solved in time* $\Delta^{O(\Delta \cdot k)} n^{O(1)}$ *on networks of max-degree* Δ *and cliquewidth* k.

Neighbourhood Diversity. Here, we show that q-MAJORITY ILLUSION is FPT parametrised by ND. 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 $\bar{x}$ does not exist.

Proposition 2. [(Lenstra and Jr. 1983; Kannan 1987; Frank and Tardos 1987)]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.*

We further rely on the following facts capturing the relation between vertices of the same module in a labelled social network (proof in the extended version of this paper).

Lemma 1. *Let* (N, E) *be a social network, and let* $\mathcal{T} = \{T_1, \ldots, T_k\}$ *denote a partition of* N *into* k *modules. Further, let* $f : N \to \{r, b\}$ *be a labelling, where* b *is the majority colour. Then, the following hold.*

1. *If a vertex of an independent module is under illusion under* f, *then every vertex of this module is under illusion.*
2. *If a blue vertex (i.e, vertex labelled* b*) 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.*

Let (N, E) be the given input social network and let $\mathcal{T} = \{T_1, \ldots, T_k\}$ be the partition of N into k modules (k is the neighborhood diversity), each of which is a clique or an independent set. The set $\mathcal{T}$ can be computed in polynomial time (Lampis 2012). For every $i \in [k]$, let $adj(i)$ be the set $\{j \in [k] : j \neq i \text{ and } \exists u \in T_i, v \in T_j : (u, v) \in E\}$. That is, $adj(i)$ comprises the indices of all modules T_j in which least

one vertex (and hence all vertices) is adjacent to a vertex of T_i (and hence to all vertices of T_i). Let $\chi = \lceil qn \rceil$ denote the required number of vertices to be under illusion to have a q-majority illusion. The main idea is to construct $2^{O(k)}$ instances of ILP-FEASIBILITY each with $O(k)$ variables such that if there is a q-majority illusion, then the solution to one of these ILP-FEASIBILITY instances can be used to get a solution to the given instance of q-MAJORITY ILLUSION.

Let $\mathcal{C}$ denote the set of all clique modules in $\mathcal{T}$ and let $\mathcal{I}$ denote the set of all independent modules in $\mathcal{T}$.

ILP Instance. We are now ready to start describing the design of the ILP-FEASIBILITY instances. For every,

- Clique-col: $\mathcal{C} \to \{\mathsf{red}, \mathsf{blue}, \mathsf{both}\}$,
- Clique-maj: $\mathcal{C} \to \{\mathsf{blue}, \mathsf{all}, \mathsf{none}\}$, and
- Ind-maj: $\mathcal{I} \to \{\mathsf{all}, \mathsf{none}\}$,

we construct one ILP-feasibility instance for which the set of variables and constraints will be discussed shortly. We first sketch the intuition behind these functions. Let $\rho : V(G) \to \{\mathsf{red}, \mathsf{blue}\}$ be a labelling that 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 ρ (in which case this module is mapped to both) or it contains only red vertices (in which case this module is mapped to red) or it contains only blue vertices (then this module is mapped to blue). 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 blue) or all vertices are under illusion (mapped to all) under ρ. Recall from the second and third statements of Lemma 1 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 from the first statement of Lemma 1 that these are the only two possibilities. If ρ 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. In order to better understand the constraints we will design, we encourage the reader to consider the three selected functions to be the "correct" ones that correspond to ρ. We will also assume that these functions are consistent with each other. That is, if Clique-col (T_i) is red (respectively, blue), then it cannot be the case that Clique-maj (T_i) is blue (respectively, red). In other words, if we guess that every vertex of T_i is labelled red, then we will not guess that all the blue vertices of T_i will be under illusion. Moreover, we have a convention that in Clique-maj, all takes "priority" over red or blue. That is, if Clique-col (T_i) is blue, then Clique-maj (T_i) is either none or all and never blue. This is because setting it to all achieves the same effect as setting it to blue since all vertices in T_i are blue. Any triple of functions where these conditions are not satisfied is discarded.

Constraints in the Instance. We now describe the ILP-FEASIBILITY instance. For every $i \in [k]$, let s_i denote the size of $V(T_i)$. We know the value of each s_i since we know $\mathcal{T}$. The set of variables in this instance is $\bigcup_{i \in [k]} \{r_i, b_i, p_i\}$. The intuitive meaning of these variables is the following. Recall that ρ is a hypothetical optimal labelling that places at least χ vertices under illusion. Then, for every module T_i, r_i represents the number of vertices of T_i labelled red by ρ. Also, b_i is the number of vertices in T_i labelled blue by ρ and p_i is used to represent the number of vertices of T_i that are under illusion under ρ. Notice that we have $3k$ variables in total. This lets us formulate a number of constraints such that if we solve the ILP-FEASIBILITY instance corresponding to some triple, then we can recover a labelling admitting q-majority illusion (which may not be the same as ρ). Among others, we encode constraints saying that the number of blue and red vertices is in each module is equal to the size of that module, and that blue is the strict majority colour. We also ensure that the number of vertices under illusion from each T_i is at least p_i and the total number of vertices under illusion is at least χ. Also, for clique and independent modules, we add constraints ensuring that the number of vertices under illusion in these modules is consistent with Ind-maj and Clique-maj. The formal description of the constraints can be found in the extended version of this paper.

The running time is bounded by the time required to compute $\mathcal{T}$ (polynomial) plus 18^k multiplied by the time required to construct an ILP-FEASIBILITY instance and execute 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, giving us the following result.

Theorem 4. *q-MAJORITY ILLUSION can be solved in time $2^{O(k \log k)} n^{O(1)}$ on networks of neighbourhood diversity k.*

Since graphs with vertex cover number (VC) at most k have ND at most $k + 2^k$ (Lampis 2012), Theorem 4 gives us an FPT algorithm parametrised by VC as a corollary.

Corollary 3. *q-MAJORITY ILLUSION can be solved in time $2^{2^{O(k)}} n^{O(1)}$ on networks with VC equal to k.*

Table 1 shows an overview of the parametrised complexity results obtained in this section.

	Parameters
FPT	$\Delta + tw$, $\Delta + cw$, ND, VC
XP	tw
Para-NP-Hard	Δ, c-closure, max-clique-size

Table 1: Main parametrised complexity results in Section 3.

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 a q-majority illusion does not hold in a labelled network by altering only a bounded number of edges.

q-ILLUSION ELIMINATION:
Input: (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 (N, E', f) such that $|\{(e \in N^2 : e \in E \text{ iff } e \notin E'\}| \leq k$ and f does not induce q-majority illusion in (N, E', f)?

We also consider two refinements of this problem, ADDITION q-ILLUSION ELIMINATION in which we restrict the possible actions to adding edges and REMOVAL q-ILLUSION ELIMINATION for removing edges.

4.1 Hardness

In this section we show that q-ILLUSION ELIMINATION is NP-complete. In fact, our construction implies that this problem is also W[1]-hard parametrised by the number of changed edges in a social network, which we denote as m. We obtain that by providing the required reduction from k-CLIQUE. There, given a graph G, the it is checked whether G includes a clique of size k as a subgraph.

Let us present the high-level description of this proof, which can be found in the extended version of this paper. First, for a given graph $G = (N_G, E_G)$, we construct an instance EN_G of q-ILLUSION ELIMINATION, i.e., a labelled social network (N, E, f), and a natural number k', such that G contains a k-clique if and only if it is necessary and sufficient to eliminate edges between the vertices in a k-clique in the structure which we call a *G-gadget* to ensure that there are at most $|N_G - k|$ vertices under illusion in the modified network. To define the G-gadget, we take G, with all vertices in G labelled r, as a subnetwork of EN_G. Then, for every vertex i in this gadget we construct a number of red and blue vertices adjacent to i such that red has the majority of $k - 1$ in i's neighbourhood. We also ensure that the only vertices under illusion are the vertices in the G-gadget by constructing additional vertices in EN_G. Setting k' to $k^2 - k$, i.e., the number of edges in a k-clique, we get that if there exists a k-clique in G, then it is sufficient to eliminate edges within this clique to ensure that k of its members are not under illusion anymore. Then, if there is no k-clique in the G-gadget, we demonstrate that eliminating illusion from k vertices in this encoding requires the modification of at least $k^2 - k + 2$ edges. This concludes the proof of the claim.

We further extend the previously discussed construction to show that the NP-hardness of q-ILLUSION ELIMINATION holds for every rational $q \in (0, 1)$. So, for every graph G, and for every such q, we provide an instance I of q-ILLUSION ELIMINATION, for which the answer is positive if and only if G contains a k-clique. Towards this end, take a graph G and consider the encoding EN_G. To ensure that the claim holds, we extend EN_G with a what we call an *m-pump-up*, or with what we call an *m-pump-down* gadget. The m-pump-up gadget is a structure such that, if an m-pump-up gadget is embedded in a network in which b is the majority winner, then $m + 4$ vertices are under illusion in the gadget, while 4 are not. Also, for every vertex i in the m-pump-up gadget which is labelled b, r has the majority of 4 in i's neighbourhood. In addition, if the 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 vertex labelled b in the gadget would be adjacent to an additional vertex labelled r, then it would be under illusion. The properties of these gadgets help us to ensure that the illusion needs to be eliminated from exactly k vertices for q-majority illusion not to hold, while it is only possible by ensuring that vertices in a k-clique are not under illusion. So, by extending EN_G with one of these gadget, which depends on the number of vertices under illusion in EN_G, we ensure that the answer to I, with $k' = k^2 - k$, is positive if and only if illusion can be eliminated from members of a k-clique in the G-gadget by removing all of the edges between its members. Thus the claim holds. Finally, we notice that the number of changed edges in the constructed instance of q-ILLUSION ELIMINATION is a quadratic function of k. So, the reduction from k-clique implies W[1]-hardness of this problem.

Theorem 5. *For all $q \in (0, 1)$ q-ILLUSION ELIMINATION is W[1]-hard parametrised by the solution size k.*

Using similar reductions we get W[1]-hardness of ADDITION and REMOVAL q-ILLUSION ELIMINATION.

Theorem 6. *For all $q \in (0, 1)$, ADDITION and REMOVAL q-ILLUSION ELIMINATION are W[1]-hard.*

5 Conclusion

We showed the algorithmic hardness of checking (Theorems 1 and 2) and eliminating (Theorems 5 and 6) majority illusion, together with a number of parametrised algorithms for the verification problem (Table 1) and W[1]-hardness for elimination (Theorems 5 and 6).

The central take away message is that even if illusion identification is a hard problem in general, there are various natural constraints that make it feasible. For elimination, the hardness persists for some natural restriction, and finding good parametrisations is still an interesting challenge. Another open challenge is to lift the assumption of binary labelling. Surprisingly there are social networks that do not admit a majority illusion but do admit a "plurality" illusion, i.e., agents have a wrong perception of the plurality winner, when more than two colours are allowed (see extended version of this paper). This is particularly relevant for voting contexts such as elections with multiple candidates. Furthermore, the problem of establishing the complexity of checking if a network admits q-majority illusion for fractions smaller than or equal to $\frac{1}{2}$ remains open (cf. Theorem 1). Finally, exploring the connections between majority illusion and opinion diffusion is a natural and important follow up.

Acknowledgments

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).

References

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

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

Auletta, V.; Caragiannis, I.; Ferraioli, D.; Galdi, C.; and Persiano, G. 2017. Information retention in heterogeneous majority dynamics. In *International Conference on Web and Internet Economics*, 30–43. Springer.

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

Berman, P.; Karpinski, M.; and Scott, A. 2004. Approximation hardness of short symmetric instances of MAX-3SAT. Technical report, Weizmann Institute of Science.

Bredereck, R.; and Elkind, E. 2017. Manipulating Opinion Diffusion in Social Networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*.

Castiglioni, M.; Ferraioli, D.; and Gatti, N. 2020. Election Control in Social Networks via Edge Addition or Removal. In *The 34th AAAI Conference on Artificial Intelligence (AAAI)*.

Castiglioni, M.; Ferraioli, D.; Gatti, N.; and Landriani, G. 2021. Election Manipulation on Social Networks: Seeding, Edge Removal, Edge Addition. *Journal of Artificial Intelligence Research*, 71: 1049–1090.

Chitnis, R.; and Talmon, N. 2018. Can We Create Large k-Cores by Adding Few Edges? In Fomin, F. V.; and Podolskii, V. V., eds., *Computer Science – Theory and Applications*, 78–89. Cham: Springer International Publishing.

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

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

Faliszewski, P.; Gonen, R.; Koutecký, M.; and Talmon, N. 2018. Opinion Diffusion and Campaigning on Society Graphs. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*.

Fox, J.; Roughgarden, T.; Seshadhri, C.; Wei, F.; and Wein, N. 2018. Finding Cliques in Social Networks: A New Distribution-Free Model. In Chatzigiannakis, I.; Kaklamanis, C.; Marx, D.; and Sannella, D., eds., *ICALP*, volume 107, 55:1–55:15.

Fox, J.; Roughgarden, T.; Seshadhri, C.; Wei, F.; and Wein, N. 2020. Finding Cliques in Social Networks: A New Distribution-Free Model. *SIAM J. Comput.*, 49(2): 448–464.

Frank, A.; and Tardos, É. 1987. An application of simultaneous Diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1): 49–65.

Granovetter, M. 1978. Threshold models of collective behavior. *American journal of sociology*, 83(6): 1420–1443.

Gurski, F.; and Wanke, E. 2000. The Tree-Width of Clique-Width Bounded Graphs Without $K_{n,\,n}$. In *26th International Workshop in Graph-Theoretic Concepts in Computer Science*, volume 1928 of *Lecture Notes in Computer Science*, 196–205. Springer.

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

Johnson, N.; Velásquez, N.; Restrepo, N.; Leahy, R.; Gabriel, N.; El Oud, S.; Zheng, M.; Manrique, P.; Wuchty, S.; and Lupu, Y. 2020. The online competition between pro- and anti-vaccination views. *Nature*, 582(7811): 230–233.

Kannan, R. 1987. Minkowski's Convex Body Theorem and Integer Programming. *Math. Oper. Res.*, 12(3): 415–440.

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

Kempe, D.; Kleinberg, J. M.; and Tardos, É. 2015. Maximizing the Spread of Influence through a Social Network. *Theory Comput.*, 11: 105–147.

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

Lampis, M. 2012. Algorithmic Meta-theorems for Restrictions of Treewidth. *Algorithmica*, 64(1): 19–37.

Lenstra, H. W.; and Jr. 1983. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4): 538–548.

Lerman, K.; Yan, X.; and Wu, X.-Z. 2016. The "majority illusion" in social networks. *PloS one*, 11(2).

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

Stewart, A. J.; Mosleh, M.; Diakonova, M.; Arechar, A. A.; Rand, D. G.; and Plotkin, J. B. 2019. Information gerrymandering and undemocratic decisions. *Nature*, 573(7772): 117–121.

Wilder, B.; and Vorobeychik, Y. 2018. Controlling Elections through Social Influence. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Zhou, X.; and Zhang, Z. 2021. *Maximizing Influence of Leaders in Social Networks*, 2400–2408. New York, NY, USA: Association for Computing Machinery.

Zhou, Z.; Zhang, F.; Lin, X.; Zhang, W.; and Chen, C. 2019. K-Core Maximization: An Edge Addition Approach. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*.