# Subspace Differential Uniformity

Sondre Rønjom[1,2], Arne Sandrib[1,2] and Joakim Sunde[1]

[1] University of Bergen, Department of Informatics, Bergen, Norway
[2] Nasjonal sikkerhetsmyndighet, Kolsås, Norway

**Abstract.** The main contribution of this paper is to introduce *Subspace Differential Uniformity* (SDU) for S-boxes and block ciphers. The SDU is essentially a measure of how well any function spreads input differences clustered in affine subspaces away from affine clusters in output differences. We provide some lower bounds for the SDU and describe an efficient algorithm for computing the SDU. Moreover, we provide results for some popular classes of S-boxes up to $n = 8$.

**Keywords:** Permutations · differential uniformity · differential cryptanalysis

## 1 Introduction

In this paper, we investigate S-box properties that facilitate attacks against block ciphers by exploiting biased difference distributions over cosets of linear subspaces. Invariant subspace attacks, introduced in [LAAZ11], and further generalized in [LMR15], have proven surprisingly effective against numerous designs despite being relatively simple to mitigate. These attacks rely on the property that input and output differences remain within the same fixed subspace of the state space across all rounds. The case where differences follow trails of distinct subspaces was addressed in [GRR17], and subsequently generalized to trails of differences in affine subspaces in [LTW18].

Most common subspace trail attacks exploit structural symmetries inherent in the cipher's architecture, such as invariant subspaces in the linear layer, and typically disregard the specific S-box definition. Such trails persist independently of the chosen S-box. Conversely, certain cases exist where the S-box itself maps elements of a fixed coset of a subspace to itself which has been leveraged to mount weak-key invariant subspace attacks [GJN+16].

To mitigate differential trail clustering within affine subspaces over multiple rounds, it is important that the S-box remains balanced regarding differences in cosets of linear subspaces. Specifically, for fixed linear subspaces $U$ and $V$, the probability that an input difference in $U + a$ maps to an output difference in $V + b$ should be as uniform as possible for all choices of $a$ and $b$. While the existence of probability-1 trails within the S-box (cf. [GJN+16]) can lead to deterministic weak-key trails over the full cipher, the more general case where an S-box is merely biased on cosets is less understood.

Consider a block cipher with a round function comprising a layer of $k$ parallel $n$-bit S-boxes, denoted by $S^{\otimes k}$, followed by a $k \times k$ binary linear layer $L$ operating on $n$-bit blocks. Assume there exist linear subspaces $U, V, W \subset \mathbb{F}_2^n$ such that input differences in a coset of $U$ do not distribute sufficiently evenly across the cosets of $V$, and similarly, differences in cosets of $V$ map unevenly to cosets of $W$. This behavior can be extended to the full state space by constructing S-box dependent probabilistic trails of the form

E-mail: sondre.ronjom@uib.no (Sondre Rønjom), arne.sandrib@student.uib.no (Arne Sandrib), joakim.sunde@uib.no (Joakim Sunde)

$U^{\otimes k}+a \to V^{\otimes k}+b \to W^{\otimes k}+c$. By fixing biased input/output coset difference distributions

$$U^{\otimes k} + a \xrightarrow{S^{\otimes k}} V^{\otimes k} + b', \tag{1}$$

the role of the linear layer $L$ is to translate these cosets. Since $L$ is binary and operates on $n$-bit blocks, it follows that

$$L(V^{\otimes k} + b') = L(V^{\otimes k}) + L(b') = V^{\otimes k} + b, \tag{2}$$

where $b = L(b')$.

However, searching for such S-box dependent subspace trails requires precise knowledge of the subspace difference distributions of the S-box itself, which is a topic that has not been thoroughly investigated in the literature. Thus, investigating what we term the *subspace differential uniformity* of S-boxes is an essential step towards understanding how block ciphers can avoid clustering of differences in affine subspaces.

### 1.1 Contributions

Here we introduce the *Subspace Difference Distribution Table* (SDT) and *Subspace Differential Uniformity* (SDU) as a measure of subspace difference uniformity of S-boxes and block ciphers. Our main contributions are as follows. First we provide concrete bounds for SDU in the generic case, followed by improved bounds for the quadratic case. Then we introduce an efficient algorithm[1] for computing the SDU for an S-box together with an analysis. We use this algorithm to compute the SDUs for popular classes of S-boxes and provide comparisons and further directions. In the end we provide examples of 64-bit block ciphers where the 16 classes of optimal 4-bit S-boxes are distinguishable based on their SDU.

## 2 Preliminaries

We give an overview of some foundational notions in the study of vectorial Boolean functions. For a more thorough treatment we refer to [Car21a].

We will use the identification of the finite field $\mathbb{F}_{2^n}$ with the vector space $\mathbb{F}_2^n$ throughout the paper, and $x \cdot y$ is the standard dot product when $x, y \in \mathbb{F}_2^n$. For a set $S \subseteq \mathbb{F}_2^n$ let $S^* = S \setminus \{0\}$.

Differential uniformity is a measure of how well an S-box behaves against differential cryptanalysis. For a function $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ and a pair $(a, b) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$, let

$$\delta_f(a, b) = |\{x \in \mathbb{F}_2^n : f(x + a) + f(x) = b\}| \tag{3}$$

denote the $2^n \times 2^n$ *difference distribution table* for $f$. The *normalized Differential Probability* (DP) for the pair $(a, b)$ is given by

$$DP_f(a, b) = \frac{\delta_f(a, b)}{2^n}. \tag{4}$$

The maximum value among the entries $(a, b) \in (\mathbb{F}_2^n)^* \times \mathbb{F}_2^n$ of $\delta_f$ is called the differential uniformity of $f$, and if this maximum value is 2 the corresponding function is called *Almost Perfect Nonlinear* (APN). We also define $DP_f^{\max} = \max_{a \neq 0, b} DP_f(a, b)$, which is equal to $2/2^n = 2^{1-n}$ if and only if $f$ is APN.

[1] A Python implementation is available at https://github.com/arnesandrib/subspace_equivalence

If we let the graph of $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ be denoted $G_f = \{(x, f(x)) \,|\, x \in \mathbb{F}_2^n\}$, then $f$ is an APN function if and only if $G_f$ is a *Sidon set* in $((\mathbb{F}_2^n)^2, +)$, i.e. $|G_f + G_f|$ attains the maximal possible value $\binom{2^n}{2} + 1$.

The *differential spectrum* $S_f$ of a function $f$ is the distribution of values in $\delta_f(a, b)$, i.e. $S_f = (m(s) \,|\, 0 \le s \le 2^n)$ where $m(s) = |\{(a, b) \in \mathbb{F}_2^n \times \mathbb{F}_2^n \mid \delta_f(a, b) = s\}|$.

Two S-boxes with equal differential uniformity can have different differential spectra, which also inspires our interest in how this affects differential uniformity in higher dimensions.

S-boxes can be classified up to various equivalence relations and there is a hierarchy of these equivalence relations preserving various cryptographically relevant properties of the S-box. Two functions $f$ and $g$ are said to be *affine equivalent* if there exist invertible affine maps $A, B$ such that $g = B \circ f \circ A$, and are called *extended-affine* equivalent (EA-equivalent) if there exists some affine map $C$ such that $g = B \circ f \circ A + C$. Further on, two functions $f$ and $g$ are called *CCZ-equivalent* if there is an affine permutation $A$ on $\mathbb{F}_2^{2n}$ such that $A(G_f) = G_g$.

Differential uniformity is preserved all the way up to CCZ equivalence, but for *SDU* tables, which we will shortly define, there are usually functions with different *SDU*-tables within the CCZ-class of a given function.

For $\alpha, \beta \in \mathbb{F}_2^n$, the *Walsh-Hadamard transform* of $f$ is defined by

$$W_f(\alpha, \beta) = \sum_{x \in \mathbb{F}_2^n} (-1)^{\beta \cdot f(x) + \alpha \cdot x},$$

where $\cdot$ denotes the standard inner product on $\mathbb{F}_2^n$.

The *linear approximation table* (LAT) of $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ is the table $\mathrm{LAT}_f(\alpha, \beta)$, $\alpha, \beta \in \mathbb{F}_2^n$, whose entries are

$$\mathrm{LAT}_f(\alpha, \beta) = |\{x \in \mathbb{F}_2^n : \alpha \cdot x = \beta \cdot f(x)\}| - 2^{n-1}.$$

Equivalently, $W_f(\alpha, \beta) = 2\,\mathrm{LAT}_f(\alpha, \beta)$, so that $W_f(\alpha, \beta) = 0$ if and only if the approximation $\alpha \cdot x = \beta \cdot f(x)$ holds with probability exactly $1/2$. The function $f$ is called *bent* (or *vectorial bent*) if $n$ is even and $\left|W_f(\alpha, \beta)\right| = 2^{n/2}$ holds for all $(\alpha, \beta) \in \mathbb{F}_2^n \times (\mathbb{F}_2^n)^*$. Equivalently $\left|\mathrm{LAT}_f(\alpha, \beta)\right| = 2^{n/2-1}$ for all $(\alpha, \beta) \in \mathbb{F}_2^n \times (\mathbb{F}_2^n)^*$.

For odd $n$, a function $f$ is called *Almost Bent* (AB) if, for every $\beta \in (\mathbb{F}_2^n)^*$, the values of the Walsh-Hadamard transform satisfy $W_f(\alpha, \beta) \in \{0, \pm 2^{(n+1)/2}\}$ for all $\alpha \in \mathbb{F}_2^n$, meaning that $\mathrm{LAT}_f(\alpha, \beta) \in \{0, \pm 2^{(n-1)/2}\}$ for all $(\alpha, \beta) \in \mathbb{F}_2^n \times (\mathbb{F}_2^n)^*$.

The *linearity* of a function $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ is defined as

$$L(f) = \max_{\substack{\alpha \in \mathbb{F}_2^n \\ 0 \neq \beta \in \mathbb{F}_2^n}} |W_f(\alpha, \beta)|$$

and its *nonlinearity* is defined as $nl(f) = 2^{n-1} - \frac{1}{2} L(f)$.

**Subspaces** Let $\mathcal{V}(n, d)$ denote the set of $d$-dimensional subspaces of $\mathbb{F}_2^n$, and let $\mathcal{A}(n, d)$ be the set of affine $d$-dimensional subspaces (or *flats*) of $\mathbb{F}_2^n$. Recall the Gaussian binomial coefficient, defined as

$$\binom{n}{d}_q = \prod_{i=0}^{d-1} \frac{1 - q^{n-i}}{1 - q^{i+1}},$$

where $q$ is a prime power. This number counts the number of $d$-dimensional linear subspaces of $\mathbb{F}_q^n$. Consequently, the number of affine $d$-dimensional subspaces of $\mathbb{F}_2^n$ is given by $|\mathcal{A}(n, d)| = 2^{n-d}\binom{n}{d}_2$.

Finally recall that any $(n-1)$-dimensional linear subspace of $\mathbb{F}_2^n$ is called a *linear hyperplane*, and any $(n-1)$-dimensional affine subspace of $\mathbb{F}_2^n$ is called an *affine hyperplane*.

When writing hyperplane, we mean affine hyperplane unless something else is explicitly specified.

# 3 Subspace Differential Uniformity

In the following, we introduce the notions that will be the focus of study for the remainder of this paper. Fix input and output dimensions $0 \leq d_1, d_2 \leq n$. For a pair of linear subspaces $(U, V) \in \mathcal{V}(n, d_1) \times \mathcal{V}(n, d_2)$ and a function $f$ on $\mathbb{F}_2^n$, we call the $2^{n-d_1} \times 2^{n-d_2}$ table $\Delta_f^{U,V}$ with entries

$$\begin{aligned}\Delta_f^{U,V}(a,b) =& |\{(x,u) \in \mathbb{F}_2^n \times U : u + a \neq 0, f(x) + f(x+u+a) \in V + b\}| \\ =& \sum_{u \in (U+a)^*, v \in V+b} \delta_f(u,v)\end{aligned} \tag{5}$$

the *Subspace Difference Distribution Table* (SDT) for $f$ taken over the unique cosets $U + a$ and $V + b$. The value $\Delta_f^{U,V}(a,b)$ maximized over all $(U, V) \in \mathcal{V}(n, d_1) \times \mathcal{V}(n, d_2)$

$$\Delta_f^{d_1,d_2} = \max_{\substack{U+a \in \mathcal{A}(n,d_1) \\ V+b \in \mathcal{A}(n,d_2)}} \Delta_f^{U,V}(a,b)$$

is called the *Subspace Differential Uniformity* (SDU) of $f$, while we call

$$\widehat{\Delta}_f^{d_1,d_2} = \max_{\substack{U \in \mathcal{V}(n,d_1) \\ V \in \mathcal{V}(n,d_2)}} \Delta_f^{U,V}(0,0)$$

the *linear Subspace Differential Uniformity* (linear SDU). We cover linear SDU as a special case of SDU, since real-world applications of subspace trail cryptanalysis most frequently focus on linear subspaces.

Analogous to the DDT and DP, we define the *Subspace Difference Probability table* (SDP) by normalizing by the number of input pairs $(x, u)$ such that $x + u \in (U + a)^*$. By Eq. (5) there are $2^n|(U + a)^*|$ such pairs. We define

$$SDP_f^{U,V}(a,b) = \frac{\Delta_f^{U,V}(a,b)}{2^n|(U+a)^*|}.$$

In turn we define the *maximal Subspace Difference Probability* (maximal SDP) to be

$$SDP_f^{d_1,d_2} = \max_{\substack{U+a \in \mathcal{A}(n,d_1) \\ V+b \in \mathcal{A}(n,d_2)}} SDP_f^{U,V}(a,b).$$

If, for a set $A \subset \mathbb{F}_2^n$ we let $e_A$ denote the indicator vector of length $2^n$ such that $e_A(u) = 1$ if $u \in A$, then the subspace differential uniformity is the maximal value of

$$e_{A^*}^T \cdot T_f \cdot e_B$$

where $A$ and $B$ range over affine subspaces and $T_f$ denotes the $2^n \times 2^n$ DDT matrix.

We use the convention that

$$\Delta_f(a,b) = \Delta_f^{\{0\},\{0\}}(a,b),$$

so $\Delta_f(a, b) = \delta_f(a, b)$, i.e. the normal differential uniformity, except that we have $\Delta_f(0, 0) = 0$.

If we think of $f$ as a block cipher, the linear invariant subspace case, i.e., when $U = V$ and $(a, b) = (0, 0)$, was introduced as invariant subspace attacks in [LAAZ11]. The further generalization to different linear subspaces $U \neq V$ was introduced as subspace trail cryptanalysis in [GRR17]. The case of affine subspaces was then investigated to some extent in [LTW18]. However, they do not treat probabilistic subspace trails over S-boxes and block ciphers. While they stress that S-boxes without linear structures behave like any generic S-box in their analysis, this only holds for probability 1 trails. In our case even functions with the same differential uniformity can behave quite differently in higher dimensions. In particular, two APN permutations can differ significantly with respect to differential properties in higher dimensions.

As a motivation, for two different S-boxes $f$ and $g$, we can have $\Delta_f^{0,0} > \Delta_g^{0,0}$ (e.g. one that is APN and one that is 4-uniform) while we can have $\Delta_f^{d_1,d_2} < \Delta_g^{d_1,d_2}$. If the differential spectrum of $g$ has a few values higher than $f$, but significantly more that are lower, the function $f$ can be worse at avoiding affine subspace difference clusters than $g$. The SDU then accounts for how well an S-box or a block cipher avoids affine clustering of differences.

# 4 Bounds for SDU

In this section we provide bounds for the SDU, both for the generic case and for the special case of quadratic permutations.

## 4.1 Generic Bounds

We begin by establishing a recursive bound on the SDU values which identifies a fundamental relationship between the values $\Delta_f^{d_1,d_2}$ (resp. $\widehat{\Delta}_f^{d_1,d_2}$) as $d_1$ or $d_2$ increases for a fixed function $f$ on $\mathbb{F}_2^n$. In particular, we specify how much the SDU can grow from one entry to the next.

**Lemma 1.** *Let $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$. Then for $t \in \{0, 1\}$, the SDU values satisfy*

$$\Delta_f^{d_1+t,d_2+(1-t)} \leq 2 \cdot \Delta_f^{d_1,d_2} \quad \text{and}$$
$$\widehat{\Delta}_f^{d_1+t,d_2+(1-t)} \leq 3 \cdot \widehat{\Delta}_f^{d_1,d_2},$$

*where the claim about the linear SDU holds only when $d_1, d_2 \geq 1$.*

*Proof.* Any $d$-dimensional affine subspace can be written as the disjoint union of two $(d-1)$-dimensional affine subspaces. Consider input dimension $d_1 + 1$ and suppose $U + a = U_1 + a_1 \cup U_2 + a_2$ for disjoint $d_1$-dimensional subspaces. Then $\Delta^{U,V}(a, b) = \Delta^{U_1,V}(a_1, b) + \Delta^{U_2,V}(a_2, b) \leq 2\Delta_f^{d_1,d_2}$. The same argument applies when increasing output dimension.

The linear case follows from exactly the same reasoning, except that it uses the fact that any linear $d$-dimensional subspace can be written as the union of three $(d-1)$-dimensional linear subspaces whenever $d > 1$. □

Observe that recursively using Lemma 1 gives the following general bound for the SDU of a function given the value of the SDU on input/output hyperplanes:

$$\Delta_f^{d_1,d_2} \geq \frac{\Delta_f^{n-1,n-1}}{2^{2n-2-(d_1+d_2)}} \qquad \text{and} \qquad \widehat{\Delta}_f^{d_1,d_2} \geq \frac{\widehat{\Delta}_f^{n-1,n-1}}{3^{2n-2-(d_1+d_2)}}.$$

For certain functions, the bounds of Lemma 1 are tight for at least some small values of $d_1$ and $d_2$. However, in general there is no guarantee that $\Delta_f^{d_1,d_2}$ grows as $d_1$ and $d_2$ grows.

Take for instance the identity function $f(x) = x$. If $d_1 < d_2$, then $\Delta_f^{d_1+1,d_2} = \Delta_f^{d_1,d_2}$. Since for any subspace $U$ if $x + y \in U$ then $f(x) + f(y) \in U$. So the SDU entry only depends on whichever of $d_1$ and $d_2$ is smaller.

Since the linear SDU is a special case of SDU, we obviously have $\widehat{\Delta}_f^{d_1,d_2} \leq \Delta_f^{d_1,d_2}$ and $\Delta_f^{d_1,d_2} \leq \widehat{\Delta}_f^{d_1+1,d_2+1}$ whenever $d_1, d_2 < n$ since any $d$-dimensional affine subspace is contained in a $(d+1)$-dimensional linear subspace.

The first baseline bound is based on a simple averaging argument over the affine subspaces. Since at least one of the affine subspaces must achieve the average value the bound follows. We do not believe that any S-box can obtain these bounds (and in fact prove that for most dimensions $d_1$ and $d_2$ it is not possible, see e.g., Theorem 5 and the discussion above Corollary 2). Achieving this bound would require the S-box to diffuse all input differences perfectly and distribute the output differences uniformly over every output subspace.

**Theorem 1.** *For a permutation $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$,*

$$\Delta_f^{d_1,d_2} \geq 2 \left\lceil \frac{2^{n+d_1+d_2}}{2^{n+1}-2} \right\rceil.$$

*Proof.* Fix some nonzero vector $w \in (\mathbb{F}_2^n)^*$ and let $d \leq n$ be some fixed dimension. Define

$$C_d^w = \{U \in \mathcal{A}(n,d) : w \in U \text{ and } 0 \notin U\} \qquad \text{and} \qquad A_d = \{U \in \mathcal{A}(n,d) : 0 \notin U\}$$

to be subsets of $\mathcal{A}(n,d)$ with $C_d^w \subseteq A_d$. Observe by symmetry that $|C_d^w|$ is independent of the choice of $w$, we may define the number $C_d = |C_d^w|$.

We count the number of pairs $(w, W)$ where $0 \neq w \in W$ and $W \in A_d$. Fixing $W$ and running through possible $w$ shows that there are $2^d|A_d|$ such pairs. Conversely, fixing nonzero $w$ (of which there are $2^n - 1$) and then running through possible spaces $W$ yields $(2^n - 1)C_d$ such pairs.

Since $n > 0$, we get

$$C_d = \frac{|A_d| \cdot 2^d}{2^n - 1}.$$

Now fix input dimension $d_1$ and output dimension $d_2$ and consider the total contribution of the elements in the affine subspaces not containing 0. Let

$$\begin{aligned} S &= \sum_{\substack{U \in \mathcal{V}(n,d_1) \\ a \in \mathbb{F}_2^n \\ 0 \notin U+a}} \sum_{\substack{V \in \mathcal{V}(n,d_2) \\ b \in \mathbb{F}_2^n \\ 0 \notin V+b}} \sum_{u \in U+a} \sum_{v \in V+b} \Delta_f(u,v) \\ &= C_{d_1} C_{d_2} \sum_{u,v \in (\mathbb{F}_2^n)^*} \Delta_f(u,v) \\ &= C_{d_1} C_{d_2} 2^n (2^n - 1) \\ &= |A_{d_1}||A_{d_2}| \frac{2^{d_1+d_2+n}}{2^n - 1}. \end{aligned}$$

Since $S$ is the sum of the values $\Delta_f(a,b)$ over all $|A_{d_1}||A_{d_2}|$ pairs of affine subspaces not containing 0, at least one of these pairs of subspaces must achieve at least the average value. We get $\Delta_f^{d_1,d_2} \geq \frac{S}{|A_{d_1}||A_{d_2}|} \geq \frac{2^{n+d_1+d_2}}{2^n-1}$. Since $\Delta_f^{d_1,d_2}$ must be an even integer we get that

$$\Delta_f^{d_1,d_2} \geq 2 \left\lceil \frac{2^{n+d_1+d_2}}{2^{n+1}-2} \right\rceil. \qquad \square$$

### 4.2 Connections Between SDU Values and the Walsh Transform

In the following, we establish a bound for the SDU of a permutation where the input/output spaces are hyperplanes connecting the SDU to the Walsh-Hadamard transform of the function. We establish two lemmas before proving the main result of this section, namely Theorem 2.

Given a function $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$, the 2-dimensional Walsh-Hadamard transform of its DDT viewed as a function $\delta_f\colon \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{Z}$ is closely related to the LAT table of $f$. Indeed, in section 2.2 of [DH18], the authors show that

$$\mathrm{LAT}_f(\alpha,\beta)^2 = \frac{1}{4}\sum_{x,y\in\mathbb{F}_2^n}(-1)^{\alpha\cdot x+\beta\cdot y}\delta_f(x,y). \tag{6}$$

Recall that while $\delta_f(0,0) = 2^n$, we use the convention that $\Delta_f(0,0) = 0$. Then the following lemma is an immediate consequence of Eq. (6).

**Lemma 2.** *Let $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a function. Then*

$$\sum_{x,y\in\mathbb{F}_2^n}\Delta_f(x,y)(-1)^{\alpha\cdot x+\beta\cdot y} = W_f(\alpha,\beta)^2 - 2^n.$$

Using this connection of the LAT of a function and its derivative, we can precisely connect the value of the SDT on hyperplanes to the Walsh transform and hence the nonlinearity of the function.

**Lemma 3.** *Let $U+a$ and $V+b$ be hyperplanes of $\mathbb{F}_2^n$. If $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ is a permutation, then*

$$\Delta_f^{U,V}(a,b) = \frac{1}{4}\left[2^{2n} - 2^n + (-1)^{\epsilon}(-2^n) + (-1)^{\eta}(-2^n) + (-1)^{\epsilon+\eta}(W_f(\alpha,\beta)^2 - 2^n)\right].$$

*Proof.* Since $U+a$ and $V+b$ are hyperplanes, there are $\alpha,\beta\in\mathbb{F}_2^n$ and $\epsilon,\eta\in\mathbb{F}_2$ such that

$$U+a = \{x\in\mathbb{F}_2^n : \alpha\cdot x = \epsilon\} \quad \text{and} \quad V+b = \{x\in\mathbb{F}_2^n : \beta\cdot x = \eta\}.$$

The associated indicator functions are

$$\mathbf{1}_{U+a}(x) = \frac{1}{2}(1+(-1)^{\alpha\cdot x+\epsilon}) \quad \text{and} \quad \mathbf{1}_{V+b}(x) = \frac{1}{2}(1+(-1)^{\beta\cdot x+\eta}).$$

We compute

$$\begin{aligned}\Delta_f^{U,V}(a,b) &= \sum_{u\in U+a}\sum_{v\in V+b}\Delta_f(u,v) = \sum_{x,y\in\mathbb{F}_2^n}\Delta_f(x,y)\mathbf{1}_{U+a}(x)\mathbf{1}_{V+b}(y)\\ &= \frac{1}{4}\sum_{x,y\in\mathbb{F}_2^n}\Delta_f(x,y)\left(1+(-1)^{\alpha\cdot x+\epsilon}+(-1)^{\beta\cdot y+\eta}+(-1)^{\alpha\cdot x+\beta\cdot y+\epsilon+\eta}\right).\end{aligned}$$

Expanding gives

$$\begin{aligned}\Delta_f^{U,V}(a,b) = \frac{1}{4}\Bigg[&\sum_{x,y\in\mathbb{F}_2^n}\Delta_f(x,y) + (-1)^{\epsilon}\sum_{x,y\in\mathbb{F}_2^n}\Delta_f(x,y)(-1)^{\alpha\cdot x}\\ &+ (-1)^{\eta}\sum_{x,y\in\mathbb{F}_2^n}\Delta_f(x,y)(-1)^{\beta\cdot y} + (-1)^{\epsilon+\eta}\sum_{x,y\in\mathbb{F}_2^n}\Delta_f(x,y)(-1)^{\alpha\cdot x+\beta\cdot y}\Bigg]\end{aligned}$$

Using the fact that $f$ is a permutation, we get

$$\sum_{x,y\in\mathbb{F}_2^n}\Delta_f(x,y)(-1)^{\alpha\cdot x}=\sum_{x\in\mathbb{F}_2^n}(-1)^{\alpha\cdot x}\sum_{y\in\mathbb{F}_2^n}\Delta_f(x,y)=\sum_{x\in(\mathbb{F}_2^n)^*}(-1)^{\alpha\cdot x}2^n=-2^n.$$

Furthermore, we have $\sum_{x,y\in\mathbb{F}_2^n}\Delta_f(x,y)=(2^n-1)2^n=2^{2n}-2^n$. Putting this into the above equation gives

$$\begin{aligned}\Delta_f^{U,V}(a,b)&=\frac{1}{4}\left[2^{2n}-2^n+(-1)^{\epsilon}(-2^n)+(-1)^{\eta}(-2^n)+(-1)^{\epsilon+\eta}\sum_{x,y}\Delta_f(x,y)(-1)^{\alpha\cdot x+\beta\cdot y}\right]\\&=\frac{1}{4}\left[2^{2n}-2^n+(-1)^{\epsilon}(-2^n)+(-1)^{\eta}(-2^n)+(-1)^{\epsilon+\eta}(W_f(\alpha,\beta)^2-2^n)\right],\end{aligned}$$

where the final equality follows by Lemma 2. □

A consequence of Lemma 3 is that the bottom right corner ($d_1=d_2=n-1$) of the SDU table of some function $f\colon\mathbb{F}_2^n\to\mathbb{F}_2^n$ completely encodes the nonlinearity of $f$. We give a precise expression for this in the following theorem:

**Theorem 2.** *If $f\colon\mathbb{F}_2^n\to\mathbb{F}_2^n$ is a permutation, then*

$$\Delta_f^{n-1,n-1}=2^{2n-2}+\frac{1}{4}\max_{\alpha,\beta\neq 0}W_f(\alpha,\beta)^2.$$

*Proof.* Let $U+a$ and $V+b$ be hyperplanes of $\mathbb{F}_2^n$. Then there are $\alpha,\beta\in(\mathbb{F}_2^n)^*$ and $\epsilon,\eta\in\mathbb{F}_2$ such that $U+a=\{x\in\mathbb{F}_2^n:\alpha\cdot x=\epsilon\}$ and $V+b=\{x\in\mathbb{F}_2^n:\beta\cdot x=\eta\}$. Using Lemma 3 we have that

$$\Delta_f^{U,V}(a,b)=\frac{1}{4}\left[2^{2n}-2^n+(-1)^{\epsilon}(-2^n)+(-1)^{\eta}(-2^n)+(-1)^{\epsilon+\eta}(W_f(\alpha,\beta)^2-2^n)\right].$$

Using that $W_f(\alpha,\beta)$ is independent of $\epsilon,\eta$, one can easily check that this expression is maximized for some $\alpha,\beta$ when $\epsilon,\eta=1$, in which case

$$\Delta_f^{U,V}(a,b)=2^{2n-2}+\frac{1}{4}W_f(\alpha,\beta)^2,$$

which reaches its maximum when $|W_f(\alpha,\beta)|$ is maximal. □

Recall that for permutations $f\colon\mathbb{F}_2^n\to\mathbb{F}_2^n$, there is a well-known bound

$$\max_{\alpha\neq 0,\beta\neq 0}|W_f(\alpha,\beta)|\geq 2^{(n+1)/2}. \tag{7}$$

Any nonzero vector $\alpha\in(\mathbb{F}_2^n)^*$ determines some hyperplane by $\{x\in\mathbb{F}_2^n:\alpha\cdot x=\epsilon\}$ for $\epsilon\in\mathbb{F}_2$. Hence, combining Eq. (7) with Theorem 2 gives the following universal bound on permutations:

**Corollary 1.** *If $f\colon\mathbb{F}_2^n\to\mathbb{F}_2^n$ is a permutation, then*

$$\Delta_f^{n-1,n-1}\geq 2^{2n-2}+2^{n-1}.$$

Another consequence of Theorem 2 is that AB permutations achieve the optimal subspace diffusion on hyperplanes. Recall that AB permutations exist only for odd $n$, having linearity $2^{(n+1)/2}$. Consequently, if $n$ is odd and $f$ is an AB permutation, then $f$ attains the lowest possible value for $\Delta_f^{n-1,n-1}$, namely

$$\Delta_f^{n-1,n-1}=2^{2n-2}+\frac{1}{4}(2^{(n+1)/2})^2=2^{2n-2}+2^{n-1}. \tag{8}$$

In particular, we observe that the bound of Corollary 1 is achieved by the AB permutations.

Furthermore we conclude that a permutation can not be perfect in the sense that it reaches the averaging bound of Theorem 1 for hyperplanes. This can easily be seen by comparing with the values of Eq. (8); the averaging bound gives $\Delta_f^{n-1,n-1} \geq 2^{2n-2}+2^{n-2}+2$ which in general is lower than $2^{2n-2}+2^{n-1}$.

Applying Theorem 2 to compute the SDU for APN permutations in general is not straightforward as the nonlinearity of APN functions generally is not well understood. The following result of Carlet provides a useful bound:

**Lemma 4** ([Car21b, Corollary 3]). *For any APN function $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ where $\{|W_f(\alpha,\beta)| : \alpha \in \mathbb{F}_2^n, \beta \in (\mathbb{F}_2^n)^*\}$ takes its maximum value for at least two different $(\alpha,\beta)$ then*

$$nl(f) \geq 2^{n-1} - \frac{1}{2}\sqrt[4]{2^{4n-1}-2^{3n}}.$$

As noted by Carlet [Car21b, p. 24], all known APN functions satisfy the condition in Lemma 4; in particular, it is provably true for all power functions and all quadratic functions. If the condition turns out to hold in general, then for all APN permutations $f$ on $\mathbb{F}_2^n$ we get that

$$\Delta_f^{n-1,n-1} \leq 2^{2n-2} + \frac{1}{4}\sqrt[4]{2^{4n-1}-2^{3n}}.$$

Also note that in [CP26] they show that APN functions satisfy

$$L(f) \leq 2^n - 4$$

where $L$ denotes the linearity of the function $f$. However this bound is only better for very small dimension $n$.

We conclude this section by linking the bounds on SDU on hyperplanes to a bound on the entire SDU, by applying Section 4.1 to the Theorem 2.

**Theorem 3.** *For a permutation $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ and $0 \leq d_1, d_2 \leq n-1$ we have*

$$\Delta_f^{d_1,d_2} \geq 2^{d_1+d_2} + 2^{d_1+d_2-2n} \max_{\alpha,\beta\neq 0} W_f(\alpha,\beta)^2.$$

*Proof.* Combining Theorem 2 with Lemma 1 gives

$$\begin{aligned}\Delta_f^{d_1,d_2} &\geq \frac{\Delta_f^{n-1,n-1}}{2^{2n-2-(d_1+d_2)}} \\ &= \frac{2^{2n-2} + \frac{1}{4}\max_{\alpha,\beta\neq 0} W_f(\alpha,\beta)^2}{2^{2n-2-(d_1+d_2)}} \\ &= 2^{d_1+d_2} + 2^{d_1+d_2-2n} \max_{\alpha,\beta\neq 0} W_f(\alpha,\beta)^2.\end{aligned}$$

□

When $d_1 = d_2 = n-1$, Theorem 3 gives exactly the same as Theorem 2. This slightly improves the averaging bound for larger values of $d_1, d_2$. Again using that $\max_{\alpha,\beta\neq 0} |W_f(\alpha,\beta)| \geq 2^{(n+1)/2}$ when $f$ is a permutation, we get the following result.

**Corollary 2.** *If $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ is a permutation, then*

$$\Delta_f^{d_1,d_2} \geq 2^{d_1+d_2} + 2^{d_1+d_2+1-n}.$$

Another consequence of Theorem 3 is that if $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ has a linear component (i.e., minimal nonlinearity), then its SDU values are abnormally high.

**Corollary 3.** *If the permutation $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ has a linear component function $\beta \cdot f(x)$ for some non-zero $\beta \in \mathbb{F}_2^n$, then, for arbitrary $d_1, d_2$ the SDU of $f$ has the following restrictions:*

$$\Delta_f^{d_1,d_2} \geq 2^{d_1+d_2+1}.$$

*I.e., in the affine case we get the highest possible values in the rightmost column and the bottom row are no better than the identity function.*

*Proof.* By Theorem 2, we have $\Delta_f^{n-1,n-1} = 2^{2n-2} + \frac{1}{4}(2^n)^2 = 2 \cdot 2^{2n-2} = 2^{2n-1}$. Combining with Lemma 1 for any $d_1, d_2$, we get $\Delta_f^{d_1,d_2} \geq 2^{2n-1}/2^{2n-2-(d_1+d_2)}$. □

Corollary 3 may be expressed in terms of maximal SDP for the permutation $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ with a linear component. We get that

$$SDP_f^{d_1,n-1} \geq \frac{2^{d_1+(n-1)+1}}{2^{n+d_1}} = 1 \qquad \text{and} \qquad SDP_f^{n-1,d_2} \geq \frac{2^{(n-1)+d_2+1}}{2^{n+(n-1)}} = 2^{d_2-n+1}.$$

## 4.3 Zero-Dimensional SDU

When $d_1 = 0$ we are just considering a single row of the DDT, and finding the maximal $d_2$-dimensional affine subspace that has the largest intersection with the support. Here we will just consider when we are guaranteed the "maximal" value given its differential uniformity. This question basically boils down to how many vectors in a set we need to guarantee it contains a $d$-dimensional affine subspace of $\mathbb{F}_2^n$. This problem has been studied quite a bit, see for example [KN25]. As far as we know the best bound given on this problem is given in [FY25], where the authors study the problem from quite a different perspective, specifically through Ramsey numbers of vector spaces. Below we state a simplified version of their result.

**Theorem 4** ([FY25, Theorem 4.2])**.** *Any set $S \subseteq \mathbb{F}_2^n$ where $|S| \geq 2^{n-\frac{n-d}{2^d-d}}$ contains a $d$-dimensional affine subspace.*

Using Theorem 4, we establish the following result.

**Lemma 5.** *Let $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ be an APN function and $n \geq 2^d$. Then*

$$\Delta_f^{0,d} = \Delta_f^{d,0} = 2^{d+1}.$$

*Proof.* If $d = 0$, the result follows directly from the fact that $f$ is APN, so suppose $d > 0$. First consider the case of $\Delta_f^{d,0}$. There must be some $v$, corresponding to an output difference, such that $\sum_{u \in \mathbb{F}_2^n} \Delta_f(u, v) \geq 2^n$. Then $\{v\}$ is a 0-dimensional affine space. Let the support of the map $u \mapsto \Delta_f(u, v)$ be denoted $S$. Then $|S| \geq 2^{n-1}$. Note (by abuse of notation) that $\Delta_f^{S,\{v\}} = 2^{d+1}$ if and only if $S$ contains a $d$-dimensional subspace, which is guaranteed by Theorem 4 if

$$|S| \geq 2^{n-\frac{n-d}{2^d-d}}. \tag{9}$$

Since $|S| \geq 2^{n-1}$ we just need to show that

$$2^{n-1} \geq 2^{n-\frac{n-d}{2^d-d}}$$

and equation 9 will hold. Note that since $n \geq 2^d$, we have

$$\frac{n-d}{2^d-d} \geq \frac{2^d-d}{2^d-d} = 1,$$

implying that

$$n - 1 \geq n - \frac{n-d}{2^d-d},$$

which means that $2^{n-1} \geq 2^{n-\frac{n-d}{2^d-d}}$ and we conclude that Eq. (9) holds.

The proof in the case of $\Delta_f^{0,d}$ is analogous, but slightly simpler. One may use *any* row $u$ of the DDT as this guarantees that $\sum_{v \in \mathbb{F}_2^n} \Delta_f(u,v) = 2^n$. Then the result follows by exactly the same argument; by setting $S$ equal to the support for $v \mapsto \Delta_f(u,v)$, the result follows from Theorem 4. □

### 4.4 One-Dimensional SDU

We consider when $\Delta_f^{1,1} \geq 8$ is guaranteed for a function $f$. Note that this is a tight bound for APN functions.

**Theorem 5.** *If $n \geq 3$ then for any function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$,*

$$\Delta_f^{1,1} \geq 8.$$

*Proof.* First note that since a 1-dimensional affine subspace of $\mathbb{F}_2^n$ is just any size 2 subset of $\mathbb{F}_2^n$, the statement is equivalent to saying that the DDT (excluding the top row) of $f$ contains a $2 \times 2$ submatrix whose entries add up to a number $\geq 8$. We demonstrate that any attempt to construct a DDT type matrix without the presence of such a submatrix leads to a contradiction, proving the theorem. We assume an APN type DDT, but for a higher differential uniformity the proof is very similar.

Since we are only concerned with the sums of entries in submatrices, we may freely permute the rows and columns of our matrix without affecting any of these sums. Hence we may assume that the left half of the first row consists of all 2s and the right half consists of all 0s.

Consider the next row of the DDT of $f$. Recall the assumption that there is no $2 \times 2$ submatrix of this DDT where every entry is 2. Thus, there may be at most one 2 in the left half of the second row, and at least $2^{n-1} - 1$ entries of value 2 in the right half. After applying suitable independent permutations of the columns within each half of the DDT of $f$, the first two rows can essentially have two possible configurations:

$$R_{12} = \begin{pmatrix} 2 & 2 & \dots & 2 & 0 & \dots & 0 & 0 \\ 2 & 0 & \dots & 0 & 2 & \dots & 2 & 0 \end{pmatrix} \quad \text{or} \quad R'_{12} = \begin{pmatrix} 2 & 2 & \dots & 2 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 2 & \dots & 2 \end{pmatrix}.$$

We construct a third row. In any configuration, placing more than one 2 on the left side gives a $2 \times 2$ submatrix of all 2s, so at least $2^{n-1} - 1 \geq 3$ of the 2s must be placed on the right hand side. However, in any of the configurations above, placing three 2s in the right half is impossible without creating a $2 \times 2$ submatrix whose entries are all 2s and we are done. □

## 5 Quadratic APN Functions

Quadratic functions are functions whose directional derivatives have affine hyperplanes as images. I.e., for some function $f \colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ and any $u \neq 0$, the set $\{D_u f(x) : x \in \mathbb{F}_2^n\}$ is an affine hyperplane. Since many well-known APN functions are quadratic, this case is of particular interest to us. Furthermore, quadraticness provides structure that in some cases allows for easier classification of functions.

We are particularly interested in permutations. If $f \colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ is a quadratic permutation then for any $u \in (\mathbb{F}_2^n)^*$ the image of the derivative $D_u f$ is an affine hyperplane not containing 0, since $f(x+u) \neq f(x)$ for all $x \in \mathbb{F}_2^n$. Consequently we denote by $\lambda_f(u) \in \mathbb{F}_2^n$ the unique vector such that

$$\mathrm{Im}(D_u f) = \{x \in \mathbb{F}_2^n : \lambda_f(u) \cdot x = 1\},$$

and we call $\lambda_f(u)$ the *normal vector of the derivative hyperplane of $f$ in direction $u$.*

## 5.1 Linear SDU of Quadratic APN Permutations

For the special case that is the linear SDU of quadratic APN permutations we are able to compute exact values for certain dimensions. First recall the following basic result about Gaussian binomial coefficients.

**Lemma 6.** *For any $n \geq 2, 1 \leq k \leq n-1$, we have*

$$(2^k - 1)\binom{n-1}{k}_2 < \binom{n}{k}_2.$$

We are now able to compute the diagonal of the SDU table for any quadratic APN permutation.

**Theorem 6.** *For any quadratic APN permutation $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ and $0 \leq k \leq n$ we have*

$$\widehat{\Delta}_f^{k,k} = 2^k(2^k - 1).$$

*Proof.* Let $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a quadratic APN permutation and $U \in \mathcal{V}(n,k)$ a fixed subspace of $\mathbb{F}_2^n$ with dimension $k$. Fix some $u \in U^*$. Then there is some $\lambda_f(u)$ such that $\mathrm{Im}(D_u f) = \{v \in \mathbb{F}_2^n : \lambda_f(u) \cdot v = 1\}$.

For any $V \in \mathcal{V}(n,k)$ define

$$\begin{aligned} S_u(V) &= |\{v \in V : \Delta_f(u,v) = 2\}| \\ &= |\{v \in V : v \in \mathrm{Im}(D_u f)\}| \\ &= |\{v \in V : \lambda_f(u) \cdot v = 1\}|. \end{aligned}$$

The map $l_u\colon v \mapsto \lambda_f(u) \cdot v$ is clearly balanced on $V$ unless $V$ is a subspace of the kernel of $l_u$. Therefore, we want to find a subspace $V$ for which $V \not\subseteq \mathrm{Ker}(l_u)$ for any $0 \neq u \in U$. We will show that such a space exists by a counting argument.

The number of $k$-dimensional subspaces of $\mathbb{F}_2^n$ is $\binom{n}{k}_2$. Similarly, the number of subspaces contained in $\mathrm{Ker}(l_u)$ is $\binom{n-1}{k}_2$. The number of such subspaces contained in at least one kernel is therefore at most $(2^k - 1)\binom{n-1}{k}_2$. From Lemma 6 we have

$$(2^k - 1)\binom{n-1}{k}_2 < \binom{n}{k}_2$$

so such a subspace must exist. Then, for these subspaces $U$ and $V$, we clearly get

$$\Delta_f^{U,V}(0,0) = \sum_{u \in U^*} 2 \cdot S_u(V) = \sum_{u \in U \setminus \{0\}} 2 \cdot 2^{k-1} = (2^k - 1)2^k$$

which completes the proof. □

## 5.2 SDU of Quadratic APN Permutations

It is desirable to compute all SDU values of quadratic functions theoretically. However, the proof technique of Theorem 6 is not directly applicable, since that map $l_u$ does not have to be balanced on affine subspaces of $\mathbb{F}_2^n$. We will now investigate the case of generic SDU, and finally prove Theorem 7 which is a partial bound of the SDU of quadratic APN permutations.

Let $H_f(u) = \mathrm{Im}(D_u f)$, $\lambda_f(u)$ the normal vector of $H_f(u)$, so $H_f(u) = \{z \in \mathbb{F}_2^n : \lambda_f(u) \cdot z = 1\}$ since the hyperplanes are affine when $f$ is a permutation. Finally, for some subspace $V \subseteq \mathbb{F}_2^n$, let $V_b^\perp = \{c \in V^\perp : b \cdot c = 1\}$. We of course have

$$\Delta_f^{U,V}(a,b) = 2 \cdot \sum_{u \in (U+a)^*} |H_f(u) \cap (V + b)|$$

**Lemma 7.** *For any quadratic APN permutation $f$ and affine subspace $V+b$ we have*

$$|H_f(u) \cap (V+b)| = \begin{cases} 2^{d_2} & \lambda_f(u) \in V_b^\perp \\ 0 & \lambda_f(u) \in V^\perp \backslash V_b^\perp \\ 2^{d_2-1} & \lambda_f(u) \notin V^\perp \end{cases}$$

*Proof.* Any element $z \in H_f(u) \cap (V+b)$ can be written as $z = v_0 + b$ with $v_0 \in V$ and satisfies $\lambda_f(u) \cdot z = 1$ which is equivalent to $\lambda_f(u) \cdot v_0 = \lambda_f(u) \cdot b + 1$. We want to decide the number of solutions $v_0 \in V$ to this equation. Let $l_u(v) = \lambda_f(u) \cdot v$ be the linear Boolean function restricted to $V$. Consider the following three cases:

- If $\lambda_f(u) \in V^\perp$ then $l_u(v_0) = 0$ for all $v_0 \in V$. So the equation is independent of $v_0$ and becomes $\lambda_f(u) \cdot b = 1$. This has a solution if and only if $\lambda_f(u) \in V_b^\perp$ in which case all of $V$ is a solution so we get $2^{d_2}$ solutions.
- $\lambda_f(u) \in V^\perp \backslash V_b^\perp$ and we get 0 solutions.
- If $\lambda_f(u) \notin V^\perp$ then $l_u$ is surjective into $\mathbb{F}_2$ and takes each of the two values for half the domain. So the number of solutions are $2^{d_2-1}$. □

**Lemma 8.** *Let $U+a$ be an affine subspace, and $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ a quadratic APN permutation. For any set $S \subseteq \mathbb{F}_2^n$, define $N_f(S) = |\{u \in (U+a)^* : \lambda_f(u) \in S\}|$. Then*

$$\Delta_f^{U,V}(a,b) = \begin{cases} 2^{d_2}(2^{d_1} + N_f(V_b^\perp) - N_f(V^\perp \backslash V_b^\perp)) & \text{if } a \notin U \\ 2^{d_2}(2^{d_1} + N_f(V_b^\perp) - N_f(V^\perp \backslash V_b^\perp)) - 2^{d_2} & \text{if } a \in U. \end{cases}$$

*Proof.* We compute

$$\begin{aligned} \Delta_f^{U,V}(a,b) &= 2 \sum_{u \in (U+a)^*} |H_f(u) \cap (V+b)| \\ &= 2(2^{d_2} \cdot N_f(V_b^\perp) + 0 \cdot N_f(V^\perp \backslash V_b^\perp) + 2^{d_2-1} N_f(\mathbb{F}_2^n \backslash V^\perp) \quad \text{By Lemma 7} \\ &= 2^{d_2+1} N_f(V_b^\perp) + 2^{d_2} N_f(\mathbb{F}_2^n \backslash V^\perp). \end{aligned}$$

Observe that

$$N_f(\mathbb{F}_2^n \backslash V^\perp) = |U+a| - N_f(V^\perp) = \begin{cases} 2^{d_1} - (N_f(V_b^\perp) + N_f(V^\perp \backslash V_b^\perp)) & \text{if } a \notin U \\ 2^{d_1} - 1 - (N_f(V_b^\perp) + N_f(V^\perp \backslash V_b^\perp)) & \text{if } a \in U. \end{cases}$$

Inserting this into the expression above gives

$$\begin{aligned} \Delta_f^{U,V}(a,b) &= 2^{d_2+1} N_f(V_b^\perp) + 2^{d_2}(2^{d_1} - (N_f(V_b^\perp) + N_f(V^\perp \backslash V_b^\perp))) \\ &= 2^{d_2}(2^{d_1} + N_f(V_b^\perp) - N_f(V^\perp \backslash V_b^\perp)) \end{aligned}$$

when $a \notin U$ and

$$\begin{aligned} \Delta_f^{U,V}(a,b) &= 2^{d_2+1} N_f(V_b^\perp) + 2^{d_2}(2^{d_1} - 1 - (N_f(V_b^\perp) + N_f(V^\perp \backslash V_b^\perp))) \\ &= 2^{d_2}(2^{d_1} + N_f(V_b^\perp) - N_f(V^\perp \backslash V_b^\perp)) - 2^{d_2} \end{aligned}$$

when $a \in U$. □

Using the Lemma 8, we can finally provide a strong upper bound on $\Delta_f^{d_1,d_2}$.

**Theorem 7.** *Let $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a quadratic APN permutation and $d_1 + d_2 > n-1$. Then,*

$$\Delta_f^{d_1,d_2} \leq 2^{d_1+d_2} + 2^{n-1}.$$

*Proof.* From Lemma 8,

$$\begin{aligned}\Delta_f^{U,V}(a,b) &\leq 2^{d_2}(2^{d_1} + N_f(V_b^\perp) - N_f(V^\perp \backslash V_b^\perp)) \\ &\leq 2^{d_2}(2^{d_1} + N_f(V_b^\perp)) \\ &\leq 2^{d_2}(2^{d_1} + \min(|U+a|, |V_b^\perp|)).\end{aligned}$$

Note that any element counted by $N_f(V_b^\perp)$ is in $U + a$, so $N_f(V_b^\perp) \leq |U + a|$, and $N_f(V_b^\perp) \leq |V_b^\perp|$ since the map $u \mapsto \lambda_f(u)$ is injective as the rows of the DDT of an APN function are all unique [BCJS18, Theorem 2]. Consequently,

$$2^{d_2}(2^{d_1} + \min(|U+a|, |V_b^\perp|)) \leq 2^{d_2}(2^{d_1} + \min(2^{d_1}, 2^{n-d_2-1}))$$

since $V_b^\perp$ is a hyperplane of $V^\perp$ if it is not empty. Finally, we have that,

$$\Delta_f^{U,V}(a,b) \leq 2^{d_2}(2^{d_1} + \min(2^{d_1}, 2^{n-d_2-1})) \leq 2^{d_2}(2^{d_1} + 2^{n-d_2-1}) = 2^{d_1+d_2} + 2^{n-1}. \quad \square$$

The bound given in Theorem 7 shows that the following is the (entrywise) largest SDU a quadratic APN permutation can achieve for e.g. $n = 7$:

$$\begin{pmatrix} & & & & & 192 \\ & & & & 192 & 320 \\ & & & 192 & 320 & 576 \\ & & 192 & 320 & 576 & 1088 \\ & 192 & 320 & 576 & 1088 & 2112 \\ 192 & 320 & 576 & 1088 & 2112 & 4160 \end{pmatrix}$$

Note that this matches the SDU values corresponding to a function with DDT of the form

$$DDT(a,b) = 2(a \cdot b),$$

which corresponds to the binary Hadamard matrix $2 \cdot H_7$.

## 5.3 Connection with Reed-Muller Codes

Since the images of directional derivatives are hyperplanes, they can be defined by inner products. This lets us describe the DDT of such functions with the Reed-Muller code. The following proposition summarizes this well-known connection:

**Proposition 1.** *Let $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a quadratic APN function. Then for each row $R_a$ of the DDT of $f$, there is a weight $2^{n-1}$ codeword $c \in RM(1,n)$ such that $R_a = 2 \cdot c$. Furthermore, if $f$ is a permutation, then each row is given by a unique such $c$, and every codeword of $RM(1,n)$ with 0 as first entry constitutes one row in the DDT. Finally, if the inverse of $f$ is also quadratic, then the columns of the DDT consist of the same codewords of $RM(1,n)$.*

Recall that $RM(1,n)$, is a constant weight code with weight $2^{n-1}$ and constant distance $2^{n-1}$ between all nonzero codewords. These considerations let us compute the SDU of $x^3$ on $\mathbb{F}_2^3$ without the need of a computer.

**Example 1.** We now have enough information to explicitly compute the complete SDU of the well known APN permutation $f(x) = x^3$ over $\mathbb{F}_2^3$. By Theorem 1, we know that

$$\Delta_f \geq \begin{bmatrix} 2 & 4 & 6 \\ 4 & 6 & 10 \\ 6 & 10 & 20 \end{bmatrix}.$$

The inverse of $x^3$, given by $x^5$, is also quadratic, so by Proposition 1 both the rows and the columns consist of non-constant codewords of $RM(1,n)$.

We compute $\Delta_f^{d_1,0}$ for $d_1 = 0,1,2$. The symmetric case follows since the inverse of $f$ is also quadratic. Fix a output space $\{v\}$ of dimension 0. Then the support of $u \mapsto \delta_f(u,v)$ constitutes an affine hyperplane $H_u$, implying that $\Delta_f^{2,0} = 4 \cdot 2 = 8$. Furthermore, $H_u$ is the union of two disjoint hyperplanes of dimension 1. Now Lemma 1 shows that $\Delta_f^{1,0} = 4$.

By Theorem 5 we know that $\Delta_f^{1,1} \geq 8$, and by the APN-ness of $x^3$, a sub-matrix of 4 elements cannot sum to a larger value.

The entries $\Delta_f^{1,2}$ and $\Delta_f^{2,1}$ are now easily computed: Choose any hyperplane $H \subset \mathbb{F}_2^n$. Then there is an $a$ and a row $R_a$ whose support is given exactly by the 4 points of $H$. To find a sub-matrix of the DDT with maximal sum, we may look at any other line in the DDT. But the distance is 4, and in particular, the overlap in nonzero values is exactly 2. Hence, for any input/output pair of hyperplane/plane, we get $2 \cdot 4 + 2 \cdot 2 = 12$.

Finally, we compute $\Delta_f^{2,2}$. Suppose that the hyperplanes $U, V$ determine a sub-matrix of the DDT where no column or row consists only of 2s. Then there are at most four 2s in every row/column, giving a value of 16 which is lower than the lower bound of 20. We conclude that there must be at least one column or row whose entries are all 2. Then by the properties of $RM(1,n)$ we have, apart from the rows/columns of all 2s, there cannot be more than one 2 on the remaining rows/columns. Hence, there are no more than 10 entries valued 2 in the sub-matrix. Hence, $\Delta_f^{2,2}$ hits the lower bound.

We end up with the following SDU:

$$\Delta_f = \begin{bmatrix} 2 & 4 & 8 \\ 4 & 8 & 12 \\ 8 & 12 & 20 \end{bmatrix}.$$

We remark that considering arbitrary input/output subsets (instead of affine subspaces) would not make a difference in the SDU apart from the $d_1 = d_2$ entry where we may reach a value of 22. This is due to $n$ being low. For higher $n$, the situation would be more complex.

# 6 Algorithm

We will now explain our algorithm for computing the SDU, i.e $\Delta_f^{d_1,d_2}$ for all $d_1, d_2$. This algorithm can of course also be used to compute the linear SDU, but we limit our focus to the SDU here. The provided implementation of the algorithm can also compute the linear SDU.

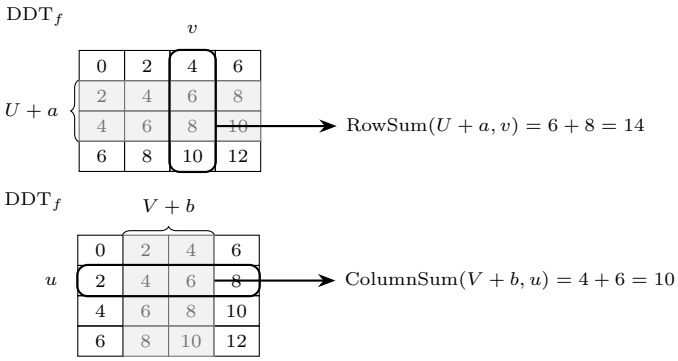
We start by introducing a simple data structure based on prefix sums that will help us both to compute $\Delta_f^{U,V}(a,b)$ faster, and to prune out a lot of subspaces.

We fix the space of input differences $U + a$ and compute the following sum for all $v \in \mathbb{F}_2^n$:

$$\text{RowSum}(U + a, v) = \sum_{u \in U + a} \Delta_f(u,v).$$

Similarly, we fix the space of output differences $V + b$ and compute for all $u \in \mathbb{F}_2^n$:

$$\text{ColumnSum}(V + b, u) = \sum_{v \in V + b} \Delta_f(u,v).$$



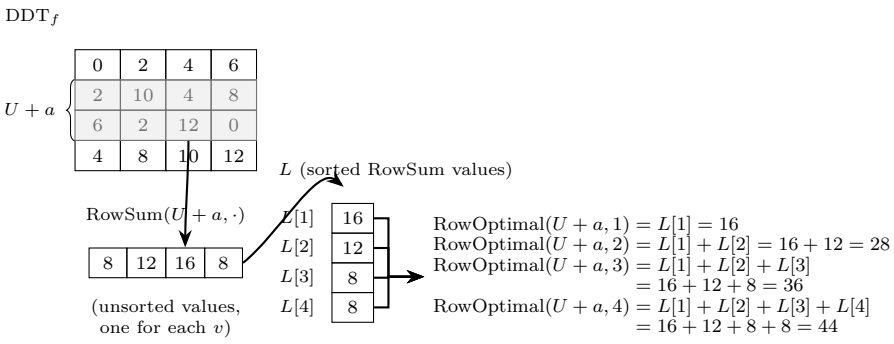

Computing $\Delta_f^{U,V}(a,b)$ naively given that the DDT has been precomputed has time complexity $\mathcal{O}(|U||V|)$. Using RowSum and ColumnSum, it can be computed much faster. To calculate $\Delta_f^{U,V}(a,b)$ we simply calculate the sum $\sum_{u\in U+a}\text{ColumnSum}(V+b,u)$ if $|U|<|V|$, otherwise we calculate the sum $\sum_{v\in V+b}\text{RowSum}(U+a,v)$. This allows us to compute $\Delta_f^{U,V}(a,b)$ in time $\mathcal{O}(\min(|U|,|V|))$.

Using RowSum and ColumnSum we define what we can call a sorted prefix sum to determine what is the highest possible value that the SDT can be if one of the subspaces $U+a$ or $V+b$ are fixed; if we have already found a subspace with a higher value, then we can discard it. For each $U+a$ we define RowOptimal$(U+a,t)$ for any $t\leq 2^n$ as follows

- Let $L=[\text{RowSum}(U+a,v):v\in\mathbb{F}_2^n]$
- Sort $L$ in decreasing order.
- Let $\text{RowOptimal}(U+a,t)=\sum_{j=1}^{t}L[j]$

Analogously we define ColumnOptimal$(V+b,t)$

- Let $L=[\text{ColumnSum}(V+b,u):u\in\mathbb{F}_2^n]$
- Sort $L$ in decreasing order.
- Let $\text{ColumnOptimal}(V+b,t)=\sum_{j=1}^{t}L[j]$




Our goal is to avoid having to consider all pairs of affine subspaces, we use the following observation. RowOptimal$(U+a,2^{d_2})$ is an upper bound on the value of $\Delta_f^{U,V}(a,b)$ for any $V+b\in\mathcal{A}(n,d_2)$. This is because when $U+a$ is fixed the best possible $V+b$ of a given dimension $d_2$ would be taking the elements representing the largest $2^{d_2}$ elements of RowSum$(U+a,\cdot)$. These $2^{d_2}$ elements are of course unlikely to form a subspace, but any other $d_2$ dimensional subspace cannot beat this value. This means that if we have already

found a pair of affine subspaces such that $\Delta_f^{U,V}(a,b) \geq \text{RowOptimal}(U'+a', 2^{d_2})$. Then we can immediately discard all the pairs $(U'+a', V'+b')$ for all $V'+b' \in \mathcal{A}(n,d_2)$.

**Algorithm 1:** SDU$(f, d_1, d_2)$

**Input:** Function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$, dimensions $d_1, d_2$
**Output:** $\Delta_f^{d_1,d_2}$

Let $U_{order}$ be the set of all $d_1$ dimensional affine subspaces sorted by their value on RowOptimal$(U+a, 2^{d_2})$ in descending order.;
Let $V_{order}$ be the set of all $d_2$ dimensional affine subspaces sorted by their value on ColumnOptimal$(V+b, 2^{d_1})$ in descending order.;
$M \leftarrow 0$;
**for** $U+a \in U_{order}$ **do**
  **if** RowOptimal$(U+a, 2^{d_2}) \leq M$ **then**
    **break loop**;
  **for** $V+b \in V_{order}$ **do**
    **if** ColumnOptimal$(V+b, 2^{d_1}) \leq M$ **then**
      **break inner loop**;
    $M \leftarrow \max(M, \Delta_f^{U,V}(a,b))$;
**return** $M$;

### 6.1 Correctness

Given $d_1$ and $d_2$ we want to compute $\Delta_f^{d_1,d_2}$ so we must find the elements $U+a \in \mathcal{A}(n,d_1)$ and $V+b \in \mathcal{A}(n,d_2)$ with maximal value of $\Delta_f^{U,V}(a,b)$. Assume $\Delta_f^{U,V}(a,b)$ obtains the maximal value. Then at any step in the algorithm before considering $U+a, V+b$, we know that RowOptimal$(U+a, 2^{d_2}) \geq M$ and ColumnOptimal$(V+b, 2^{d_1}) \geq M$ at this point in the algorithm. This means that the algorithm will consider $\Delta_f^{U,V}(a,b)$ at some point.

### 6.2 Complexity

The runtime of the algorithm depends heavily on how successful the preprocessing removes subspaces (how early the loop breaks), which again depends heavily on the specific function as we can see in our runtime results.

To construct $U_{order}$ we must first compute RowSum$(U+a)$ for all $U+a \in \mathcal{A}(n,d_1)$, each $U+a$ can be done in $\mathcal{O}(2^n)$ time. Then constructing RowOptimal for this $U+a$ is simply sorting and summing over the elements, done in $\mathcal{O}(2^n \log(2^n)) = \mathcal{O}(n2^n)$. So we see that the preprocessing step of constructing RowOptimal can be done in $\mathcal{O}(|\mathcal{A}(n,d_1)|n2^n)$, the same is true for computing ColumnOptimal; thus the preprocessing is done in time $\mathcal{O}(n2^n(|\mathcal{A}(n,d_1)| + |\mathcal{A}(n,d_2)|))$.

After that the worst case is when we have to iterate through every subspace (so the loops do not terminate before considering more or less all the subspaces). Then we need to consider all pairs of subspaces, and then do $2^{\min(d_1,d_2)}$ work to compute $\Delta_f^{U,V}(a,b)$. The total complexity is $\mathcal{O}(2^{\min(d_1,d_2)}|\mathcal{A}(n,d_1)||\mathcal{A}(n,d_2)|)$.

## 7 SDU Optimal DDT Indicators

A DDT indicator is a Boolean function $\gamma_f(x,y)$ taking on the value 0 if and only if $\delta_f(x,y) = 0$ (or when $x = 0$) and 1 otherwise. A natural question is to determine which

DDT indicators achieve optimal subspace differential uniformity, i.e. the indicators for which the table with entries

$$M(x, y) = 2\gamma_f(x, y)$$

obtain the best (lowest) SDU values.

The challenge of finding S-boxes having the lowest SDU values is nontrivial as one APN permutation can be better than another in this regard. One approach to possibly improving lower bounds and understand the relationship between the SDU table values for a single function, is to determine designs of DDT tables which provides the lowest possible SDU values overall. Although not all such DDT matrices may actually be possible for a permutation, it still provides us with fundamental lower and upper bounds that may guide us. Computing the SDU of a function reduces to adding certain elements in the DDT table of the function. This motivates the matrix combinatorial question: Finding a DDT-like matrix whose SDU table has the lowest (or highest) achievable values in the class of APN valid DDT matrices. A natural place to look is in the toolbox of combinatorial designs. One type of matrix that would fit our restrictions, is the *binary* Hadamard matrix $H_{2^n}$, with entries

$$2 \cdot H_{2^n}(u, v) = 2 \cdot (u \cdot v).$$

which has rows and columns constituting the Hadamard code, which is a constant weight code for which all codewords have equal distance to each other. The distance properties should in part explain the low SDU of the matrix, but even though the Hadamard matrix seems to provide a low values, computations indicate that it is not optimal. Below we compare our general lower bound on the SDUs together with the ones achieved with the Hadamard matrix, juxtaposed with the SDUs for Niho and Gold APN power functions in $\mathbb{F}_{2^7}$.

**Lower Bound**

$$\begin{pmatrix} 2 & 4 & 6 & 10 & 18 & 34 & 66 \\ 4 & 6 & 10 & 18 & 34 & 66 & 130 \\ 6 & 10 & 18 & 34 & 66 & 130 & 260 \\ 10 & 18 & 34 & 66 & 130 & 260 & 518 \\ 18 & 34 & 66 & 130 & 260 & 518 & 1034 \\ 34 & 66 & 130 & 260 & 518 & 1034 & 2066 \\ 66 & 130 & 260 & 518 & 1034 & 2066 & 4130 \end{pmatrix}$$

**Hadamard**:

$$\begin{pmatrix} 2 & 4 & 8 & 16 & 32 & 64 & 128 \\ 4 & 8 & 16 & 32 & 64 & 128 & 192 \\ 8 & 16 & 32 & 64 & 128 & 192 & 320 \\ 16 & 32 & 64 & 128 & 192 & 320 & 576 \\ 32 & 64 & 128 & 192 & 320 & 576 & 1088 \\ 64 & 128 & 192 & 320 & 576 & 1088 & 2112 \\ 128 & 192 & 320 & 576 & 1088 & 2112 & 4160 \end{pmatrix}$$

**Niho** $x^{2^3+2^5-1}$

$$\begin{pmatrix} 2 & 4 & 8 & 16 & 30 & 44 & 72 \\ 4 & 8 & 16 & 32 & 54 & 88 & 144 \\ 8 & 16 & 32 & 60 & 98 & 168 & 288 \\ 16 & 32 & 58 & 104 & 180 & 320 & 576 \\ 30 & 54 & 98 & 180 & 328 & 592 & 1088 \\ 44 & 88 & 168 & 320 & 592 & 1104 & 2112 \\ 72 & 144 & 288 & 576 & 1088 & 2112 & 4160 \end{pmatrix}$$

**Gold** $x^3$

$$\begin{pmatrix} 2 & 4 & 8 & 16 & 32 & 64 & 128 \\ 4 & 8 & 16 & 32 & 64 & 128 & 192 \\ 8 & 16 & 32 & 64 & 128 & 192 & 320 \\ 16 & 32 & 60 & 112 & 192 & 320 & 576 \\ 30 & 56 & 100 & 184 & 320 & 576 & 1088 \\ 48 & 94 & 176 & 320 & 576 & 1088 & 2112 \\ 84 & 156 & 296 & 576 & 1088 & 2112 & 4160 \end{pmatrix}$$

We remark that the only observed differences between the Hadamard SDU and the Gold SDU occur on and below the diagonal, where the values for $x^3$ are slightly lower. Moreover, $x^3$ obtains with a few exceptions the worst case upper-bounds for quadratic

APNs (see Section 5.2), i.e. the entries below the anti-diagonal. The Niho APN function, on the other hand, is closer to our lower bound. In general, a bivariate Boolean function, $f(x,y) : \mathbb{F}_2^{2n} \to \mathbb{F}_2$ and

$$M(u,v) = 2\mathrm{Tr}(u^r v^s)$$

seems to be a good candidate for a DDT indicator whose table achieves optimal SDU values. Thus, (a) determining a DDT matrix that attains the minimum possible SDU, and (b) determining whether any APN permutation can have such a DDT, are left as two open problems.

# 8 Practical Results

Below we provide run-times (in seconds) of the algorithm presented in Section 6 for computing the full SDU of $x^{-1}$ and $x^3$ on a MacBook Pro with M2 chip with 12 CPU cores and 32 GBs of RAM with a Python implementation using the module *Numba* for near C-performance and parallelization.

Computing the SDU for the AES S-box took 4553 seconds, including the pre-processing step. In contrast, computing the linear SDU of the same S-box took only 151 seconds on the same computer. Even though the elimination of subspaces in the algorithm results in a massive speedup, there are certain input/output dimensions for which we cannot eliminate any subspaces, and must compute the value $\Delta_f^{U,V}(a,b)$ for a very high number of subspaces. The significance of this becomes clear when considering the function $x^3$ whose SDU was computed in only 320 seconds.

We conclude that the run-time depends heavily on the specific function, which determines the success of the pre-processing in eliminating subspaces. Consult Table 1 and Table 2 for the exact run-times.

**Table 1:** Average run-times to compute subspace differential uniformity $\Delta_f^{d_1,d_2}$ for all $d_1, d_2$ for $x^3$ and $x^{-1}$ across different values of $n$.

| $n$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| $x^3$ | 0.004 s | 0.006 s | 0.01 s | 0.02 s | 0.2 s | 11 s | 320 s |
| $x^{-1}$ | 0.001 s | 0.003 s | 0.009 s | 0.04s | 0.4 s | 72 s | 4500 s |

**Table 2:** Average run-times to compute linear subspace differential uniformity $\widehat{\Delta}_f^{d_1,d_2}$ for all $d_1, d_2$ for $x^3$ and $x^{-1}$ across different values of $n$.

| $n$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| $x^3$ | 0.0003 s | 0.001 s | 0.002 s | 0.008 s | 0.05 s | 1.8 s | 43 s |
| $x^{-1}$ | 0.0003 s | 0.001 s | 0.003 s | 0.009 s | 0.07 s | 2 s | 151 s |

## 8.1 Results for Some Popular S-boxes

We have computed the subspace differential uniformity for several popular S-boxes from literature. The set of 16 equivalence classes of optimal 4-bit S-boxes $G_0$ through $G_{15}$ taken from [LP07] falls into 8 SDU classes. We have included both linear SDU and SDU for Dillon's APN and the Streebog S-box for comparison. Note that dimension 0 is not included for the linear SDU, while it is included in the affine case.

**Dillon's APN [BDMW10] $n = 6$**

**Linear SDU**

$$\begin{pmatrix} 2 & 6 & 14 & 28 & 40 \\ 6 & 18 & 36 & 72 & 112 \\ 14 & 36 & 80 & 144 & 256 \\ 28 & 72 & 144 & 288 & 544 \\ 40 & 112 & 256 & 544 & 1024 \end{pmatrix}$$

**SDU**

$$\begin{pmatrix} 2 & 4 & 8 & 16 & 32 & 64 \\ 4 & 8 & 16 & 32 & 64 & 104 \\ 8 & 16 & 32 & 64 & 112 & 176 \\ 16 & 32 & 64 & 112 & 192 & 320 \\ 32 & 64 & 112 & 192 & 320 & 576 \\ 64 & 104 & 176 & 320 & 576 & 1088 \end{pmatrix}$$

**The $\chi$-function**

$n = 5$:

$$\begin{pmatrix} 0.25 & 0.50 & 1 & 1 & 1 \\ 0.19 & 0.38 & 0.75 & 1 & 1 \\ 0.14 & 0.28 & 0.50 & 0.75 & 1 \\ 0.09 & 0.19 & 0.31 & 0.50 & 0.75 \\ 0.06 & 0.11 & 0.20 & 0.38 & 0.63 \end{pmatrix}$$

$n = 7$

$$\begin{pmatrix} 0.25 & 0.50 & 1 & 1 & 1 & 1 & 1 \\ 0.19 & 0.38 & 0.75 & 1 & 1 & 1 & 1 \\ 0.14 & 0.28 & 0.50 & 0.75 & 1 & 1 & 1 \\ 0.08 & 0.16 & 0.31 & 0.50 & 0.75 & 1 & 1 \\ 0.05 & 0.10 & 0.18 & 0.31 & 0.50 & 0.75 & 1 \\ 0.03 & 0.05 & 0.10 & 0.19 & 0.31 & 0.50 & 0.75 \\ 0.02 & 0.03 & 0.06 & 0.11 & 0.20 & 0.38 & 0.63 \end{pmatrix}$$

**Figure 1:** SDP-table of $\chi$

**APN functions in $\mathbb{F}_2^7$**

Gold: $x^3$:

$$\begin{pmatrix} 2 & 4 & 8 & 16 & 32 & 64 & 128 \\ 4 & 8 & 16 & 32 & 64 & 128 & 192 \\ 8 & 16 & 32 & 64 & 128 & 192 & 320 \\ 16 & 32 & 60 & 112 & 192 & 320 & 576 \\ 30 & 56 & 100 & 184 & 320 & 576 & 1088 \\ 48 & 94 & 176 & 320 & 576 & 1088 & 2112 \\ 84 & 156 & 296 & 576 & 1088 & 2112 & 4160 \end{pmatrix}$$

Kasami:

$$\begin{pmatrix} 2 & 4 & 8 & 16 & 30 & 44 & 72 \\ 4 & 8 & 16 & 32 & 54 & 88 & 144 \\ 8 & 16 & 32 & 58 & 98 & 168 & 288 \\ 16 & 32 & 60 & 104 & 180 & 320 & 576 \\ 30 & 54 & 98 & 180 & 328 & 592 & 1088 \\ 44 & 88 & 168 & 320 & 592 & 1104 & 2112 \\ 72 & 144 & 288 & 576 & 1088 & 2112 & 4160 \end{pmatrix}$$

Welch:

$$\begin{pmatrix} 2 & 4 & 8 & 16 & 30 & 44 & 72 \\ 4 & 8 & 16 & 32 & 54 & 88 & 144 \\ 8 & 16 & 32 & 58 & 100 & 168 & 288 \\ 16 & 32 & 58 & 104 & 184 & 320 & 576 \\ 30 & 54 & 100 & 184 & 320 & 592 & 1088 \\ 44 & 88 & 168 & 320 & 592 & 1104 & 2112 \\ 72 & 144 & 288 & 576 & 1088 & 2112 & 4160 \end{pmatrix}$$

Niho:

$$\begin{pmatrix} 2 & 4 & 8 & 16 & 30 & 44 & 72 \\ 4 & 8 & 16 & 32 & 54 & 88 & 144 \\ 8 & 16 & 32 & 60 & 98 & 168 & 288 \\ 16 & 32 & 58 & 104 & 180 & 320 & 576 \\ 30 & 54 & 98 & 180 & 328 & 592 & 1088 \\ 44 & 88 & 168 & 320 & 592 & 1104 & 2112 \\ 72 & 144 & 288 & 576 & 1088 & 2112 & 4160 \end{pmatrix}$$

Inverse:

$$\begin{pmatrix} 2 & 4 & 8 & 16 & 32 & 64 & 128 \\ 4 & 8 & 16 & 32 & 64 & 128 & 192 \\ 8 & 16 & 32 & 64 & 128 & 192 & 320 \\ 16 & 32 & 60 & 112 & 192 & 320 & 576 \\ 30 & 56 & 100 & 184 & 320 & 576 & 1088 \\ 48 & 94 & 176 & 320 & 576 & 1088 & 2112 \\ 84 & 156 & 296 & 576 & 1088 & 2112 & 4160 \end{pmatrix}$$

Entrywise average over 100 randomly sampled S-boxes:

$$\begin{pmatrix} 11 & 17 & 25 & 34 & 49 & 69 & 97 \\ 17 & 27 & 40 & 59 & 85 & 125 & 189 \\ 25 & 40 & 62 & 94 & 144 & 224 & 352 \\ 34 & 59 & 94 & 152 & 248 & 406 & 671 \\ 49 & 85 & 144 & 247 & 430 & 744 & 1284 \\ 69 & 124 & 224 & 406 & 743 & 1359 & 2450 \\ 96 & 184 & 351 & 670 & 1283 & 2444 & 4642 \end{pmatrix}$$

**4-uniform functions in $\mathbb{F}_2^6$**

Gold: $x^5$:

$$\begin{pmatrix} 4 & 8 & 16 & 32 & 64 & 64 \\ 8 & 16 & 32 & 48 & 80 & 128 \\ 16 & 24 & 48 & 80 & 128 & 192 \\ 28 & 40 & 64 & 112 & 192 & 320 \\ 40 & 64 & 112 & 176 & 320 & 576 \\ 48 & 96 & 176 & 320 & 576 & 1088 \end{pmatrix}$$

Modified version of Dillon:

$$\begin{pmatrix} 4 & 8 & 12 & 20 & 36 & 64 \\ 8 & 14 & 22 & 38 & 64 & 104 \\ 12 & 22 & 40 & 72 & 112 & 184 \\ 20 & 38 & 68 & 114 & 200 & 340 \\ 36 & 66 & 112 & 196 & 348 & 612 \\ 64 & 104 & 184 & 340 & 612 & 1124 \end{pmatrix}$$

**SDU Streebog/Kuznyechik [Per19] $n = 8$**

**Linear SDU**

$$\begin{pmatrix} 8 & 18 & 30 & 48 & 74 & 112 & 176 \\ 18 & 42 & 68 & 110 & 180 & 298 & 500 \\ 30 & 78 & 138 & 214 & 368 & 644 & 1116 \\ 60 & 138 & 262 & 450 & 700 & 1244 & 2268 \\ 76 & 186 & 378 & 700 & 1302 & 2412 & 4552 \\ 110 & 296 & 642 & 1238 & 2416 & 4676 & 8816 \\ 176 & 488 & 1076 & 2228 & 4432 & 8724 & 16912 \end{pmatrix}$$

**SDU**

$$\begin{pmatrix} 8 & 14 & 22 & 32 & 46 & 72 & 114 & 176 \\ 16 & 28 & 42 & 66 & 98 & 140 & 216 & 344 \\ 24 & 40 & 62 & 98 & 152 & 244 & 390 & 660 \\ 34 & 62 & 104 & 176 & 258 & 430 & 714 & 1240 \\ 48 & 92 & 184 & 308 & 450 & 748 & 1326 & 2380 \\ 74 & 136 & 242 & 438 & 764 & 1364 & 2494 & 4592 \\ 116 & 214 & 398 & 718 & 1322 & 2492 & 4752 & 8904 \\ 176 & 336 & 640 & 1216 & 2352 & 4596 & 8872 & 17168 \end{pmatrix}$$

**SDU SKIPJACK**, [ski98] $n = 8$

$$\begin{pmatrix} 12 & 18 & 26 & 36 & 52 & 78 & 124 & 182 \\ 20 & 30 & 46 & 66 & 94 & 142 & 220 & 348 \\ 24 & 44 & 66 & 100 & 156 & 244 & 386 & 660 \\ 36 & 64 & 100 & 160 & 262 & 430 & 714 & 1268 \\ 50 & 94 & 156 & 256 & 432 & 756 & 1342 & 2428 \\ 80 & 144 & 244 & 426 & 748 & 1360 & 2508 & 4640 \\ 118 & 220 & 392 & 738 & 1342 & 2522 & 4702 & 8976 \\ 180 & 346 & 666 & 1262 & 2412 & 4602 & 8936 & 17224 \end{pmatrix}$$

**SDU AES S-box**, $n = 8$

$$\begin{pmatrix} 4 & 6 & 10 & 18 & 34 & 52 & 88 & 144 \\ 6 & 12 & 20 & 36 & 64 & 104 & 172 & 288 \\ 10 & 20 & 38 & 72 & 128 & 208 & 332 & 572 \\ 18 & 36 & 72 & 140 & 256 & 392 & 640 & 1136 \\ 34 & 64 & 128 & 256 & 480 & 704 & 1176 & 2208 \\ 52 & 104 & 208 & 392 & 704 & 1216 & 2242 & 4292 \\ 88 & 172 & 332 & 640 & 1176 & 2242 & 4338 & 8416 \\ 144 & 288 & 572 & 1136 & 2208 & 4292 & 8416 & 16640 \end{pmatrix}$$

We also computed the SDUs of all 4-bit S-boxes up to affine equivalence. Every pair of S-boxes with the same SDU table are considered to belong to the same class in terms of SDU. We summarize the 8 classes that our computations provided. We round every probability off to 2 decimals.

**G0, G1, G2, G8**

**SDU**

$$\begin{pmatrix} 4 & 8 & 16 & 16 \\ 8 & 16 & 24 & 32 \\ 16 & 24 & 32 & 48 \\ 16 & 32 & 48 & 80 \end{pmatrix}$$

**SDP**

$$\begin{pmatrix} 0.25 & 0.50 & 1 & 1 \\ 0.25 & 0.50 & 0.75 & 1 \\ 0.25 & 0.38 & 0.50 & 0.75 \\ 0.13 & 0.25 & 0.38 & 0.63 \end{pmatrix}$$

**G3**

**SDU**

$$\begin{pmatrix} 4 & 6 & 10 & 12 \\ 6 & 12 & 16 & 24 \\ 10 & 16 & 26 & 48 \\ 12 & 24 & 48 & 80 \end{pmatrix}$$

**SDP**

$$\begin{pmatrix} 0.25 & 0.38 & 0.63 & 0.75 \\ 0.19 & 0.38 & 0.50 & 0.75 \\ 0.16 & 0.25 & 0.41 & 0.75 \\ 0.09 & 0.19 & 0.38 & 0.63 \end{pmatrix}$$

**G4, G7, G12, G13**

**SDU**

$$\begin{pmatrix} 4 & 8 & 10 & 12 \\ 8 & 12 & 18 & 24 \\ 10 & 18 & 32 & 48 \\ 12 & 24 & 48 & 80 \end{pmatrix}$$

**SDP**

$$\begin{pmatrix} 0.25 & 0.50 & 0.63 & 0.75 \\ 0.25 & 0.38 & 0.56 & 0.75 \\ 0.16 & 0.28 & 0.50 & 0.75 \\ 0.09 & 0.19 & 0.38 & 0.63 \end{pmatrix}$$

**G5**

**SDU**

$$\begin{pmatrix} 4 & 6 & 10 & 12 \\ 6 & 12 & 18 & 24 \\ 10 & 18 & 32 & 48 \\ 12 & 24 & 48 & 80 \end{pmatrix}$$

**SDP**

$$\begin{pmatrix} 0.25 & 0.38 & 0.63 & 0.75 \\ 0.19 & 0.38 & 0.56 & 0.75 \\ 0.16 & 0.28 & 0.50 & 0.75 \\ 0.09 & 0.19 & 0.38 & 0.63 \end{pmatrix}$$

**G6, G11**

**SDU**

$$\begin{pmatrix} 4 & 8 & 10 & 12 \\ 8 & 14 & 18 & 24 \\ 10 & 18 & 32 & 48 \\ 12 & 24 & 48 & 80 \end{pmatrix}$$

**SDP**

$$\begin{pmatrix} 0.25 & 0.50 & 0.63 & 0.75 \\ 0.25 & 0.44 & 0.56 & 0.75 \\ 0.16 & 0.28 & 0.50 & 0.75 \\ 0.09 & 0.19 & 0.38 & 0.63 \end{pmatrix}$$

**G9, G10**

**SDU**

$$\begin{pmatrix} 4 & 8 & 12 & 16 \\ 8 & 14 & 20 & 32 \\ 12 & 20 & 32 & 48 \\ 16 & 32 & 48 & 80 \end{pmatrix}$$

**SDP**

$$\begin{pmatrix} 0.25 & 0.50 & 0.75 & 1 \\ 0.25 & 0.44 & 0.63 & 1 \\ 0.19 & 0.31 & 0.50 & 0.75 \\ 0.13 & 0.25 & 0.38 & 0.63 \end{pmatrix}$$

**G14**

**SDU**

$$\begin{pmatrix} 4 & 8 & 12 & 16 \\ 8 & 12 & 20 & 32 \\ 12 & 24 & 32 & 48 \\ 16 & 32 & 48 & 80 \end{pmatrix}$$

**SDP**

$$\begin{pmatrix} 0.25 & 0.50 & 0.75 & 1 \\ 0.25 & 0.38 & 0.63 & 1 \\ 0.19 & 0.38 & 0.50 & 0.75 \\ 0.13 & 0.25 & 0.38 & 0.63 \end{pmatrix}$$

**G15**

**SDU**

$$\begin{pmatrix} 4 & 8 & 12 & 16 \\ 8 & 12 & 24 & 32 \\ 12 & 20 & 32 & 48 \\ 16 & 32 & 48 & 80 \end{pmatrix}$$

**SDP**

$$\begin{pmatrix} 0.25 & 0.50 & 0.75 & 1 \\ 0.25 & 0.38 & 0.75 & 1 \\ 0.19 & 0.31 & 0.50 & 0.75 \\ 0.13 & 0.25 & 0.38 & 0.63 \end{pmatrix}$$

## 8.2 Remarks on SDU Tables

We provide interpretations of some of the numbers appearing in the SDU tables above.

First, we note that since $\chi$ is affine equivalent to $x^3$ over $\mathbb{F}_2^3$, its SDU is just the one computed in Example 1 so we do not include it here. By design $\chi$ has many probability 1 subspace trails for larger $n$, which is observed very clearly above the diagonal in the corresponding SDU in Fig. 1.

A perhaps more interesting observation about the SDU of $\chi$ is the diagonal values. In particular, for dimension $1 < d < n-1$, we observe that there are subspaces $a+U, b+V$ for which any difference in $a+U$ has output difference in $b+V$ with probability $1/2$, and slightly higher when $d = n-1$.

We also remark that among the Kasami APN permutations, given by exponents $2^{2t} - 2^t + 1$ for $\gcd(t, n) = 1$, we tried $t = 1, 2, 3, 4, 5, 6$. This yielded the same SDU for

$t = 1$, $t = 2, 5$ and $t = 3, 4$. We listed the latter one in the table above, as it provided the lowest values. Finally, observe that the Niho function has SDU table which is the transpose of the Kasami one. This is no coincidence, as one readily shows that for $t = 4$, the Kasami function is just the Welch function.

The comparison of the two 4-uniform functions illustrates a crucial point: the differential uniformity of a function does not reveal every aspect of the resistance an S-box provides to cryptanalysis. In our example, we consider a 4-uniform Gold-function and a modified version of the Dillon permutation, where we swapped the output of 0 and 1. While the differential spectrum of $x^5$ contained the value 4 a total of 1008 times, the value only appeared 54 times in the modified Dillon case. While intuitively one would expect that having a differential spectrum with fewer 4s would lead to lower values in the SDU, to our surprise, the Gold function has lower values on the latter half of the diagonal. One conclusion to draw from this is that differential uniformity does not tell the whole story about an S-box.

We observe that among the optimal 4-bit S-boxes, $G3$ has the lowest (sometimes shared) SDU in every input/output dimension. In particular, it performs significantly better for input/output dimensions $d_1 = d_2 = n - 2$.

### 8.3 SDU and Block Ciphers

Among the 16 classes of optimal 4-bit S-boxes presented on the previous page, 8 of them exhibit deviating behavior in terms of their subspace differential uniformity table, in the sense that they map input differences from one subspace to output differences in another subspace with probability 1. Thus it is of interest to measure whether there are measurable differences in practice when used as S-boxes in an otherwise strong block cipher construction.

To test this, consider a basic conservative block cipher construction using a round function with an S-box layer consisting of 16 copies of one of the 4-bit S-boxes $G_0, G_1, \ldots, G_{15}$ and a $16 \times 16$ binary layer $L$ derived from a $(16, 8, 8)$ linear code, which provides optimal diffusion over $\mathbb{F}_2$. We add random round keys before, in-between and at the end.

To simplify, we pick the hyperplanes $U \subset \mathbb{F}_2^4$ and consider the distribution of input/output differences between cosets of the same product space $U^{\otimes 16}$ over three full rounds. If the block cipher struggles to spread fixed coset differences uniformly, this can in principle be distinguished by measuring the number of output cosets activated in total.

In our experiments, for each of the S-boxes $G_i$, we evaluate $2^{16}$ random differences with exactly one active nibble (e.g. the first) and compare the number of unique cosets we hit after 3 full rounds. There are $2^{16}$ cosets and we have $2^{16}$ output differences, thus the expected number of cosets these output differences should hit is $2^{16}(1 - (1 - 1/2^{16})^{16}) \approx 41427$ and with a standard deviation of 80. For each S-box, we go through each of the 15 hyperplanes $U \subset \mathbb{F}_2^4$ and determine the hyperplane for which the set of output differences are contained in fewest cosets of $U^{\otimes 16}$. In the supplied code referenced in Section 1.1, we print a) the minimal unique set of cosets (`minuq`), b) the number of standard deviations from expected, and c) the highest number of output differences observed in a single coset.

The eight 4-bit S-boxes with degenerate SDU tables, $G_0, G_1, G_2, G_8, G_9, G_{10}, G_{14}$ and $G_{15}$ in the tables displayed on previous page, are immediately detected in the experiment by hitting up to 30 standard deviations below the expected number of output cosets. The remaining 8 classes are consistently between 3 and 5 in the experiments.

## 9 Conclusion

In this paper we have introduced the subspace DDT and established bounds for subspace differential uniformity in the generic case, as well as improved bounds for quadratic

permutations. We have presented an efficient algorithm for computing the SDU together with practical results. Several open questions remain unanswered. In particular, it is not yet clear what functions achieve optimal SDU, nor which DDT design gives the lowest possible SDU values. Finally there is the question of how these new insights may be leveraged for cryptanalysis. We leave these open questions for future research.

## Acknowledgments

We thank the anonymous referees for their valuable feedback, and Nikolay Kaleyski for helpful discussions along the way. Finally, we are grateful to the organizers of the BFA 2025 workshop for organizing such an interesting event.

## References

[BCJS18] Christina Boura, Anne Canteaut, Jérémy Jean, and Valentin Suder. Two notions of differential equivalence on sboxes. Cryptology ePrint Archive, Paper 2018/617, 2018. URL: https://eprint.iacr.org/2018/617, doi:10.1007/s10623-018-0496-z.

[BDMW10] Keith A Browning, John F Dillon, MT McQuistan, and Alan J Wolfe. An APN permutation in dimension six. *Finite Fields: theory and applications*, 518:33–42, 2010.

[Car21a] Claude Carlet. *Boolean Functions for Cryptography and Coding Theory*. Cambridge University Press, Cambridge, 2021.

[Car21b] Claude Carlet. On the properties of the Boolean functions associated to the differential spectrum of general APN functions and their consequences. *IEEE Transactions on Information Theory*, 67(10):6926–6939, 2021.

[CP26] Ingo Czerwinski and Alexander Pott. On large sidon sets, 2026. URL: https://arxiv.org/abs/2411.12911, arXiv:2411.12911.

[DH18] Orr Dunkelman and Senyang Huang. Reconstructing an S-box from its Difference Distribution Table. *Cryptology ePrint Archive*, 2018.

[FY25] Bryce Frederickson and Liana Yepremyan. Vector space Ramsey numbers and weakly Sidorenko affine configurations. *The Quarterly Journal of Mathematics*, 76(1):77–94, 2025.

[GJN+16] Jian Guo, Jérémy Jean, Ivica Nikolic, Kexin Qiao, Yu Sasaki, and Siang Meng Sim. Invariant Subspace Attack Against Midori64 and The Resistance Criteria for S-box Designs. *IACR Transactions on Symmetric Cryptology*, 2016(1):33–56, Dec. 2016. doi:10.13154/tosc.v2016.i1.33-56.

[GRR17] Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom. Subspace Trail Cryptanalysis and its Applications to AES. *IACR Transactions on Symmetric Cryptology*, 2016(2):192–225, Feb. 2017. URL: https://tosc.iacr.org/index.php/ToSC/article/view/571, doi:10.13154/tosc.v2016.i2.192-225.

[KN25] Benedek Kovács and Zoltán Lóránt Nagy. Avoiding intersections of given size in finite affine spaces AG (n, 2). *Journal of Combinatorial Theory, Series A*, 209:105959, 2025.

[LAAZ11] Gregor Leander, Mohamed Ahmed Abdelraheem, Hoda AlKhzaimi, and Erik Zenner. A Cryptanalysis of PRINTcipher: The Invariant Subspace Attack. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, pages 206–221, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[LMR15] Gregor Leander, Brice Minaud, and Sondre Rønjom. A Generic Approach to Invariant Subspace Attacks: Cryptanalysis of Robin, iSCREAM and Zorro. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, pages 254–283, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[LP07] G. Leander and A. Poschmann. On the Classification of 4 Bit S-Boxes. *Arithmetic of Finite Fields*, pages 159–176, 2007.

[LTW18] Gregor Leander, Cihangir Tezcan, and Friedrich Wiemer. Searching for Subspace Trails and Truncated Differentials. *IACR Transactions on Symmetric Cryptology*, 2018(1):74–100, Mar. 2018. URL: https://tosc.iacr.org/index.php/ToSC/article/view/845, doi:10.13154/tosc.v2018.i1.74-100.

[Per19] Léo Perrin. Partitions in the S-Box of Streebog and Kuznyechik. *IACR Transactions on Symmetric Cryptology*, 2019(1):302–329, Mar. 2019. URL: https://tosc.iacr.org/index.php/ToSC/article/view/7405, doi:10.13154/tosc.v2019.i1.302-329.

[ski98] SKIPJACK and KEA Algorithm Specifications. Technical report, National Institute of Standards and Technology, May 1998. https://csrc.nist.gov/presentations/1998/skipjack-and-kea-algorithm-specifications.