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

David Ponarovsky

April 2, 2023

1 Background.

1.1 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 $n - d + 1$ points. By define the code to be the subspace contains all the polynomials at degree at most d encoded by n numbers associated with their values. Formally we define:

Definition 1. *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:*

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

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 vectors $e_1, e_2..e_d = 1, x..x^d$ 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 $n - d$. □

Notice that encoding naively the aleph-bet of $\mathbb{F}_p$ in binary strings require to pay a factor $\log n$ bits, So the asymptotic rate of the code attends to zero. **[COMMENT]** Add a statement about the vanishing rate of the binary encoded version. And add a paragraph about Tanner code in which each edge correspond to a non binary alpha-bet.

1.1.1 Note On 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:

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

[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.

2 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.

$$
\begin{aligned}
P(x, y) &= \prod_{i \neq \Delta - 1} (x + iy) = \prod_{i \neq 1} (x - iy) \\
P(x, x) &= \prod_{i \neq \Delta - 1} (x + ix) = x^{\Delta - 1} \prod_{i \neq \Delta - 1} (1 + i) = (\Delta - 1)! =_{\Delta} -1 \neq_{\Delta} 0
\end{aligned}
$$

□