

# A Complete Linear Programming Hierarchy for Linear Codes

Leonardo Nagami Coregliano ✉🏠

Institute for Advanced Study, Princeton, NJ, USA

Fernando Granha Jeronimo ✉🏠

Institute for Advanced Study, Princeton, NJ, USA

Chris Jones ✉🏠

University of Chicago, IL, USA

---

## Abstract

A longstanding open problem in coding theory is to determine the best (asymptotic) rate  $R_2(\delta)$  of binary codes with minimum constant (relative) distance  $\delta$ . An existential lower bound was given by Gilbert and Varshamov in the 1950s. On the impossibility side, in the 1970s McEliece, Rodemich, Rumsey and Welch (MRRW) proved an upper bound by analyzing Delsarte’s linear programs. To date these results remain the best known lower and upper bounds on  $R_2(\delta)$  with no improvement even for the important class of *linear* codes. Asymptotically, these bounds differ by an exponential factor in the blocklength.

In this work, we introduce a new hierarchy of linear programs (LPs) that converges to the true size  $A_2^{\text{Lin}}(n, d)$  of an optimum *linear* binary code (in fact, over any finite field) of a given blocklength  $n$  and distance  $d$ . This hierarchy has several notable features:

1. It is a natural generalization of the Delsarte LPs used in the first MRRW bound.
2. It is a hierarchy of linear programs rather than semi-definite programs potentially making it more amenable to theoretical analysis.
3. It is *complete* in the sense that the optimum code size can be retrieved from level  $O(n^2)$ .
4. It provides an answer in the form of a hierarchy (in larger dimensional spaces) to the question of how to cut Delsarte’s LP polytopes to approximate the true size of *linear* codes.

We obtain our hierarchy by generalizing the Krawtchouk polynomials and MacWilliams inequalities to a suitable “higher-order” version taking into account interactions of  $\ell$  words. Our method also generalizes to translation schemes under mild assumptions.

**2012 ACM Subject Classification** Theory of computation

**Keywords and phrases** Coding theory, code bounds, convex programming, linear programming hierarchy

**Digital Object Identifier** 10.4230/LIPIcs.ITCS.2022.51

**Related Version** *Full Version*: <https://arxiv.org/abs/2112.09221>

**Funding** This material is based upon work supported by the National Science Foundation under grant numbers listed below. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation (NSF).

*Leonardo Nagami Coregliano*: This material is based upon work supported by the National Science Foundation, and by the IAS School of Mathematics.

*Fernando Granha Jeronimo*: This material is based upon work supported by the National Science Foundation under Grant No. CCF-1900460.

*Chris Jones*: This material is based upon work supported by the National Science Foundation under Grant No. CCF-2008920.

**Acknowledgements** We would like to thank the anonymous reviewers for their comments and helpful feedback.



© Leonardo Nagami Coregliano, Fernando Granha Jeronimo, and Chris Jones; licensed under Creative Commons License CC-BY 4.0

13th Innovations in Theoretical Computer Science Conference (ITCS 2022).

Editor: Mark Braverman; Article No. 51; pp. 51:1–51:22

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 1 Introduction

A fundamental question in coding theory is the maximum size of a binary code given a blocklength parameter  $n$  and a minimum distance parameter  $d_n$ . This value is typically denoted by  $A_2(n, d_n)$ . A particularly important regime occurs when  $\lim_{n \rightarrow \infty} d_n/n = \delta$  for some absolute constant  $\delta \in (0, 1/2)$ . In this regime,  $A_2(n, d_n)$  is known to grow exponentially in  $n$ . However, the precise rate of this exponential growth remains an elusive major open problem. It is often convenient to denote the asymptotic basis of this growth as  $2^{R_2(\delta)}$ , where the rate  $R_2(\delta)$  is defined as

$$R_2(\delta) := \limsup_{n \rightarrow \infty} \frac{1}{n} \log_2 (A_2(n, \lfloor \delta n \rfloor)).$$

An equivalent way of defining  $A_2(n, d)$  is as the independence number of the graph  $H_{n,d}$  whose vertex set is  $V(H_{n,d}) := \mathbb{F}_2^n$  and two vertices  $x, y \in V(H_{n,d})$  are adjacent if and only if their Hamming distance  $\Delta(x, y)$  lies in  $\{1, \dots, d-1\}$ . Note that there is a one-to-one correspondence between independent sets in this graph and binary codes of blocklength  $n$  and minimum distance  $d$ . Most of the literature about  $A_2(n, d)$  takes advantage of this graph-theoretic interpretation.

A lower bound on  $A_2(n, d)$  follows from the trivial bound on the independence number of a graph, namely,  $\alpha(H_{n,d}) \geq |V(H_{n,d})| / (\deg(H_{n,d}) + 1)$ , which gives  $\alpha(H_{n,d}) \geq 2^{(1-h_2(d/n)+o(1))n}$  where  $h_2$  is the binary entropy function. First discovered by Gilbert [7] and later generalized to linear codes by Varshamov [20], this existential bound is now known as the Gilbert–Varshamov (GV) bound. Observe that the GV bound readily implies that  $R_2(\delta) \geq 1 - h_2(\delta)$ . Despite its simplicity, this bound remains the best (existential) lower bound on  $R_2(\delta)$ .

The techniques to upper bound  $A_2(n, d)$  are oftentimes more involved, with the most prominent being the Delsarte linear programming method that we now describe. A binary code  $C \subseteq \mathbb{F}_2^n$  is *linear* if it is a subspace of  $\mathbb{F}_2^n$  and its weight distribution is the tuple  $(a_0, a_1, \dots, a_n) \in \mathbb{N}^{n+1}$ , where  $a_i$  is the number of codewords of  $C$  of Hamming weight  $i$ . MacWilliams [10] showed that the weight distribution  $(b_0, b_1, \dots, b_n)$  of the dual code  $C^\perp$  can be obtained by applying a linear transformation to  $(a_0, a_1, \dots, a_n)$ . More precisely, the MacWilliams identities establish that

$$b_j = \frac{1}{|C|} \sum_{i=0}^n K_j(i) \cdot a_i,$$

where the coefficients  $K_j(i)$  are evaluations of the so-called Krawtchouk (or Kravchuk) polynomial of degree  $j$ . The Krawtchouk polynomials form a family of orthogonal polynomials under the measure  $\mu_n(i) = \binom{n}{i}/2^n$  and they play an important role in coding theory [19, Chapter 1]. Since the weight distribution of the dual  $C^\perp$  is non-negative, the MacWilliams identities can be relaxed to inequalities

$$\sum_{i=0}^n K_j(i) \cdot a_i \geq 0$$

for  $j = 0, \dots, n$ . This naturally leads to the following linear program (LP) relaxation for  $A_2(n, d)$  when  $C$  is a linear code (recall that for a linear code, having distance at least  $d$  is equivalent to having no words of Hamming weight 1 through  $d-1$ ).

$$\begin{aligned}
\max \quad & \sum_{i=0}^n a_i \\
\text{s.t.} \quad & a_0 = 1 && \text{(Normalization)} \\
& a_i = 0 && \text{for } i = 1, \dots, d-1 \quad \text{(Distance constraints)} \\
& \sum_{j=1}^n K_i(j) \cdot a_j \geq 0 && \text{for } i = 0, \dots, n \quad \text{(MacWilliams inequalities)} \\
& a_i \geq 0 && \text{for } i = 0, \dots, n \quad \text{(Non-negativity)}.
\end{aligned}$$

■ **Figure 1** Delsarte’s linear program for  $A_2(n, d)$ .

The  $a_i$  can be suitably generalized to codes which are not necessarily linear (by setting  $a_i := |\{(x, y) \in C^2 \mid \Delta(x, y) = i\}| / |C|^2$ ). The MacWilliams inequalities hold for these generalized  $a_i$ ’s as proven by MacWilliams, Sloane and Goethals [11]. Therefore, the same linear program above also bounds  $A_2(n, d)$  for general codes. This family of linear programs was first introduced by Delsarte in [2], where it was obtained in greater generality from the theory of association schemes. We refer to the above linear program as Delsarte’s linear program, or, more formally, as DelsarteLP( $n, d$ ).

The best known upper bound on  $R_2(\delta)$  for distances  $\delta \in (0.273, 1/2)$  is obtained by constructing solutions to the dual program of Delsarte’s linear program, as first done by McEliece, Rodemich, Rumsey and Welch (MRRW) [12] in their first linear programming bound. In the same work, McEliece et al. also gave the best known bound for  $\delta \in (0, 0.273)$  via a second family of linear programs. Since our techniques are more similar to their first linear programming bound, we restrict our attention to it in this discussion. In the first linear programming bound, they showed that  $R_2(\delta) \leq h_2(1/2 - \sqrt{\delta(1-\delta)})$  with a reasonably sophisticated argument using properties of general orthogonal polynomials and also particular properties of Krawtchouk polynomials. Simpler perspectives on the first LP bound analysis were found by Navon and Samorodnitsky [13] and by Samorodnitsky [15].

Instead of linear programming, one can use more powerful techniques based on semi-definite programming (SDP) to upper bound  $A_2(n, d)$ . For instance, the Sum-of-Squares/Lasserre SDP hierarchy was suggested for this problem by Laurent [9]. The value of the program equals  $\alpha(H_{n,d})$  for a sufficiently high level of the hierarchy, so in principle analyzing these programs could give  $A_2(n, d)$  exactly. Analyzing SDP methods to improve  $R_2(\delta)$  seems challenging and we do not even know how to analyze the simplest of them [17], which is weaker than degree-4 Sum-of-Squares (see related work below for more details on SDP methods).

On the one hand, we have reasonably simple linear programs of Delsarte already requiring a non-trivial theoretical analysis for proving upper bounds on  $R_2(\delta)$ . On the other hand we have more sophisticated SDP methods which are provably stronger than the Delsarte LP, but for which no theoretical analyses are known.

## 1.1 Our Contribution

In this work, we refine the Delsarte linear programming method used in the first LP bound for  $A_2(n, d)$  by designing a *hierarchy* of linear programs. For a parameter  $\ell \in \mathbb{N}_+$ , the hierarchy is based on specific higher-order versions of Krawtchouk polynomials and MacWilliams inequalities that take advantage of  $\ell$ -point interactions of words. We denote by KrawtchoukLP( $n, d, \ell$ ) the linear programming relaxation for  $A_2(n, d)$  at level  $\ell$  of our hierarchy.

We define  $A_2^{\text{Lin}}(n, d)$  analogously to  $A_2(n, d)$  as the maximum size of a *linear* binary code of blocklength  $n$  and minimum distance  $d$ . For linear codes, we impose additional “semantic” constraints on the programs in the hierarchy taking advantage of the linear structure of the code. We denote by  $\text{KrawtchoukLP}_{\text{Lin}}(n, d, \ell)$  the linear program relaxation for  $A_2^{\text{Lin}}(n, d)$  with these additional constraints. Both  $\text{KrawtchoukLP}_{\text{Lin}}(n, d, 1)$  and  $\text{KrawtchoukLP}(n, d, 1)$  coincide with  $\text{DelsarteLP}(n, d)$  at the first level of our hierarchy.

There is a known gap between the value of Delsarte’s linear programs and the GV bound. In particular when  $\delta = 1/2 - \varepsilon$ , Delsarte’s linear programs do not yield an upper bound tighter than  $R_2(1/2 - \varepsilon) \leq \Theta(\varepsilon^2 \log(1/\varepsilon))$ , as shown by Navon and Samorodnitsky [13], whereas the GV bound establishes a lower bound of  $R_2(1/2 - \varepsilon) \geq \Omega(\varepsilon^2)$ . There are no known improvements to these bounds even for the important class of *linear* codes. If the GV bound is indeed tight, then analyzing DelsarteLP is not sufficient to prove it. The goal of our hierarchy is to give tighter and tighter upper bounds on  $A_2(n, d)$  as the level of the hierarchy increases.

We show that for *linear* codes the hierarchy  $\text{KrawtchoukLP}_{\text{Lin}}(n, d, \ell)$  is *complete*, meaning that the value of the hierarchy converges to  $A_2^{\text{Lin}}(n, d)$  as  $\ell$  grows larger. We prove that level roughly  $\ell = O(n^2)$  is enough to retrieve the correct value of  $A_2^{\text{Lin}}(n, d)$ . More generally, for linear codes over  $\mathbb{F}_q$ , we have the following completeness theorem for  $A_q^{\text{Lin}}(n, d)$ .

► **Theorem 1** (Completeness - Informal version of Theorem 39). *For  $\ell \geq \Omega_{\varepsilon, q}(n^2)$ , we have*

$$A_q^{\text{Lin}}(n, d) \leq \text{val}(\text{KrawtchoukLP}_{\text{Lin}}^{\mathbb{F}_q}(n, d, \ell))^{1/\ell} \leq (1 + \varepsilon) \cdot A_q^{\text{Lin}}(n, d).$$

We think that the  $\text{KrawtchoukLP}_{\text{Lin}}(n, d, \ell)$  hierarchy is an extremely interesting object for the following reasons.

1. It takes advantage of higher-order interactions of codewords by naturally computing Hamming weight statistics of subspaces spanned by  $\ell$  codewords (see Definition 5).
2. It is a generalization of the Delsarte LP used in the first MRRW bound and the two share strong structural similarities (see Section 3).
3. It is a hierarchy of linear programs rather than semi-definite programs (see Definition 15 and Section 4.2).
4. It is a *complete* hierarchy (see Theorem 39).
5. It provides an answer in the form of a hierarchy (in larger dimensional spaces) to the question of how to cut Delsarte’s LP polytopes [13] to approximate the true size of *linear* codes.

We hope this hierarchy will fill an important gap in the coding theory literature between Delsarte’s LP, for which theoretical analyses are known, and more powerful SDP methods, for which we seem to have no clue how to perform asymptotic analysis.

Not unexpectedly, the hierarchy  $\text{KrawtchoukLP}(n, d, \ell)$  corresponding to general (not necessarily linear) codes does not improve on Delsarte’s linear program. Without the extra structure of linearity, the number of constraints we can add to our LP hierarchy is limited. We prove that solutions of  $\text{DelsarteLP}(n, d)$  (easily) lift to solutions of  $\text{KrawtchoukLP}(n, d, \ell)$  with the same value as follows.

► **Proposition 2** (Hierarchy Collapse - Informal version of Proposition 40). *For  $\ell \in \mathbb{N}$ , we have*

$$\text{val}(\text{KrawtchoukLP}(n, d, \ell))^{1/\ell} = \text{DelsarteLP}(n, d).$$

This contrast between the hierarchies  $\text{KrawtchoukLP}_{\text{Lin}}(n, d, \ell)$  and  $\text{KrawtchoukLP}(n, d, \ell)$  reinvigorates the question of whether the maximum sizes of general and linear codes are substantially different or not.

Though we give special attention to the binary case since it may be the most important one, we prove completeness and lifting results more generally in the language of association schemes. For example, a suitable modification of the linear programming hierarchy also converges to the maximum size of a  $D$ -code in the Hamming scheme over any finite field (see full version of this paper); this in particular covers other types of codes that may be of interest such as  $\varepsilon$ -balanced codes.

## More on Related Work

Although quantitatively the McEliece et al. [12] upper bound on  $R_2(\delta)$  has not improved, our qualitative understanding of this upper bound is now substantially better. Friedman and Tillich [4] designed generalized Alon–Boppana theorems in order to bound the size of linear binary codes. Inspired by [4], Navon and Samorodnitsky [13, 14] rederived the McEliece et al. bound on  $R_2(\delta)$  for general codes using a more intuitive proof based on Fourier analysis. Despite a seemingly different language, the proof in [13] also yields feasible solutions to the dual of Delsarte’s LP as in MRRW. More recently, Samorodnitsky [15] gave yet a new interpretation of the McEliece et al. upper bound and conjectured interesting hypercontractivity inequalities towards improving the upper bound on  $R_2(\delta)$ .

Schrijver [16] showed that the seemingly artificial Delsarte LP has the same value<sup>1</sup> as the Lovász  $\vartheta'$  relaxation for  $\alpha(H_{n,d})$ , which is also essentially the degree-2 Sum-of-Squares/Lasserre relaxation of  $\alpha(H_{n,d})$  (with additional non-negativity constraints on the entries of the matrix). Schrijver showed that this holds generally for commutative association schemes, a connection that allows us to also express KrawtchoukLP as  $\vartheta'$  of a certain graph.

A line of work (similar in motivation to the current work) is to strengthen a convex relaxation of  $A_2(n, d)$ . In Delsarte’s approach, only the distance between pairs of points is taken into account in the optimization. For this reason, Delsarte’s approach is classified as a 2-point bound [18]. Nonetheless, there is no reason to restrict oneself to just 2-point interactions. Schrijver [17] constructed a family of semi-definite programs (SDPs) for  $A_2(n, d)$  designed to take into account the 3-point interactions. Extending Schrijver’s result to a 4-point interaction bound, Gijswijt, Mittelmann and Schrijver [5] gave another tighter family of SDPs for  $A_2(n, d)$  (they also give a description of their hierarchy for arbitrary  $\ell$ ). A complete SDP hierarchy for  $\alpha(H_{n,d})$  is the Sum-of-Squares/Lasserre hierarchy, which was proposed for code upper bounds by Laurent [8], building on de Klerk et al. [1].

Since the Sum-of-Squares hierarchy is guaranteed to find the correct value of  $\alpha(H_{n,d})$  when the level is sufficiently high (precisely, level  $2\alpha(H_{n,d})$ ), in principle it would be enough to analyze this SDP to compute  $A_2(n, d)$ . Unfortunately, studying the performance of SDPs on a fixed instance is a notoriously difficult task. In particular, the global positive semi-definiteness constraint is nontrivial. Unfortunately, no theoretical analysis is known for “genuine” SDP methods even for the simplest of them, the 3-point bound of Schrijver [17] mentioned above.

In summary, the state of affairs on upper bounding  $A_2(n, d)$  or  $A_2^{\text{Lin}}(n, d)$  is as follows. On one hand, we have a thorough theoretical understanding of techniques based on Delsarte’s LP, but if the true value of  $A_2(n, d)$  or  $A_2^{\text{Lin}}(n, d)$  is closer to the GV bound, then these techniques fall short of providing tight bounds. On the other hand, we have  $\ell$ -point bounds from SDP techniques capable of yielding the correct value of  $A_2(n, d)$ , but (apparently) no

<sup>1</sup> In fact, by a symmetrization of the  $\vartheta'$  SDP on  $H_{n,d}$  using its graph automorphisms, one obtains DelsarteLP( $n, d$ ) exactly, see Section 4.

clue how to theoretically analyze them to bound  $R_2(\delta)$  for general codes or linear codes. We hope that our hierarchy will open a new angle of attack on this elusive problem for the important class of *linear* binary codes.

## 1.2 Outline of the Paper

Section 2 contains some notation and basic facts.

Section 3 shows the construction of the LP hierarchy for the binary code case. In this section, we introduce the notion of an  $\ell$ -*configuration*, which roughly capture the Hamming weights of all words in the subspace spanned by the  $\ell$  points. In analogy with the usual the Delsarte LP, we then analyze statistics of codes called  $\ell$ -*configuration profiles*, which capture the number of  $\ell$ -tuples in each possible  $\ell$ -configuration. In the rest of the section we construct higher-order Krawtchouk polynomials, show MacWilliams identities, define the LP hierarchy and prove that its restrictions can be computed in  $O(n^{2^{\ell+1}-2})$  time (for  $\ell \in \mathbb{N}_+$  fixed).

Section 4 shows how the LP hierarchy admits several other interpretations. The LP hierarchy is a *symmetrization* of an exponential-size hierarchy, which has a natural interpretation either as checking non-negativity of Fourier coefficients of the code, or as  $\vartheta'(G)$  for a large graph  $G$ .

We study our construction in more generality through the theory of *association schemes* (see full paper). Our construction can be seen as adding extra constraints to the  $\ell$ -fold tensor product of the Delsarte LP. More specifically, the underlying association scheme is a *refinement* of the  $\ell$ -fold tensor product scheme in which “semantic” constraints can be added due to linearity of the code in the original translation scheme. We study this type of refinement, giving conditions under which it is still a bona fide translation scheme. The other sections may be read mostly independently of this section.

In Section 5, we show the main results: that the LP hierarchy is complete for linear codes, and no better than the Delsarte LP in the general (not necessarily linear) case.

We conclude in Section 6 with some open problems. Please, see the full version of this paper for details and complete proofs.

## 2 Preliminaries

A binary code  $C$  of block length  $n$  is a subset of  $\mathbb{F}_2^n$ . For a word  $x \in \mathbb{F}_2^n$ , we denote by  $|x| := |\{i \in [n] \mid x_i \neq 0\}|$  its *Hamming weight*. Given two words  $x, y \in \mathbb{F}_2^n$ , we denote by  $\Delta(x, y) := |x - y|$  their *Hamming distance*. The (*minimum*) *distance* of  $C$  is defined by  $\Delta(C) := \min\{\Delta(x, y) \mid x, y \in C \wedge x \neq y\}$ . The *rate* of  $C$  is defined by  $r(C) := \log_2(|C|)/n$ . The maximum size of a code of blocklength  $n$  and minimum distance at least  $d$  is defined as

$$A_2(n, d) := \max\{|C| \mid C \subseteq \mathbb{F}_2^n, \Delta(C) \geq d\}.$$

We denote the *asymptotic rate* of codes of relative distance at least  $\delta$  and alphabet size  $q$  as

$$R_2(\delta) := \limsup_{n \rightarrow \infty} \frac{1}{n} \log_2(A_2(n, \lfloor \delta n \rfloor)).$$

We define  $A_2^{\text{Lin}}(n, d)$  and  $R_2^{\text{Lin}}(\delta)$  for linear codes in an analogous way, by further requiring the code  $C$  to be *linear* (i.e., an  $\mathbb{F}_2$ -linear subspace of  $\mathbb{F}_2^n$ ).

Note that a code of distance at least  $d$  can alternatively be viewed as an independent set in the *Hamming cube graph of distance less than  $d$* ,  $H_{n,d}$ , whose vertex set is  $V(H_{n,d}) := \mathbb{F}_2^n$  and whose edge set is  $E(H_{n,d}) := \{\{x, y\} \in \binom{\mathbb{F}_2^n}{2} \mid \Delta(x, y) \leq d - 1\}$ .

Let  $f, g : \mathbb{F}_2^n \rightarrow \mathbb{R}$ . We denote by  $\langle f, g \rangle := \mathbb{E}_{x \in \mathbb{R}\mathbb{F}_2^n} [f(x)g(x)]$  the *inner product* of  $f$  and  $g$  under the uniform measure and we denote by  $f * g$  their convolution given by  $(f * g)(x) := \mathbb{E}_{y \in \mathbb{R}\mathbb{F}_2^n} [f(y) \cdot g(x - y)]$  ( $x \in \mathbb{F}_2^n$ ). The *Fourier transform*  $\widehat{f}$  of  $f$  is given by  $\widehat{f}(x) := \langle f, \chi_x \rangle = \mathbb{E}_{y \in \mathbb{R}\mathbb{F}_2^n} [f(y) \cdot \chi_x(y)]$ , where  $\chi_x(y) := (-1)^{\langle x, y \rangle}$ . The (simple) Plancherel identity will be used in our computations.

► **Fact 3** (Plancherel). *Let  $f, g : \mathbb{F}_2^n \rightarrow \mathbb{R}$ . Then  $\langle f, g \rangle = \sum_{x \in \mathbb{F}_2^n} \widehat{f}(x) \cdot \widehat{g}(x)$ .*

Given a linear code  $C \subseteq \mathbb{F}_2^n$ , the *dual code* of  $C$  is defined as  $C^\perp := \{x \in \mathbb{F}_2^n \mid \forall y \in C, \chi_x(y) = 1\}$ . The Fourier transform of the indicator of a linear code maps it to a multiple of the indicator of its dual code in the following way.

► **Fact 4**. *If  $C \subseteq \mathbb{F}_2^n$  is a linear code and  $\mathbb{1}_C$  is its indicator function, then  $\widehat{\mathbb{1}_C} = |C| \cdot \mathbb{1}_{C^\perp} / 2^n = \mathbb{1}_{C^\perp} / |C^\perp|$ .*

### 3 Krawtchouk Hierarchies for Binary Codes

In this section, we describe the LP hierarchy for the standard case of binary codes. We opt for an ad hoc derivation from boolean Fourier analysis to show how the higher-order Krawtchouk polynomials nicely parallel the usual Krawtchouk polynomials. We will generalize the construction using the language of association schemes (see full version of this paper).

#### 3.1 Higher-order Krawtchouk polynomials

As we alluded to previously, we want to incorporate  $\ell$ -point interactions in our optimization problem for  $A_2(n, d)$  similar in spirit to the Sum-of-Squares semi-definite programming hierarchy for the independence number of a graph but in the simpler setting of *linear* programming. To accomplish this goal, we measure the profile of “configurations” of  $\ell$ -tuples of codewords from the code.

We start with the definition of symmetric difference configurations. In plain English, the symmetric difference configuration of an  $\ell$ -tuple  $(z_1, \dots, z_\ell) \in (\mathbb{F}_2^n)^\ell$  of words captures all information of  $(z_1, \dots, z_\ell)$  corresponding to Hamming weights of linear combinations of the words.

► **Definition 5**. *The symmetric difference configuration of the  $\ell$ -tuple  $(z_1, \dots, z_\ell) \in (\mathbb{F}_2^n)^\ell$  is the function  $\text{Config}_{n,\ell}^\Delta(z_1, \dots, z_\ell) : 2^{[\ell]} \rightarrow \mathbb{R}$  defined by*

$$\text{Config}_{n,\ell}^\Delta(z_1, \dots, z_\ell)(J) := \left| \sum_{j \in J} z_j \right|,$$

for every  $J \subseteq [\ell]$ , that is, the value of the function at  $J \subseteq [\ell]$  is the Hamming weight of the linear combination  $\sum_{j \in J} z_j$ .

By viewing  $\text{Config}_{n,\ell}^\Delta$  as a function  $(\mathbb{F}_2^n)^\ell \rightarrow \mathbb{R}^{2^{[\ell]}}$  (i.e., a function from the space of  $\ell$ -tuples of words to the space of functions  $2^{[\ell]} \rightarrow \mathbb{R}$ ), the set of (valid) symmetric difference configurations of  $\ell$ -tuples of elements of  $\mathbb{F}_2^n$  is captured by its image  $\text{im}(\text{Config}_{n,\ell}^\Delta)$ .

Given a symmetric difference configuration  $g \in \text{im}(\text{Config}_{n,\ell}^\Delta)$ , we will also abuse notation and write  $(z_1, \dots, z_\ell) \in g$  to mean that  $(z_1, \dots, z_\ell) \in (\mathbb{F}_2^n)^\ell$  has configuration  $\text{Config}_{n,\ell}^\Delta(z_1, \dots, z_\ell) = g$ . In other words, this abuse of notation consists of thinking of a

## 51:8 A Complete Linear Programming Hierarchy for Linear Codes

configuration as the set of all  $\ell$ -tuples of words that have this configuration (see also Lemma 8 below). We also let  $|g|$  be the size of this set, i.e., the number of  $\ell$ -tuples whose configuration is  $g$ .

The trivial symmetric difference configuration is the constant 0 function (denoted by  $0$ ), which is the symmetric configuration of the tuple  $(0, \dots, 0) \in (\mathbb{F}_2^n)^\ell$ .

► **Remark 6.** A configuration measures the Hamming weights of vectors in the subspace of  $\mathbb{F}_2^n$  spanned by  $z_1, \dots, z_\ell$ . However, Definition 5 depends on the choice of basis for the subspace. With more technical difficulty one can define configurations in a basis-independent way; see the discussion at the end of Section 4.1.

Even though the space  $(\mathbb{F}_2^n)^\ell$  has exponential size in  $n$  (for a fixed  $\ell$ ), the next lemma says that the number of configurations is polynomial in  $n$  (for a fixed  $\ell$ ).

► **Lemma 7.** *We have*

$$|\text{im}(\text{Config}_{n,\ell}^\Delta)| = \binom{n + 2^\ell - 1}{2^\ell - 1}.$$

The next lemma provides an alternative way of viewing configurations: for each symmetric difference configuration  $g \in \text{im}(\text{Config}_{n,\ell}^\Delta)$ , the set of  $\ell$ -tuples with a certain configuration  $g$  is precisely one of the orbits of the natural (diagonal) right action of the symmetric group  $S_n$  on  $n$  points on  $(\mathbb{F}_2^n)^\ell$ .

► **Lemma 8.** *Let  $n, \ell \in \mathbb{N}_+$  and consider the natural (diagonal) right action of  $S_n$  on  $(\mathbb{F}_2^n)^\ell$  given by  $(x_1, \dots, x_\ell) \cdot \sigma := (y_1, \dots, y_\ell)$ , where  $(y_j)_i := (x_j)_{\sigma(i)}$   $((x_1, \dots, x_\ell), (y_1, \dots, y_\ell) \in (\mathbb{F}_2^n)^\ell, \sigma \in S_n, j \in [\ell], i \in [n])$ .*

*The following are equivalent for  $(x_1, \dots, x_\ell), (y_1, \dots, y_\ell) \in (\mathbb{F}_2^n)^\ell$ .*

1.  $(x_1, \dots, x_\ell)$  and  $(y_1, \dots, y_\ell)$  are in the same  $S_n$ -orbit.
2.  $\text{Config}_{n,\ell}^\Delta(x_1, \dots, x_\ell) = \text{Config}_{n,\ell}^\Delta(y_1, \dots, y_\ell)$ .

Similarly to the weight profile of a code, we can define a higher-order configuration profile.

► **Definition 9.** *The  $\ell$ -configuration profile of a code  $C \subseteq \mathbb{F}_2^n$  is the sequence  $(a_g^C)_{g \in \text{im}(\text{Config}_{n,\ell}^\Delta)}$  defined by*

$$a_g^C := \frac{1}{|C|^\ell} \left| \left\{ ((x_1, \dots, x_\ell), (y_1, \dots, y_\ell)) \in C^\ell \times C^\ell \mid \text{Config}_{n,\ell}^\Delta(x_1 - y_1, \dots, x_\ell - y_\ell) = g \right\} \right|.$$

► **Remark 10.** Note that if  $C$  is linear,  $a_g^C$  can alternatively be computed as

$$a_g^C = |\{(z_1, \dots, z_\ell) \in C^\ell \mid \text{Config}_{n,\ell}^\Delta(z_1, \dots, z_\ell) = g\}|.$$

Recall that the (usual) Krawtchouk polynomial  $K_i$  of degree  $i$  is defined by

$$\begin{aligned} K_i(t) &:= 2^n \mathbb{E}_{x \in \mathbb{F}_2^n} [\mathbf{1}_{W_i}(x) \cdot \chi_y(x)] \\ &= \sum_{x \in W_i} \chi_y(x), \end{aligned}$$

where  $W_i \subseteq \mathbb{F}_2^n$  is the set of all words of Hamming weight  $i$ ,  $\mathbf{1}_{W_i} : \mathbb{F}_2^n \rightarrow \{0, 1\}$  is its indicator function and  $y \in W_t$  is any element with of Hamming weight  $t$ .



► **Definition 11** (Higher-order Krawtchouk). *Let  $h \in \text{im}(\text{Config}_{n,\ell}^\Delta)$  be a symmetric difference configuration. The higher-order Krawtchouk polynomial indexed by  $h$  is the function  $K_h: \text{im}(\text{Config}_{n,\ell}^\Delta) \rightarrow \mathbb{R}$  defined by*

$$\begin{aligned} K_h(g) &:= 2^{\ell n} \mathbb{E}_{(y_1, \dots, y_\ell) \in (\mathbb{F}_2^n)^\ell} \left[ \mathbb{1}_h(y_1, \dots, y_\ell) \cdot \prod_{j=1}^{\ell} \chi_{x_j}(y_j) \right] \\ &= \sum_{(y_1, \dots, y_\ell) \in h} \prod_{j=1}^{\ell} \chi_{x_j}(y_j), \end{aligned} \tag{1}$$

for every symmetric difference configuration  $g \in \text{im}(\text{Config}_{n,\ell}^\Delta)$ , where  $(x_1, \dots, x_\ell) \in g$  is any  $\ell$ -tuple of words with symmetric difference configuration  $g$  and  $\mathbb{1}_h$  is the indicator function of the set of  $\ell$ -tuples whose symmetric difference configuration is  $h$  (Lemma 23 shows this is well-defined).

► **Remark 12.** Another way to see  $K_h$  above is as the unique function (see Lemma 23 below) such that

$$\widehat{\mathbb{1}_h} = \frac{K_h \circ \text{Config}_{n,\ell}^\Delta}{2^{n\ell}}.$$

Note that when  $\ell = 1$ , a symmetric difference configuration  $\text{Config}_{n,1}^\Delta(x)$  of a word  $x \in \mathbb{F}_2^n$  only tracks the Hamming weight  $\text{Config}_{n,1}^\Delta(x)(\{1\}) = |x|$  of  $x$  (as  $\text{Config}_{n,1}^\Delta(x)(\emptyset)$  is always equal to 0) thus we recover the univariate Krawtchouk polynomials.

For explicit computations of the higher-order Krawtchouk polynomials, the formula (1) is quite inconvenient as it involves a sum of  $2^{n\ell}$  terms. We will provide an alternative formula in Section 3.4.

## 3.2 Higher-order MacWilliams Identities and Inequalities

In this section, we show a higher-order analogue of MacWilliams identities and inequalities using only basic Fourier analysis. Later we are going to define a suitable family of association schemes from which MacWilliams identities and inequalities follow from the general theory of association schemes of Delsarte [2, 3].

The MacWilliams identities show a surprising combinatorial fact: the weight profile of the dual  $C^\perp$  of a linear code  $C \subseteq \mathbb{F}_2^n$  is completely determined by the weight profile of  $C$ . The higher-order MacWilliams identities generalize this fact to  $\ell$ -configuration profiles.

► **Lemma 13** (Higher-order MacWilliams identities). *Let  $C \in \mathbb{F}_2^n$  be a linear code and let  $h \in \text{im}(\text{Config}_{n,\ell}^\Delta)$  be a symmetric difference configuration. Then*

$$a_h^{C^\perp} = \frac{1}{|C|^\ell} \sum_{g \in \text{im}(\text{Config}_{n,\ell}^\Delta)} K_h(g) \cdot a_g^C.$$

Just as the usual MacWilliams inequalities hold for arbitrary codes, we can prove that the same transformation at least yields non-negative numbers.

► **Lemma 14** (Higher-order MacWilliams inequalities). *Let  $C \in \mathbb{F}_2^n$  be an arbitrary code. For  $h \in \text{im}(\text{Config}_{n,\ell}^\Delta)$ , we have*

$$\sum_{g \in \text{im}(\text{Config}_{n,\ell}^\Delta)} K_h(g) \cdot a_g^C \geq 0.$$

### 3.3 Higher-order Delsarte's Linear Programs

Now we have all the elements to define a hierarchy of linear programs for  $A_2(n, d)$  parameterized by the size of the interactions  $\ell \in \mathbb{N}_+$  in analogy to  $\text{DelsarteLP}(n, d)$ .

► **Definition 15.** For  $n, \ell \in \mathbb{N}_+$  and  $d \in \{0, 1, \dots, n\}$ , we let  $\text{KrawtchoukLP}(n, d, \ell)$  be the following linear program.

$$\begin{aligned}
& \max && \sum_{g \in \text{im}(\text{Config}_{n, \ell}^\Delta)} a_g \\
& \text{s.t.} && a_0 = 1 && (\text{Normalization}) \\
& && a_g = 0 && \forall g \in \text{ForbConfig}(n, d, \ell) && (\text{Distance constraints}) \\
& && \sum_{g \in \text{im}(\text{Config}_{n, \ell}^\Delta)} K_h(g) \cdot a_g \geq 0 && \forall h \in \text{im}(\text{Config}_{n, \ell}^\Delta) && (\text{MacWilliams inequalities}) \\
& && a_g \geq 0 && \forall g \in \text{im}(\text{Config}_{n, \ell}^\Delta) && (\text{Non-negativity}),
\end{aligned}$$

where the variables are  $(a_g)_{g \in \text{im}(\text{Config}_{n, \ell}^\Delta)}$  and

$$\text{ForbConfig}(n, d, \ell) := \{g \in \text{im}(\text{Config}_{n, \ell}^\Delta) \mid \exists j \in [\ell], g(\{j\}) \in \{1, \dots, d-1\}\}.$$

We also define  $\text{KrawtchoukLP}_{\text{Lin}}(n, d, \ell)$  as the linear program obtained by replacing the set  $\text{ForbConfig}(n, d, \ell)$  with

$$\text{ForbConfig}_{\text{Lin}}(n, d, \ell) := \{g \in \text{im}(\text{Config}_{n, \ell}^\Delta) \mid \exists J \subseteq [\ell], g(J) \in \{1, \dots, d-1\}\}.$$

► **Proposition 16.** The linear programs  $\text{KrawtchoukLP}(n, d, \ell)$  and  $\text{KrawtchoukLP}_{\text{Lin}}(n, d, \ell)$  are sound, that is, we have

$$\begin{aligned}
& \text{val}(\text{KrawtchoukLP}(n, d, \ell))^{1/\ell} \geq A_2(n, d), \\
& \text{val}(\text{KrawtchoukLP}_{\text{Lin}}(n, d, \ell))^{1/\ell} \geq A_2^{\text{Lin}}(n, d).
\end{aligned}$$

**Proof.** Recall that for  $C \subseteq \mathbb{F}_2^n$ , we have

$$a_g^C := \frac{1}{|C|^\ell} |\{(x_1, \dots, x_\ell), (y_1, \dots, y_\ell) \in C^\ell \times C^\ell \mid \text{Config}_{n, \ell}^\Delta(x_1 - y_1, \dots, x_\ell - y_\ell) = g\}|.$$

If  $C$  is an arbitrary code of distance at least  $d$ , then Lemma 14 implies that the  $\ell$ -configuration profile  $a^C$  satisfies the MacWilliams inequalities. On the other hand, if  $g \in \text{ForbConfig}(n, d, \ell)$ , that is, we have  $g(\{j\}) \in \{1, \dots, d-1\}$  for some  $j \in [\ell]$ , then clearly no pair of  $\ell$ -tuples of codewords  $(x_1, \dots, x_\ell), (y_1, \dots, y_\ell) \in C^\ell$  can satisfy  $\text{Config}_{n, \ell}^\Delta(x_1 - y_1, \dots, x_\ell - y_\ell) = g$  as it would imply  $|x_j - y_j| = g(\{j\}) \in \{1, \dots, d-1\}$ , thus the distance constraints are also satisfied.

All other restrictions follow trivially from the definition of  $a^C$ , thus  $a^C$  is a feasible solution of  $\text{KrawtchoukLP}(n, d, \ell)$ . Since the objective value of  $a^C$  is  $\sum_{g \in \text{im}(\text{Config}_{n, \ell}^\Delta)} a_g^C = |C|^\ell$ , it follows that  $\text{val}(\text{KrawtchoukLP}(n, d, \ell))^{1/\ell} \geq A_2(n, d)$ .

If we further assume that  $C$  is linear and  $g \in \text{ForbConfig}_{\text{Lin}}(n, d, \ell)$  is such that  $g(J) \in [d-1]$  for some  $J \subseteq [\ell]$ , then no tuple  $(z_1, \dots, z_\ell) \in C^\ell$  can satisfy  $\text{Config}_{n, \ell}^\Delta(z_1, \dots, z_\ell) = g$  as it would imply  $|\sum_{j \in J} z_j| = g(J) \in \{1, \dots, d-1\}$ . By Remark 10, we get  $a_g^C = 0$ , so  $a^C$  is also a feasible solution of  $\text{KrawtchoukLP}_{\text{Lin}}(n, d, \ell)$  and thus  $\text{val}(\text{KrawtchoukLP}_{\text{Lin}}(n, d, \ell))^{1/\ell} \geq A_2^{\text{Lin}}(n, d)$ . ◀

### 3.4 Properties of higher-order Krawtchouk polynomials

In this section, we explore more properties of the higher-order Krawtchouk polynomials in order to show that the objective and restrictions of the linear programs  $\text{KrawtchoukLP}(n, d, \ell)$  and  $\text{KrawtchoukLP}_{\text{Lin}}(n, d, \ell)$  can be algorithmically computed in  $O(n^{2^{\ell+1}-2})$  time for a fixed  $\ell \in \mathbb{N}_+$  (see Proposition 25).

Even though symmetric difference configurations are more natural from the point of view of linear codes, for computations and properties with the higher-order Krawtchouk polynomials, it is more convenient to work with Venn diagram configurations defined below. In plain English, each word  $z \in \mathbb{F}_2^n$  induces a partition of  $[n]$  into its support  $\text{supp}(z) := \{i \in [n] \mid z_i \neq 0\}$  and its complement  $[n] \setminus \text{supp}(z)$ ; the Venn diagram configuration of a tuple  $(z_1, \dots, z_\ell) \in (\mathbb{F}_2^n)^\ell$  then encodes the information about the sizes of each of the cells of the Venn diagram of the coarsest common refinement of the partitions induced by the  $z_i$ .

► **Definition 17.** *The Venn diagram configuration of the  $\ell$ -tuple  $(z_1, \dots, z_\ell) \in (\mathbb{F}_2^n)^\ell$  is the function  $\text{Config}_{n,\ell}^V(z_1, \dots, z_\ell): 2^{[\ell]} \rightarrow \mathbb{R}$  defined by*

$$\begin{aligned} \text{Config}_{n,\ell}^V(z_1, \dots, z_\ell)(J) &:= \left| \bigcap_{j \in J} \text{supp}(z_j) \cap \bigcap_{j \in [\ell] \setminus J} ([n] \setminus \text{supp}(z_j)) \right| \\ &= \left| \left\{ i \in [n] \mid \{j \in [\ell] \mid (z_j)_i = 1\} = J \right\} \right|, \end{aligned}$$

for every  $J \subseteq [\ell]$ .

By viewing  $\text{Config}_{n,\ell}^V$  as a function  $(\mathbb{F}_2^n)^\ell \rightarrow \mathbb{R}^{2^{[\ell]}}$ , the set of (valid) Venn diagram configurations of  $\ell$ -tuples of elements of  $\mathbb{F}_2^n$  is  $\text{im}(\text{Config}_{n,\ell}^V)$ .

The next lemma gives an easy description of the set of Venn diagram configurations of  $\ell$ -tuples of elements of  $\mathbb{F}_2^n$  as the set of all functions  $2^{[\ell]} \rightarrow \mathbb{R}$  whose values are non-negative integers that add up to  $n$ . Combining it with Lemma 19 below gives an explicit description of the set of symmetric difference configurations.

► **Lemma 18.** *For every  $n, \ell \in \mathbb{N}_+$ , we have*

$$\text{im}(\text{Config}_{n,\ell}^V) = \left\{ g: 2^{[\ell]} \rightarrow \mathbb{R} \mid \sum_{J \subseteq [\ell]} g(J) = n \wedge \forall J \subseteq [\ell], g(J) \in \mathbb{N} \right\}. \quad (2)$$

The next lemma provides a pair of linear transformations that transform a symmetric difference configuration into a Venn diagram configuration and vice-versa.

► **Lemma 19.** *Let  $n, \ell \in \mathbb{N}_+$ , let*

$$S_{n,\ell} \stackrel{\text{def}}{=} \left\{ g \in \mathbb{R}^{2^{[\ell]}} \mid \sum_{J \subseteq [\ell]} g(J) = n \right\}, \quad Z_{n,\ell} \stackrel{\text{def}}{=} \{g \in \mathbb{R}^{2^{[\ell]}} \mid g(\emptyset) = 0\}$$

and let  $V_{n,\ell}: Z_{n,\ell} \rightarrow S_{n,\ell}$  and  $D_{n,\ell}: S_{n,\ell} \rightarrow Z_{n,\ell}$  be given by

$$D_{n,\ell}(g)(J) \stackrel{\text{def}}{=} \sum_{\substack{T \subseteq [\ell] \\ |T \cap J| \text{ odd}}} g(T), \quad (3)$$

$$V_{n,\ell}(g)(J) \stackrel{\text{def}}{=} n \cdot \mathbb{1}[J = \emptyset] + 2^{1-\ell} \sum_{T \subseteq [\ell]} (-1)^{|T \cap J|} g(T), \quad (4)$$

for every  $J \subseteq [\ell]$ .

Then  $V_{n,\ell}$  and  $D_{n,\ell}$  are inverses of each other and  $\text{Config}_{n,\ell}^\Delta = D_{n,\ell} \circ \text{Config}_{n,\ell}^V$  and  $\text{Config}_{n,\ell}^V = V_{n,\ell} \circ \text{Config}_{n,\ell}^\Delta$ .

## 51:12 A Complete Linear Programming Hierarchy for Linear Codes

Making use of Venn diagram configurations, we can also easily compute the number of  $\ell$ -tuples with a given configuration as a multinomial.

► **Lemma 20.** *For a symmetric difference configuration  $g \in \text{im}(\text{Config}_{n,\ell}^\Delta)$ , we have*

$$|g| = K_g(0) = \binom{n}{V_{n,\ell}(g)} = \frac{n!}{\prod_{J \subseteq [\ell]} V_{n,\ell}(g)(J)!},$$

where  $V_{n,\ell}$  is given by (4).

The following lemma says that, similarly to the univariate case, the higher-order Krawtchouk polynomials are orthogonal with respect to the natural discrete measure on symmetric configurations in which each  $g \in \text{im}(\text{Config}_{n,\ell}^\Delta)$  has measure  $|g| = \binom{n}{V_{n,\ell}(g)}$  (see Lemma 20), i.e., the number of  $\ell$ -tuples with configuration  $g$ .

► **Lemma 21 (Orthogonality).** *For  $n, \ell \in \mathbb{N}_+$  and  $h, h' \in \text{im}(\text{Config}_{n,\ell}^\Delta)$ , we have*

$$\sum_{g \in \text{im}(\text{Config}_{n,\ell}^\Delta)} |g| \cdot K_h(g) \cdot K_{h'}(g) = 2^{\ell n} \cdot |h| \cdot \mathbb{1}[h = h'].$$

Also similarly to the univariate case, the higher-order Krawtchouk polynomials satisfy the following reflection property.

► **Lemma 22 (Reflection).** *For  $n, \ell \in \mathbb{N}_+$  and  $g, h \in \text{im}(\text{Config}_{n,\ell}^\Delta)$ , we have*

$$\frac{K_h(g)}{|h|} = \frac{K_g(h)}{|g|}.$$

The next lemma provides an alternative formula for the higher-order Krawtchouk polynomial in which the sum involves only  $O(n^{2^{2\ell}})$  terms (as opposed to the  $2^{\ell n}$  terms in (1)).

► **Lemma 23.** *For every  $n, \ell \in \mathbb{N}_+$  and every  $g, h \in \text{im}(\text{Config}_{n,\ell}^\Delta)$ , we have*

$$K_h(g) = \sum_{F \in \mathcal{F}} \prod_{J \subseteq [\ell]} \frac{V_{n,\ell}(g)(J)!}{\prod_{K \subseteq [\ell]} F(J, K)!} \cdot \prod_{j=1}^{\ell} \prod_{\substack{J, K \subseteq [j] \\ j \in J \cap K}} (-1)^{F(J, K)},$$

where  $\mathcal{F}$  is the set of functions  $F: 2^{[\ell]} \times 2^{[\ell]} \rightarrow \{0, 1, \dots, n\}$  such that

$$\begin{aligned} \forall J \subseteq [\ell], \quad \sum_{K \subseteq [\ell]} F(J, K) &= V_{n,\ell}(g)(J), \\ \forall K \subseteq [\ell], \quad \sum_{J \subseteq [\ell]} F(J, K) &= V_{n,\ell}(h)(K), \end{aligned}$$

and  $V_{n,\ell}$  is given by (4).

The next lemma allows the computation of the Krawtchouk polynomials even faster via dynamic programming.

► **Lemma 24.** *Let  $n, \ell \in \mathbb{N}_+$  with  $n \geq 2$ , let  $g, h \in \text{im}(\text{Config}_{n,\ell}^\Delta)$  be symmetric difference configurations and let  $J_0 \subseteq [\ell]$  be such that  $V_{n,\ell}(g)(J_0) > 0$  for  $V_{n,\ell}$  given by (4). Then*

$$K_h(g) = \sum_{\substack{K_0 \subseteq [\ell] \\ V_{n,\ell}(h)(K_0) > 0}} (-1)^{|J_0 \cap K_0|} \cdot K_{h \ominus K_0}(g \ominus J_0), \quad (5)$$

$$K_h(g) = - \sum_{\substack{K_0 \subseteq [\ell] \\ V(h)(K_0) > 0 \\ K_0 \neq \emptyset}} K_{h \oplus \emptyset \ominus K_0}(g) + \sum_{\substack{K_0 \subseteq [\ell] \\ V(h)(K_0) > 0}} (-1)^{|J_0 \cap K_0|} \cdot K_{h \oplus \emptyset \ominus K_0}(g \oplus \emptyset \ominus J_0), \quad (6)$$

where

$$\begin{aligned} h \ominus K_0 &:= D_{n-1,\ell}(V_{n,\ell}(h) - \mathbb{1}_{\{K_0\}}), & g \ominus J_0 &:= D_{n-1,\ell}(V_{n,\ell}(g) - \mathbb{1}_{\{J_0\}}), \\ h \oplus \emptyset &:= D_{n+1,\ell}(V_{n,\ell}(h) + \mathbb{1}_{\{\emptyset\}}), & g \oplus \emptyset &:= D_{n+1,\ell}(V_{n,\ell}(g) + \mathbb{1}_{\{\emptyset\}}), \end{aligned}$$

and  $D_{n-1,\ell}$  and  $D_{n+1,\ell}$  are given by (3).

► **Proposition 25.** *The objective and restrictions of the linear programs  $\text{KrawtchoukLP}(n, d, \ell)$  and  $\text{KrawtchoukLP}_{\text{Lin}}(n, d, \ell)$  can be algorithmically computed in  $O(n^{2^{\ell+1}-2})$  time for a fixed  $\ell \in \mathbb{N}_+$ .*

**Proof.** The number of variables and restrictions of these linear programs is the number of configurations at level  $\ell$ , which is  $O(n^{2^\ell-1})$  by Lemma 7. Furthermore, converting between symmetric difference configurations and Venn diagram configurations using Lemma 19 can be done in time  $O(2^\ell) = O(1)$  and using Lemma 23 and (6) in Lemma 24, we can compute all values of all Krawtchouk polynomials of level  $\ell$  in time  $O((n^{2^\ell-1})^2) = O(n^{2^{\ell+1}-2})$ . ◀

## 4 Unsymmetrized Formulations of the Krawtchouk Hierarchies

In this section we give other formulations for  $\text{KrawtchoukLP}$ . These formulations are *unsymmetrized* versions of the same hierarchy. Working with the unsymmetrized hierarchy can be easier, since it avoids the technical definitions of the Krawtchouk polynomials  $K_h(g)$ , but computationally the number of variables and constraints of these hierarchies is huge.

### 4.1 The Hierarchy as Checking Non-negativity of Fourier Coefficients

The LP hierarchy for linear codes can be simply described as checking non-negativity of products of Fourier coefficients. Define the linear programming hierarchy  $\text{FourierLP}_{\text{Lin}}(n, d, \ell)$  with the variables  $a_x$  ( $x \in (\mathbb{F}_2^n)^\ell$ ):

$$\begin{aligned} \max \quad & \sum_{x \in (\mathbb{F}_2^n)^\ell} a_x \\ \text{s.t.} \quad & a_0 = 1 && \text{(Normalization)} \\ & a_{(x_1, \dots, x_\ell)} = 0 \quad \exists w \in \text{span}(x_1, \dots, x_\ell), |w| \in \{1, \dots, d-1\} && \text{(Distance constraints)} \\ & \sum_{x \in (\mathbb{F}_2^n)^\ell} a_x \chi_\alpha(x) \geq 0 \quad \forall \alpha \in (\mathbb{F}_2^n)^\ell && \text{(Fourier coefficients)} \\ & a_x \geq 0 \quad \forall x \in (\mathbb{F}_2^n)^\ell && \text{(Non-negativity)}. \end{aligned}$$

► **Proposition 26.** *For every  $n, \ell \in \mathbb{N}_+$  and  $d \in \{0, 1, \dots, n\}$ ,  $\text{val}(\text{FourierLP}_{\text{Lin}}(n, d, \ell)) \geq A_2^{\text{Lin}}(n, d)^\ell$ .*

## 51:14 A Complete Linear Programming Hierarchy for Linear Codes

The corresponding hierarchy for non-linear codes  $\text{FourierLP}(n, d, \ell)$  is defined over the variables  $a_x$  ( $x \in (\mathbb{F}_2^n)^\ell$ ) as:

$$\begin{aligned}
 \max \quad & \sum_{x \in (\mathbb{F}_2^n)^\ell} a_x \\
 \text{s.t.} \quad & a_0 = 1 && \text{(Normalization)} \\
 & a_{(x_1, \dots, x_\ell)} = 0 && \exists i \in [\ell], |x_i| \in \{1, \dots, d-1\} \quad \text{(Distance constraints)} \\
 & \sum_{x \in (\mathbb{F}_2^n)^\ell} a_x \chi_\alpha(x) \geq 0 && \forall \alpha \in (\mathbb{F}_2^n)^\ell \quad \text{(Fourier coefficients)} \\
 & a_x \geq 0 && \forall x \in (\mathbb{F}_2^n)^\ell \quad \text{(Non-negativity)}.
 \end{aligned}$$

It turns out that  $\text{KrawtchoukLP}$  is a symmetrization of  $\text{FourierLP}$  (and likewise for the programs  $\text{KrawtchoukLP}_{\text{Lin}}$  and  $\text{FourierLP}_{\text{Lin}}$ ). We will briefly describe the technique of symmetrizing convex programs, which is also described in the survey article by Vallentin [18]. The proof that  $\text{KrawtchoukLP}$  and  $\text{FourierLP}$  are equivalent continues at Proposition 30.

The technique exploits the fact that convex relaxations for the independence number  $\alpha(H_{n,d})$  of the Hamming cube graph  $H_{n,d}$  of distance less than  $d$  are highly symmetric, that is, programs that are invariant under large permutation groups as defined below.

► **Definition 27 (Program invariance).** *Let  $\mathcal{P}$  be a linear program with variables  $(a_x)_{x \in X}$  for some set  $X$ . We say that  $\mathcal{P}$  is invariant under a permutation  $\sigma$  of  $X$  if for all feasible solutions  $(a_x)$ , the point  $a \cdot \sigma$  defined by  $(a \cdot \sigma)_x := a_{\sigma(x)}$  is also feasible, and the objective value is the same.*

*Similarly, a semi-definite program  $\mathcal{P}$  with variable  $M \in \mathbb{R}^{X \times X}$  is invariant under  $\sigma$  if for all feasible  $M$ , the matrix  $M \cdot \sigma$  defined by  $(M \cdot \sigma)[x, y] := M[\sigma(x), \sigma(y)]$  is also feasible, and the objective value is the same.*

*The group of permutations of  $X$  under which  $\mathcal{P}$  is invariant is called the automorphism group of  $\mathcal{P}$  and is denoted  $\text{Aut}(\mathcal{P})$ .*

If the input of a program  $\mathcal{P}$  is a graph  $G$  and the program only depends on the isomorphism class of  $G$ , then the program is invariant under the automorphism group  $\text{Aut}(G)$  of the graph  $G$ . For convex relaxations such as the Lovász  $\vartheta$ -function or the Sum-of-Squares hierarchy, the variables of the program are indexed by tuples of vertices from  $G$ , and thus a case of interest is when  $\text{Aut}(G)$  acts diagonally on tuples of vertices.

By symmetrizing solutions, i.e., by averaging the values of the variables over the automorphism group  $\text{Aut}(\mathcal{P})$ , we may assume that the solution has the same symmetry:

► **Fact 28.** *For any  $H \subseteq \text{Aut}(\mathcal{P})$ , the value  $\text{val}(\mathcal{P})$  equals the value of  $\mathcal{P}$  with the additional constraints  $\forall \sigma \in H, \forall x \in X, a_x = a_{\sigma(x)}$  (or  $\forall \sigma \in H, \forall x, y \in X, M[x, y] = M[\sigma(x), \sigma(y)]$  for an SDP).*

A symmetrized solution is constant on each orbit of the group action on  $X$  or  $X^2$ . Therefore, the “effective” number of variables in the convex program is only the number of orbits, which may be significantly smaller than even  $|V(G)|$ .

For example, the graph  $H_{n,d}$  has a large symmetry group:

► **Fact 29.** *For  $1 < d < n$ ,  $\text{Aut}(H_{n,d})$  is the hyperoctahedral group, which is the semidirect product  $\mathbb{F}_2^n \rtimes S_n$  in which  $S_n$  permutes the coordinates and  $\mathbb{F}_2^n$  applies a bit flip.*

Even though the hypercube has size  $2^n$  and thus  $|V(H_{n,\ell})^\ell| = 2^{n\ell}$ , the number of orbits of the diagonal action of  $\text{Aut}(H_{n,d})$  on  $\ell$ -tuples is only  $\text{poly}(n)$  for constant  $\ell$ . For example, for  $\ell = 4$ , viewing the hypercube momentarily as  $\{-1, +1\}^n$ , the orbit of  $(x_1, x_2, x_3, x_4)$  essentially only depends on the angles between the vectors: it is determined by the seven numbers

$$\langle x_1, x_2 \rangle, \langle x_1, x_3 \rangle, \langle x_1, x_4 \rangle, \langle x_2, x_3 \rangle, \langle x_2, x_4 \rangle, \langle x_3, x_4 \rangle, \sum_{i=1}^n x_{1,i}x_{2,i}x_{3,i}x_{4,i}. \quad (7)$$

Equivalently, it is determined by  $\text{Config}_{n,\ell}^\Delta(x_2 - x_1, x_3 - x_1, x_4 - x_1)$  (see Lemma 8).

Since each of the numbers in (7) takes at most  $n + 1$  values, the effective number of variables in the degree-4 Sum-of-Squares relaxation for  $\alpha(H_{n,d})$  is at most  $O(n^7)$ . Thus, the search for an upper bound on an exponential-size object is reduced to a polynomial-size convex program! Of course, to actually run this in polynomial time, one also needs to show that this polynomial-size convex program can be computed in polynomial time (which rules out explicitly computing the original program then taking a quotient).

We use the symmetrization technique to show that  $\text{KrawtchoukLP}$  and  $\text{FourierLP}$  are equivalent.

► **Proposition 30.** *For every  $n, \ell \in \mathbb{N}_+$  and every  $d \in \{0, 1, \dots, n\}$ , we have*

$$\begin{aligned} \text{val}(\text{FourierLP}(n, d, \ell)) &= \text{val}(\text{KrawtchoukLP}(n, d, \ell)), \\ \text{val}(\text{FourierLP}_{\text{Lin}}(n, d, \ell)) &= \text{val}(\text{KrawtchoukLP}_{\text{Lin}}(n, d, \ell)). \end{aligned}$$

► **Remark 31.** The linear programs  $\text{FourierLP}$  and  $\text{FourierLP}_{\text{Lin}}$  are invariant under  $S_n$  but are *not* invariant under the other automorphisms of the hypercube of the form  $x \mapsto x + z$  ( $z \in \mathbb{F}_2^n$ ), because of the normalization constraint and the distance constraints. It makes more sense to view the underlying space as  $\mathbb{F}_2^n$  instead of the hypercube, which does not have the  $\mathbb{F}_2^n$  automorphism because the origin is treated specially.

There is actually more symmetry in the programs than just  $S_n$ . In the case of the program for non-linear codes, there is a symmetry under the right action of  $S_\ell$  on  $(\mathbb{F}_2^n)^\ell$  that permutes the *words*  $x_1, \dots, x_\ell$ , that is, we have  $(x_1, \dots, x_\ell) \cdot \tau := (x_{\tau(1)}, \dots, x_{\tau(\ell)})$  ( $(x_1, \dots, x_\ell) \in (\mathbb{F}_2^n)^\ell$ ,  $\tau \in S_\ell$ ). In the case of the program for linear codes, we have symmetry under the action of  $\text{GL}_\ell(\mathbb{F}_2)$  that applies a basis change to  $(x_1, \dots, x_\ell)$ , that is, it is given by

$$(A \cdot x)_i := \sum_{j \in [\ell]} A[i, j] \cdot x_j \in \mathbb{F}_2^n$$

for every  $A \in \text{GL}_\ell(\mathbb{F}_2)$ , every  $x \in (\mathbb{F}_2^n)^\ell$  and every  $i \in [\ell]$ . The distance constraints are evidently invariant under this action as it does not change the linear subspace spanned by  $(x_1, \dots, x_\ell)$ . The Fourier constraints are invariant since

$$\chi_\alpha(A \cdot x) = \chi_{A^\top \cdot \alpha}(x)$$

for every  $x, \alpha \in \mathbb{F}_2^\ell$ .

Note that the actions of  $\text{GL}_\ell(\mathbb{F})$  and  $S_n$  commute with each other and thus induce an action of the direct product  $\text{GL}_\ell(\mathbb{F}) \times S_n$ . Another reasonable definition of the higher-order Krawtchouk polynomials and linear program symmetrizes under this larger group action of  $\text{GL}_\ell(\mathbb{F}) \times S_n$ . There is one Krawtchouk polynomial and one free variable for each orbit of this action.

## 51:16 A Complete Linear Programming Hierarchy for Linear Codes

► **Definition 32** (Fully symmetrized higher-order Krawtchouks). *Let  $O := (\mathbb{F}_2^n)^\ell / (\text{GL}_\ell(\mathbb{F}_2) \times S_n)$  be the set of orbits of the  $(\text{GL}_\ell(\mathbb{F}_2) \times S_n)$ -action as above. For each  $h \in O$  we define the higher-order Krawtchouk polynomial  $K_h: O \rightarrow \mathbb{R}$  by*

$$K_h(g) := \sum_{(\alpha_1, \dots, \alpha_\ell) \in h} \prod_{j=1}^{\ell} \chi_{\alpha_j}(x_j),$$

where  $(x_1, \dots, x_\ell)$  is any element in the orbit  $g \in O$ .

Since the symmetry group is larger and the number of orbits is smaller, the size of the resulting LP is smaller. However, since  $|\text{GL}_\ell(\mathbb{F}_2)| = \prod_{t=0}^{\ell-1} (2^\ell - 2^t) = O_\ell(1)$ , for a constant  $\ell$ , this would only decrease the size of KrawtchoukLP by a constant factor. For practical computations, constant factors make a difference and this symmetrization should likely be performed. We chose our definition of Krawtchouks in Section 3 because the orbits are simpler to describe (being captured by explicit combinatorial objects, configuration functions) and we can compute the set of orbits and the Krawtchouk polynomials efficiently (see Proposition 25).

There is an equivalent interpretation of  $(\text{GL}_\ell(\mathbb{F}_2) \times S_n)$ -orbits as “subspace weight profiles” as follows. The right action of  $S_n$  naturally induces an action over linear subspaces of  $\mathbb{F}_2^n$  given by

$$W \cdot \sigma := \{w \cdot \sigma \mid w \in W\} \quad (W \leq \mathbb{F}_2^n, \sigma \in S_n).$$

It is straightforward to see that two  $\ell$ -tuples  $(x_1, \dots, x_\ell), (y_1, \dots, y_\ell) \in (\mathbb{F}_2^n)^\ell$  are in the same  $(\text{GL}_\ell(\mathbb{F}_2) \times S_n)$ -orbit if and only if  $\text{span}\{x_1, \dots, x_\ell\}$  and  $\text{span}\{y_1, \dots, y_\ell\}$  are in the same  $S_n$ -orbit, which in turn is equivalent to saying that both spaces have the same dimension, say  $k$ , and there are ordered bases  $b^x = (b_1^x, \dots, b_k^x)$  and  $b^y = (b_1^y, \dots, b_k^y)$  of these spaces respectively such that  $\text{Config}_{n,k}^\Delta(b^x) = \text{Config}_{n,k}^\Delta(b^y)$ . Thus, the hierarchy corresponding to the  $(\text{GL}_\ell(\mathbb{F}_2) \times S_n)$ -action has an interesting interpretation as measuring weight statistics of linear subspaces of the linear code of dimension at most  $\ell$ .

### 4.2 The Hierarchy as an SDP

The LP hierarchy is also equivalent to an SDP relaxation with the harsh constraint that the SDP matrix must be *translation invariant*.

Define the semi-definite program  $\text{TranslationSDP}(n, d, \ell)$  as

$$\begin{aligned} \max \quad & \sum_{x \in (\mathbb{F}_2^n)^\ell} M[0, x] \\ \text{s.t.} \quad & M[0, 0] = 1 && \text{(Normalization)} \\ & M[0, (x_1, \dots, x_\ell)] = 0 \quad \exists i \in [\ell], |x_i| \in \{1, \dots, d-1\} && \text{(Distance constraints)} \\ & M[x, y] = M[0, y-x] \quad \forall x, y \in (\mathbb{F}_2^n)^\ell && \text{(Translation symmetry)} \\ & M \succeq 0 && \text{(PSD-ness)} \\ & M[x, y] \geq 0 \quad \forall x, y \in (\mathbb{F}_2^n)^\ell && \text{(Non-negativity),} \end{aligned}$$

where the variable is  $M \in \mathbb{R}^{(\mathbb{F}_2^n)^\ell \times (\mathbb{F}_2^n)^\ell}$ .

To form  $\text{TranslationSDP}_{\text{Lin}}(n, d, \ell)$ , replace the distance constraints by

$$M[0, (x_1, \dots, x_\ell)] = 0 \quad \exists w \in \text{span}(x_1, \dots, x_\ell), |w| \in \{1, \dots, d-1\}.$$



The crucial translation symmetry property of TranslationSDP ensures  $M$  lies in the commutative matrix algebra  $\text{span}\{D_z \mid z \in (\mathbb{F}_2^n)^\ell\}$ , where

$$D_z[x, y] := \mathbb{1}[y - x = z].$$

The coefficient of  $M$  on  $D_z$  is  $M[0, z]$ .

Since the matrices  $D_z$  commute, they are simultaneously diagonalizable. More specifically, their common eigenvectors are the Fourier characters.

► **Fact 33.** *The matrices  $D_z$  are simultaneously diagonalized by  $(\chi_\alpha \mid \alpha \in (\mathbb{F}_2^n)^\ell)$  with the eigenvalue of  $D_z$  on  $\chi_\alpha$  being  $\chi_\alpha(z)$ .*

Therefore, the PSD-ness constraint in TranslationSDP is particularly simple: to check that  $\lambda_z D_z \succcurlyeq 0$ , it is equivalent to check  $\sum_{z \in (\mathbb{F}_2^n)^\ell} \lambda_z \chi_\alpha(z) \geq 0$  for all  $\alpha \in (\mathbb{F}_2^n)^\ell$ . This is a linear constraint on the  $\lambda_z$ , and hence we can express the SDP as an LP, giving yet another formulation of the hierarchy.

► **Proposition 34.** *For every  $n, \ell \in \mathbb{N}_+$  and every  $d \in \{0, 1, \dots, n\}$ , we have*

$$\begin{aligned} \text{val}(\text{FourierLP}(n, d, \ell)) &= \text{val}(\text{TranslationSDP}(n, d, \ell)), \\ \text{val}(\text{FourierLP}_{\text{Lin}}(n, d, \ell)) &= \text{val}(\text{TranslationSDP}_{\text{Lin}}(n, d, \ell)). \end{aligned}$$

► **Remark 35.** In previous convex relaxations for  $A_2(n, d)$ , in order to implement the program efficiently, a key technical step has been finding an explicit block diagonalization of the SDP matrix (which reduces the program size). This step requires significant technical work [17, 5, 6]. An advantage of the LP hierarchy is that complete diagonalization is trivial.

### 4.3 The Hierarchy as $\vartheta'$

The hierarchy can also be seen as computing the (modified) Lovász  $\vartheta'$  function on progressively larger graphs, whose definition is recalled below. In fact, this formulation of the hierarchy holds for any *association scheme* (see full version of the paper).

► **Definition 36** ( $\vartheta'$  Program). *The (modified) Lovász  $\vartheta'$  function is defined as follows. For a graph  $G$ ,  $\vartheta'(G)$  is the optimum value of the semi-definite program  $\mathcal{S}(G)$  given by*

$$\begin{aligned} \max \quad & \langle J, M \rangle \\ \text{s.t.} \quad & \text{tr } M = 1 && \text{(Normalization)} \\ & M[u, v] = 0 && \forall \{u, v\} \in E(G) \quad \text{(Independent set)} \\ & M \succcurlyeq 0 && \text{(PSD-ness)} \\ & M[u, v] \geq 0 && \forall u, v \in V(G) \quad \text{(Non-negativity)}, \end{aligned}$$

where the variable is  $M \in \mathbb{R}^{V \times V}$  symmetric,  $J$  is the all ones matrix and  $\langle A, B \rangle := \text{tr}(A^\top B)$ .

By strong duality  $\vartheta'(G)$  is also the optimum value of the dual semi-definite program  $\mathcal{S}'(G)$  given by

$$\begin{aligned} \min \quad & \beta \\ \text{s.t.} \quad & \beta I - N \succcurlyeq 0 && \text{(PSD-ness)} \\ & N[u, v] \geq 1 && \forall u, v \in V(G) \text{ with } \{u, v\} \notin E \quad \text{(Independent set)}, \end{aligned}$$

where the variables are  $N \in \mathbb{R}^{V \times V}$  symmetric and  $\beta \in \mathbb{R}$ .

## 51:18 A Complete Linear Programming Hierarchy for Linear Codes

It is straightforward to see that  $\vartheta'(G)$  is an upper bound for the independence number of the graph  $G$  since if  $A \subseteq V(G)$  is an independent set, then  $\mathbb{1}_A \mathbb{1}_A^\top / |A|$  is a feasible solution of  $\mathcal{S}(G)$  with value  $|A|$ .

In the same way that a code  $C \subseteq \mathbb{F}_2^n$  of distance at least  $d$  can be seen as an independent set in the graph  $H_{n,d}$ , we can see  $C^\ell$  as an independent set in exclusion graphs defined below based on the sets  $\text{ForbConfig}(n, d, \ell)$  and  $\text{ForbConfig}_{\text{Lin}}(n, d, \ell)$  of Definition 15.

► **Definition 37** (Exclusion Graph). *We define the exclusion graph  $H_{n,d,\ell}$  to have vertex set  $(\mathbb{F}_2^n)^\ell$  and edge set*

$$E(H_{n,d,\ell}) := \left\{ (x, y) \in \binom{(\mathbb{F}_2^n)^\ell}{2} \mid \text{Config}_{n,\ell}^\Delta(x - y) \in \text{ForbConfig}(n, d, \ell) \right\}.$$

*We define  $H_{n,d,\ell}^{\text{Lin}}$  analogously replacing  $\text{ForbConfig}(n, d, \ell)$  with  $\text{ForbConfig}_{\text{Lin}}(n, d, \ell)$ .*

► **Lemma 38.** *For every  $n, \ell \in \mathbb{N}_+$  and every  $d \in \{0, 1, \dots, n\}$ , we have*

$$\begin{aligned} \text{val}(\text{TranslationSDP}(n, d, \ell)) &= \text{val}(\vartheta'(H_{n,d,\ell})), \\ \text{val}(\text{TranslationSDP}_{\text{Lin}}(n, d, \ell)) &= \text{val}(\vartheta'(H_{n,d,\ell}^{\text{Lin}})). \end{aligned}$$

## 5 Main Properties of the Krawtchouk Hierarchies

This section presents our main results on the linear programming hierarchy. The first result is the completeness of the higher-order linear programming hierarchies for *linear* codes. The second result is the collapse of the hierarchies for *general* codes.

### 5.1 Completeness for Linear Codes

In this section, we show the (approximate) completeness of our linear programming hierarchy for linear codes over a finite field  $\mathbb{F}$ .

We will show completeness at level  $O(n^2)$  via a counting argument. The intuition is that the hierarchy is likely already complete at level  $n$  (and we conjecture this to be the case). At level  $n$ , the feasible region of the LP already encodes  $A_q^{\text{Lin}}(n, d)$ . That is, since at level  $n$  there is a variable for each possible basis of a subspace of  $\mathbb{F}_q^n$ , just writing down the distance constraints of  $\text{KrawtchoukLP}_{\text{Lin}}^{\mathbb{F}}(n, d, n)$  allows one to deduce the true value of  $A_q^{\text{Lin}}(n, d)$ . Of course this property is not sufficient to imply that the *value* of  $\text{KrawtchoukLP}_{\text{Lin}}^{\mathbb{F}}(n, d, n)$  is correct. At an intuitive level, the below proof shows that at level  $O(n^2)$  the large-dimensional subspaces outweigh the small-dimensional subspaces enough to deduce the correct value of  $A_q^{\text{Lin}}(n, d)$ .

► **Theorem 39** (Completeness). *Let  $\mathbb{F}$  be a finite field, let  $q := |\mathbb{F}|$ , let  $\varepsilon \in (0, 1)$  and let  $\ell \geq 9(n^2 \ln(q) + 1) / (\ln(1 + \varepsilon))^2$ . Then for every  $d \in \{0, 1, \dots, n\}$ , we have*

$$\text{val}(\text{KrawtchoukLP}_{\text{Lin}}^{\mathbb{F}}(n, d, \ell))^{1/\ell} \leq (1 + \varepsilon) \cdot A_q^{\text{Lin}}(n, d).$$

Before proving this theorem, note that since  $\mathbb{F}$ -linear codes must necessarily have size of the form  $q^k$  for some  $k \in \mathbb{N}$ , by taking  $\varepsilon < q - 1$ , we get

$$A_q^{\text{Lin}}(n, d) = q^{\lfloor (\log_q \text{val}(\text{KrawtchoukLP}_{\text{Lin}}^{\mathbb{F}}(n, d, \ell))) / \ell \rfloor}$$

whenever  $\ell > 9(n^2 \ln(q) + 1) / (\ln(q))^2$ .

**Proof.** For simplicity, we only consider the binary case  $q = 2$  (see the full version for the general case<sup>2</sup>). By Lemma 38 we have  $\text{val}(\text{KrawtchoukLP}_{\text{Lin}}^{\mathbb{F}}(n, d, \ell)) = \text{val}(\vartheta'(G))$  where  $G = H_{n,d,\ell}^{\text{Lin}}$ . Recall that  $\text{val}(\vartheta'(G))$  is the optimum value of the semi-definite program

$$\begin{aligned} \max \quad & \langle J, M \rangle \\ \text{s.t.} \quad & \text{tr } M = 1 && \text{(Normalization)} \\ & M[u, v] = 0 && \forall \{u, v\} \in E(G) \quad \text{(Independent set)} \\ & M \succ 0 && \text{(PSD-ness)} \\ & M[u, v] \geq 0 && \forall u, v \in V(G) \quad \text{(Non-negativity),} \end{aligned}$$

where the variables is  $M \in \mathbb{R}^{V \times V}$  symmetric and

$$E(G) := \left\{ \{x, y\} \in \binom{(\mathbb{F}^n)^\ell}{2} \mid \forall k \in \mathbb{F}^\ell, \left| \sum_{i=1}^{\ell} k_i(x_i - y_i) \right| \notin [d-1] \right\},$$

where  $|z| := |\{j \in [n] \mid z_j \neq 0\}|$  is the Hamming weight of  $z$ .

Let  $k_0$  be the maximum dimension of an  $\mathbb{F}$ -linear code of distance  $d$  in  $\mathbb{F}^n$  (that is, let  $k_0 := \log_q A_q^{\text{Lin}}(n, d)$ ), let  $M$  be a feasible solution of the program above and let us provide an upper bound for the objective value  $\langle J, M \rangle$ . Note that symmetrizing  $M$  under the automorphism group  $\text{Aut}(G)$  of the Cayley graph  $G$  does not change the objective value  $\langle J, M \rangle$  (and preserves all restrictions), so we may suppose that  $M$  is  $\text{Aut}(G)$ -invariant, which in particular implies that all diagonal entries of  $M$  are equal and since the trace of  $M$  is 1, it follows that all diagonal entries of  $M$  are equal to  $q^{-n\ell}$ . On the other hand, since  $M$  is positive semi-definite, any  $2 \times 2$  principal minor of  $M$  is non-negative and thus all off-diagonal entries of  $M$  have absolute value at most  $q^{-n\ell}$ , that is, we have  $\|M\|_\infty = q^{-n\ell}$ .

Since the objective value  $\langle J, M \rangle$  is simply the sum of all entries of  $M$ , we can provide an upper bound for it by simply giving an upper bound on how many entries of  $M$  are allowed to be non-zero.

Note that for an entry  $M_{xy}$  indexed by  $(x, y) \in (\mathbb{F}^n)^\ell \times (\mathbb{F}^n)^\ell$  to be non-zero, the difference vectors  $z_1, \dots, z_\ell \in \mathbb{F}^n$  given by  $z_i := x_i - y_i$  ( $i \in [\ell]$ ) must span an  $\mathbb{F}$ -linear subspace of dimension at most  $k_0$  (as any subspace of larger dimension necessarily has distance smaller than  $d$  and thus some  $k \in \mathbb{F}^\ell$  will have  $|\sum_{i=1}^{\ell} k_i(x_i - y_i)| \in [d-1]$ ).

By letting  $\gamma_{n,\ell,k}$  be the number of tuples  $(z_1, \dots, z_\ell)$  that span a subspace of dimension  $k \in \{0, 1, \dots, n\}$ , since each difference  $(z_1, \dots, z_\ell)$  is realized as  $z_i = x_i - y_i$  for exactly  $q^{n\ell}$  pairs  $(x, y) \in (\mathbb{F}^n)^\ell \times (\mathbb{F}^n)^\ell$ , we get

$$\langle J, M \rangle \leq \sum_{k=0}^{k_0} \gamma_{n,\ell,k} \cdot q^{n\ell} \cdot \|M\|_\infty \leq \sum_{k=0}^{k_0} \gamma_{n,\ell,k}.$$

We claim that

$$\gamma_{n,\ell,k} \leq \binom{\ell}{k} \cdot \beta_{n,k} \cdot (q^k)^{\ell-k}, \quad (8)$$

where

$$\beta_{n,k} := \prod_{j=0}^{k-1} (q^n - q^j)$$

<sup>2</sup> The proof of the general case is similar.

is the number of linearly independent ordered  $k$ -tuples in  $\mathbb{F}^n$ . Indeed, the upper bound in (8) follows by picking  $k$  out of the  $\ell$  vectors to have a linearly independent ordered  $k$ -tuple, then picking each of the other  $\ell - k$  positions to be a linear combination of these  $k$  vectors.

Using this bound along with  $\binom{\ell}{k} \leq \ell^k$  and  $\beta_{n,k} \leq q^{nk}$ , we get

$$\begin{aligned} \langle J, M \rangle &\leq \sum_{k=0}^{k_0} \gamma_{n,\ell,k} \leq \sum_{k=0}^{k_0} \ell^k q^{nk} q^{k\ell} \leq \ell^{k_0} q^{nk_0} \left( q^{k_0\ell} + \sum_{k=0}^{k_0-1} q^{k\ell} \right) \\ &= \ell^{k_0} q^{nk_0} \left( q^{k_0\ell} + \frac{q^{k_0\ell} - 1}{q^\ell - 1} \right) \leq 2\ell^n q^{n^2} q^{k_0\ell}. \end{aligned}$$

Taking the  $\ell$ th root and recalling that  $q^{k_0} = A_q^{\text{Lin}}(n, d)$  we conclude that

$$\text{val}(\text{KrawtchoukLP}_{\text{Lin}}^{\mathbb{F}}(n, d, \ell))^{1/\ell} \leq (2\ell^n q^{n^2})^{1/\ell} A_q^{\text{Lin}}(n, d).$$

Finally, the hypothesis  $\ell \geq 9(n^2 \ln(q) + 1)/(\ln(1 + \varepsilon))^2$  implies that  $(2\ell^n q^{n^2})^{1/\ell} \leq 1 + \varepsilon$ , which concludes the proof (a detailed computation is included in the appendix of the full version).  $\blacktriangleleft$

## 5.2 Hierarchy Collapse for General Codes

In this section, we show that without the additional semantic linearity constraints imposed by  $\text{KrawtchoukLP}_{\text{Lin}}(n, d, \ell)$ , the associated hierarchy  $\text{KrawtchoukLP}(n, d, \ell)$  does not give any improvement over the original Delsarte linear programming approach.

► **Proposition 40** (Lifting). *For every finite field  $\mathbb{F}$  and every  $\ell \in \mathbb{N}_+$ , we have*

$$\text{val}(\text{KrawtchoukLP}^{\mathbb{F}}(n, d, \ell))^{1/\ell} = \text{val}(\text{DelsarteLP}^{\mathbb{F}}(n, d)).$$

Once we recall that our hierarchy comes from a refinement of a tensor power of an association scheme, the proof is in two steps: first, we show that just tensoring the program does not change the relative value. Second, we show that refining the scheme and adding only natural (non-semantic) constraints does not change the value of the associated Delsarte linear program.

As a secondary corollary, we can also show that in the linear case, the logarithm of the value of the hierarchy is subadditive. Let us note that this is also true of the hierarchy for non-linear codes for trivial reasons.

► **Corollary 41.** *For every finite field  $\mathbb{F}$  and every  $\ell_1, \ell_2 \in \mathbb{N}_+$ , we have*

$$\begin{aligned} \text{val}(\text{KrawtchoukLP}_{\text{Lin}}^{\mathbb{F}}(n, d, \ell_1 + \ell_2)) \\ \leq \text{val}(\text{KrawtchoukLP}_{\text{Lin}}^{\mathbb{F}}(n, d, \ell_1)) \cdot \text{val}(\text{KrawtchoukLP}_{\text{Lin}}^{\mathbb{F}}(n, d, \ell_2)), \end{aligned}$$

## 6 Conclusion

In this paper, we presented a pair of hierarchies of linear programs  $\text{KrawtchoukLP}^{\mathbb{F}}(n, d, \ell)$  and  $\text{KrawtchoukLP}_{\text{Lin}}^{\mathbb{F}}(n, d, \ell)$  that provide upper bounds for the maximum size of codes and linear codes, respectively, of distance  $d$  in the weak Hamming scheme  $\mathbb{H}_n(\mathbb{F})$  over a finite field  $\mathbb{F}$ . We showed that while the first hierarchy  $\text{KrawtchoukLP}^{\mathbb{F}}(n, d, \ell)$  collapses, the second hierarchy obtains the true value of the maximum code up to rounding by level  $\ell = O(n^2)$ . In the full version of this paper, we also showed how to extend these hierarchy constructions to translation schemes under the mild assumption of factoring through types.

As we mentioned in the introduction, we view the main contribution of  $\text{KrawtchoukLP}_{\text{Lin}}$  as being a hierarchy that is sufficiently powerful to ensure completeness while still being sufficiently simple to remain a hierarchy of linear programs (as opposed to SDPs), and bearing enough similarities with the original Delsarte's LP to be amenable to theoretical analysis. Thus the main open problem is to provide better upper or lower bounds to the optimum value of  $\text{KrawtchoukLP}_{\text{Lin}}$ .

The contrast between completeness of  $\text{KrawtchoukLP}_{\text{Lin}}$  and collapse of  $\text{KrawtchoukLP}$  also surfaces a very natural question: are optimum codes very far from being linear? Along these lines, note that at level  $\ell$ ,  $\text{KrawtchoukLP}_{\text{Lin}}(n, d, \ell)$  does not require full linearity of a code; namely, if  $C \subseteq \mathbb{F}_2^n$  satisfies

$$\Delta \left( \sum_{j=1}^t x_j, \sum_{j=1}^t y_j \right) \notin [d-1], \quad (9)$$

for every  $t \leq \ell$  and every  $x_1, \dots, x_t, y_1, \dots, y_t \in C$ , then  $a^C$  is a feasible solution of the program  $\text{KrawtchoukLP}_{\text{Lin}}(n, d, \ell)$ . For constant  $\ell$ , the condition (9) is extremely mild and much weaker than  $C$  being a linear (or even affine) code. For example, if  $0 \in C$ , then (9) boils down to requiring sums of at most  $2\ell$  codewords from  $C$  to not have Hamming weight in  $[d-1]$ . This makes studying  $\text{KrawtchoukLP}_{\text{Lin}}(n, d, \ell)$  at constant levels  $\ell$  quite interesting.

In Theorem 39, we showed the (approximate) completeness of  $\text{KrawtchoukLP}_{\text{Lin}}^{\mathbb{F}}(n, d, \ell)$  at level  $O(n^2)$ , via an unusual counting argument. We conjecture that at level  $n$  the hierarchy would have exact completeness. The hierarchy does not have the same conceptual structure as Sum-of-Squares or Sherali–Adams, so completeness does not follow in the same simple intuitive way. Another open problem is to compare the hierarchy with the Sum-of-Squares hierarchy enhanced with linear semantic constraints for linear codes. Showing that this enhanced SOS hierarchy and  $\text{KrawtchoukLP}_{\text{Lin}}^{\mathbb{F}}(n, d, \ell)$  converge to the true size of a code at a comparable rate would show that  $\text{KrawtchoukLP}_{\text{Lin}}^{\mathbb{F}}(n, d, \ell)$  (essentially) unifies other approaches based on convex programming hierarchies.

As we mentioned in the introduction, our techniques provide a higher-order version of the linear program responsible for the first linear programming bound in [12]. The second linear programming bound in [12] also consists of analyzing a Delsarte LP but for the Johnson scheme instead of the Hamming scheme. However, since the Johnson scheme is not a translation scheme, one cannot apply the theory developed in full version of this paper directly. It is then natural to ask if there is a suitable generalization of this construction that would apply to non-translation schemes such as the Johnson scheme.

---

## References

- 1 Etienne de Klerk, Dmitrii V. Pasechnik, and Alexander Schrijver. Reduction of symmetric semidefinite programs using the regular  $*$ -representation. *Math. Program.*, 109(2-3, Ser. B):613–624, 2007. doi:10.1007/s10107-006-0039-7.
- 2 P. Delsarte. *An Algebraic Approach to the Association Schemes of Coding Theory*. Philips Journal of Research / Supplement. N.V. Philips' Gloeilampenfabrieken, 1973.
- 3 P. Delsarte and V. I. Levenshtein. Association schemes and coding theory. *IEEE Transactions on Information Theory*, 44(6):2477–2504, 1998.
- 4 Joel Friedman and Jean-Pierre Tillich. Generalized Alon–Boppana theorems and error-correcting codes. *SIAM J. Discret. Math.*, 19(3):700–718, July 2005.
- 5 D. C. Gijswijt, H. D. Mittelmann, and A. Schrijver. Semidefinite code bounds based on quadruple distances. *IEEE Transactions on Information Theory*, 58(5):2697–2705, 2012.

- 6 Dion Gijswijt. Block diagonalization for algebra's associated with block codes, 2009. [arXiv:0910.4515](#).
- 7 E.N. Gilbert. A comparison of signalling alphabets. *Bell System Technical Journal*, 31:504–522, 1952.
- 8 Monique Laurent. Strengthened semidefinite programming bounds for codes. *Mathematical Programming*, 109:1436–4646, 2007.
- 9 Monique Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging Applications of Algebraic Geometry (of IMA Volumes in Mathematics and its Applications)*. Springer, 2009.
- 10 Jessie MacWilliams. A theorem on the distribution of weights in a systematic code†. *Bell System Technical Journal*, 42(1):79–94, 1963.
- 11 Mrs. F. J. MacWilliams, N. J. A. Sloane, and J.M. Goethals. The MacWilliams identities for nonlinear codes. *The Bell System Technical Journal*, 51(4):803–819, 1972.
- 12 R. McEliece, E. Rodemich, H. Rumsey, and L. Welch. New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities. *IEEE Transactions on Information Theory*, 23(2):157–166, 1977.
- 13 M. Navon and A. Samorodnitsky. On Delsarte's linear programming bounds for binary codes. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 327–336, 2005.
- 14 Michael Navon and Alex Samorodnitsky. Linear programming bounds for codes via a covering argument. *Discrete Comput. Geom.*, 41(2):199–207, March 2009.
- 15 Alex Samorodnitsky. One more proof of the first linear programming bound for binary codes and two conjectures, 2021. [arXiv:2104.14587](#).
- 16 A. Schrijver. A comparison of the Delsarte and Lovász bounds. *IEEE Transactions on Information Theory*, 25(4):425–429, 1979.
- 17 A. Schrijver. New code upper bounds from the Terwilliger algebra and semidefinite programming. *IEEE Transactions on Information Theory*, 51(8):2859–2866, 2005.
- 18 Frank Vallentin. Semidefinite programming bounds for error-correcting codes. *CoRR*, abs/1902.01253, 2019. [arXiv:1902.01253](#).
- 19 Jacobus H. van Lint. *Introduction to Coding Theory*. Springer-Verlag, 1999.
- 20 R.R. Varshamov. Estimate of the number of signals in error correcting codes. *Doklady Akademii Nauk SSSR*, 117:739–741, 1957.