To appear in Algorithmica, Special Issue of ISAAC 2009

Editing Graphs into Disjoint Unions of Dense Clusters

Jiong Guo* Iyad A. Kanj† Christian Komusiewicz‡
Johannes Uhlmann§

Universität des Saarlandes,
Campus E 1.7, D-66123 Saarbrücken, Germany
jguo@mmci.uni-saarland.de
School of Computing, DePaul University,
243. S. Wabash Avenue, Chicago, IL 60604, USA
ikanj@cs.depaul.edu
Institut für Informatik, Friedrich-Schiller-Universität Jena
Ernst-Abbe-Platz 2, D-07743 Jena, Germany
{c.komus,johannes.uhlmann}@uni-jena.de

Abstract

In the Π-Cluster Editing problem, one is given an undirected graph G, a density measure Π, and an integer $k \geq 0$, and needs to decide whether it is possible to transform G by editing (deleting and inserting) at most k edges into a dense cluster graph. Herein, a dense cluster graph is a graph in which every connected component $K = (V_K, E_K)$ satisfies Π. The well-studied Cluster Editing problem is a special case of this problem with Π :=“being a clique”. In this work, we consider three other density measures that generalize cliques: 1) having at most s missing edges (s-defective cliques), 2) having average degree at least $|V_K| - s$ (average-s-plexes), and 3) having average degree at least $\mu \cdot (|V_K| - 1)$ (μ-cliques), where s and μ are a fixed integer and a fixed rational number, respectively. We first show that the Π-Cluster Editing problem is NP-complete for all three density measures. Then, we study the fixed-parameter tractability of the three clustering problems, showing that the first two problems are fixed-parameter tractable with respect to the parameter (s, k)

*Supported by the Excellence Cluster on Multimodal Computing and Interaction (MMCI). Main work was done while the author was with the Friedrich-Schiller-Universität Jena.

†Part of this work was done while the author was visiting the Friedrich-Schiller-Universität Jena.

‡Supported by a PhD fellowship of the Carl-Zeiss-Stiftung and the DFG, research project PABI, NI 369/7.

§Supported by the DFG, research project PABI, NI 369/7.

and that the third problem is W[1]-hard with respect to the parameter k for $0 < \mu < 1$.

1 Introduction

Graph-based data clustering is an important tool for analyzing real-world data, ranging from biological to social network data. In such applications, data items are represented as vertices, and there is an edge between two vertices if and only if the interrelation between the two corresponding data items exceeds some threshold value. A clustering with respect to such a graph is a partition of the vertex set into dense subgraphs, also called clusters, such that there are few edges between the clusters. When formulated as a graph modification problem, one thus asks for a minimum-cardinality set of edge modifications, such that the resulting graph is a graph in which every connected component is a cluster. More precisely, the algorithmic task can be formalized as follows:

> Π-Cluster Editing:
> **Input:** An undirected graph $G = (V, E)$, a density measure Π, and an integer $k \geq 0$.
> **Task:** Find a set of at most k edge modifications to transform G into a Π-cluster graph, that is, a graph in which every connected component satisfies Π.

Herein, an edge modification is an insertion or deletion of an edge. Analogously, one defines Π-Cluster Deletion by allowing only edge deletions and Π-Cluster Addition by allowing only edge insertions.

One of the most prominent problems in this context is the NP-hard Cluster Editing problem (also known as Correlation Clustering) [3, 23], where the required density measure is Π :="being a clique". Cluster Editing finds applications in various fields, such as computational biology [4] and machine learning [3], and has been intensively studied from the viewpoints of polynomial-time approximability as well as parameterized algorithmics. In terms of approximability, the currently best known approximation factor is 2.5 [2, 25]. Cluster Editing can be solved in $O(1.83^k + |E|)$ time [4] and several kernelization (data reduction) algorithms for this problem have been proposed [6, 7, 11, 16]. Successful experimental studies of the parameterized algorithms for Cluster Editing have been conducted mainly in the context of computational biology [4, 9]. The related Cluster Deletion problem is also NP-hard [23].

The requirement of being a clique has been often criticized for its overly restrictive nature and modeling disadvantages [8, 22]. Consequently, less restrictive models for dense graphs have been proposed in various application scenarios [1, 22, 24]. In this work, we study the combination of clique relaxations and graph-based data clustering by charting the tractability borderlines of Π-Cluster Editing when the density requirement is relaxed. We consider three relaxed density measures, namely, *s-defective cliques*, *average-s-plexes*, and *μ-cliques*. The corresponding modification problems are s-Defective Clique

Table 1: The complexity of the problems considered in this work. For s-defective cliques and average-s-plexes, the considered parameter is (s, k), for μ-cliques, the considered parameter is k.

	Deletion	Editing	Addition
s-Defective Clique	NP-c. (Thm. 1) FPT (Thm. 3)	NP-c. (Thm. 1) FPT (Thm. 3)	P
Average-s-Plex	NP-c. (Thm. 4) FPT (Thm. 5)	NP-c. (Thm. 4) FPT (Thm. 5)	P
μ-Clique	NP-c. (Thm. 7)	W[1]-hard (Thm. 6)	P

Editing, Average-s-Plex Editing, and μ-Clique Editing. For all three density measures we also consider the deletion variants s-Defective Clique Deletion, Average-s-Plex Deletion, and μ-Clique Deletion of the clustering problems. We study the classical and the parameterized complexity of the aforementioned problems; this complements previous work where we considered a different density measure, so-called s-plexes [17]. The proposed density measures may provide more realistic models for practical applications and fixed-parameter tractability (FPT) results can serve as a first step in a series of algorithmic improvements, eventually leading to applicability in practice, as it was the case for Cluster Editing [4, 9, 11, 16]. An overview of our results is given in Table 1. Note that for all three density measures the polynomial-time solvability of the addition problem can be easily seen, and is included only for the sake of completeness. In the following, we give the exact definitions of the density measures studied in this work, point to related work, and describe our results.

Defective Cliques. The concept of defective cliques has been used in biological networks to represent a clique with exactly one edge missing [24]. Here, we generalize this notion[1] by allowing up to s missing edges: A graph $G = (V, E)$ is called an *s-defective clique*, if G is connected and $|E| \geq |V| \cdot (|V| - 1)/2 - s$. On the negative side, we prove that s-Defective Clique Deletion and Editing are NP-complete. On the positive side however, we show that s-defective cliques can be characterized by forbidden subgraphs of size at most $2(s + 1)$, thus establishing the fixed-parameter tractability of s-Defective Clique Deletion and Editing with respect to the parameter (s, k).

Average-s-Plexes. With average-s-plexes, we propose a density measure that concerns the *average* degree of a graph $G = (V, E)$, which is defined as $\bar{d} = 2|E|/|V|$. We call a connected graph $G = (V, E)$ an *average-s-plex* if the average degree $\bar{d}$ of G is at least $|V| - s$ for an integer $1 \leq s \leq |V|$. This density measure is a relaxation of the s-plex notion, which demands that the *minimum* degree of

[1]Note that Yu et al. [24] introduced a different generalization of defective cliques that is more restrictive than the one considered here.

a graph $G = (V, E)$ be $|V| - s$. s-Plexes find applications, for example, in social network analysis [22]. For s-plexes, the clustering problem s-Plex Editing has been previously shown to be NP-hard, but fixed-parameter tractable with respect to the parameter (s, k) [17]. Here, we complement this result by showing that Average-s-Plex Deletion and Editing are also NP-hard, and that they are fixed-parameter tractable with respect to the parameter (s, k). The fixed-parameter tractability result is achieved by a reduction to a more general problem and a subsequent polynomial-time data reduction for the general problem that produces a graph with at most $4k^2 + 8sk$ vertices.

μ-Cliques. With this density measure, we capture the ratio of edges in a graph versus the number of edges in a complete graph with the same number of vertices. More precisely, the *density* of a graph $G = (V, E)$ is defined as $2|E|/(|V|(|V| - 1))$. A connected graph $G = (V, E)$ is then called a *μ-clique* for a rational constant $0 \leq \mu \leq 1$ if the density of G is at least μ. We assume that μ is represented by two constant integers a, b such that $\mu = a/b$ (note that a and b are not part of the input). Observe that for $\mu = 0$ every graph is a μ-clique, and that a graph is a 1-clique if and only if it is a clique. The μ-clique concept was studied, for example, by Abello et al. [1] and is sometimes also referred to as μ-dense graph [19]. We show that—in contrast to s-Defective Clique Editing and Average-s-Plex Editing—μ-Clique Editing is W[1]-hard with respect to the parameter k for *any* fixed $0 < \mu < 1$. Note that for $\mu = 1$, the problem is equivalent to Cluster Editing and is thus fixed-parameter tractable. For μ-Clique Deletion we show the NP-hardness, the parameterized complexity remains open.

Preliminaries. We only consider *undirected* graphs $G = (V, E)$, where $n := |V|$ and $m := |E|$. The *(open) neighborhood* $N(v)$ of a vertex $v \in V$ is the set of vertices that are adjacent to v in G. The *degree* of a vertex v, denoted by $\deg(v)$, is the cardinality of $N(v)$. For a set U of vertices, $N(U) := \bigcup_{v \in U} N(v) \setminus U$. We use $N[v]$ to denote the *closed* neighborhood of v, that is, $N[v] := N(v) \cup \{v\}$. For a set of vertices $V' \subseteq V$, the *induced subgraph* $G[V']$ is the graph over the vertex set V' with edge set $\{\{v, w\} \in E \mid v, w \in V'\}$. For $V' \subseteq V$ we use $G - V'$ as an abbreviation for $G[V \setminus V']$ and for a vertex $v \in V$ let $G - v$ denote $G - \{v\}$. A vertex $v \in V(G)$ is called a *cut-vertex* if $G - v$ has more connected components than G. For a graph $G = (V, E)$ let $\overline{G} := (V, \overline{E})$ with $\overline{E} := \{\{u, v\} \mid u, v \in V \wedge u \neq v \wedge \{u, v\} \notin E\}$ denote the *complement graph* of G.

A parameterized problem is *fixed-parameter tractable (FPT)* with respect to a parameter k, if there exists an algorithm solving the problem in $f(k) \cdot n^{O(1)}$ time, where n denotes the overall input size and f is a computable function. Downey and Fellows [10] developed a formal framework to show *fixed-parameter intractability* by means of *parameterized reductions*. A parameterized reduction from a parameterized problem P to another parameterized problem P' is a function that, given an instance (x, k), computes in $f(k) \cdot |x|^{O(1)}$ time an instance (x', k') (with k' only depending on k) such that (x, k) is a yes-instance of problem P if and only if (x', k') is a yes-instance of problem P'. The basic complexity class for fixed-parameter intractability is called $W[1]$ and it is a common belief among researchers in parameterized complexity theory that $W[1]$-hard problems are not FPT [10, 21].

2 Defective Cliques

In this section, we focus on the s-Defective Clique Editing problem. A graph is called an *s-defective clique graph* if every connected component forms an s-defective clique. An edge $\{u, v\} \in \overline{E}$ is called a *missing* edge of G. Our contribution is twofold. First, we prove the NP-hardness of s-Defective Clique Editing by a reduction from Cluster Editing. Second, we present a characterization of s-defective clique graphs by means of forbidden induced subgraphs whose size is bounded by $O(s)$, directly leading to fixed-parameter tractability of s-Defective Clique Editing with respect to the parameter (s, k).

Theorem 1. *For any constant $s \geq 0$, s-Defective Clique Deletion and Editing are NP-complete.*

Proof. For $s = 0$, the problems are equivalent to Cluster Editing and Cluster Deletion, and hence NP-complete [20, 23]. For $s > 1$, we present a reduction from Cluster Editing to s-Defective Clique Editing. Then, we observe that the same construction also yields a reduction from Cluster Deletion to s-Defective Clique Deletion.

Given a graph $G = (V, E)$ and a nonnegative integer k we construct a graph $H = (W, F)$ and an integer k' as follows. Let $n := |V|$ and $\gamma := 16(n^2+s)$. Initially, we set $H := G$. Then, for every $1 \leq i \leq n$ we add a set D_i of γ vertices to H and make every vertex in D_i adjacent to all vertices in V. Within each D_i we add all but s edges such that the missing edges do not share endpoints. Finally, we set $k' := k + n(n-1)\gamma$.

For the correctness of the reduction, we show that (G, k) is a yes-instance of Cluster Editing if and only if (H, k') is a yes-instance of s-Defective Clique Editing.

$\Rightarrow$: Assume that G can be transformed by up to k edge modifications into a cluster graph G' and let $K_1, \ldots, K_\ell$ denote the cliques (connected components) of G'. We show how to transform H into an s-defective clique graph H' using at most k' edge modifications. First, the edges between vertices of V are modified in H in the same way as in G. Then, consider each vertex $v \in V$ and let K_j denote the clique containing v. For every $i \neq j$, $1 \leq i \leq n$, delete the edges between v and D_i. Let H' denote the resulting graph. Clearly, H' is an s-defective clique graph. Moreover, in order to transform H to H' one needs at most k edge modifications between vertices in V and for every vertex $v \in V$ exactly $(n-1)\gamma$ further edge deletions.

$\Leftarrow$: Let S denote an optimal solution for H of size at most $k' = k+n(n-1)\gamma$. Let H' denote the graph that results by modifying H according to S, and let $K_1, \ldots, K_\ell$ denote the connected components of H'. Clearly, each K_j for $1 \leq j \leq \ell$ is an s-defective clique.

First, we show that for every D_i with $1 \leq i \leq n$ there is a K_j with $D_i \subseteq K_j$. Assume towards a contradiction that this is not the case. Then there is one D_i, say D_1, such that there are several connected components of H' that contain vertices from D_1. Without loss of generality assume that these are the first t K_j's for an integer $t > 1$. Let $B_j := D_1 \cap K_j$, $1 \leq j \leq t$. First, observe that there exists a B_j with $|B_j| > \gamma/2$: if $|B_j| \leq \gamma/2$ for all $1 \leq j \leq t$, then we need

$$\begin{aligned} -s + 1/2 \cdot \sum_{j=1}^{t} |B_j| \cdot (\gamma - |B_j|) &\geq \; -s + 1/2 \cdot \sum_{j=1}^{t} |B_j| \cdot \gamma/2 \\ &\geq \; -s + \gamma/4 \cdot \sum_{j=1}^{t} |B_j| = \gamma^2/4 - s > k' \end{aligned}$$

edge deletions; this is not allowed. Hence, there is a B_j with $|B_j| > \gamma/2$, say B_1. Using this fact, we can show that $D_j \cap K_1 = \emptyset$ for every $2 \leq j \leq n$: Let $X := \cup_{j \neq 1}(D_j \cap K_1)$. If $X \neq \emptyset$, then, compared to putting B_1 and X together in K_1, separating X from K_1 saves, for each vertex $v \in X$, at least $\gamma/2 - s > n$ edge insertions that are needed to put B_1 and X together, and needs at most n additional edge deletions to separate v from $K_1 \setminus (B_1 \cup X)$. From the optimality of S it follows that D_j and K_1 are disjoint for $j > 1$. Next, consider B_j with $j > 1$. We count the number of edge modifications needed for separating B_j from K_j and putting B_j together with B_1 in K_1. First we need to delete the edges between B_j and $K_j \cap V$ and to reinsert the edges between B_j and $K_1 \cap V$, which together amount to at most $n \cdot |B_j|$ edge modifications. Then, to ensure that $K_1 \cup B_j$ is an s-defective clique we insert all edges between vertices in $B_1 \cup B_j$. However, by doing so, we save at least $|B_1| \cdot |B_j| - s > n \cdot |B_j| + s$ edge deletions which are needed to separate B_1 and B_j. This means that in an optimal solution $B_j = \emptyset$ for $j > 1$. Therefore, $B_1 = D_1 \subseteq K_1$, contradicting the assumption that there are $t > 1$ connected components that have nonempty intersection with D_1. Hence, we have shown that for every D_i there is a K_j such that $D_i \subseteq K_j$.

Furthermore, observe that $D_{i_1} \subseteq K_j$ and $D_{i_2} \subseteq K_j$ implies $i_1 = i_2$ since, otherwise, we would need at least $\gamma^2 > k'$ edge insertions. Hence, we can assume without loss of generality that $D_i \subseteq K_i \subseteq D_i \cup V$ for every $1 \leq i \leq n$ and thus that in H' every vertex $v \in V$ is adjacent to the vertices of at most one D_i. Next, we show that a vertex $v \in V$ is adjacent to the vertices of at least one D_i. Assume that there exist α vertices in V that are nonadjacent to all vertices in D_i's. Then at least $\alpha \cdot n\gamma + (n - \alpha)(n-1)\gamma = \alpha \cdot \gamma + n(n-1)\gamma > k'$ edge deletions are necessary, contradicting the assumption that $|S| \leq k'$. In summary, every vertex in V is adjacent to all vertices of exactly one D_i and exactly $n(n-1)\gamma$ edge deletions are used to achieve this. That is, $\ell = n$ and $D_i \subseteq K_i \subseteq D_i \cup V$ for $1 \leq i \leq \ell$.

We conclude the proof by showing that it can be assumed that the sets $V_i := V \cap K_i$ form cliques in $H'[V]$. Assume that there exists a missing edge in $H'[V_i]$.

Since $K_i = V_i \cup D_i$ forms an s-defective clique, S must contain an edge insertion within D_i (note that $H[D_i]$ has exactly s missing edges). Clearly, we can undo this edge insertion and insert an edge between two vertices of V_i instead. This can be done iteratively until $H'[V_i]$ is a clique. Hence, modifying G in the same way as $H[V]$ needs at most k edge modifications and yields a disjoint union of cliques.

It is straightforward to verify that all steps of the proof can be adapted to work for the case that only edge deletions are allowed. Hence, the given construction yields a reduction from Cluster Deletion to s-Defective Clique Deletion, as well. □

Next, we show that s-defective clique graphs are characterized by forbidden induced subgraphs with at most $2(s+1)$ vertices. This characterization directly leads to fixed-parameter tractability of s-Defective Clique Editing with respect to the parameter (s, k) by means of a search tree based algorithm.

We start with some preliminaries. If we delete an arbitrary vertex of an s-defective clique graph, then clearly the resulting graph is still an s-defective clique graph. A graph property that is closed under the operation of deleting vertices (and hence, taking induced subgraphs) is called *hereditary*. It is well known that hereditary graph properties can be described by forbidden induced subgraphs [15]. This means that there exists a set $\mathcal{F}$ of graphs such that a given graph G is an s-defective clique graph if and only if G is $\mathcal{F}$-free, that is, G does not contain any graph from $\mathcal{F}$ as induced subgraph. A forbidden induced subgraph is *minimal* if each of its proper induced subgraphs is an s-defective clique graph. Clearly, a graph is an s-defective clique graph if and only if it does not contain any minimal forbidden induced subgraph. Next, we show that, for $s \geq 1$, every minimal forbidden induced subgraph of s-defective clique graphs contains at most $2(s+1)$ vertices. Note that for $s = 0$ the only forbidden induced subgraph is a path on 3 vertices [14, 23].

Theorem 2. *For $s \geq 1$, every minimal forbidden induced subgraph of s-defective clique graphs contains at most $2(s+1)$ vertices. Given a graph that is not an s-defective clique graph, a minimal forbidden induced subgraph can be found in $O(nm)$ time.*

Proof. Assume towards a contradiction that there exists a minimal forbidden induced subgraph $G = (V, E)$ with $|V| > 2(s+1)$. Clearly, we can assume that G is connected, since otherwise we can keep one connected component that is not an s-defective clique and delete all other connected components.

First, we consider the case when G contains a cut-vertex v. Let U denote a set of $s + 2$ vertices which together with v induce a connected graph $G' := G[U \cup \{v\}]$ and v remains a cut-vertex in G'. We show that G' is not an s-defective clique graph, a contradiction to the fact that G is minimal (note that $s+3 \leq 2s+2$ for $s \geq 1$). Let $U_1, \ldots, U_\ell$ denote the connected components of $G' - v$. It is not hard to see that there are at least $\frac{1}{2}\sum_{i=1}^{\ell} |U_i| \cdot (|U \setminus U_i|) > s$ edges missing in G', and, hence, G' is not an s-defective clique graph.

In the following, we assume that G does not contain any cut-vertex. Moreover, we can assume that no vertex of G is adjacent to all other vertices of G, since otherwise we can delete it to get a connected graph that has the same number of missing edges as G, thus contradicting the minimality of G. Hence, there are more than $s+1$ missing edges in G, since every vertex is incident to at least one missing edge. Let v be an arbitrary vertex of G and let $A := V \setminus N[v]$. Since the deletion of v results in an s-defective clique graph, it follows that in $G - v$ there are at most s missing edges. Hence, there exists a vertex u that is adjacent to all vertices of $G - v$. Clearly, $u \in A$, since, otherwise, u would be adjacent to all vertices in G. Then, the deletion of u reduces the number of missing edges by one. Thus, $G - u$ is connected and has at least $s + 1$ missing edges, and, hence, is not an s-defective clique graph, contradicting the fact that G is minimal.

To find a minimal forbidden induced subgraph proceed as follows. Given a graph $G = (V, E)$ that is not an s-defective clique graph we check for every $v \in V$ whether $G - v$ is an s-defective clique graph in $O(n + m)$ time and delete v if this is not the case. Observe that we have to consider every vertex at most once, since if $G - v$ is an s-defective clique graph, then $G' - v$ is an s-defective clique graph for every induced subgraph G' of G containing v. Hence, the overall running time is $O(nm)$. ☐

The forbidden subgraph characterization given in Theorem 2 directly leads to a search tree algorithm for s-Defective Clique Editing [5]. As long as the given graph is not an s-defective clique graph find a minimal forbidden induced subgraph and branch into all cases (at most $\binom{2s+2}{2}$) to destroy this subgraph by adding or deleting an edge between two of its vertices. Since in each case we can decrease the parameter k by one, the size of the search tree is bounded by $O\left(\binom{2s+2}{2}^k\right)$. Putting all together, we arrive at the following.

Theorem 3. *s-Defective Clique Editing and Deletion can be solved in $O(\binom{2s+2}{2}^k \cdot nm)$ time and hence are fixed-parameter tractable with respect to the parameter (s, k).*

3 Average-s-Plexes

Here, we consider the Average-s-Plex Editing problem, showing its NP-completeness and its fixed-parameter tractability with respect to (s, k). To prove the NP-hardness, we make use of the following problem.

Equal-Size Clique Editing:
Input: An undirected graph $G = (V, E)$ and two integers $k, d \geq 0$.
Question: Can G be transformed by at most k edge modifications into a vertex-disjoint union of d cliques which have the same size ?

The edge deletion version of this problem allows only edge deletions. We show the NP-hardness of Equal-Size Clique Editing and Deletion by a reduction from the well-known NP-hard Clique problem.

Lemma 1. Equal-Size Clique Editing *and* Deletion *are NP-complete.*

Proof. Clearly, both problems are contained in NP. Next, we prove only the NP-hardness of the edge deletion version. The same reduction works also for the editing version. The reduction is from the NP-complete Clique problem [13]: Given a graph $G = (V, E)$ and an integer $\ell > 0$, decide whether there exists a clique of size ℓ. The construction of the Equal-Size Clique Deletion instance is simple: Add to the given Clique instance $G = (V, E)$ a set of $|V| - \ell$ vertex-disjoint size-$(\ell - 1)$ cliques and make all vertices in these cliques adjacent to all vertices in G. Set $d := |V| - \ell + 1$ and $k := |E| - \ell(\ell - 1)/2 + \ell(|V| - \ell)(\ell - 1) + (|V| - \ell)(|V| - \ell - 1)(\ell - 1)$.

To see the equivalence between the solutions, consider a size-ℓ clique K of G. A corresponding solution of the Equal-Size Clique Deletion instance consists of the deletions of the edges between K and $V \setminus K$, the deletions of the edges between the vertices in $V \setminus K$, the deletions of the edges between K and all introduced $|V| - \ell$ cliques, and, for each vertex $v \in (V \setminus K)$, the deletions of the edges between v and all but one introduced cliques. These edge deletions amount to k. Clearly, we have d size-ℓ cliques at the end.

The other direction can be shown as follows: Clearly, each of the newly introduced cliques needs exactly one vertex from V, which must be separated from all other vertices of the newly introduced cliques and from all other vertices in V. A simple calculation shows that this requires already k edge deletions. Hence, the remaining ℓ vertices of V must form a clique. □

Reducing from Equal-Size Clique Editing and Deletion we can show the following.

Theorem 4. *For any constant $s \geq 1$,* Average-s-Plex Editing *and* Deletion *are NP-complete.*

Proof. Clearly both problems are in NP. We show only that the deletion version is NP-hard by giving a reduction from Equal-Size Clique Deletion; the reduction for the editing version is based on the same idea and therefore omitted. For $s = 1$, the problems are equivalent to Cluster Editing and Cluster Deletion, and hence NP-complete [20, 23]. In the following, we therefore assume that $s > 1$.

Given an Equal-Size Clique Deletion instance $(G = (V, E), k, d)$, we add d identical connected components to G. Each of these components, denoted by (V_C, E_C), satisfies $|V_C| = |V|^4$ and $|E_C| = 1/2 \cdot [|V_C|(|V_C| - 1) - (s - 1) \cdot (|V_C| + |V|/d)]$. Note that graphs with this property exist since we can assume that $s < |V|$ and $|V| \geq 2$. Then, make all vertices in these introduced components adjacent to all vertices in G and set $k' := k + (d - 1)|V_C| \cdot |V|$. The key observation to show the equivalence between the solutions is that each of the newly introduced components does not satisfy the condition for average-s-plex and adding a size-$(|V|/d)$ clique to such a component will transform it into an average-s-plex. In particular, adding a size-$|V|/d$ clique leads to a graph with average degree *exactly* $|V_C| + |V|/d - s$ and adding fewer vertices is therefore not sufficient. Thus, given a size-k solution S of size k for the Equal-Size Clique Deletion instance, we match the resulting size-($|V|/d$) cliques to the introduced components and for each of the cliques delete the edges between it and all components except the matched one. Adding these edge deletions to S will give us a solution to the Average-s-Plex Deletion instance. The same argument applies for the reverse direction. □

In the following, we describe a fixed-parameter algorithm for Average-s-Plex Editing parameterized by (s, k). Our algorithm consists of two main steps. First, we reduce the original problem to a weighted version. Then, we show the fixed-parameter tractability of the weighted version by describing two polynomial-time data reduction rules that yield instances which contain at most $4k^2 + 8sk$ vertices. Note that being an average-s-plex graph is not a hereditary graph property. Hence, the fixed-parameter tractability of Average-s-Plex Editing and Average-s-Plex Deletion cannot be shown by a forbidden subgraph characterization as in the case of s-Defective Clique Deletion and s-Defective Clique Editing.

We begin with describing a weighted version of Average-s-Plex Editing. We introduce three types of weights: two vertex weights and one edge weight. The idea behind these weight types is the following: whenever there are two vertices in G that cannot be separated by at most k edge modifications, we can merge them into a new "super-vertex", since it is clear that they must end up in the same connected component of the solution. We say that a super-vertex v "comprises" a vertex u of the input graph, if u is merged into v. When doing so, we must remember for each such super-vertex v:

- How many vertices of the input graph v comprises,
- How many edges there are between the vertices that v comprises, and
- For each vertex w outside v, how many vertices that v comprises are adjacent to w.

The first two aspects can be remembered by introducing two weights for v, $\sigma(v)$ which keeps track of the number of vertices comprised by v, and $\delta(v)$ which keeps track of the number of edges between these vertices. The third aspect can be stored as the edge weight $\omega(e)$ for the edge $e = \{w, v\}$. Herein, we call a vertex pair having no edge between them a "nonedge". Then, edges have edge weight at least one and nonedges have edge weight zero. In the following, we will assume that the weight functions δ and ω have the following limits on their values: For every $v \in V$, it must hold that $\delta(v) \leq \sigma(v) \cdot (\sigma(v) - 1)/2$. For every edge $\{u, v\}$ it must hold that $\omega(\{u, v\}) \leq \sigma(u) \cdot \sigma(v)$. We call this property *σ-limited*. Informally, it ensures that there is indeed an undirected graph from which the weighted graph can be obtained by merging vertices and it is also necessary for showing the size-bound on the reduced instance.

We introduce the following two notions for these weighted graphs. The "size" of a vertex set S is simply defined as $\sigma(S) := \sum_{v \in S} \sigma(v)$. The average degree $\bar{d}(V_i)$ of a connected component $G[V_i]$ can be computed as follows:

$$\bar{d}(V_i) = \frac{2 \sum_{v \in V_i} \delta(v) + \sum_{v \in V_i} \sum_{u \in N(v)} \omega(\{u, v\})}{\sigma(V_i)}.$$

Similar to the definition of average-s-plex graphs, we say that a graph is a weighted average-s-plex graph, if for each connected component $G[V_i]$, the average degree $\bar{d}(V_i)$ is at least $\sigma(V_i) - s$. For modifying the weighted graph, we allow the following modifications:

- increasing $\delta(u)$ by one for some $u \in V$,
- increasing $\omega(\{u, v\})$ by one for some $\{u, v\} \in E$,
- decreasing $\omega(\{u, v\})$ by one for some $\{u, v\} \in E$ with $\omega(\{u, v\}) > 1$,
- deleting some $\{u, v\} \in E$ with $\omega(\{u, v\}) = 1$, and
- adding some edge $\{u, v\}$ to E and setting $\omega(\{u, v\}) := 1$.

Each of these operations has cost one, and the overall cost of a modification set S is thus exactly $|S|$. We assume that all these operations can only be applied when the weight function that is changed remains σ-limited after the modification.

The weighted problem version is then defined as

> Weighted Average-s-Plex Editing
> **Input:** A graph $G = (V, E)$, with a vertex-weight function $\sigma : V \to [1, n]$, a σ-limited vertex-weight function $\delta : V \to [0, n^2]$, a σ-limited edge-weight function $\omega : E \to [1, n^2]$, and a nonnegative integer k.
> **Question:** Is there a set of edge modifications S such that applying S to G yields a weighted average-s-plex graph and $|S| \leq k$?

Observe that we can easily reduce an instance $((V, E), k)$ of Average-s-Plex Editing to an instance of Weighted Average-s-Plex Editing, by setting $\sigma(v) := 1$ and $\delta(v) := 0$ for each $v \in V$, and $\omega(\{u, v\}) := 1$, if $\{u, v\} \in E$; otherwise, $\omega(\{u, v\}) := 0$. Note that this reduction is parameter-preserving, that is, s and k are not changed.

In the following, we present two data reduction rules for Weighted Average-s-Plex Editing which (as we show in Theorem 5) yield instances that contain at most $4k^2 + 8sk$ vertices.

Rule 1. *Remove connected components that are weighted average-s-plexes from G.*

The rule is obviously correct, since no optimal solution modifies any edges incident to vertices of such a connected component.

The second reduction rule identifies two vertices that have a large common neighborhood, or a "heavy" edge between them and "merges" these vertices into a new super-vertex.

Rule 2. *If G contains two vertices u and v such that $\omega(\{u, v\}) > k$ or u and v have more than k common neighbors, then remove u from G and set*

- $\sigma(v) := \sigma(u) + \sigma(v)$,
- $\delta(v) := \delta(u) + \delta(v) + \omega(\{u, v\})$, *and*
- $\omega(\{v, w\}) := \omega(\{v, w\}) + \omega(\{u, w\})$ *for each $w \in V \setminus \{u, v\}$.*

To see the correctness of the rule, consider the following: we cannot separate u and v using at most k edge modifications; thus, they must end up in the same connected component. Hence, we can remove one of them, and store the information about its adjacency in the vertex weights and edge weights of the other vertex. Note that Rule 2 preserves the property of being σ-limited for both δ and ω.

With these two reduction rules we can show our main result of this section.

Theorem 5. (Weighted) Average-s-Plex Editing *and* Deletion *are fixed-parameter tractable with respect to the parameter* (s, k).

Proof. We first show that a yes-instance I of Weighted Average-s-Plex Editing that is reduced with Rules 1 and 2 contains at most $4k^2 + 8sk$ vertices. Let I be such a reduced instance, and let G be the input graph of I. Since I is a yes-instance, there is a weighted average-s-plex graph G' that can be obtained from G by applying at most k edge modifications. We now bound the size of G'. Herein, we call a vertex v "affected" if v is an endpoint of a modified edge or if $\delta(v)$ has been increased.

First, since G is reduced with respect to Rule 1, there is at least one affected vertex in each connected component of G'. Hence, there can be at most $2k$ connected components in G'.

Next, we show that each connected component of G' contains at most $2k+4s$ vertices. Suppose towards a contradiction that there is a connected component V_i of G' such that $|V_i| > 2k + 4s$. Let $u \in V_i$ be a vertex that has a maximum number of neighbors in V_i. Since G' is a weighted average-s-plex graph, the average vertex degree $\bar{d}(V_i)$ of $G'[V_i]$ is at least $\sigma(V_i) - s$. Since $|V_i| \leq \sigma(V_i)$, u has at least $|V_i| - s \geq 2k+3s$ neighbors in $G'[V_i]$. We consider two cases for $\sigma(u)$.
Case 1: $\sigma(u) \geq \sigma(V_i)/2$**.** We show that the average degree $\bar{d}(V_i)$ of $G'[V_i]$ is less than $\sigma(V_i) - s$, contradicting the assumption that G' is a weighted average-s-plex graph. Since G is reduced with respect to Rule 2, each edge in G has weight at most k. Furthermore, the *overall* edge weight increase of edges incident to u is at most k. The average edge weight of edges incident to u in G' is thus at most $k + 1$, since u has at least $2k + 3s$ neighbors in G'. However, with $\sigma(u) \geq \sigma(V_i \setminus \{u\}) \geq 2k + 3s$, this means that the average weight of edges incident to u is at most $\sigma(u)/2$. This leads to a low average degree. More precisely, we can bound the average degree of $G'[V_i]$ as follows:

$$
\begin{aligned}
\bar{d}(V_i) &\overset{(*)}{<} \frac{\sigma(V_i) \cdot (\sigma(V_i) - 1) - (\sigma(u)/2) \cdot \sigma(V_i \setminus \{u\})}{\sigma(V_i)} \\
&\overset{(**)}{<} \frac{\sigma(V_i) \cdot (\sigma(V_i) - 1) - (\sigma(u)/2) \cdot 4s}{\sigma(V_i)} \\
&\overset{(***)}{\leq} \frac{\sigma(V_i) \cdot (\sigma(V_i) - 1) - (\sigma(V_i)/4) \cdot 4s}{\sigma(V_i)} < \sigma(V_i) - s.
\end{aligned}
$$

Inequality (*) can be obtained from the following observations: The maximum value that can be achieved for the sum of δ and ω of $G'[V_i]$ is $\sigma(V_i) \cdot (\sigma(V_i) - 1)$ because both δ and ω are σ-limited. From this we have to subtract the missing weight for the edges incident to u, which, as described above, have average weight at most $\sigma(u)/2$. Inequality (**) follows from $\sigma(V_i \setminus \{u\}) \geq |V_i| - 1 > 4s$,

and inequality (***) follows from $\sigma(u) \geq \sigma(V_i)/2$.
Case 2: $\sigma(u) < \sigma(V_i)/2$**.** First, we show that there must be at least one other vertex $w \in V_i$ that has at least $|V_i| - 2s$ neighbors in $G'[V_i]$. Suppose otherwise. Then the average degree of $G'[V_i]$ can be bounded as follows

$$\bar{d} \overset{(*)}{\leq} \frac{\sigma(V_i) \cdot (\sigma(V_i) - 1) - 2s \cdot (\sigma(V_i) - \sigma(u))}{\sigma(V_i)}$$
$$\overset{(**)}{<} \frac{\sigma(V_i) \cdot (\sigma(V_i) - s)}{\sigma(V_i)}.$$

Inequality (*) can be obtained from the following observations: The maximum possible value that can be achieved for the sum of δ and ω of $G'[V_i]$ is $\sigma(V_i) \cdot (\sigma(V_i) - 1)$ because both δ and ω are σ-limited. From this we have to subtract the at least $2s$ edges that are missing for each vertex $v \in V_i \setminus \{u\}$. Inequality (**) follows from $\sigma(V_i) - \sigma(u) > \sigma(V_i)/2$. We have thus shown that there is at least one other vertex w that is adjacent to at least $|V_i| - 2s$ vertices in G'. Since u has at least $|V_i| - s$ neighbors in $G'[V_i]$, there must be at least $|V_i| - 3s > 2k + 4s - 3s = 2k + s$ vertices in $G'[V_i]$ that are common neighbors of u and w. Clearly, more than $k + s$ of those vertices are common neighbors of u and w in G. This contradicts the assumption that G is reduced with respect to Rule 2.

We have thus shown that a reduced yes-instance contains at most $4k^2 + 8sk$ vertices. We obtain a fixed-parameter algorithm for Weighted Average-s-Plex Editing as follows. First we exhaustively apply the reduction rules, which can clearly be done in polynomial time. If the reduced instance contains more than $4k^2 + 8sk$ vertices, then it is a no-instance. Otherwise, we can solve the problem with running time only depending on s and k, for example by brute-force generation of all possible partitions of the graph. The fixed-parameter tractability of Average-s-Plex Editingthen directly follows from the described reduction to Weighted Average-s-Plex Editing. □

4 μ-Cliques

The main result of this section is that, in contrast to s-Defective Clique Editing and Average-s-Plex Editing, μ-Clique Editing is fixed-parameter intractable with respect to the number k of allowed edge modifications. We show the parameterized and classical intractability by a parameterized polynomial-time reduction from the W[1]-complete Multicolored Clique problem [12]:

> **Input:** A graph $G = (V, E)$ with a proper k-coloring $c : V \mapsto \{1, \ldots, k\}$ of the vertices.
> **Question:** Is there a size-k clique in G consisting of exactly one vertex from each color class?
> Parameter: k.

First, we briefly describe the basic idea of the reduction. Construct three types of dense connected components, all of which fulfill the condition of μ-cliques.

There are k connected components of the first type, called "color components", each corresponding to a color in the Multicolored Clique instance. The components of the second type correspond to pairs of colors, called "color pair components". The components of the third type are the "vertex components": for each vertex in G, there is a vertex component. Finally, we connect the vertex component for a vertex v to the color component corresponding to $c(v)$ by $2(k-1)$ "bridges" (the exact definition of bridges will be given in the following): For each color c' with $c' \neq c(v)$, we use two bridges between the vertex component for v and the color component for $c(v)$ to encode the edges in G between v and the vertices colored by c'. Each of these bridges is sparse, that is, a bridge alone does not satisfy the condition of μ-cliques. This is the rough construction of the μ-Clique Editing instance.

The decisive trick of the reduction is the following: each color class of G corresponds to a connected component that contains a color component for this color and vertex components for the vertices with this color. This connected component is not a μ-clique due to the sparse vertex components and it is so large compared to our new parameter k', that we cannot transform it into a μ-clique by adding edges. However, cutting *exactly one* vertex component corresponding to some vertex v turns this connected component into a μ-clique. This means we have to cut $2(k-1)$ bridges connecting the vertex component for v to the color component. The construction of the bridges and the densities of the color and vertex components again force that cutting one bridge has to separate two small parts of the bridge, called "edge tails", from the color and vertex components and the bridge. An edge tail does not fulfill the condition of μ-cliques and corresponds to an edge between v and a vertex from the color class that the bridge represents. Finally, the separated edge tails are connected to the color pair components by edge insertions. The density of the color pair components allow them to be connected to exactly two edge tails which represent an edge between the two corresponding color classes. Altogether, the vertex components separated from the color components correspond to the vertices in the clique sought for in Multicolored Clique, and the edge tails separated from the vertex components and added to the color pair components correspond to the edges between the vertices in the clique.

In our reduction, we need to ensure that some of the constructed components are highly connected and have the property that adding a prespecified number of edges and vertices to these graphs results in a graph that has density *exactly* μ. In the following lemma, we show that the construction of these graphs is always possible and that it can be performed in polynomial time.

Lemma 2. *Given four positive integers a, b, c and d, where $a < b$ and $d \leq c(c-1)/2$, we can construct in $\text{poly}(a,b,c,d)$ time a graph G, such that*
− G is $2(a-1)$-connected, and
− adding c vertices and d edges to G results in a graph that has density exactly a/b and has average degree more than a.

Proof. Without loss of generality, assume that $b - a > 1$ and that $a > 1$. Otherwise, we can show the claim using $2a$ instead of a and $2b$ instead of b.

We set the number of vertices of G to $n := (2b-1)c$ and the number of edges to $m := ac(2bc-1) - d$. First, since

$$\begin{aligned} 2m < 2ac(2bc-1) &\leq 2(b-2)c(2bc-1) \\ &= (2bc-4c)(2bc-1) \\ &< (n-3c)(n+c) \\ &< n(n-1) \end{aligned}$$

there is indeed a simple graph with the claimed number of edges. Moreover, since

$$\frac{m}{n} = \frac{ac(2bc-1)-d}{(2bc-1)c} = a - \frac{d}{(2bc-1)c} > a-1,$$

G has at least $(a-1)\cdot n$ edges. This means that we can construct G in polynomial time such that it is $2(a-1)$-connected [18]. The density of the graph G' that results from adding c vertices and d edges to G is

$$\frac{2(m+d)}{(n+c)(n+c-1)} = \frac{2(ac(2bc-1)-d+d)}{(2bc-c+c)(2bc-c-1+c)} = \frac{2ac(2bc-1)}{(2bc)(2bc-1)} = \frac{a}{b}.$$

The average degree of G' follows directly. □

With this "subroutine" at hand, we can show the main theorem of this section.

Theorem 6. *For any fixed* $0 < \mu < 1$, μ-Clique Editing *is NP-complete and W[1]-hard with respect to the number* k *of allowed edge modifications.*

Proof. Let a and b be two fixed integers such that $\mu := a/b$ and assume without loss of generality that $a > n^6$ and $b > n^6$. Let C with $|C| = k$ be the set of colors in the Multicolored Clique instance $G = (V, E)$. Let $V_c := \{v \in V \mid c(v) = c\}$. Without loss of generality, assume that $\forall c \neq c'$, $|V_c| = |V_{c'}|$ and let $l := |V_c|$. In the following, we describe in detail the construction of the components of the μ-Clique Editing instance.We first describe the construction of "bridges" which are used to connect vertex and color components and then the construction of the different types of components.

Construction of Bridges. The bridges encode information about the adjacencies of G. This is done by adding tails of a specific length to the bridges. To this end, we assign to each ordered vertex pair (u, v) with $c(u) \neq c(v)$ an integer π_{uv} between 1 and x where $x = 12n^2 + 1$. Herein, the following rules should be respected:

1. No two pairs get the same number.
2. If there is an edge between u and v, then $|\pi_{uv} - \pi_{vu}| = 1$ and all numbers z with $|z - \pi_{uv}| \leq 2$ or $|z - \pi_{vu}| \leq 2$ should be reserved, that is, they should not be assigned to any other vertex pair.
3. If there is no edge between u and v, then $|\pi_{uv} - \pi_{vu}| = 2$ and all numbers z with $|z - \pi_{uv}| \leq 1$ or $|z - \pi_{vu}| \leq 1$ should be reserved.

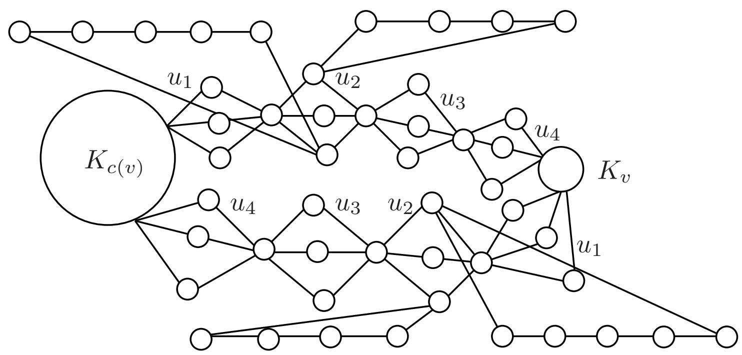

Figure 1: An example of a bridge pair between a vertex component K_v and a color component $K_{c(v)}$. These two bridges correspond to a color class c' with $c' \neq c(v)$ and $V_{c'} = \{u_1, u_2, u_3, u_4\}$. Thus, each bridge has four intervals. Here, for the sake of clarity, only three length-two paths are drawn in each interval. Note that, in the construction, there are $4k(k-1)+1$ such paths in each interval. To each of the bridges, two edge tails are added for u_2 with $\pi_{vu_2} = 5$, $x = 11$, and $y = 0$.

4. If a number i is assigned to a vertex pair or reserved, then $x - i$ should not be assigned to any vertex pair.

Observe that, since $x > 12n^2$, such an assignment is always computable in polynomial time. These numbers will be needed to identify the edges in E.

For a vertex v with color $c(v)$, we connect the vertex component K_v of v to the color component $K_{c(v)}$ with $2(k-1)$ bridges. For each color $c' \neq c(v)$ we add a pair of bridges as illustrated in Figure 1. Each bridge consists of $4k(k-1)+1$ edge-disjoint paths all of length $2l$. A bridge is divided by the common vertices of the paths into l “intervals”; each interval corresponds to a vertex in $V_{c'}$ and consists of $4k(k-1)+1$ length-two paths. The vertices lying on these paths with the exception of the two endpoints are called “middle” vertices in this interval. However, the orders of the vertices in $V_{c'}$ according to which they appear on the two corresponding bridges are reversed as shown in Figure 1. Note that all bridges have the same endpoints, one in K_v and the other in $K_{c(v)}$. Since the length of bridges can be extended to n^d for an arbitrary large constant d, we can assume that each of the bridges has density less than μ.

Next we add “edge tails” to the bridges. For each interval of a bridge which corresponds to a vertex w, we add two edge tails that are two cycles with length $y + \pi_{vw}$ and $y + x - \pi_{vw}$, respectively. Herein, y is a large integer such that each of the two cycles alone has density less than μ. This can be achieved for example by setting $y := ab$. See Figure 1 for an example of a color component and its bridges to a vertex component.

Now we describe the construction of the three types of components.

Color Pair Components. For each (unordered) pair of colors, add two color pair components K^1 and K^2, where K^1 should satisfy the following requirements:

- Adding two edges to connect two vertex-disjoint cycles that together have length *at most* $x + 2y + 1$ with K^1 results in a graph with density at least μ.
- Connecting two cycles with total length *more than* $x+2y+1$ will decrease the density below μ.

Lemma 2 shows that K^1 can be constructed in polynomial time such that adding two vertex-disjoint cycles that together have a length equal to $x+2y+1$ results in a graph that has density exactly μ. Note that by Lemma 2 and since $a > n^6$, the second part of the requirement also holds: adding a cycle of length more than $x + 2y + 1$ decreases—compared to adding a shorter cycle—the average degree of the connected component while further increasing the size of the connected component. Hence, such a component has density less than μ. The same requirement should also be fulfilled by K^2 with the length threshold of the cycles being $x + 2y - 1$.

Vertex Components. For each vertex v, add a component K_v satisfying the following requirements:

- K_v is $2(a - 1)$-connected.
- The graph that contains K_v and $\alpha := (k-1)((l-1)(4k(k-1)+x+2y)+8k(k-1)-2+x+2y)$ vertices and $\beta := (k-1)((l-1)(8k(k-1)+x+2y+2)+8k(k-1)+x+2y)$ edges of the bridges incident to K_v has density exactly μ.
- The density will decrease below μ if more than α vertices of the bridges are included.

These requirements are needed to argue that the graph resulting from disconnecting K_v from the corresponding color component $K_{c(v)}$ has density μ. By Lemma 2, we can construct in polynomial time a graph that fulfills the first two requirements. The fulfillment of the third requirement again follows from Lemma 2 and the observation that adding more bridges decreases the average degree and increases the number of vertices in the connected component, resulting in a graph that has density less than μ.

Color components. For each color c the color component K_c has to fulfill the following requirements. Herein, v is an arbitrary vertex from V_c.

- K_c is $2(a - 1)$-connected.

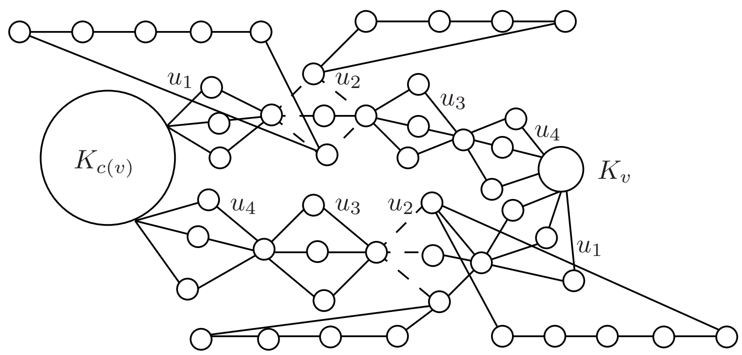

Figure 2: Illustration of the situation where the second requirement for the vertex components applies. The dashed lines represent edge deletions. Here, the two bridges between K_v and $K_{c(v)}$ are destroyed by deleting the edges in the intervals corresponding to u_2. Moreover, the middle vertices of these intervals belong to the connected component containing K_v with the only exception of two middle vertices of the first bridge. Observe that two edge tails corresponding to the vertex pair (v, u_2) are separated from K_v and $K_{c(v)}$.

- The graph that consists of 1) K_c, 2) the $l-1$ vertex components corresponding to the vertices of $V_c \setminus \{v\}$ together with all bridges that connect these vertex components with K_c, and 3) further $\gamma := (k-1)(l-1)(4k(k-1)+x+2y)$ vertices and $\delta := (k-1)(l-1)(8k(k-1)+x+2y+2)$ edges from the bridges between K_c and K_v has density *exactly* μ.
- Adding more vertices from the bridges between K_c and K_v to this graph results in a graph with density *less than* μ.

By Lemma 2, we can construct in polynomial time a graph that fulfills the first two requirements. The fulfillment of the third requirement again follows from the observation that adding more than γ vertices from the bridges between K_c and K_v decreases the average degree while increasing the number of vertices. Hence, the graph has density less than μ in this case.

We complete the construction of the μ-Clique Editing instance by setting its parameter to $k' := 2k(k-1)(4k(k-1)+3)$. Let (H, k') denote the constructed instance of μ-Clique Editing. We prove the theorem by showing the following claim:

> (G, k) is a yes-instance of Multicolored Clique $\Leftrightarrow$ (H, k') is a yes-instance of μ-Clique Editing.

$\Rightarrow$: Given a size-k clique $X := \{v_1, v_2, \ldots, v_k\}$ of the Multicolored Clique instance, we disconnect the vertex components which correspond to the vertices in X from their color components.

Observe that, in order to disconnect a vertex component from a color component, one needs $2(k-1)(4k(k-1)+1)$ edge deletions, that is, the edge deletions to cut every bridge at some interval. More precisely, to disconnect the vertex component K_{v_i} from $K_{c(v_i)}$, we cut the bridges incident to K_{v_i} at the intervals corresponding to the vertices in $X \setminus \{v_i\}$ as shown in Figure 2. Note that in Figure 2 we use two additional edge deletions to separate two edge tails from the vertex component and the color component. Altogether, we use $2(k-1)$ additional edge deletions to separate $2(k-1)$ edge tails which correspond to the edges between v_i and $X \setminus \{v_i\}$. By the construction of K_{v_i} and $K_{c(v_i)}$ we can conclude that the resulting two connected components, one containing K_{v_i} and the other containing $K_{c(v_i)}$ fulfill the condition of μ-cliques. Further, we add edges between the separated edge tails and the color pair components. According to the assignment of integers from the interval $[1, x]$ to the ordered vertex pairs, the edge tails can be pairwise put together such that each pair has a total length of either $x + 2y - 1$ or $x + 2y + 1$. Then, we connect each pair with a total length of $x+2y+1$ to one color pair component K^1 and each pair with a total length of $x+2y-1$ to one K^2. According to the construction of the color pair components, all resulting components are μ-cliques. Here, we need altogether $2k(k-1)$ edge insertions. Putting edge deletions and insertions together, we have made $2k(k-1)(4k(k-1)+1)+4k(k-1) = 2k(k-1)(4k(k-1)+3)$ edge modifications.

$\Leftarrow$: By construction, the connected components of the μ-Clique Editing instance that contain l vertex components and one color component have density less than μ. We cannot transform these connected components into μ-cliques by splitting the vertex or color components, since these components are $2(a-1)$-connected and $a > n^6 > k^4$. By the same reason, we cannot transform such a connected component into a μ-clique with at most k' edge insertions. Therefore, at least one vertex component has to be disconnected from each color component.

Since there are $2(k-1)(4k(k-1)+1)$ edge-disjoint paths between a vertex component and a color component, we need at least $2(k-1)(4k(k-1)+1)$ edge deletions for each color. This means that from a solution for (H, k') only $4k(k-1)$ modifications remain to be specified. Therefore, for at least one color, we have at most $4(k-1)$ edge modifications.

From the construction of vertex and color components, we know that when separating a vertex component K_v from a color component $K_{c(v)}$, there have to be exactly $(k-1)(x+2y)$ vertices that are separated from both the connected component that contain K_v and the connected component that contains $K_{c(v)}$. Since K_v and $K_{c(v)}$ are inseparable, these vertices come from the bridges between K_v and $K_{c(v)}$ and edge tails attached to the bridges. Note that, since each bridge consists of $4k(k-1)+1$ edge-disjoint paths, we cannot separate enough vertices by deleting at most $4(k-1)$ edges of bridges. The only choice is to separate some edge tails from the bridges. From Figure 2 we can observe that by deleting two edges we can separate at most $x + 2y$ vertices from the bridges, namely, separating the two edge tails, one having length $y + i$ and the other having length $x + y - i$, attached to the intervals whose length-two paths are destroyed while separating K_v from $K_{c(v)}$ (in Figure 2, this is the interval corresponding to u_2). Separating other edge tails or separating only a part of an edge tail requires at least two edge deletions and leaves less than $x+2y$ vertices separated. Thus, all separated edge tails come from the intervals where K_v is separated from $K_{c(v)}$.

Moreover, observe that these separated edge tails are not μ-cliques and connecting them to each other by at most $2k(k-1)$ edge insertions cannot transform them into μ-cliques, because their sizes are at least $y+1$ for a large integer y. Then, the only possibility is to connect them to the $2k(k-1)$ color pair components. This means one edge insertion for each edge tail. From the construction of color pair components, vertex-disjoint cycles with a total length at most $x+2y+1$ can be attached to K^1's, while vertex-disjoint cycles with a total length at most $x+2y-1$ can be attached to K^2's. Thus, we are forced to group the edge tails into at most $k(k-1)$ groups such that the groups can be partitioned to two same-size subsets; in one, each group has a total length at most $x+2y+1$ and, in the other, each has a total length at most $x+2y-1$. According to the assignment for the ordered vertex pairs described above, this is only possible when each group consists of two edge tails, one corresponding to the vertex pair (u,v) and the other corresponding to the vertex pair (v,u), and there is an edge between u and v in the original instance. This means that the k vertices whose corresponding vertex components are separated from the color components form a clique. □

The reduction used in the proof of Theorem 6 does not work for the edge deletion case, because we need the color pair components to check whether the separated edge tails for each pair of colors correspond to the same edge. The check requires edge insertions between edge tails and color pair components.

However, we can establish the NP-hardness of μ-Clique Deletion by a reduction from Equal-Size Clique Deletion which is NP-hard by Lemma 1.

Theorem 7. *For any fixed $0 < \mu < 1$, μ-*Clique Deletion *is NP-complete.*

Proof. Let $(G=(V,E),k,d)$ be an instance of Equal-Size Clique Deletion and let $\ell := |V|/d$ denote the size of the cliques that shall be obtained. We construct an instance of μ-Clique Deletion for $\mu = a/b$ as follows. Assume without loss of generality that $b > a > |V|^4$. We add d new graphs $G_i = (V_i, E_i)$, $1 \le i \le d$, to G. Each of these graphs is constructed such that it fulfills the following properties:

- G_i is a-connected.
- Adding ℓ vertices and $e := |V|^2 \cdot \ell + \binom{\ell}{2}$ edges to G_i results in a graph that has density exactly μ.

In analogy to the construction in the proof of Lemma 2, we can construct each G_i in poly(a,b,ℓ,e) time such that it contains $n := (2b-1)\ell$ vertices and $m := a\ell(2b\ell-1)-e$ edges, and fulfills both requirements.

For each G_i, we choose an arbitrary subset of $|V|^2$ vertices $S_i \subset V_i$ and insert an edge between every vertex in S_i and every vertex in V. Let G' denote the graph that results from this construction. We set the number of allowed edge deletions to $k' := |V|^3 \cdot (d-1) + k$. It remains to show the following:

(G, k) is a yes-instance of Equal-Size Clique Deletion $\Leftrightarrow$ (G', k') is a yes-instance of μ-Clique Deletion.

$\Rightarrow$: Let $S \subseteq E$ be a size-k edge set whose removal transforms G into a graph that consists of d vertex-disjoint cliques K_i, $1 \leq i \leq d$, each of size ℓ. A size-k' solution for the μ-Clique Deletion instance (G', k') can be constructed by performing the edge deletions of S to $G'[V]$ and removing for each V_i the edges between V_i and $V \setminus K_i$. Let G'' be the graph that is obtained from G' by these edge deletions. Observe that G'' consists of d connected components whose vertex sets are exactly $K_i \cup V_i$, $1 \leq i \leq d$. The constructed solution has size $|V|^3 \cdot (d-1) + k = k'$ since we first perform k edge deletions in $G'[V]$ and then for each vertex $v \in V$ we perform $(d-1) \cdot |V|^2$ edge deletions (v is "cut" from $d-1$ of the G_i's and has exactly $|V|^2$ neighbors in each G_i). Each of the connected components of G'' has density exactly μ, which can be seen as follows: Each K_i is a size-ℓ clique and there are $|V|^2 \cdot \ell$ edges between V_i and K_i. The density of $G''[K_i \cup V_i]$ then follows directly from the second requirement of the construction of G_i.

$\Leftarrow$: Let S be a size-k' edge set whose removal transforms G' into a μ-clique-cluster graph G''. First, note that each of the G_i's is a-connected and that $k' < |V|^4 < a$. Therefore, each of the G_i's is completely contained in one connected component of G''. Second, a connected component that contains two or more G_i's is not μ-dense which can be shown as follows. Let G^* be a connected component of G'' that contains $t > 1$ of the G_i's and x vertices of V. The number of edges in G^* is at most $t \cdot (m + |V|^2 \cdot x) + \binom{x}{2}$ and the number of vertices in G^* is exactly $t \cdot n + x$. The density of G^* is thus at most

$$\begin{aligned}
\frac{2t \cdot (m + |V|^2 \cdot x) + 2\binom{x}{2}}{(t \cdot n + x)(t \cdot n + x - 1)} &\overset{(*)}{<} \frac{2 \cdot (m + |V|^2 \cdot x + \binom{x}{2})}{(2n) \cdot n} \\
&= \frac{2 \cdot (a\ell(2b\ell - 1) - e + |V|^2 \cdot x + \binom{x}{2})}{2(2b-1)\ell \cdot (2b-1)\ell} \\
&\overset{(**)}{<} \frac{2 \cdot (a\ell(2b\ell))}{2(2b-1)\ell \cdot (2b-1)\ell} \\
&\overset{(***)}{<} \frac{2a \cdot (\ell + 1)}{2(2b-1)\ell} \\
&< \frac{a \cdot (\ell+1)}{(2b-1)\ell} < \frac{3a}{4b-2} < \frac{a}{b} = \mu
\end{aligned}$$

Inequality $(*)$ follows from $t > 1$ and $x \geq 1$ (which can be assumed since there are no edges between the G_i's). Inequality $(**)$ follows from $a\ell + e > |V|^2 \cdot x + \binom{x}{2}$. Inequality $(***)$ follows from $2b > \ell + 1$. It thus follows that G^* has density less than μ if it contains more than one of the G_i's. Hence, each connected component of G'' contains at most one of the G_i's. Therefore, each vertex $v \in V$ is in G'' adjacent to vertices of at most one G_i. In other words, for each $v \in V$ we need to delete the edges to at least $d-1$ of the G_i's. Overall, this amounts to at least $(d-1) \cdot |V|^3$ edge deletions. This, however, means that there cannot be a vertex $v \in V$ that is cut from *all* G_i's: then at least $(d-1) \cdot |V|^3 + |V|^2$ are necessary, which cannot be afforded since $k < |V|^2$. In summary, each connected component of G'' contains exactly one of the G_i's.

We now show that each connected component contains exactly ℓ vertices from V and that these vertices form a clique. First, if there is a connected component that contains more than ℓ vertices from V, then this connected component has density less than μ since compared to the graph that contains G_i and ℓ vertices from V (which has, by the second requirement its construction, density exactly μ if the ℓ vertices form a clique) we add vertices that have lower average degree (which reduces the density). Second, if the ℓ vertices from V do not form a clique, then the number of edges that are added to G_i is less than d. By the second requirement of the construction of G_i, the density of the connected component is then less than μ. Hence, we can conclude that each of the connected components of G'' contains ℓ vertices from V that form a clique. By observing that the number of edge deletions that are performed in $G'[V]$ is at most k, it follows that (G, k) is a yes-instance of Equal Size Clique Deletion. □

5 Outlook

There are numerous topics for future research, we only point out some of them. For s-Defective Clique Editing and Average-s-Plex Editing clearly further algorithmic improvements are necessary. For instance, s-Defective Clique Editing is still missing a nontrivial kernelization algorithm, whereas for Average-s-Plex Editing a solution algorithm other than the brute-force one that can be applied to the reduced instance is needed. As a next step, experimental studies should then be undertaken regarding the running time of the algorithms and the quality of the produced clusterings. For μ-Clique Editing, other parameterizations should be studied. Moreover, the parameterized complexity of μ-Clique Deletion remains open. Finally, it is also interesting to consider other density measures that are useful in practice, and study the classical and parameterized complexity of the cluster editing problem with respect to these measures.

Acknowledgment We thank the anonymous referees of *Algorithmica* for several comments that have improved the presentation of this work.

References

[1] J. Abello, M. G. C. Resende, and S. Sudarsky. Massive quasi-clique detection. In *Proceedings of the 5th Latin American Symposium on Theoretical Informatics (LATIN '02)*, volume 2286 of *Lecture Notes in Computer Science*, pages 598–612. Springer, 2002.

[2] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information:

Ranking and clustering. *Journal of the ACM*, 55(5), 2008. Article 23 (October 2008), 27 pages.

[3] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56(1–3):89–113, 2004.

[4] S. Böcker, S. Briesemeister, Q. B. A. Bui, and A. Truß. Going weighted: Parameterized algorithms for cluster editing. *Theoretical Computer Science*, 410(52): 5467–5480, 2009.

[5] L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996.

[6] Y. Cao and J. Chen. Weighted cluster editing: Kernelization based on edge-cuts. In *Proceedings of the 5th International Symposium on Parameterized and Exact Computation (IPEC '10)*, Lecture Notes in Computer Science. Springer, 2010.

[7] J. Chen and J. Meng. A $2k$ kernel for the cluster editing problem. In *Proceedings of the 16th Annual International Conference on Computing and Combinatorics (COCOON '10)*, volume 6196 of *Lecture Notes in Computer Science*, pages 459–468. Springer, 2010.

[8] E. J. Chesler, L. Lu, S. Shou, Y. Qu, J. Gu, J. Wang, H. C. Hsu, J. D. Mountz, N. E. Baldwin, M. A. Langston, D. W. Threadgill, K. F. Manly, and R. W. Williams. Complex trait analysis of gene expression uncovers polygenic and pleiotropic networks that modulate nervous system function. *Nature Genetics*, 37 (3):233–242, 2005.

[9] F. K. H. A. Dehne, M. A. Langston, X. Luo, S. Pitre, P. Shaw, and Y. Zhang. The cluster editing problem: Implementations and experiments. In *Proceedings of the 2nd International Workshop on Parameterized and Exact Computation (IWPEC '06)*, volume 4169 of *Lecture Notes in Computer Science*, pages 13–24. Springer, 2006.

[10] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.

[11] M. R. Fellows, M. A. Langston, F. A. Rosamond, and P. Shaw. Efficient parameterized preprocessing for Cluster Editing. In *Proceedings of the 16th International Symposium on Fundamentals of Computation Theory (FCT '07)*, volume 4639 of *Lecture Notes in Computer Science*, pages 312–321. Springer, 2007.

[12] M. R. Fellows, D. Hermelin, F. A. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009.

[13] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[14] J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Graph-modeled data clustering: Exact algorithms for clique generation. *Theory of Computing Systems*, 38 (4):373–392, 2005.

[15] D. L. Greenwell, R. L. Hemminger, and J. B. Klerlein. Forbidden subgraphs. In *Proceedings of the 4th Southeastern Conference on Combinatorics, Graph Theory and Computing*, pages 389–394, 1973.

[16] J. Guo. A more effective linear kernelization for Cluster Editing. *Theoretical Computer Science*, 410(8–10):718–726, 2009.

[17] J. Guo, C. Komusiewicz, R. Niedermeier, and J. Uhlmann. A more relaxed model for graph-based data clustering: s-plex cluster editing. *SIAM Journal on Discrete Mathematics*, 24(4):1662–1683, 2010.

[18] F. Harary. The maximum connectivity of a graph. *Proceedings of the National Academy of Science of the United States of America*, 48(7):1142–1146, 1962.

[19] S. Kosub. Local density. In *Network Analysis*, volume 3418 of *Lecture Notes in Computer Science*, pages 112–142. Springer, 2004.

[20] M. Křivánek and J. Morávek. NP-hard problems in hierarchical-tree clustering. *Acta Informatica*, 23(3):311–323, 1986.

[21] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[22] S. B. Seidman and B. L. Foster. A graph-theoretic generalization of the clique concept. *Journal of Mathematical Sociology*, 6:139–154, 1978.

[23] R. Shamir, R. Sharan, and D. Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144(1–2):173–182, 2004.

[24] H. Yu, A. Paccanaro, V. Trifonov, and M. Gerstein. Predicting interactions in protein networks by completing defective cliques. *Bioinformatics*, 22(7):823–829, 2006.

[25] A. van Zuylen and D. P. Williamson. Deterministic pivoting algorithms for constrained ranking and clustering problems. *Mathematics of Operations Research*, 34(3):594–620, 2009.