Discrete Applied Mathematics 365 (2025) 138–159

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Interplay between resiliency and polynomial degree — Recursive amplification, higher order sensitivity and beyond ☆

Subhamoy Maitra [a], Chandra Sekhar Mukherjee [b], Pantelimon Stănică [c], Deng Tang [d,*]

[a] Applied Statistics Unit, Indian Statistical Institute, 203 B T Road, Kolkata 700108, India
[b] University of Southern California, USA
[c] Naval Postgraduate School, Applied Mathematics Department, Monterey, CA 93943, USA
[d] Shanghai Jiao Tong University, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 2 September 2024
Received in revised form 10 December 2024
Accepted 1 January 2025
Available online xxxx

MSC:
06E30
94A60
94D10

Keywords:
Higher order sensitivity
Maiorana-McFarland construction
Polynomial degree
Resiliency
Sensitivity
Separation

ABSTRACT

Boolean functions are important primitives in different domains of cryptology, complexity and coding theory, and far beyond in different areas of science and technology. In this paper we connect the tools of cryptology and complexity theory in the domain of resilient Boolean functions. It is well known that the resiliency of a Boolean function and its polynomial degree are directly connected. We first show that borrowing an idea from complexity theory, one can implement resilient Boolean functions on a large number of variables with little amount of circuit. Further, we also look into the search techniques used in the construction of resilient Boolean functions to show the existence and non-existence results of functions with low polynomial degree and high sensitivity on small number of variables. In the process, we settle some previously open problems. Finally, we extend the notion of sensitivity to higher order and present a construction with low polynomial degree and higher order sensitivity exploiting Maiorana-McFarland functions. The questions we raise identify novel combinatorial problems in the domain of Boolean functions.

1. Introduction

Two of the central properties of Boolean functions studied extensively over the past three decades in complexity theory are the real polynomial degree $\mathrm{pdeg}(f)$ and sensitivity $s(f)$ (see Sections Section 2.1, respectively 2.2, for their definitions). These notions have important implications in the domain of query complexity [2,3] (and not only), where finding functions with lower polynomial degree than sensitivity generates more candidates for obtaining super-linear separation between two of the query complexity models, the classical deterministic and exact quantum models (we say that there is a super-linear separation between the functions f and g, if $|f(x) - g(x)| = \omega(x)$ as $x \to \infty$, where the Landau symbol $\omega(\cdot)$ has its usual meaning, that is, $\lim_{x\to\infty} \frac{|f(x)-g(x)|}{x} = \infty$).

☆ This is a substantially extended and revised version of the paper "On Boolean Functions with Low Polynomial Degree and Higher Order Sensitivity (Extended Abstract)" presented in WCC 2022. We rewrote and added computational results in the Sections 1–3, the part presented in WCC 2022. Sections 4 and 5 contain mostly additional results over the workshop version.

* Corresponding author at: Shanghai Jiao Tong University, Shanghai 200240, China.
E-mail addresses: subho@isical.ac.in (S. Maitra), chandrasekhar.mukherjee07@gmail.com (C.S. Mukherjee), pstanica@nps.edu (P. Stănică), dtang@foxmail.com (D. Tang).

https://doi.org/10.1016/j.dam.2025.01.003
0166-218X/

An important open problem, which was studied extensively, was to determine the maximum possible separation between sensitivity and polynomial degree. We see that the problem reduces to finding the separation between the number of variables and the real polynomial degree in a fully sensitive function (a function f on n variables with $s(f) = n$). Simply put, a necessary and sufficient condition for have full sensitivity in a function f on n variables is to have an input point $\mathbf{x} \in \{0,1\}^n$ such that $f(\mathbf{x}) = \overline{f(\mathbf{x}^i)}$ (the over-line is the complement of the input), $1 \le i \le n$, where $\mathbf{x}^i$ is obtained by altering the value of the ith bit of $\mathbf{x}$. Thus, we fix the output corresponding to some $n+1$ input points, of the total 2^n input points, which, interestingly, greatly restricts the real polynomial degree of the function. Without any restriction, a function that depends on all of its variables can have $\mathsf{pdeg}(f)$ as low as $\mathcal{O}(\log n)$ (the Landau symbol $\mathcal{O}(\cdot)$ has its usual meaning; specifically, $f = \mathcal{O}(g)$ means $|f(x)| < cg(x)$ holds with some constant $c > 0$, for x sufficiently large). However, one of the seminal papers [14] in the study of Boolean functions dictate that $\mathsf{pdeg}(f) = \Omega\left(s(f)^{\frac{1}{2}}\right)$ (the Knuth symbol Ω is the asymptotic lower bound, that is, $f = \Omega(g)$ if and only if $g = \mathcal{O}(f)$). Furthermore, apart from the famous recursive amplification method (which is also known as the function composition) there does not exist any known method to obtain functions with non-constant separation between $s(f)$ and $\mathsf{pdeg}(f)$. In fact, the maximum known separation between $s(f)$ and $\mathsf{pdeg}(f)$ is achieved by finding and recursively amplifying a 6 variable function with $s(f) = 6$ and $\mathsf{pdeg}(f) = 3$ (due to Kushilevitz [13]). This results in a function f on $n = 6^d$ variables with full sensitivity n and polynomial degree of 3^d so that $s(f) = \mathsf{pdeg}(f)^{\log_3 6} \approx \mathsf{pdeg}(f)^{1.63}$ [14].

Interestingly, the real polynomial degree is intrinsically connected to the cryptographically important property of resiliency $r(f)$ (see Section 2.1 for the definition). That is, if an n-variable function f has $\mathsf{pdeg}(f) = m$, then the function $g = f \oplus \mathcal{L}_n$ is $(n-m-1)$-resilient, where $\mathcal{L}_n$ is the all variable (symmetric) linear function. In this paper we refer to g and f as each other's dual, and we define the dual sensitivity $ds(f)$ property, where a function g has full dual sensitivity if and only if there exists a point $\mathbf{x} \in \{0,1\}^n$ such that $g(\mathbf{x}) = g(\mathbf{x}^i)$, $1 \le i \le n$. This results in a one-to-one connection between the $\mathsf{pdeg}(f)$-$s(f)$ relationship and $r(g)$-$ds(g)$ relationship, where g is the dual of f. Resilient Boolean functions have received a lot of interest in construction of symmetric ciphers as evident from [4,8,11,17] and references therein.

In this paper, we show that the techniques from complexity theory and cryptology can supplement each other with respect to combinatorial aspects of Boolean functions. Further we obtain an efficient construction of resilient Boolean functions from the recursive amplification method referred above, and also settle several questions on the separation for Boolean functions on small number of variables through the analysis of resilient functions. Further, we exploit a well-known Maiorana-McFarland construction [5] of highly nonlinear Boolean functions in obtaining certain results related to higher order sensitivity that we introduce here.

To elaborate, we extend the definition of sensitivity to a stronger notion of kth order sensitivity (dual sensitivity), where the known measure of sensitivity is interpreted as the first order sensitivity (dual sensitivity). Obtaining a kth order sensitive function requires fixing the outputs corresponding to $\sum_{i=1}^{k}\binom{n}{i}$ input points, compared to $n+1$ points in the case of full sensitivity. We discuss the recursive amplification method in this regard and then observe that this method can be used to obtain super-linear separation between the number of variables and the polynomial degree in functions with constant order of sensitivity. However, the recursive amplification method cannot be used in its current form to obtain separation between n and $\mathsf{pdeg}(f)$ for functions with super-constant orders of sensitivity and we need certain tweaks. Next we study kth order sensitive functions on n variables that have lower polynomial degree using the Maiorana-McFarland (MM) class of functions, which has been a source of cryptographically important functions for more than two decades (see [17] for details about Boolean functions, the MM construction, and its cryptographic properties). In this direction, we develop a construction to obtain super-constant separation between the number of variables n and real polynomial degree of functions with super-constant orders $o(\log n)$ of sensitivity (the Landau symbol $o(\cdot)$ is used in $f = o(g)$ to mean $\lim_{x\to\infty}\frac{f(x)}{g(x)} = 0$). That is, we use different properties of resiliency and polynomial degree to obtain results and constructions that apply to both paradigms of cryptology and complexity theory.

1.1. Contribution and organization

In this paper we study the relationship between sensitivity and real polynomial degree of Boolean functions using the connection between real polynomial degree and resiliency, a widely studied property in the domain of complexity theory and cryptology, respectively. We extend the study of the relationship between sensitivity and polynomial degree by looking into the resiliency-dual sensitivity relationship as well as the higher order sensitivity, which we define by naturally extending the notion of sensitivity.

In Section 2, which explains certain background materials, we describe the properties of sensitivity, resiliency and real polynomial degree of Boolean functions. We then define dual sensitivity, so that there is a one-to-one connection between "resiliency-dual sensitivity" and "polynomial degree-sensitivity". Next, we extend the notion of sensitivity to higher kth order sensitivity and correspondingly, we define higher order dual sensitivity and finally describe the need for understanding the hierarchy that is thus created. We also refer to some of the cryptographically and complexity theoretically important properties of Boolean functions, like noise sensitivity and variable influence.

In Section 3 we describe the recursive amplification (function composition) method, which has been used to obtain super-linear separation between sensitivity $s(f)$ and polynomial degree $\mathsf{pdeg}(f)$ in the domain of complexity theory, among other applications. We first tweak this method so that it can admit a larger class of base functions to give the desired $s(f) - \mathsf{pdeg}(f)$ advantage, which is not directly possible in its original form. We then show how this technique

can be used to obtain functions with very high resiliency so that the corresponding circuit size is $\mathcal{O}(n)$ and the depth is $\mathcal{O}(\log n)$, for a function on n input variables. To the best of our knowledge, this is the most efficient implementation for certain classes of resilient functions. We further discuss the nonlinearity profile of such functions and obtain some nontrivial recursive lower bounds. We complete this section by showing that the recursive amplification method can be used to obtain super-linear separation between n and $\mathsf{pdeg}(f)$ for functions with constant higher order sensitivity. We also present some technical results on the nonlinearity of the functions obtained via the recursive amplification to conclude this part.

Next in Section 4 we observe the $s(f)$-$\mathsf{pdeg}(f)$ relation for functions on small number of variables (up to 10). The study of Boolean functions on small number of variables towards obtaining interesting properties has been the subject of study for the last two decades (see [8] and the references therein). In this regard we use the properties of polynomial degree and resiliency to give answers to some open questions. Specifically we obtain the full characterization of all 6 variable fully sensitive function with $\mathsf{pdeg}(f) = 3$ and also show that there does not exist any 7 variable fully sensitive functions with $\mathsf{pdeg}(f) = 3$, which were previously not known. The result for $n = 7$ is particularly important as this implies that the smallest n, for which we can search for functions that can be recursively amplified to overcome the currently known separation between sensitivity and polynomial degree, is $n = 10$. For $n = 8, 9$ and 10 we check the rotation symmetric classes and obtain previously unknown 9 variable fully sensitive functions with $\mathsf{pdeg}(f) = 4$.

Finally in Section 5 we discuss functions with high orders of sensitivity. We first observe that the recursive amplification method does not give us non-constant separation between n and $\mathsf{pdeg}(f)$ in such cases. In this direction, we use the Maiorana-McFarland constructions, which has widely been utilized to construct functions with good cryptographic properties. We use the real polynomial structure of this class to obtain fully sensitive functions on n variables with $\mathsf{pdeg}(f) = n - \Theta(\log n)$ (the Landau symbol $\Theta(\cdot)$ has its usual meaning – see the preliminaries section, below). Note that this is not surprising as we have much larger separation using the recursive amplification method. However, we show that this method can be extended for super-constant orders of sensitivity and design a construction to obtain non-constant separation between n and $\mathsf{pdeg}(f)$ in $o(\log n)$ order sensitive functions. Specifically, we design kth order sensitive functions on n variables with $\mathsf{pdeg}(f) = \mathcal{O}\left(n - \frac{\log(\frac{n}{2}-k)-\log(k)}{k}\right)$, which is the final contribution of this paper.

We conclude the paper in Section 6 by summarizing our results and laying out future directions to better understand the hierarchy of how higher order sensitivity restricts the polynomial degree of Boolean functions and discuss possible cryptographic implication of the mentioned characterization.

2. Preliminaries

Let us first define some notations that we use frequently in the paper. We denote by $\mathbf{x}$ an element of $\mathbb{F}_2^n$ (the n-dimensional vector space over the two-element field $\mathbb{F}_2$) and its individual bits are expressed as x_i, $1 \le i \le n$, where $x_i \in \mathbb{F}_2$. We use $\oplus$ to denote the addition in both $\mathbb{F}_2$ and $\mathbb{F}_2^n$. We denote by $\mathbf{1}_n$ and $\mathbf{0}_n$ the all one, respectively, all zero element in $\mathbb{F}_2^n$. We denote by $\mathbf{1}_{n-i}||\mathbf{0}_i$ the element in $\mathbb{F}_2^n$ whose first $n-i$ bits are one and the last i bits are zero. Further, we denote by $[n]$ the set $\{1, 2, \ldots, n\}$ and by $|S|$, the cardinality of a set S. In what follows, we use the Landau symbols $\mathcal{O}$ and o, as well as Knuth symbols Ω, Θ, with their usual meanings. As defined earlier, recall that $A(x) = \mathcal{O}(B(x))$, $A(x) = \Omega(B(x))$ mean that the inequality $|A(x)| < cB(x)$, respectively, $|A(x)| > cB(x)$, hold with some positive constant c, when their variable x is sufficiently large. Further, $A(x) = \Theta(B(x))$ means that both $A(x) = \mathcal{O}(B(x))$ and $A(x) = \Omega(B(x))$ hold when x is sufficiently large, and $A(x) = o(B(x))$ means that $A(x)/B(x)$ tends to zero, when x approaches ∞.

Now let us look at the properties of real polynomial degree and resiliency of Boolean functions and the relationship between these.

2.1. Resiliency and polynomial degree

Corresponding to any Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ (which can be represented via its truth table, that is, the output vector, or the algebraic normal form, that is, the multivariate polynomial representation) there exists a unique polynomial $p : \mathbb{F}_2^n \to \mathbb{R}$ so that f and p output the same value for all possible inputs. This polynomial is unique for any representation of $\mathbb{F}_2$ (the additive, $\{0, 1\}$, or multiplicative, $\{-1, 1\}$). In all such cases, the corresponding real polynomial is also multilinear and the degree of the real polynomial is the same for any such representation. In this paper, we take the real polynomial of a Boolean function f as $p : \mathbb{F}_2^n \to \mathbb{R}$ such that $p(\mathbf{x}) = f(\mathbf{x})$, $\forall \mathbf{x} \in \mathbb{F}_2^n$. As expected, the real polynomials degree is the degree of this polynomial.

The real polynomial degree of a Boolean function is intrinsically connected to the cryptographic property of the resiliency. The connection stems from the structure of the Walsh spectrum (the output multiset of the Walsh transform – defined below) of a Boolean function. For $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, $\omega = (\omega_1, \omega_2, \ldots, \omega_n)$ in $\mathbb{F}_2^n$ and $\mathbf{x} \cdot \omega$, the usual inner product in $\mathbb{F}_2^n$, we define the Walsh transform (the discrete version of the Fourier transform) of a function $g : \mathbb{F}_2^n \to \mathbb{F}_2$ at the point $\omega \in \mathbb{F}_2^n$ by

$$W_g(\omega) = \sum_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{g(\mathbf{x}) + \omega \cdot \mathbf{x}}.$$

A Boolean function g is called m-resilient if $W_g(\omega) = 0$, for $0 \le wt(\omega) \le m$. Here, $wt(\cdot)$ denotes the Hamming weight, that is, the number of nonzero components in the input. The nonlinearity of an n-variable Boolean function g, denoted by $NL(g)$, is the minimum Hamming distance from g to the set of all the affine functions, where any affine function $l(\mathbf{x})$ has the form $\omega \cdot \mathbf{x} \oplus c$ with $\omega \in \mathbb{F}_2^n$ and $c \in \mathbb{F}_2$. It can be easily checked that the nonlinearity of g can be computed by $NL(g) = 2^{n-1} - \frac{1}{2}\max_{\omega\in\mathbb{F}_2^n} |W_g(\omega)|$. We now look into the unique real polynomial corresponding to any Boolean function. If we transform the inputs x_i to y_i where $y_i = (-1)^{x_i}$ (the same as $y_i = 1 - 2x_i$), then we have another unique polynomial $\hat{p}_f : \{-1, 1\}^n \to \mathbb{R}$ such that $p(\mathbf{x}) = \hat{p}_f(\mathbf{y})$ where $\mathbf{y} = (y_1, \ldots, y_n)$. It is easy to see that $\deg(p) = \deg(\hat{p}_f)$, and this value is called the polynomial degree of f, denoted by $\mathsf{pdeg}(f)$. Now $\hat{p}_f$ can be expressed using the Walsh spectrum of the function f in the following manner. With $\chi^{\omega} = \prod_{\omega_i=1} y_i$, we have

$$\hat{p}_f(\mathbf{y}) = \sum_{\omega\in\{0,1\}^n} \frac{W_g(\omega)}{2^n}\chi^{\omega}.$$

Thus, a function f has a polynomial degree k if and only if $W_f(\omega) = 0,\ \forall\omega$ with $wt(\omega) > k$. Furthermore, a function f is k-resilient if and only if $W_f(\omega) = 0,\ \forall\omega$ with $wt(\omega) \le k$. Finally, we note that given a function f, $W_f(\omega) = W_{f\oplus\mathcal{L}_n}(\bar{\omega})$, where $\bar{\omega} = \omega \oplus \mathbf{1}_n$ and $\mathcal{L}_n = \oplus_{i=1}^n x_i$. These structural arguments gave rise to the famous result connecting the resiliency and polynomial degree of Boolean functions.

Theorem 2.1 ([15], page 150)**.** *If a function g is k-resilient then the function $f = g \oplus \mathcal{L}_n$ will have a polynomial degree of $n - k - 1$, where $\mathcal{L}_n = \oplus_{i=1}^n x_i$.*

Thus, it is clear that a Boolean function g is m-resilient if and only if $\mathsf{pdeg}(f) = n - m - 1$.

Next, we study the notion of sensitivity and also, that of dual sensitivity, which we define to better understand the connection between resiliency and polynomial degree.

2.2. Sensitivity

For any $\mathbf{x} \in \mathbb{F}_2^n$, we let $\mathbf{x}^i$ to be $\mathbf{x}$ with the ith bit of $\mathbf{x}$ flipped (complemented). The sensitivity of a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ at a point $\mathbf{x}$ can be defined as $s(f, \mathbf{x}) = |\{i \in [n] : f(\mathbf{x}) \ne f(\mathbf{x}^i)\}|$, and the sensitivity of a function is

$$s(f) = \max_{\mathbf{x}\in\mathbb{F}_2^n} s(f, \mathbf{x}).$$

Sensitivity, and different variants of it, such as Gaussian noise sensitivity [9] and average sensitivity [6] are well studied metrics of Boolean functions with implications in complexity theory, learning theory [10], and beyond.

It is natural to consider the situation where we want the function to have the same value even if multiple input bits of $\mathbf{x}$ are flipped regardless of their position. In this direction, we define the kth order sensitivity of a Boolean function. For any set $S \subseteq [n]$ and the input point $\mathbf{x} \in \mathbb{F}_2^n$ we define $\mathbf{x}^{(S)}$ as the input point obtained by flipping the jth bit of $\mathbf{x}$ for all $j \in S$. Sensitivity is defined around the notion of flipping any single component corresponding to a given input where the output of the function remains unchanged. In this regard we define kth order sensitivity of a function in the following manner.

Definition 2.2. We call a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ kth order sensitive if there exists $\mathbf{x} \in \mathbb{F}_2^n$ such that

$$f(\mathbf{x}) \ne f\left(\mathbf{x}^{(S)}\right),\ \forall S \subseteq [n],\ 1 \le |S| \le k.$$

That is, f is kth order sensitive if there exists an input so that flipping any $i \le k$ of the component bits of the input, changes the function's output. Thus a first order sensitive function is simply a function with $s(f) = n$. The main implication of kth order sensitivity is that it indeed further restricts how low the degree of the real polynomial corresponding to the function can be. Without any restrictions we know $\mathsf{pdeg}(f)$ can be as low as $\log n$ for functions that depend on n variables. If we fix $s(f) = n$ then the polynomial degree is $\Omega\left(\sqrt{n}\right)$.

In relation to kth order sensitivity we show that there are MM type constructions such that there exists non-constant separation between n and real polynomial degree of functions with $t(n)$-order sensitivity, where $t(n) = o(\log n)$. Studying this hierarchy and motivating understanding the relationship between sensitivity order and the minimum possible polynomial degree is the central theme of the paper.

2.3. Dual sensitivity

Given a function f on n variables with polynomial degree m, we call the function $g = f \oplus \mathcal{L}_n$, the dual of f, which has $n - m - 1$ resiliency. The dual sensitivity of a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ at a point $\mathbf{x}$ is defined as $ds(f, \mathbf{x}) = |\{i \in [n] : f(\mathbf{x}) = f(\mathbf{x}^i)\}|$. The dual sensitivity of f is

$$ds(f) = \max_{\mathbf{x}\in\mathbb{F}_2^n} ds(f, \mathbf{x}).$$

This notion can be extended to kth order dual sensitivity in the following manner.

Definition 2.3. We say a function g is kth order dual sensitive if there exists $\mathbf{x} \in \mathbb{F}_2^n$, such that, for all $j : 1 \le j \le k$ we have:

- If $j \equiv 0 \bmod 2$, then $f(\mathbf{x}) \neq f\left(\mathbf{x}^{(S)}\right),\ \forall S \subseteq [n]$ with $|S| = j$.
- If $j \equiv 1 \bmod 2$, then $f(\mathbf{x}) = f\left(\mathbf{x}^{(S)}\right),\ \forall S \subseteq [n]$ with $|S| = j$.

That is, a function is kth order dual sensitive if there is an input point such that if we flip the values of any odd number $\le k$ of input bits then the function's output remains unchanged and if we flip any even number $\le k$ of input bits then the function's output gets complemented. The next proposition is immediate.

Proposition 2.4. *A function f on n variables is kth order sensitive if and only if its dual $g = f \oplus \mathcal{L}_n$ is kth order dual sensitive.*

Note 2.5. *In this paper we only talk about fully sensitive functions (for first order sensitive functions) and higher order sensitive functions and their resilient kth order dual sensitive counterparts. We use the following notations to describe these properties:*

- *An (n, k, d)-function is a Boolean function of n variables that is kth order sensitive and has real polynomial degree d.*
- *An $[n, k, m]$-function is a Boolean function of n variables that is kth order dual sensitive and m-resilient.*

Thus, if f is an (n, k, d)-function then $g = f \oplus \mathcal{L}_n$ is an $[n, k, n - d - 1]$-function.

Beyond understanding the hierarchy of how restricting outputs for inputs around a point restrict the polynomial degree, there are many concepts that touch in one way or another our notion of kth order sensitivity, like (variable) influence, or noise sensitivity, to name only two such. The influence of the ith variable x_i of f of n variables is defined as $\mathrm{Inf}_i(f) = \mathrm{Prob}[f(\mathbf{x}) \neq f(\mathbf{x}^i)]$ where $\mathbf{x} \in \mathbb{F}_2^n$ is chosen equiprobably, and the *total influence* is then $\mathrm{Inf}(f) = \sum_{i=1}^n \mathrm{Inf}_i(f)$. The influence of the ith variable and the total influence on f is related to the Walsh coefficients of f (cf. [16]) as $\mathrm{Inf}_i(f) = 2^{-n} \sum_{\mathbf{u} \in \mathbb{F}_2^n} u_i W_f^2(\mathbf{u})$, and $\mathrm{Inf}(f) = \sum_{i=1}^n \mathrm{Inf}_i(f) = 2^{-n} \sum_{\mathbf{u} \in \mathbb{F}_2^n} wt(\mathbf{u}) W_f^2(\mathbf{u})$. Surely, it is well known that the total influence is equivalent to edge-boundary size for subsets of the Hamming cube, and ultimately, it is a measure of the computational complexity of a function, which is very important in cryptography. Also, the Fourier Entropy-Influence (FEI) conjecture (Friedgut–Kalai [7]) claims that there exists a universal constant C such that for any Boolean function f we have $\mathbb{H}(f) \le C \cdot \mathrm{Inf}(f)$, where $\mathbb{H}(f) = \sum_{\mathbf{u} \in \mathbb{F}_2^n} \widehat{f}(\mathbf{u})^2 \log_2 \frac{1}{\widehat{f}(\mathbf{u})^2}$, is the entropy of f. The FEI conjecture implies a version of Mansour's conjecture (see [12,16]), which states that given a Boolean function f whose disjunctive normal form (DNF) has t terms, then most of the nonzero Walsh coefficients are also concentrated on polynomial poly(t) number of coefficients.

Furthermore, our concept of kth order sensitivity can be regarded as the local version of the well known noise sensitivity, which is a global concept. For a function $f : \{-1, 1\}^n \to \{-1, 1\}$ and $\delta \in [0, 1]$, the noise sensitivity $\mathbf{NS}_\delta[f]$ is the probability that $f(\mathbf{x}) \neq f(\mathbf{x}^\delta)$, when $\mathbf{x}$ is uniformly random and $\mathbf{x}^\delta$ is formed from $\mathbf{x}$ by flipping every bit independently with probability δ [15]. For example, it can be shown that the noise stability of the constant function is 1, and that of the majority function, Maj_n satisfies $\lim_{\substack{n \to \infty \\ n \text{ odd}}} \mathbf{NS}_\delta[\mathrm{Maj}_n] = \frac{2}{\pi}\sqrt{\delta} + \mathcal{O}(\delta^{3/2})$. The noise sensitivity was used in low-degree algorithms [10], which are very important in cryptography and learning, coding theory and complexity as a local list-decoding algorithms for the Hadamard codes [15]. These are just a few samples on how our concept is connected to cryptography and coding theory.

3. The recursive amplification method and related results

We noted already that fixing the value of a function corresponding to $n + 1$ input points to make a function fully sensitive, will restrict the polynomial degree to $\Omega(\sqrt{n})$. This implies that a fully dual sensitive function has resiliency $n - \Omega(\sqrt{n})$. We next explore these restrictions by performing a case study of functions on small number of variables $n \le 6$ along with some basic results. We further look into the recursive amplification method, which is also commonly known as function composition. This is a well known technique that is used to obtain super-linear separation between n and $\mathrm{pdeg}(f)$ and is also used to obtain super-linear separation between $s(f)$ and $\mathrm{pdeg}(f)$. In this section we use this technique and obtain the following results:

1. A slight modification of the recursive amplification method to obtain super-linear separation between $s(f)$ and $\mathrm{pdeg}(f)$ by starting from any candidate base function.
2. We build highly resilient functions with good nonlinearity, $\mathcal{O}(n)$ circuit size and $\mathcal{O}(\log n)$ circuit depth.
3. We obtain super-linear separation between number of variables n and polynomial degree $\mathrm{pdeg}(f)$ for functions with constant order sensitivity.

It is easy to see that if a function is nth order sensitive then its real polynomial degree is also n. This is because a function can be nth order sensitive if for a point $\mathbf{a} \in \mathbb{F}_2^n$, $f(\mathbf{a}) = 0$ and $f(\mathbf{x}) = 1,\ \forall \mathbf{x} \in \mathbb{F}_2^n \setminus \{\mathbf{a}\}$. Then such a function will have an odd number of ones in its truth table and thus will have $\mathrm{pdeg}(f) = n$.

The first separation between n and polynomial degree can be found for $(n - 1)$-st order sensitivity when n is odd, and for $(n - 2)$-nd order, otherwise.

Lemma 3.1. *The maximum value of k such that there exists a kth order sensitive function with polynomial degree less than n is $n-1$, where n is odd, and $n-2$, otherwise.*

Proof. We prove this result via the resiliency idea. The dual of an $(n, k, n-1)$-function is an $[n, k, 0]$-function, which is a balanced kth order dual sensitive function.

Let g be an $(n-1)$-st order dual sensitive function with respect to a point $\mathbf{y} \in \mathbb{F}_2^n$. Without loss of generality we assume that $f(\mathbf{y}) = 0$. Then, corresponding to all the points that can be obtained by flipping an odd number of bits, the output is 0, and 1, otherwise, where the maximum number bits that can be flipped is $n-1$. That is, $g(\overline{\mathbf{y}})$ can be both zero or one, which does not affect the sensitivity order, and the output corresponding to all other $2^n - 1$ points is fixed.

Case n odd: The minimum number of points $\mathbf{x}$ for which $f(\mathbf{x})$ must be zero is

$$1 + \binom{n}{1} + \binom{n}{3} + \cdots + \binom{n}{n-2} = 2^{n-1}.$$

We can then fix $f(\mathbf{y}) = 1$ and the other $2^{n-1} - 1$ points have output 1, by definition, which gives us an $n-1$-th order dual sensitive 0-resilient function whose dual $f = g \oplus \mathcal{L}_n$ is then an $(n, n-1, n-1)$-function.

Case n even: Here, the minimum number of points $\mathbf{x}$ for which $f(\mathbf{x}) = 0$ is

$$1 + \binom{n}{1} + \binom{n}{3} + \cdots + \binom{n}{n-1} = 2^{n-1} + 1$$

and thus the corresponding function cannot be balanced.

On the other hand if we try to form an $(n-2)$-nd order dual sensitive function, then the restriction reduces by $\binom{n}{n-1}$ and becomes $2^{n-1} - n + 1$. Meanwhile, the restriction on the minimum number of points with $f(\mathbf{x}) = 1$ remains the same and then the remaining n points can be fixed accordingly to make the function balanced, which gives us an $[n, n-2, 0]$-function whose dual is an $(n, n-2, n-1)$-function. □

It is easy to see that if there exists a kth order sensitive function f with polynomial degree p, then there exist functions of the same or smaller polynomial degree for $(j < k)$-th order sensitive functions as well, and the standard implications about resiliency-dual sensitivity follow.

Now we discuss the recursive amplification method and observe how it gives us functions f^d on $n = d^u$ variables with low real polynomial degree starting from a function f on small number of variables d, that we call as the base function. We further observe that this method in its normal form does not always guarantee full sensitivity in f^u even when f is fully sensitive. In this regard we slightly modify the recursive amplification process so that it always preserves the sensitivity in the function f^u formed as a result of the process. Before describing the method and the modifications, let us first observe the sensitivity-polynomial degree (dual sensitivity-resiliency) relations in functions on 3 variables to motivate the need for such modifications.

Dimension $n = 3$:

There cannot be a 3 variable function with $\mathtt{pdeg}(f) = 1$ and therefore the possible function categories are as follows:

1. $[3, 1, 0]$-functions (correspondingly, $(3, 1, 2)$-functions).
2. $[3, 2, 0]$-functions (correspondingly, $(3, 2, 2)$-functions).

$[3, 1, 0]$-*Functions:*

$g_1(x) = x_1x_2 \oplus x_1x_3 \oplus x_2x_3$, $g_2(x) = x_2 \oplus x_3 \oplus x_1x_2 \oplus x_1x_3 \oplus x_2x_3$, $g_3(x) = x_1 \oplus x_3 \oplus x_1x_2 \oplus x_1x_3 \oplus x_2x_3$, $g_4(x) = x_1 \oplus x_2 \oplus x_1x_2 \oplus x_1x_3 \oplus x_2x_3$, and their complements. Therefore the $(3, 1, 2)$-functions will be their duals obtained by adding the all variable linear function $\mathcal{L}_4 = x_1 \oplus x_2 \oplus x_3 \oplus x_4$ to g and this gives us the $(3, 1, 2)$-functions $f_i = g_i \oplus \mathcal{L}_n$, $1 \le i \le 4$.

Here it should be noted that the function f_1 is a symmetric function and the other functions f_2, f_3 and f_4 are all isomorphic up to permutation of variable names.

$[3, 2, 0]$-*Functions:*

Interestingly, all the four aforementioned functions and their complements are also $[3, 2, 0]$-functions (and their duals are second order sensitive functions of polynomial degree of 2). That is, all first order sensitive functions are also second order sensitive (where the polynomial degree of the functions is 3). However, we will observe that is indeed not the case for higher values of n, as one should expect.

3.1. The recursive amplification method and its modification

Recursive amplification was used to obtain the largest known separation between sensitivity and polynomial degree of Boolean functions [3], as well as the first example of separation between exact quantum query complexity and deterministic query complexity [1], among other separation results. Let f be a function on d variables $x_1, x_2 \dots x_d$ with polynomial degree p. Then the function f^2 of d^2 variables defined as

$$f^2(x_1, \dots, x_{d^2}) = f\big(f(x_1, \dots, x_d), f(x_{d+1}, \dots, x_{2d}), \dots, f(x_{d^2-d+1}, \dots, x_{d^2})\big)$$

has polynomial degree p^2. This process can be generalized as follows

$$f^{i+1}(x_1, \dots, x_{d^{i+1}}) = f\big(f^i(x_1, \dots, x_{d^i}), \dots, f^i(x_{d^{i+1}-d^i+1}, x_{d^{i+1}})\big).$$

Then we have the function f^u on $n = d^u$ variables such that $\mathsf{pdeg}(f^u) = p^u$ for any function f with d variables. Here, we note that if we replace any instance of f^i with $\overline{f^i}$ or any function with the same polynomial degree as f^i in any step of the recursion the results remain unchanged, as expressed by the following fact.

Fact 3.2. *Let f be a function defined on d variables with $\mathsf{pdeg}(f) = p_1$ and $f_1, f_2, \dots, f_d$ be functions each of d variables of polynomial degrees p_2. Then the function*

$$\hat{f} = f\big(f_1(x_1, \dots, x_d), \dots, f_k(x_{d^2-d+1}, \dots, x_{d^2})\big)$$

has $\mathsf{pdeg}(\hat{f}) = p_1 p_2$.

1. $f^1 = f$, where f is a function on d variables.
2. $f^{i+1}\left(x_1, \dots, x_{d^{i+1}}\right) = f\left(f^i\left(x_1, \dots, x_{d^i}\right), \dots, f^i\left(x_{(d-1)d^i+1}, \dots, x_{d^{i+1}}\right)\right)$.

Thus if we can also increase the sensitivity of the function in the recursive amplification process in the same rate, then starting from a base (d, x, y)-function f we can recursively amplify it to get a (d^u, x^u, y^u)-function f^u, which in fact gives us super-linear separation whenever $x > y$. However, although the polynomial degree of f^u is guaranteed, the case of sensitivity is not so straightforward. Consider the functions f_1 and f_2 defined on 3 variables, which have the properties:

- $\mathsf{pdeg}(f_1^2) = 4$.
- $s(f_1^2) = 9$.
- $\mathsf{pdeg}(f_2^2) = 4$.
- $s(f_2^2) = 7$.

The difference lies in which points of the function, the full sensitivity is achieved and its relation to the truth table of the function. We call such points maximally/fully sensitive.

The fully sensitive points (x_1, x_2, x_3) of f_1 are $(0, 0, 0)$ and $(1, 1, 1)$ with $f_1(0, 0, 0) = 0$ and $f_1(1, 1, 1) = 0$. Whereas the fully sensitive points of f_2 are $(0, 0, 1)$ and $(1, 1, 0)$ with $f_2(0, 0, 1) = 1$ and $f_2(1, 1, 0) = 1$.

Then the function f_1^2 is fully sensitive at the point $\mathbf{x}_0 = (0, 0, 0, 0, 0, 0, 0, 0, 0)$, since $f_1^2(\mathbf{x}_0) = f_1\big(f_1(0, 0, 0), f_1(0, 0, 0), f_1(0, 0, 0)\big)$. So if any one bit is changed the corresponding inner f_1 function's value becomes 1, which in turn changes the value of the outer f_1 function as f_1 is fully sensitive at the point $(0, 0, 0)$.

However, this situation does not arise for the function f_2^2. This is because in f_2^2, for the function to have full sensitivity, the inner functions need to act on fully sensitive points, i.e. $(x_1, x_2, x_3), (x_4, x_5, x_6), (x_7, x_8, x_9) \in \{(0, 0, 1), (1, 1, 0)\}$. But since both for both the points f_2 outputs 1, the outer f_2 function in f_2^2 always gets $(1, 1, 1)$ as input and it is not fully sensitive at that point. But instead, if we define $\hat{f}_2^2 = f_2\big(f_2(x_1, x_2, x_3), f_2(x_4, x_5, x_6), \overline{f_2(x_7, x_8, x_9)}\big)$, then if we fix the input for each inner function to be a fully sensitive point, then the corresponding input point for the outer f_2 is $(1, 1, 0)$ and the function is fully sensitive at this point. This gives a sufficient condition for successfully applying recursive amplification for any base function. This result also holds for higher order sensitivity.

Theorem 3.3. *Let f be a kth order sensitive function on d variables with $\mathsf{pdeg}(f_1) = p$ and $\mathbf{y} = (y_1, y_2, \dots, y_d)$ being the input with respect to which the function exhibits kth order sensitivity. We define the function f^u on d^u variables such that:*

1. $\mathbf{y}^1 = \mathbf{y}$
2. $\mathbf{y}^i \in \mathbb{F}_2^{d^i}$ *is obtained by concatenating d copies of $\mathbf{y}^{i-1}$.*
3. $f^1 = f$
4. $f^i = f\Big(f^{i-1}(x_1, \dots, x_{d^{i-1}}) \oplus f^{i-1}(\mathbf{y}^{i-1}) \oplus y_1, \dots, f^{i-1}(x_{jd^{i-1}+1}, \dots, x_{(j+1)d^{i-1}}) \oplus f^{i-1}(\mathbf{y}^{i-1}) \oplus y_j, \dots, f^{i-1}(x_{(d-1)d^{i-1}+1}, \dots, x_{d^i}) \oplus f^{i-1}(\mathbf{y}^{i-1}) \oplus y_d\Big)$.

Then f^u is a kth order sensitive function on $n = d^u$ variables and $\mathsf{pdeg}(f) = p^u$ with kth order sensitivity achieved at the input point $\mathbf{y}^u$.

Proof. Here we call the function f as the base function. Let us denote by $[\mathbf{x}]_k$ any input point that can be obtained by flipping at least 1 and at most k variables of $\mathbf{x} \in \mathbb{F}_2^n$. Thus if a function f is kth order sensitive at the point $\mathbf{y}$ then $f([\mathbf{y}]_k) = \overline{f(\mathbf{y})}$ by definition. We now prove the result using induction on u. The result holds for $u = 1$ by definition. Assume the result holds for $u-1$ and we need to show that the function f^u has kth order sensitivity at y^u. The value of the function at $\mathbf{y}^u$ is

$$f^u(\mathbf{y}^u) = f\Big(f^{u-1}(\mathbf{y}^{u-1}) \oplus f^{u-1}(\mathbf{y}^{u-1}) \oplus y_1, \ldots, f^{u-1}(\mathbf{y}^{u-1}) \oplus f^{u-1}(\mathbf{y}^{u-1}) \oplus y_d\Big) = f(\mathbf{y}).$$

Let us now select any $i \le k$ variables whose value we wish to flip resulting in an input point of the form $[\mathbf{y}^u]_k$. We define the d tuple $S = (s_1, s_2, s_d)$ where s_i denotes the number of bits to be flipped between $x_{(i-1)d^{u-1}+1}$ and $x_{id^{u-1}}$. Thus $0 \le s_i \le k,\ \forall i$. If $s_i = 0$ then

$$\begin{aligned} &f^{u-1}(x_{id^{u-1}+1}, \ldots, x_{(i+1)d^{u-1}}) \oplus f^{u-1}(\mathbf{y}^{u-1}) \oplus a_i \\ &= f^{u-1}(\mathbf{y}^{u-1}) \oplus f^{n-1}(\mathbf{y}^{u-1}) \oplus a_i = a_i. \end{aligned}$$

If $1 \le s_i \le k$ then

$$\begin{aligned} &f^{u-1}(x_{id^{u-1}+1}, \ldots, x_{(i+1)d^{u-1}}) \oplus f^{u-1}(\mathbf{y}^{u-1}) \oplus a_i \\ &= f^{u-1}([\mathbf{y}^{u-1}]_k) \oplus f^{u-1}(y^{u-1}) \oplus a_i = \overline{a_i}. \end{aligned}$$

The number of nonzero values in S are at most k, which would change at most k of the d points y_i in the base function's input to $\overline{y_i}$ and result in an input to f of the form of $f([y]_k)$. Thus for the function f^u we have $f^u([\mathbf{y}^u]_k) = f([\mathbf{y}]_k) = \overline{f(\mathbf{y})} = \overline{f^u(\mathbf{y}^u)}$.

The polynomial degree result holds from the basic definition of recursive amplification as $\mathsf{pdeg}(f) = \mathsf{pdeg}(\overline{f})$ and this completes the proof. □

It turns out that for the NAE function and the Kushilevitz's function, the normally known recursive amplification process works, without the modifications. Thus starting with any $(3, 3, 2)$ function f we can use the recursive amplification to obtain a $(3^u, 3^u, 2^u)$ function f^u and for this we have $s(f^u) = \mathsf{pdeg}(f^u)^{\frac{\log 3}{\log 2}} \approx \mathsf{pdeg}(f^u)^{1.58}$.

Now we look into the functions of 4 variables and then discuss how the recursive amplification method can be used to obtain highly resilient functions with good nonlinearity.

3.2. Low cost resilient functions with recursive amplification

Let us consider a function f on d variables with $\mathsf{pdeg}(f) = p < d$, where d is a constant. Then we can recursively amplify the function to obtain a function f^u on $n = d^u$ variables with $\mathsf{pdeg}(f^u) = p^u = n^{\frac{\log p}{\log d}}$. Now if we add the all variables linear function to it we get an n-variable function with $n - n^{\frac{\log p}{\log d}} - 1$ resiliency. However, there already exist many methods of obtaining Boolean functions with high resiliency and other cryptographically important properties such as high nonlinearity.

Here the advantage of the recursive amplification method is the circuit size for building such functions. Building efficient low depth circuits for cryptographically important functions with large number of input variables is a challenging problem. In this regard, the work by Sarkar et al. [18] is important. This work shows how to start with an m-resilient function on some d variables and generate an $m+u$-resilient function on $n = d+u$ variables that requires $\mathcal{O}(u)$ depth, which is effectively $\mathcal{O}(n)$ as d is constant for any given construction. In fact, this has been the best known result in this direction for almost two decades in building efficient circuits for resilient functions on large variables starting from base functions. In this direction we observe that the recursive amplification method can be used to obtain highly resilient functions on large number of variables with just $\mathcal{O}(\log n)$ depth.

Theorem 3.4. *Given a function f on d variables with $\mathsf{pdeg}(f) = p < d$ we can obtain a function on g^u on $n = d^u$ variables with resiliency $n - n^{\frac{\log p}{\log d}} - 1$ such that there is a circuit of linear size and logarithmic depth (in n) for it.*

Proof. We first define f^u as the function obtained recursively amplifying the function f, u times, which gives us a function on d^u variables with $\mathsf{pdeg}(f^u) = n^{\frac{\log p}{\log d}}$. Let us assume the circuit corresponding to the base function on d variables consists of some c_d gates and has a depth of t_d. This circuit takes in d input variable bits and outputs a single bit. Then the circuit corresponding to f^u can be built using the circuits for f in a layered manner in the following way.

- In the first layer there are total d^{u-1} circuits each taking in d variables each as input bits.
- In the ith layer there d^{u-i-1} circuits each taking as input d of the $du-i$ output bits from the previous layer.

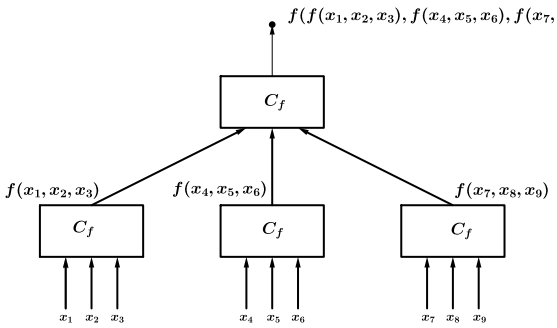

Fig. 1. Example of a circuit corresponding to recursive amplification.

- The final layer contains a single circuit, whose output is the output of the final function.

Then the total number of circuit instances of f to be used is $\sum_{i=0}^{u-1} d^i = \frac{d^u-1}{u-1}$ and the gate count is $c_d \times \frac{d^u-1}{u-1} = \mathcal{O}(d^u) = \mathcal{O}(n)$. Moreover, the depth of this circuit is $u \times t_d$ as the circuit for f is set up in u layers, which gives as a circuit for f^u with $\mathcal{O}(\log_d n)$ depth.

Now if we XOR the parity of all the input bits to this output we obtain a $n - n^{\frac{\log p}{\log d}} - 1$ function g^u via the resiliency-polynomial degree connection. The parity of the input bits can be simply obtained in parallel using n gates and $\log n$ depth, which gives us the result. □

Fig. 1 gives an example of building a 9 variable function using 4 instances of the circuit C_f corresponding to a 3 variable function f. Here observe that if the base function f has full sensitivity, then the function g^u will also be fully dual sensitive. Given our modified recursive amplification process in Theorem 3.3, we can have this property starting from any base function, and this does not the depth of the circuit at all and the gate count beyond a constant factor.

3.3. Other cryptographic properties

Having discussed the efficiency of this method, let us now look into the nonlinearity of such functions, which is another very important cryptographic property. We consider the following examples:

- Recursively amplifying a $(3, 1, 2)$-variable function f_3 to get a 9 variable function $f_9(\mathbf{x}) = f_3\big(f_3(x_1, x_2, x_3) \oplus a_1, f_3(x_4, x_5, x_6) \oplus a_2, f_3(x_7, x_8, x_9) \oplus a_3\big)$, $a_i \in \{0, 1\}$ and then adding the all variable linear function to obtain g_9, which is 4 resilient, irrespective of the choice of a_i. However, depending on the choice of a_i, the nonlinearity can either be 96 or 192. Here the circuit for f_3 needs 5 XOR gates and 3 AND gates and thus the circuit for f_9 only requires 20 XOR gates and 12 AND gates, and we can obtain g_9 by adding the all variable parity function, which requires a further 9 XOR gates, which makes the total gate count to be only 41.
- Recursively amplifying a 4 variable function f_4 to get a 16 variable function f_{16} by defining $f_{16}(\mathbf{x}) = f_4\big(f_4(x_1, x_2, x_3, x_4) \oplus a_1, f_4(x_4, x_6, x_7, x_8) \oplus a_2, f_4(x_9, x_{10}, x_{11}, x_{12}) \oplus a_3, f_4(x_{13}, x_{14}, x_{15}, x_{16}) \oplus a_4\big)$, $a_i \in \{0, 1\}$ and then adding the all variable linear function to obtain g_{16}. Here we obtain 16 variable 11-resilient functions with nonlinearity as high as $24576 = 2^{15} - 2^{13}$ where the best possible nonlinearity for 16 variable functions is $2^{15} - 2^7$ (bent functions).
- We also find (6, 1, 3)-functions in Section 4.3 that require 24 AND gates and 21 XOR gates to compute. We can then get a 36 variable 26 resilient fully dual sensitive function with only 168 AND gates and 147 XOR gates.

Here it should be noted that the algebraic degree of the function g^u is upper bounded by the real polynomial degree of f^u, which is $n^{\frac{\text{pdeg}(f)}{d}}$. However, we can also have an algebraic degree-resiliency trade off in this construction if we use two kinds of base functions on d variables, one with algebraic degree (and thus polynomial degree) of d and the other having lower polynomial degree, and we can add these functions in different stages of the recursion to obtain functions with various resiliency and algebraic degree values. Thus we have two problems that need further attention.

1. How much can we increase the nonlinearity in f^u when the functions used in recursion are of the form f or $\overline{f}$, where f is the base function defined on some d variables?
2. How good cryptographic profiles can we obtain using this method and using multiple functions on d variables throughout the recursion process?

These properties should be studied more elaborately later and is beyond the scope of this present discussion.

3.4. Constant higher order sensitivity

We now show a simple but fundamental implication of Lemma 3.1 and Theorem 3.3. These two results directly indicate there is super-linear separation between the number of variables and polynomial degree for any kth order sensitive function, where k is some constant.

Theorem 3.5. *Given any constant k there exists a kth order sensitive function f on n variables such that $\mathsf{pdeg}(f) = n^{\frac{\log k}{\log k+1}}$, if k is even and $\mathsf{pdeg}(f) = n^{\frac{\log k+1}{\log k+2}}$, if k is odd.*

Proof. Given any k, we form a function f in the following manner:

- If k is odd we form a $(k+2)$-variable balanced kth order dual sensitive function g which is a $[k+2, k, 0]$-function as per Lemma 3.1. Then we take its dual $f = g \oplus \mathcal{L}_n$, which is a $(k+2, k, k+1)$-function.
- If k is odd we form a $(k+1)$-variable balanced kth order dual sensitive function g which is a $[k+1, k, 0]$-function. Then we take its dual $f = g \oplus \mathcal{L}_n$, which is a $(k+1, k, k)$-function, via Lemma 3.1.

Next we use the recursive amplification process described in Theorem 3.3. This gives a $\left((k+2)^u, k, (k+1)^u\right)$-function, when k is even and a $\left((k+1)^u, k, k^u\right)$-function, otherwise, which gives us the desired super-linear advantage. □

We will soon observe the sensitivity-polynomial degree relations in functions from 4 up to 10 variables in Section 4. To conclude this section, let is now present the results on nontrivial lower bounds on the nonlinearity of functions based on the recursive amplification method.

3.5. Results on the nonlinearity of functions obtained with recursive amplification

Let f be a Boolean function on s variables and $g_1, g_2, \ldots, g_s$ be s Boolean functions on u variables. We now define a us-variable Boolean function $\hat{f}$ by

$$\hat{f}(x_1, \ldots, x_{us}) = f(g_1(x_1, \ldots, x_u), g_2(x_{u+1}, \ldots, x_{2u}), \ldots, g_s(x_{us-u+1}, \ldots, x_{us})). \tag{3.1}$$

Theorem 3.6 ([19]). *Let $\hat{f}$ be the function defined in (3.1), then for any $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_s) \in \mathbb{F}_2^{us}$ (where $\mathbf{w}_i \in \mathbb{F}_2^u$ for any $1 \le i \le s$), we have*

$$W_{\hat{f}}(\mathbf{w}) = 2^{-s} \sum_{\mathbf{v} \in \mathbb{F}_2^s} \left(W_f(\mathbf{v}) \prod_{i=1}^{s} W_{v_i g_i}(\mathbf{w}_i) \right),$$

where $\mathbf{v} = (v_1, v_2, \ldots, v_s) \in \mathbb{F}_2^s$.

Theorem 3.7. *Let $\hat{f}$ be the function defined in (3.1) by taking $g_1, g_2, \ldots, g_k \in \mathbb{F}_2^k$ to be balanced functions. Then for any $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_s) \in \mathbb{F}_2^{ks}$ (where $\mathbf{w}_i \in \mathbb{F}_2^k$ for any $1 \le i \le s$), we have*

$$W_{\hat{f}}(\mathbf{w}) = 2^{-s} W_f(\mathbf{v}') \prod_{i=1}^{s} \left(\frac{1 + (-1)^{v_i'}}{2} \cdot 2^k + \frac{1 - (-1)^{v_i'}}{2} \cdot W_{g_i}(\mathbf{w}_i) \right),$$

where $\mathbf{v}' = (v_1', v_2', \ldots, v_s') \in \mathbb{F}_2^s$ with $v_i' = 1$ if and only if $\mathbf{w}_i \neq \mathbf{0}_k$. Furthermore, we have

$$NL(\hat{f}) \ge 2^{ks-1} - 2^{ks-k-s-1} \left(2^s - 2NL(f) \right) \left(2^k - 2 \min_{1 \le i \le s} \{NL(g_i)\} \right),$$

where NL denotes the nonlinearity.

Proof. Note that the value of the Walsh transform at any nonzero point of a constant function is equal to null. Thus, for any $1 \le i \le s$, we have $W_0(\mathbf{w}_i) = 0$ if $\mathbf{w}_i \neq \mathbf{0}_k$. As g_i's are balanced, we have $W_{g_i}(\mathbf{0}_k) = 0$ for any $1 \le i \le s$. So we have $\prod_{i=1}^{s} W_{v_i g_i}(\mathbf{w}_i) = 0$ if $v_i \neq v_i'$. Then by Theorem 3.6 we have

$$W_{\hat{f}}(\mathbf{w}) = 2^{-s} \sum_{\mathbf{v} \in \mathbb{F}_2^s} \left(W_f(\mathbf{v}) \prod_{i=1}^{s} W_{v_i g_i}(\mathbf{w}_i) \right) = 2^{-s} W_f(\mathbf{v}') \prod_{i=1}^{s} W_{v_i' g_i}(\mathbf{w}_i).$$

Then our first assertion comes from the fact $W_{v_i' g_i}(\mathbf{w}_i)$ equals 2^k if $v_i' = 0$ and $W_{g_i}(\mathbf{w}_i)$ if $v_i' = 1$. Note that $|W_{g_i}(\mathbf{w}_i)| \le \max_{\mathbf{w}_i' \in \mathbb{F}_2^k} \{|W_{g_i}(\mathbf{w}_i')|\} \le 2^k$ for any $1 \le i \le s$. Then we have $|\prod_{i=1}^{s} W_{v_i g_i}(\mathbf{w}_i)| \le 2^{ks-k} \cdot \max_{1 \le i \le s, \mathbf{w}_i' \in \mathbb{F}_2^k} \{|W_{g_i}(\mathbf{w}_i')|\}$ by setting the

Hamming weight of $\mathbf{v}'$ to be 1. Then we can obtain that the nonlinearity of $\hat{f}$ is at least $2^{ks-1} - 2^{ks-k-s-1} \max_{\mathbf{u}\in\mathbb{F}_2^s}\{|W_f(\mathbf{u})|\} \cdot \max_{1\le i\le s, \mathbf{w}_i'\in\mathbb{F}_2^k}\{|W_{g_i}(\mathbf{w}_i')|\}$, which gives our second assertion. This finishes the proof. □

Lemma 3.8. *Let f be an n-variables function and $\hat{f}$ be an n^2-variable function defined as $\hat{f}(x_1, \ldots, x_{n^2}) = f\left(f(x_1, \ldots, x_n) + a_1, \ldots, f(x_{n^2-n+1}, \ldots, x_{n^2}) + a_n\right)$, where a_i's belong to $\mathbb{F}_2$. Then the nonlinearity of $\hat{f}$ is at least $2^{n^2-n+1}NL(f) - 2^{n^2-2n+1}(NL(f))^2$. Moreover, if f is m resilient, then $\hat{f}$ is $(m+1)^2 - 1$ resilient; if $w_f(\mathbf{w}) = 0$ for any $\mathbf{w}$ with Hamming weight no less than m', then f is $n^2 - (m'-1)^2 - 1$ resilient.*

Proof. The lower bound on nonlinearity of $\hat{f}$ directly follows from Theorem 3.7. By observing the value of $W_f(\mathbf{v}')\prod_{i=1}^s W_{v_i'g_i}(\mathbf{w}_i)$ in the proof of Theorem 3.7, then we can easily obtain the rest assertions. This completes the proof. □

Corollary 3.9. *Let f be an n-variables Boolean function and f_i $(i \ge 2)$ be the function defined above. Then we have*

$$NL(f^i) \ge 2^{n^i-n^{i-1}}NL(f^{i-1}) + 2^{n^i-n}NL(f) - 2^{n^i-n^{i-1}-n+1}NL(f^{i-1})NL(f),$$

with $NL(f^1) = NL(f)$.

4. $s(f)$-pdeg(f) results for functions on small number of variables

We have already discussed the case of functions of 3 variables. Let us now observe the situations for functions on higher number of variables. The functions on up to 5 variables can be exhaustively searched to obtain all existing combinations. However, for functions on 6 and more variables, an exhaustive search is not possible given the size of search space (2^{64} for $n = 6$, 2^{128} for $n = 7$ and so on), and we instead use the properties of resiliency and dual sensitivity to completely exhaust the case of fully sensitive functions for $n = 6$ and $n = 7$ in terms of obtaining all functions and proving non-existence respectively.

4.1. [4, −, 0]-*Functions:*

It can be checked with a simple search that there does not exist any fully sensitive function on 4 variables with $\mathsf{pdeg}(f) = 2$. In fact, if that would have been the case then we could use the recursive amplification method to obtain a function f^u on 4^u variables with $s(f^u) = 4^u$ and $\mathsf{pdeg}(f^u) = 2^u$, which would give us quadratic separation between sensitivity and polynomial degree.

Thus we look into the (4, −, 3) functions, which are duals of [4, −, 0] functions. We have the following counts.

- There are approximately 3760($\approx 2^{11.87}$) many [4, 1, 0]-functions.
- Only 256($= 2^8$) of these functions are [4, 2, 0]-functions.
- There are no [4, 3, 0]-functions.

4.2. [5, −, 1]-*Functions:*

In the case of 5 variable functions, the possible polynomial degree is between 3 and 4. As we have already observed the case of 0-resiliency ($n - 1$ polynomial degree in the dual) for 3 and 4, we compute the 1-resilient functions in this case, and get the following counts:

$$\#[5,1,1]\text{-functions} = 12304(\approx 2^{13.58}), \qquad \#[5,3,1]\text{-functions} = 0$$
$$\#[5,2,1]\text{-functions} = 2464(\approx 2^{11.2}), \qquad \#[5,4,1]\text{-functions} = 0.$$

Finally let us look into the case of 6 variable functions, for which we have the best base function for first order sensitivity, which is the Kushilevitz function.

4.3. [6, −, 2]-*Functions:*

There are total 2^{64} Boolean functions on 6 variables, and checking the resiliency and sensitivity of all possible functions requires computational resources that is unattainable. We instead use properties of dual sensitivity and resiliency to obtain all possible [6, 1, 2]-functions by concatenating the truth tables of two 5 variable functions. Any 6 variable function f can be written as $f(x_1, \ldots, x_6) = (1 \oplus x_6)f_2(x_1, \ldots, x_5) \oplus x_6 f_2(x_1, \ldots, x_5)$ Where f_1 and f_2 are functions on 5 variables. Then we have the following constraints on the properties of f.

1. If f is 2-resilient then either both f_1 and f_2 are 1-resilient or both are 2-resilient [11].
2. If f is fully dual sensitive then at least one of f_1 and f_2 are fully dual sensitive. This is easy to see as if neither f_1 nor f_2 are dual sensitive then there is no input point for which the whole function can have full dual sensitivity.

Using these constraints we get the full characterization of $[6, -, 2]$-functions, which was not previously reported.

- We find that there are $33632 (\approx 2^{15.03})$ many $[6, 1, 2]$-functions. Here it should be noted that the dual of any such function is a $(6, 1, 3)$-function. We can use the modified recursive amplification technique of Theorem 3.3 on all such functions to obtain $(6^u, 1, 3^u)$-functions, which gives us the best known separation between sensitivity and polynomial degree, same as the function by Kushilevitz [13].
- We also get $192 (\approx 2^{7.6})$ many $[6, 2, 2]$-functions, and this gives us the maximum super-linear separation between number of variables and real polynomial degree in second order sensitive functions, which is $\mathsf{pdeg}(f) = n^{\frac{\log 3}{\log 6}}$. Furthermore, there is no $[6, > 2, 2]$-functions.

4.4. Nonexistence of $(7, 1, 3)$-functions (as well as $[7, 1, 3]$-functions)

The existence or non-existence of a $(7, 1, 3)$-function is central to understanding the maximum separation between $s(f)$ and $\mathsf{pdeg}(f)$. If there does exist a $(7, 1, 3)$-function then we can obtain a $(7^u, 1, 3^u)$-function using the recursive amplification method, which gives $s(f) = \mathsf{pdeg}(f)^{\frac{\log 7}{\log 3}}$, improving on the best known result. However the total number of functions on 7 variables is 2^{128} and therefore checking all functions for this profile is not possible. In this direction we use a mixed integer linear program to check existence of such function.

Let f be a Boolean function on 7 variables. f has full sensitively and polynomial degree 3 if and only if there is a vector $\mathbf{x} \in \mathbb{F}_2^7$ such that $f(\mathbf{x}) \oplus (\mathbf{x}^i) = 1$ for every $1 \leq i \leq 7$ and $W_f(\mathbf{u}) = 0$ for every $\mathbf{u} \in \mathbb{F}_2^7$ with Hamming weight no less than 4. For every $\mathbf{x} \in \mathbb{F}_2^7$, by mixed integer linear program method with constraints on Walsh spectrum, GUROBI software shows that there is no 7 variable function with full sensitively and polynomial degree 3. Ranging over all vectors $\mathbf{x}$ in $\mathbb{F}_2^7$, we confirm that there is no such function.

4.5. Rotation symmetric functions for up to 10 variables

Even with our strategy, it not possible to search for all fully sensitive and higher order sensitive functions on more than 7 variables because of the size of the search space. In this regard we search 8,9 and 10 variable rotation symmetric functions, which is another cryptographically important class of functions to obtain with fully sensitive (first order sensitive functions) using the least possible polynomial degree (maximum resiliency in the dual function). We thus obtain the following results:

- There exists only 12 functions in the rotation symmetric class that are $[8, 1, 3]$-functions and none are $[8, 2, 3]$-functions. There cannot exist any $[8, \ell, 4]$-function, where $\ell > 0$.
- We also find 29 many $[9, 1, 4]$ functions rotation symmetric functions, out of which 27 are also $[9, 4, 2]$-functions. Furthermore, there cannot exist any $[9, \ell, 5]$-function, where $\ell > 0$. Here one should note that one can also obtain $[9, 1, 4]$ and $[9, 2, 4]$-functions by recursively amplifying a $(3, 1, 2)$ or a $(3, 2, 2)$-function respectively and then taking its dual, and these were the only known $(9, 1, 4)$-functions before now. However, we obtain $[9, 1, 4]$-rotation symmetric functions with a nonlinearity of 224, whereas the nonlinearity of the functions obtained through recursive amplification is 192. Thus, we obtain previously unknown $(9, 1, 4)$-functions. The advantage of using the recursive amplification process is its efficient circuit size and depth.
- There does not exist any $[10, 1, 5]$-rotation symmetric function. It should be noted that if we can obtain a $[10, 1, 5]$-function (provided such a function exists) then that would improve on the best known separation between sensitivity and polynomial degree. This is because we can then get a $(10, 1, 4)$-function and then recursively amplify the function using the modified amplification process described in Theorem 3.3 to get a $(10^u, 1, 4^u)$-function, thus giving $s(f) = (\mathsf{pdeg}(f))^{\frac{\log 10}{\log 4}}$ and since $\frac{\log 10}{\log 4} > \frac{\log 6}{\log 3}$ this would be an improvement on the best known result.
  In this regard it is interesting to observe that there are no $[6, 1, 2]$-rotation symmetric functions but there are many $[5, 1, 1]$-function, and we have indeed observed that there are $[9, 1, 4]$-rotation symmetric functions.

This concludes the study of sensitivity-polynomial degree (and higher order sensitivity-polynomial degree) study of functions on up to 10 variables. Next, we discuss how the recursive amplification method does not help us in providing non-constant separation between n and $\mathsf{pdeg}(f)$ in functions with super-constant order of sensitivity and then we design a construction based on the Maiorana-McFarland class of functions to obtain such results.

5. Super-constant higher order sensitivity and Maiorana-McFarland constructions

Until now we have discussed functions with a constant higher order sensitivity, and have found classes of functions f defined on the number of variables n for which we could obtain super-linear separation between n and $\mathsf{pdeg}(f)$ using the recursive amplification method. However, we cannot obtain any $t(n)$-order sensitive (or dual sensitive) function, where $t(n)$ is an increasing function on n.

Theorem 5.1. *The recursive amplification process cannot obtain a function that has super-constant order of sensitivity, where the process is as follows*

- *A base function f^1 on some d variables.*
- $f^k = f^1\left(\hat{f}^{k-1}, \ldots, \hat{f}^{k-1}\right)$ *where* $\hat{f}^{k-1} \in \{f^{k-1}, \overline{f^{k-1}}\}$.

Proof. Let us consider any function f on d variables and $t(n)$-order sensitivity where $t(n) = \Omega(1)$. Then f^n is a function on d^n variables and there exists $n_0 \in \mathbb{N}$ such that $t(d^{n_0}) > d$. However it is easy to see (via induction) that any function built with a base function on d variables cannot be dth order sensitive. □

Thus the recursive amplification process does not help us anymore when we consider $\Omega(1)$-order sensitive functions. In this regard we next explore the class of Maiorana-McFarland (MM) constructions, which is a cryptographically important method that has been used to obtain functions of desirable properties, such as resiliency and nonlinearity. Here, we use it from the perspective of polynomial degree-sensitivity to obtain results in the domain of functions with non-constant order of sensitivity. The layout of this section is as follows. We first discuss the Maiorana McFarland construction and then obtain separation between n and $\mathsf{pdeg}(f)$ while making the function first order sensitive (fully sensitive). We extend this construction while discussing first order sensitivity only for ease of understanding. We analyze the polynomial structure of these functions and obtain logarithmic separation between n and $\mathsf{pdeg}(f)$. Finally we show that this construction can be modified for super-constant orders of sensitivity (up to $o(\log n)$) with only a few tweaks.

5.1. Maiorana-McFarland construction

The Maiorana-McFarland (MM) construction [5] is based on dividing the input variable space into two parts and attaching different linear functions from one subspace to each point in the other subspace. This kind of construction gives rise to Boolean functions with different interesting properties, such as maximum possible first order nonlinearity for functions on even number of variables and high resiliency.

Definition 5.2. A function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is called an MM function if it can be expressed as $f(\mathbf{x}, \mathbf{y}) = (\phi(\mathbf{x}) \cdot \mathbf{y}) \oplus g(\mathbf{x})$, where:

- $\mathbf{x} \in \mathbb{F}_2^{n_1}$, $\mathbf{y} \in \mathbb{F}_2^{n_2}$, $n = n_1 + n_2$;
- ϕ is a mapping of the form $\phi : \mathbb{F}_2^{n_1} \to \mathbb{F}_2^{n_2}$;
- $a \cdot b = a_1b_1 \oplus a_2b_2 \oplus \ldots \oplus a_kb_k$, where $a = (a_1, \ldots, a_k)$, $b = (b_1, \ldots, b_k) \in \mathbb{F}_2^k$;
- $g : \mathbb{F}_2^{n_1} \to \mathbb{F}_2$ is an arbitrary Boolean function defined on the subspace $\mathbb{F}_2^{n_1}$.

Different restrictions on the map ϕ yield functions with different cryptographic properties. For example, if ϕ is a permutation, then n is even and the corresponding function is a bent function, for any choice of g. Similarly, if the weight of $\phi(\mathbf{x})$, denoted as $|\phi(\mathbf{x})|$ is strictly greater than m for all $\mathbf{x}$, then f is m-resilient, etc.

The Boolean functions via the MM construction can be visualized in different ways. We view them as different linear functions defined on $\mathbf{y}$ attached to activating values in $\mathbf{x}$. Let there be an MM Boolean function with any arbitrary map $\phi : \mathbb{F}_2^{n_1} \to \mathbb{F}_2^{n_2}$ and some Boolean functions $g : \mathbb{F}_2^{n_1} \to \mathbb{F}_2$. Corresponding to any $\mathbf{a} \in \mathbb{F}_2^{n_1}$, the quantity $\phi(\mathbf{a}) \cdot \mathbf{y}$ is essentially the outcome of the linear equation $\bigoplus_{\phi(\mathbf{a})_i=1} \mathbf{y}_i$. Thus $\phi(\mathbf{a}) \cdot \mathbf{y} \oplus g(\mathbf{a})$ equals $\left(\bigoplus_{\phi(\mathbf{a})_i=1} y_i\right) \oplus g(\mathbf{a})$ for all $\mathbf{y}$ in $\mathbb{F}_2^{n_2}$. Let us denote this function defined on $\mathbb{F}_2^{n_2}$ as $Lin_{\phi(\mathbf{a}),g(\mathbf{a})}$. Then we can express the function in the way described in Table 1.

This representation gives us a structural way of describing the real polynomial for any MM type function, which is essential to the functions we construct later.

We first define two real polynomial structures $\mathcal{P}_\mathbf{a} : \mathbb{F}_2^{n_1} \to \mathbb{R}$, $\mathbf{a} \in \mathbb{F}_2^{n_1}$ and $\mathcal{L}_{(\mathbf{b},c)} : \mathbb{F}_2^{n_2} \to \mathbb{R}$, $\mathbf{b} \in \mathbb{F}_2^{n_2}, c \in \mathbb{F}_2$. We let $\mathcal{P}_\mathbf{a}(\mathbf{x}) = \left(\prod_{\mathbf{a}_i=0}(1 - x_i)\right)\left(\prod_{\mathbf{a}_i=1} x_i\right)$, so

$$\mathcal{P}_\mathbf{a}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{a} \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

Now we define a polynomial structure that expresses the linear function $Lin_{\phi(\mathbf{a}),g(\mathbf{a})}$ corresponding to each $\mathbf{a} \in \mathbb{F}_2^{n_1}$. Any linear function on $\mathbf{y}$ can be expressed as $\oplus_{b_i=1} y_i$, $\mathbf{b} \in \mathbb{F}_2^{n_2}$, and the corresponding real polynomial is $\frac{1}{2}\left(1 - \prod_{b_i=1}(1 - 2y_i)\right)$.

Table 1
An arbitrary MM type Boolean function.

$\mathbf{x} \in \mathbb{F}_2^{n_1}$	$\mathbf{y} \in \mathbb{F}_2^{n_2}$	$f(\mathbf{x}, \mathbf{y}) : \mathbb{F}_2^{n_1+n_2} \to \mathbb{F}_2$
$000\ldots00(\mathbf{0}_{n_1})$	$000\ldots00$	$Lin_{\phi(\mathbf{0}_{n_1}),g(\mathbf{0}_{n_1})}(000\ldots00)$
	$000\ldots01$	$Lin_{\phi(\mathbf{0}_{n_1}),g(\mathbf{0}_{n_1})}(000\ldots01)$
	$\vdots$	$\vdots$
	$111\ldots11(\mathbf{1}_{n_2})$	$Lin_{\phi(\mathbf{0}_{n_1}),g(\mathbf{0}_{n_1})}(111\ldots11)$
$000\ldots01(\mathbf{0}_{n_1-1}1)$	$\mathbf{y}$	$Lin_{\phi(\mathbf{0}_{n_1-1}1),g(\mathbf{0}_{n_1-1}1)}$
$\vdots$	$\vdots$	$\vdots$
$111\ldots11(\mathbf{1}_{n_1})$	$\mathbf{y}$	$Lin_{\phi(\mathbf{1}_{n_1}),g(\mathbf{1}_{n_1})}$

Then the real polynomial corresponding to $\oplus_{b_i=1} y_i \oplus 1$ is $\frac{1}{2}\left(1+\prod_{b_i=1}(1-2y_i)\right)$. Thus we define $\mathcal{L}_{(\mathbf{b},c)}(\mathbf{y}) = \frac{1}{2} - \frac{(-1)^c}{2}\prod_{b_i=1}(1-2y_i)$, where the MM function $\mathbf{b}$ is defined by ϕ, and c is defined by g with respect to different points.

These structures allow us to generically express the real polynomial corresponding to any MM function, which we describe next.

Proposition 5.3. *Given an MM function $f(\mathbf{x}, \mathbf{y}) = (\phi(\mathbf{x})\cdot\mathbf{y})\oplus g(\mathbf{x})$ on n variables with $\mathbf{x} \in \mathbb{F}_2^{n_1}$ and $\mathbf{y} \in \mathbb{F}_2^{n_2}$, the corresponding real polynomial can be defined as*

$$p(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{a}\in\mathbb{F}_2^{n_1}} \mathcal{P}_{\mathbf{a}}(\mathbf{x})\, \mathcal{L}_{(\phi(\mathbf{a}),g(\mathbf{a}))}(\mathbf{y}). \tag{5.2}$$

Let us now note down a simple result that this polynomial structure entails.

Note 5.4. *For any $\mathbf{b} \in \mathbb{F}_2^{n_2}$ we have $\mathcal{L}_{\mathbf{b},c} + \mathcal{L}_{\mathbf{b},\bar{c}} = 1$ and $\mathsf{pdeg}(\mathcal{L}_{\mathbf{b},c}) = wt(\mathbf{b})$.*

Having discussed the Maiorana-McFarland construction and the structural results related to the polynomials we now show the first instance of separation. From here on we denote by $\mathbf{x}$ and $\mathbf{y}$ variables in $\mathbb{F}_2^{n_1}$ and $\mathbb{F}_2^{n_2}$, respectively, unless stated otherwise.

We first impose some constraints on the MM construction so that it has full sensitivity. We emphasize that while these constraints are sufficient they are not necessary.

Lemma 5.5. *Let f be an MM function defined by $f(\mathbf{x}, \mathbf{y}) = (\phi(\mathbf{x})\cdot\mathbf{y})\oplus g(\mathbf{x})$, where $\mathbf{x} \in \mathbb{F}_2^{n_1}$ and $\mathbf{y} \in \{0,1\}^{n_2}$ so that n_2 is even. Let us denote by $\mathbf{a} = \mathbf{1}_{n_1}$. If the function ϕ satisfies: $\phi(\mathbf{1}_{n_1}) = \mathbf{1}_{n_2}$, $g(\mathbf{1}_{n_1}) = 0$, $\phi(\mathbf{1}_{n_1-2}00) = \mathbf{1}_{n_2}$, $g(\mathbf{1}_{n_1-2}00) = 1$, $wt(\phi(\mathbf{a}^i)) \equiv 1 \mod 2$, $1 \le i \le n_1$, $g(\mathbf{a}^i) = 0$, $1 \le i \le n_1$, then $s(f) = n$.*

Proof. It suffices to show that $s(f, \mathbf{1}_n) = n$. By definition we have $\phi(\mathbf{1}_{n_1}) = \mathbf{1}_{n_2}$. Thus for any input of the form $\mathbf{1}_{n_1}||\mathbf{y}$ we have $f(\mathbf{1}_{n_1}||\mathbf{y}) = \bigoplus_{i=1}^{n_2} y_i$. Then if $y_i = 1$, for all i, we get $f(\mathbf{1}_{n_1}||\mathbf{1}_{n_2}) = 0$ as $n_2 \equiv 0 \mod 2$.

Corresponding to this point, if any one of the components in $\mathbf{y}$ is flipped, then the function evaluates to 1 as the output is the XOR of an odd number of variables. If any of the points (say $i : 1 \le i \le n_1$) in $\mathbf{x}$ is flipped then the output of the function is $f(\mathbf{a}^i||\mathbf{1}_{n_2})$. Since the number of variables of the linear functions corresponding to all such points $\mathbf{a}^i$ is odd, $g(\mathbf{a}^i)$ is 1 and all the input variables in $\mathbf{y}$ that are to be considered are set as 1, the output of the function at this point is 1. This shows that $s(f, \mathbf{1}_n) = n$ and the proof is complete. □

Let us formally define MM type functions of this kind to avoid redundancy going forward.

Definition 5.6. For an MM type function on $f(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})\cdot\mathbf{y} \oplus g(\mathbf{x})$ defined on $n = n_1 + n_2$ variables, if $\phi(\mathbf{1}_{n_1}) = \mathbf{1}_{n_2}$ and $wt(\phi(\mathbf{1}^i_{n_1})) \equiv n_2 + 1 \mod 2$, $1 \le i \le n_1$ we say that f belongs to $\mathbb{MM}_n$.

We denote the point $\mathbf{1}_n$ as the 0-th critical point and the point $\mathbf{1}^i_n$, $1 \le i \le n$ to be the ith critical point, where for $1 \le i \le n_1$ the bit flipped belongs to $\mathbf{x}$ and for $n_1 + 1 \le i \le n$ the bit flipped belongs to $\mathbf{y}$. We develop our construction in such a manner that the constraints described with respect to ϕ and g for these $n+1$ critical points are met. We now describe the construction technique. We first show a method of defining ϕ and g so that the corresponding function f in $\mathbb{MM}_n$ has $\mathsf{pdeg}(f) = n-1$. We then analyze in the next section how we can achieve non-constant (logarithmic) separation between $s(f)$ and $\mathsf{pdeg}(f)$.

5.2. An MM function with $s(f) = n$ and $\mathrm{pdeg}(f) = n - 1$

Let us first denote some notations that we shall use in the proof. We denote by $\mathbf{X_i^j}$ the monomial $\left(\prod_{k=i}^{j} x_k\right)$ and by $\mathbf{X^i}$ the monomial $\left(\prod_{k=1}^{i} x_k\right)$. We define the polynomial $\mathbf{L_{n_2}} = \prod_{i=1}^{n_2}(1-2y_i)$. Then we can write $\mathcal{L}_{\mathbf{1}_{\mathbf{n_2}}\mathbf{,0}} = \frac{1-\mathbf{L_{n_2}}}{2}$ and $\mathcal{L}_{\mathbf{1}_{\mathbf{n_2}}\mathbf{,1}} = \frac{1+\mathbf{L_{n_2}}}{2}$.

Theorem 5.7. *Let $f \in \mathbb{MM}_n$ be an MM function defined on $n = n_1 + n_2$ variables of input $(\mathbf{x}, \mathbf{y}) \in \mathbb{F}_2^{n_1} \times \mathbb{F}_2^{n_2}$, so that $n_1 \le n_2 \le n_1 + 1$ with even n_2. If f is defined using $\phi(\mathbf{1}_{n_1-2}00) = \mathbf{1}_{n_2}$, $g(\mathbf{x}) = \prod_{i=1}^{n_1-2} x_i(1-x_{n-1})(1-x_n)$, the sensitivity of f is n and the polynomial degree is $n-1$.*

Proof. First, $s(f) = n$, which follows directly from the constraints described and the subsequent proof of Lemma 5.5.

We now show that $\mathrm{pdeg}(f) = n - 1$. Apart from the constraints given, we have two more constraints as $f \in \mathbb{MM}_n$, namely, $wt(\phi(\mathbf{1}^i_{n_1})) \equiv 1 \mod 2$, $\phi(\mathbf{1}_{n_1}) = \mathbf{1}_{n_2}$. We have defined the real polynomial $p(\mathbf{x}, \mathbf{y})$ that represents f in Proposition 5.3. The degree of the polynomial is $\le \max(wt(\phi)) + n_1$. We know that $p(\mathbf{x}, \mathbf{y}) = \sum_{a\in\{0,1\}^{n_1}} \mathcal{P}_a(\mathbf{x})\left(\mathcal{L}_{(\phi(\mathbf{a}),g(\mathbf{a}))}(\mathbf{y})\right)$ has monomials of degree n only in the terms $\mathcal{L}_{b,c}$ where $wt(b) = n_2$, which is the case only when $b = \mathbf{1}_{n_1-2}00$ or $\mathbf{1}_{n_1}$. Additionally $g(\mathbf{1}_{n_1-2}00) = 1$ and $g(\mathbf{1}_{n_1}) = 0$. Therefore, we can write the polynomial as

$$p(\mathbf{x}, \mathbf{y}) = p'(\mathbf{x}, \mathbf{y}) + \mathcal{P}_{(\mathbf{1}_{n_1-2}00)}(\mathbf{x})\mathcal{L}_{\phi(\mathbf{1}_{n_1-2}00),g(\mathbf{1}_{n_1-2}00)}(\mathbf{y}) + \mathcal{P}_{(\mathbf{1}_{n_1})}(\mathbf{x})\mathcal{L}_{\phi(\mathbf{1}_{n_1}),g(\mathbf{1}_{n_1})}(\mathbf{y})$$

$$\text{where } p'(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{a}\in\mathbb{F}_2^{n_1}\setminus\{\mathbf{1}_{n_1},\mathbf{1}_{n_1-2}00\}} \mathcal{P}_{\mathbf{a}}(\mathbf{x})\left(\mathcal{L}_{(\phi(\mathbf{a}),g(\mathbf{a}))}(\mathbf{y})\right).$$

Thus in this case $\mathrm{pdeg}(p') = n - 1$. Now, expanding $p(\mathbf{x}, \mathbf{y})$ we get

$$\begin{aligned}
&p(\mathbf{x}, \mathbf{y}) = \\
&p'(\mathbf{x}, \mathbf{y}) + \left(\prod_{i=1}^{n_1-2} \mathbf{x}_i\right)(1-x_{n_1-1})(1-x_{n_1})\mathcal{L}_{(\mathbf{1}_{n_2},1)}(\mathbf{y}) + \left(\prod_{i=1}^{n_1} x_i\right)\mathcal{L}_{(\mathbf{1}_{n_2},0)}(\mathbf{y}) \\
&= p'(\mathbf{x}, \mathbf{y}) + \mathbf{X^{n_1-2}}(1 - x_{n_1-1} - x_{n_1} + x_{n_1-1}x_{n_1})\mathcal{L}_{(\mathbf{1}_{n_2},1)}(\mathbf{y}) + \mathbf{X^{n_1}}\mathcal{L}_{(\mathbf{1}_{n_2},0)}(\mathbf{y}) \\
&= p'(\mathbf{x}, \mathbf{y}) + \left(\mathbf{X_1^{n_1-2}} - \mathbf{X_1^{n_1-2}}x_{n_1-1} - \mathbf{X_1^{n_1-2}}x_{n_1}\right)\mathcal{L}_{(\mathbf{1}_{n_2},1)}(\mathbf{y}) \\
&+ \mathbf{X^{n_1}}\left(\mathcal{L}_{(\mathbf{1}_{n_2},0)}(\mathbf{y}) + \mathcal{L}_{(\mathbf{1}_{n_2},1)}(\mathbf{y})\right) \\
&= p'(\mathbf{x}, \mathbf{y}) + \left(\mathbf{X_1^{n_1-2}} - \mathbf{X_1^{n_1-2}}x_{n_1-1} - \mathbf{X_1^{n_1-2}}x_{n_1}\right)\mathcal{L}_{(\mathbf{1}_{n_2},1)}(\mathbf{y}) + \mathbf{X^{n_1}}.
\end{aligned}$$

In this case, the polynomial degree of the terms are:

- $p'(\mathbf{x}, \mathbf{y})$ has a degree of $n - 1$;
- $\left(\mathbf{X_1^{n_1-2}} - \mathbf{X_1^{n_1-2}}x_{n_1-1} - \mathbf{X_1^{n_1-2}}x_{n_1}\right)\mathcal{L}_{(\mathbf{1}_{n_2},1)}(\mathbf{y})$ has a degree of $n - 1$.
- the degree of X^{n_1} is n_1.

Thus, the polynomial degree of $p(\mathbf{x}, \mathbf{y})$ is $n - 1$ and the proof is complete. □

In this case, any function in $\mathbb{MM}_n$ with the added constraint that $\phi(\mathbf{1}_{n_1-1}00) = \mathbf{1}_{n_2}$ and $g(\mathbf{x}) = \left(\prod_{i=1}^{n_1-2} x_i\right)(1 - x_{n_1-1})(1-x_n)$ would have this property of $s(f) = n$ and $\mathrm{pdeg}(f) = n - 1$. Except for $\mathbf{1}_{n_2}00$ and $\mathbf{1}_{n_2}$, all the points can have any linear function of weight less than n_2, with the only restriction being that n points in $\mathbf{x}$ should have odd weighted linear functions attached to them. This gives us a loose lower bound on the number of functions that we can recover with this construction technique with the given property. We can in fact construct $2^{\mathrm{EXP}(n)}$ functions with such property, which we state next.

Corollary 5.8. *For any n, there are at least $\Omega\left(2^{2^{\frac{n}{2}}}\right)$ MM type functions with $s(f) = n$ and $\mathrm{pdeg}(f) = n - 1$.*

Proof. By definition we have $\lceil \frac{n}{2} \rceil \le n_2 \lceil \frac{n}{2} \rceil + 1$. For simplicity, let us assume that n is even. Then each distinct mapping ϕ that satisfies the constraint of Theorem 5.7 will represent an MM type function with $s(f) = n$ and $\mathrm{pdeg}(f) = n - 1$. We have already defined ϕ at the points $\mathbf{1}_{n_1-2}00$ and $\mathbf{1}_{n_1}$. Let us consider the mapping where $\phi(\mathbf{1}^i_{n_1}) = 0^i_{n_2}$. That is, the function attached corresponding to the ith critical point $(1 \le i \le n)$ is y_i and since $n_2 > n_1$, this is feasible. Therefore, corresponding to the other $2^{\frac{n}{2}} - 2 - \frac{n}{2}$ points in $\mathbf{x}$ we can put any of the $2^{2^{\frac{n}{2}}}$ linear functions defined on $\mathbf{y}$, and each such function would have the desired property.

Table 2
A Boolean function f with $s(f) = 7$ and $\mathrm{pdeg}(f) = 6$.

x	The linear function on **y**
000	$y_2 \oplus y_3$
001	y_1
010	y_2
011	$y_1 \oplus y_2$
100	$y_1 \oplus y_2 \oplus y_3 \oplus y_4 \oplus 1$
101	$y_1 \oplus y_3$
110	y_3
111	$y_1 \oplus y_2 \oplus y_3 \oplus y_4$

Thus, the number of such functions will be $\left(2^{\frac{n}{2}} - 2 - \frac{n}{2}\right)^{2^{\frac{n}{2}}}$, which we denote by $C(n, n-1)$. Then for $n > 6$ we have $C(n, n-1) > \left(2^{\frac{n}{2}-1}\right)^{2^{\frac{n}{2}}} = \Omega\left(2^{2^{\frac{n}{2}}}\right)$. □

We give now an example of a function with $s(f) = 1$ and $\mathrm{pdeg}(f) = n - 1$. We let $n = 7$ and $n_1 = 3, n_2 = 4$ and show the structure of the Boolean function. Observe that for the inputs 001, 010 and 110 in **x** the corresponding linear function contains only a single variable and the linear function corresponding to 111 contains 4 variables, implying that f is in the class $\mathbb{MM}_7$. Additionally the linear function at 100 is the complement of the linear function at 111, which satisfies the conditions of Theorem 5.7 (see Table 2).

We now further develop this construction method to get functions with $s(f) = n$ and $\mathrm{pdeg}(f) = n - \Theta(\log n)$. Finally, we check the nature of super-linear separation that can be achieved for this class using recursive amplification.

Next we generalize our construction method, by building on the construction in Theorem 5.7. The difficulty in designing Boolean functions with any desired polynomial degree is that modifying the polynomial directly may lead to a situation where the modified polynomial does not represent a Boolean function at all. Moreover, even valid modifications may affect the sensitivity of the function. Even if these two issues are somehow dealt with, recovering the Boolean function structure from the modified polynomial can be a tedious task. Against this backdrop we first observe some valid real polynomial modifications to MM type functions and their effect on the Boolean function structure.

5.3. Interpreting real polynomial terms via the MM construction

We saw in Proposition 5.3 that the real polynomial corresponding to any MM type function can be expressed as $p(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{a}\in\mathbb{F}_2^{n_1}} \mathcal{P}_{\mathbf{a}}(\mathbf{x})\left(\mathcal{L}_{(\phi(\mathbf{a}),g(\mathbf{a}))}(\mathbf{y})\right)$. We know that $\mathcal{P}_{\mathbf{a}}(\mathbf{x}) = \left(\prod_{a_i=1} x_i\right)\left(\prod_{a_j=0}(1-x_j)\right)$. If $\mathbf{z} = (z_1, z_2, \ldots, z_k) \in \mathbb{F}_2^k$, it is easy to see $\sum_{\mathbf{a}\in\mathbb{F}_2^k}\left(\left(\prod_{a_i=0} z_i\right)\left(\prod_{a_j=0}(1-z_j)\right)\right) = 1$. Corresponding to a Boolean function defined on n variables $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, we define three non-empty mutually disjoint sets S_1, S_2 and S_3 such that $S_1 \cup S_2 \cup S_3 = [n]$ with $|S_i| = s_i$. Let the variables indexed by elements in S_i be denoted as x_{i_j} and $\mathbf{z} \in \mathbb{F}_2^{s_3}$ be represented as $(z_1, z_2, \ldots, z_{s_3})$. Then the real polynomial $\left(\prod_{i\in S_1} x_i\right)\left(\prod_{j\in S_2}(1-x_j)\right)$ can be represented as

$$\begin{aligned}
&\left(\prod_{i\in S_1} x_i\right)\left(\prod_{j\in S_2}(1-x_j)\right) \qquad (5.3)\\
&= \left(\prod_{i\in S_1} x_i\right)\left(\prod_{j\in S_2}(1-x_j)\right)\left(\sum_{\mathbf{z}\in\mathbb{F}_2^{s_3}}\left(\left(\prod_{z_i=0} x_{3_i}\right)\left(\prod_{z_j=0}(1-x_{3_j})\right)\right)\right)\\
&= \sum_{\mathbf{z}\in\mathbb{F}_2^{s_3}}\left(\left(\prod_{i\in S_1} x_i\right)\left(\prod_{j\in S_2}(1-x_j)\right)\left(\prod_{z_i=0} x_{3_i}\right)\left(\prod_{z_j=0}(1-x_{3_j})\right)\right).
\end{aligned}$$

This implies that corresponding to an MM type function defined on $n = n_1 + n_2$ variables, the polynomial $\mathcal{L}_{(\mathbf{b},c)}(\mathbf{y})\left(\prod_{i=1}^{k_1} x_i\right)\left(\prod_{j=k_1+1}^{k_2}(1-x_j)\right)$ can be interpreted as

$$\sum_{\mathbf{a}\in\mathbb{F}_2^{n_1-k_1-k_2}}\left(\left(\prod_{i\in_1}^{k_1} x_i\right)\left(\prod_{j=k_1+1}^{k_2}(1-x_j)\right)\left(\prod_{a_i=1} x_{k_1+k_2+i}\right)\left(\prod_{a_j=0}(1-x_{k_1+k_2+j})\right)\right)\mathcal{L}_{(\mathbf{b},c)}(\mathbf{y}).$$

We represent it as $\sum_{\mathbf{a}\in\mathbb{F}_2^{n_1-k_1-k_2}} \mathcal{P}_{(\mathbf{1}_{k_1}\mathbf{0}_{k_2}\mathbf{a})}(\mathbf{x})\mathcal{L}_{(\mathbf{b},c)}(\mathbf{y})$. Thus, we get

$$\left((\prod_{i\in_1}^{k_1} x_i)(\prod_{j=k_1+1}^{k_2}(1-x_j))\right)\mathcal{L}_{(\mathbf{b},c)}(\mathbf{y}) = \sum_{\mathbf{t}\in\mathbb{F}_2^{n_1-k_1-k_2}} \left(\mathcal{P}_{(\mathbf{1}_{k_1}\mathbf{0}_{k_2}\mathbf{t})}(\mathbf{x})\mathcal{L}_{(\mathbf{b},c)}(\mathbf{y})\right). \tag{5.4}$$

These considerations imply the following result.

Proposition 5.9. *Let f be an MM function $f(\mathbf{x},\mathbf{y}) = \mathbf{x}\cdot\phi(\mathbf{y}) + g(\mathbf{x})$ such that if $\mathbf{x} = \mathbf{1}_{k_1}\mathbf{0}_{k_2}\mathbf{t}$ then $\phi(\mathbf{x}) = 0$ and $g(\mathbf{1}_{k_1}\mathbf{0}_{k_2}\mathbf{t}) = 0,\ \forall \mathbf{t}\in\mathbb{F}_2^{n_1-k_1-k_2}$. Then the polynomial corresponding to the Boolean function f can be written as*

$$p(\mathbf{x},\mathbf{y}) = \sum_{\mathbf{a}\in\mathbb{F}_2^{n_1},\mathbf{a}\neq\mathbf{1}_{k_1}\mathbf{0}_{k_2}\mathbf{t}} \mathcal{P}_{\mathbf{a}}(\mathbf{x})\left(\mathcal{L}_{(\phi(\mathbf{a}),g(\mathbf{a}))}(\mathbf{y})\right),$$

and the polynomial $p'(\mathbf{x},\mathbf{y}) = p(\mathbf{x},\mathbf{y}) + \left((\prod_{i\in_1}^{k_1} x_i)(\prod_{j=k_1+1}^{k_2}(1-x_j))\right)\mathcal{L}_{(\mathbf{b},c)}(\mathbf{y})$ can be written as $p'(\mathbf{x},\mathbf{y}) = \sum_{\mathbf{a}\in\mathbb{F}_2^{n_1}} \mathcal{P}_{\mathbf{a}}(\mathbf{x})\,\mathcal{L}_{(\phi'(\mathbf{a}),g'(\mathbf{a}))}(\mathbf{y})$, where

$$\phi'(\mathbf{x}) = \begin{cases}\phi(\mathbf{x}) & \text{if } \mathbf{x}\neq\mathbf{1}_{k_1}\mathbf{0}_{k_2}\mathbf{t} \text{ for some } \mathbf{t}\\ b & \text{otherwise,}\end{cases}$$

$$g'(\mathbf{x}) = \begin{cases}g(\mathbf{x}) & \text{if } \mathbf{x}\neq\mathbf{1}_{k_1}\mathbf{0}_{k_2}\mathbf{t} \text{ for some } \mathbf{t}\\ c & \text{otherwise}\end{cases}$$

and this represents another MM type function f' which differs from f only in the points $\mathbf{1}_{k_1}\mathbf{0}_{k_2}\mathbf{t}$.

Using this result we can attempt to obtain an MM type Boolean function with a pre-decided real polynomial structure. We start with the real polynomial of a particular MM type function, and then modifying its corresponding polynomial by adding $\left(\prod_{i\in_1}^{k_1} x_i\ \prod_{j=k_1+1}^{k_2}(1-x_j)\right)\mathcal{L}_{(\mathbf{b},c)}(\mathbf{y})$, keeping in mind the respective necessary constraints we have discussed in terms of ϕ and g. This gives us another function f' whose structure and its properties can be recovered from f.

5.4. Logarithmic separation

We finally design a method of constructing an MM type function in $\mathbb{MM}_n$ that has polynomial degree of $n-\Theta(\log n)$. Before proceeding to the proof, we derive some more results related to the structure of different polynomials and their interaction upon addition.

In the last subsection we observed how given an MM type function f, where ϕ and g are defined to be zero in many points, say all points of the form $\mathbf{1}_{k_1}\mathbf{0}_{k_2}\mathbf{t}\in\mathbb{F}_2^{n_1}$ then adding a polynomial term of the form $\left(\prod_{i=1}^{k_1} x_i \prod_{j=k_1+1}^{k_2}(1-x_j)\right)\mathcal{L}_{\mathbf{b},c}(\mathbf{y})$ transforms it to another MM type function f' whose map ϕ' and sub-function on g' differs from f only for the points $\mathbf{1}_{k_1}\mathbf{00}_{k_2}\mathbf{t}\in\mathbb{F}_2^{n_1}$. With these observations in mind we analyze a function $f_1\in\mathbb{MM}_n$ such that its polynomial satisfies the constraint of Theorem 5.7 and then further add some constraints so as to be able to make modifications as discussed in Proposition 5.9. We define the corresponding map ϕ_1 and sub-function g_1 in the lines of Theorem 5.7:

$$\begin{aligned}\phi_1(\mathbf{x}) &= \begin{cases}\mathbf{1}_{n_2} & \text{if } \mathbf{x}=\mathbf{1}_{n_1} \text{ or } \mathbf{x}=\mathbf{1}_{n_1-2}00\\ \mathbf{0}_{n_2}^i & \text{if } \mathbf{x}=\mathbf{0}_{n_1}^i\\ 0 & \text{otherwise,}\end{cases}\\ g_1(\mathbf{x}) &= \left(\prod_{i=1}^{n_1-2} x_i\right)(1-x_{n_1-1})(1-x_{n_1}).\end{aligned} \tag{5.5}$$

This definition differs from the function in Theorem 5.7 in that the map ϕ_1 is defined as 0 in all but n_1+2 positions. However it satisfies all the constraints of Theorem 5.7. This means that the real polynomial corresponding to f_0 can be expressed as

$$\begin{aligned}p_1(\mathbf{x},\mathbf{y}) &= p_1'(\mathbf{x},\mathbf{y}) + \left(\mathbf{X_1^{n_1-2}} - \mathbf{X_1^{n_1-2}}x_{n_1-1} - \mathbf{X_1^{n_1-2}}x_{n_1}\right)\mathcal{L}_{(\mathbf{1}_{n_2},1)}(\mathbf{y}) + \mathbf{X^{n_1}}\\ &= p_1'(\mathbf{x},\mathbf{y}) + \left(\mathbf{X_1^{n_1-2}} - \mathbf{X_1^{n_1-2}}x_{n_1-1} - \mathbf{X_1^{n_1-2}}x_{n_1}\right)\frac{1+\mathbf{L_{n_2}}}{2} + \mathbf{X^{n_1}},\end{aligned}$$

where $p_1'(\mathbf{x},\mathbf{y})$ defined in the same manner as p' in Theorem 5.7. Consider the term $-\mathbf{X_1^{n_1-2}}x_{n_1-1}\frac{1+\mathbf{L_{n_2}}}{2}$ which has a degree of $n-1$. If we add the polynomial $\mathbf{X_1^{n_1-4}}(1-x_{n_1-3})(1-x_{n_1-2})x_{n_1-1}\frac{1-\mathbf{L_{n_2}}}{2}$ to this term we get

$$\begin{aligned}&\mathbf{X_1^{n_1-4}}(1-x_{n_1-3}-x_{n_1-2}+x_{n_1-3}x_{n_1-2})x_{n_1-1}\frac{1-\mathbf{L_{n_2}}}{2} - \mathbf{X_1^{n_1-2}}x_{n_1-1}\frac{1+\mathbf{L_{n_2}}}{2}\\ &= \mathbf{X_1^{n_1-4}}x_{n_1-1} - \mathbf{X_1^{n_1-4}}x_{n_1-3}x_{n_1-1}\frac{1-\mathbf{L_{n_2}}}{2} - \mathbf{X_1^{n_1-4}}x_{n_1-2}x_{n_1-1}\frac{1-\mathbf{L_{n_2}}}{2}\end{aligned}$$

$$+\mathbf{X_1^{n_1-2}}x_{n_1-1}\frac{1-\mathbf{L_{n_2}}}{2}-\mathbf{X_1^{n_1-2}}x_{n_1-1}\frac{1+\mathbf{L_{n_2}}}{2}$$
$$=\left(\mathbf{X_1^{n_1-4}}x_{n_1-1}-\mathbf{X_1^{n_1-4}}x_{n_1-3}x_{n_1-1}-\mathbf{X_1^{n_1-4}}x_{n_1-2}x_{n_1-1}\right)\frac{1-\mathbf{L_{n_2}}}{2}+\mathbf{X_1^{n_1-2}}x_{n_1-1}.$$

This polynomial has degree $n-2$, which reduces the degree of the polynomial by 1 and creates 2 terms of the same type of degree $n-2$ and one of degree $n-3$. Similarly, if we could modify the other degree $n-1$ polynomial in the same way it would bring down the overall degree of the real polynomial of the function to $n-2$. This modification is central to our construction method and we now generalize it, starting with defining two generic polynomial structures:

1 $\hat{p}(i,s)=(\prod_{j=1}^{i}x_j)(1-x_{i+1})(1-x_{i+2})(\prod_{j=s_i}x_j)$, $s\subset[n]\setminus[i+2]$,
2 $\tilde{p}(i,u)=(\prod_{j=1}^{i}x_j)(\prod_{j=u_i}x_j)$, $u\subset[n]\setminus[i]$.

From these definitions we get the following two relations.

$$\mathsf{pdeg}(\tilde{p}(i,u))=\mathsf{pdeg}(\hat{p}(i-2,u))=i+|u|, \tag{5.6}$$

$$\tilde{p}(i,u)=\tilde{p}(i-k,u^k)\text{ where }u^k=u\cup\{i,i-1,\ldots,i-k+1\}. \tag{5.7}$$

Then corresponding to any $\tilde{p}(i,u)\frac{1+\mathbf{L_{n_2}}}{2}$, we have

$$\begin{aligned}&\tilde{p}(i,u)\frac{1+\mathbf{L_{n_2}}}{2}+\hat{p}(i-2,u)\frac{1-\mathbf{L_{n_2}}}{2}\\ =&\tilde{p}(i-2,u)\frac{1+\mathbf{L_{n_2}}}{2}-\tilde{p}(i-2,u)x_{i-1}\frac{1+\mathbf{L_{n_2}}}{2}-\tilde{p}(i-2,u)x_i\frac{1+\mathbf{L_{n_2}}}{2}+\tilde{p}(i,u)\\ =&\tilde{p}(i-2,u)\frac{1+\mathbf{L_{n_2}}}{2}-\tilde{p}(i-2,u_1)\frac{1+\mathbf{L_{n_2}}}{2}-\tilde{p}(i-2,u_2)\frac{1+\mathbf{L_{n_2}}}{2}+\tilde{p}(i,u),\end{aligned} \tag{5.8}$$

where $|u_1|=|u_2|=|u+1|$. Then this addition of the polynomial $\hat{p}(i-2,u)\frac{1-\mathbf{L_{n_2}}}{2}$ reduces the degree of the polynomial by 1 and forms two similar terms of degree 1 less, one term of degree 2 less, and one term that is only defined on $\mathbf{x}$ and therefore has degree less than n_1. Next we note another property of $\hat{p}$ before proceeding with the final proof.

Lemma 5.10. *Corresponding to the polynomial p_1 of an MM function f_1 where ϕ_1 is nonzero in only the points $\mathbf{1}_{n_1}$, $\mathbf{1}_{n_1-2}\mathbf{0}^i_{n_1}$ such that $f_1\in\mathbb{MM}_n$ ($s(f,\mathbf{1}_n)=n$) then adding of any arbitrary number of polynomial terms $\hat{p}(i,u)(\frac{1\pm\mathbf{L_{n_2}}}{2})$ to p_1 transforms it to another MM Boolean function in $\mathbb{MM}_n$, provided that for any two added polynomial terms $\hat{p}(i_1,u_1)$ and $\hat{p}(i_2,u_2)$ we have $|i_1-i_2|\geq 2$ and $i<n_1-2$, $\forall\,\hat{p}(i,u)$.*

Proof. Consider any two $\hat{p}(i_1,u_1)$ and $\hat{p}(i_2,u_2)$ with $i_1\leq i_2-2$. Then, from Eq. (5.4) we have that adding $\hat{p}(i_1,u_1)(\frac{1\pm\mathbf{L_{n_2}}}{2})$ and $\hat{p}(i_2,u_2)(\frac{1\pm\mathbf{L_{n_2}}}{2})$ to a polynomial corresponding to an MM function only works (the resultant function is still in $\mathbb{MM}_n$) if $\phi(\mathbf{x})=0$ for all $\mathbf{x}$ such that $\hat{p}(i_1,u_1)(\mathbf{x})=1$ or $\hat{p}(i_2,u_2)(\mathbf{x})=1$ and moreover $\hat{p}(i_1,u_1)$ and $\hat{p}(i_1,u_1)$ should not both return 1 for any input $\mathbf{x}$.

The first condition is true as any polynomial of the type $\hat{p}(i,u)$ can only return 1 if both x_{i+1} and x_{i_2} are set to 0. This cannot happen for any critical point as their weight is minimum n_1-1. And since we have defined $i<n_1-2$ none of these polynomials can return 1 for the point $\mathbf{1}_{n_1-2}00$.

Also for any two polynomials $p_1=\hat{p}(i_1,u_1)$ and $p_2=\hat{p}(i_2,u_2)$ with $i_1\leq i_2-2$ they cannot both be 1 for any input $\mathbf{a}\in\mathbb{F}_2^{n_1}$ for the simple reason that for $p_1(\mathbf{x})$ to be 1, x_{i_1} and x_{i_1+1} should be set as 1 and for $p_2(\mathbf{x})$ to be 1 both x_{i_1} and x_{i_2} should be set to 0, by definition. This completes the proof. □

Finally, we use the results and equations we have defined so far to develop a stepwise construction technique to obtain a function with $s(f)=n$ and $\mathsf{pdeg}(f)=n-\Theta(\log n)$.

Theorem 5.11. *There exists a Boolean function $f\in\mathbb{MM}_n$ with $\mathsf{pdeg}(f)=n-\Theta(\log n)$.*

Proof. We define the function f_1 as in Eq. (5.5) and the corresponding polynomial is $p_1=p_1'(\mathbf{x},\mathbf{y})+\left(\mathbf{X_1^{n_1-2}}-\mathbf{X_1^{n_1-2}}x_{n_1-1}-\mathbf{X_1^{n_1-2}}x_{n_1}\right)(\frac{1+\mathbf{L_{n_2}}}{2})+\mathbf{X_1^{n_1}}$ where $\deg(p')=n_1+1$.

We start with f_1 and recursively reduce the polynomial degree by adding polynomials of the form $\hat{p}(i,u)$ where i starts with n_1-3 and decreases by 2 with every new polynomial that we add. The terms $\mathbf{X_1^{n_1-2}}(\frac{1+\mathbf{L_{n_2}}}{2})$, $\mathbf{X_1^{n_1-2}}x_{n_1}(\frac{1+\mathbf{L_{n_2}}}{2})$ and $\mathbf{X_1^{n_1-2}}x_{n_1-1}(\frac{1+\mathbf{L_{n_2}}}{2})$ can all be expressed polynomials $\tilde{p}(i,u_i)(\frac{1\pm\mathbf{L_{n_2}}}{2})$ for some u_i. Here $i=n_1-2$. We also know from Eq. (5.6) that any such polynomial can also be represented as some $\tilde{p}(i-k,u_i^k)(\frac{1\pm\mathbf{L_{n_2}}}{2})$.

Then corresponding to the term $(-1)^c\tilde{p}(i-k,u_i^k)(\frac{1+(-1)^d\mathbf{L_{n_2}}}{2})$ we add the polynomial $\hat{p}(i-k-2,u_i^k)(\frac{1+(-1)^{c+d+1}\mathbf{L_{n_2}}}{2})$ and this results in at most 3 new polynomial terms of the form $\tilde{p}(i-k-1,u_i^{k'})$ of degree at least one less (specifically one polynomial of degree 2 less and two polynomials of degree 1 less). For each new polynomial we add, if the last polynomial that we added was $\hat{p}(i,u)(\frac{1\pm\mathbf{L_{n_2}}}{2})$, then the next polynomial is $\hat{p}(i-2,u')(\frac{1\mp\mathbf{L_{n_2}}}{2})$, where u and u' depend on the lower degree polynomials that form. Therefore for any defined n_1 we can at most add $\lfloor\frac{n_1-2}{2}\rfloor$ polynomials. This polynomial term addition is valid and the modified function is still in $\mathbb{MM}_n$ as we have shown in Lemma 5.10.

At the first step we have 3 such polynomial terms to which we add a $\hat{p}$ polynomial. Then, at the ith step, if we have 3^i polynomials of the form $\tilde{p}$ of degree $n-i$. Corresponding to each such, we add polynomials $\hat{p}(k,u)$ where $n_1-3^i\geq k\geq n_1-3^{i+1}$ for each polynomial $\tilde{p}(k+2,u)$, and this would form 3^{i+1} polynomials of degree at most $n-(i+1)$. This step can at most be continued for z iterations, where $\sum_{i=1}^{z}2\times 3^z\leq n_1-1$, that is, $z\leq\log_3(n_1-1)$. If we fix $n_1=\frac{n}{2}$ this gives us a function in $\mathbb{MM}_n$ with polynomial degree $n-\log_3(\frac{n}{2})=n-\Theta(\log n)$. □

It is easy to see that when we have not used some k bits of $\mathbf{x}$ as the zero bits in $\hat{p}$, our modifications do not affect the map ϕ in the point with at least any two of those k-bits being zero. This gives us the following corollary.

Corollary 5.12. *There are $\Omega\left(2^{2^k}\right)$ functions in $\mathbb{MM}_n$ with polynomial degree $n-\log_3(\frac{n}{2})+\log_3(k)$ that we can recover using the construction method of Theorem 5.11.*

Furthermore, any such function can then be recursively amplified to obtain a class of functions with super-linear separation between $s(f)$ and $\mathsf{pdeg}(f)$. On an interesting note, the recursive amplification generally works on a single function to obtain f^n. In this case, we can use different functions f_i^d to get f^{d+1}, because of the fact that we are able to recover large number of functions for a given polynomial degree, albeit within logarithmic distance and all of these functions will have full sensitivity in the all 1 input point. We thus have the following theorem.

Theorem 5.13. *For any k, we can construct several functions with $s(f)=k^n$ and $\mathsf{pdeg}(f)=d^n$ where $k-\Theta(\log k)\leq d\leq k-1$.*

Proof. We defined the class $\mathbb{MM}_n$ such that $s(f,\mathbf{1}_n)=n$ so that $f(\mathbf{1}_n)=0$ and $f(\mathbf{1}_n^i)=1$. Then the function $\overline{f}(\mathbf{x})=f(\mathbf{x})\oplus 1$ also has full sensitivity at the point $\mathbf{1}_n$ and $\overline{f}(\mathbf{1}_n)=1$ and $\overline{f}(\mathbf{1}_n^i)=0$. The corresponding polynomial $\overline{p}=\frac{1-p}{2}$ and the degree remain unchanged.

For convenience, we write $\overline{f}$ in lieu of f below. It is easy to see by induction if $s(f^d,\mathbf{1}_{n^d})=n$ with $f(\mathbf{1}_{n^d})=1$ then the same follows for f^{d+1}. This is because

$$f^{d+1}=\left(f(f^d(x_1,\ldots,x_{n^d}),f^d(x_{n^d+1},\ldots,x_{2n^d}),\ldots,f^d(x_{(n-1)n^d+1},\ldots,x_{n^{d+1}})\right).$$

Then $f^{d+1}(\mathbf{1}_{n^{d+1}})=1$ and for any point $f^{d+1}(\mathbf{1}^i_{n^{d+1}})$ any one of the functions f^d will return 0 and the rest will return 1. Thus the input to the function f is $\mathbf{1}_n^i$ and thus f will output 0. It is easy to see that if $\mathsf{pdeg}(f^d)=k^d$ then $\mathsf{pdeg}(f^{d+1})=k^{d+1}$ and this does not require further elaboration.

In fact, for any $s(f)=k$ and $\mathsf{pdeg}(f)=n-\log_3(\frac{k}{2})+l$, we can choose any of the 2^{2^l} functions we have and that will lead to constructions of this kind. Therefore the total number of such functions will be $(k^n)^{2^{2^l}}=\Omega(k^{n2^k})$ (as $l=\mathcal{O}(\log k)$). □

The fully sensitive function with $s(f)=n-\Theta(\log n)$ that is $\left(\lfloor\frac{n}{2}\rfloor-2\right)$-resilient:

Note that the resulting function in Theorem 5.11 is still a Maiorana McFarland Type function $f\in\mathbb{MM}_n$ where the minimum weight of ϕ can be as high as $\lfloor\frac{n}{2}\rfloor-1$ as the condition on $\mathbb{MM}_n$ is that $wt(\phi_x)\in\{n_2,z\}$ where $z\equiv 0 \mod n_2+1$. Now if we fix $n_2=\frac{n}{2}$ then we can have $\min_{\mathbf{x}\in\mathbb{F}_2^{n_1}}wt(\phi(\mathbf{x}))=\lfloor\frac{n}{2}\rfloor-1$. Now this directly implies the function is $\left(\lfloor\frac{n}{2}\rfloor-2\right)$-resilient as it belongs to the Maiorana McFarland class.

Finally we extend our constructions and results for super-constant orders of sensitivity.

5.5. Extending to super-constant higher order sensitivity via the MM construction

We have so far observed the situation where we have defined a function on n variables in the MM class as $f(\mathbf{x},\mathbf{y})=(\phi(\mathbf{x})\cdot\mathbf{y})\oplus g(\mathbf{x})$ where $\mathbf{x}\in\mathbb{F}_2^{n_1}$ and $\mathbf{y}\in\mathbb{F}_2^{n_2}$ with $n_1+n_2=n$. The simplest interpretation is choosing a linear function in $\mathbf{y}$ (or its complement depending on g) corresponding to each point $\mathbf{x}\in\mathbb{F}_2^{n_1}$. We can extend this to nonlinear functions in $\mathbf{y}$ being fixed with respect to the points in $\mathbf{y}$.

In order to define this, we first define 2^{n_1} activator functions in $\mathbb{F}_2^{n_1}$, defined as $ac_{\mathbf{a}}(\mathbf{x})=\prod_{i=1}^{n_1}(\mathbf{x}_i\oplus\bar{\mathbf{a}})$, $\mathbf{a}\in\mathbb{F}_2^{n_1}$. Thus, we have $ac(\mathbf{x})_{\mathbf{a}}=1$ if and only if $\mathbf{x}=\mathbf{a}$. Then, we can define any MM function with nonlinear functions in $\mathbf{y}$ in the algebraic normal form as $f(\mathbf{x},\mathbf{y})=\bigoplus_{\mathbf{a}\in\mathbb{F}_2^{n_1}}(ac_{\mathbf{a}}(\mathbf{x})\cdot g_{\mathbf{a}}(\mathbf{y}))$, where $g_{\mathbf{a}}:\mathbb{F}_2^{n_2}\to\mathbb{F}_2$, $\forall\mathbf{a}$. Then the real polynomial corresponding to the function $f(\mathbf{x},\mathbf{y})$ can be written as $p(\mathbf{x},\mathbf{y})=\sum_{\mathbf{a}\in\mathbb{F}_2^{n_1}}\mathcal{P}_{\mathbf{a}}(\mathbf{x})\hat{g}_{\mathbf{a}}(\mathbf{y})$ where $\mathcal{P}_{\mathbf{a}}$ is as defined in Eq. (5.1) and $\hat{g}_{\mathbf{a}}:\mathbb{F}_2^{n_2}\to\mathbb{R}$ is the real polynomial corresponding to the function $g_{\mathbf{a}}$. Now let us discuss some sufficient conditions to obtain kth order sensitivity by choosing the $g_{\mathbf{a}}$ functions.

5.6. Obtaining kth order sensitivity

We start by defining a function in $\mathbb{F}_2^{n_2}$ that is kth order sensitive itself. We define this function as $\mathsf{sym}_{n_2}^k : \mathbb{F}_2^{n_2} \to \mathbb{F}_2, k \leq n$. The algebraic normal form of the function contains all degree $i, 1 \leq i \leq k$ monomials. For an example $\mathsf{sym}_4^2(\mathbf{y}) = y_1 \oplus y_2 \oplus y_3 \oplus y_4 \oplus y_1y_2 \oplus y_1y_3 \oplus y_1y_4 \oplus y_2y_3 \oplus y_2y_4 \oplus y_3y_4$. Next we observe the sensitivity order of this function.

Lemma 5.14. *The function sym_m^k is a function defined on m variables. kth order sensitive around the all zero input point $\mathbf{0}_m$.*

Proof. We have $\mathsf{sym}_m^k(\mathbf{0}_m) = 0$ by definition. Now it suffices to see for any input point $\mathbf{b} \in \mathbb{F}_2^m$ that can be obtained by flipping $i \leq k$ bits of $\mathbf{0}_m$ returns true for an odd number of monomials of sym_m^k, in which case we will have $\mathsf{sym}_m^k(\mathbf{b}) = 1$. This can be verified as follows.

Any input point obtained by flipping i points in $\mathbf{0}_m$ has exactly i variables set as one and the rest as zero. These i values will return 1 for i monomials of degree 1, $\binom{i}{2}$ monomials of degree 2, and so on. Thus the total number of monomials for which it will return 1 is $\sum_{j=1}^{i}\binom{i}{j} = 2^i - 1$, which is odd for all values of $i > 0$. This completes the proof. □

Next we define an MM type function $\mathbb{MM}_n^k$ with nonlinear functions in $\mathbf{y}$, which is kth order sensitive.

Lemma 5.15. *Any function $f : \mathbb{F}_2^{n_1+n_2} \to \mathbb{F}_2$ with the algebraic normal form $f(\mathbf{x}, \mathbf{y}) = \bigoplus_{\mathbf{a}\in\mathbb{F}_2^{n_1}} (c_{\mathbf{a}}(\mathbf{x}) \cdot g_{\mathbf{a}}(\mathbf{y}))$, where $g_{\mathbf{1}_{n_1}} = \mathsf{sym}_{n_2}^k$ and $g_{\mathbf{a}} = 1$ for all $\mathbf{a} \in \mathbb{F}_2^{n_1} : n_1 > wt(\mathbf{a}) \geq n_1 - k$ is kth order sensitive.*

Proof. We show that the function is kth order sensitive in the point $(\mathbf{a}, \mathbf{b}) = (\mathbf{1}_{n_1}, \mathbf{0}_{n_2})$. We have $f(\mathbf{1}_{n_1}, \mathbf{1}_{n_2}) = \mathsf{sym}_{n_2}^k(\mathbf{0}_{n_2}) = 0$. Now, we consider any input that can be obtained by flipping at most k bits of $(\mathbf{1}_{n_1}, \mathbf{0}_{n_2})$. There can be two cases.

All bits are flipped in y: In this case the input point is of the form $(\mathbf{x}', \mathbf{y}') = \left(\mathbf{1}_{n_1}, \left[\mathbf{0}_{n_2}\right]_k\right)$ for which the output is of the form $f(\mathbf{x}', \mathbf{y}') = \mathsf{sym}_{n_2}^k\left(\left[\mathbf{0}_{n_2}\right]_k\right)$ which is 1 from Lemma 5.14.

At least one bit is flipped in x: Any such point is of the form $(\mathbf{x}', \mathbf{y}') = \left(\left[\mathbf{1}_{n_1}\right]_k, \left[\mathbf{0}_{n_2}\right]_{k-1}\right)$. Then the output of the function is of the form $f(\mathbf{x}', \mathbf{y}') = g_{\left[\mathbf{1}_{n_1}\right]_k}(\mathbf{y}) = 1$.

This shows that the output of the function is flipped if any $1 \leq i \leq k$ of the input bits are flipped at the point $(\mathbf{1}_{n_1}, \mathbf{0}_{n_2})$. □

From here on we fix $n_1 = \lceil \frac{n}{2} \rceil$ and $n_2 = \lfloor \frac{n}{2} \rfloor$. One should note that these choices are quite flexible up to a point, with no significant effect on the results.

5.7. Reducing the polynomial degree

Finally we extend the technique of Section 5.4 to obtain non-constant separation between number of variables and real polynomial degree in functions with super-constant order of sensitivity.

Theorem 5.16. *There exists a kth order sensitive function in $\mathbb{MM}_n^k$ with $n - \frac{\log(\frac{n}{2}-k)-\log k}{k}$ real polynomial degree.*

Proof. We construct the corresponding function analogous to the result for first order sensitive functions defined in Theorem 5.11.

We start with partial description of an MM type function $f : \mathbb{F}_2^{n_1+n_2} \to \mathbb{F}_2$ with nonlinear functions in $\mathbb{F}_2^{n_2}$ attached to the points in $\mathbb{F}_2^{n_1}$. As we have discussed, this function can be written as $f(\mathbf{x}, \mathbf{y}) = \bigoplus_{\mathbf{a}\in\mathbb{F}_2^{n_1}} (ac_{\mathbf{a}}(\mathbf{x}) \cdot g_{\mathbf{a}}(\mathbf{y}))$, where $g_{\mathbf{a}} : \mathbb{F}_2^{n_2} \to \mathbb{F}_2$, $\forall \mathbf{a}$, and the corresponding real polynomial is written as $p(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{a}\in\mathbb{F}_2^{n_1}} \mathcal{P}_{\mathbf{a}}(\mathbf{x})\hat{g}_{\mathbf{a}}(\mathbf{y})$ where $\hat{g}_{\mathbf{a}}(\mathbf{y}) : \mathbb{F}_2^{n_2} \to \mathbb{R}$ is the real polynomial corresponding to $g_{\mathbf{a}}(\mathbf{y})$.

The function f has $g_{\mathbf{a}}$ defined for all points with $wt(\mathbf{a}) \geq n-k$, which consists of $\sum_{i=0}^{k}\binom{k}{i}$ points, with $g_{\mathbf{1}_{n_2}} = \mathsf{sym}_k^n$. Let us denote by $\widehat{sym}_k^n(\mathbf{y})$ the corresponding real polynomial. Then the real polynomial corresponding to $\overline{\mathsf{sym}_k^n}$ is $1 - \widehat{sym}_k^n(\mathbf{y})$.

From this point our construction is analogous to that of Theorem 5.11. In case of first order sensitivity, the polynomial terms we added to reduce the degree was of the form

$$\left(\prod_{i=1}^{j_1}(x_i)(1-x_{j_1+1})(1-x_{j_1+2})\prod_{j\in S}x_j\right)g(\mathbf{y}),$$

where $g(\mathbf{y}) \in \{\frac{1+\mathbf{L}_{\mathbf{n_2}}}{2}, \frac{1-\mathbf{L}_{\mathbf{n_2}}}{2}\}$ and $S \subseteq [n_1] - [j_1 + 2]$ so that high degree polynomial terms cancel out.

In case of kth order sensitive function, any new polynomial term that we add will be of the form

$$T(j,S)=\left(\prod_{i_1=1}^{j}(x_{i_1})\prod_{i_2=j+1}^{j+1+k}\left(1-x_{i_2}\right)\prod_{i_3\in S}x_{i_3}\right)g(\mathbf{y}),$$

where $g(\mathbf{y})\in\{\widehat{sym}_k^n(\mathbf{y}),1-\widehat{sym}_k^n(\mathbf{y})\}$ and $S\subseteq[n_1]-[j+k+1]$.

Let us suppose the total degree of the constructed polynomials at some point is n'. If we add a term of the form $T(j,S)$ that cancels out a polynomial term of degree n', then it creates $\binom{k}{i-1}$ terms of degree $n'-i$ where $0<i<k$. Moreover with a new polynomial term that is added, the value of j reduces by k and as we know $j\geq 0$. Then we have the following lower bound on how much can we reduce the polynomial degree of the function depending on the value of k.

Let us assume the polynomial degree of the function is $n-p$. To reduce the polynomial degree by p we have to cancel $(2^k-1)^p$ terms, and adding such a term will reduce the value of j by k. Let us assume that $n_1=\frac{n}{2}$. Then we have

$$\begin{aligned}
&k\cdot(2^k-1)^p=\frac{n}{2}-k\\
\implies&\log(k)+p\log(2^k-1)=\log\left(\frac{n}{2}-k\right)\\
\implies&\log(k)+pk\geq\log\left(\frac{n}{2}-k\right)\\
\implies&p\geq\frac{\log\left(\frac{n}{2}-k\right)-\log(k)}{k}.
\end{aligned}$$

Thus, even if we have $k=o(\log n)$ we get a non-constant separation between number of variables n and the real polynomial degree $n-p$. □

6. Conclusion

In this paper we study the interplay between cryptology and complexity theory through the immediate connection of resiliency and polynomial degree and use it to study the sensitivity-polynomial degree trade-off. In this direction we extend the notion of sensitivity to higher order for Boolean functions f of n variables (a first order sensitive function is a fully sensitive function). Through the connection between the techniques from two domains, we obtained the following results.

First, using the recursive amplification method we design highly resilient functions with good nonlinearity that has only $\mathcal{O}(n)$ circuit size and $\mathcal{O}(\log n)$ depth. These functions are good candidates for implementation of Boolean functions with large number of variables in lightweight cryptographic primitives.

Next using results in resiliency we study the $s(f)$-$\mathsf{pdeg}(f)$ relation in small number of variables (upto 10). We obtain complete characterization for $n=6$ and settle the nonexistence of $(7,1,3)$-functions. We also obtain new $(9,1,4)$-functions by studying rotation symmetric functions that differ from the functions obtained through recursively amplifying a $(3,1,2)$-function.

Then we study the relation between the number of variables n and the polynomial degree of a kth order sensitive function. Because of resiliency-polynomial degree connections, adding the all variable linear function to f gives us a kth order dual sensitive function g. Using a recursive amplification method borrowed from complexity theory, we can get super-linear separation between n and $\mathsf{pdeg}(f)$ for functions with constant order sensitivity, but that it fails whenever the sensitivity order is super-constant. Then we use a connection between resiliency and polynomial degree, along with the Maiorana-McFarland construction, when k is either a constant or grows slowly with the number of variables n ($k=o(\log(n))$), to obtain several results such as non-constant separation between n and the real polynomial degree for functions with super-constant order of sensitivity. To the best of our knowledge this is the first time such generalized results have been shown. Our extension to higher order sensitivity is connected to variable (total) influence, as well as noise sensitivity of Boolean functions, which are relevant to cryptography and other areas. It would be interesting if some other constructions can be developed that strengthens the separation between the mentioned concepts.

Acknowledgments

We wish to thank the anonymous reviewers and the editors for their insightful and instructive suggestions that improved the technical as well as editorial quality of this paper. Subhamoy Maitra acknowledges the support of MeitY, Government of India, related to the initiative "Cluster - Cryptography, Information Security Education and Awareness (ISEA) Project Phase - III". The work of Deng Tang was supported in part by the National Key R&D Program of China (No. 2020YFA0712300) and the National Natural Science Foundation of China (grants 62272303 and 12031011).

Data availability

The data that has been used is confidential.

References

[1] A. Ambainis, Superlinear advantage for exact quantum algorithms, in: Proceedings of the forty-fifth annual ACM symposium on Theory of Computing, STOC'13, 2013, pp. 891–900.
[2] A. Ambainis, K. Balodis, A. Belovs, T. Lee, M. Santha, J. Smotrovs, Separations in query complexity based on pointer functions, J. ACM 64 (5) (2017) 1–24.
[3] H. Buhrman, R. De Wolf, Complexity measures and decision tree complexity: a survey, Theoret. Comput. Sci. 288 (1) (2002) 21–43.
[4] C. Carlet, P. Charpin, Cubic Boolean functions with highest resiliency, IEEE Trans. Inform. Theory 51 (2) (2005) 562–571.
[5] J.F. Dillon, Elementary Hadamard Difference Sets (Ph.D. Dissertation), Univ. of Maryland, 1974.
[6] E. Friedgut, Boolean functions with low average sensitivity depend on few coordinates, Combinatorica 18 (1) (1998) 27–36.
[7] E. Friedgut, G. Kalai, Every monotone graph property has a sharp threshold, Proc. Amer. Math. Soc. 124 (10) (1996) 2293–3002.
[8] S. Kavut, S. Maitra, M.D. Yucel, Search for Boolean functions with excellent profiles in the rotation symmetric class, IEEE Trans. Inform. Theory 53 (5) (2007) 1743–1751.
[9] G. Kindler, R. O'Donnell, Gaussian noise sensitivity and Fourier tails, in: Proceedings of the 27th Annual Computational Complexity Conference, 2012, pp. 137–147.
[10] A. Klivans, R. O'Donnell, R. Servedio, Learning intersections and thresholds of half-spaces, J. Comput. System Sci. 68 (4) (2004) 808–840.
[11] S. Maitra, P. Sarkar, Highly nonlinear resilient functions optimizing Siegenthaler's inequality, in: Adv. Crypt. — CRYPTO' 99. CRYPTO 1999, in: LNCS 1666, Springer, Berlin, Heidelberg, pp. 198–215, http://dx.doi.org/10.1007/3-540-48405-1_13.
[12] Y. Mansour, Learning Boolean functions via the Fourier transform, in: V. Roychowdhury, K.-Y. Siu, A. Orlitsky (Eds.), Theoretical Advances in Neural Computation and Learning, Kluwer Academic Publishers, 1994, pp. 391–424, Chapter 11.
[13] N. Nisan, M. Szegedy, On the degree of Boolean functions as real polynomials, Comput. Complexity 4 (1994) 301–313.
[14] N. Nisan, A. Wigderson, On rank vs. communication complexity, Comb. 15 (1995) 557–565.
[15] R. O'Donnell, Analysis of Boolean Functions, Cambridge University Press, 2014.
[16] R. O'Donnell, J. Wright, Y. Zhou, The Fourier entropy–influence conjecture for certain classes of Boolean functions, in: Proc. of Automata, Languages and Programming – 38th International Colloquium, 2011, pp. 330–341.
[17] P. Sarkar, S. Maitra, Construction of nonlinear Boolean functions with important cryptographic properties, in: Adv. Crypt. - EUROCRYPT 2000, in: LNCS 1807, Springer-Verlag, 2000, pp. 485–506.
[18] P. Sarkar, S. Maitra, Efficient implementation of cryptographically useful "large" Boolean functions, IEEE Trans. Comp. 52 (4) (2003) 410–417.
[19] Qichun Wang, Chik How Tan, Cryptographic Boolean functions with a large number of variables, in: 2014 IEEE International Symposium on Information Theory, Honolulu, HI, USA, June 29 - July 4, 2014, pp. 1534–1538.