The Dual-Tensor Polynomial Code Is Not w-Robust.

David Ponarovsky

April 6, 2023

Abstract

w-Robust codes are among the main ingredients in the novel constructions of good Quantum LDPC and LTC codes made by [Din+22], [LZ22], and [PK21]. The Robustness property grants that any small-weight local view of the codeword will spread a fraction of it in both directions of the Left-Right Cayley Complex. On our way to construct Locally Testable Quantum Codes, we have tested a particular case on which the small code set on each local view is the polynomial code and focus on whether it can be w-robust code. Unfortunately, our answer to that question is negative. In this work, we share our experiences, ideas, and insights. We hope that all those would serve others in bringing closer a Quantum PCP Theorem.

1 Preambles

Locally Testable Codes (LTC) are error correction codes such that verifying a uniformly chosen check would be enough to detect any error with probability proportional to its size. Simply put, one can imagine puzzle parts such that any trial to connect pieces in order far from a correct assignment would fail (w.p) at an early step of the process. The analogy for not testability is the case in which the contradiction is observed only in the attempt to putting the last piece.

Likewise, Quantum LDPC codes (qLDPC) are also error correction codes; though that qLDPC encode qubits instead bits, for been considered good, they have to protect against two types of error, and obviously, their decoders have to be designed such that any attempt to detect or fix a one type error would not cause a second type error.

Good LTC and qLDPC have more in common besides the fact that their existences were open questions for a long time that were solved at once. For example, It has shown that sampling uniformly a code would be, with probability 1, neither LTC [BHR03] nor qLDPC code. That stands in total contrast to many other valuable entities in computer science, such as good classic LDPC codes, which a random process can achieve. Thus, it is unsurprising that the recent constructions hinge on a complex that is relatively rich with algebraic structure. And even though those results are indeed used for proving the NLTS conjecture [ABN22], one could expect that the construction of a qLTC will follow soon after them.

Here we shatter light on that wondering by point on one reason that cause the straightforward approach to fail. In detail,.. **[COMMENT]** complete this.

2 Background.

2.1 Polynomial Code.

Consider the field $\mathbb{F}_m$ for an arbitrary prime power $m = q^l$ greater than n. The polynomial codes relay on the fact that any two different polynomials in the ring $\mathbb{F}_m[x]$ at degree at most d different by at least $m-d+1$ points. For example consider a polynomials pair at degree 1, namely two linear straight lines. If they are not identical than they have at most single intersection point, and the disagree on each of the $n-1$ remaining points.

So by define the code to be the subspace contains all the polynomials at degree at most d, in such way that any codeword is an image of such polynomail encoded by n numbers, one can garntee a lower bound on the code's distance. Formally we define:

Definition 1 (Polynomial Code. [RS60])**.** *Fix $m > n$ to be a prime power and let $a_0, a_1, a_2, \dots a_n$ distinct points of the field $\mathbb{F}_m = R$ and define the code $C \subset R$ as follows:*

$$C = \{p(a_0), p(a_1), p(a_2), \cdots p(a_n) : p \text{ is polynomial at degree at most } d\}$$

Observe that C is a linear code at length n over the aleph-bet $\mathbb{F}_m$. The following Lemma states the realtion between the maximal degree of the polynomials and the properites of the code.

Lemma 1. *Fix the degree of the polynomial code to be at most d. Then the parameters of the code are $[n, d+1, n-d]$.*

Proof. The dimension of the code equals to the dimension of the polynomials space at degree at most d which is spanned by the monomial base $e_0, e_1, e_2 ... e_d = 1, x ... x^d$ and therefore is $d+1$. In addition suppose that f, g are different polynomials i.e $f \neq g$.

Hence $h = f - g$ is a non-0 polynomial at degree at most d and therefore has at most d roots. Namely at most d points in which f equals g and at least $n - d$ in which they disagree. Put in another way the distance between any two different codewords of the code is at least $n - d$. □

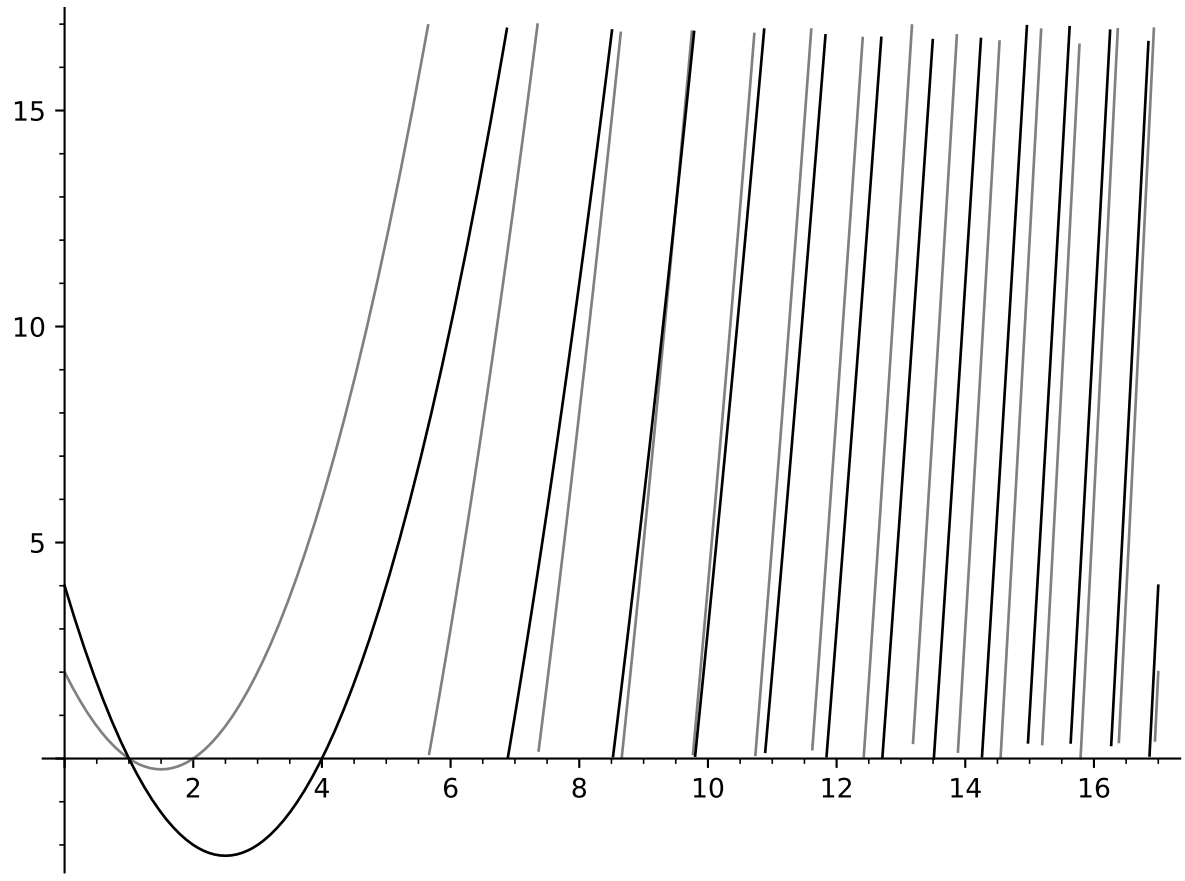

Figure 1: The plot presents the extension of the polynomials **??** and **??** in the filed $\mathbb{F}_{17}$.

Fact 1. *Given $d+1$ points, there is a unique polynomial at degree at most d that pass through all those points. Nevertheless, there is an algorithm G that takes those points as input and outputs the corresponding polynomial.*

Lemma 2 (Testability Of Polynomail Code.)**.** *The polynomial code is $(d \cdot \log(n), \varepsilon)$ testable.*

Proof. Denote by $f \in \mathbb{F}_m^n$ a codeword candidate and consider the following test. First, choose uniformly at random $d+1$ points $x_1, x_2, x_3, ...x_{d+1}$ and use G to output a polynomial that agree on that points, denote it by $G(f)$. Then uniformly choose an additional point, return **True** if $G(f)(x_{d+2}) = f(x_{d+2})$ and **False** otherwise.

Now, observe that if $f \in C$ then by fact 1 we have that $G(f) = f$. In particular $G(f)(x_{d+2}) = f(x_{d+2})$, Namely the test validate a codeword with probability 1.

Now consider the case that $f \neq C$. And observes that $1 - \frac{1}{n}d(f, C)$ is the maximal fraction η such that there exists a polynomial agree with f on at least ηn points. Then we have:

$$\begin{aligned} \eta &\leq \mathbf{Pr}_{x_1,x_2...x_d \sim U, x_{d+1} \sim \mathbb{F}}\left[(Gf)(x) = f(x)\right] \\ &\leq \mathbf{Pr}\left[(Gf)(x) = f(x)\right] \leq \varepsilon \\ 1 - \eta &\geq 1 - \varepsilon \end{aligned}$$

□

Notice that encoding naively the aleph-bet of $\mathbb{F}_p$ in binary strings require to pay a factor $\log n$ bits, So in general the asymptotic rate of the polynomial code attends to zero. Yet in our case, as we use the code for encoding only local views, The total factor that we have to pay is $\log \Delta$, which is constant relative to the number of bits (faces in the complex).

2.2 Quantum Polynomial Code.

Let's define the code C such that any state in C is a coset of the polynomials at degree at most d shifted by $x \in \mathbb{F}_p$. In other words the codeword associated with x is the state $|\underline{c}\rangle = \sum_{\substack{f \in \mathbb{F}_d[x] \\ f(0)=0}} |c + f\rangle$.

The inner product between any d-degree polynomial with zero free coefficient is:

$$\langle f|x^j\rangle = \sum_{i \leq d} \langle a_i x^i | x^j \rangle = \sum_{i \leq d} a_i \mathbf{E}\left[x^i x^j\right] = \sum_{i \leq d} a_i \mathbf{1}_{i+j=_n 0}$$

[COMMENT] Say some words about the classily testability of the polynomial code, and why for quantum it doesn't work. (The dual space of polynomials of low degree is the subspace of all the polynomials with heigh degree.)

Next, we will review Tanner's construction, that in addition to being a critical element to our proof, also serves as an example of how one can construct a code with arbitrary length and positive rate.

Definition 2 (w-Robustness)**.** *Let C_0 be code of length Δ over the aleph-bet Σ with minimum distance $\delta_0 \Delta$. $C = C_0 \otimes \mathbb{F} + \mathbb{F} \otimes C_0^{\perp}$ will be said w-robust if any codeword $c \in C$ at weight less than w it follows that c is supported on at most $2 \cdot w/\delta_0 \Delta$ rows and cols.*

This definition is exactly identical to the one found in [LZ22], expect that here we leave a room for consider also a non-binary codes. We note that, at least for proving the existence of negative vertex adjoins to many normal vertices via heavy edges, the aleph-bet is not matter.

3 The Polynomial-Code Is Not w-Robust.

One idea for constructing is to use the polynomial code instead C_0, The follow form the fact that if one pick degree strictly greater than $\Delta/2$ then $C_0^{\perp} \subset C_0$ and therefore one could choose C_z to be the same code defined on the negative vertices of the graph.

Here we prove that the dual-tensor code, in that case, is not w-robust, meaning that any such construction should be consider other way for proving the reduction Lemma.

Claim 1. *Let C_0 be the $[\Delta, d, \Delta - d]$ polynomial code. Then any code word in $\left(C_0^{\perp} \otimes C_0^{\perp}\right)^{\perp}$ is a polynomial in $F[x, y]$ at degree at most $\Delta + d$*

Proof. Consider base element $C_0 \otimes \mathbb{F}$, denote it by $c = g_i \otimes e_j$. And notice that c has representation in $F[x, y]$ of $\prod_{y' \neq j} (y - y') g_i(x)$. By the fact that $g_i(x) \in C_0$ we have that degree of c is at most $\Delta + \delta$. Hence any element in the subspace of $C_0 \otimes \mathbb{F}$ is a polynomial at degree at most $\Delta + d$. □

Claim 2. *The dual-tensor polynomial code is not w-robust.*

Proof. Consider the following polynomial

$$P(x, y) = \prod_{i \neq \Delta - 1} (x + iy) = \prod_{i \neq 1} (x - iy)$$

The degree of any monomial is at most $\Delta - 1$, Thus it clear that $P \in \left(C_0^{\perp} \otimes C_0^{\perp}\right)^{\perp}$. And by the fact that for any $x \neq y$ there exists $i \neq 1$ such that $x = iy$ we have that $P(x, y) = 0$. Hence the weight of P is at most $|\{(x, y) : x = y\}| = \Delta$. Yet, for any $x = y$ it follows:

$$P(x, x) = \prod_{i \neq \Delta - 1} (x + ix) = x^{\Delta - 1} \prod_{i \neq \Delta - 1} (1 + i) = (\Delta - 1)! =_{\Delta} -1 \neq_{\Delta} 0$$

Put it differently, the diagonal of the matrix has only non-zero values, therefore the P is supported on the entire collection of rows and columns. Namely, we found a codeword in the dual tensor code at weight less than $\Delta^{1\frac{1}{2}}$ which is not restricted to at most $\Delta^{1\frac{1}{2}}/\delta_0 \Delta$. So, the dual-tensor polynomial code is not a w-robust code, for $w = \Delta^{1\frac{1}{2}}$. □

References

[RS60] Irving S. Reed and Gustave Solomon. "Polynomial Codes Over Certain Finite Fields". In: *Journal of The Society for Industrial and Applied Mathematics* 8 (1960), pp. 300–304.

[BHR03] Eli Ben-Sasson, Prahladh Harsha, and Sofya Raskhodnikova. "Some 3CNF Properties Are Hard to Test". In: *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*. STOC '03. San Diego, CA, USA: Association for Computing Machinery, 2003, pp. 345–354. ISBN: 1581136749. DOI: 10.1145/780542.780594. URL: https://doi.org/10.1145/780542.780594.

[PK21] Pavel Panteleev and Gleb Kalachev. *Asymptotically Good Quantum and Locally Testable Classical LDPC Codes*. 2021. DOI: 10.48550/ARXIV.2111.03654. URL: https://arxiv.org/abs/2111.03654.

[ABN22] Anurag Anshu, Nikolas P. Breuckmann, and Chinmay Nirkhe. *NLTS Hamiltonians from good quantum codes*. 2022. arXiv: 2206.13228 [quant-ph].

[Din+22] Irit Dinur et al. *Good Locally Testable Codes*. 2022. DOI: 10.48550/ARXIV.2207.11929. URL: https://arxiv.org/abs/2207.11929.

[LZ22] Anthony Leverrier and Gilles Zémor. *Quantum Tanner codes*. 2022. arXiv: 2202.13641 [quant-ph].