Raymond and Beverly Sackler Faculty of Exact Sciences
Blavatnik School of Computer Science

Unbalanced Expanders from Multiplicity Codes

Thesis submitted in partial fulfillment
of graduate requirements for the degree
Master of Science
in the School of Computer Science, Tel-Aviv University

by
Itay Kalev

The research for this thesis has
been carried out under the supervision of
Prof. Amnon Ta-Shma

August 2022

Acknowledgments

I wish to express my deepest gratitude to my advisor, Prof. Amnon Ta-Shma, for his guidance, knowledge, and most of all for his endless time and energy dedicated to me and to this research. Without your constant help non of this would have been possible, and for that I am grateful, Thank you Amnon.
Secondly, I wish to thank all of my friends from the Arazim program, and in particular to my roommates during the writing of this thesis: Dror Frid, Nick Kushnir, and Sean Landsberg for their enjoyable company throughout this year.
Last, but definitely not least, I wish to thank my family for their constant love and support.

Abstract

In 2007 Guruswami, Umans and Vadhan gave an explicit construction of a lossless condenser based on Parvaresh-Vardy codes. This lossless condenser is a basic building block in many constructions, and, in particular, is behind the state of the art extractor constructions.

We give an alternative construction that is based on Multiplicity codes. While the bottom-line result is similar to the GUV result, the analysis is very different. In GUV (and Parvaresh-Vardy codes) the polynomial ring is closed to a finite field, and every polynomial is associated with related elements in the finite field. In our construction a polynomial from the polynomial ring is associated with its iterated derivatives. Our analysis boils down to solving a differential equation over a finite field, and uses previous techniques, introduced by Kopparty (in [Kop15]) for the list-decoding setting. We also observe that these (and more general) questions were studied in differential algebra, and we use the terminology and result developed there.

We believe these techniques have the potential of getting better constructions and solving the current bottlenecks in the area.

Contents

1 Introduction

A condenser is a probabilistic mapping from a large universe $\{0,1\}^n$ to a smaller universe $\{0,1\}^m$ that preserves the entropy of not too large sets. More formally, $C : \{0,1\}^n \times [D] \to \{0,1\}^m$ is a (k_1, k_2, ϵ) condenser, if for every distribution X on $\{0,1\}^n$ with k_1 min-entropy, the output distribution $C(X, U_D)$ is ϵ-close to having k_2 min-entropy (see definition 2.3 for a formal definition).

Ideally, we would like to explicitly build a condenser for any n, $k_1 < n$, and $\epsilon = \epsilon(n) > 0$ and have D as small as possible, k_2 as close as possible to $k_1 + \log(D)$, and have k_2 as close as possible to m. Let us call $d = \log(D)$ the *seed length* of C, it measures the amount of randomness the probabilistic construction uses, and clearly the smaller the better. Similarly, let us call $k_1 + d - k_2$ the *entropy loss* of C. The entropy loss measures the difference between the amount of entropy in the system $(k_1 + d)$ and the amount of entropy we preserve (k_2), and we want it small. Finally, let us call $m - k_2$ the *entropy gap* of C. The entropy gap measures how dense the output distribution $C(X, U_D)$ is in its ambient space $\{0,1\}^m$, and the smaller the better. Thus, in this terminology, given n, k_1 and ϵ we would like to find an explicit construction simultaneously minimizing the seed length, entropy loss and entropy gap of the condenser.

An important special case is when the entropy gap $m - k_2$ is 0, and then C is called a (k_1, ϵ) extractor. Non-explicitly, there are extractors (and so the entropy gap zero) with entropy loss $2\log(\frac{1}{\epsilon}) + O(1)$ and seed length $\log(n - k_1) + 2\log(\frac{1}{\epsilon}) + O(1)$, and each one of these bounds is tight (even individually) [RT00].

Dodis et al. [DPW14] observe that if we allow some entropy gap (and in particular even if it is only a constant) then non-explicitly the entropy loss dramatically drops to $O(\log\log(\frac{1}{\epsilon}))$ and the seed length to $\log(n - k) + 1 \cdot \log(\frac{1}{\epsilon}) + O(1)$. With larger entropy gaps, the entropy loss continues to drop until it basically turns into zero, and then we get a *lossless* condenser. For the dependence of the entropy loss on the entropy gap see [DPW14] (and also [AT19]).

The GUV lossless condenser [GUV09] has logarithmic seed length and constant fraction entropy gap. Specifically,

Theorem 1.1. *(The GUV condenser)[GUV09, Theorem 1.7] For every* $n \in \mathbb{N}, k_{max} \leq$

$n, \epsilon > 0$, and $0 < \alpha \leq 1$, there exists an $m \leq 2d + (1+\alpha)k_{max}$ and an explicit function

$$C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$$

with $d = (1 + 1/\alpha) \cdot (\log n + \log k_{max} + \log 1/\epsilon) + O(1)$ such that for all $k \leq k_{max}$, C is an $(n, k) \to_\epsilon (m, k + d)$ (lossless) condenser.

The GUV condenser has found numerous applications (as can be easily seen by looking at the hundreds of papers that cite it). In particular, GUV present an extractor construction by first applying the GUV lossless condenser, and then an extractor construction specifically designed for high min-entropy sources (see [GUV09, Section 4]). Roughly speaking, this extractor construction inherits its entropy loss from the *entropy gap* of the lossless condenser. As a result, the extractor construction presented in [GUV09] has linear entropy loss.

The problem of constructing explicit extractors with short seed length and small entropy loss is widely open and there has been only modest improvement over the extractor of [GUV09] that has linear entropy loss. Specifically, [DKSS13] construct explicit extractors with the slightly sub-linear entropy loss $\frac{k}{polylog(k)}$. Their construction uses improved mergers that are obtained using the polynomial method with multiplicities. In another work, [TU12] modify the GUV condenser construction and using again the multiplicity method of [DKSS13] together with other ideas, give a condenser with small entropy loss and the slightly sub-linear *entropy gap* $\frac{m}{polylog(n)}$. This condenser implies an explicit extractor with a short seed and the same slightly sub-linear entropy loss. Constructing an extractor with a short seed and a better entropy loss is still a major open problem.

In this paper we give another explicit construction of a GUV like lossless condenser. While we do not improve the parameters, our construction uses a different analysis that we believe has the potential to substantially improve current state of the art results. Specifically, we prove:

Theorem 1.2. *(Our condenser) For every $n \in \mathbb{N}, k_{max} \leq n, \epsilon > 0$, and $\frac{16 \log \frac{n}{\epsilon}}{\sqrt{k_{max}}} \leq \alpha \leq 1$, there is an $m \leq d + (1 + \alpha)k_{max}$ and an explicit function*

$$C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$$

with $d = (1 + 1/\alpha) \cdot (\log n + \log k_{max} + \log 1/\epsilon) + O(1)$ *such that for all* $k \leq k_{max}$, C *is an* $(n, k) \to_\epsilon (m, k + d)$ *(lossless) condenser.*

In a similar fashion to [GUV09], our condenser follows from a new construction of an unbalanced bipartite expander graph.

Theorem 1.3. *For every field* $\mathbb{F}_q, n, s \in \mathbb{N}$ *such that* $15 \leq s + 2 \leq n \leq char(\mathbb{F}_q)$, *there exists an explicit graph* $\Gamma : \mathbb{F}_q^n \times \mathbb{F}_q \to \mathbb{F}_q^{s+2}$, *which is a* (K, A) *expander for every* $K > 0$ *with*

$$A = q - \frac{n(s+2)}{2} \cdot (qK)^{\frac{1}{s+2}}. \tag{1}$$

In [GUV09] there is a similar expression with $A = q - (n-1)(s+1)(K^{\frac{1}{s+1}} - 1)$.

While the bound on m in theorem 1.2 is slightly better than the one in theorem 1.1, the former has more restrictions on α then the latter. In any case, those two differences are minor, and as stated before, the main contribution of theorem 1.2 is the method used to prove it, which is very different then the one used in [GUV09], as we next explain.

1.1 Our construction and the GUV construction

Both our construction and the GUV construction have the following structure. The input that we want to condense is interpreted as a degree $n-1$ uni-variate polynomial over $\mathbb{F}_q$, i.e., as an element f from $\mathbb{F}_q^{<n}[X]$. Given the output length $s + 2 \in \mathbb{N}$ (with $s + 2 < n$) both constructions associate f with $s + 1$ different polynomials $f_0, \ldots, f_s$ where $f_i \in \mathbb{F}_q^{<n}[X]$. In GUV the association is done as follows:

1. First, put a field structure on $\mathbb{F}_q^{<n}[X]$ and fix $h \in \mathbb{N}$, that way f^{h^i} (where multiplication and powering is in the field) can also be interpreted as a degree less than n polynomial.

2. Define $f_i = f^{h^i}$.

For example, one may choose a degree n irreducible polynomial $E \in \mathbb{F}_q[X]$ and define the field $\mathbb{F} = \mathbb{F}_q[X] \bmod E$. Then, the condenser construction is as follows:

The condenser C

Parameters: Fix a field $\mathbb{F}_q$, $n, s \in \mathbb{N}$, $n, s \geq 1$. Identify the elements of $\mathbb{F}_q^n$ with univariate polynomials of degree less than n.

Construction: Define $C : \mathbb{F}_q^n \times \mathbb{F}_q \to \mathbb{F}_q^{(s+2)}$ by:

$$C(f, y) = (y, f_0(y), f_1(y), \ldots, f_s(y)) \tag{2}$$

Our construction has the same structure, but our choice of the associated functions $f_0, \ldots, f_s$ is different. Instead of choosing $f_0, \ldots, f_s$ as in GUV, we choose

$$f_i = f^{(i)},$$

i.e., $f^{(i)}$ is the i'th iterated derivative of f in $\mathbb{F}_q[X]$.

To see why our construction is natural, let us look at it from a coding theory perspective. We can associate a function $C : V \times [D] \to \Sigma$ with a linear code of length D and alphabet Σ, where for every $v \in V$ we have the codeword

$$(c(v)_1, \ldots, c(v)_D) \in \Sigma^D$$

where $c(v)_i = C(v, i)$. Using this translation, the GUV construction exactly corresponds to the PV code [PV05] and our construction exactly corresponds to Multiplicity codes [KSY14, GW13].

PV codes and Multiplicity codes are among the few explicit constructions of ECC with close to optimal list-decoding capacity. In the list-decoding problem our goal is to find a construction such that for every given word $(w_1, \ldots, w_D) \in \Sigma^D$ there are few $v \in V$ such that $c(v)$ is close to w. In the condenser construction problem we wish to solve a problem similar to the *list-recoverability* problem, our input is a large subset $W \subseteq \Sigma$, and the output should be the (hopefully few) $v \in V$ such that $c(v)_i \in W$ for every $i \in [D]$ (or the variant where $c(v)_i \in W$ for most $i \in [D]$). Indeed, GUV write that the known connection between codes and extractors (pointed out, e.g., in [TZ04]) and the fact that PV codes have list-decoding close to capacity motivated them to explore whether PV codes give condensers with good list-recoverability.

Looking at it from this perspective, in this paper we ask whether Multiplicity

codes, which are known to have list-decoding close to capacity, also have good list-recoverability and hence give good condensers. In theorems 1.2 and 1.3 we show that this is indeed the case.

Another code which has close to optimal list-decoding capacity is the Folded Reed-Solomon code defined in [GR08]. Consequently, the condenser it produces has been analyzed in [GUV09, Section 6], and achieved worse parameters than the PV based condenser. Interestingly, the parameters are also worse than the ones achieved by our Multiplicity condenser, making this the first time, to the best of our knowledge, that a construction based on Multiplicity codes achieves better results than one based on FRS codes.

While our construction and the GUV construction are similar in structure, they are very different in implementation. In GUV the ring of polynomials $\mathbb{F}_q^{\leq n}[X]$ is "lifted" to a finite field, and the associated functions f_i are chosen so that they lie on a curve, specifically, over the extension field $\mathbb{F}$, all the functions f_i are just polynomials in one common variable. The challenge is proving that if $Q(y, f_0, \ldots, f_m)$ is a non-zero polynomial in the polynomial ring, then Q composed with the curve is a *non-zero*, univariate polynomial over the extension field $\mathbb{F}$. In general, proving that a non-zero polynomial composed with a given curve remains non-zero is a non-trivial challenge, and GUV solve it with a specific trick, that works, but gives constant entropy gap.

In contrast, our construction does not lift to an extension field. Instead the associated functions are just the derivatives of the given input. Thus, we completely avoid the question of proving that a non-zero polynomial composed with a curve remains non-zero, and, instead, we are left with a question similar to interpolation from derivatives. This leads to a widely different analysis as we explain next. We hope that further extensions of it might lead to constructions better than the current state of the art.

1.2 The proof technique

We give a proof sketch of theorem 1.3 (the expanding graph). It is enough to prove that for every $W \subseteq \mathbb{F}_q^{s+2}$ of size at most $AK - 1$ we have $|\mathrm{LIST}(W)| < K$. Fix a set $W \subseteq \mathbb{F}_q^{s+2}$ of size $AK - 1$. Our goal is to bound the number of degree $n - 1$ polynomials f such that $\Gamma(f) \subseteq W$.

Our starting point is to find a non-zero, low-degree, multi-variate polynomial $Q(X, Y_0, \ldots, Y_s)$ such that $Q(w) = 0$ for every $w \in W$. This step is identical to the first step in the proof of GUV. The total degree of Q is $O(|W|^{1/(s+2)}s)$. It is a standard observation that for every f with $\Gamma(f) \subseteq W$ it must be that

$$Q \circ \overline{df} = Q(x, f(x), f'(x), \ldots, f^{(s)}(x))$$

is the zero polynomial, i.e., f solves the differential equation Q. The challenge now is to bound that number of functions f such that $\Gamma(f) \subseteq W$.

To bound the number of degree $n-1$ polynomials such that $\Gamma(f) \subseteq W$ we adapt the list-decoding algorithm of [Kop15] to the list-recovery setting (much the same as GUV adapt the [PV05] list-decoding algorithm to the list-recovery setting). The main lemma Kopparty uses is that given $(y, w_0, \ldots, w_s) \in \mathbb{F}_q \times \mathbb{F}_q^{s+1}$, there is usually at most one degree $n-1$ polynomial f such that:

- The first s derivatives of f at y agree with $w_0, \ldots, w_s$, i.e., $f^{(i)}(y) = w_i$, for $i = 1, \ldots, s$, and,
- $Q \circ \overline{df}$ is the zero polynomial.

Formally, this is true whenever the *Separant* of the equation, $\frac{\partial Q}{\partial Y_s}$, is *non-singular* at $\mathbf{w}$, i.e.,

$$\frac{\partial Q}{\partial Y_s}(y, w_0, \ldots, w_s) \neq 0.$$

Kopparty proves this lemma using Hensel lifting. We rephrase the proof using differential algebra terminology and intuition from [Rit50]. We believe our proof is simpler, and also more amenable to generalizations. Furthermore, this theory was generalized in [Lim15, FZV22], where generalized Separants were introduced, and we believe these generalization might be useful for future improvements of the analysis.

Going back to the list-recovery problem, and following the list-decoding algorithm from [Kop15], let us denote by W_1 the set of all $\mathbf{w} \in W$ such that $\frac{\partial Q}{\partial Y_s}(\mathbf{w}) \neq 0$. We see that for every f such that $\Gamma(f) \subseteq W$ and $\Gamma(f) \cap W_1 \neq \emptyset$, we can recover f by going over all $\mathbf{w} \in W_1$, and for each such $\mathbf{w}$ output the unique suitable degree $n-1$ polynomial, given by the above main lemma.

We are then left with the task of outputting all the degree $n-1$ polynomials such that $\Gamma(f) \subseteq W_0 = W \setminus W_1$. We notice that each of these polynomials solve the lower

degree differential equation $\frac{\partial Q}{\partial Y_s}(x, f(x), \ldots, f^{(s)}) = 0$. Reiterating the process we get a new list of solution. As each time we get a lower degree differential equation, we can iterate the process at most $\deg(Q)$ times. Doing the calculation more carefully (as is done in [Kop15]) saves even this loss, and, furthermore, shows expansion by a factor of about $q - sn\sqrt[s+2]{|W|}$. We explain the thin details in section 4.

2 Preliminaries

We use the following notation:

$$(n)_t = n \cdot (n-1) \cdot \ldots \cdot (n-t+1) = \frac{n!}{(n-t)!},$$

where for $t = 0$, $(n)_0 = 1$. Thus, $(n)_t = t!\binom{n}{t}$.

Also, for $\mathbf{J} = (j_1, \ldots, j_m)$ and $\mathbf{I} = (i_1, \ldots, i_m)$ we define

$$\begin{aligned}(\mathbf{J})_{\mathbf{I}} &= \prod_{\ell=1}^{m} (j_\ell)_{i_\ell}, \\ \binom{\mathbf{J}}{\mathbf{I}} &= \prod_{\ell=1}^{m} \binom{j_\ell}{i_\ell}, \text{ and,} \\ \mathbf{I}! &= \prod_{\ell=1}^{m} i_\ell!.\end{aligned}$$

Thus, $(\mathbf{J})_{\mathbf{I}} = \mathbf{I}!\binom{\mathbf{J}}{\mathbf{I}}$. Finally, $\mathbf{J} - \mathbf{I} = (j_1 - i_1, \ldots, j_m - i_m)$.

2.1 Multi-variate derivatives

Let $R = \mathbb{F}[X_1, \ldots, X_m]$ be the ring of polynomials in m variables over $\mathbb{F}$. For $\mathbf{I} = (i_1, \ldots, i_m)$ with $i_1, \ldots, i_m \in \mathbb{N}$ we define the *partial derivative* in direction $\mathbf{I}$ as the linear operator on R defined by $\frac{\partial X^{\mathbf{J}}}{\partial \mathbf{I}} = (\mathbf{J})_{\mathbf{I}} \cdot \mathbf{X}^{\mathbf{J}-\mathbf{I}}$. We denote

$$Q^{(\mathbf{I})}(\mathbf{X}) = \frac{\partial Q}{\partial \mathbf{I}}(\mathbf{X}).$$

The order of $\mathbf{I}$ is $w(\mathbf{I}) = i_1 + \ldots + i_m$. Notice that for uni-variate polynomials $Q(X)$, $Q^{(i)}(X)$ coincides with the i'th iterated derivative.

Let $\mathbf{w} = (w_1, \ldots, w_m)$ where $w_i \in \mathbb{N}$. The $\mathbf{w}$-weighted degree of a monomial $\mathbf{X^J} = X_1^{j_1} \cdot \ldots \cdot X_m^{j_m}$ is $\sum_{i=1}^m w_i \cdot j_i$. The $\mathbf{w}$-weighted degree of Q, denoted $\deg_{\mathbf{w}}(Q)$, is the largest $\mathbf{w}$-weighted degree of a monomial in Q. We let $|\mathbf{w}|$ denote $\sum w_i$, $\Pi(\mathbf{w}) = \Pi w_i$, and $M_{\mathbf{w},t}$ the number of monomials $\mathbf{X^J}$ with $\mathbf{w}$-weighted degree at most t. Beged-Dov gave upper and lower bounds on $M_{\mathbf{w},t}$:

Lemma 2.1. *[Beg72]*

$$\frac{t^m}{m! \cdot \Pi(\mathbf{w})} \;\leq\; M_{\mathbf{w},t} \;\leq \frac{(t + |\mathbf{w}|)^m}{m! \cdot \Pi(\mathbf{w})}$$

2.2 Condensers

In this subsection let $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$.

Definition 2.2. *We say C is a* (K, A) expander *if for every $S \subseteq \{0,1\}^n$ of cardinality K the set*

$$\Gamma(S) = \bigcup_{s \in S, y \in \{0,1\}^d} C(s, y)$$

has cardinality at least $K \cdot A$.

We next define a condenser:

Definition 2.3. *We say C is an* $(n, k) \to_\epsilon (m, k')$ condenser *if for all distributions X with min-entropy at least k, the distribution $C(X, U_d)$ is ϵ-close to a distribution with min-entropy at least k'. The condenser is* explicit *if C can be computed in time* $\mathrm{poly}(n, \frac{1}{\epsilon})$.

To prove that a function is a condenser or an expander, we use the "list-decoding" approach described in [GUV09]. For $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ and $T \subseteq \{0,1\}^m$ define:

$$\mathrm{LIST}(T) = \{x : \Gamma(x) \subseteq T\}$$
$$\mathrm{LIST}(T, \epsilon) = \left\{x : \Pr_y[C(x, y) \in T] \geq \epsilon\right\}$$

Lemma 2.4. *[GUV09, Lemma 3.2] C is a (K, A) expander iff for every set $T \subseteq \{0,1\}^m$ of cardinality at most $AK - 1$, $\mathrm{LIST}(T)$ has cardinality at most $K - 1$.*

And for condensers:

Lemma 2.5. *[TUZ07, Theorem 8.1],[GUV09, Lemma 5.4] Let $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a function.*

- *If C is a $(K, (1-\epsilon)2^d)$ expander, then C is a $(n, k) \to_\epsilon (m, k + d)$ condenser, i.e., it is a lossless condenser with error ϵ,*
- *If for all $T \subseteq \{0,1\}^m$ of size at most L the set $\mathrm{LIST}(T, \epsilon)$ has cardinality at most H, then C is a $(n, \log(\frac{H}{\epsilon})) \to_{2\epsilon} (m, \log(\frac{L}{\epsilon}) - 1)$ condenser.*

3 The Separant

Let $Q \in \mathbb{F}_q[X, Y_0, \ldots, Y_s]$. When we think of Q as a differential equation, we look for all (low-degree) polynomials $f \in \mathbb{F}_q[X]$ such that

$$Q(X, f(X), f^{(1)}(X), \ldots, f^{(s)}(X)) = 0 \in \mathbb{F}_q[X].$$

Let us define

$$\overline{df} = (X, f(X), f^{(1)}(X), \ldots, f^{(s)}(X), \ldots, f^{(n)}(X), \ldots)$$

Notice that if $f \in \mathbb{F}_q^{<n}[X]$, then $f^{(i)}(X)$ is identically zero for all $i \geq n$. Let us also think of Q as a polynomial $Q \in \mathbb{F}_q[X, Y_0, \ldots, Y_s, \ldots, Y_n \ldots]$ that depends only on X and $Y_0, \ldots, Y_s$. In this notation f solves the differential equation Q iff $Q \circ \overline{df} = 0 \in \mathbb{F}_q[X]$.

A differential equation Q can be itself derived. While formally Q depends on X and $Y_0, \ldots, Y_n, \ldots$, we think of Y_0 as a function depending on X, $Y_0 = f(X)$ and of Y_{i+1} as $\frac{\partial Y_i}{\partial X}$. This motivates the following definition:

Definition 3.1. *Let $Q \in \mathbb{F}_q[X, Y_0, \ldots, Y_s]$, define the infinite sequence of polynomials*

$Q^{(0)}, Q^{(1)}, \ldots$ *where* $Q^{(k)} \in \mathbb{F}[X, Y_0, \ldots, Y_{k+s}]$ *is defined by:*

$$\begin{aligned} Q^{(0)} &= Q \\ Q^{(k+1)} &= \frac{\partial Q^{(k)}}{\partial X} + \sum_{i=0}^{k+s} \frac{\partial Q^{(k)}}{\partial Y_i} \cdot Y_{i+1}. \end{aligned}$$

The motivation behind this definition is apparent given:

Lemma 3.2. *For every* $f \in \mathbb{F}_q[X]$ *and* $\ell \geq 0$

$$(Q \circ \overline{df})^{(\ell)} = Q^{(\ell)} \circ \overline{df}.$$

Proof. By induction. The case $\ell = 0$ is immediate. Assume for ℓ and let us prove for $\ell + 1$. Using the chain rule:

$$\begin{aligned} (Q \circ \overline{df})^{(\ell+1)} =& ((Q \circ \overline{df})^{(\ell)})' = (Q^{(\ell)} \circ \overline{df})' \\ =& \frac{\partial Q^{(\ell)}}{\partial X} \circ \overline{df} + \sum_{i=0}^{s+\ell} \frac{\partial Q^{(\ell)}}{\partial Y_i} \circ \overline{df} \cdot \frac{\partial f^{(i)}}{\partial X} \\ =& \frac{\partial Q^{(\ell)}}{\partial X} \circ \overline{df} + \sum_{i=0}^{s+\ell} \frac{\partial Q^{(\ell)}}{\partial Y_i} \circ \overline{df} \cdot f^{(i+1)} \\ =& (\frac{\partial Q^{(\ell)}}{\partial X} + \sum_{i=0}^{s+\ell} \frac{\partial Q^{(\ell)}}{\partial Y_i} \cdot Y_{i+1}) \circ \overline{df} \\ =& Q^{(\ell+1)} \circ \overline{df}, \end{aligned}$$

where the first equality is because we use iterated derivations, the second is induction, the third is the chain rule (and notice that $Q^{(\ell)}$ depends on $X, Y_0, \ldots, Y_{s+\ell}$). □

We call $Q^{(\ell)}$ the *ℓ-th derivative* of Q. This operation comes from differential algebra [Rit50]. As its name suggests, this operator has some properties similar to regular derivative

Claim 3.3. *[Rit50]*

1. *(linearity) For every* $Q, P \in \mathbb{F}_q[X, Y_0, \ldots], \lambda, \mu \in \mathbb{F}_q, \ell \geq 0$

$$(\lambda Q + \mu P)^{(\ell)} = \lambda Q^{(\ell)} + \mu P^{(\ell)}$$

2. *(Leibniz product rule) For every* $Q, P \in \mathbb{F}_q[X, Y_0, \dots]$

$$(P \cdot Q)^{(1)} = P^{(1)} \cdot Q + P \cdot Q^{(1)}$$

3. *(repeated derivation) For every* $Q \in \mathbb{F}_q[X, Y_0, \dots], \ell_1, \ell_2 \geq 0$

$$(Q^{(\ell_1)})^{(\ell_2)} = Q^{(\ell_1+\ell_2)}$$

Claim 3.4. *Let* $Q \in \mathbb{F}_q[X, Y_0, \dots, Y_s]$ *and* $\ell \in \mathbb{N}$.

- $\deg_{(0,1,1,\dots)} Q^{(\ell)} = \deg_{(0,1,1,\dots)} Q$, *and,*
- $\deg_{(0^{s+2},1,2,3,\dots)} Q^{(\ell)} \leq \ell$. *I.e., if we give* $X, Y_0, \dots, Y_s$ *weight* 0, *and* Y_{s+j} *weight* j, *then the* ℓ*'th derivative degree is at most* ℓ.

Proof. For the first item notice that $\frac{\partial Q^{(\ell)}}{\partial X}$ is either zero or does not change the degree in $Y_0, \dots$. Also, the effect of $\frac{\partial Q}{\partial Y_i} \cdot Y_{i+1}$ is to reduce the degree in Y_i by one and increase the degree in Y_{i+1} by one.

For the second item, we prove by induction. The case $\ell = 0$ is immediate. For the induction step, $\frac{\partial Q^{(\ell)}}{\partial X}$ and $\frac{\partial Q^{(\ell)}}{\partial Y_i} \cdot Y_{i+1}$ for $i < s$, are either zero or do not change the weighted degree, while $\frac{\partial Q^{(\ell)}}{\partial Y_i} \cdot Y_{i+1}$ for $i \geq s$ increase the weighted degree by one. □

One consequence of claim 3.4 is that $Y_{s+\ell}$ appears with degree at most 1 in $Q^{(\ell)}$ and that the coefficient of $Y_{s+\ell}$ in $Q^{(\ell)}$ is a function of $X, Y_0, \dots, Y_s$ alone. Indeed, we next prove the coefficient of $Y_{s+\ell}$ in $Q^{(\ell)}$ is $\frac{\partial Q}{\partial Y_s}$.

Definition 3.5. *(Separant) Let* $Q \in \mathbb{F}[X, Y_0, \dots, Y_s]$. *The* Separant *of* Q, *denoted* S_Q, *is*

$$S_Q = \frac{\partial Q}{\partial Y_s}.$$

A classical lemma from differential algebra (see [Rit50, Page 30]) states that:

Lemma 3.6. *For every* $\ell \geq 1$,

$$Q^{(\ell)} = S_Q \cdot Y_{s+\ell} + R_\ell$$

where $R_\ell \in \mathbb{F}[X, Y_0, \dots, Y_{s+\ell-1}]$ *does not depend on* $Y_{s+\ell}$.

Proof. By induction. For $\ell = 1$, the only way to get Y_{s+1} in $Q^{(1)}$ is in the term $\frac{\partial Q}{\partial Y_s} \cdot Y_{s+1}$. Assume for ℓ and let us prove for $\ell+1$. The only way to get $Y_{s+\ell+1}$ in $Q^{(\ell+1)}$ is by taking $\frac{\partial Q^{(\ell)}}{\partial Y_{s+\ell}}$. By induction, $Y_{s+\ell}$ only appears in $Q^{(\ell)}$ in the linear term $S_Q \cdot Y_{s+\ell}$. Thus, the only term involving $Y_{s+\ell+1}$ in $Q^{(\ell+1)}$ is $\frac{\partial (S_Q \cdot Y_{s+\ell})}{\partial Y_{s+\ell}} \cdot Y_{s+\ell+1} = S_Q \cdot Y_{s+\ell+1}$. $\square$

Lemma 3.7. *Fix $Q \in \mathbb{F}_q[X, Y_0, \ldots, Y_s]$, $(\alpha, \mathbf{b}) = (\alpha, b_0, \ldots, b_s) \in \mathbb{F}_q^{s+2}$ and $S_Q(\alpha, \mathbf{b}) \neq 0$. Suppose $f \in \mathbb{F}_q[X]$ such that:*

- *$f^{(i)}(\alpha) = b_i$, for $i = 0, \ldots, s$, and*
- *$Q \circ \overline{df} = 0$.*

Then there are unique values $b_{s+1}, \ldots, b_n$ such that $f^{(i)}(\alpha) = b_i$.

Proof. We prove by induction on n. The base case $n = s$ is clear. Assume for n and let us prove for $n + 1$. By assumption we know there are unique values $b_{s+1}, \ldots, b_n$ such that $b_i = f^{(i)}(\alpha)$ for $i = s+1, \ldots, n$. Our goal is to show there is a unique value possible for $f^{(n+1)}(\alpha)$.

We will use $Q^{(n-s+1)}$ and the fact that Y_{n+1} appears linearly in it with coefficient S_Q, and that at $(\alpha, \mathbf{b})$, $S_Q(\alpha, \mathbf{b}) \neq 0$. First we notice that

$$\begin{aligned} Q^{(n-s+1)}(\alpha, b_0, \ldots, b_n, f^{(n+1)}(\alpha)) =& Q^{(n-s+1)}(\alpha, f(\alpha), \ldots, f^{(n+1)}(\alpha)) \\ =& Q^{(n-s+1)} \circ \overline{df}(\alpha) \\ =& (Q \circ \overline{df})^{(n-s+1)}(\alpha) = 0, \end{aligned}$$

where the first equality is by induction, the second by definition, the third using lemma 3.2, and the last equality because we know $Q \circ \overline{df}$ is the zero polynomial in $\mathbb{F}_q[X]$.

Next we recall that by lemma 3.6

$$Q^{(n-s+1)}(X, Y_0, \ldots, \ldots, Y_{n+1}) = S_Q(X, Y_0, \ldots, Y_s) \cdot Y_{n+1} + R(X, Y_0, \ldots, Y_n),$$

and therefore

$$\begin{aligned} 0 =& Q^{(n-s+1)}(\alpha, b_0, \ldots, b_n, f^{(n+1)}(\alpha)) \\ =& S_Q(\alpha, \mathbf{b}) \cdot f^{(n+1)}(\alpha) + R(\alpha, b_0, \ldots, b_n). \end{aligned}$$

Thus, $f^{(n+1)}(\alpha) = -\frac{R(\alpha,b_0,\ldots,b_n)}{S_Q(\alpha,\mathbf{b})}$ is uniquely determined. □

In words, this means the following. f solves the differential equation if $Q \circ \overline{df} = 0$. We can think of the conditions $f^{(i)}(\alpha) = b_i$, for $i = 0, \ldots, s$, as $s+1$ initial conditions on the Taylor expansion of f at α. In this terminology, lemma 3.7 says that that if the separant S_Q is non-zero at the point $(\alpha, \mathbf{b})$ then there can be at most one solution to the differential equation Q with degree smaller than the characteristic, satisfying the initial conditions $(\alpha, \mathbf{b})$.

4 Reconstruction with the Polynomial Method

In this section we present a "de-condensing" procedure that given $\Gamma : \mathbb{F}_q^n \times \mathbb{F}_q \to \mathbb{F}_q^{s+2}$ and a set $W \subseteq \mathbb{F}_q^{s+2}$ outputs LIST(W). Throughout this section we assume that $n \leq char(\mathbb{F}_q)$. The de-condensing algorithm works as follows. Given W we first find a low-degree polynomial Q that vanishes over W, namely,

Claim 4.1. *There exists a non-zero polynomial $Q \in \mathbb{F}_q[X, Y_0, \ldots, Y_s]$ with*

$$\deg_{(1,n,\ldots,n-s)} Q \leq D = \left\lceil n \cdot \left[|W| \cdot (s+2)! \right]^{\frac{1}{s+2}} \right\rceil$$

that vanishes on W.

Proof. By lemma 2.1 the number of monomials in $\mathbb{F}_q[X, Y_0, \ldots, Y_s]$ with $(1, n, n-1, \ldots, n-s)$-weighted degree at most D is some value F such that

$$F \geq \frac{D^{s+2}}{(s+2)! \cdot \prod_{j=0}^{s}(n-j)} > |W|.$$

To find a polynomial Q that vanishes on W, we write a homogeneous linear system over $\mathbb{F}_q$ where the variables are the coefficients of the above monomials, and for every $w \in W$ we have a linear equation forcing that the polynomial vanishes on w. As the number of variables is larger than the number of constraints, there is a non-zero solution. □

It then follows that every $f \in \mathbb{F}_q^{<n}[T]$ with $\Gamma(f) \subseteq W$ satisfies the differential equation $Q(x, f(x), \ldots, f^{(s)}(x)) = 0$. Formally,

Claim 4.2. *If $f \in \mathrm{LIST}(W)$, and $q > D$, than*

$$R_f(T) = Q(T, f(T), \ldots, f^{(s)}(T)) \in \mathbb{F}_q[T]$$

is the zero polynomial.

Proof. As $\deg_{(1,n,\ldots,n-s)}(Q) \leq D$ and $\deg(f^{(i)}) < n - i$, R_f has degree at most D. Also, for every $\alpha \in \mathbb{F}_q$,

$$R_f(\alpha) = Q(\alpha, f(\alpha), \ldots, f^{(s)}(\alpha)) = 0.$$

As $q > D$ we must have $R_f = 0$ in $\mathbb{F}_q[T]$. □

The main challenge is proving the number of low-degree solutions to the differential equation Q with starting conditions W is small, and designing an algorithm finding all such solutions. For that we define algorithm $Solve$. The input to the algorithm is a polynomial $\dot{Q} \in \mathbb{F}_q[X, Y_0, \ldots, Y_s]$ and $\dot{W} \subseteq \mathbb{F}_q^{s+2}$. The output contains all polynomials $f \in \mathbb{F}_q^{<n}[X]$ such that $\Gamma(f) \subseteq \dot{W}$ and $\dot{Q} \circ \overline{df} = 0$. The algorithm works as follows:

Algorithm 1: $Solve(\dot{Q}, \dot{W})$

If $\dot{Q}$ does not depend on $Y_0, \ldots, Y_s$ return $\emptyset$.

Let s^* be the largest $j \in \{0, \ldots, s\}$ for which $\dot{Q}$ depends on Y_j.

Set $\mathcal{L}_1 \leftarrow \emptyset$ and

$$\dot{W}_1 \leftarrow \left\{ w \in \dot{W} \mid \frac{\partial \dot{Q}}{\partial Y_{s^*}}(w) \neq 0 \right\}.$$

for $w = (\alpha, w_0, \ldots, w_s) \in \dot{W}_1$ **do**

Assuming there exits some polynomial $g \in \mathbb{F}_q[X]$ such that $\dot{Q} \circ \overline{dg} = 0 \in \mathbb{F}_q[X]$ and $g^{(i)}(\alpha) = w_i$ for all $0 \leq i \leq s$, find the unique values $w_{s+1}, \ldots, w_{n-1}$ such that $g^{(i)}(\alpha) = w_i$ for all $0 \leq i < n$. Such a unique solution exits by lemma 3.7.

Define

$$f(x) = \sum_{i=0}^{n-1} \frac{w_i}{i!}(x - \alpha)^i.$$

If $\Gamma(f) \subseteq \dot{W}$ add f to $\mathcal{L}_1$.

Set

$$\dot{W}_0 \leftarrow \left\{ w \in \dot{W} \mid \frac{\partial \dot{Q}}{\partial Y_{s^*}}(w) = 0 \right\}.$$

$\mathcal{L}_0 \leftarrow Solve(\frac{\partial \dot{Q}}{\partial Y_{s^*}}, W_0)$

return $\mathcal{L}_0 \cup \mathcal{L}_1$

With that the de-condensing algorithm is:

Algorithm 2: Decondensing

Input: Parameters q, s, n, the condenser $\Gamma : \mathbb{F}_q^n \times \mathbb{F}_q \to \mathbb{F}_q^{(s+2)}$, and a set $W \subseteq \mathbb{F}_q^{s+2}$

Output: All $f \in \mathbb{F}_q^{<n}[X]$ such that $\Gamma(f) \subseteq W$

1 Set $D \leftarrow \left\lceil n \cdot \left[|W| \cdot (s+2)!\right]^{\frac{1}{s+2}} \right\rceil$

2 Construct a non-zero polynomial $Q \in \mathbb{F}_q[X, Y_0, \ldots, Y_s]$ with

$$\deg_{(1,n,\ldots,n-s)} Q \leq D$$

that vanishes on W.

3 **return** $Solve(Q, W)$

4.1 Analysis of *Solve*

Lemma 4.3. *(Correctness of Solve) Fix a non-zero polynomial $Q \in \mathbb{F}_q[X, Y_0, \ldots, Y_s]$ such that $\deg_{(1,n,\ldots,n-s)}(Q) < q$, and $W \subseteq \mathbb{F}_q^{s+2}$. Every $f \in \mathbb{F}_q^{<n}[T]$ for which $Q(x, f(x), \ldots, f^{(s)}(x)) = 0$ and $\Gamma(f) \subseteq W$ appears in the output of $Solve(Q, W)$.*

Proof. The proof is by induction on the degree of Q as a polynomial in $Y_0, \ldots, Y_s$, i.e., $\deg_{(0,1,\ldots,1)}(Q)$. In the base case Q depends only on X, thus $Q = Q(X)$. As $Q \neq 0$, there are no solutions to $Q(T, f(T), \ldots, f^{(s)}(T)) = Q(T) = 0$ and $\mathcal{L} = \emptyset$ is the correct output.

Now let $f(T) \in \mathbb{F}_q^{<n}[T]$ such that $\Gamma(f) \subseteq W$ and $Q(T, f(T), \ldots, f^{(s)}(T)) = 0$. We have two cases:

1. $\frac{\partial Q}{\partial Y_{s^*}}(T, f(T), \ldots, f^{(s)}(T)) \neq 0$. Note that

$$\begin{aligned} \deg\left(\frac{\partial Q}{\partial Y_{s^*}}(T, f(T), \ldots, f^{(s)}(T))\right) &\leq \deg_{(1,n,\ldots,n-s)}\left(\frac{\partial Q}{\partial Y_{s^*}}(X, Y_0, \ldots, Y_s)\right) \\ &\leq \deg_{(1,n,\ldots,n-s)}\left(Q(X, Y_0, \ldots, Y_s)\right) < q. \end{aligned}$$

Therefore there must be some $\alpha \in \mathbb{F}_q$ for which

$$\frac{\partial Q}{\partial Y_{s^*}}(\alpha, f(\alpha), \ldots, f^{(s)}(\alpha)) \neq 0.$$

As $(\alpha, f(\alpha), \ldots, f^{(s)}(\alpha)) \in \Gamma(f) \subseteq W$, in the for loop we iterate over this vector and therefore in line 5 we find the unique solution of the ODE with these initial conditions, and because of the uniqueness this solution must be f. As $\Gamma(f) \subseteq W$ we add it to the list $\mathcal{L}$ in line 7.

2. $\frac{\partial Q}{\partial Y_{s^*}}(T, f(T), \ldots, f^{(s)}(T)) = 0$. We notice that in this case $\Gamma(f) \subseteq W_0$, as for every $\alpha \in \mathbb{F}_q$ we have $\frac{\partial Q}{\partial Y_{s^*}}(\alpha, f(\alpha), \ldots, f^{(s)}(\alpha)) = 0$. Also $\deg_{(0,1,\ldots,1)}(\frac{\partial Q}{\partial Y_{s^*}}) < \deg_{(0,1,\ldots,1)}(Q)$, hence by induction $f \in \mathcal{L}_0$.

□

Lemma 4.4. *(List size of Solve) For every non-zero $Q \in \mathbb{F}_q[X, Y_0, \ldots, Y_s]$ with $\deg_{(1,n,n-1,\ldots,n-s)}(Q) \leq D < q$ and every $W \subseteq \mathbb{F}_q^{s+2}$, the size of the output of $Solve(Q, W)$ is at most $\frac{|W|}{q-D}$.*

Proof. We prove by induction on the $(0, 1, \ldots, 1)$-degree of Q. If $\deg_{(0,1,\ldots,1)}(Q)$ is zero, the list is empty, the list size is zero and the claim holds. We next prove the induction step.

For every $w = (\alpha, w_0, \ldots, w_s) \in W_1$, there exists a unique f that may be joined to the list. Furthermore, since $w \in W_1$ we have that:

$$\frac{\partial Q}{\partial Y_{s^*}}(\alpha, f(\alpha), \ldots, f^{(s)}(\alpha)) = \frac{\partial Q}{\partial Y_{s^*}}(\alpha, w_0, \ldots, w_s) \neq 0,$$

thus $\frac{\partial Q}{\partial Y_{s^*}}(T, f(T), \ldots, f^{(s)}(T)) \neq 0$, and its degree is at most D, meaning that it equals 0 for at most D values of T, hence it is non-zero for at least $q - D$ values of $T \in \mathbb{F}_q$. Also, if f appears in the list then $\Gamma(f) \subseteq W$. Hence, each of those $q - D$ values lies in W (and therefore in W_1) and reconstructs f. We conclude that f is reconstructed from at least $q - D$ different points in W_1, thus $|\mathcal{L}_1| \leq \frac{|W_1|}{q-D}$.

We remain with the list size of $\mathcal{L}_0$ which is obtained from $Solve(\frac{\partial Q}{\partial Y_{s^*}}, W_0)$. Since $\deg_{(0,1,\ldots,1)}(\frac{\partial Q}{\partial Y_{s^*}}) < \deg_{(0,1,\ldots,1)}(Q)$, and the $(1, n, \ldots, n-s)$-weighted degree of $\frac{\partial Q}{\partial Y_{s^*}}$ is at most D, we know by induction that $|\mathcal{L}_0| \leq \frac{|W_0|}{q-D}$. Altogether, $|\mathcal{L}| \leq \frac{|W_1|}{q-D} + \frac{|W_0|}{q-D} = \frac{|W|}{q-D}$. □

4.2 Putting it together

Proof. (of theorem 1.3) By lemma 2.4 it is enough to prove that for every $W \subseteq \mathbb{F}_q^{s+2}$ of size at most $AK-1$ we have $|\mathrm{LIST}(W)| < K$. Fix a set $W \subseteq \mathbb{F}_q^{s+2}$ of size $AK-1 < qK$. Let Q be as in claim 4.1, with

$$\begin{aligned} D &= \left\lceil n \cdot \left[qK \cdot (s+2)!\right]^{\frac{1}{s+2}} \right\rceil \leq n \cdot (qK)^{\frac{1}{s+2}} \cdot (((s+2)!)^{\frac{1}{s+2}} + 1) \\ &\leq \frac{n(s+2)}{2} \cdot (qK)^{\frac{1}{s+2}} = q - A \end{aligned}$$

Where the second to last inequality is due to the fact that $(k!)^{1/k} + 1 \leq \frac{k}{2}$ for every $k \geq 15$. Let $\mathcal{L}$ be the output list of $Solve(Q, W)$. Then,

$$\mathrm{LIST}(W) \leq |\mathcal{L}| \leq \frac{|W|}{q-D} \leq \frac{AK-1}{q-D} < K,$$

where the first inequality is by lemma 4.3, the second by lemma 4.4 and the last inequality by using the fact that $A \leq q - D$. □

By choosing the parameters in the same way as done in [GUV09, Theorem 3.5] we get the following expander

Theorem 4.5. *For every positive integers N, $K_{max} \leq N$, all $\epsilon > 0$, and $\frac{16 \log\left(\frac{\log N}{\epsilon}\right)}{\sqrt{\log K_{max}}} \leq \alpha \leq 1$, there is an $M \leq D \cdot K_{max}^{1+\alpha}$ and an explicit $(\leq K_{max}, (1-\epsilon)D)$ expander $\Gamma : [N] \times [D] \to [M]$ with degree $D = O(((\log N (\log K_{max}))/\epsilon)^{1+1/\alpha})$.*

For completeness we repeat the proof from [GUV09].

Proof. Let $n = \log N$ and $k = \log K_{max}$. Let $h_0 = (2nk/\epsilon)^{1/\alpha}$, $h = \lceil h_0 \rceil$, and let q be a prime in the interval $(h^{1+\alpha}/2, h^{1+\alpha}]$.

Set $s + 2 = \lceil k / \log h \rceil$, so that $h^{s+1} \leq K_{max} \leq h^{s+2}$. As $15 \leq s+2 \leq n \leq q = char(\mathbb{F}_q)$, by theorem 1.3, the graph $\Gamma : \mathbb{F}_q^n \times \mathbb{F}_q \to \mathbb{F}_q^{s+2}$ is a $(\leq h^{s+2}, A)$ expander for $A = q - \frac{n(s+2)}{2} \cdot (qK)^{\frac{1}{s+2}}$, because $K_{max} \leq h^{s+2}$, it is also a $(\leq K_{max}, A)$ expander.

Note that the number of left-vertices in Γ is $q^n \geq N$, and the number of right-vertices is

$$M = q^{s+2} \leq q \cdot h^{(1+\alpha)(s+1)} \leq q \cdot K_{max}^{1+\alpha}$$

The degree is

$$\begin{aligned} D = q &\leq h^{1+\alpha} \leq (h_0+1)^{1+\alpha} \\ &= O(h_0^{1+\alpha}) = O((nk/\epsilon)^{1+1/\alpha}) \end{aligned}$$

Lastly, we consider the expansion factor, $A = q - \frac{n(s+2)}{2} \cdot (qK)^{\frac{1}{s+2}} \geq q - \frac{nkhq^{\frac{1}{s+2}}}{2}$, of the graph, first notice

$$nkh \leq \epsilon \frac{h^{1+\alpha}}{2} \leq \epsilon q$$

where the first equality is due to the fact that $nk/\epsilon \leq h^\alpha/2$. Secondly, we can convert our lower bound on α to a lower bound on k

$$k \geq \frac{256}{\alpha^2} \log\left(\frac{n}{\epsilon}\right)$$

and by using it we get

$$\begin{aligned} s+2 \geq \frac{k}{\log h} &\geq \frac{\frac{256}{\alpha^2}\log^2\left(\frac{n}{\epsilon}\right)}{\log h} \geq \frac{\frac{64}{\alpha^2}\log^2\left(\frac{nk}{\epsilon}\right)}{\log h} \geq \frac{\frac{16}{\alpha^2}\log^2\left(2\frac{nk}{\epsilon}\right)}{\log h} \\ &= \frac{16\log^2 h_0}{\log h} \geq \frac{4\log^2 h}{\log h} = 4\log h \geq (1+\alpha)\log h \geq \log q \end{aligned}$$

by combining the two inequalities

$$\frac{nkhq^{1/(s+2)}}{2} = nkh \cdot \frac{q^{1/(s+2)}}{2} \leq \epsilon q.$$

By substituting back to A we get $A \geq (1-\epsilon)q = (1-\epsilon)D$, which concludes the proof. □

Finally, theorem 1.2 is an immediate consequence of lemma 2.5 applied to theorem 4.5.

Bibliography

[AT19] Nir Aviv and Amnon Ta-Shma. On the entropy loss and gap of condensers. *ACM Trans. Comput. Theory*, 11(3):15:1–15:14, 2019.

[Beg72] Aharon Gavriel Beged-Dov. Lower and upper bounds for the number of lattice points in a simplex. *SIAM Journal on Applied Mathematics*, 22(1):106–108, 1972.

[DKSS13] Zeev Dvir, Swastik Kopparty, Shubhangi Saraf, and Madhu Sudan. Extensions to the method of multiplicities, with applications to kakeya sets and mergers. *SIAM J. Comput.*, 42(6):2305–2328, 2013.

[DPW14] Yevgeniy Dodis, Krzysztof Pietrzak, and Daniel Wichs. Key derivation without entropy waste. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 93–110. Springer, 2014.

[FZV22] Sebastian Falkensteiner, Yi Zhang, and Thieu N. Vo. On existence and uniqueness of formal power series solutions of algebraic ordinary differential equations. *Mediterranean Journal of Mathematics*, 19(2):74, Feb 2022.

[GR08] Ariel Gabizon and Ran Raz. Deterministic extractors for affine sources over large fields. *Comb.*, 28(4):415–440, 2008.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. Unbalanced expanders and randomness extractors from parvaresh-vardy codes. *J. ACM*, 56(4):20:1–20:34, 2009.

[GW13] Venkatesan Guruswami and Carol Wang. Linear-algebraic list decoding for variants of reed-solomon codes. *IEEE Trans. Inf. Theory*, 59(6):3257–3268, 2013.

[Kop15] Swastik Kopparty. List-decoding multiplicity codes. *Theory Comput.*, 11:149–182, 2015.

[KSY14] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. *J. ACM*, 61(5):28:1–28:20, 2014.

[Lim15] Maksim Amiryanovich Limonov. Generalized separants of differential polynomials. *Moscow University Mathematics Bulletin*, 70(6):248–252, 2015.

[PV05] Farzad Parvaresh and Alexander Vardy. Correcting errors beyond the guruswami-sudan radius in polynomial time. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 285–294. IEEE Computer Society, 2005.

[Rit50] Joseph Fels Ritt. *Differential algebra*, volume 33. American Mathematical Soc., 1950.

[RT00] Jaikumar Radhakrishnan and Amnon Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM J. Discret. Math.*, 13(1):2–24, 2000.

[TU12] Amnon Ta-Shma and Christopher Umans. Better condensers and new extractors from parvaresh-vardy codes. In *Proceedings of the 27th Conference on Computational Complexity, CCC 2012, Porto, Portugal, June 26-29, 2012*, pages 309–315. IEEE Computer Society, 2012.

[TUZ07] Amnon Ta-Shma, Christopher Umans, and David Zuckerman. Lossless condensers, unbalanced expanders, and extractors. *Comb.*, 27(2):213–240, 2007.

[TZ04] Amnon Ta-Shma and David Zuckerman. Extractor codes. *IEEE Trans. Inf. Theory*, 50(12):3015–3025, 2004.

תקציר

בשנת 2007 Guruswami, Umans וVadhan הראו בניה מפורשת של מכווץ ללא-אובדן המתבססת על קודי Parvaresh-Vardy. מכווץ זה מהווה אבן בניה בבניות רבות של אובייקטים פסידו-אקראיים, ובפרט עומד מאחורי הבניות הטובות ביותר של מחלצים.

בעבודה זו, אנו מראים בניה אלטרנטיבית אשר מתבססת על קודי כפילויות. למרות שהשורה התחתונה של עבודה זו דומה לשל הבניה של GUV, האנליזה שונה מאוד. בבניה של GUV (ובקודי PV), חוג הפולינומים נסגר לשדה סופי, וכל פולינום נקשר למספר איברים בשדה הסופי. בבניה שלנו פולינום מחוג הפולינומים נקשר לנגזרות שלו. האנליזה שלנו מסתכמת לכדי פתירה של משוואה דיפרנציאלית מעל שדה סופי, ומתבססות על טכניקות קודמות, שהוצגו על ידי Kopparty עבור המקרה של פענוח רשימה. אנחנו מראים כי שאלות אלו (כמו גם שאלות כלליות יותר) נחקרו בתחום של אלגברה-דיפרנציאלית, ואנו משתמשים בטרמינולוגיות ותוצאות שפותחו שם.

אנחנו מאמינים כי לטכניקות אלו יש את הפוטנציאל להביא לבניות טובות יותר ולפתור את צווארי-הבקבוק הקיימים כיום בתחום.

הפקולטה למדעים מדויקים ע״ש ריימונד ובברלי סאקלר

בית הספר למדעי המחשב ע״ש בלבטניק

גרפים מרחיבים לא-מאוזנים מקודי כפילויות

חיבור זה הוגש כחלק מהדרישות לקבלת התואר

״מוסמך אוניברסיטה״ – M.Sc. באוניברסיטת תל-אביב

בית הספר למדעי המחשב ע״ש בלבטניק

ע״י

איתי כלב

העבודה הוכנה בהנחיית

פרופסור אמנון תא-שמע

אוגוסט 2022