MASTER

Efficient Kernels for Q-Coloring

Schalken, M.M.R.

Award date:
2020

Technische Universiteit Eindhoven
Department of Mathematics and Computer Science

Efficient Kernels for Q-Coloring

Master Thesis

M.M.R. Schalken
m.m.r.schalken@student.tue.nl
0807486

Supervisor:
dr. B.M.P. Jansen

Committee:
dr. B.M.P. Jansen
dr. K.A. Buchin
dr. K.A.B. Verbeek

Eindhoven, August 4, 2020

Abstract

With parameterized q-Coloring problems, the existence of a polynomial kernel is dictated by which parameter is chosen. In recent results it was found that using the size of a minimum vertex cover as parameter, q-Coloring admits a kernel of bitsize $\mathcal{O}(k^{q-1} \log k)$ and the problem does not admit a kernel of bitsize $\mathcal{O}(k^{q-1-\varepsilon})$ for any $\varepsilon > 0$ unless $coNP \subseteq NP/poly$. This paper analyzes if there are parameters that are strictly smaller than Vertex Cover for which the q-Coloring problem still admits polynomial kernel. For q-Coloring parameterized by the size k of a vertex set whose removal splits the graph into isolated vertices and independent edges, we present a polynomial kernel with $\mathcal{O}(k^{2q-2})$ vertices. We also show that this problem does not admit a kernel of bitsize $\mathcal{O}(k^{2q-3-\varepsilon})$ for any $\varepsilon > 0$ unless $coNP \subseteq NP/poly$. We then also show how for $q = 3$ and $q = 4$ we can find a kernel with $\mathcal{O}(k^{2q-3})$ vertices which can be encoded with $\mathcal{O}(k^{2q-3} \log k)$ bits. We show how the difficulty of generalizing this kernel for $q \geq 5$ is reduced to finding polynomial equalities that adhere to a set of defined properties. We furthermore show that we can reduce our parameter further by presenting a kernel with $\mathcal{O}(k^{q^2})$ vertices for q-Coloring parameterized by the size k of a vertex set whose removal splits the graph into disjoint cliques.

1 Introduction

In graph theory q-Coloring is a well-known algorithmic problem. Given an input graph it asks if the graph can be *properly colored* with q colors. A *proper coloring* consists of a coloring function f that assigns one of the available q colors to each vertex in the graph, such that if two vertices are adjacent in the graph, they do not get assigned the same color. q-Coloring is well-known to be *NP-hard* for all $q \geq 3$ [7]. This means that there is no exact algorithm for q-Coloring that runs in time bounded by a polynomial in the size of the input graph, unless $P = NP$ [2, Ch. 34].

With graph coloring being a well-known NP-hard problem, it is well positioned to be studied within the subject of *Parameterized Algorithms* [3]. With parameterized algorithms we extend an algorithm's input with a parameter k. We then ask the same question, but we want to find an algorithm that has a time-bound of the shape $f(k) \cdot n^{\mathcal{O}(1)}$. The main idea is that this parameter is chosen such that it captures how complex the given input is. The result would then be an algorithm that grows in running-time polynomially based on the size of the input and only non-polynomially based on the chosen complexity parameter. All the parameterized problems for which such an algorithm exists, form the collection of *Fixed Parameter Tractable (FPT)* problems.

For the parameterized version of q-Coloring, the complexity is dictated by which parameter is chosen. For example if we pick our parameter to be the amount of available colors, the problem will not be in FPT. If we would have an FPT algorithm for this, then we could look at 3-Coloring and have a polynomial time algorithm, which would imply $P = NP$. Because of this, a common research question is for which parameters the problem is in FPT and for which it is not. This can be proven by either directly proving the existence of an FPT algorithm for the problem, or by showing that there is a different parameter that is always smaller or equal to your chosen parameter for which it is already proven that the problem is in FPT.

In a 2012 paper by Doucha and Kratochvíl [5], it is shown that q-Coloring is in FPT with parameter *Unbounded Cluster Vertex Deletion*, the smallest set of vertices such that after deletion of the vertices, the remainder of the graph is a disjoint union of complete graphs. As such we know that for any parameter for which we can show that it is always greater or equal to *UCVD*, then we know that the problem is in FPT. This paper furthermore discusses the parameter *c-Bounded Cluster Vertex Deletion*, the size of a vertex deletion set such that the remainder is a disjoint union of complete graphs where each clique has size at most c. In our research we used the parameter *UCVD* and $BCVD_2$, although we are using different terminology.

Furthermore, we can pose the question for which parameters there is a *kernelization* for the problem. Kernelizations are a specific type of pre-processing algorithms that provide a polynomial time reduction of the given input, such that the reduced graph is bounded in size by a function of the parameter k. If this size bounding function is a polynomial, then we call this a *polynomial kernel*. This type of pre-processing can be combined with any exact solving algorithm on the reduced graph to get a complete algorithm that is bounded in time polynomial in the size of the original input and non-polynomial in the size of the parameter, thus giving an FPT algorithm. Because of this we know that if for a given parameter, there exists a kernelization for q-Coloring, then q-Coloring with this parameter is also in FPT. This does not always hold in the other direction. More specifically for all problems in FPT there exists a kernelization, however not for all these problems there exists a *polynomial* kernel. There are parameters for which it is proven that q-Coloring is in FPT, but it does not admit a polynomial kernel unless $coNP \subseteq NP/poly$. Note that for all problems in FPT, there exists a kernelization. For some problems in FPT however a kernelization is only possible with a size bound that is superpolynomial.

In selecting parameters we will use the notation of q-Coloring on $\mathcal{F} + kv$. Here $\mathcal{F}$ is a family of graphs, for example *Independent* graphs (graphs without edges), and $\mathcal{F} + kv$ is the graph family formed by all graphs for which there is a set $X \subset V(G)$ of size at most k, such that $G - X$ is in $\mathcal{F}$. We call such a subset X a modulator to $\mathcal{F}$ for the graph G. Note that this is mostly a different way to define the same parameter. As an example, q-Coloring on Independent $+ kv$ is equivalent to q-Coloring with as parameter the size of a *vertex cover*, which is also equivalent to 1-*Bounded Cluster Vertex Deletion* from [5]. In working with this definition to find kernelizations, we will assume that such a modulator X is provided along with the input. This allows us to decouple the difficulty of finding such a modulator from the difficulty of finding a kernelization. For all parameters that we will consider there are polynomial-time constant-factor approximation algorithms. As such if we find a polynomial kernelization, then we can use this kernelization by first running such an approximation algorithm on the input to find a modulator for our kernelization algorithm. Thus, we can safely assume a modulator to be part of the input.

A paper by Jansen and Kratsch [8] introduces a hierarchical classification of parameters for q-Coloring. This hierarchy depends on certain parameters always being smaller or equal to another parameter and as such creating an hierarchical relation. The paper also defines sets within this hierarchy of parameters for all problems that are FPT and all problems that admit a polynomial kernel. In this hierarchy it is shown that q-Coloring with parameter *Vertex Cover*, equivalent to q-Coloring on Independent $+ kv$ admits a kernel of $\mathcal{O}(k^q)$ vertices. Furthermore, the paper shows that 3-coloring on LinearForest $+ kv$ graphs does not admit a polynomial kernel unless $coNP \subseteq NP/poly$. Here the LinearForest graph class is all graphs that are a disjoint union of paths, in other words an acyclic graph where each vertex has at most degree 2.

In a work by Jansen and Pieterse [9], an improvement on the $\mathcal{O}(k^q)$ vertices kernel for q-Coloring on Independent + kv is introduced. The paper shows a kernel of $\mathcal{O}(k^{q-1})$ vertices and shows that this can be encoded in $\mathcal{O}(k^{q-1} \log k)$ bits. This gives a factor k improvement on the previous kernel. Furthermore, the paper shows that a kernel of $\mathcal{O}(k^{q-1-\varepsilon})$ bits does not exist for $\varepsilon > 0$ unless $coNP \subseteq NP/poly$. Thus, the paper matches the upper and lower bound for kernels for q-Coloring on Independent + kv to $k^{o(1)}$ factors.

Notably all the lower-bound results come with the assumption that $coNP \not\subseteq NP/poly$. Similar to the assumption $P \neq NP$, this assumption is generally accepted to hold [4].

The scope of this thesis will be to show two smaller parameters compared to q-Coloring on Independent + kv and show that these do still admit polynomial kernels. Furthermore, we will show a lower-bound on possible kernel-size for one of the parameters.

The first parameter that we introduce is q-Coloring on Independent Edge + kv. Here the Independent Edge graph family is all graphs where every connected component is either an independent vertex or a single connected vertex pair. Our first result is that q-Coloring on Independent Edge + kv admits a kernel bounded in size by $\mathcal{O}(k^{2q-2})$ vertices. We show how we define our reduced graph G' by starting with the subgraph defined by the modulator X. We then look for every set of q vertices in the modulator if all q vertices were adjacent to at-least one vertex in the remainder of our original graph. If they were then we add a new vertex to our reduced graph G' to make sure that there will be at least one color available for the vertices in the remainder of our original graph. We then look at every 2 sets of $q-1$ vertices in the modulator to see if we can find an adjacent vertex pair $\{u, v\}$ in the remainder such that the entire first set is adjacent to u and the second set to v. We then add a new adjacent vertex pair in our reduced graph such that there will be different colors available for u and v.

Our second result shows that q-Coloring on Independent Edge + kv does not admit a kernel of size $\mathcal{O}(k^{2q-3-\varepsilon})$ bits for all $\varepsilon > 0$ unless $coNP \subseteq NP/poly$. This is achieved by showing two linear-parameter transformations; from q-cnf-sat to $(q+1)$-nae-sat with common last literal, and from $(2q-2)$-nae-sat with common last literal to q-Coloring on Independent Edge + kv. By combination of these transformations we can show how any $(2q-3)$-cnf-sat problem can be transformed into a q-Coloring on Independent Edge + kv problem. This gives us that if a kernel of $\mathcal{O}(k^{2q-3-\varepsilon})$ bits exists for q-Coloring on Independent Edge + kv, then there exists a *generalized* kernel of $\mathcal{O}(n^{q-\varepsilon})$ bits for q-cnf-sat parameterized by n, the number of variables. It is well-known that q-cnf-sat admits no kernel of size $\mathcal{O}(n^{q-\varepsilon})$ bits for $\varepsilon > 0$ and $q \geq 3$ unless $CoNP \subseteq NP/poly$ [4].

Thirdly we will introduce our second parameter to define q-Coloring on Clique + kv. Here the Clique graph family consists of all graphs where every connected component forms a clique. Notably this parameter is equivalent to the *UCVD* parameter from [5]. We will show that this problem admits a kernel bounded in size by $\mathcal{O}(k^{q^2})$ vertices. This result was achieved using the same method as we used for q-Coloring on Independent Edge + kv. We iteratively look for vertex sets in the modulator and see if there is a clique in their common neighborhood in the remainder of the graph that could lead to a coloring conflict when extending a coloring from our reduced graph to the original. If such a node exists then we add a new clique of vertices to our reduced graph to prevent such conflicts from occurring.

For our fourth result, we will show that for q-COLORING on INDEPENDENT EDGE + kv for $q = 3$ and $q = 4$ it is possible to make a smaller kernel to get closer to our lower bound. To achieve this we use a tactic that was shown in the earlier mentioned paper by Jansen and Pieterse [9]. We create polynomial equalities that given a partial coloring of the modulator can be evaluated such that if all equalities are satisfied, then the coloring can be extended to the original graph. We then use a theorem for d-POLYNOMIAL ROOT CSP to reduce polynomial equalities to a (sub)set of equalities such that the amount of equalities is bounded by $f(k)$ for a function f. We then create a reduced graph with the modulator with vertices added back for the reduced set of equalities. This gives us a reduced graph that is limited in size by $\mathcal{O}(k^{2q-3})$ vertices which we can encode with $\mathcal{O}(k^{2q-3} \log k)$ bits. The main difficulty in this approach remains to find the proper polynomial equalities that preserve the constraints that our remainder graph puts on valid colorings. We present such polynomials for $q = 3$ and $q = 4$, but we were not able to find nor compute polynomials for $q \geq 5$.

Related Work

Bodlaender, Jansen, and Kratsch [1] introduced a framework for proving kernelization lower bounds. Furthermore, they use this framework to prove lower bounds for multiple parameterized problems. One key result for us is that they showed that q-COLORING on INDEPENDENT + kv with q not as a fixed constant does not admit a polynomial kernel. This is why for our research we always consider q to be a fixed constant and as such it is allowed to appear in the exponent of our size bounds.

In a 2011 paper by Fiala, Golovach, and Kratochvíl [6], coloring problems are compared with regards to *Vertex Cover* and *Treewidth* as parameter. In this comparison it is shown for multiple variations of coloring problems if they are in FPT or not with regards to these two parameters.

2 Preliminaries

We will use $[q]$ to denote the set of integers 1 to q for some natural number q, thus $[q] := \{1, \ldots, q\}$. We will write $x \equiv_\ell y$ for $x, y \in \mathbb{Z}$, to denote that x and y are congruent modulo ℓ. For some finite set X and some integer k, let $\binom{X}{k}$ denote the collection of all subsets of X of size exactly k. A graph G consists of a vertex set $V(G)$ and an edge set $E(G)$. We assume all graphs to be simple and undirected. We denote the neighborhood, all vertices adjacent, of a vertex $v \in V(G)$ by $N_G(v) := \{u \in V(G) \mid \{u, v\} \in E(G)\}$. Note that we use this notation to denote the *open neighborhood*, all neighbors of v with v not included. We let $G[X]$ for $X \subseteq V(G)$ denote the subgraph of G induced by X, thus $V(G[X]) := X$ and $E(G[X]) := \{\{u, v\} \mid \{u, v\} \in E(G) \wedge u, v \in X\}$. For $X \subseteq V(G)$, we use $G - X$ to denote the graph obtained from G by deleting all vertices of X and their incident edges. A *proper q-coloring* of G is a function $f \colon V(G) \to [q]$ such that for all $\{u, v\} \in E(G)$ it holds that $f(u) \neq f(v)$.

A *parameterized problem* $\mathcal{Q}$ is a subset of $\Sigma^* \times \mathbb{N}$ where Σ is a finite alphabet. Here $\mathbb{N}$

is the parameter for problem which captures some complexity measure of the input. A parameterized problem is *Fixed Parameter Tractable* if there exists an algorithm that decides if $(x, k) \in \mathcal{Q}$ in time $f(k)|x|^{\mathcal{O}(1)}$ for some computable function f.

Let $\mathcal{Q}, \mathcal{Q}' \subseteq \Sigma^* \times \mathbb{N}$ be parameterized problems and let $h\colon \mathbb{N} \to \mathbb{N}$ be a computable function. A *generalized kernel* for $\mathcal{Q}$ into $\mathcal{Q}'$ of size $h(k)$ is an algorithm that given an input $(x, k) \in \Sigma^* \times \mathbb{N}$, takes time polynomial in $|x| + k$ and outputs an instance (x', k') such that both:

1. $|x'|$ and k' are bounded by $h(k)$.
2. $(x', k') \in \mathcal{Q}'$ if and only if $(x, k) \in \mathcal{Q}$.

The algorithm is a kernel for $\mathcal{Q}$ if $\mathcal{Q} = \mathcal{Q}'$. It is a *polynomial kernel* if $h(k)$ is a polynomial function.

Given two parameterized problems $\mathcal{P}$ and $\mathcal{Q}$, we say that $\mathcal{P}$ is *polynomial-parameter reducible* to $\mathcal{Q}$, if there exists a polynomial time computable function $f\colon \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$ and a polynomial $p\colon \mathbb{N} \to \mathbb{N}$, such that for all $(x, k) \in \Sigma^* \times \mathbb{N}$ it holds that both:

1. $(x, k) \in \mathcal{P}$ if and only if $(x', k') = f(x, k) \in \mathcal{Q}$.
2. $k' \leq p(k)$.

We call the function f a *polynomial-parameter transformation.* If p is a linear function then f is a *linear-parameter transformation.*

In this paper we will consider as parameters the number of vertices that need to be removed from the input graph, such that the remainder of the graph is part of a specified family of graphs. We define the generalized definition as such.

q-COLORING ON $\mathcal{F} + kv$ GRAPHS **Parameter:** $k := |X|$

Input: A graph G and a modulator $X \subseteq V(G)$ such that $G - X \in \mathcal{F}$

Question: Is there a function $f\colon V(G) \to [q]$ such that for all $\{u, v\} \in E(G)\colon f(u) \neq f(v)$.

In this paper we will use the q-LIST-COLORING problem to prove some properties when trying to extend a proper partial coloring to the rest of the graph. We will first give a proper definition for the problem.

q-LIST-COLORING

Input: A graph G and for every vertex $v \in V(G)$ a list $L(v) \subseteq [q]$ of allowed colors.

Question: Is there a function $f\colon V(G) \to [q]$ such that for all $\{u, v\} \in E(G)\colon f(u) \neq f(v)$ and for all $v \in V(G)\colon f(v) \in L(v)$.

For any partial coloring f for q-COLORING, we can define extending this coloring to the rest of the graph as a q-LIST-COLORING problem. If we are given a q-COLORING problem with input graph G and a partial coloring of $G[S]$ for some $S \subseteq V(G)$. We can define a q-LIST-COLORING instance (G', L) by defining $G' := G - S$ and $L(v) := [q] \setminus \{f(u) \mid u \in N_G(v) \cap S\}$. Answering if we can extend f to the entire graph G is then equivalent to answering the q-LIST-COLORING instance (G', L).

Later in the paper we will use a theorem for POLYNOMIAL ROOT CSP, thus we will properly state the definition of the problem here for completeness. There is a different version of this problem defined for each efficient field $\mathcal{F}$. The mathematical structure of a field is called an efficient field if the field operations (addition, subtraction, multiplication, division) can be performed in polynomial time in the length of a reasonable encoding of the input.

d-POLYNOMIAL ROOT CSP OVER $\mathcal{F}$
Input: A list L of polynomial equalities and a set of variables $V := \{x_1, \ldots, x_n\}$. Equalities are of form $f(x_1, \ldots, x_n) \equiv_{\mathcal{F}} 0$, where f is a multivariate polynomial with max degree d and $\equiv_F$ denotes congruent over the field $\mathcal{F}$.
Question: Is there an assignment $\tau\colon V \to \{0, 1\}$ that satisfies all equalities in L.

3 q-Coloring on Independent Edge + kv

We introduce the graph class INDEPENDENT EDGE of graphs where each connected component has at-most one edge. Note that INDEPENDENT $\subset$ INDEPENDENT EDGE. As such also INDEPENDENT + kv $\subset$ INDEPENDENT EDGE + kv. We will start by showing that there exists a polynomial kernel for the q-COLORING on INDEPENDENT EDGE + kv problem with size of $\mathcal{O}(k^{2q-2})$ vertices.

Lemma 1. *q-COLORING on INDEPENDENT EDGE + kv admits a polynomial kernel with $\mathcal{O}(k^{2q-2})$ vertices for every $q \geq 3$.*

Proof. Let (G, X) be an instance of q-COLORING on INDEPENDENT EDGE + kv. We create an equivalent instance (G', X) in the following steps:

- Set $G' := G[X]$.
- For every $S \in \binom{X}{q}$, if $\bigcap_{v \in S} N_G(v) \setminus X \neq \emptyset$ add v_S to G' with $N_{G'}(v_S) := S$.
- For every $S_1 \in \binom{X}{q-1}$, for every $S_2 \in \binom{X}{q-1}$, if there exists a vertex $u \in \bigcap_{v \in S_1} N_G(v) \setminus X$ and a vertex $w \in \bigcap_{v \in S_2} N_G(v) \setminus X$ with $(u, w) \in E(G)$, then add u_S and w_S to G' with $N_{G'}(u_S) := S_1 \cup \{w_S\}$ and $N_{G'}(w_S) := S_2 \cup \{u_S\}$.

To show that (G', X) is equivalent to (G, X) we show that a proper q-coloring f' of G' implies the existence of a proper coloring of G. We do this by showing how if given a proper q-coloring f' of G' we can from this construct a proper coloring f of G.

Given a proper coloring f' of G' we first extract a proper partial coloring f of $G[X]$. We will now show how we extend f to the entire graph of G by showing how we can color each connected component in $G - X$. Since $G - X \in$ INDEPENDENT EDGE we know that each connected component has at-most size 2. We will do a case distinction on the size of the connected component. For each connected component $C \in G - X$ we do the following:

(a) If $|C| = 1$, then let C be the single vertex v. Since v has no neighbors in C, the entire neighborhood of v is in X and is thus already colored. Let $S := \{f(u) \mid u \in N_G(v)\}$ be the set of colors used by the neighbors of v. If $[q] \setminus S \neq \emptyset$, then let $i \in [q] \setminus S$ and set $f(v) := i$. Since there is no neighbor of v that uses color i, this is a proper extension. If $[q] \setminus S = \emptyset$ then there is a set of vertices $V_v = \{v_1, \ldots, v_q\} \subseteq N_G(v)$ with $f(v_i) = i$ for $i \in [q]$. Since $|V_v| = q$ and $V_v \subseteq X$, we know that $V_v \in \binom{X}{q}$. Since all vertices in V_v are adjacent to v we have that $\bigcap_{x \in V_v} N_G(x) \setminus X \neq \emptyset$ and as such there is a $x_s \in G'$ with $N_{G'}(x_s) = V_v$. Since for all $v_i \in V_v\colon f(v_i) = f'(v_i) = i$ and there is some color c for which $f'(x_s) = c$ we know that $f'(x_s) = f'(v_c) = c$ while $\{x_s, v_c\} \in E(G')$, thus contradicting that f' is a proper coloring.

(b) If $|C| = 2$, then let C be the adjacent vertex pair u, v. Since u and v only have each other as neighbors in C and the rest of their neighborhood is in X we know that their entire neighborhood except for each other is already colored. Let $S_u := \{f(x) \mid x \in N_G(u) \setminus \{v\}\}$ and $S_v := \{f(x) \mid x \in N_G(v) \setminus \{u\}\}$.

If $[q] \setminus S_u = \emptyset$, there is a set of vertices $V_u = \{u_1, \ldots, u_q\} \subseteq N_G(u) \setminus \{v\}$ with $f(u_i) = i$ for $i \in [q]$. Since $|V_u| = q$ and $V_u \subseteq X$, we know that $V_u \in \binom{X}{q}$. Since all vertices in V_u are adjacent to u we have that $\bigcap_{x \in V_u} N_G(x) \setminus X \neq \emptyset$ and as such there is a $x_s \in G'$ with $N_{G'}(x_s) = V_u$. Since for all $u_i \in V_u\colon f(u_i) = f'(u_i) = i$ and there is some color c for which $f'(x_s) = c$ we know that $f'(x_s) = f'(u_c) = c$ while $\{x_s, u_c\} \in E(G')$, thus contradicting that f' is a proper coloring.

If $[q] \setminus S_v = \emptyset$, there is a set of vertices $V_v = \{v_1, \ldots, v_q\} \subseteq N_G(v) \setminus \{u\}$ with $f(v_i) = i$ for $i \in [q]$. Since $|V_v| = q$ and $V_v \subseteq X$, we know that $V_v \in \binom{X}{q}$. Since all vertices in V_v are adjacent to v we have that $\bigcap_{x \in V_v} N_G(x) \setminus X \neq \emptyset$ and as such there is a $x_s \in G'$ with $N_{G'}(x_s) = V_v$. Since for all $v_i \in V_v\colon f(v_i) = f'(v_i) = i$ and there is some color c for which $f'(x_s) = c$ we know that $f'(x_s) = f'(v_c) = c$ while $\{x_s, v_c\} \in E(G')$, thus contradicting that f' is a proper coloring.

As such we know there is $i \in [q] \setminus S_u$ and $j \in [q] \setminus S_v$ such that if we assign color i to u then there is no neighbor of u in X that also uses color i and equally if we assign color j to v, then there is no neighbor of v in X that also uses color j. Thus, $f(u) = i$ and $f(v) = j$ would be a proper coloring unless $i = j$. If there is no $i \in [q] \setminus S_u$ and $j \in [q] \setminus S_v$ such that $i \neq j$, then it has to be that $[q] \setminus S_u = [q] \setminus S_v = \{i\}$ for some color i. Since the order of the colors is arbitrary, we can choose i to be the q-th color and as such reference it as q. There then is a set of vertices $V_u = \{u_1, \ldots, u_{q-1}\} \subseteq N_G(u) \setminus \{v\}$ with $f(u_i) = i$ for $i \in [q-1]$. There is also a set of vertices $V_v = \{v_1, \ldots, v_{q-1}\} \subseteq N_G(v) \setminus \{u\}$ with $f(v_i) = i$ for $i \in [q-1]$. Since $|V_u| = |V_v| = q - 1$ and $V_u, V_v \subseteq X$, we know that $V_u, V_v \in \binom{X}{q-1}$. Since all vertices in V_u are adjacent to u and all vertices in V_v are adjacent to v, we have that $u \in \bigcap_{x \in V_u} N_G(x) \setminus X$ and $v \in \bigcap_{x \in V_v} N_G(x) \setminus X$. As such there are x_u and x_v in G' with $N_{G'}(x_u) = V_u \cup \{x_v\}$ and $N_{G'}(x_v) = V_v \cup \{x_u\}$. Since for all $u_i \in V_u\colon f(u_i) = f'(u_i) = i$ and there is some color c_1 for which $f'(x_u) = c_1$. Since f' is a proper coloring, then it has to hold that $c_1 = q$. Furthermore, for all $v_i \in V_v\colon f(v_i) = f'(v_i) = i$ and there is some color c_2 for which $f'(x_v) = c_2$. Since f' is a proper coloring, then it has to hold that $f'(x_u) \neq f'(x_v)$ thus $c_2 \neq q$. Therefore we know that $f'(x_v) = c_2 = f(v_{c_2})$ while $\{x_v, v_{c_2}\} \in E(G')$, thus contradicting that f' is a proper coloring. As such we know that there always is $i \in [q] \setminus S_u$ and $j \in [q] \setminus S_v$ such that $i \neq j$ and we can set $f(u) = i$ and $f(v) = j$.

Thus, by showing how we extend a coloring f' of G' to a coloring f of G, we have shown that

(G', X) is a YES-instance $\Rightarrow (G, X)$ is a YES-instance.

For the other direction it rather straightforward that (G', X) is a YES-instance $\Leftarrow (G, X)$ is a YES-instance. Given a proper coloring f of G we can extract a proper partial coloring f' of $G'[X]$ since $G'[X] \subseteq G$. For every v_s added to G' with $N_{G'}(v_s) = S$ there was a vertex $x \in G - X$ with $S \subseteq N_G(x)$, as such we can properly extend f' with $f'(v_s) = f(x)$. For every adjacent vertex pair u_s, w_s added to G' with $N_{G'}(u_s) = S_1 \cup \{w_s\}$ and $N_{G'}(v_s) = S_2 \cup \{u_s\}$, there was an adjacent vertex pair $x_1, x_2 \in G - X$ with $S_1 \cup \{x_2\} \subseteq N_G(x_1)$ and $S_2 \cup \{x_1\} \subseteq N_G(x_2)$, as such we can properly extend f' with $f'(u_s) = f(x_1)$ and $f'(w_s) = f(x_2)$. This way we can get a proper complete coloring of G' from a coloring of G and as such the implication holds.

To constrain the size of our graph G' we observe that the graph consists of $G'[X]$ which contains k vertices, at-most k^q independent vertices in $G' - X$, and at-most $2 \cdot k^{q-1} \cdot k^{q-1} = 2k^{2q-2}$ connected vertex pairs in $G' - X$. As such G' has at-most $k + k^q + 2k^{2q-2}$ vertices. Thus, we have shown a kernel with $\mathcal{O}(k^{2q-2})$ vertices for all $q \geq 3$. □

Next we will prove a lower-bound on the size of possible kernels for q-COLORING on INDEPENDENT EDGE + kv. For the lower-bound proof we will first introduce a new problem definition.

q-NAE-SAT WITH COMMON LAST LITERAL	**Parameter:** n

Input: A formula in CNF with clauses $\{C_1, \ldots, C_m\}$ of size exactly q, on variables $x_1, \ldots, x_n$, such that for each clause C_i for all $i \in [m]$ the last literal of C_i is x_n.

Question: Is there a truth assignment $f\colon \{x_1, \ldots, x_n\} \to \{0, 1\}$ such that for each clause C_i for all $i \in [m]$ there is at least one literal in C_i that evaluates to 0, and at least one literal that evaluates to 1?

We will now show two linear-parameter transformations to prove a generalized kernel for q-CNF-SAT to give us a bound on the possible size of a kernel. The following lemma is inspired by [8, Theorem 3]

Lemma 2. *There is a linear-parameter transformation from q-CNF-SAT parameterized by n to $(q+1)$-NAE-SAT WITH COMMON LAST LITERAL, parameterized by n'.*

Proof. Let ϕ be a q-CNF-SAT-formula on n variables. We obtain the formula ϕ' on $n+1$ variables by adding a new variable x_{n+1}. We then add the literal x_{n+1} to each clause of ϕ. Because x_{n+1} is in each clause, ϕ' is a valid input formula for $(q+1)$-NAE-SAT WITH COMMON LAST LITERAL. Any satisfying assignment of ϕ is transformed into a satisfying not-all-equal assignment of ϕ' by setting x_{n+1} to FALSE. In the other direction, any not-all-equal assignment that satisfies ϕ' and sets x_{n+1} to FALSE, is also a satisfying CNF-SAT assignment for ϕ. If we have a not-all-equal assignment that satisfies ϕ' and sets x_{n+1} to TRUE, then the inverse of this assignment also satisfies ϕ' and sets x_{n+1} to FALSE and as such this inverse also is a satisfying CNF-SAT assignment for ϕ. Thus, the two instances are equivalent and since $n' = n + 1$, we have a linear-parameter transformation. □

Lemma 3. *There is a linear-parameter transformation from $(2q-2)$-NAE-SAT WITH COMMON LAST LITERAL, parameterized by n, to q-COLORING on INDEPENDENT EDGE + kv.*

Proof. For our transformation we will use a similar approach as used in [8, Thm. 3]. We consider an instance of $(2q-2)$-NAE-SAT on n variables with formula $\phi = \{C_1, \ldots, C_m\}$. We will show how we build a graph G and modulator $X \subseteq V(G)$ such that $G - X \in$ INDEPENDENT EDGE. We first construct a clique P to use as a palette. Using the q colors we construct the corresponding adjacent vertices $p_1, \ldots, p_q$.

For each variable x_i for $i \in [n]$, we construct a gadget V_i consisting of $2q$ vertices $T_{i,1}, \ldots, T_{i,q}, F_{i,1}, \ldots, F_{i,q}$. For all $j \in [q]$ we make $T_{i,j}$ adjacent to $F_{i,j}$. We then add a cycle through successive vertices $T_{i,1}, \ldots, T_{i,q}$ and back to $T_{i,1}$. We then connect these vertices to our palette to restrict the colors that they can get. For all $j \in [q]$ we make $T_{i,j}$ and $F_{i,j}$ adjacent to all vertices in our palette except p_j and p_{j+1} ensuring that $T_{i,j}$ and $F_{i,j}$ can only take colors j or $j+1$. These numbers are all evaluated modulo q such that $T_{i,q}$ and $F_{i,q}$ are adjacent to all but p_q and p_1.

Now any color assigned to $T_{i,j}$ will take one of the available colors for either $T_{i,j+1}$ or $T_{i,j-1}$. Since all nodes $T_{i,j}$ for $j \in [q]$ form a cycle, this means that coloring one vertex of a gadget, locks all other vertices of that gadget into one color to still have a proper coloring. Because of this we get that each gadget will have two possible colorings; a "True" coloring where $T_{i,j}$ is colored j and $F_{i,j}$ is colored $j+1$ for all $j \in [q]$ and a "False" coloring where $T_{i,j}$ is colored $j+1$ and $F_{i,j}$ is colored j for all $j \in [q]$. From this we will get our key property: For all $i \in [n]$ a coloring of V_i where $T_{i,j}$ is colored j for all $j \in [q]$ will correspond to an assignment where x_i is TRUE. Similarly, a coloring of V_i where $F_{i,j}$ is colored j for all $j \in [q]$ will correspond to an assignment where x_i is FALSE.

Now for each clause $C_k \in \phi$ with literals $(\ell_1, \ldots, \ell_{2q-2})$, let $C_{k,1} = (\ell_1, \ldots, \ell_{q-1})$ and $C_{k,2} = (\ell_q, \ldots, \ell_{2q-2})$. We will add adjacent vertices $c_{k,1}, c_{k,2}$ to our graph. For each $\ell_j \in C_{k,1}$ with $\ell_j = x_i$, we connect $c_{k,1}$ to $T_{i,j}$. For each $\ell_j \in C_{k,1}$ with $\ell_j = \neg x_i$, we connect $c_{k,1}$ to $F_{i,j}$. For each $\ell_j \in C_{k,2}$ with $\ell_j = x_i$, we connect $c_{k,2}$ to $T_{i,j-(q-1)}$. For each $\ell_j \in C_{k,2}$ with $\ell_j = \neg x_i$, we connect $c_{k,2}$ to $F_{i,j-(q-1)}$.

We can now see that this will have the desired effect of preserving the answer of an assignment into a coloring of the graph. If all first $q-1$ literals of a clause are TRUE, then the corresponding $c_{k,1}$ is adjacent to all colors except for color q. If all first $q-1$ literals of a clause are FALSE, then the corresponding $c_{k,1}$ is adjacent to all colors except for color 1. This holds the same way for the second $q-1$ literals of a clause and $c_{k,2}$. Thus, if all literals in a clause evaluate to TRUE, then both $c_{k,1}$ and $c_{k,2}$ can only be colored q, thus a proper coloring is not possible. If all literals of a clause evaluate to FALSE, then both $c_{k,1}$ and $c_{k,2}$ can only be colored 1, thus a proper coloring is not possible.

To prove; if we have a proper coloring of our graph, then from the coloring of our variable gadgets, we can extract a corresponding not-all-equal assignment of all variables. If this assignment does not satisfy ϕ, then there is a clause where either all variables are TRUE, or all are FALSE. As shown above this implies that there is a $c_{k,1}$ and $c_{k,2}$ which can either both only be colored 1 or q and as such prevent the graph from being properly colorable. Thus, by contradiction this implies that the assignment does satisfy ϕ.

In the other direction, given a satisfying not-all-equal assignment f, we need to show that there is a coloring function f' that properly colors the graph G. We start by defining f' for the palette and the variable gadgets. For all $p_i \in P = \{p_1, \ldots, p_q\}$ we define $f'(p_i) = i$. For $i \in [n]$ if $f(x_i) = True$, for all $j \in [q]$ we assign $f'(T_{i,j}) = j$ and $f'(F_{i,j}) = (j+1) \mod q$. If

$f(x_i) = False$, for all $j \in [q]$ $f'(T_{i,j}) = (j+1) \mod q$ and $f'(F_{i,j}) = j$. This gives f' as a valid partial coloring of $G - \{c_{k,1}, c_{k,2} | \text{for } C_k \in \phi\}$. It rests to show that f' can be properly extended to the vertices that we have added for each clause.

Based on the property that for each clause $C_k \in \phi$ with $C_k = \{\ell_1, \ldots, \ell_{2q-2}\}$ the last literal $\ell_{2q-2} = x_n$ and as such $c_{k,2}$ is adjacent to $T_{n,q-1}$, we will distinguish between two cases; $f(x_n) = False$ and $f(x_n) = True$.

If $f(x_n) = False$ then we can get an inverse assignment of f: $g(x_i) = \neg f(x_i)$. If f not-all-equal satisfies ϕ then g also not-all-equal satisfies ϕ. Since g is a satisfying assignment that has $g(x_n) = True$ we know that we can safely assume that $f(x_n) = True$.

Given $f(x_n) = True$ then we know that $f'(T_{n,q-1}) = q-1$, and as such for each clause $C_k = \{\ell_1, \ldots, \ell_{2q-2}\}$, $f'(c_{k,2}) = q$ would be a valid coloring. If for C_k any of the first $q-1$ literals are FALSE based on f then either ℓ_1 is FALSE and then $f'(c_{k,1}) = 1$ is a valid coloring for $c_{k,1}$, or we can find an $i \in [q-2]$ such that $\ell_i = True$ and $\ell_{i+1} = False$. We then have that $f'(c_{k,1}) = i+1$ is a valid coloring for $c_{k,1}$, thus f would be properly coloring of G. If for C_k all the first $q-1$ literals are TRUE based on f then at-least one of the second $q-1$ literals has to be FALSE, else f would not satisfy ϕ. If ℓ_q is FALSE then $f'(c_{k,2}) = 1$ is a valid coloring for $c_{k,2}$. Otherwise, we can find an $i \in [q-2]$ such that $\ell_{i+(q-1)} = True$ and $\ell_{i+q} = False$. We then have that $f'(c_{k,2}) = i+1$ is a valid coloring for $c_{k,2}$. Since all first $q-1$ literals being true implies $f'(c_{k,1}) = q$ is a valid coloring for $c_{k,1}$, we know that G can be properly colored.

As such we have shown that the nae-sat instance and our graph-coloring instance are equivalent.

Lastly to have our graph be a valid instance of q-COLORING on INDEPENDENT EDGE + kv we specify our modulator X to be $G - \{c_{k,1}, c_{k,2} | \text{for } C_k \in \phi\}$. The size of X is $|X| = q + 2qn$, which is linear in n for fixed q. Since it is trivial that the vertices $c_{k,i}$ form a subgraph that is in INDEPENDENT EDGE, we have that $G - X \in$ INDEPENDENT EDGE. Thus, we have shown a linear-parameter transformation, proving our claim. □

Lemma 4. *For all $q \geq 3$, q-COLORING on INDEPENDENT EDGE + kv does not have a kernel of bitsize $\mathcal{O}(k^{2q-3-\varepsilon})$ for any $\varepsilon > 0$, unless $coNP \subseteq NP/poly$.*

Proof. Let there be a kernel for q-COLORING on INDEPENDENT EDGE + kv, with some constant q, of bitsize $\mathcal{O}(k^{2q-3-\varepsilon})$ for an $\varepsilon > 0$. We can then make a generalized kernel for $(2q-3)$-CNF-SAT parameterized by the number of variables of size $\mathcal{O}(k^{2q-3-\varepsilon})$ by following the following steps:

1. Given an n variable instance Φ of $(2q-3)$-CNF-SAT, apply Lemma 2 to obtain an equivalent instance of Φ' on $n' \leq O(n)$ variables of $(2q-2)$-NAE-SAT parameterized by the number of variables n'.

2. By applying Lemma 3 on Φ', we get an equivalent instance (G, X) of q-COLORING on INDEPENDENT EDGE + kv with a modulator of size $k \leq \mathcal{O}(n') \leq \mathcal{O}(n)$.

3. By using the assumed kernel for q-COLORING on INDEPENDENT EDGE + kv on (G, X), we get an equivalent instance (G', X') of size $\mathcal{O}(k^{2q-3-\varepsilon})$ bits.

Hence a kernel for q-Coloring on Independent Edge + kv of bitsize $\mathcal{O}(k^{2q-3-\varepsilon})$ leads to a generalized kernel for $(2q-3)$-cnf-sat parameterized by the number of variables of bitsize $\mathcal{O}(k^{2q-3-\varepsilon})$ which implies a generalized kernel for q-cnf-sat of bitsize $\mathcal{O}(n^{q-\varepsilon})$.

In [4, Thm. 1] it is shown that for all $q \geq 3$ any generalized kernel of size $\mathcal{O}(n^{q-\varepsilon})$ for any $\varepsilon > 0$ implies $coNP \subseteq NP/poly$. As such we have shown that for $q \geq 3$ q-Coloring on Independent Edge + kv does not have a kernel of bitsize $\mathcal{O}(k^{2q-3-\varepsilon})$ for any $\varepsilon > 0$, unless $coNP \subseteq NP/poly$. □

4 q-Coloring on Clique + kv

To further explore kernelizations for q-Coloring, we introduce another graph class Clique where each connected component is a clique. Clearly we have Independent Edge $\subset$ Clique and as such Independent Edge + $kv \subset$ Clique + kv.

Lemma 5. *Given a q-list-coloring instance (C, L) on a clique C with $|C| \leq q$ that is not colorable, there exists a non-colorable (sub)clique $S \subseteq C$ where for all $v \in S$ we have $|L(v)| \leq |S| - 1$.*

Proof. We prove by induction on $|C|$.

Base case ($|C| = 1$): C is not colorable by assumption. As such the vertex $v \in C$ has $L(v) = \emptyset$. Since $|L(v)| = 0 = |C| - 1$, we have that $S = C$ is a valid subclique satisfying the lemma.

Inductive Step ($|C| > 1$): If for all $v \in C$, $|L(v)| \leq |C| - 1$ holds, then $S = C$ already satisfies the lemma, thus we assume that this does not hold. There is a $v \in C$ with $|L(v)| \geq |C|$. Let $S = C \setminus \{v\}$. If S does not satisfy the lemma, then either S is colorable or there exists a $v' \in S$ with $L(v') \geq |S|$. If S is colorable then there is a coloring f' for S which uses $|C| - 1$ colors. Since $|L(v)| \geq |C|$ there is at-least one color in $L(v)$ that is not used in f. As such we can extend f' to a coloring for C which contradicts C being not colorable. If there exists a $v' \in S$ with $L(v') \geq |S|$, then by the inductive step we find a $S' \subseteq S$ which is not colorable and for all $v \in S'$, $|L(v)| \leq |S'| - 1$. □

Lemma 6. *q-Coloring on Clique + kv admits a polynomial kernel with $\mathcal{O}(k^{q^2})$ vertices for every $q \geq 3$.*

Proof. Let (G, X) be an instance of q-Coloring on Clique + kv. We create an equivalent instance (G', X) by the following steps:

- If G contains a connected component C that is a clique of size $\leq q$ set $G := G \setminus C$. Repeat this until this is no longer the case.
- If $G - X$ contains a clique of size $> q$ return a trivial no answer.
- Set $G' := G[X]$.
- For $\ell \in [q]$, for every $S_1, \ldots, S_\ell \in \binom{X}{q+1-\ell}$ if there is a clique $\{v_1, \ldots, v_\ell\}$ of size ℓ in $G - X$ where, for all $i \in [\ell]$, $v_i \in \bigcap_{v \in S_i} N_G(v) \setminus X$, we add $\{u_1, \ldots, u_\ell\}$ to G' with $N_{G'}(u_i) := S_i \cup \{u_1, \ldots, u_\ell\} \setminus \{u_i\}$.

If we have a proper q-coloring f for G then from f we can extract a proper partial q-coloring f' for $G[X]$. Since $G'[X] = G[X]$, f' forms a proper partial q-coloring for $G'[X]$. By the way we constructed G' we know that $G' - X$ is a collection of cliques. For each such clique $\{u_1, \ldots, u_\ell\}$, we know that there is a clique $\{v_1, \ldots, v_\ell\}$ in $G - X$ such that for all $i \in [\ell]$, $N_{G'}(u_i) \cap X \subseteq N_G(v_i) \cap X$. As such let $f'(u_i) = f(v_i)$. This gives us a proper q-coloring f' of G'.

To now show that (G', X) is equivalent to (G, X) we show that a proper q-coloring f' of G' implies a proper coloring of G. From f' we can extract a proper partial q-coloring f of $G'[X]$. Since $G'[X] = G[X]$, f is a proper partial q-coloring of $G[X]$. If f can not be extended to the rest of G, then there has to be at-least one clique in $G - X$ to which this coloring can not be extended. Let this clique be C. For every vertex $v \in C$ we can get a list of admittable colors $L(v) = [q] \setminus \{f'(u) \mid u \in N_G(v) \cap X\}$. By applying lemma 5 on (C, L) we get that there is a $C' \subseteq C$, such that (C', L) does not have a proper coloring and for all $v' \in C'$, $|L(v')| \leq |C'| - 1$. By complement this means that for each $v_i \in C'$, there is a set $S_i = \{u_1, \ldots, u_{q+1-|C'|}\} \subseteq N_G(v_i) \cap X$ with $\forall_{u_a, u_b \in S_i, u_a \neq u_b} f(u_a) \neq f(u_b)$. Since this gives us a multiset $\mathbf{S} = \{S_1, \ldots, S_{|C'|}\}$ with each set having size $q + 1 - |C'|$, this would satisfy the last step in our construction phase for $\ell = |C'|$. This gives us a set $U = \{u_1, \ldots, u_\ell\}$ in G' with, for all $i \in [\ell]$, $N_{G'}(u_i) = S_i \cup \{u_1, \ldots, u_\ell\} \setminus \{u_i\}$. However since the neighborhood of this clique U is the set $\mathbf{S}$, we know that the list of admittable colors for U will be equal to the list of admittable colors for C'. Since C' is not properly list-colorable, we have a contradiction with f' being a proper coloring of G'.

Thus, we have that (G', X) is equivalent to (G, X).

To bound the size of our graph G' we sum the amount of vertices added at each iteration of our construction process. This gives us a bound of $k + \sum_{i=1}^{q} ik^{i(q+1-i)}$. This is bounded by $\mathcal{O}(k^{q^2})$. □

5 Closing the bound-gap for q-Coloring on Independent Edge + kv

In Chapter 3 of this paper, we showed a kernel of $\mathcal{O}(k^{2q-2})$ vertices for q-Coloring on Independent Edge + kv and we showed a lower-bound on possible kernel size of $\mathcal{O}(k^{2q-3})$ bits. This leaves us with a factor k gap. We will use [9] as inspiration to try to find a smaller kernel. We will use some of the definitions that were introduced in this paper, which we will repeat for completeness.

Our main idea is to look at the connected components in $G - X$ and their neighborhoods. We then formulate polynomial constraints based on a coloring, such that if there is a proper coloring of $G[X]$ that also satisfies all our constraints, then there is a proper coloring for G. We then use a theorem from the mentioned paper to show that we can reduce the set of constraints to be of size bounded by our parameter.

Theorem 1. *There is a polynomial-time algorithm that, given an instance (L, V) of d-Polynomial root CSP over an efficient field F, outputs $L' \subseteq L$ with at most $n^d + 1$ constraints such that any $0/1$-assignment to V satisfies L' if and only if it satisfies L.*

Similar to the mentioned paper, we will show how the coloring constraints expressed by our

$G - X$ graph can be encoded as a set of polynomial equalities. We will encode the color of a vertex v_i by q boolean variables $y_{i,1}, \ldots, y_{i,q}$. Here variable $y_{i,k}$ is set to TRUE if vertex v_i has color k. The property that each vertex gets one and only one color is expressed by the following definition that we use.

Definition 1. *Let $\{y_{i,k} \mid i \in [n], k \in [q]\}$ be a set of boolean variables and let $\mathbf{y}$ be the vector containing all these variables. We define our boolean variables such that $y_{i,k} = 0$ represents* FALSE *and $y_{i,k} = 1$ represents* TRUE. *We say $\mathbf{y}$ is given a choice assignment if for all $i \in [n]$: $\sum_{k=1}^{q} y_{i,k} = 1$.*

We then needed to find suitable polynomial equalities to capture the desired constraints. We will first formulate what constraints need to be captured.

Lemma 7. *Given an instance of q-COLORING on INDEPENDENT EDGE + kv. G is properly colorable if there is a proper partial coloring $f\colon X \to [q]$ of $G[X]$ such that both of the following properties are satisfied:*

1. *For all $u \in G - X$ for all $S \in \binom{N_G(u)\cap X}{q}$, there are $v, w \in S$ such that $f(v) = f(w)$*
2. *For all $\{u, v\} \in E(G - X)$ for all $S_u \in \binom{N_G(u)\cap X}{q-1}$ and $S_v \in \binom{N_G(v)\cap X}{q-1}$, it holds that $\{f(s_u) \mid s_u \in S_u\} \neq \{f(s_v) \mid s_v \in S_v\}$ or there are $w, z \in S_u$ such that $f(w) = f(z)$.*

Proof. Let (G, X) be an instance of q-COLORING on INDEPENDENT EDGE + kv and f a proper partial coloring of $G[X]$, with the above described properties. We will show how to extend f to a proper coloring of G. For each connected component $C \in G - X$ we will show how to define colors for the vertices of C such that no vertex in C will be assigned the same color as one of its neighbors. Since C is a connected component of $G - X$, vertices of C can only be adjacent to other vertices of C and to X. As such we can define colors for each C independently. We will distinguish between cases based on the number of vertices in C. Since $G - X \in$ INDEPENDENT EDGE, we know that C has at most two vertices.

(a) If $|C| = 1$, then let C be the single vertex v. Since v has no neighbors in C, the entire neighborhood of v is in X and thus is already colored. Let $S := \{f(u) \mid u \in N_G(v)\}$ be the set of colors used by the neighbors of v. If $[q]\setminus S \neq \emptyset$, then let $i \in [q]\setminus S$ and $f(v) := i$. Since there is no neighbor of v that uses color i, this is a proper extension. If $[q] \setminus S = \emptyset$, then there is a set of vertices $V_v = \{v_1, \ldots, v_q\}\} \subseteq N_G(v)$ with $f(v_i) = i$ for $i \in [q]$. Since $|V_v| = q$ and since v has no neighbors outside of X thus $N_G(v) = N_G(v) \cap X$, we know that $V_v \in \binom{N_G(v)\cap X}{q}$. As there are no distinct $v_i, v_j \in V_v$ for which $f(v_i) = f(v_j)$ we get a contradiction with f satisfying property 1.

(b) If $|C| = 2$, then let C be the adjacent vertex pair u, v. Since u and v only have each other as neighbors in C and the rest of their neighborhood is in X we know that their entire neighborhood except for each other is already colored. Let $S_u := \{f(x) \mid x \in N_G(u) \setminus \{v\}\}$ and $S_v := \{f(x) \mid x \in N_G(v) \setminus \{u\}\}$.

If $[q] \setminus S_u = \emptyset$, there is a set of vertices $V_u = \{u_1, \ldots, u_q\} \subseteq N_G(u) \setminus \{v\}$ with $f(u_i) = i$ for $i \in [q]$. Since $|V_u| = q$ and since u has no neighbors besides v outside of X thus $N_G(u) \setminus \{v\} = N_G(u) \cap X$, we know that $V_u \in \binom{N_G(u)\cap X}{q}$. Since there are no distinct $u_i, u_j \in V_u$ for which $f(u_i) = f(u_j)$, we know by contradiction with property 1 that $[q] \setminus S_u \neq \emptyset$.

If $[q] \setminus S_v = \emptyset$, there is a set of vertices $V_v = \{v_1, \ldots, v_q)\} \subseteq N_G(v) \setminus \{u\}$ with $f(v_i) = i$ for $i \in [q]$. Since $|V_v| = q$ and since v has no neighbors besides u outside of X thus $N_G(v) \setminus \{u\} = N_G(v) \cap X$, we know that $V_v \in \binom{N_G(v)\cap X}{q}$. Since there are no distinct $v_i, v_j \in V_v$ for which $f(v_i) = f(v_j)$, we know by contradiction with property 1 that $[q] \setminus S_v \neq \emptyset$.

Now we know that there is $i \in [q] \setminus S_u$ and $j \in [q] \setminus S_v$ such that if we assign color i to u, then there is no neighbor of u in X that also uses color i and equally if we assign color j to v, then there is no neighbor of v in X that also uses color j. Thus, $f(u) = i$ and $f(v) = j$ would be a proper coloring unless $i = j$. If there is no $i \in [q] \setminus S_u$ and $j \in [q] \setminus S_v$ such that $i \neq j$, then it has to be that $[q] \setminus S_u = [q] \setminus S_v = \{i\}$ for some color i. Since the order of the colors is arbitrary, we can choose i to be the q-th color and as such reference it as q. There then is a set of vertices $V_u = \{u_1, \ldots, u_{q-1}\} \subseteq N_G(u) \setminus \{v\}$ with $f(u_i) = i$ for $i \in [q-1]$. Since $|V_u| = q-1$ and since $N_G(u) \setminus \{v\} = N_G(u) \cap X$, we know that $V_u \in \binom{N_G(u)\cap X}{q-1}$. There is then also a set of vertices $V_v = \{v_1, \ldots, v_{q-1}\} \subseteq N_G(v) \setminus \{u\}$ with $f(v_i) = i$ for $i \in [q-1]$. Since $|V_v| = q-1$ and since $N_G(v) \setminus \{u\} = N_G(v) \cap X$, we know that $V_v \in \binom{N_G(v)\cap X}{q-1}$.

Since there are no distinct $u_i, u_j \in V_u$ for which $f(u_i) = f(u_j)$ and we know that $\{f(u_i) \mid u_i \in V_u\} = \{f(v_i) \mid v_i \in V_v\} = [q-1]$, we have a contradiction with f satisfying property 2.

By this we have shown that given a proper partial coloring f that satisfies properties 1 and 2, we can extend this to a complete proper coloring of G and as such G is properly colorable. □

Using this lemma we will show in a later proof that finding a proper coloring of G can be solved by finding a proper coloring of $G[X]$ for which properties 1 and 2 hold. In order to create a kernelization that finds such colorings, we will need to use polynomial equalities. We first need to find the polynomials that our kernelization is going to use. Finding these polynomials is separate from the kernelization. Once we have found the polynomials, they can be hardcoded into the algorithm.

Our next step is to show how we can create polynomial equalities that capture these two properties. We already have that property 1 is captured by the polynomial found in [9].

Lemma 8 ([9, Lemma 6]). *Let $q > 0$ be an integer and let $y_{i,k}$ for $i \in [q]$, $k \in [q]$ be boolean variables. Then there exists a polynomial p of degree $q-1$ such that for any choice assignment to $\mathbf{y}$, we have $p(\mathbf{y}) \equiv_2 0$ if and only if there are $i, j, k \in [q]$ such that $y_{i,k} = y_{j,k} = 1$.*

$$p(\mathbf{y}) := \sum_{\substack{i_1,\ldots,i_{q-1}\in[q] \\ \textit{distinct}}} \prod_{k=1}^{q-1} y_{i_k,k}$$

As a demonstration of Lemma 8 we will show $q = 3$ as an example. For $q = 3$ we get the following polynomial: $p_{q=3}(\mathbf{y}) := \sum_{i_1 \neq i_2 \in [3]} \prod_{k=1}^{2} y_{i_k,k} = y_{1,1} \cdot y_{2,2} + y_{1,1} \cdot y_{3,2} + y_{2,1} \cdot y_{1,2} + y_{2,1} \cdot y_{3,2} + y_{3,1} \cdot y_{1,2} + y_{3,1} \cdot y_{2,2}$. Since we only consider choice assignments, we will demonstrate an assignment by showing which 3 variables will have value 1. By the lemma we should have that $p_{q=3}(\mathbf{y}) \equiv_2 0$ if and only if there are $i, j, k \in [3]$ such that $y_{i,k} = y_{j,k} = 1$. Let us consider an assignment with $y_{1,1} = y_{2,2} = y_{3,3} = 1$, for this we get $p_{q=3}(\mathbf{y}) = 1 \equiv_2 1$, thus we get the expected result. Now let us consider an assignment with $y_{1,1} = y_{2,2} = y_{3,2} = 1$, for this we get $p_{q=3}(\mathbf{y}) = 2 \equiv_2 0$, thus we also get the expected result.

To find polynomials that capture property 2 we use a linear-solver for $q = 3$ and $q = 4$. Since it might not be intuitive how we use a linear-solver to find a non-linear polynomial, we will explain step-by-step how we created our inputs for the solver. We start by defining the inputs for the polynomial and determining the max degree that we want the polynomial to have. The polynomial will concern two set of $q - 1$ vertices and we want our max degree to be $2q - 3$. For our example we will use $q = 3$. We define the vertices as $\{a, b, c, d\}$ where a, b are the first $q - 1$ vertices and c, d the second set. Since our polynomial will take boolean variables as input, we define the boolean variables associated with our vertices as $\{a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3, d_1, d_2, d_3\}$. Our next step will be to generate all possible terms that we want in our polynomial. We do this by for each degree of term that we want, generate all possible combinations of our input variables. We ignore terms where there are multiple variables for the same vertex, since a vertex can only get one color in a coloring and thus such a term could never be true. In example for $q = 3$ there would be the following $6 \cdot 9 = 54$ terms of degree 2:

$$\begin{gathered}
a_1b_1, a_1c_1, a_1d_1, b_1c_1, b_1d_1, c_1d_1,\\
a_1b_2, a_1c_2, a_1d_2, b_1c_2, b_1d_2, c_1d_2,\\
a_1b_3, a_1c_3, a_1d_3, b_1c_3, b_1d_3, c_1d_3,\\
a_2b_1, a_2c_1, a_2d_1, b_2c_1, b_2d_1, c_2d_1,\\
a_2b_2, a_2c_2, a_2d_2, b_2c_2, b_2d_2, c_2d_2,\\
a_2b_3, a_2c_3, a_2d_3, b_2c_3, b_2d_3, c_2d_3,\\
a_3b_1, a_3c_1, a_3d_1, b_3c_1, b_3d_1, c_3d_1,\\
a_3b_2, a_3c_2, a_3d_2, b_3c_2, b_3d_2, c_3d_2,\\
a_3b_3, a_3c_3, a_3d_3, b_3c_3, b_3d_3, c_3d_3
\end{gathered}$$

Doing this for all degrees from 1 to $2q - 3$ gives us the vector T with all possible terms of our polynomial. We now want to get a polynomial from this by finding the coefficients for each term. To find these coefficients we want to generate a matrix M and vector B such that solving $M \cdot X = B$ gives us a vector of coefficients X.

To generate M we start by first generating all possible choice assignments for our input variables for which we want the polynomial to evaluate to 0. As an example let us consider a situation where vertices a, b, c, d all get color 1, then $a_1 = b_1 = c_1 = d_1 = 1$ and all the other variables are 0. As a simplified example we will show what the part of this row associated with the terms shown in the example above would be in the matrix M.

$$\begin{gathered}
1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\\
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
\end{gathered}$$

Alongside the matrix M that is filled with rows like this, we for each row of M have an entry with 0 to form our vector B. This gives us a linear equation such that for each row $M[i]$ we have $M[i] \cdot X = 0$. Having this M we now have that if we find coefficients M then we have a polynomial that equates to 0 if the given choice assignment satisfies our desired constraint.

We now want to also have that if the assignment does not satisfy our constraint, then the polynomial should equate to non-zero. To do this we generate all possible choice assignments for which we want the polynomial to not evaluate to 0. As an example let us consider a situation where vertices a, c get color 1 and b, d get color 2, then $a_1 = b_2 = c_1 = d_2 = 1$ and all the other variables are 0. As a simplified example we will show again what the row in M associated with this assignment would look like for the terms given above.

$$0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,$$
$$0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0$$

We can then add such row for each choice assignment to M along with a non-zero value entry for B. For $q = 4$ we were able to find a single polynomial thus we created one matrix M with all possible choice assignments with $B_i = 0$ if entry $M[i]$ corresponded with an assignment that did satisfy the constraint and $B_i = 1$ if entry $M[i]$ corresponded with an assignment that did not satisfy the constraint. For $q = 3$ we were not able to find a single polynomial, but we were able to find two polynomials. For this we created two matrices M_1 and M_2. Both M_1 and M_2 had rows for all possible assignments that did satisfy the constraint. For $q = 3$ there are 12 choice assignments that do not satisfy the constraint. M_1 had entries for the first 6 assignments and M_2 had entries for the last 6 assignments. Alongside M_1 and M_2 we had a vector B with $B[i] = 0$ if entry $M[i]$ corresponded with an assignment that did satisfy the constraint and $B_i = 1$ if entry $M[i]$ corresponded with an assignment that did not satisfy the constraint.

We then had complete matrices M and B to find our coefficients. Using the LinearSolve method in *Wolfram Mathematica 12.0* we found a vector X from which we get the coefficients that give us our polynomials. In the LinearSolve method there is a parameter to allow the solver to solve $M \cdot X \equiv_\ell B$ for some integer B. We got results by finding polynomials over integers modulo 2.

For $q = 3$ found the following two polynomials.

$$\begin{aligned} p_{q=3,1}(\mathbf{y}) := {} & d_1b_1 + d_1a_1 + c_1b_1 + c_1a_1 + d_1c_2b_1 + c_1d_2b_1 + d_1a_2b_1 + d_1a_1b_2 \\ & + d_1c_2a_1 + c_1d_2a_1 + c_1a_2b_1 + c_1a_1b_2 \end{aligned}$$

$$\begin{aligned} p_{q=3,2}(\mathbf{y}) := {} & d_1b_1 + b_2d_2 + b_1c_1 + b_2c_2 + a_1d_1 + d_2a_2 + a_1c_1 + a_2c_2 + d_1a_2b_1 \\ & + d_2a_2b_1 + b_2a_1d_1 + b_2a_1d_2 + d_1b_1c_2 + d_2b_1c_1 + b_2d_1c_2 + b_2d_2c_1 + a_2b_1c_1 \\ & + a_2b_1c_2 + b_2a_1c_1 + b_2a_1c_2 + a_1d_1c_2 + a_2d_1c_2 + d_2a_1c_1 + d_2a_2c_1 \end{aligned}$$

Given an assignment $\mathbf{y}$ that satisfies our constraint, then both $p_{q=3,1}(\mathbf{y}) \equiv_2 0$ and $p_{q=3,2}(\mathbf{y}) \equiv_2 0$. If $\mathbf{y}$ does not satisfy the assignment then either $p_{q=3,1}(\mathbf{y}) \not\equiv_2 0$ or $p_{q=3,2}(\mathbf{y}) \not\equiv_2 0$.

We will show an example let us consider an assignment with $a_1 = b_1 = c_1 = d_1 = 1$ and all other variables set to 0. We then have $p_{q=3,1}(\mathbf{y}) = 4 \equiv_2 0$ and $p_{q=3,2}(\mathbf{y}) = 4 \equiv_2 0$, thus satisfying the desired property. Let us now consider and assignment with $a_1 = b_2 = c_1 = d_2 = 1$ and all other variables set to 0. We then have $p_{q=3,1}(\mathbf{y}) = 3 \equiv_2 1$ and $p_{q=3,2}(\mathbf{y}) = 6 \equiv_2 0$. Here we see that even though this is an assignment that does not satisfy our constraints, there is still one of the equalities that equates to 0. This is where it is important that only when an assignment does satisfy our constraints will both of the equalities equate to 0. Lastly let us consider an assignment with $a_3 = b_2 = c_2 = d_3 = 1$ and all other variables set to 0. We now have $p_{q=3,1}(\mathbf{y}) = 0 \equiv_2 0$ and $p_{q=3,2}(\mathbf{y}) = 1 \equiv_2 1$. This is an example of where the first polynomial does equate to 0 but the second one does not.

For $q = 4$ we found the polynomial $p_{q=4}(\mathbf{y})$ shown in *Appendix A*.

For this polynomial we have that given an assignment $\mathbf{y}$ that satisfies our constraint, we have $p_{q=4}(\mathbf{y}) \equiv_2 0$. If $\mathbf{y}$ does not satisfy the constraint, then $p_{q=4}(\mathbf{y}) \not\equiv_2 0$

The source code that we used to generate the inputs for the solver and the results that we found can be downloaded from `https://github.com/MSchalken/Efficient-Kernels-for-Q-Coloring`. Here we have also presented a Mathematica notebook that loads in the data an shows how the coefficients we found solve $M \cdot X = B$.

To formalize the properties of these polynomials we have the following lemma.

Lemma 9. *Let $q = 3, \ell = 3$ or $q = 4, \ell = 2$ and let $y_{i,k}$ for $i \in [2q-2]$ and $k \in [q]$ be boolean variables. Then there exists a set of polynomials $\{p_1, \ldots, p_m\}$ of degree $2q-3$ such that for any choice assignment to $\mathbf{y}$, we have all $p_i(\mathbf{y}) \equiv_\ell 0$ for $i \in [\ell]$ if and only if one of the holds:*

- *There are $i, j \in [q-1]$ and $k \in [q]$ such that $y_{i,k} = y_{j,k} = 1$.*
- *There is $i \in [q-1]$ and $k \in [q]$ with $y_{i,k} = 1$ such that there is no $j \in \{q, \ldots, 2q-2\}$ for which $y_{i,k} = y_{j,k} = 1$.*

In order to be able to use Theorem 1 we need to show that integers modulo 2 and integers modulo 3 form efficient fields.

Lemma 10. *The element set of integers modulo 2 forms an efficient field.*

Proof. We first show that the integers modulo 2 form a finite field by showing the addition, subtraction, multiplication, and division tables for this element set:

+	0	1
0	0	1
1	1	0

-	0	1
0	0	1
1	1	0

×	0	1
0	0	0
1	0	1

/	0	1
0	-	0
1	-	1

Table 1: The addition, subtraction, multiplication, and division tables for the element set formed by integers modulo 2.

This shows that the field is finite. For every finite field it is possible to hard code the operation tables such that performing the operation is simply a look up in these tables. As such any finite field is trivially an efficient field. □

Having these polynomials we can now prove a smaller kernel for $q = 3$ and $q = 4$.

Lemma 11. *For $q = 3$ and $q = 4$, q-Coloring on Independent Edge + kv admits a kernel of vertex size $\mathcal{O}(k^{2q-3})$ which can be encoded in $\mathcal{O}(k^{2q-3} \log k)$ bits.*

Proof. Let G be a given input graph with modulator X with $|X| = k$. For each vertex $v \in X$ we create boolean variables $\{B_{v,i} \mid i \in [q]\}$. Let $\mathbf{B} = \{B_{v,i} \mid i \in [q], v \in X\}$ be the complete collection of all our boolean variables.

For each vertex $u \in G - X$, for each $S \in \binom{N_G(u) \cap X}{q}$, let $\mathbf{B_{u,S}}$ be the subset of $\mathbf{B}$ with all $B_{v,i}$ for all $v \in S$, for all $i \in [q]$. Using Lemma 8 we obtain a polynomial $p_{u,S}$ of degree $q-1$ such that for any choice assignment we have $p_{u,S}(B_{u,S}) \equiv_2 0$ if and only if property 1 of Lemma 7 is satisfied.

Let L_1 be this set of created polynomial equalities, thus $L_1 := \{p_{u,S}(\mathbf{B_{u,S}}) \equiv_2 0 \mid u \in G - X \wedge S \in \binom{N_G(u) \cap X}{q}\}$. Since the maximum degree of these polynomials is $q-1$, L_1 forms a valid instance of $(q-1)$-Polynomial root CSP over the integers modulo 2. Using Theorem 1 we can find $L'_1 \subseteq L_1$ with $|L'| \leq (qk)^{q-1}+1$ such that any assignment of $\mathbf{B}$ satisfies L'_1 if and only if it satisfies L_1.

For each adjacent vertex pair $u, v \in G - X$, for each $S_1 \in \binom{N_G(u) \cap X}{q-1}$, for each $S_2 \in \binom{N_G(v) \cap X}{q-1}$, let $\mathbf{B_{u,S_1}}$ be the subset of $\mathbf{B}$ with all $B_{w,i}$ for all $w \in S_1$, for all $i \in [q]$ and let $\mathbf{B_{v,S_2}}$ be the subset of $\mathbf{B}$ with all $B_{w,i}$ for all $w \in S_2$, for all $i \in [q]$.

We now make a case distinction between $q = 3$ and $q = 4$:

$q = 3$: We use polynomials $p_{q=3,1}(\mathbf{y})$ and $p_{q=3,2}(\mathbf{y})$ that we found with as variables the concatenation of $\{\mathbf{B_{u,S_1}}, \mathbf{B_{v,S_2}}\}$ to get polynomials $p_{u,v,S_1,S_2,1}$ and $p_{u,v,S_1,S_2,2}$ of degree $2q-3$ such that for any choice assignment we have $p_{u,v,S_1,S_2,1}(\mathbf{B_{u,S_1}} \cup \mathbf{B_{v,S_2}}) \equiv_2 0 \wedge p_{u,v,S_1,S_2,2}(\mathbf{B_{u,S_1}} \cup \mathbf{B_{v,S_2}}) \equiv_2 0$ if and only if property 2 of Lemma 7 is satisfied.

$q = 4$: We use the polynomial $p_{q=4}(\mathbf{y})$ that we found with as variables the concatenation of $\{\mathbf{B_{u,S_1}}, \mathbf{B_{v,S_2}}\}$ to get a polynomial p_{u,v,S_1,S_2} of degree $2q-3$ such that for any choice assignment we have $p_{u,v,S_1,S_2}(\mathbf{B_{u,S_1}} \cup \mathbf{B_{v,S_2}}) \equiv_2 0$ if and only if property 2 of Lemma 7 is satisfied.

Let L_2 be this second set of created polynomial equalities, thus:

For $q = 3$:

$$L_2 := \{p_{u,v,S_1,S_2,1}(\mathbf{B_{u,S_1}} \cup \mathbf{B_{v,S_2}}) \equiv_2 0 \cup p_{u,v,S_1,S_2,2}(\mathbf{B_{u,S_1}} \cup \mathbf{B_{v,S_2}}) \equiv_2 0 \mid u, v \in G - X \wedge \{u,v\} \in E(G) \wedge S_1 \in \binom{N_G(u) \cap X}{q-1} \wedge S_2 \in \binom{N_G(v) \cap X}{q-1}\}$$

For $q = 4$:

$$L_2 := \{p_{u,v,S_1,S_2}(\mathbf{B_{u,S_1}} \cup \mathbf{B_{v,S_2}}) \equiv_2 0 \mid u, v \in G - X \wedge \{u,v\} \in E(G) \wedge S_1 \in \binom{N_G(u) \cap X}{q-1} \wedge S_2 \in \binom{N_G(v) \cap X}{q-1}\}$$

Since the maximum degree of these polynomials is $2q-3$, L_2 forms a valid instance of $(2q-3)$-Polynomial root CSP over the integers modulo 2 for $q = 3$ and a valid instance of $(2q-3)$-Polynomial root CSP over the integers modulo 2 for $q = 4$. Using Theorem 1 we can find $L'_2 \subseteq L_2$ with $|L'_2| \leq (qk)^{2q-3} + 1$ such that any assignment of $\mathbf{B}$ satisfies L'_2 if and only if it satisfies L_2.

To now show our kernelization we will show how to construct G'. Start by setting $G' = G[X]$. For every equality $p_{u,S}(B_{u,S}) \equiv_2 0 \in L'_1$, add u to G' if it is not yet in G'. For every $v \in S$, add $\{u, v\}$ to $E(G')$ if it is not yet in $E(G')$. For every equality $p_{u,v,S_1,S_2,\_}(\mathbf{B_{u,S_1}} \cup \mathbf{B_{v,S_2}}) \equiv_2 0 \in L'_2$, add u to G' if it is not yet in G', add v to G' if it is not yet in G', add $\{u, v\}$ to $E(G')$ if it is not yet in $E(G')$. For every $w \in S_1$, add (u, w) to $E(G')$ if it is not yet in

$E(G')$. For every $w \in S_2$, add (v, w) to $E(G')$ if it is not yet in $E(G')$. For every equality $p_{u,v,S_1,S_2}(\mathbf{B_{u,S_1}} \cup \mathbf{B_{v,S_2}}) \equiv_2 0 \in L'$, add u to G' if it is not yet in G', add v to G' if it is not yet in G', add $\{u, v\}$ to $E(G')$ if it is not yet in $E(G')$. For every $w \in S_1$, add (u, w) to $E(G')$ if it is not yet in $E(G')$. For every $w \in S_2$, add (v, w) to $E(G')$ if it is not yet in $E(G')$.

Correctness proof: Since G' is a subgraph of G it is trivial that if G is q-colorable then G' is q-colorable. Thus, it remains to show that if G' is q-colorable, then G is q-colorable.

Let f' be a proper coloring of G'. By the construction of G' we know that f' is also a proper partial coloring of $G[X]$. We will prove that G is colorable by showing that f' satisfies properties 1 and 2 from Lemma 7. We start by creating a choice assignment $\mathbf{y}$ corresponding to f'. For all vertices $v_i \in X$, for all $j \in [q]$, let $y_{i,j} := 1$ if $f'(v_i) = j$ and let $y_{i,j} := 0$ if $f'(v_i) \neq j$. Let $\mathbf{y}$ be the collection of all $y_{i,j}$. Since each vertex will only have one color in f', it will hold for all $v_i \in X$ that $\sum_{k=1}^{q} y_{i,k} = 1$. As such $\mathbf{y}$ is a choice assignment.

Claim 1. $\mathbf{y}$ *satisfies* L'_1 *and* L_1*.*

Let there be a polynomial equality $p_{u,S}(\mathbf{B_{u,S}}) \equiv_2 0 \in L'_1$ that is not satisfied. Then by Lemma 8 we have that for all $v_i \in S$ there is no $v_j \in S \setminus \{v_i\}$ and $k \in [q]$ such that $y_{i,k} = y_{j,k} = 1$. This means by our construction of $\mathbf{y}$ that there are no distinct $v, w \in S$ such that $f'(v) = f'(w)$. By the construction of G' we know that since $p_{u,S}(\mathbf{B_{u,S}}) \equiv_2 0 \in L'_1$, there is $u \in G'$ and $S \subseteq N_{G'}(u)$. Now since $|S| = q$ and there are no distinct $v, w \in S$ such that $f'(v) = f'(w)$ then there has to be a $z \in S$ such that $f'(u) = f'(z)$. However we know that $(u, z) \in E(G')$, thus contradicting that f' is a proper coloring. As such we know that $\mathbf{y}$ satisfies L'_1. By Theorem 1 we know that $\mathbf{y}$ also satisfies L_1.

Claim 2. f' *satisfies property* 1 *from Lemma 7.*

Let us assume that f' does not satisfy property 1 from Lemma 7. We then have that there is a $u \in G - X$, for which there is a $S \in \binom{N_G(u) \cap X}{q}$ for which there are no distinct $v, w \in S$ such that $f'(v) = f'(w)$. By the construction of G' we know that there has to be a polynomial equality $p_{u,S}(B_{u,S}) \equiv 0 \in L_1$. From Lemma 8 we know that since L_1 is satisfied, there has to be $v_i, v_j \in S$ and $k \in [q]$ such that $y_{i,k} = y_{j,k} = 1$. As such we would have $f'(v_i) = f'(v_j)$. This contradicts with our statement that there are no distinct $v, w \in S$ such that $f'(v) = f'(w)$. Therefore we know that f' satisfies property 1 from Lemma 7.

Claim 3. $\mathbf{y}$ *satisfies* L'_2 *and* L_2*.*

Let there be a polynomial equality $p_{u,v,S_1,S_2}(\mathbf{B_{u,S_1}} \cup \mathbf{B_{v,S_2}}) \equiv_\ell 0$ in L'_2 that is not satisfied. By Lemma 9 we have that there are no $v_i, v_j \in S_1$ and $k \in [q]$ such that $y_{i,k} = y_{j,k} = 1$. As such there is no $v_i, v_j \in S_1$ such that $f'(v_i) = f'(v_j)$. Furthermore, we have that there is no $v_i \in S_1$ and $k \in [q]$ for which there exists no $v_j \in S_2$ such that $y_{i,k} = y_{j,k} = 1$. As such for all $v_i \in S_1$, there is a $v_j \in S_2$ such that $f'(v_i) = f'(v_j)$. By the construction of G' we know that since $p_{u,v,S_1,S_2}(\mathbf{B_{u,S_1}} \cup \mathbf{B_{v,S_2}}) \equiv_\ell 0 \in L'_2$, there are $u, v \in G'$ and $S_1 \subseteq N_{G'}(u)$ and $S_2 \subseteq N_{G'}(v)$. Now since $|S_1| = q - 1$ and there are no $w, z \in S_1$ such that $f'(w) = f'(z)$ then there is only one color c_1 for which there is no $w \in S_1$ with $f'(w) = c_1$. As such $f'(u) = c_1$ has to be the case otherwise f' would not be a proper coloring. Let $f'(v) = c_2$. Since u and v are adjacent in G' it has to hold that $c_1 \neq c_2$. As such there has to be $w \in S_1$ such that

$f'(w) = c_2$. Because of this there has to be a $z \in S_2$ with $f'(z) = f'(w) = c_2$. Since in G' we have that v and z are adjacent and $f'(v) = f'(w) = c_2$, we have a contradiction with f' being a proper coloring. Thus, have that $\mathbf{y}$ satisfies L'_2. By Theorem 1 we know that $\mathbf{y}$ also satisfies L_2.

Claim 4. *f' satisfies property* 2 *from Lemma 7.*

Let us assume that f' does not satisfy property 2 from Lemma 7. We then have that there is an adjacent vertex pair $u, v \in G - X$, for which there are $S_u \in \binom{N_G(u) \cap X}{q-1}$ and $S_v \in \binom{N_G(v) \cap X}{q-1}$, for which there are no $w, z \in S_u$ such that $f'(w) = f'(z)$ and for each $s_u \in S_u$ there is $s_v \in S_v$ for which $f'(u) = f'(v)$. By the construction of G' we know there has to be a polynomial equality $p_{u,v,S_1,S_2}(\mathbf{B_{u,S_1}} \cup \mathbf{B_{v,S_2}}) \equiv_\ell 0 \in L_2$. From Lemma 9 we know that since L_2 is satisfied, there either have to be $v_i, v_j \in S_1$ and $k \in [q]$ for which $y_{i,k} = y_{j,k} = 1$ and as such $f'(v_i) = f'(v_j)$. Or there is a $v_i \in S_1$ and $k \in [q]$ with $y_{i,k} = 1$, such that there is no $v_j \in S_2$ for which $y_{i,k} = y_{j,k} = 1$ and as such there is a $v_i \in S_1$ for which there is no v_j with $f'(v_i) = f'(v_j)$. This contradicts with our statement that there are no $w, z \in S_u$ such that $f'(w) = f'(z)$ and for each $s_u \in S_u$ there is $s_v \in S_v$ for which $f'(u) = f'(v)$. Therefore we know that f' satisfies property 2 from Lemma 7.

Since we have that f' is a proper partial coloring of $G[X]$ that satisfies properties 1 and 2 from Lemma 7, we have that G is properly colorable. As such we have that G' being properly colorable implies G being properly colorable.

Kernel size: For both $q = 3$ and $q = 4$ we have that the vertex size of our kernel is equal to the size of k plus the amount of vertices that we added based on the polynomial constraints. This is smaller or equal to: $k + |L'_1| + 2 \cdot |L'_2| \leq k + (qk)^{q-1} + 1 + 2 \cdot (qk)^{2q-3} + 2$. This is bounded by $\mathcal{O}(k + k^{q-1} + k^{2q-3}) = \mathcal{O}(k^{2q-3})$ vertices.

Thus, for $q = 3$ and $q = 4$ our kernel size is bounded by $\mathcal{O}(k^{2q-3})$ vertices.

To from this bound get a size bound in bits we have to look at the complete size of G'. We have that $|G'| = |V(G')| + |E(G')|$. Since we have a bound on the amount on vertices, we want to find a bound on the amount of edges in G' expressed in some function of $|V(G')|$. By the construction of G' we have that $|E(G')| \leq E(G[X]) + q \cdot |L'_1| + (2q-1) \cdot |L'_2| \leq k^2 + q \cdot ((qk)^{q-1} + 1) + (2q-1) \cdot ((qk)^{2q-3} + 1) = \mathcal{O}(k^{2q-3})$. Using an adjacency list encoding our graph will have a size $\mathcal{O}(|E| \log |V| + |V|)$. For our graph this means that we can encode the graph with $\mathcal{O}(k^{2q-3} \cdot \log k^{2q-3}) = \mathcal{O}(k^{2q-3} \log k)$ with a constant q. □

It is notable that this proof structure works for also for $q \geq 5$ if we are able to find polynomials that capture the required constraints. Thus, if such polynomials were to be found, this same approach could be used to show a kernel for generalized q.

6 Conclusion

We have introduced the parameterized problem of q-Coloring with as parameter the deletion distance to an Independent Edge graph. We have shown a kernelization for this problem with a size-bound of $\mathcal{O}(k^{2q-2})$ vertices. Furthermore, we have proven that for this problem no kernels of size $\mathcal{O}(k^{2q-3-\varepsilon})$ bits exist unless $coNP \subseteq NP/poly$. We have also used a tactic introduced by Jansen and Pieterse to show that at-least for $q = 3$ and $q = 4$ we can show a kernel of $\mathcal{O}(k^{2q-3} \log k)$ bits, thus closing the gap between our lower and upper bound to $k^{O(1)}$ factors.

We have also introduced the parameterized problem of q-Coloring with as parameter the deletion distance to a Clique graph. Using the same method as we used for our first kernelization, we have shown a kernel for q-Coloring on Clique + kv of $\mathcal{O}(k^{q^2})$ vertices.

For future research one could try and find proper polynomials to capture coloring constraints for $q \geq 5$ for q-Coloring on Independent Edge + kv. By proving that proper polynomial equalities exist for a generalized q one could prove that a kernel exists for generalized q.

Another direction for future research could be to prove a lower bound on the size of kernelizations for q-Coloring on Clique + kv. It will be likely that there will not be lower bound that is tight to the size of our existing kernel, thus such research could be followed with proving the existence of smaller kernel than we have presented.

References

[1] Bodlaender, H. L., Jansen, B. M., and Kratsch, S. Kernelization lower bounds by cross-composition. *SIAM Journal on Discrete Mathematics 28*, 1 (2014), 277–305.

[2] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to algorithms*. MIT press, 2009.

[3] Cygan, M., Fomin, F. V., Kowalik, Ł., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., and Saurabh, S. *Parameterized algorithms*, vol. 4. Springer, 2015.

[4] Dell, H., and Van Melkebeek, D. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *Journal of the ACM (JACM) 61*, 4 (2014), 1–27.

[5] Doucha, M., and Kratochvíl, J. Cluster vertex deletion: A parameterization between vertex cover and clique-width. In *Mathematical Foundations of Computer Science 2012* (Berlin, Heidelberg, 2012), B. Rovan, V. Sassone, and P. Widmayer, Eds., Springer Berlin Heidelberg, pp. 348–359.

[6] Fiala, J., Golovach, P. A., and Kratochvíl, J. Parameterized complexity of coloring problems: Treewidth versus vertex cover. *Theoretical Computer Science 412*, 23 (2011), 2513–2523.

[7] Garey, M., Johnson, D., and Stockmeyer, L. Some simplified NP-complete graph problems. *Theoretical Computer Science 1*, 3 (1976), 237 – 267.

[8] Jansen, B. M. P., and Kratsch, S. Data reduction for graph coloring problems. *Information and Computation 231* (2013), 70 – 88. Fundamentals of Computation Theory.

[9] Jansen, B. M. P., and Pieterse, A. Optimal data reduction for graph coloring using low-degree polynomials. *Algorithmica 81*, 10 (Oct 2019), 3865–3889.

Appendices

A: Polynomial for $q = 4$

$$
\begin{aligned}
p_{q=4}(\mathbf{y}) := {} & f_2c_1a_2e_1 + a_3f_3c_1e_1 + e_2c_1a_2f_1 + a_3c_1e_3f_1 + c_2e_1f_2a_1 + c_2f_1e_2a_1 + c_2f_3a_3e_2 \\
& + c_2a_3e_3f_2 + f_3e_1c_3a_1 + e_2a_2f_3c_3 + e_3f_1c_3a_1 + a_2e_3c_3f_2 + d_2c_1a_2e_1 + a_3c_1d_3e_1 \\
& + e_2d_1c_1a_2 + a_3d_1c_1e_3 + c_2d_2e_1a_1 + c_2d_1e_2a_1 + c_2a_3e_2d_3 + c_2d_2a_3e_3 + e_1d_3c_3a_1 \\
& + e_2a_2d_3c_3 + e_3d_1c_3a_1 + a_2e_3d_2c_3 + b_2c_1f_2e_1 + f_3c_1b_3e_1 + e_2c_1b_2f_1 + b_3c_1e_3f_1 \\
& + c_2e_1f_2b_1 + c_2b_1e_2f_1 + c_2f_3e_2b_3 + c_2b_3e_3f_2 + f_3e_1c_3b_1 + e_2f_3b_2c_3 + e_3b_1c_3f_1 \\
& + e_3b_2c_3f_2 + d_2c_1b_2e_1 + b_3c_1d_3e_1 + e_2d_1c_1b_2 + b_3d_1c_1e_3 + c_2d_2e_1b_1 + c_2d_1e_2b_1 \\
& + c_2d_3e_2b_3 + c_2d_2b_3e_3 + e_1d_3c_3b_1 + e_2b_2d_3c_3 + e_3d_1c_3b_1 + d_2e_3b_2c_3 + d_2c_1a_2f_1 \\
& + d_1c_1a_2f_2 + a_3c_1d_3f_1 + a_3f_3d_1c_1 + c_2d_2f_1a_1 + c_2d_1f_2a_1 + c_2a_3d_3f_2 + c_2f_3a_3d_2 \\
& + d_3f_1c_3a_1 + f_3d_1c_3a_1 + a_2d_3c_3f_2 + a_2f_3d_2c_3 + d_2c_1b_2f_1 + b_3c_1d_3f_1 + b_2d_1c_1f_2 \\
& + f_3d_1c_1b_3 + c_2d_2b_1f_1 + c_2d_1f_2b_1 + c_2b_3d_3f_2 + c_2f_3b_3d_2 + b_1d_3c_3f_1 + b_2d_3c_3f_2 \\
& + f_3d_1c_3b_1 + f_3d_2b_2c_3 + b_2e_1f_2a_1 + f_3e_1b_3a_1 + b_1e_1a_2f_2 + a_3f_3e_1b_1 + e_2f_1b_2a_1 \\
& + e_2b_1a_2f_1 + e_2f_3b_3a_2 + e_2f_3a_3b_2 + b_3f_1e_3a_1 + a_2b_3e_3f_2 + a_3b_1e_3f_1 + a_3b_2e_3f_2 \\
& + d_2e_1b_2a_1 + b_3e_1d_3a_1 + d_2e_1a_2b_1 + a_3e_1d_3b_1 + e_2d_1b_2a_1 + e_2d_1a_2b_1 + e_2b_3d_3a_2 \\
& + e_2a_3d_3b_2 + b_3d_1e_3a_1 + a_2d_2b_3e_3 + a_3d_1e_3b_1 + a_3d_2b_2e_3 + d_2f_1b_2a_1 + b_3d_3f_1a_1 \\
& + d_1b_2f_2a_1 + f_3d_1b_3a_1 + d_2b_1a_2f_1 + b_1d_1a_2f_2 + b_3d_3a_2f_2 + f_3d_2b_3a_2 + a_3b_1d_3f_1 \\
& + a_3b_2d_3f_2 + a_3f_3d_1b_1 + a_3f_3b_2d_2 + e_1f_2a_2c_1d_3 + e_1f_3a_2d_2c_1 + e_1f_2a_3c_1d_3 \\
& + e_1f_3a_3d_2c_1 + e_2f_1a_2c_1d_3 + e_2d_1f_3a_2c_1 + e_2f_1a_3c_1d_3 + e_2d_1f_3a_3c_1 + f_1a_2d_2c_1e_3 \\
& + d_1f_2a_2c_1e_3 + f_1a_3d_2c_1e_3 + d_1f_2a_3c_1e_3 + e_1f_2c_2d_3a_1 + e_1f_3c_2d_2a_1 + e_1f_2a_3c_2d_3 \\
& + e_1f_3a_3c_2d_2 + e_2f_1c_2d_3a_1 + e_2d_1f_3c_2a_1 + e_2a_3f_1c_2d_3 + e_2d_1f_3a_3c_2 + f_1c_2d_2e_3a_1 \\
& + d_1f_2c_2e_3a_1 + a_3f_1c_2d_2e_3 + d_1f_2a_3c_2e_3 + e_1f_2c_3d_3a_1 + e_1f_3c_3d_2a_1 + e_1f_2a_2c_3d_3 \\
& + e_1f_3a_2c_3d_2 + e_2f_1c_3d_3a_1 + e_2d_1f_3c_3a_1 + e_2f_1a_2c_3d_3 + e_2d_1f_3a_2c_3 + f_1c_3d_2e_3a_1 \\
& + d_1f_2c_3e_3a_1 + f_1a_2c_3d_2e_3 + d_1f_2a_2c_3e_3 + e_1f_2a_2b_3c_1 + e_1f_3a_2b_3c_1 + e_1f_2b_2a_3c_1 \\
& + e_1f_3b_2a_3c_1 + e_2f_1a_2b_3c_1 + e_2f_3a_2b_3c_1 + e_2f_1b_2a_3c_1 + e_2f_3b_2a_3c_1 + f_1a_2b_3c_1e_3 \\
& + f_2a_2b_3c_1e_3 + f_1b_2a_3c_1e_3 + f_2b_2a_3c_1e_3 + e_1f_2b_3c_2a_1 + e_1f_3b_3c_2a_1 + e_1f_2a_3c_2b_1 \\
& + e_1f_3a_3c_2b_1 + e_2f_1b_3c_2a_1 + e_2f_3b_3c_2a_1 + e_2a_3f_1c_2b_1 + e_2f_3a_3c_2b_1 + f_1b_3c_2e_3a_1 \\
& + f_2b_3c_2e_3a_1 + a_3f_1c_2e_3b_1 + f_2a_3c_2e_3b_1 + e_1f_2b_2c_3a_1 + e_1f_3b_2c_3a_1 + e_1f_2a_2c_3b_1 \\
& + e_1f_3a_2c_3b_1 + e_2f_1b_2c_3a_1 + e_2f_3b_2c_3a_1 + e_2f_1a_2c_3b_1 + e_2f_3a_2c_3b_1 + f_1b_2c_3e_3a_1 \\
& + f_2b_2c_3e_3a_1 + f_1a_2c_3e_3b_1 + f_2a_2c_3e_3b_1 + e_1a_2b_3d_2c_1 + e_1a_2b_3c_1d_3 + e_1b_2a_3d_2c_1 \\
& + e_1b_2a_3c_1d_3 + e_2d_1a_2b_3c_1 + e_2a_2b_3c_1d_3 + e_2d_1b_2a_3c_1 + e_2b_2a_3c_1d_3 + d_1a_2b_3c_1e_3 \\
& + a_2b_3d_2c_1e_3 + d_1b_2a_3c_1e_3 + b_2a_3d_2c_1e_3 + e_1b_3c_2d_2a_1 + e_1b_3c_2d_3a_1 + e_1a_3c_2d_2b_1 \\
& + e_1a_3c_2d_3b_1 + e_2d_1b_3c_2a_1 + e_2b_3c_2d_3a_1 + e_2d_1a_3c_2b_1 + e_2a_3c_2d_3b_1 + d_1b_3c_2e_3a_1 \\
& + b_3c_2d_2e_3a_1 + d_1a_3c_2e_3b_1 + a_3c_2d_2e_3b_1 + e_1b_2c_3d_2a_1 + e_1b_2c_3d_3a_1 + e_1a_2c_3d_2b_1 \\
& + e_1a_2c_3d_3b_1 + e_2d_1b_2c_3a_1 + e_2b_2c_3d_3a_1 + e_2d_1a_2c_3b_1 + e_2a_2c_3d_3b_1 + d_1b_2c_3e_3a_1 \\
& + b_2c_3d_2e_3a_1 + d_1a_2c_3e_3b_1 + a_2c_3d_2e_3b_1 + e_1f_2b_2c_1d_3 + e_1f_2b_3c_1d_3 + e_1f_3b_2d_2c_1 \\
& + e_1f_3b_3d_2c_1 + e_2f_1b_2c_1d_3 + e_2f_1b_3c_1d_3 + e_2d_1f_3b_2c_1 + e_2d_1f_3b_3c_1 + f_1b_2d_2c_1e_3
\end{aligned}
$$

$$
\begin{aligned}
&+ f_1b_3d_2c_1e_3 + d_1f_2b_2c_1e_3 + d_1f_2b_3c_1e_3 + e_1f_2c_2d_3b_1 + e_1f_2b_3c_2d_3 + e_1f_3c_2d_2b_1 \\
&+ e_1f_3b_3c_2d_2 + e_2f_1c_2d_3b_1 + e_2f_1b_3c_2d_3 + e_2d_1f_3c_2b_1 + e_2d_1f_3b_3c_2 + f_1c_2d_2e_3b_1 \\
&+ f_1b_3c_2d_2e_3 + d_1f_2c_2e_3b_1 + d_1f_2b_3c_2e_3 + e_1f_2c_3d_3b_1 + e_1f_2b_2c_3d_3 + e_1f_3c_3d_2b_1 \\
&+ e_1f_3b_2c_3d_2 + e_2f_1c_3d_3b_1 + e_2f_1b_2c_3d_3 + e_2d_1f_3c_3b_1 + e_2d_1f_3b_2c_3 + f_1c_3d_2e_3b_1 \\
&+ f_1b_2c_3d_2e_3 + d_1f_2c_3e_3b_1 + d_1f_2b_2c_3e_3 + f_1a_2b_3d_2c_1 + f_1a_2b_3c_1d_3 + d_1f_2a_2b_3c_1 \\
&+ f_2a_2b_3c_1d_3 + d_1f_3a_2b_3c_1 + f_3a_2b_3d_2c_1 + f_1b_2a_3d_2c_1 + f_1b_2a_3c_1d_3 + d_1f_2b_2a_3c_1 \\
&+ f_2b_2a_3c_1d_3 + d_1f_3b_2a_3c_1 + f_3b_2a_3d_2c_1 + f_1b_3c_2d_2a_1 + f_1b_3c_2d_3a_1 + d_1f_2b_3c_2a_1 \\
&+ f_2b_3c_2d_3a_1 + d_1f_3b_3c_2a_1 + f_3b_3c_2d_2a_1 + a_3f_1c_2d_2b_1 + a_3f_1c_2d_3b_1 + d_1f_2a_3c_2b_1 \\
&+ f_2a_3c_2d_3b_1 + d_1f_3a_3c_2b_1 + f_3a_3c_2d_2b_1 + f_1b_2c_3d_2a_1 + f_1b_2c_3d_3a_1 + d_1f_2b_2c_3a_1 \\
&+ f_2b_2c_3d_3a_1 + d_1f_3b_2c_3a_1 + f_3b_2c_3d_2a_1 + f_1a_2c_3d_2b_1 + f_1a_2c_3d_3b_1 + d_1f_2a_2c_3b_1 \\
&+ f_2a_2c_3d_3b_1 + d_1f_3a_2c_3b_1 + f_3a_2c_3d_2b_1 + e_1f_2b_2d_3a_1 + e_1f_2b_3d_3a_1 + e_1f_3b_2d_2a_1 \\
&+ e_1f_3b_3d_2a_1 + e_1f_2a_2d_3b_1 + e_1f_2a_2b_3d_3 + e_1f_3a_2d_2b_1 + e_1f_3a_2b_3d_2 + e_1f_2a_3d_3b_1 \\
&+ e_1f_2b_2a_3d_3 + e_1f_3a_3d_2b_1 + e_1f_3b_2a_3d_2 + e_2f_1b_2d_3a_1 + e_2f_1b_3d_3a_1 + e_2d_1f_3b_2a_1 \\
&+ e_2d_1f_3b_3a_1 + e_2f_1a_2d_3b_1 + e_2f_1a_2b_3d_3 + e_2d_1f_3a_2b_1 + e_2d_1f_3a_2b_3 + e_2f_1a_3d_3b_1 \\
&+ e_2f_1b_2a_3d_3 + e_2d_1f_3a_3b_1 + e_2d_1f_3b_2a_3 + f_1b_2d_2e_3a_1 + f_1b_3d_2e_3a_1 + d_1f_2b_2e_3a_1 \\
&+ d_1f_2b_3e_3a_1 + f_1a_2d_2e_3b_1 + f_1a_2b_3d_2e_3 + d_1f_2a_2e_3b_1 + d_1f_2a_2b_3e_3 + f_1a_3d_2e_3b_1 \\
&+ f_1b_2a_3d_2e_3 + d_1f_2a_3e_3b_1 + d_1f_2b_2a_3e_3
\end{aligned}
$$