2016

Cameron-Liebler line classes and partial difference sets

Uthaipon Tantipongipat
University of Richmond

CAMERON-LIEBLER LINE CLASSES AND PARTIAL DIFFERENCE SETS

UTHAIPON TANTIPONGPIPAT

Honors Thesis

Department of Mathematics and Computer Science

University of Richmond

29 April 2016

Advisor: Dr. James A Davis

The signatures below, by the thesis advisor, the departmental reader, and the honors coordinator for mathematics, certify that this thesis, prepared by Uthaipon Tantipongpipat, has been approved, as to style and content.

(Dr. James Davis, thesis advisor)

(Dr. Heather Russell, departmental reader)

(Dr. Van Nall, honors coordinator)

Abstract. The work consists of three parts. The first is a study of Cameron-Liebler line classes which receive much attention recently. We studied a new construction of infinite family of Cameron-Liebler line classes presented in the paper by Tao Feng, Koji Momihara, and Qing Xiang (first introduced in 2014), and summarized our attempts to generalize this construction to discover any new Cameron-Liebler line classes or partial difference sets (PDSs) resulting from the Cameron-Liebler line classes. The second is our approach to finding PDS in non-elementary abelian groups. Our attempt eventually led to the same general construction of PDS presented in John Polhill's PhD Thesis. The third presents a proof that any PDS in $\mathbb{Z}_p^3$ for $p \equiv 3 \mod 4$ must be trivial, and any PDS in $\mathbb{Z}_p^3$ for $p \equiv 1 \mod 4$ must be a Paley-type PDS. We also show that finding all PDSs in $\mathbb{Z}_p^3$ for $p \equiv 1 \mod 4$ reduces to a computational problem of solving a linear equation under some integer constraints. Up to the writing and best of the author's knowledge, the result of the third part is new to the Mathematics community.

Contents

1. Cameron-Liebler Line Classes

1.1. Background.

In this paper, $\mathbb{Z}_p$ represents a group $\{0, 1, \dots, p-1\}$ under addition (same as $\mathbb{Z}/p\mathbb{Z}$ in some other papers), and F_q represents a finite field of q elements with q being a prime power. A field or a vector space over a field with a star $(*)$ indicates a field or a vector space over a field without the additive identity. For example, $F_q^* = F_q \backslash \{\mathbf{0}\}$.

1.1.1. *Motivation.*

Cameron-Liebler line classes are objects in Projective Geometry that relate to some objects in Combinatorics. Examples of those objects are Partial Difference Sets (PDS) in coding theory, and Cayley Graphs in graph theory.

Coding theory is a study of codes that are used to transmit information in telecommunication with an error-correcting capability. Examples of applications are obvious, such as sending pictures from Mars and 3G internet data. Cameron-Liebler line classes always have a corresponding PDS, and these PDS can be used to generate two-weight projective codes.

Graph theory is a study of an abstract discrete incidence structure of vertices and edges. This abstract structure can represent many real-world structure. For example, in logistics, a vertex can represent a place, and an edge represents a transportation between places, or in computer network, a vertex can represent a machine (computer or server), and an edge represents a communication of data. Cayley graphs are graphs with particular symmetric structure that can be obtained from Cameron-Liebler line classes.

1.1.2. *Partial Difference Sets (PDS).*

Definition 1. A partial difference set (PDS) is a subset D of a group G such that the multiset $\{d_1 d_2^{-1} : d_1, d_2 \in D\}$ contains each nonidentity element in D λ times and each nonidentity element in $G \backslash D$ μ times. The parameters of the PDS is (v, k, λ, μ), with $|G| = v$ and $|D| = k$.

It is conventional to assume that $\lambda \neq \mu$ for D to be a PDS. If $\lambda = \mu$, then D is called difference set, another object of extensive study on its own.

Example 2. $D = \{(2,1), (0,2), (1,2), (0,1)\} \subset \mathbb{Z}_3 \times \mathbb{Z}_3$ is a PDS. By trying all 12 pairs of $d_1 d_2^{-1}$ in lexicographic order, we have

$$\{d_1 d_2^{-1} : d_1, d_2 \in D\} = \{(2,2), (1,2), (2,0), (1,1), (2,0), (0,1), (2,1), (1,0), (1,1), (1,0), (0,2), (2,2)\}$$

which contains each element in D exactly once each, and each nonidentity element in $\mathbb{Z}_3 \times \mathbb{Z}_3 \backslash D$ exactly twice each. Therefore, D is $(9, 4, 1, 2)$ PDS.

Note that adding or removing the identity element from PDS D does not change whether D is a PDS. By convention, we assume that D does not contain the identity element.

Definition 3. A character χ on an abelian group G is a homomorphism from G to a group of complex numbers $\mathbb{C}$ under multiplication. χ is called principal if it is a trivial homomorphism; that is $\chi(g) = 1$ for all $g \in G$.

Example 4. For $G = \mathbb{Z}_3 \times \mathbb{Z}_3$, we may take $\chi(1,0)$ to be $1, e^{2\pi i/3}$, or $e^{4\pi i/3}$, but nothing else, since $1 = \chi(0,0) = (\chi(1,0))^3$. Similarly for $\chi(0,1)$. Once we determine $\chi(0,1)$ and $\chi(1,0)$, then we completely determine χ.

For a finite subset $A \subset G$, we define $\chi(A) := \sum_{a \in A} \chi(a)$ a character sum of A. We say χ is principal on A if $\chi(a) = 1$ for all $a \in A$.

Example 5. Let G be an abelian group, and H be a finite subgroup of G. Then, $\chi(H) = 0$ if χ is not principal on H, and $\chi(H) = |H|$ otherwise.

Proof. Let n be an exponent of the group G (the exponent n of G is the least non-negative integer m such that $a^m = 1$ for all $a \in G$). Because χ is a homomorphism, $\chi(g)$ must be an nth root of unity for all $g \in G$. Because H is a group and χ is a homomorphism from H to C, $(\chi(h))_{h \in H}$ takes the same number of values $e^{2\pi i/n}$ for each $i = 0, 1, 2, ..., n-1$, and so $\chi(H) = 0$. □

Theorem 6. *[1, Theorem 1.6] A subset D of abelian group G is a (v, k, λ, μ) PDS if and only if, for all nonprincipal character χ,*

$$\chi(D) = \frac{\lambda - \mu \pm \sqrt{(\lambda - \mu)^2 + 4(k - \mu)}}{2}$$

1.1.3. *Projective Plane* .

Definition 7. A projective plane is a set L of lines together with a set P of points, and a relation $\in$ between points and lines called incidence, having the following properties [5]:

(1) Given any two distinct points, there is exactly one line incident with both of them.
(2) Given any two distinct lines, there is exactly one point incident with both of them.
(3) There exists a set of four points, no three of which are on a same line.

Note that the first condition is as expected in the usual Euclidean plane, but the second is not. The third condition is only to "prevent" the trivial cases to be considered as a projective plane. The empty (P, L) (no lines or points), (P, L) with only one line consisting of $|P|$ points, and (P, L) with only one point with $|L|$ lines passing that point are example of trivial point-line incident structure that satisfy axioms 1 and 2, but not 3.

One way to "fix" the Euclidean plane to satisfy the second condition is to, for each parallel class of lines, add a point at infinity where those parallel lines intersect, and define all those points at infinity to form a line in this projective plane. This is explained more in [5, Example 11]. There are also projective planes that have finite number of points and lines, which are called finite projective planes.

Example 8. [5, Example 10] Let $\Pi = (P, L)$ where $P = \{1, 2, 3, 4, 5, 6, 7\}$ and $L = \{l_1, l_2, l_3, l_4, l_5, l_6, l_7\}$, where $l_1, l_2, ..., l_7$ are {1, 2, 3}, {1, 4, 5}, {1, 6, 7}, {2, 4, 6}, {2, 5, 7}, {3, 4, 7} and {3, 5, 6} respective. Then Π is a projective plane. It is easily seen to satisfy all three conditions in Definition 7. This plane is called the Fano plane. See Figure 1.1.

Another example with 13 points and 13 lines can be found in [5, Example 17]. Finite projective planes have a well-studied numeric structure. For any finite projective plane $\Pi = (P, L)$, there must exist a positive integer n, called the order of the projective plane, such that:

- $|P| = |L| = n^2 + n + 1$
- Each line $l \in L$ contains exactly $n + 1$ points in P

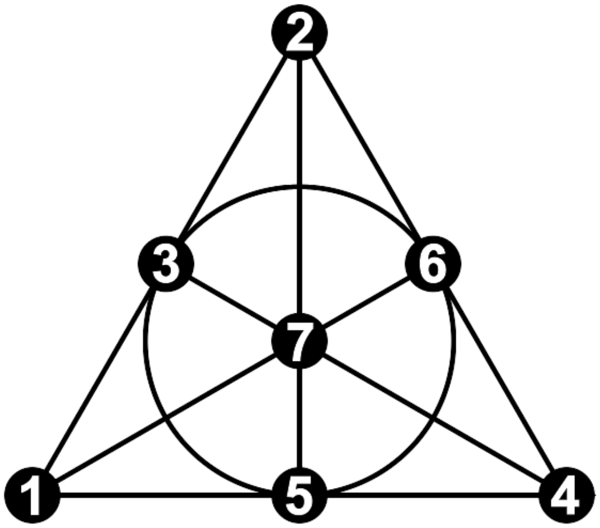

FIGURE 1.1. Fano Plane
By Watchduck (a.k.a. Tilman Piesk) - Own work, Public Domain,
https://commons.wikimedia.org/w/index.php?curid=17240472

- Each point $p \in P$ is on exactly $n+1$ lines in L

There is a well-known construction method to a get finite projective plane of order q using a three-dimensional vector space over a finite field F_q, which will be described as a special case in the next section.

1.1.4. *Projective Geometry $PG(n, q)$.*

Definition 9. (Whitehead's Axioms) A projective geometry is a set L of lines together with a set P of points, and a relation $\in$ between points and lines called incidence, having the following properties:

(1) Given any two distinct points, there is exactly one line incident with both of them.
(2) If lines AB and CD intersect, then so do lines AC and BD (where it is assumed that A and D are distinct from B and C).
(3) Every line contains at least 3 points

The third condition is only to "prevent" some trivial cases, similar to axiom 3 in Definition 7 of projective plane. The first condition is the same as in Def 7 for projective plane. The second condition is more relaxed than in projective plane: we only require some pairs of lines to intersect, not all. We call projective geometry finite if P is finite. The following method defines $\mathrm{PG}(n, q)$, which will be the space that Cameron-Liebler line classes are in.

Let $S = F_q^{n+1}$ be a vector space over F_q of dimension $n+1$. Let P and L be the sets of all subspace of dimension one and two of S, respectively. For each element

$l \in L$,we identify the subspace l by listing all one-dimensional subspace contained in l. That is, we view L as

$$L = \{\{p \in P : p \subset l\} : l \text{ is subspace of dimension two in } S\}$$

Define $\Pi = (P, L)$, viewing P and L as a set of points and lines (lines are now sets of points in P), respectively.

Theorem 10. $\Pi = (P, L)$ *as defined above from* $S = F_q^{n+1}$ *is a projective geometry.*

The proof is not that hard by checking axioms. The resulted projective geometry from S is called $\mathrm{PG}(n, q)$, and S is called the **underlying (vector) space** of $\mathrm{PG}(n, q)$.

Example 11. Let $S = F_2^3$. Write $S = \mathbb{Z}_2^3$. Then

$$P = \{\langle(1, 0, 0)\rangle, \langle(0, 1, 0)\rangle, \langle(0, 0, 1)\rangle, \langle(1, 1, 0)\rangle, \langle(0, 1, 1)\rangle, \langle(1, 0, 1)\rangle, \langle(1, 1, 1)\rangle\}$$

Consider a subspace of dimension two $\langle(1, 0, 0), (0, 1, 0)\rangle$, which contains three subspaces $\langle(1, 0, 0)\rangle, \langle(0, 1, 0)\rangle, \langle(1, 1, 0)\rangle$. This means $\{\langle(1, 0, 0)\rangle, \langle(0, 1, 0)\rangle, \langle(1, 1, 0)\rangle\} \in L$. If we consider $\langle(0, 1, 0).(0, 0, 1)\rangle$, then $\{\langle(0, 1, 0)\rangle, \langle(0, 0, 1)\rangle, \langle(0, 1, 1)\rangle\} \in L$. By considering all planes in S, we get all seven elements of L. If we identify $\langle(1, 0, 0)\rangle, \langle(0, 1, 0)\rangle, \langle(0, 0, 1)\rangle, \langle(1, 1, 0)\rangle, \langle(0, 1, 1)\rangle, \langle(1, 0, 1)\rangle, \langle(1, 1, 1)\rangle$ as numbers $1, 2, 4, 3, 6, 5, 7$ in that order and compare the result to Example 8, we see that S generates the same example.

In fact, the construction using underlying vector space will give a projective plane if the dimension of S is three. We can summarize the result from this as follows:

Corollary 12. *For each prime power q, there exists a projective plane of order q.*

Proof. Construct $\mathrm{PG}(2, q)$ from $S = F_q^3$. □

We can find numeric properties of $\mathrm{PG}(n, q)$ from underlying subspace point of view.

Lemma 13. *In* $PG(n, q) = (P, L)$*:*

- $|P| = \frac{q^{n+1}-1}{q-1} = q^n + q^{n-1} + ... + 1$
- $|L| = \frac{(q^{n+1}-1)(q^{n+1}-q)}{(q^2-1)(q^2-q)}$
- *Each line* $l \in L$ *contains* $q + 1$ *points*

A point in $\mathrm{PG}(n, q)$ corresponds to a one-dimensional subspace in the underlying space S. A line in $\mathrm{PG}(n, q)$ corresponds to a two-dimensional subspace in the underlying space. We define a **plane** in $\mathrm{PG}(n, q)$ as all points and lines in $\mathrm{PG}(n, q)$ whose underlying structure is contained in a same three-dimensional subspace.

Note that projective plane of order m (as in Subsection 1.1.3) is exactly the same as $\mathrm{PG}(2, m)$.

1.1.5. *Cameron-Liebler Line Classes.*

Definition 14. A Cameron-Liebler line class with parameter x, where x is a non-negative integer, is a set of $x(q^2 + q + 1)$ lines $\mathcal{L}$ in $\mathrm{PG}(3, q)$ such that

- For all $l \in \mathcal{L}$, $|\{m \in \mathcal{L} : m \cap l \neq \emptyset\}| = (q + 1)x + q^2$
- For all $k \notin \mathcal{L}$, $|\{m \in \mathcal{L} : m \cap k \neq \emptyset\}| = (q + 1)x$

Some author (such as [12]) may define a Cameron-Liebler line class with parameter x to be a set of $x(q^2+q+1)$ lines $\mathcal{L}$ in $\mathrm{PG}(3,q)$ such that every spread (a partition of points into sets of lines) of $\mathrm{PG}(3,q)$ contains x lines in $\mathcal{L}$. These two definitions are equivalent [3, Definition 1.1].

Example 15. Let $\Pi = \mathrm{PG}(n,q)$ with $n \geq 2$. Let p be a point in Π, and define $\mathrm{star}(p)$ to be the set of all lines through p. Let π be a plane in Π, and define line(π) to be the set of all lines contained in the plane π. We have the following trivial examples [12, Section 1]:

(1) The empty set is a Cameron-Liebler line class with parameter $x = 0$
(2) Both $\mathrm{star}(p)$ and $\mathrm{line}(\pi)$ are Cameron-Liebler line classes with parameter with $x = 1$
(3) If p is not inside the plane π, then $\mathrm{star}(p) \cup \mathrm{line}(\pi)$ is a Cameron-Liebler line class with $x = 2$

The complement of a Cameron-Liebler line class with parameter$x$ in the set of all lines of $\mathrm{PG}(3,q)$ is a Cameron-Liebler line class with parameter q^2+1-x. Hence, without loss of generality we may assume that $x \leq \frac{q^2+1}{2}$.

1.1.6. *Coordinate System in PG(3, q) and Klein Correspondence.* A coordinate system for $\mathrm{PG}(n,q) = (P,L)$ is defined by identifying each point $p \in P$ with a non-identity element p_v in the underlying subspace structure. However, any scalar multiple ap_v $(a, \in F_q)$ represents the same point p in $\mathrm{PG}(n,q)$, so we define an equivalence relation $x \sim y \iff xy^{-1} \in F_q$ over all non-identity elements in the underlying vector space S of $\mathrm{PG}(n,q)$ (that is, $x, y \in S = F_q^{n+1} \backslash \{\mathbf{0}\}$).

We, however, are interested in labeling projective lines, since the object of interest (Cameron-Liebler line classes) are lines. Projective lines are planes in the underlying space, which can be identified by two lines that span the plane. Of course this will require a more complicated equivalence relation. In this paper, we focus an equivalence relation on $\mathrm{PG}(3,q)$, which is where Cameron-Liebler line classes are.

Suppose we have a two-dimensional subspace P_0 inside the underlying space $S = F_q^4$, which is spanned by two nonzero vectors $u = (u_0, u_1, u_2, u_3)$ and $v = (v_0, v_1, v_2, v_3)$. Form a matrix $M_{u,v} = \begin{bmatrix} u_0 & u_1 & u_2 & u_3 \\ v_0 & v_1 & v_2 & v_3 \end{bmatrix}$. For each $0 \leq i \langle j \leq 3$, define $r_{ij}(u,v) = \det \begin{bmatrix} u_i & u_j \\ v_i & v_j \end{bmatrix}$. Form $R(u,v) = (r_{01}, r_{02}, r_{03}, r_{12}, r_{13}, r_{23})$. Note that $R(u,v)$ is an element in $S' = F_q^6$. We define (u,v) (which we use to identify a plane in S) to be equivalent to (s,t) if $R(u,v) = aR(s,t)$ for some $a \in F_q^*$. We have mapped a line in $\mathrm{PG}(3,q)$ into an element in F_q^6, where this F_q^6 has an equivalence relation that elements are invariant under scalar multiples. This means a line in $\mathrm{PG}(3,q)$ corresponds to a unique point in $\mathrm{PG}(5,q)$. This mapping from a line in $\mathrm{PG}(3,q)$ to a point in $\mathrm{PG}(5,q)$ is called the **Klein Correspondence**.

To show that the Klein Correspondence is a valid map, suppose (u,v) and (s,t) both span the same plane P_0 in F_q^4. Then each of s,t is a linear combination of u,v, so there exists 2×2 matrix X such that $\begin{bmatrix} s \\ t \end{bmatrix} = XM_{u,v}$. X must be invertible, since (s,t) spans a plane. Therefore, $r_{ij}(s,t) = \det(X)\det\begin{bmatrix} u_i & u_j \\ v_i & v_j \end{bmatrix}$. Hence, $R(s,t) = \det(X)R(u,v)$, and so $R(s,t) \sim R(u,v)$.

The Klein correspondence is also one-to-one (we will not prove it here), but not onto. It turns out, however, that the image of Klein correspondence is in a quadric, which we will define.

Definition 16. A quadric of a quadratic form Q are points x on the space such that $Q(x) = 0$.

The definition of quadratic form is in [2, Definition 7.1]. For more background on quadratic forms and the associated bilinear form, see [2, Chapter 1,2]. As $\begin{bmatrix} u_0 & u_1 & u_2 & u_3 \\ v_0 & v_1 & v_2 & v_3 \\ u_0 & u_1 & u_2 & u_3 \\ v_0 & v_1 & v_2 & v_3 \end{bmatrix}$ has rank 2, we have

$$r_{01}r_{23} + r_{02}r_{13} + r_{03}r_{12} = \det\begin{bmatrix} u_0 & u_1 & u_2 & u_3 \\ v_0 & v_1 & v_2 & v_3 \\ u_0 & u_1 & u_2 & u_3 \\ v_0 & v_1 & v_2 & v_3 \end{bmatrix} = 0$$

This means any point $R(u,v) = (r_{01}, r_{02}, r_{03}, r_{12}, r_{13}, r_{23})$ from the Klein correspondence must lie in a quadric $Q(x_1,x_2,x_3,x_4,x_5,x_6) = x_1x_6 + x_2x_5 + x_3x_4$. This quadric is called the **Klein Quadric**. The notation $Q^+(5,q)$ refers to the set of points in $\mathrm{PG}(5,q)$ whose underlying points lie in the Klein quadric. The Klein correspondence is in fact a bijection between the set of lines in $\mathrm{PG}(3,q)$ and $Q^+(5,q)$ [9, Chapter 12].

Example 17. We will find the Klein correspondence of a trivial Cameron-Liebler line class in $\mathrm{PG}(3,2)$ from $S = F_2^4 = \mathbb{Z}_2^4$. Let $\mathcal{L} = \mathrm{star}(p)$, where p has an underlying line $\langle(1,0,0,0)\rangle$. One element l_1 of $\mathcal{L}$ is a line represented by two lines $u =\langle(1,0,0,0)\rangle$,$v = \langle(0,1,0,0)\rangle$ in S, so we represent l_1 by its basis - $\{(1,0,0,0),(0,1,0,0)\}$. Then, $M_{u,v} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$, and hence $R(u,v) = (1,0,0,0,0,0) \in \mathbb{Z}_2^6$. If we had used other basis, such as $\{(1,0,0,0),(1,1,0,0)\}$, we want get $(1,0,0,0,0,0)$ as well. Pick a second element l_2 with basis $\{(1,0,0,0),(0,1,1,1)\}$. Then $M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$, and $R = (1,1,1,0,0,0)$. In general, all elements in $\mathcal{L}$ will be represented by $s = \langle(1,0,0,0)\rangle$ and $t = \langle w\rangle$ with $w \neq s$ a nonzero element in S. As $M_{s,t} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ w_0 & w_1 & w_2 & w_3 \end{bmatrix}$, $r_{12}, r_{13,}r_{23} = 0$ regardless of choice of w. In fact, it is not hard to check by hand that we will get all possible elements in the form $(a,b,c,0,0,0)$ with $a,b,c \in \mathbb{Z}_2$ and $(a,b,c) \neq (0,0,0)$ (zero is not a point in $\mathrm{PG}(5,q)$).

1.1.7. *Relationship of Cameron Liebler Line Classes to PDSs.* The following theorem gives a derivation of PDS from Cameron-Liebler line classes.

Theorem 18. *[12, Result 2.2] Let $\mathcal{L}$ be a set of $x(q^2+q+1)$ lines in $PG(3,q)$ with $0 < x \leq \frac{q^2+1}{2}$, and let $\mathcal{M} \subset Q^+(5,q)$ be the image of $\mathcal{L}$ under Klein correspondence. Let D be the underlying structure of $\mathcal{M}$ under addition; that is,*

$$D = \{av : a \in F_q^*, < v > \in \mathcal{M}\} \subset (F_q^6, +)$$

If $\mathcal{L}$ is a Cameron-Liebler line classes of parameter x, then $|D| = x(q^3-1)$ and D is a PDS in $(F_q^6, +)$ with $\chi(D) \in \{-x, -x+q^3\}$ for nonprincipal character χ.

In [12, Result 2.2], the result also specify what character χ gives $\chi(D) = -x$ and $\chi(D) = -x+q^3$. Note that not all PDS from $(F_q^6, +)$ can be mapped back to Cameron-Liebler line classes; one has to verify that PDS D is closed under scalar multiples (so that D can be projected into $\mathrm{PG}(5,q)$), and all points in D must lie in some quadric as well.

Example 19. Let's take a Klein correspondence from a trivial Cameron-Liebler line class described in Example 17. The image $\mathcal{M}$ is $\{\langle(a,b,c,0,0,0)\rangle : (a,b,c) \neq (0,0,0), a,b,c \in \mathbb{Z}_2\}$, and so $D = \{(a,b,c,0,0,0) : (a,b,c) \neq (0,0,0), a,b,c \in \mathbb{Z}_2\}$. But D is simply a subgroup with identity taken out, and the quotient of elements in a group is in the same group, so D is a PDS (in a trivial way) with $\mu = 0$.

1.2. Current work.

The introduction of [12] has summarized the history, existence and nonexistence results. Nontrivial Cameron-Liebler line classes are rare to find, so much so that there were once conjectured not to exist. There, however, have been existence results starting in the last two decades. Feng, Momihara, and Xiang [12] recently found a new construction that gives an infinite family of Cameron-Liebler line classes with $x = \frac{q^2-1}{2}$ for $q \equiv 5, 9 \mod 12$. That result was neatly simultaneous with Morgan Rodgers' PhD thesis [11] in which he found new constructions. Rodger's work was a combination of projective geometry and computer search, whereas Feng, Momihara, and Xiang's work involve more algebraic construction and proof.

Cameron-Liebler line classes are objects in projective geometry $Q^+(5,q)$, but have a connection to PDS. Our attempt is to work on PDSs using a difference set perspective. There are two possible directions:

(1) Relate potentially new PDSs back to Cameron-Liebler line classes.
(2) Study the new class of PDSs obtained from new construction, and generalize the new PDSs into other groups.

We focus specifically on the case $q = 5$, which is the smallest case that Feng, Momihara, and Xiang's new construction applies. In this case, the second goal is to "put" $D \subset \mathbb{Z}_5^6$ into other groups, such as non-elementary abelian groups $\mathbb{Z}_{25}^3$ or $\mathbb{Z}_{125}^2$.

1.3. Attempts to generalize an example of $q = 5$: relating PDS to Cameron-Liebler line classes.

The obtained PDSs from new construction are very complicated (it contains 1488 points inside $\mathbb{Z}_5^6$, scattered around the space) and is only produced by computer computation. We then consider a simpler example of PDS, and see whether that PDS can correspond to a Cameron-Liebler line class.

Theorem 20. *(Partial Spread Construction) Let $G = \mathbb{Z}_p^{2k}$. View G as $GF(q)^2$ with $q = p^k$ a prime power. A line in $GF(q)^2$ is $\langle x\rangle = \{ax : a \in F_q\}$ where $x \in G\backslash\{0\}$. Let $D = (\bigcup_{x\in X}\langle x\rangle)\backslash\{0\}$ be an arbitrary union of lines with additive identity taken out. Then D is a PDS.*

To achieve the first goal, the PDS must be in some quadric. Feng, Momihara, and Xiang uses a quadratic form $Q : E \times E \to F_5$, where $E = F_{125}$, defined by

$$Q(x,y) = \mathrm{Tr}(xy), (x,y) \in E \times E$$

Where a trace function from $F_q = F_{p^h}$ to a base field F_p is defined by $\mathrm{Tr}(x) = x + x^p + x^{p^2} + ... + x^{p^{h-1}}$. They show that this quadric can be a model for $Q^+(5, q)$, and it is convenient for us due to its simple algebraic formula, so it will be the quadratic form we use as well.The partial spread construction (Theorem 20) requires a line to be present in D, and we want to observe whether a line can be inside the quadric. The next Lemma tells us that the answer is no.

Lemma 21. *A subset D of $\mathbb{Z}_p^6 = GF(E)^2$ where $E = F_{p^3}$ inside a quadric defined by quadratic formula $Q(x, y) = Tr(xy)$ cannot contain any punctured line in $E \times E$ (a line with additive identity element taken out)*

Proof. Suppose $\langle x\rangle\backslash\{0\} \in D$. Write $x = (x_1, x_2) \in E \times E$. Then $\langle x\rangle\backslash\{0\}$ contains elements in the form (ax_1, ax_2) with $a \in E^*$. By definition, $Q(ax_1, ax_2) = \mathrm{Tr}(a^2x_1x_2)$. As a ranges over E^*, a^2 takes $\frac{p^3-1}{2}$ possible values in E^*, so $a^2x_1x_2$ also takes $\frac{p^3-1}{2}$ values. But Tr is a homomorphism from $E = F_{p^3}$ to $F_p = \mathbb{Z}_p$, so there are only p^2 values in E which has trace zero. Since $\frac{p^3-1}{2}\rangle p^2$ for all primes p, it is impossible that all $a^2x_1x_2$ have trace zero. □

The proof of Lemma 21 shows that there are no more than roughly $2/p$ of any line in $E \times E$ in the quadric $\mathrm{Tr}(xy)$. We think it is hard to find a way to delete most of the line and still get a PDS, so we change direction in the following section.

1.4. Attempts to generalize an example of $q = 5$: generalize PDS from Cameron-Liebler line classes to non-elementary abelian groups.

In order to put objects in $\mathbb{Z}_5^6$ into $\mathbb{Z}_{25}^3$ or $\mathbb{Z}_{125}^2$, we think that the multiplicative structure will be very different, but maybe not as much as the additive structure. As a result, we first try to find other ways to explain the new PDS, not by trace and multiplicative structure as the paper [12] has, but other new ways, especially any insight into the PDS's additive structure. The actual construction in detail is lengthy and complicated, so we do not repeat it here, but refer readers to [12, Section 3,4] whenever we use borrow the notations from that paper.

1.4.1. *An attempt to simplify the constructed example* .

The first idea is to look at what the PDS from the new construction looks like in $q = 5$ case. Using the same notation for I_X from [12, Section 4], the PDS is

$$D = \{(xy, xy^{-1}zw^l : x \in F_5^* =<\omega^{31}>, y \in <\omega^4>, z = 1, l \in I_X\} \tag{1.1}$$

where ω is a generator of F_{125}^*. I_X has size $2(q + 1) = 12$. Let $N = q^2 + q + 1$ (which is 26 for $q = 5$). The following are observations:

(1) The paper [12] writes $\overline{X} \subset \mathbb{Z}_{2N}$ defined in [12, Equation (3.5)] as

$$\overline{X} = 2A \cup (2B + N) \mod 2N$$

for some sets $A, B \subset \mathbb{Z}_n$ and

$$I_X = 4A \cup (4A + N) \cup (4B + 2N) \cup (4B + 3N) \tag{1.2}$$

where $aS = \{as : s \in S\}$ and $S + a = \{s + a : s \in S\}$ for a set S and a number a. Observe that 1.2 means $I_X = 2\overline{X} \cup (2\overline{X} + N)$. This simplifies ω^l (as $l \in I_X$) to be $\omega^{8d_i}(\mathrm{Tr}(\omega^{4(d_0+d_i)}))^2$ (same d_i as defined in [12, Equation (3.3)]) or $4\omega^{16}$ for the case $l \in 2\overline{X}$, or 3 times of those for the case $l \in 2\overline{X}+N$ (3 comes from $\omega^N = \omega^{31} = 3 \in F_5^*$). Explicitly from computation, we know

$\{\omega^{8d_i} : i = 1, ..., 5\} = \{122, 102, 423, 321, 112\}$, and $\{\omega^{8d_i}(\mathrm{Tr}(\omega^{4(d_0+d_i)}))^2 : i = 1, ..., 5\} = \{122, 403, 132, 321, 112\}$, and $4\omega^{16} = 020$, where the notation abc represents $ax^2 + bx + c \in GF(5, 3) = F_5[x]/\langle x^3 + 3x + 2\rangle$. Hence, $\omega^l \in \{122, 403, 132, 321, 112, 020, 311, 204, 341, 413, 331, 010\}$.

(2) The trace part $(\mathrm{Tr}(\omega^{4(d_0+d_i)}))^2$ can be simplified a little bit more to

$$(\mathrm{Tr}(\omega^{4(d_0+d_i)}))^2 = \mathrm{Tr}(\omega^{4(d_0+d_i)}\mathrm{Tr}(\omega^{4(d_0+d_i)}))$$

$$= \mathrm{Tr}(g(g + g^5 + g^{25})) = \mathrm{Tr}(g^2 + g^6 + g^{26}) = \mathrm{Tr}(g^2) + 2\mathrm{Tr}(g^6)$$

where $g = \omega^{4(d_0+d_i)}$, and the last equation is by $\mathrm{Tr}(g^{26}) = \mathrm{Tr}(g^{130}) = \mathrm{Tr}(g^6)$.

(3) From 1.1, we may treat $a = xy$ as an independent variable ranging in F_{125}^* (the map $f : F_5 \times \langle\omega^4\rangle \to F_{125}^*$, $f(x, y) = xy$ is bijective), then we can treat xy^{-1} as a dependent variable of a. Write $a = \omega^{4i+j}$, $i \in \{0, 1, ..., 30\}, j \in \{0, 1, 2, 3\}$. $\omega^{4i+j} = \omega^{4(i+8j)+(4-j)31}$, so $x = (\omega^{31})^{4-j}$ and $y = (\omega^4)^{i+8j}$. Then

$$b := xy^{-1} = (\omega^{31})^{4-j}/(\omega^4)^{i+8j} = \omega^{-4i-63j} = a^{-1}\omega^{-62j} = (-1)^j a^{-1}$$

The last equation comes from $\omega^{62} = -1$. So we have biject the original (x, y) into $(x, y) \mapsto (a, (-1)^{(\log_\omega a \mod 4)}a^{-1})$ where a can be treated as an arbitrary element in F_{125}^*.

We have tried all these algebraic simplifications, but saw no insight into how these can be helpful in understanding this PDS. We stopped and tried to prove that D is PDS by other way, which is handled in the next subsection.

1.4.2. *An attempt to prove D is PDS by character theory.* Roughly speaking, there are two main ways to see PDSs D. First, view D as a set with "difference set" properties by observing $\{d_1d_2^{-1} : d_1, d_2 \in D\}$, and second, view D as a set satisfying the character sum equation (from Theorem 6). We have tried to look at an equation $a-b = c$, with $a, b \in D$, and observe how many pairs of (a, b) can give a fixed element c. This attempt gave us no insight.

We then tried to understand D from the character perspective by ourselves, with the hope that we may come to find a different proof from the one in the paper [12].

The first step is to understand that a character on $E \times E$ can be thought of as

$$\chi_{a,b}((x, y)) := \chi_{1,E}(ax + by) = e^{2\pi i\mathrm{Tr}(ax+by)/5}, (a, b) \in E \times E \tag{1.3}$$

where $\chi_{1,E}$ is a principal character on E, and the trace function is from F_{125} to F_5 [12, Equation (2.1)]. This is because a character on $E \times E$ gives a value on (x, y) equal to product of $\chi_a(x)$ and $\chi_b(y)$. Those two are $e^{2\pi i\mathrm{Tr}(ax)/5}$ and $e^{2\pi i\mathrm{Tr}(by)/5}$, respectively. Multiplying them gives $e^{2\pi i\mathrm{Tr}(ax+by)/5}$.

For each punctured line (a subspace of dimension 1 with additive identity taken out) in F_5^6, the character sum on that line is either -1 or 4. This is because, as mentioned in Example 1.1.2, the whole line $\langle c\rangle$ has character sum 0 or 5. With identity taken out, the character sum decreases by 1. As a result, to calculate $\chi_{a,b}(D) = \sum_{x\in F_5^*, y\in\langle\omega^4\rangle, k\in K} \chi_{a,b}(xy, xy^{-1}k)$, where $K = \{\omega^l : l \in I_X\}$, which is just a union of punctured lines, it is enough to count how many punctured lines will give value 4, which happens if and only if one point on that line has character value 1. So it is enough to count how many $(y, k) \in \langle\omega^4\rangle \times K$ gives $\chi_{a,b}(y, y^{-1}k) = 1$, which by 1.3, happens if and only if $\mathrm{Tr}(ay+by^{-1}k) = 0$. Focusing on $\mathrm{Tr}(ay + by^{-1}k) = 0$ in order to find character value leads to a desired result for $a = 0$ or $b = 0$.

Lemma 22. *If $a = 0$ or $b = 0$ but not both, then there are exactly 72 pairs $(y, k) \in \langle\omega^4\rangle \times K$ satisfying $Tr(ay + by^{-1}k) = 0$.*

Proof. Observe that if ω^t is a solution to $\mathrm{Tr}(x) = 0$ (x as a variable), so are $\omega^{t+31}, \omega^{t+62}, \omega^{t+93}$. Those three solutions together with ω^t gives 4 solutions, each has different remainder mod 4 in its exponent (or its discrete log value). There are 24 non-zero solution to $\mathrm{Tr}(x) = 0$ (as trace is an onto homomorphism $\mathrm{Tr} : F_{125} \to \{e^{2\pi ik/5} : k = 0, 1, 2, 3, 4\}$, the size of kernal of Tr is $125/5 = 25$). These 24 solutions must then be partitioned into 6 sets, each of which is in the form $\{\omega^t, \omega^{t+31}, \omega^{t+62}, \omega^{t+93}\}$ for some t.

If $a = 0$, then $\mathrm{Tr}(ay + by^{-1}k) \equiv 0$ if and only if $\mathrm{Tr}(by^{-1}k) = 0$. For each $k \in K$, we can set $bk = \omega^{4i+j}$ $(0 \leq j \leq 3)$. Then the number of solutions is the same to $\mathrm{Tr}(\omega^j y^{-1}) = 0$, which is 6 because there are 6 solutions to $\mathrm{Tr}(\omega^t) = 0$ with the exponent $t \in \{0, 1, 2, ..., 123\}$, $t \equiv j \pmod 4$. Therefore, there are 6 solutions of y to the equation for each k. As we range over all possible 12 values of k, we get the number (y, k) pairs to be $12 * 6 = 72$. The case $b = 0$ is similar: $\mathrm{Tr}(ay) = 0$ clearly has 6 solutions for y for any fixed $a \neq 0$ by the same reasoning. □

Lemma 22 tells us that the character sum of D when a or b (but not both) is zero will always be $4(72) + (-1)(372 - 72) = -12$ (D is a union of $31 * 12 = 372$ lines, so the rest 300 lines has character sum -1). Comparing to [12, Result 2.2], we get character value of -12, which agrees with the parameter $x = 12$ of Cameron-Liebler line class generating this PDS. The other possible character sum over D is 123, which means there are 97 lines with character sum 4 with that particular character.

22 only explains when $a = 0$ or $b = 0$. But what if $a, b \neq 0$? The simplest case to move forward is $a = b = 1$. But even with this case we cannot find any insight to solve the problem by hand. As a result, we moved on to using computer to generate some data, and hope that some may inspire a new direction.

1.4.3. *Computer calculations related to the number of solutions to* $\mathrm{Tr}(ay+by^{-1}k) = 0$. For each $k \in K$, we can compute the number $n(k)$ of $y \in \langle\omega^4\rangle$ that satisfies $\mathrm{Tr}(ay + by^{-1}k) = 0$ by a computer by ranging all 31 values of y. For the actual code, see Algorithms 2 in Appendix A.The following is an example when $a = b = 1$ of $n(k)$ as $k \in K$:

$$7, 4, 4, 7, 7, 4, 5, 8, 8, 5, 5, 8$$

Other examples can give more seemingly “ugly” result. For example, for $a = 1, b = \omega^4$, we get

$$7, 4, 8, 5, 3, 12, 5, 8, 4, 7, 4, 5$$

Note that K can be partitioned into 6 pairs, each of which has a form $\{k_0, 3k_0\}$ for some $k_0 \in F^*_{125}$. We pair $n(k)$ with $n(3k)$. When $a = b = 1$, we have $(n(k))_{k\in K} =$

$$\begin{matrix} 7 & 4 & 4 & 7 & 7 & 4 \\ 5 & 8 & 8 & 5 & 5 & 8 \end{matrix}$$

where numbers of the same column are in the same pair of partitioning. If we observe the sum $n(k) + n(3k)$, and put the sum below that column, we get

$$12, 12, 12, 12, 12, 12$$

which looks promising to lead to some pattern. For $a = 1, b = \omega^4$, we first write 12 numbers as

$$\begin{matrix} 7 & 4 & 8 & 5 & 3 & 12 \\ 5 & 8 & 4 & 7 & 4 & 5 \end{matrix}$$

and then get the sum $n(k) + n(3k)$ as

$$12, 12, 12, 12, 7, 17$$

We do the same for other examples of (a, b). Interestingly, the only numbers that appear are 7,12,17, though the numbers of 7,12, and 17 that appears vary. The other possible numbers of total solutions besides 72 is 97, which then can only come from 17+17+17+17+17+12 (or in other order). We thought there may be some structure leading to "7,12,17" pattern, but we were unable to find one.

Another computer calculation we did is, instead of counting the number of y satisfying $\mathrm{Tr}(ay+by^{-1}k) = 0$ for each k, we count the number of k satisfying $\mathrm{Tr}(ay+by^{-1}k) = 0$ for each y. For example, for $a = b = 1$ and as $y = 1, \omega^4, \omega^8, ..., \omega^{120}$, we find the numbers of k to be

$$0, 1, 2, 4, 3, 1, 2, 3, 2, 2, 2, 1, 2, 4, 2, 4, 4, 3, 4, 2, 3, 2, 3, 3, 1, 1, 2, 1, 4, 2, 2$$

Varying a, b gives similarly "ugly" result - we cannot see any pattern in these numbers.

1.4.4. *Computer calculations on additive structure of D* . Finally, we try to observe the additive structure of D using computer computation. We first write a program to check how many points $(p_1, p_2, p_3, p_4, p_5, p_6) \in F_5^6$ in D have the first few coordinates as given, such as $p_1 = 1$ and $p_2 = 2$. That is, given numbers $a_1, \ldots, a_t$, what is the number $m(a_1, \ldots, a_t)$ of points in D such that $p_i = a_i$ for all $i = 1, ..., t$? For the actual code, see Algorithms 3 in Appendix A.

For $t = 2, 3$ we found a very symmetric result: for every $(a_1, a_2) \neq (0, 0)$, $m(a_1, a_2) = 60$, and for every $(a_1, a_2, a_3) \neq (0, 0, 0)$, $m(a_1, a_2, a_3) = 12$. But this is due to the observation 1 in 1.4.1 that the first three coordinates of D (not all zeroes) will range over E exactly $|K| = 12$ times.

For $t = 4$, we see how 12 points within fixed a_1, a_2, a_3 split into 5 cases of 5 values of a_4. Most of the time the numbers are

$$1, 2, 3, 4, 2$$

in some order. Occasionally the value is $12, 0, 0, 0, 0$. We observe no other pattern besides these two.

An interesting observation from computer computation is that, for each (a_1, a_2, a_3, a_4) such that $m(a_1, a_2, a_3, a_4) = 4$, four points in D with $p_i = a_i$ are corners of a plane (i.e. that D has four distinct points p, q, r, s such that $p + q = r + s$ under $\mathbb{Z}_5^6$). For the actual code, see Algorithms 4 in Appendix A.We were not sure what this observation might lead, nor where we should go next. We, however, give more thoughts on the "plane concepts" - anything related to a plane. We then observe some additive structure on K itself.

1.4.5. *Additive structure on K* . Recall that we can find explicit points in K: $K = \{122, 403, 132, 321, 112, 020, 311, 204, 341, 413, 331, 010\}$ (see observation 1 in 1.4.1). We also use computer to help observe the structure. For the actual code, see Algorithms 5 and 6 in Appendix A.

The first observation is that K in fact lie in a plane: embed $K = \mathbb{Z}_5^3$ with first, second, and third coordinates being x, y, z, then every point in K lie on the plane $x + 2z = 0$.

Second, choose any subspace of dimension two (or plane passing through the origin)P in $\mathbb{Z}_5^3$ not paralleled to $x + 2z = 0$. Let cosets of P be $P_i, i = 1, 2, 3, 4, 5$. Then we observe that $|K \cap P_i|, i = 1, 2, 3, 4, 5$ is always 1,2,3,4,2, the same pattern found in 1.4.4. For example, choosing $P : x = 0$ and so $P_i : x = i$ gives $|K \cap P_i|$ as $i = 1, 2, 3, 4, 5$ to be

$$2, 3, 1, 4, 2$$

This pattern inspired us to try defining K in additive term instead of multiplicative term as in the paper [12]. That is, instead of finding $X, A, B, \overline{X}$ as outlined in the paper, we may define K as a subset of $\mathbb{Z}_p^3$ such that for each plane passing through the origin P, the intersection number $|K \cap P_i|$ of each coset P_i must obey certain pattern. Unfortunately we did not see a way to use "additive" property to prove that, given K obeys certain structure and D is constructed some way from K, then D is a PDS. The main obstacle is that we have no insight (beyond the original, complicated algebraic proof using trace and character theory in the paper) the proof that D is a PDS using our own perspective or additive structure.

2. Finding PDS in Non Elementary Abelian Group

We failed to find new PDSs in non elementary abelian group $\mathbb{Z}_{25}^3$ or $\mathbb{Z}_{125}^3$, as we did not see enough additive structure in D in previous section to understand the structure making D a PDS. We then started to look at other non elementary abelian groups. The simple one we want to start with is $\mathbb{Z}_{p^2}^2$ with $p = 3$ because the group behaves differently when $p = 2$, and the group on only one dimension is not as interesting. The starting motivation comes from Davis' construction. The construction uses a generalization of "punctured" line, in a sense that we may "puncture" more points on a line than just the origin (but the way we puncture will still be symmetric, not arbitrary). In this section we will use additive notation for groups, so, for example, $nx := x + x + \ldots + x$ (n times) and $\langle a \rangle = \{na : n \in \mathbb{Z}^+\}$.

2.1. Background and a starting example.

Definition 23. In an abelian group $\mathbb{Z}_{p^t}^n$, for each $x \in \mathbb{Z}_{p^t}^n$, let $\langle x \rangle_{p^r} = \{ax : a \in \mathbb{Z}_{p^t}, ax$ has order strictly more than $p^r\}$.

Example 24. In $G = \mathbb{Z}_9^2$, $\langle (1, 0) \rangle_3 = \{(1, 0), (2, 0), (4, 0), (5, 0), (7, 0), (8, 0)\}$. In $G = \mathbb{Z}_{27}^2$, $\langle (1, 0) \rangle_3 = \langle (1, 0) \rangle \backslash \{(0, 0), (9, 0), (18, 0)\}$ and $\langle (1, 0) \rangle_9 = \langle (1, 0) \rangle \backslash \{(0, 0), (3, 0), (6, 0), ..., (24, 0)\}$.

Another way to define $\langle a \rangle_{p^r}$ is to take $\langle a \rangle$ and then delete all p^r elements in $\langle a \rangle$ which has order equal to or less than p^r. In $G = \mathbb{Z}_{p^t}^n$, we have $\langle x \rangle_{p^r} = \langle x \rangle \backslash \langle p^{t-r} x \rangle$.

[1, Theorem 3.1-3.3] uses these "generalized" punctured lines to construct $(p^4, (t+ep)(p^2-1), p^2+(t+ep)^2-3(t+ep), (t+ep)^2-(t+ep))$ PDS in $\mathbb{Z}_{p^2}^2$ for $3 \leq t \leq p+1$ and $1 \leq e \leq p-1$. As we will start to generalize from small example first, we will begin by giving a PDS from [1, Theorem 3.1-3.3] that applies to $\mathbb{Z}_9^2$.

Example 25. In $G = \mathbb{Z}_9^2$, let $D = \langle (1, 1) \rangle_3 \cup \langle (1, 2) \rangle_3 \cup \langle (1, 3) \rangle_3 \cup \langle (3, 1) \rangle_3$. Then D is a PDS.

Proof. Let χ be a nonprincipal character. Then χ is of order either 3 or 9 (the order n of homomorphism χ is the least positive integer m such that χ^m maps every element to the identity). We look at the case when χ has order 9 first.

If χ has order 9, then $K := \ker\chi$ has size 9 and is abelian, so $K \cong \mathbb{Z}_9$ or $\mathbb{Z}_3^2$. If $K \cong \mathbb{Z}_3^2$, then K has 8 elements of order 3, but G has exactly 8 elements of order 3, so K are exactly those elements of order 3 in G. But then it is easy to see that χ will have order 3, a contradiction. Therefore, $K \cong \mathbb{Z}_9$. That is, K is a line $\langle(a_1, a_2)\rangle$ for some $a = (a_1, a_2) \in G$ of order 9.

Consider $\chi(\langle(b_1, b_2)\rangle_3)$, where $b = (b_1, b_2) \in G$ of order 9. If $(b_1, b_2) \in K$ (i.e. $\langle(b_1, b_2)\rangle = K$), then χ is principal on $\langle(b_1, b_2)\rangle$, so $\chi(\langle(b_1, b_2)\rangle_3) = 6$. If $(b_1, b_2) \notin \langle(a_1, a_2)\rangle$ but $3(b_1, b_2) \in \langle(a_1, a_2)\rangle$, that is χ is principal on $\langle 3(b_1, b_2)\rangle$ but not $\langle(b_1, b_2)\rangle$, then $\chi(\langle(b_1, b_2)\rangle_3) = \chi(\langle(b_1, b_2)\rangle) - \chi(\langle 3(b_1, b_2)\rangle) = 0 - 3 = -3$. Finally, if $3(b_1, b_2) \notin \langle(a_1, a_2)\rangle$, then $\chi(\langle(b_1, b_2)\rangle_3) = \chi(\langle(b_1, b_2)\rangle) - \chi(\langle 3(b_1, b_2)\rangle) = 0 - 0 = 0$.

If $(a_1, a_2) = (1, 1)$, then $\chi(\langle(1, 1)\rangle_3) = 6$, and because $3(1, 2), 3(1, 3), 3(3, 1) \notin \langle(1, 1)\rangle$, we must have $\chi(\langle(b_1, b_2)\rangle_3) = 0$ for $(b_1, b_2) \in \{(1, 2), (1, 3), (3, 1)\}$. As a result $\chi(D) = 6 + 0 + 0 + 0 = 6$. Similar argument will show that for $(a_1, a_2) \in \{(1, 1), (1, 2), (1, 3), (3, 1)\}$, $\chi(D) = 6$.

If $(a_1, a_2) = (1, 4)$, then $3(1, 1) \in \langle(a_1, a_2)\rangle$ but $3(1, 2), 3(1, 3), 3(3, 1) \notin \langle(1, 4)\rangle$. Therefore, $\chi(\langle(1, 1)\rangle_3) = -3$ and $\chi(\langle(b_1, b_2)\rangle_3) = 0$ for $(b_1, b_2) \in \{(1, 2), (1, 3), (3, 1)\}$. As a result, $\chi(D) = -3 + 0 + 0 + 0 = -3$. Similar argument will show that for all $(a_1, a_2) \in \{(1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (1, 0), (6, 1), (0, 1)\}$, $\chi(D) = -3$.

If χ has order 3, then $K = \ker\chi$ has size 27 and is abelian, and it must be that $K \cong \mathbb{Z}_9 \times \mathbb{Z}_3$. One way to visualize K is to represent K with a line $\langle(a_1, a_2)\rangle, (a_1, a_2) \in G$ of order 9, together with two other lines with "distance" 3 and 6 apart paralleled to it. For example, if $(a_1, a_2) = (1, 1)$, then $K = \langle(1, 1)\rangle \cup (\langle(1, 1)\rangle + (0, 3)\rangle) \cup (\langle(1, 1)\rangle + (0, 6))$. Note that the "direction" that we move $\langle(1, 1)\rangle$ by $(0, 3)$ and $(0, 6)$ may be different: if we have chosen $(a_1, a_2) = (0, 1)$, then we would have moved $\langle(a_1, a_2)\rangle$ by $(3, 0)$ and $(6, 0)$ instead, but we always add elements of order 3.

Consider $\chi(\langle(b_1, b_2)\rangle_3)$, where $(b_1, b_2) \in G$ of order 9. If $(b_1, b_2) \in \langle(a_1, a_2)\rangle$, then $\chi(\langle(b_1, b_2)\rangle_3) = 6$. If $3(b_1, b_2) \in \langle(a_1, a_2)\rangle$, then $3(b_1, b_2) = k(a_1, a_2)$ for some $k \in \mathbb{Z}$ divisible by 3 (since (a_1, a_2) has order 9 but $3(b_1, b_2)$ has order 3), so write $k = 3r, r \in \mathbb{Z}$. Then, $3(b_1, b_2) = 3r(a_1, a_2)$, so $(b_1, b_2) - r(a_1, a_2)$ is an element of order 3. Therefore, $\chi((b_1, b_2) - r(a_1, a_2)) = 1$, and so $\chi((b_1, b_2)) = \chi(r(a_1, a_2)) = 1$ since $r(a_1, a_2) \in \langle(a_1, a_2)\rangle$. This means χ is principal on $\langle(b_1, b_2)\rangle$, and so $\chi(\langle(b_1, b_2)\rangle_3) = 6$. If $3(b_1, b_2) \notin \langle(a_1, a_2)\rangle$, then χ is not principal on $\langle(b_1, b_2)\rangle$, but on $\langle 3(b_1, b_2)\rangle$ (K includes all elements of order 3), and so $\chi(\langle(b_1, b_2)\rangle_3) = \chi(\langle(b_1, b_2)\rangle) - \chi(\langle 3(b_1, b_2)\rangle) = 0 - 3 = -3$.

For $K \cong \mathbb{Z}_9 \times \mathbb{Z}_3$, χ will be principal on exactly one of $\langle(1, 1)\rangle, \langle(1, 2)\rangle, \langle(1, 3)\rangle, \langle(3, 1)\rangle$, and none on the rest. Therefore, $\chi(D) = 6 - 3 - 3 - 3 = -3$.

Since $\chi(D)$ is either 6 or -3 for all nonprincipal character, D is a PDS. □

It is important to note that idea of **representing** $K = \ker\chi$ even with $K \cong \mathbb{Z}_9 \times \mathbb{Z}_3$ simply by a line $\langle a\rangle$. One advantage on considering χ as its representation $\langle a\rangle$ is that we can specify all nonprincipal χ by simply choose one direction $a \in G$ for a line to be a representative of $\ker\chi$ (for both χ of order 3 and 9). We can formalize this idea as followed.

Definition 26. Let χ be a character in $G = \mathbb{Z}_{p^m}^2$. A **representation** of χ is a line $\langle a\rangle \subset G$ such that $\langle a\rangle \subset \ker\chi$.

TABLE 1. The value of $\chi(L)$ on some sets L for χ of order 9. Represent $\ker(\chi) = \langle a \rangle$

Sets	If $b \in \langle a \rangle$	If $b \notin \langle a \rangle, 3b \in \langle a \rangle$	If $3b \notin \langle a \rangle$
$\langle b \rangle$	9	0	0
$\langle b \rangle_3$	6	-3	0
$\langle 3b \rangle$	3	3	0

TABLE 2. The value of $\chi(L)$ on some sets L for χ of order 3. $\ker(\chi)$ is represented by $\langle a \rangle$

Sets	If $b \in \langle a \rangle$	If $b \notin \langle a \rangle, 3b \in \langle a \rangle$	If $3b \notin \langle a \rangle$
$\langle b \rangle$	9	9	0
$\langle b \rangle_3$	6	6	-3
$\langle 3b \rangle$	3	3	3

If χ is of order 3, then χ have more than one representation, but we can observe if two representations are equivalent easily. For example, if χ is order 3 and χ is represented by $\langle (1,1) \rangle$, then it is the same if χ is represented by $\langle (1,4) \rangle$ or $\langle (1,7) \rangle$. In general, if χ is a character of order p^r in $G = \mathbb{Z}_{p^m}^2$, and $\langle a \rangle$ represents χ, then $\langle b \rangle$ represents χ if there is nonzero element $c \in \langle b \rangle$ such that $a - c$ has order p^{m-r}.

2.2. **Visualization of character values on generalized punctured lines.** In this section we let $G = \mathbb{Z}_9^2$. From the proof of Example 25, we can summarize how nonprinciple character interacts with generalized punctured line in Tables 2 and 1. We list the character values $\chi(L)$ on each set L.

Note that Table 2 can be obtained by shifting column of Table 1 to the right by 1, and keep the leftmost column the same. Also, we can calculate$\chi(\langle b \rangle_3)$ by realizing that $\chi(\langle b \rangle_3) = \chi(\langle b \rangle) - \chi(\langle 3b \rangle)$.

One way to visualize how a, b interact in the tables is as followed. First, note that we only focus on lines or generalized punctured lines, and they can all be represented by $\langle a \rangle_{p^r}$ for some elements $a \in G$ of order 9. a is a direction of the line, and we may list all possible directions in G to be

$$S = \{(1,0), (1,1), \ldots, (1,8), (0,1), (3,1), (6,1)\}$$

It is not hard to check that any element must lie on a line generated by at least one element in S (so that S actually includes all possible directions).

Second, we try to find a necessary and sufficient conditions on (a, b) that determines the case among three columns in Table 2 and 1. Let

$$\begin{aligned} R_1 &= \{a, b \in S : b \in < a >\} \\ R_2 &= \{a, b \in S : b \notin < a >, 3b \in < a >\} \\ R_3 &= \{a, b \in S : 3b \notin < a >\} \end{aligned}$$

By restricting directions to only elements of S and going through simple arguments to check, we can simplify R_i to be more useful to:

$$\begin{aligned} R_1 &= \{a, b \in S : a - b = 0\} \\ R_2 &= \{a, b \in S : a - b \neq 0, 3(a - b) = 0\} \\ R_3 &= \{a, b \in S : 3(a - b) \neq 0\} \end{aligned}$$

TABLE 3. Visualizing $b \in S$ which determines $l \in \{1,2,3\}$ that $(a,b) \in R_l$ where $a = (1,1)$

	(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)	(0,7)	(0,8)
(0,0)		3							
(1,0)	3	**1**	3	3	**2**	3	3	**2**	3
(2,0)									
⋮		3							
		3							
(8,0)									

The element of G or cell (i,j) is represented at row $i+1$ and column $j+1$, counting from top to bottom, and left to right. If a number l is in the cell (i,j), that means $(a,(i,j)) \in R_l$. If a number is not written at cell (i,j), that means $(i,j) \notin S$.

Let's see an example how this can be helpful.

Problem 27. Let χ have order 9 and $\ker\chi = \langle(2,2)\rangle$. Let $D = \langle(1,1)\rangle_3 \cup \langle(1,2)\rangle_3 \cup \langle(1,3)\rangle_3 \cup \langle(3,1)\rangle_3 \cup \langle(1,4)\rangle$. What is the value of $\chi(D)$?

$\ker\chi = \langle(2,2)\rangle$ means χ is represented by $\langle a\rangle$ with $a = (1,1) \in S$. We can visualize values of $b \in S$ that (a,b) is in each R_i as followed.

We use Table 1 and 3:

- Since 3 appears in $(1,2), (1,3)$ in Table 3, using the third column of the Table 1, $\chi(\langle(1,2)\rangle_3) = \chi(\langle(1,3)\rangle_3) = 0$
- Since 2 appears in $(1,4)$ in Table 3, using the second column of the Table 1, $\chi(\langle(1,4)\rangle) = 0$
- Since 1 appears in $(1,1)$ in Table 3, using the first column of the Table 1, $\chi(\langle(1,1)\rangle_3) = 6$

Therefore, $\chi(D) = 0 + 0 + 0 + 6 = 6$.

Problem 28. Let χ have order 3 and $\langle(1,1)\rangle \subset \ker\chi$. Let $D = \langle(1,1)\rangle_3 \cup \langle(1,2)\rangle_3 \cup \langle(1,3)\rangle_3 \cup \langle(3,1)\rangle_3 \cup \langle(1,4)\rangle$. What is the value of $\chi(D)$?

We progress exactly the same way as the previous problem, but we use Table 2 instead of Table 1 (or equivalently use the same Table 1 but read the number on one column to the left from normal). We get $\chi(\langle(1,2)\rangle_3) = \chi(\langle(1,3)\rangle_3) = -3$, $\chi(\langle(1,4)\rangle) = 9$, and $\chi(\langle(1,1)\rangle_3) = 6$. Therefore, $\chi(D) = -3 - 3 + 9 + 6 = 9$.

Using table may seem to only represent trivial facts, but it helps to generalize from $\mathbb{Z}_9$ to $\mathbb{Z}_{27}$.

2.3. **Attempts to generalize from $\mathbb{Z}_9$ to $\mathbb{Z}_{27}$.** When moving from finding PDS in $\mathbb{Z}_9^2$ to $\mathbb{Z}_{27}^2$, there are two things that will be changed: the tables of $\chi(L)$ for each set L (Table 2 and 1), and the set R_i that helps us visualize and calculate $\chi(D)$ (as we see in by using Table 3).

Again, it is not hard to see that:

TABLE 4. The value of $\chi(L)$ on some sets L for χ of order 27. Represent $\ker(\chi) = \langle a \rangle$

Sets	If $b \in \langle a \rangle$	If $b \notin \langle a \rangle, 3b \in \langle a \rangle$	If $3b \notin \langle a \rangle, 9b \in \langle a \rangle$	If $9b \notin \langle a \rangle$
$\langle b \rangle$	27	0	0	0
$\langle b \rangle_3$	18	-9	0	0
$\langle b \rangle_9$	24	-3	-3	0
$\langle 3b \rangle$	9	9	0	0
$\langle 9b \rangle$	3	3	3	0

TABLE 5. Visualizing $b \in S$ which determines $l \in \{1, 2, 3, 4\}$ that $(a, b) \in R_l$ where $a = (1, 1)$

	(0,0)	(0,1)	(0,2)	. . .																							(0,26)
(0,0)		4																									
(1,0)	4	1	4	4	3	4	4	3	4	4	2	4	4	3	4	4	3	4	4	2	4	4	3	4	4	3	4
(2,0)																											
(3,0)		4																									
⋮	⋮	⋮	⋮																								⋮
(24,0)																											
(25,0)		4																									
(26,0)																											

The element of G or cell (i, j) is represented at row $i + 1$ and column $j + 1$, counting from top to bottom, and left to right. If a number l is in the cell (i, j), that means $(a, (i, j)) \in R_l$. If a number is not written at cell (i, j), that means $(i, j) \notin S$.

- If χ has order 9 and is represented by $\langle a \rangle$, then the Table 4 can be used as usual by shifting values one column to the right
- If χ has order 3 and is represented by $\langle a \rangle$, then the Table 4 can be used as usual by shifting values two columns to the right

To use the table to find character value, we can define R_i in a similar way as before:

$$\begin{aligned} R_1 &= \{a, b \in S : a - b = 0\} \\ R_2 &= \{a, b \in S : a - b \neq 0, 3(a - b) = 0\} \\ R_3 &= \{a, b \in S : 3(a - b) \neq 0, 9(a - b) = 0\} \\ R_4 &= \{a, b \in S : 9(a - b) \neq 0\} \end{aligned}$$

Then, we can construct a similar table (shown in Table 3) as the Table 3.

Once we have tools to evaluate character values of order 3,9,27 on any union of generalized lines pretty efficiently, we start testing different intuitive unions of generalized lines. We tested by trial and error many unions. The following are examples of what we tested if the set is a PDS:

(1) $\langle(1, 1)\rangle_3 \cup \langle(1, 2)\rangle_3 \cup \langle(1, 3)\rangle_3 \cup \langle(3, 1)\rangle_3$
(2) $\langle(1, 1)\rangle_9 \cup \langle(1, 2)\rangle_9 \cup \langle(1, 3)\rangle_9 \cup \langle(3, 1)\rangle_9$
(3) $\langle(1, 1)\rangle_9 \cup \langle(1, 2)\rangle_9 \cup \langle(1, 3)\rangle_9 \cup \langle(1, 4)\rangle_9 \cup \langle(1, 5)\rangle_9 \cup \langle(1, 6)\rangle_9 \cup \langle(1, 7)\rangle_9 \cup \langle(1, 8)\rangle_9 \cup \langle(1, 9)\rangle_9 \cup \langle(3, 1)\rangle_9 \cup \langle(6, 1)\rangle_9 \cup \langle(9, 1)\rangle_9$

(4) $\langle(1,1)\rangle_9 \cup \langle(1,2)\rangle_9 \cup \langle(1,3)\rangle_3 \cup \langle(1,4)\rangle_9 \cup \langle(1,5)\rangle_9 \cup \langle(1,6)\rangle_3 \cup \langle(1,7)\rangle_9 \cup \langle(1,8)\rangle_9 \cup \langle(1,9)\rangle_3 \cup \langle(3,1)\rangle_9 \cup \langle(6,1)\rangle_9 \cup \langle(9,1)\rangle_3$

All of these are not PDS, but finally, we are able to find one PDS in $\mathbb{Z}_{27}^2$.

Claim 29. Let $D = \langle(1,1)\rangle_9 \cup \langle(1,2)\rangle_9 \cup \ldots \cup \langle(1,9)\rangle_9 \cup \langle(1,10)\rangle_3 \cup \langle(1,11\rangle_3 \cup \langle(1,12)\rangle_3 \cup \langle(3,1)\rangle_9 \cup \langle(6,1)\rangle_9 \cup \langle(9,1)\rangle_9 \cup \langle(12,1)\rangle_3$. Then D is a PDS.

Proof. The main idea is find all character values of D on all nonprincipal characters (of order 3,9,27). Using Tables 4 and 5 can be helpful.

(1) Character χ of order 27

We know $|\ker\chi| = 27$, and it is not hard to see that $\ker\chi \cong \mathbb{Z}_{27}$.

(a) If $K = \langle(1,1)\rangle$, then $\chi(\langle(1,1)\rangle_9) = 18$; $\chi(\langle(1,i)\rangle_9) = 0$ for $i = 2,3,...,9$; $\chi(\langle(1,10)\rangle_3) = -3$; $\chi(\langle(1,i)\rangle_3) = 0$ for $i = 1,2$; $\chi(\langle(i,1)\rangle_9) = 0$ for $i = 3,6,9$; $\chi(\langle(12,1)\rangle_3) = 0$. Therefore, $\chi(D) = 18 - 3 = 15$. The argument is similar and gives the same $\chi(D)$ if $K = \langle(1,k)\rangle$ for $k = 2,3,...,9$.

(b) If $K = \langle(1,10)\rangle$, $\chi(\langle 1,1\rangle_9) = -9$; $\chi(\langle(1,10)\rangle_3) = 24$; and the other generalized lines have character value 0. Therefore, $\chi(D) = 15$. The argument is similar and gives the same $\chi(D)$ if $K = \langle(1,11)\rangle, \langle(1,12)\rangle$.

(c) If $K = \langle(1,0)\rangle$, then $\chi(\langle(1,9)\rangle_9) = -9$; and $\chi(\langle(1,12)\rangle_3) = -3$. Therefore, $\chi(D) = -12$. The argument is similar and gives the same $\chi(D)$ if $K = \langle(1,k)\rangle$ for $k = 13,14,...,26$.

Similar analysis can be made with $K = \langle(i,1)\rangle$ for $i = 0,3,6,...,24$.

(2) Character χ of order 9

It is enough to consider when χ is represented by $\langle(1,i)\rangle$ for $i = 0,1,...,8$ and $\langle(i,1)\rangle$ for $i = 0,3,6$.

(a) If χ is represented by $\langle(1,1)\rangle$, then $\chi(\langle(1,1)\rangle_9) = 18$; $\chi(\langle(1,4)\rangle_9) = \chi(\langle(1,7)\rangle_9) = -9$; $\chi(\langle(1,10)\rangle_3) = 24$; $\chi(\langle(1,11)\rangle_3) = \chi(\langle(1,12)\rangle_3) = \chi(\langle(12,1)\rangle_3) = -3$ and the rest has character value 0. Therefore, $\chi(D) = 18 - 9 - 9 + 24 - 3 - 3 - 3 = 15$. The argument is similar and gives the same $\chi(D)$ if χ is represented by $\langle(1,2)\rangle$ or $\langle(1,3)\rangle$.

(b) If χ is represented by $\langle(1,4)\rangle$, then $\chi(\langle(1,4)\rangle_9) = 18$; $\chi(\langle(1,1)\rangle_9) = \chi(\langle(1,7)\rangle_9) = -9$; $\chi(\langle(1,101)\rangle_3) = \chi(\langle(1,11)\rangle_3) = \chi(\langle(1,12)\rangle_3) = \chi(\langle(12,1)\rangle_3) = -3$; and the rest has character 0. Therefore, $\chi(D) = 18 - 9 - 9 - 3 - 3 - 3 - 3 = 1 - 12$. The argument is similar and gives the same $\chi(D)$ if χ is represented by $\langle(1,i)\rangle$ for $i = 4,5,...,0$.

Similar analysis can be made with χ represented by $\langle(i,1)\rangle$ for $i = 0,3,6$.

(3) Character χ of order 3

It is enough to consider when χ is represented by $\langle(1,i)\rangle$ for $i = 0,1,2$ and $\langle(0,1)\rangle$. If χ is represented by $\langle(1,1)\rangle$, then $\chi(\langle(1,1)\rangle_9) = \chi(\langle(1,4)\rangle_9) = \chi(\langle(1,7)\rangle_9) = 18$; $\chi(\langle(1,10)\rangle_3) = 24$. All other 9 generalized lines in D in the form $\langle(a,b)\rangle_9$ has character value -9 and all other 3 generalized in D in the form $\langle(a,b)\rangle_3$ has character value 3. This gives $\chi(D) = -12$. All four cases gives $\chi(D) = -12$ as well with similar argument.

□

We were able to generalize this result to all prime $p \geq 3$ to get a PDS in $\mathbb{Z}_{p^2}^2$. Furthermore, we were able to generalize this construction to get a PDS in $\mathbb{Z}_{p^t}^2$ for all $t \geq 2$. Dr. Davis, however, realized that the general construction obtained was familiar, and found that the result we got has already been discovered during the work for PhD thesis of one of his PhD student, John Polhill, in 1999. Polhill has this construction for $\mathbb{Z}_{p^r}^{2t}$ for all $r \geq 2, t \geq 1$ [4, Theorem 11]. He also finds other PDS that is disjoint from PDS in [4, Theorem 11] and union it with the original PDS to get a new PDS. His work and the general construction can be found in [4, Theorem 11-13].

At this point, we want to change a direction of the research again. Previously we have used generalized lines as "building blocks" of a PDS, and ended up with a result included in the theorem already found. We want to start from a different point. Instead of restricting a PDS to only unions of generalized lines, we want to know all PDS in a group G. This, of course, seems like a much harder task. If $|G| = n$, then there are 2^n possible sets to check whether they are PDS. We, however, still tried to exhaustively find PDS in a group whose size is not too big.

2.4. **Computer result on the search of PDS on $\mathbb{Z}_4^3$.** Which group should we start the search? Non-elementary abelian group has an interesting structure, and is harder to find PDS than elementary one. Also, Polhill construction applies to spaces of even dimensions ($\mathbb{Z}_{p^r}^{2t}$). There are not many known constructions of PDS on spaces of odd dimensions. The starting point for us is hence $\mathbb{Z}_{p^2}^3$. Since p behaves differently when p is odd and even, we decide to start with $\mathbb{Z}_9^2$, the smallest case with p odd, hoping that $p = 3$ example can generalize to other primes. We could not find a way to search the space in a reasonable time, so we start even with $\mathbb{Z}_4^3$. We will focus on the search of PDS on $\mathbb{Z}_4^3$ in this subsection. The following fact is helpful to reduce the search space in finding all PDS in abelian group G.

Lemma 30. *Let D be a PDS in abelian G. Then $a \in D$ if and only if $-a \in D$.*

Proof. Suppose that D is a (v, k, λ, μ) PDS. Let $a \in G$. Let $(x_i, y_i), i = 1, 2, ..., t$ be all the solutions $(x, y) \in D \times D$ to $x - y = a$. Then, all the solutions $(x, y) \in D \times D$ to $x - y = -a$ are exactly (y_i, x_i), $i = 1, 2, ..., t$. If $a \in D$, then $t = \lambda$; otherwise, $t = \mu$. Similarly, if $-a \in D$, then $t = \lambda$; otherwise, $t = \mu$. Since $\lambda \neq \mu$ (as a convention for definition of PDS), $a \in D$ if and only if $-a \in D$. □

Lemma 30 implies that, if we have selected $a \in G$ to be either in D or not, then we have fixed whether $-a$ is in D as well. This reduces the number of variables to iterate by at most half.

Consider a search on $G = \mathbb{Z}_4^3$. If $a \in G$ has order 4, then we can pair $(a, -a)$ together so that the algorithms decides whether $a \in D$ simultaneously with whether $-a \in D$. There are 56 elements of order 4 in G, so there are 28 pairs to consider. There are 7 elements of order 2 in G, and each can be selected to be in D or not. The identity element of G can be assumed not to be in D. Hence, in total there are 35 variables (28 pairs and 7 elements of order 2) that determine all possible D that can be a PDS. The search space is now $2^{35} \approx 33.6 \times 10^9$. An algorithms (written in Java) to check if a given set D is a PDS empirically can check about one million sets a second, so we approximated that the search would take about half a day. We can reduce the search space a bit more using the following obvious fact.

Lemma 31. *Let D be a PDS in G. Let ϕ be an automorphism of G. Then the image $\phi(D)$ of the automorphism ϕ on D is also a PDS in G.*

This lemma allows us to assume without loss of generality something about D. This reduces the search space by a factor of about 20. For the actual code, see Algorithms 7 in Appendix A.

After the run of the algorithms, we get the following result:

```
All execution of this method takes: 2431.068312381 seconds.
The number of sets checked whether it is PDS is: 1453825484
The total number of PDS found is: 6464
All PDS are broken down into difference categories:
31 PDS are found as type 1 - as subgroups
6432 PDS are found as type 2 - as (64,28,12) Difference Set
1 PDS are found as type 3 - as others. They are: [[[0, 0, 0, 0], [0, 0, 0, 0],
    [0, 0, 0, 0], [0, 0, 0, 0]], [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0,
    0, 0, 0]], [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]], [[0, 0,
    0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]]
```

Hadamard difference set is a difference set on $\mathbb{Z}_4^3$ that is already known. A difference set with identity taken out will always be a PDS. In $\mathbb{Z}_4^3$, Hadamard difference set is a (64,28,12) difference set (same definition as a (64,28,12,12) PDS defined in this paper). Those 6432 type-2 PDSs in the result are from this difference set.

Type-1 PDSs in the result indicates that PDS found is a subgroup with an identity taken out, so that PDS is trivial.

Type-3 PDSs are other PDSs. The only one we found is the empty set (all 64 zeroes in the result indicates that D does not have any element on any of 64 points of $\mathbb{Z}_4^3$), which is trivial as well.

We summarized the result as follows:

Fact 32. *Let D be a PDS in $G = \mathbb{Z}_4^3$. Then D is either:*

- *A subgroup of G*
- *A translate of a Hadamard difference set with an identity taken out*

If the result on an algorithms shows some PDS D that is not known, we could have studied that PDS and generalized that set. The run does not show any new PDS, so we do not proceed to find PDS in $\mathbb{Z}_{p^2}^3$.

3. Finding PDS in $\mathbb{Z}_p^3$

It is, however, interesting to think about possibility of finding PDS in $\mathbb{Z}_p^3$ instead of $\mathbb{Z}_{p^2}^3$. Though $\mathbb{Z}_p^3$ is elementary abelian, it has dimension three, and there has been little known results about PDS on odd-dimensional spaces. In this section, we summarize the approach to find all PDS in $\mathbb{Z}_p^3$, and state a partial success in proving nonexistence result of PDS in $\mathbb{Z}_p^3$ for $p \equiv 3 \mod 4$.

3.1. **Any nontrivial union of lines in $\mathbb{Z}_p^3$ is not a PDS.** Consider a union of lines $D = (\bigcup_{i=1}^{j} L_i) \backslash \{(0,0,0)\} \subset \mathbb{Z}_p^3$, where $L_i = \langle a_i \rangle$ for some element $a_i \in \mathbb{Z}_p^3$ of order p. Let χ be a nonprincipal character on $\mathbb{Z}_p^3$. A line L_i has character sum $\chi(L_i) = p$ if χ is principle on L_i (i.e. $L_i \subset \ker\chi$), otherwise $\chi(L_i) = 0$. Let l_χ be the number of lines L_i such that $L_i \subset \ker\chi$. Since $\chi(0,0,0) = 1$ always, $\chi(D) = pl_\chi - j$. If D is PDS, then $\chi(D)$ has two possible values, so l_χ can have two possible values. We can summarize this into the following:

Lemma 33. *Let $D = (\bigcup_{i=1}^{j} L_i) \backslash \{(0,0,0)\}$, where $L_i = \langle a_i \rangle$ for some element $a_i \in \mathbb{Z}_p^3$ of order p, be a union of lines. Then D is a PDS if and only of $l_\chi = |\{i : L_i \subset ker\chi\}|$ has exactly two possible values over all nonprincipal character χ.*

In $\mathbb{Z}_p^3$, kerχ is a hyperplane and L_i is a line. We can think of relationship between kerχ,L_i, and "$L_i \subset \ker\chi$" as a line-plane incident structure. There already exists such the incident structure in $\mathbb{Z}_p^3$ we need: the Desaurgesian projective plane PG$(2,p)$. Each point in PG$(2,p)$ corresponds to a line L_i in $\mathbb{Z}_p^3$, and each line in PG$(2,p)$, which consists of points in PG$(2,p)$, corresponds to a hyperplane in $\mathbb{Z}_p^3$. A point m_i in PG$(2,p)$ is an element of a line P_j in PG$(2,p)$ if and only if the line in $\mathbb{Z}_p^3$ represented by m_i is contained in the plane in $\mathbb{Z}_p^3$ represented by P_j. There are p^2+p+1 points $m_1, m_2, ..., m_{p^2+p+1}$ and p^2+p+1 lines $P_1, P_2, ..., P_{p^2+p+1}$ in PG$(2,p)$.

The incident structure can be represented by (p^2+p+1)-by-(p^2+p+1) matrix A, where $a_{ij} = \begin{cases} 1 & \text{if } m_j \in P_i \\ 0 & \text{otherwise} \end{cases}$ If we have $D = (\bigcup_{i=1}^{j} L_i)\backslash\{(0,0,0)\}$, we can represent D by a (p^2+p+1)-by-1 column vector x where $x_{j1} = \begin{cases} 1 & \text{if } L_j \subset D \\ 0 & \text{otherwise} \end{cases}$ The ith entry in column vector $A\mathbf{x}$ is the number of $L_j \subset D$ such that $L_j \subset P_i$. Since all possible kerχ are exactly all P_i, the statement that $l_\chi = |\{i : L_i \subset \ker\chi\}|$ has exactly two possible values over all nonprincipal characters χ is equivalent to saying that the components of $A\mathbf{x}$ have only two possible values. To show that this is impossible, we first show a weaker claim: that $A\mathbf{x}$ can't have only one value, unless $\mathbf{x}$ corresponds to some trivial subset of $\mathbb{Z}_p^3$. Define $\vec{\mathbf{0}}$ to be a (p^2+p+1)-by-1 column vector with entries 0, and $\vec{\mathbf{1}}$ to be a (p^2+p+1)-by-1 column vector with entries 1.

Claim 34. Let A be a (p^2+p+1)-by-(p^2+p+1) matrix representing point-line incident structure in PG$(2,p)$. If $A\mathbf{x} = a\vec{\mathbf{1}}$ for some integer a and a column vector $\mathbf{x}$ with components $x_i \in \{-m, -m+1, ..., n-1, n\}$, then $\mathbf{x} = b\vec{\mathbf{1}}$ for some integer $b \in \{-m, -m+1, ..., n-1, n\}$.

Proof. Let t be the sum of components in $\mathbf{x}$. Each value 1 in $\mathbf{x}$ will increase the sum of components in $A\mathbf{x}$ by $p+1$, since each point in PG$(2,p)$ is in $p+1$ lines in PG$(2,p)$. Therefore, the sum of components in $A\mathbf{x}$ is $t(p+1)$. The sum of components in $a\vec{\mathbf{1}}$ is $a(p^2+p+1)$. As $t(p+1) = a(p^2+p+1)$, and $\gcd(p+1, p^2+p+1) = \gcd(p+1, 1) = 1$, we must have $p^2+p+1|t$ and $p+1|a$. Let $t = b(p^2+p+1)$ for some integer b, and hence $a = b(p+1)$. Multiply by A^T on the left of $A\mathbf{x} = a\vec{\mathbf{1}}$ and use $A^TA = AA^T = p\mathbf{I} + \mathbf{J}$, where $\mathbf{I}$ is the $(p^2+p+1) \times (p^2+p+1)$ identity matrix and $\mathbf{J}$ is the $(p^2+p+1) \times (p^2+p+1)$ matrix with all entries being one [5, Lemma 28] to get

$$\begin{aligned}
A^TA\mathbf{x} &= b(p+1)A^T\vec{\mathbf{1}} \\
(p\mathbf{I}+\mathbf{J})\mathbf{x} &= b(p+1)(p+1)\vec{\mathbf{1}} \\
p\mathbf{x}+\mathbf{J}\mathbf{x} &= b(p^2+2p+1)\vec{\mathbf{1}} \\
p\mathbf{x}+b(p^2+p+1)\vec{\mathbf{1}} &= b(p^2+2p+1)\vec{\mathbf{1}} \\
p\mathbf{x} &= bp\vec{\mathbf{1}} \\
\mathbf{x} &= b\vec{\mathbf{1}}
\end{aligned}$$

□

Remark 35. Claim 34 is also true for $\mathrm{PG}(2,q)$ in general (with the base field being $\mathrm{GF}(q)$ where q is a power of prime), not just in $\mathrm{PG}(2,p)$. The proof follows exactly the same.

Now we show that $A\mathbf{x}$ cannot have only two possible values either.

Lemma 36. *Let A be a (p^2+p+1)-by-(p^2+p+1) matrix representing point-line incident structure in $PG(2,p)$. If $A\mathbf{x} = a\vec{\mathbf{1}} + b\mathbf{y}$ for some integer a, b and $b \neq 0$, and for some (p^2+p+1)-by-1 zero-one column vector $\mathbf{y}$, then either*

(1) $\mathbf{x} = \vec{\mathbf{1}}$ *or* $\vec{\mathbf{0}}$
(2) $\mathbf{x}^T$ *is identical to one of the rows of A, or the complement of one of the rows of A*

Proof. Let t and s be the number of $'1's$ in $\mathbf{x}$ and $\mathbf{y}$, respectively. We multiply $A\mathbf{x} = a\vec{\mathbf{1}} + b\mathbf{y}$ on the left with A^T. We know that $A^TA = AA^T = pI + J$, where I is the $(p^2+p+1)\times(p^2+p+1)$ identity matrix and J is the $(p^2+p+1)\times(p^2+p+1)$ matrix with all entries being one [5, Lemma 28]. Therefore, we get

$$(pI + J)\mathbf{x} = aA^T\vec{\mathbf{1}} + bA^T\mathbf{y}$$

Since each row of A^Thas sum $p+1$, $A^T\vec{\mathbf{1}} = (p+1)\vec{\mathbf{1}}$. Also, $(pI+J)\mathbf{x} = pI\mathbf{x} + J\mathbf{x} = p\mathbf{x} + t\vec{\mathbf{1}}$. Therefore,

$$\begin{aligned} p\mathbf{x} + t\vec{\mathbf{1}} &= a(p+1)\vec{\mathbf{1}} + bA^T\mathbf{y} \\ p\mathbf{x} - bA^T\mathbf{y} &= (a(p+1)-t)\,\vec{\mathbf{1}} \qquad (3.1) \end{aligned}$$

Equation 3.1 is over $\mathbb{Z}$. Reduce every entry in 3.1 over $\mod p$. We get

$$bA^T\mathbf{y} = (t-a)\vec{\mathbf{1}} \mod p \qquad (3.2)$$

Since the entries in $A\mathbf{x}$ are in the range $\{0,1,...,p+1\}$, we have $b \in \{0,1,...,p+1\}$. If $b=0$, then we have $A\mathbf{x} = a\vec{\mathbf{1}}$, which by Claim 34 implies $\mathbf{x} = \vec{\mathbf{1}}$ or $\vec{\mathbf{0}}$. If $b = p$, then the entries of $A\mathbf{x}$ are either 1) $p+1$ and 1, or 2) p and 0.

In the first case, suppose m rows of $A\mathbf{x}$ have values $p+1$, and the rest have value 1. If $2 \le m \le p^2+p$, choose two distinct rows r_1, r_2 that have value $p+1$ and one row r_3 that has value 1. Label columns of A by $\{1,2,...,p^2+p+1\}$. Let C_i be the set of all columns that the row i has '1' in. Because of projective plane structure, $|C_i| = p+1$, $|C_i \cap C_j| = 1$, and $|C_i \cap C_j \cap C_k| = 0$ for distinct rows i,j,k. In order for rows r_1, r_2 to have value $p+1$, $\mathbf{x}$ must be 1 at all rows in C_{r_1} and C_{r_2}. There are two distinct columns $c_1 \in C_{r_1} \cap C_{r_3}$ and $c_2 \in C_{r_2} \cap C_{r_3}$. Since $\mathbf{x}$ has value 1 at rows c_1, c_2, $\mathbf{x}$ must have value at least 2 at row r_3. But this is a contradiction to row r_3 being chosen to have value 1. Hence, $m = 0, 1, p^2+p+1$. If $m=0$ or p^2+p+1, all components of $A\mathbf{x}$ have the same value. By Claim 34 ,$\mathbf{x} = \vec{\mathbf{1}}$ or $\vec{\mathbf{0}}$. If $m=1$, then $\mathbf{x}^T$ is identical to one of the rows of A.

In the second case, define $\mathbf{x}' = \vec{\mathbf{1}} - \mathbf{x}$, a complement of $\mathbf{x}$. Then $A\mathbf{x}'$ has the property that its entries are either $p+1$ or 1, so by the first case, $\mathbf{x}'$ must be $\vec{\mathbf{1}}$,$\vec{\mathbf{0}}$, or identical to one of the rows of A.

Finally, if $b \neq 0, p$, then $\gcd(b,p) = 1$, and so equation 3.2 $\mod p$ is

$$A^T\mathbf{y} = ((t-a)/b)\vec{\mathbf{1}} \mod p$$

where b^{-1} is multiplicative inverse in $\mod p$. As components of $A^T\mathbf{y}$ are in $\{0,1,...,p+1\}$, they either

(1) have same values
(2) are 0 or p, or
(3) are 1 or $p+1$

The case 1) is dealt with the same way as in Claim 34. Cases 2) and 3) are exactly the same as what we just did. □

Note what the result that $\mathbf{x} = \vec{\mathbf{1}}$ or $\vec{\mathbf{0}}$ or $\mathbf{x}^T$ is identical to one of the rows of A, or complement of one of the rows of A means. $\mathbf{x} = \vec{\mathbf{1}}$ or $\vec{\mathbf{0}}$ implies $D = \mathbb{Z}_p^3$ or ϕ, the trivial PDSs. $\mathbf{x}^T$ being identical to one of the rows of A means that D has exactly all lines contained in one particular plane, so D is a plane with the identity taken out. We can summarize everything in the following theorem:

Theorem 37. *Let $D \subset \mathbb{Z}_p^3$ be a union of lines with no identity element. If D is a PDS, then D is either a subgroup of $\mathbb{Z}_p^3$ with identity taken out, or the complement of a subgroup of $\mathbb{Z}_p^3$.*

3.2. Nonexistence result of PDS on $\mathbb{Z}_p^3$ for $p \equiv 3 \mod 4$.

If D is a PDS in elementary abelian group G, there is a study on automorphisms of G that will fix D. This can be found in [7, Section 4].

Definition 38. Let σ be an automorphism of $G = \mathbb{Z}_p^t$, and D be a PDS in G. σ is called a **multiplier** of D if $\sigma(D) = D$.

For example, when D is a PDS, because $x \in D$ if and only if $-x \in D$, $\sigma : x \mapsto -x$ is always a multiplier of D (See Lemma 30).

Theorem 39. *Let $t \in \mathbb{Z}_p \backslash \{0\}$. Let σ_t be an automorphism $\sigma : x \mapsto tx$ of $G = \mathbb{Z}_p^n$. Let D be a PDS in G. If t is a square modulo p, then σ_t is a multiplier of D. [7, Theorem 4.1]*

There are $\frac{p-1}{2}$ squares in $\mathbb{Z}_p$ for odd prime p. If D is a PDS in $G = \mathbb{Z}_p^n$, consider each line $\langle a \rangle \subset G$. If one element $x \in \langle a \rangle$ is in D, then for all non-zero squares q in $\mathbb{Z}_p$, $xq \in \langle a \rangle$ is also in D. We may partition $\langle a \rangle \backslash \{\mathbf{0}\}$ into two parts: $P_1 = \{aq : q \text{ is a non-zero square in } \mathbb{Z}_p\}$ and the rest $P_2 = \langle a \rangle \backslash P_1 \cup \{\mathbf{0}\}$. D must either contain every element in P_1 or nothing in P_1. Similarly D either contains everything in P_2 or nothing in P_2, since any pair of two elements in P_2 can be mapped to one another by σ_t for some square t).

By Lemma 30, σ_{-1} is a multiplier of D. If $p \equiv 1 \mod 4$, -1 is a square, so by Theorem 39, σ_{-1} is a multiplier of D. This means Lemma 30 does not give us anything new apart from Theorem 39.

However, if $p \equiv 3 \mod 4$, Lemma 30 and Theorem 39 give different tools that can be combined to know the structure of D.

Lemma 40. *Let D be a PDS in $G = \mathbb{Z}_p^n$. If $p \equiv 3 \mod 4$, then D is a union of punctured lines, i.e. $D = (\bigcup_{i=1}^{j} L_i) \backslash \{\mathbf{0}\} \subset \mathbb{Z}_p^n$ where $L_i = \langle a_i \rangle$ for some element $a_i \in G$ of order p and $j \geq 0$.*

Proof. From Theorem 39, σ_t is a multiplier of D for all t nonzero squares in $\mathbb{Z}_p$. From Lemma 39, σ_{-1} is a multiplier of D. By definition, it is obvious to see that if σ_a, σ_b are multipliers of D, then $\sigma_a \circ \sigma_b = \sigma_{ab}$ is also a multiplier of D. As we range q over all non-zero squares of $\mathbb{Z}_p$, $(-1)q$ will range over all nonzero non-squares of $\mathbb{Z}_p$, so σ_t is a multiplier for t nonzero non-squares as well. Since σ_t is a multiplier for all $t \in \mathbb{Z}_p \backslash \{0\}$, this means for each punctured line $\langle a \rangle \backslash \{\mathbf{0}\} \subset G$, D either contains the whole punctured line or nothing in that line. $\square$

For $G = \mathbb{Z}_p^n$ with $n = 3$, combining this result with Theorem 37 gives the nonexistence result.

Theorem 41. *If D is a PDS in $G = \mathbb{Z}_p^3$ for $p \equiv 3 \mod 4$, then D is either a subgroup of $\mathbb{Z}_p^3$ with identity taken out, or the complement of a subgroup of $\mathbb{Z}_p^3$.*

3.3. **Attempts on the search of all PDS in $\mathbb{Z}_p^3$ for $p \equiv 1 \mod 4$.** In the case $p \equiv 1 \mod 4$, Theorem 39 only tells us that a PDS $D \in \mathbb{Z}_p^3$ is a union of "half lines" - for each punctured line $\langle a \rangle$ which splits into two parts, D may contain all or nothing of each part. We, in fact, do not expect any nonexistence result, since there is a construction of PDS on $\mathbb{Z}_p^3$ with $p \equiv 1 \mod 4$: it is known that taking D as nonzero squares in any finite field F_q gives D a PDS in additive group in F_q for $q \equiv 1 \mod 4$ [8, Proposition 3.5].

There are some approaches we attempted that seem to be general to all primes $p \equiv 1 \mod 4$, but the problem reduces solving a {-1,1}-matrix equation with some integer constraint. We were unable to solve the problem at the end. We tried an exhaustive search on $p = 5$, which made us suspect that solutions may be harder to categorize than we thought.

3.3.1. *The matrix approach.* For each punctured line $L_a = \langle a \rangle \backslash \{\mathbf{0}\} \in G = \mathbb{Z}_p^3$, we split L_a into two parts: $Q_a = \{aq : q \text{ is a non-zero square in } \mathbb{Z}_p\}$ and the rest $R_a = \langle a \rangle \backslash Q_a \cup \{\mathbf{0}\}$. For each punctured line L_a, there are 4 possibilities of $D \cap L_a$:L_a, Q_a, R_a,or $\emptyset$. If we know which of these 4 possibilities is for all $p^2 + p + 1$ lines in G, then we have completely determined D. This motivates us to "encode" D into $p^2 + p + 1$ column vector, each row corresponding to a line in G.

Second, we try to represent nonprincipal χ by $K = \ker\chi$, and see how K interacts with all $p^2 + p + 1$ lines in G. Even if we know K, which is a hyperplane in G, we still have to specify what character value of each coset of K is. Formally, write $K = \langle (a, b, c) \rangle^\perp$. We can define cosets of K to be $K_i = \{(x, y, z) \in G : ax + by + cz = i\}$ for $i = 0, 1, ..., p - 1$. There are still $p - 1$ possible nonprincipal characters with $\ker\chi = K$: χ can map elements in K_1 to ω^t where $\omega = e^{2\pi i/p}$ for any choice of $t = 1, 2, ..., p - 1$.

It turns out that, due to the structure of D being unions of half lines, we do not need to consider all $p - 1$ nonprincipal characters after fixing $K = \langle (a, b, c) \rangle^\perp$. Let $L = \langle (x, y, z) \rangle$ be a line in G. Let S be a set of all nonzero squares in $\mathbb{Z}_p$. Consider the following possibilities (all equations are under $\mod p$):

(1) $ax + by + cz = 0$
(2) $ax + by + cz \in S$
(3) $ax + by + cz \notin S$ and not zero

There are 4 possibilities for $D \cap L_a$. Each combination of possibilities give different value of $\chi(D \cap L_a)$. See Table 6. First and last columns in Table 6 is obvious. The first row means $\ker\chi$ is principal on L_a.

By Table 6, we only have to care two possible values of t: $t \in S$ or $t \notin S$ The table shows that a character value on a half line can either be $Q = \sum_{i \in S} \omega^i$ or $R = \sum_{i \in \mathbb{Z}_p \backslash (S \cup \{0\})} \omega^i$. Choosing a different t (from $t \in S$ to $t \notin S$ or the other way around) will switch value Q to R, and vice versa. Before we move on to the

TABLE 6. Values of $\chi(D \cap L_a)$ for ker$\chi = \langle (a,b,c) \rangle^{\perp}$ and $L_a = \langle (x,y,z) \rangle$

Cases	$D \cap L_a = Q_a \cup R_a$	$D \cap L_a = Q_a$	$D \cap L_a = R_a$	$D \cap L_a = \emptyset$
$ax+by+cz=0$	$p-1$	$\frac{p-1}{2}$	$\frac{p-1}{2}$	0
$ax+by+cz \in S$	-1	$\sum_{i \in S} \omega^{ti}$	$\sum_{i \in \mathbb{Z}_p \setminus (S \cup \{0\})} \omega^{ti}$	0
$ax+by+cz \notin S$ and not zero	-1	$\sum_{i \in \mathbb{Z}_p \setminus (S \cup \{0\})} \omega^{ti}$	$\sum_{i \in S} \omega^{ti}$	0

Note: ω^t is the root of unity that χ sends elements of K_1 to.

construction of the matrix equation, we can actually compute exactly what Q, R are. We state it here without the proof.

Lemma 42. *Let $p \equiv 1 \mod 4$, and $S \subset \mathbb{Z}_p^*$ be a set of all nonzero squares modulo p. Then, $Q = \sum_{i \in S} \omega^i = \frac{-1+\sqrt{p}}{2}$ and $R = \sum_{i \in \mathbb{Z}_p \setminus (S \cup \{0\})} \omega^i = \frac{-1-\sqrt{p}}{2}$.*

Construct a (p^2+p+1)-by-(p^2+p+1) matrix B by labeling each column with lines $\langle (x,y,z) \rangle$ and rows with plane $\langle (a,b,c) \rangle^{\perp}$, and define

$$b_{<(a,b,c)>^{\perp},<(x,y,z)>} = \begin{cases} 1 & , ax+by+cz \in S \\ -1 & , ax+by+cz \neq 0, ax+by+cz \notin S \\ 0 & , ax+by+cz = 0 \end{cases}$$

Define a (p^2+p+1) column vector $\mathbf{x} = \mathbf{x}_D$ by labeling each row with lines $\langle (x,y,z) \rangle$, and define the component of each row i

$$x_i = \begin{cases} 1 & , D \cap L_a = Q_a \\ -1 & , D \cap L_a = R_a \\ 0 & , \text{otherwise} \end{cases}$$

where a is the element generating line i. For simplicity we can define so that row i and column i are always orthogonal of each other, and that column i of B is the same label (same line) as row i of $\mathbf{x}$ (if at row i of B it is $\langle (a,b,c) \rangle^{\perp}$, then the column i of B and row i of $\mathbf{x}$ are $\langle (a,b,c) \rangle$).

Consider each row of the column vector $\mathbf{z} = B\mathbf{x}$. Fix row i. To compute row i of $\mathbf{z}$, and say row i of B represents $\langle (a,b,c) \rangle^{\perp}$, we need to compute $\sum_{j=1}^{p^2+p+1} b_{ij} x_j$. For each term $b_{ij} x_j$, this is 1 for each column $j = \langle (x,y,z) \rangle$ of B such that

- $ax+by+cz \in S$ and $D \cap L_{(x,y,z)} = Q_{(x,y,z)}$, or
- $ax+by+cz \neq 0, ax+by+cz \notin S$ and $D \cap L_{(x,y,z)} = R_{(x,y,z)}$

From Table 6, this corresponds to the case that character value of a half line evaluated to $\sum_{i \in S} \omega^{ti}$. Similarly, $b_{ij} x_j$ is -1 for each column j if character value on a half line is $\sum_{i \in \mathbb{Z}_p \setminus (S \cup \{0\})} \omega^{ti}$. If we add 0, this means we add an integral number -1 or $p-1$.

As we sum $b_{ij} x_j$ (which are -1,0,1) over all columns j of B to get component at row i of $\mathbf{z}$, we are fixing $K = \ker\chi$ to be exactly what row i of B represents, and "add" value of character on all p^2+p+1 half lines, lines, and empty sets that D have. The component z_i of $\mathbf{z} = B\mathbf{x}$ represents something about $\chi(D)$. It does not represent exactly what $\chi(D)$ is, since we ignore the rational part of $\chi(D)$. It, however, tells that the number of times $\sum_{i \in S} \omega^{ti}$ is produced in $\chi(D)$ subtracted by number of times $\sum_{i \in \mathbb{Z}_p \setminus (S \cup \{0\})} \omega^{ti}$ is produced is z_i. That is, $\sum_{i \in S} \omega^{ti}$ term is produced $\frac{(p^2+p+1+z_i)}{2}$ and $\sum_{i \in \mathbb{Z}_p \setminus (S \cup \{0\})} \omega^{ti}$ is produced $\frac{(p^2+p+1-z_i)}{2}$ times. If

t is chosen so that $Q = \sum_{i \in S} \omega^{ti}$ and $R = \sum_{i \in \mathbb{Z}_p \setminus (S \cup \{0\})} \omega^{ti}$, then the irrational part of $\chi(D)$ is determined by $\frac{(p^2+p+1+z_i)}{2}Q + \frac{(p^2+p+1-z_i)}{2}R = \frac{p^2+p+1}{2}(Q+R) + \frac{z_i}{2}(Q-R) = \frac{p^2+p+1}{2}(-1) + \frac{z_i}{2}\sqrt{p}$, which means $\chi(D) = \frac{\beta_i}{2} + \frac{z_i}{2}\sqrt{p}$ for some integer β_i. If t is chosen so that $R = \sum_{i \in S} \omega^{ti}$ and $Q = \sum_{i \in \mathbb{Z}_p \setminus (S \cup \{0\})} \omega^{ti}$, we would have the irrational part of $\chi(D)$ be the same as $\frac{(p^2+p+1+z_i)}{2}R + \frac{(p^2+p+1-z_i)}{2}Q = \frac{p^2+p+1}{2}(-1) - \frac{z_i}{2}\sqrt{p}$ instead., which means $\chi(D) = \frac{\beta_i}{2} - \frac{z_i}{2}\sqrt{p}$ for some integer β_i.

At this point, since we know $\chi(D)$ may only take two possible values, there are two cases:

(1) $\chi(D)$ takes two distinct rational values. This means z_i in $\mathbf{z}$ is all zero (for otherwise the term $\pm\frac{z_i}{2}\sqrt{p}$ we obtained earlier will make $\chi(D)$ irrational), and β_i takes two different integers as we range over p^2+p+1 possible $K = \ker\chi$.

(2) $\chi(D)$ takes two distinct irrational values. This can only happen when $z_i \neq 0$ for some i ($\sqrt{p}$ is the only possible irrational term we found). Then, $\chi(D) = \frac{\beta_i}{2} \pm \frac{z_i}{2}\sqrt{p}$ are two possible values, and so every nonprincipal character must have one of these two values.

Case 1: Since $\mathbf{z}$ is an all-zero column vector, we have an equation $B\mathbf{x} = \vec{\mathbf{0}}$. If B is invertible, we would have a solution for $\mathbf{x}$ easily, and it is.

Lemma 43. *Let B be a (p^2+p+1)-by-(p^2+p+1) matrix constructed by labeling each column with lines $\langle(x,y,z)\rangle$ and rows with plane $\langle(a,b,c)\rangle^{\perp}$, where row i and column j are orthogonal to each other, and define*

$$b_{<(a,b,c)>^{\perp},<(x,y,z)>} = \begin{cases} 1 & , ax+by+cz \in S \\ -1 & , ax+by+cz \neq 0, ax+by+cz \notin S \\ 0 & , ax+by+cz = 0 \end{cases}$$

Where S is the set of all nonzero squares modulo p. Then $B^2 = p^2\mathbf{I}$, where $\mathbf{I}$ is a (p^2+p+1)-by-(p^2+p+1) identity matrix.

Proof. First, observe that $B = B^T$, so the entries c_{ij} in $C = B^2$ is a dot product of two distinct rows i, j of B. If $i = j$, it is obvious that $c_{ij} = p^2$. Now let $i \neq j$, and denote i, j with $\langle(a_1,b_1,c_1)\rangle^{\perp}$ and $\langle(a_2,b_2,c_2)\rangle^{\perp}$, respectively. We know c_{ij} is the sum of $g((x,y,z)) = \left(\frac{(a_1,b_1,c_1)\cdot(x,y,z)}{p}\right) \times \left(\frac{(a_2,b_2,c_2)\cdot(x,y,z)}{p}\right)$ over all columns (x,y,z) of B (the $\cdot$ is a dot product of two vectors, and the parenthesis is a Lagrange symbol). Note that $g(s(x,y,z)) = g((x,y,z))$ for all $s \in F_p^*$, so $\sum_{t \in \langle(x,y,z)\rangle} g(t) = (p-1)g((x,y,z))$. Therefore, summing $g((x,y,z))$ over all $(x,y,z) \in F_p^3 \setminus \{\mathbf{0}\}$ will give the sum exactly $p-1$ times of summing $g((x,y,z))$ over columns (x,y,z) of B. Clearly $g(\mathbf{0}) = 0$. Therefore, $c_{ij} = \frac{\sum_{t \in F_p^3} g(t)}{p-1}$.

Clearly $\sum_{t \in F_p^3}\left(\frac{(a_1,b_1,c_1)\cdot(x,y,z)}{p}\right) = 0$: one hyperplane $H : (a_1,b_1,c_1)\cdot(x,y,z) = 0$ gives the term 0, half of $p-1$ cosets of that hyperplane give the term value 1, and the other half gives the term value -1. Consider each coset of the hyperplane $H' : (a_2,b_2,c_2)\cdot(x,y,z) = 0$. Each coset hH' intersects each all p cosets of H exactly at p points. Among these p intersection, $\frac{p-1}{2}$ times is when it intersects a coset of H which $\left(\frac{(a_1,b_1,c_1)\cdot(x,y,z)}{p}\right) = 1$, $\frac{p-1}{2}$ times on cosets of H which $\left(\frac{(a_1,b_1,c_1)\cdot(x,y,z)}{p}\right) =$

-1, and 1 time with $\left(\frac{(a_1,b_1,c_1)\cdot(x,y,z)}{p}\right) = 0$ (i.e. on H itself). If the coset hH' has $\left(\frac{(a_2,b_2,c_2)\cdot(x,y,z)}{p}\right) = 1$, $\left(\frac{(a_1,b_1,c_1)\cdot(x,y,z)}{p}\right)$ and $g((x,y,z))$ are identical, so the sum is the same over hH'. If $\left(\frac{(a_2,b_2,c_2)\cdot(x,y,z)}{p}\right) = -1$, $\left(\frac{(a_1,b_1,c_1)\cdot(x,y,z)}{p}\right)$ and $g((x,y,z))$ have opposite value. But the sum of $\left(\frac{(a_1,b_1,c_1)\cdot(x,y,z)}{p}\right)$ over hH' is originally 0, so the sum $g((x,y,z))$ over H' is also 0. If $\left(\frac{(a_2,b_2,c_2)\cdot(x,y,z)}{p}\right) = 0$, i.e. the coset hH' is H', then $g((x,y,z))$ is always 0 on hH'. In all cases, $g((x,y,z))$ has sum 0 over any coset hH'. Therefore, $\sum_{t\in F_p^3} g(t)$, and so $c_{ij} = 0$. □

By Lemma 43, the first case implies $\mathbf{x} = \vec{\mathbf{0}}$. This means for each line $\langle(x,y,z)\rangle \subset \mathbb{Z}_p^3$, D either contains the whole punctured line or nothing in that line. That is, D is a union of punctured lines. But by Theorem 37, D must be a trivial PDS, so we are done in this case.

Case 2: Since $\chi(D)$ is irrational, we know $\Delta = (\lambda-\mu)^2 + 4(k-\mu)$ is not a square (from Theorem 6). By [7, Corollary 6.3], D is a Paley type partial difference set with parameter $(v,k,\lambda,\mu) = (p^3, \frac{p^3-1}{2}, \frac{p^3-5}{4}, \frac{p^3-1}{4})$ (see [6] and [10] about Paley type partial difference sets).

Lemma 44. *Let D be a nontrivial PDS in $\mathbb{Z}_p^3$, with $p \equiv 1 \mod 4$. Then D is a Paley type PDS.*

As we know $\lambda = \frac{p^3-5}{4}, \mu = \frac{p^3-1}{4}$, so we can calculate $\chi(D) = \frac{-1\pm p\sqrt{p}}{2}$ by Theorem 6. Comparing this with our result $\chi(D) = \frac{\beta_i}{2} \pm \frac{z_i}{2}\sqrt{p}$, this means

(1) $\beta_i = -1$ for all row i of $\mathbf{z}$
(2) $z_i \in \{-p, p\}$ for all row i of $\mathbf{z}$

$\beta_i = -1$ across all i will give us some information about D, and we focus on z_i later on. We now focus on the rational part of $\chi(D)$, which should always be $-\frac{1}{2}$.

2.1 Using $\beta_i = -\frac{1}{2}$: focus on the rational part of $\chi(D)$. Suppose that we have fixed D, but we range over all nonprincipal χ. From Table 6 (fixing D is like we already fixed a column that we look at), there is no difference of the rational part whether $ax+by+cz \in S$ or not if $ax+by+cz = 0$. The difference only comes whether $ax+by+cz = 0$. If $|D \cap L_{(x,y,z)}| = \frac{p-1}{2}$, then the difference is $\frac{p}{2}$. If $|D \cap L_{(x,y,z)}| = p-1$, then the difference is p. This motivates us to use:

(1) Projective plane incident structure matrix A as define earlier. This captures the necessary and sufficient condition when $ax+by+cz = 0$ happens.
(2) Construct p^2+p+1 column vector $\mathbf{v} = \mathbf{v}_D$ by labeling each row i of $\mathbf{v}$ with all lines $\langle(x,y,z)\rangle$ in $\mathbb{Z}_p^3$, and define

$$v_i = |D \cap L_{(x,y,z)}|/(\frac{p-1}{2})$$

We know $D \cap L_{(x,y,z)}$ has only four possibilities $\emptyset, Q_{(x,y,z)}, R_{(x,y,z)}, L_{(x,y,z)}$, and so $v_i \in \{0,1,2\}$. $\mathbf{v}$ captures the size of $|D \cap L_{(x,y,z)}|$ which impacts the rational parts of $\chi(D)$.

Consider the product $\mathbf{w} = A\mathbf{v}$. To get component w_i of $\mathbf{w}$, we look at row $i = \langle(a,b,c)\rangle^\perp$ of A, and for each column $j = \langle(x,y,z)\rangle$ of A, we add the term $a_{ij}v_j$. By Table 6, the rational part of $\chi_{\langle(a,b,c)\rangle^\perp}(D \cap \langle(x,y,z)\rangle)$, where χ_K is the character with ker$\chi = K$, is the same as $\frac{pa_{ij}v_j - v_j}{2}$ (can be checked easily). From the character perspective, $\chi(D) = \sum_{\text{lines } \langle(x,y,z)\rangle \text{ in } \mathbb{Z}_p^3}(D \cap \langle(x,y,z,)\rangle)$ which has rational part $\sum_j \frac{pa_{ij}v_j - v_j}{2} = \sum_j \frac{pa_{ij}v_j}{2} - \sum_j \frac{v_j}{2}$. From the matrix equation, component $w_i = \sum_j a_{ij}v_j$. Therefore, the rational part of $\chi_{i=\langle(a,b,c)\rangle^\perp}(D)$ is $\frac{w_i}{2} - \sum_j \frac{v_j}{2} = \frac{w_i}{2} - \frac{|D|}{p-1}$. Since $|D|$ is fixed, and rational part of $\chi_i(D)$ is fixed across all i, w_i is fixed across all i. Since entries in A and $\mathbf{v}$ are integers, $\mathbf{w}$ must be integral, so we must have

$$A\mathbf{v} = d\vec{\mathbf{1}}$$

for some integer d. By Claim 34, $\mathbf{v}$ is either all-zero, all-one, or all-two column vector. All-zero and all-two vectors represent trivial D. The all-one vector means that D must be a union of half lines, with half line in each line of $\mathbb{Z}_p^3$. Note that under $v_j = 1$, we have the rational part of $\chi_i(D)$ to be $\sum_j \frac{pa_{ij}v_j}{2} - \sum_j \frac{v_j}{2} = p\sum_j \frac{a_{ij}}{2} - \frac{p^2+p+1}{2} = p\frac{p+1}{2} - \frac{p^2+p+1}{2} = -\frac{1}{2}$, which is exactly what it should be. Also, $|D| = \frac{p^3-1}{2}$. We state the result as follow (non-triviality means D is not a subgroup with identity taken out or complement of a subgroup).

Theorem 45. *Let D be a nontrivial PDS in $\mathbb{Z}_p^3$, with $p \equiv 1 \mod 4$. Then D is a union of half lines in each line of $\mathbb{Z}_p^3$. That is, for each line $\langle(a,b,c)\rangle \subset \mathbb{Z}_p^3$, $D \cap \langle(a,b,c)\rangle \in \{Q_{\langle(a,b,c)\rangle}, R_{\langle a,b,c\rangle}\}$, where $Q_{\langle(a,b,c)\rangle} = \{q(a,b,c) : q \text{ is a nonzero square modulo } p\}$ and $R_{\langle(a,b,c)\rangle} = \{r(a,b,c) : r \text{ is a nonzero nonsquare modulo } p\}$.*

2.2 Using $z_i \in \{-p, p\}$: focus on the irrational part of $\chi(D)$. Recall the equation

$$B\mathbf{x} = \mathbf{z}$$

By Theorem 45, $\mathbf{x}$ must be a $\{-1,1\}$-column vector. Also, $z_i \in \{-p, p\}$ means that $\mathbf{z} = p\mathbf{y}$ for some $\{-1,1\}$-column vector $\mathbf{y}$ of length p^2+p+1. The work we show so far proves that if D is a nontrivial PDS in $\mathbb{Z}_p^3$, then $\mathbf{x} = \mathbf{x}_D$ satisfies $B\mathbf{x} = p\mathbf{y}$ for some $\{-1,1\}$-column vector $\mathbf{y}$. Since B captures all possible kerχ (by looking all rows i, which represents all possible kerχ), $B\mathbf{x} = p\mathbf{y}$ is sufficient that the irrational part of $\chi(D)$ is in fact $\frac{\pm p\sqrt{p}}{2}$. Also, as we just noted earlier that $\chi(D)$ will always have a correct $-\frac{1}{2}$ rational part given $\mathbf{x}$ a $\{-1,1\}$-column vector, the equation $B\mathbf{x} = p\mathbf{y}$ is necessary and sufficient for D to be a nontrivial PDS.

Theorem 46. *Let $G = \mathbb{Z}_p^3$, and define matrices B and $\mathbf{x} = \mathbf{x}_D$ as earlier. Then there is a bijection between the set of all nontrivial PDS D and the set of solutions $(\mathbf{x}, \mathbf{y})$ of pairs of $\{-1,1\}$ column vectors such that*

$$B\mathbf{x} = p\mathbf{y}$$

Theorem 46 reduces finding PDS in $\mathbb{Z}_p^3$ to solving a matrix equation $B\mathbf{x} = p\mathbf{y}$ under some integer constraint.

3.4. **Solving a matrix equation $B\mathbf{x} = p\mathbf{y}$.** Unfortunately we are unable to find a general categorization of solutions $(\mathbf{x}, \mathbf{y})$ to

$$B\mathbf{x} = p\mathbf{y} \tag{3.3}$$

The following was the approach we did.

Motivated from Lemma 43, multiply B on the left of 3.3 to get

$$p\mathbf{x} = B\mathbf{y} \tag{3.4}$$

Adding and subtracting between 3.3 and 3.4 gives, respectively,

$$B(\mathbf{x}+\mathbf{y}) \quad = \quad p(\mathbf{x}+\mathbf{y}) \tag{3.5}$$

$$B(\mathbf{x}-\mathbf{y}) = -p(\mathbf{x}-\mathbf{y}) \tag{3.6}$$

Hence, $(\mathbf{x},\mathbf{y})$ is the solution to 3.3 if and only if $\mathbf{x}+\mathbf{y}$ and $\mathbf{x}-\mathbf{y}$ are eigenvectors of B with eigenvalues $p,-p$ respectively. The task seems obvious: find eigenvalues of B and all associated eigenspaces, and use those spaces of eigenvalues $p,-p$ as candidates for $\mathbf{x}+\mathbf{y}$ and $\mathbf{x}-\mathbf{y}$. The problem we found with $p=5$ is that B has two eigenvalues 5,-5, and the eigenspaces has dimension 15,16 respectively. The eigenspaces are too big to go through computationally. Furthermore, the constraint that $\mathbf{x},\mathbf{y}$ are {1,-1} column vectors also gives an "integer programming" sense in the problem we want to compute.

3.5. **Computer search result .** We tried to explicitly find all PDS in $\mathbb{Z}_p^3$ for $p=5$, with the hope that we may see some structure that can generalize to other p. We wrote a program to find all PDS in $\mathbb{Z}_5^3$ and find the size of stabilizers of each PDS found. For the actual code, see Algorithms 8 and 9 in Appendix A. We can get the number of equivalent PDS of a particular class to be the number of automorphism of G divided by the size of the stabilizer.

We found 3 numbers of nontrivial equivalent PDS sets on a class: 62000 (stabilizer size 24), 8000 (stabilizer size 186), and 12400 (stabilizer 120). The set of all nonzero squares in F_{125} has stabilizer of size 186. The number of nontrivial PDS found (ignoring equivalency) is 94800. This gives us high certainty that there is one PDS class of 62000 equivalent sets, one class of 8000 equivalent sets, and two classes of 12400 equivalent sets.

We later found a relevant literature. [13, Table 1] shows all Paley type set in group of order 125, which is $\mathbb{Z}_p^3$. They found only three equivalence classes, and each has automorphism group of size 15000 and 3000, which are different from our result. We are in the process of understanding Chen and Feng's paper, and checking our computations.

4. Summary and Future Direction

Results that are new are Theorem 41, 45, and 46 in Section 3 on PDS in $\mathbb{Z}_p^3$. There are many directions this research can move forward:

(1) The "7,12,17" and "1,2,3,4,2" patterns in 1.4.3,1.4.4, and 1.4.5 seem to point to some additive structure that needs an explanation. It may have a potential that leads to a new or a variant of construction of Cameron-Liebler line classes.

(2) We see that the ideas of using matrices (A,B) in Section 3 to encapsulate character equation in $\mathbb{Z}_p^3$ is partially successful. We can generalize this idea by looking at a similar structure of projective plane, but on a Galois Ring instead on a finite field. This may give us some insight about PDS in $\mathbb{Z}_{p^r}^3$ for $r \geq 2$.

(3) The search for all PDS in $\mathbb{Z}_p^3$ for $p \equiv 1 \mod 4$ is equivalent to solving $B\mathbf{x} = p\mathbf{y}$ in 3.4. There is still potential result by examining the structure of B in order to find an efficient algorithm to finding $(\mathbf{x},\mathbf{y})$. Algorithms may count or give more example of solutions $(\mathbf{x},\mathbf{y})$. If the algorithms seems hard to find, one may also try to define the problem precisely and put the

problem into some computational complexity class. One other potential direction is to describe each set D in terms of $\mathbf{x}_D$ instead of D. There is a potential that a new "matrix" view of finding PDS instead of from the viewpoint of groups or vector spaces may lead to new insight of PDSs. For example, one may examine how two equivalent PDSs D_1, D_2 in $\mathbb{Z}_p^3$ relate to one another in the matrix viewpoint (i.e. see how $\mathbf{x}_{D_1}$ relates to $\mathbf{x}_{D_2}$). In general, what will the equivalence relations of set D that is defined from automorphism on $\mathbb{Z}_p^3$ be in the space of $\{-1, 1\}$ column vector $\mathbf{x}_D$? Is there any obvious transformation of solution $\mathbf{x}_D$ to another solution of $B\mathbf{x} = p\mathbf{y}$, and if so, what would that transformation in the space of $\{-1, 1\}$ column vector $\mathbf{x}_D$ look like in the $\mathbb{Z}_p^3$ space?

(4) The computer result in 3.5 still needs some more study to explain the difference between our result and the paper [13, Table 1].

(5) Describe each equivalence class of D in $\mathbb{Z}_5^3$ found in 3.5 as its discrete log: $\log D = \{\log_\omega d : d \in D\}$ where $\log_\omega$ is the discrete log in finite field F_{125}^*, and try to find any pattern in $\log D$ for each specific class of PDS, hoping that some will generalize to $\mathbb{Z}_p^3$ for general $p \equiv 1 \mod 4$.

References

[1] James A Davis. Partial difference sets in p-groups. *Arch. Math.*, 63:103–110, 1994.

[2] Richard Elman, Nikita Karpenko, and Alexander Merkurjev. The algebraic and geometric theory of quadratic forms. Online.

[3] Patrick Govaerts and Leo Storme. On cameron-liebler line classes. *Advances in Geometry*, 4:279–286, 2004.

[4] Jr. John B. Polhill. *Constructing Difference Sets and Partial Difference Sets Using Galois Rings*. PhD thesis, University of Virginia, April 1999.

[5] Johan Kahrstrom. On projective planes. Mid Sweden University, online, February 2002.

[6] Ka Hin Leung and Siu Lun Ma. Partial difference sets with paley parameters. *Bull. London Math. Soc.*, 27(6):553–564, 1995.

[7] S. L. Ma. A survey of partial difference sets. *Designs, Codes, and Cryptography*, 4:221–261, 1994.

[8] S.L. Ma. Partial difference sets. *Discrete Mathematics*, 52:75–89, 1984.

[9] S. E. Payne. Topics in finite geometry: Ovals, ovoids and generalized quadrangles. University of Colorado Denver, class note, May 2007.

[10] John Polhill. Paley type partial difference difference sets in non p-groups. *Designs, Codes, and Cryptography*, 52:163–169, 2009.

[11] Morgan J. Rodgers. *On Some New Examples Of Cameron-Liebler Line Classes*. PhD thesis, University of Colorado Denver, 2012.

[12] Koji M Tao F and Qing X. Cameron-liebler line classes with parameter $x = \frac{q^2-1}{2}$. arXiv:1406.6526v2, February 2015.

[13] Tao Feng Yu Qing Chen. Paley type sets from cyclotomic classes and arasudillon-player difference sets. *Designs, Codes, and Cryptography*, 74(3):581–600, 2013.

5. Appendix A: Codes Of Algorithms

This section contains all the actual relevant codes used in this work. The programming languages used are Mathematica and Java. If a description of an algorithms in this section does not specify the language, it is Mathematica.

Algorithms 1: Mathematica functions

The following codes are Mathematica function. These should be run before any other run of later Mathematica Algorithms in this paper.

```
(*getting the set K in polynomial form*)
halfValueForK={x^2+2x+2,4x^2+3,x^2+3x+2,3x^2+2x+1,x^2+x+2,2x};
secondHalfOfK=PolynomialMod[3*halfValueForK ,5];
valueForK=Union[halfValueForK ,secondHalfOfK];
(*temp2=PolynomialMod[4*valueForK ,5];
temp=Union[valueForK ,temp2];
valueForK=temp*) (*only when what all multiples 1,2,3,4*)
(*Essential Function used*)
EvaluateInField[y_]:=PolynomialMod[PolynomialMod[y,x^3+3x+2],5];
Tra[y_]:=PolynomialMod[PolynomialMod[y+y^5+y^(25),x^3+3x+2],5];
(*More functions helping the program*)
list[a_,b_,K_]:=Table[{n,Mod[Tra[a*x^n]+Tra[b*x^(124-n)*K],5]},{n,0,123,4}]
countZero[list_]:=Module[{zero=0},For[i=1,i<=Length[list],i++,
    If[list[[i]][[2]]==0,zero++;, ]
    ];zero]
sumResult[list_]:=Module[{sum=0},For[i=1,i<=Length[list],i++,
   sum=list[[i]][[2]]+sum;
    ];sum]

(*constructing difference set in polynomial form*)
timeStart=AbsoluteTime[];
DSetPolyForm=Module[{set={},firstComponent,secondComponent,y},For[i=1,i<=4,i++,
    For[j=0,j<= 30,j++,
     For[k=1,k<=Length[valueForK],k++,
      y=EvaluateInField[x^(4j)];
      firstComponent=EvaluateInField[i*y];
      secondComponent=EvaluateInField[i*x^(124-4j)*valueForK[[k]]];
      AppendTo[set,firstComponent*x^3+secondComponent]
      ]
     ]
    ];set]
timeUsed=AbsoluteTime[]-timeStart (*Display time used*)

(*Turning DSetPolyForm into vector form and see it in other fields*)
TurnToVector[A_]:={Coefficient[A,x,0]  ,Coefficient[A,x,1]  ,Coefficient[A,x,2]  ,
      Coefficient[A,x,3]   ,Coefficient[A,x,4]  ,Coefficient[A,x,5]};
TurnToVector3[A_]:={Coefficient[A,x,0]  ,Coefficient[A,x,1]  ,Coefficient[A,x,2]  };
DSetVectorForm = Module[{theVectors={}},For[i=1,i<=Length[DSetPolyForm],i++,
   AppendTo[theVectors,TurnToVector[DSetPolyForm[[i]]]]];theVectors]
TurnSetToVector3[theSet_] :=     Module[{theVectors = {}}, For[i = 1, i <= Length[
      theSet], i++,
          AppendTo[theVectors, TurnToVector3[theSet[[i]]]]]; theVectors];


(*breakdown in any coordinates I choose*)
Breakdown3FixedCoor[coor1_,a_,coor2_,b_,coor3_,c_,theSetToBreak_]:=
  Module[{theVectors={}},For[i=1,i<=Length[theSetToBreak],i++,
     If[ theSetToBreak[[i]][[coor1]]==a&&theSetToBreak[[i]][[coor2]]==b&&
          theSetToBreak[[i]][[coor3]]==c,AppendTo[theVectors,theSetToBreak[[i]] ] ]
     ]; theVectors];
GetListBreakdown3FixedCoor[coor1_,a_,coor2_,b_,coor3_,c_,theSetToBreak_]:=Module
      [{theSet=Breakdown3FixedCoor[coor1,a,coor2,b,coor3,c,theSetToBreak]},For[i=1,
      i<= Length[theSet],i++,
    theSet[[i]]=Delete[theSet[[i]],{{coor1},{coor2},{coor3}}];
    ];theSet]
Breakdown3FixedCoor[1,1,3,1,5,1,DSetVectorForm]
GetListBreakdown3FixedCoor[1,1,3,1,5,1,DSetVectorForm]
```

Algorithms 2: Finding the number of solutions $y \in \langle\omega^4\rangle$ to $\mathrm{Tr}(ay + by^{-1}k) = 0$ given $a, b \in F^*_{125}$ and $k \in K$

The actual run is at the end. The first two methods are defining functions needed.

```
(*The counting solution part. Fixing k in K and or y in <w^4> and see how many
    solutions there are, and also try to see the 7,12,17 pattern*)
countSolInK[a_,b_,n_,K_]:=Module[{numSol=0},For[i=1,i<=Length[K],i++,
  If[Mod[Tra[a x^n]+Tra [b x^(124-n) K[[i]]],5]==0,numSol++];];numSol]
Table[countSolInK[2,2,4i,valueForK],{i,0,123,4}]

(*matching where k and 3k is. e.g. position 1 with 2, and 3 with 9, and 4 with 7,
     and so on, inside set K we created*)
sumInPairs[list_]:={list[[1]][[2]]+list[[2]][[2]],list[[3]][[2]]+list[[9]][[2]],
    list[[4]][[2]]+list[[7]][[2]],list[[5]][[2]]+list[[10]][[2]],list[[6]][[2]]+
    list[[11]][[2]],list[[8]][[2]]+list[[12]][[2]]}
allNumSol[a_,b_]:=Module[{c},c=Table[{valueForK[[i]],countZero[list[a,b,valueForK
    [[i]]]]},{i,1,Length[valueForK]}];first={c[[2]][[2]],c[[3]][[2]],c[[4]][[2]],
    c[[5]][[2]],c[[11]][[2]],c[[8]][[2]]};
  second={c[[1]][[2]],c[[9]][[2]],c[[7]][[2]],c[[10]][[2]],c[[6]][[2]],c
      [[12]][[2]]};Print[first];Print[second];Print[sumInPairs[c]];c]
numOfYKPairs[a_,b_]:=sumResult[allNumSol[a,b]]


For[i=0,i<=123,i++,Print["a=x^0 b=x^",i];Print[allNumSol[1,x^i]]];
For[i=0,i<=123,i++,Print["a=x^1 b=x^",i];Print[allNumSol[x,x^i]]];
For[i=0,i<=123,i++,Print["a=x^2 b=x^",i];Print[allNumSol[x^2,x^i]]];
(*We can keep on this for other a as well. But this takes long time, so we can
    stop here*)
```

Algorithms 3: Breaking down a PDS D in Section 1 by fixing the first 2, 3, or 4 coordinates

Breaking into 2 coordinates:

```
(*now splitting the DSetVectorForm based on the first two coordinates*)
For[i = 0, i <= 4, i++,
 For[j = 0, j <= 4, j++,
  Print["Check for first two coordinates being (", i, ",", j, "):"];
  DSetVectorFormFixedTwoCoordinates[i, j] =
   Module[{theVectors = {}},
    For[k = 1, k <= Length[DSetVectorForm], k++,
     If[ DSetVectorForm[[k]][[1]] == i &&
        DSetVectorForm[[k]][[2]] == j,
       AppendTo[theVectors, DSetVectorForm[[k]]] ];
     ]; theVectors];
  Print["Size of the list is: ",
   Length[DSetVectorFormFixedTwoCoordinates[i, j]]];
  Print[DSetVectorFormFixedTwoCoordinates[i, j]];
  ]
 ]
(*Try to break down 2 more coordinates. Choose one set and break down into 25
    sets*)
BreakFromMiddleTwoCoordinates[listOfVectors_] :=
  For[i = 0, i <= 4, i++,
   For[j = 0, j <= 4, j++,
    Print["Check for third and fourth coordinates being (", i, ",", j,
      "):"];
    DSetVectorFormFixedFourCoordinates[i, j] =
     Module[{theVectors = {}},
      For[k = 1, k <= Length[listOfVectors], k++,
       If[
         listOfVectors[[k]][[3]] == i && listOfVectors[[k]][[4]] == j,
          AppendTo[theVectors, listOfVectors[[k]]] ];
       ]; theVectors];
    Print["Size of the list is: ",
     Length[DSetVectorFormFixedFourCoordinates[i, j]]];
    Print[DSetVectorFormFixedFourCoordinates[i, j]];
    ]
   ];
BreakFromMiddleTwoCoordinates[DSetVectorFormFixedTwoCoordinates[0, 0]];
BreakFromMiddleTwoCoordinates[DSetVectorFormFixedTwoCoordinates[0, 1]];
BreakFromMiddleTwoCoordinates[DSetVectorFormFixedTwoCoordinates[1, 1]];
(*Can try more than these three too*)
```

Breaking into 3 and 4 coordinates:

```
SetVectorFormFixedThreeCoordinates[a_,b_,c_]:=
  Module[{theVectors={}},For[i=1,i<=Length[DSetVectorForm],i++,
    If[ DSetVectorForm[[i]][[1]]==a&&DSetVectorForm[[i]][[2]]==b&&DSetVectorForm
         [[i]][[3]]==c,AppendTo[theVectors,DSetVectorForm[[i]]] ];
    ];
   (*for testing
   Print["Size of the list starting (a,b,c) is: ", Length[theVectors]];
   Print["The list is: " ,theVectors];
   *)
   theVectors];
(*now splitting the DSetVectorForm based on the first four coordinates*)
SetVectorFormFixedFourCoordinates[a_,b_,c_,d_]:=
  Module[{theVectors={}},For[i=1,i<=Length[DSetVectorForm],i++,
    If[ DSetVectorForm[[i]][[1]]==a&&DSetVectorForm[[i]][[2]]==b&&DSetVectorForm
         [[i]][[3]]==c&&DSetVectorForm[[i]][[4]]==d,AppendTo[theVectors,
         DSetVectorForm[[i]]] ];
    ];
   (*for testing
   Print["Size of the list starting (a,b,c) is: ", Length[theVectors]];
   Print["The list is: " ,theVectors];
   *)
   theVectors];


(*Now I want to observe the 12342 pattern and 0000'12' pattern*)
For[a=0,a<=4,a++,
 For[b=0,b<=4,b++,
  For[c=0,c<=4,c++,
   (*check the pattern*)
   Print["Here fix first 3 coordinates to be (",a,",",b,",",c,") :"];
   breakdown={};
   For[d=0,d<=4,d++,
     AppendTo[breakdown,Length[SetVectorFormFixedFourCoordinates[a,b,c,d]]];
     ]
    Print["The breakdown from 4th coordinate is: ",Sort[breakdown]];
    ]
  ]
]
```

Algorithms 4: Observe whether the 4 points in D with the same first four coordinates is a plane

```
(*Find all planes in ValueForK*)
valueForKVectorForm=Module[{theVectors={}},For[i=1,i<=Length[valueForK],i++,
    AppendTo[theVectors,TurnToVector3[valueForK[[i]]]]];theVectors];
FindPlanes[theSet_]:=Module[{setOfPlanes={}},For[i=1,i<= Length[theSet],i++,
   For[j=i+1,j<= Length[theSet],j++, (*choose 2 already*)
    For[k=i+1,k<= Length[theSet]&&k!= j,k++,
     For[l=k+1,l<=Length[theSet]&&l!= j,l++,
      If[Mod[theSet[[i]]+theSet[[j]],5]==Mod[theSet[[k]]+theSet[[l]],5],AppendTo[
          setOfPlanes,{theSet[[i]],theSet[[j]],theSet[[k]],theSet[[l]]}]
       ]
      ]
     ]
    ]
   ]; setOfPlanes]
FindPlanesIndex[theSet_]:=Module[{setOfPlanes={}},For[i=1,i<= Length[theSet],i++,
   For[j=i+1,j<= Length[theSet],j++, (*choose 2 already*)
    For[k=i+1,k<= Length[theSet]&&k!= j,k++,
     For[l=k+1,l<=Length[theSet]&&l!= j,l++,
      If[Mod[theSet[[i]]+theSet[[j]],5]==Mod[theSet[[k]]+theSet[[l]],5],AppendTo[
          setOfPlanes,{i,j,k,l}]
       ]
      ]
     ]
    ]
   ]; setOfPlanes]
(*For testing*)
valueForKVectorForm
FindPlanesIndex[valueForKVectorForm]
FindPlanes[valueForKVectorForm]
Length[FindPlanes[valueForKVectorForm]]


(*I want to observe in that 12342 pattern, what is '4' looking like? Plane?*)
For[a=0,a<=4,a++,
 For[b=0,b<=4,b++,
  For[c=0,c<=4,c++,
   (*check the pattern*)
   Print["Here fix first 3 coordinates to be (",a,",",b,",",c,") :"];
   For[d=0,d<=4,d++,
    listToCheck=SetVectorFormFixedFourCoordinates[a,b,c,d];
    If[Length[listToCheck]==4,Print["The fixed 4 coordinates (",a,",",b,",",c
        ,",",d,") gives list of size 4: ", listToCheck];
     (*I want to check if this list of 4 elements is a plane. If so, d=a+(b-a)+(c
         -a), i.e. a+d = b+c, when a,d are opposites from each other (diagonally)
         *)
     Print[Mod[listToCheck[[1]]+listToCheck[[2]]-listToCheck[[3]]-listToCheck
         [[4]],5]=={0,0,0,0,0,0} ||Mod[listToCheck[[1]]+listToCheck[[3]]-
         listToCheck[[2]]-listToCheck[[4]],5]=={0,0,0,0,0,0} ||Mod[listToCheck
         [[1]]+listToCheck[[4]]-listToCheck[[3]]-listToCheck
         [[2]],5]=={0,0,0,0,0,0}];
     ];
    ]
   ]
  ]
 ]
```

Algorithms 5: Investigating the additive structure of K.

The way to investigate is to let Mathematica plot 3-D picture of K. The checking (whether 4 points is a plane) is at the end of the code, where we check if any sum of two points equals the sum of other two.

```
valueForKVectorForm
ListPointPlot3D[valueForKVectorForm, Filling->Bottom]
(*These attempted to find plane equation fails - probably because Linear Solve
     can't be done in mod 5*)
Print["Basis for Nullspace (equation for plane passing origin) is: ", NullSpace[
     valueForKVectorForm]];
Print["Equation for plane not passing origin is: ",LinearSolve[
     valueForKVectorForm,Table[1,Length[valueForKVectorForm]]]];
 valueForK


multiplyInField[a_,b_]:=PolynomialMod[PolynomialMod[a b, x^3+3x+2],5];
(*Try shifting K to see any more visual insight*)
TestXTimesK[a_]:=Module[{},Print[TurnSetToVector3[multiplyInField[a,valueForK]]];
     ListPointPlot3D[TurnSetToVector3[multiplyInField[a,valueForK]],Filling->
     Bottom]]
TestXTimesK[1]
```

Algorithms 6: Miscellaneous functions for K.

These functions are not directly related to the work in this paper, because they do not lead to any significant fruitful insights.

```
(*testing*)
ListExpo = Table[{i,EvaluateInField[x^i]},{i,0,124}]
ListConcerned = {};
For[i=1,i<=Length[ListExpo],i++,
 If[Coefficient[ListExpo[[i]][[2]],x,2] == 0,AppendTo[ListConcerned,ListExpo[[i
     ]]]]
 ]

(*FieldExp[t_]:=TurnToVector3[EvaluateInField[x^t]]*)
FieldExp=Table[TurnToVector3[EvaluateInField[x^t]],{t,1,135}];
(*Find pair of w^a-w^b so that it gives a specific difference. Will output all
    such (a,b)*)
FindAllPairsOfSpecificDifference[theDifference_]:=Module[{allPairs={},i=0,j=0},
  For[i=1,i<= 124,i++
    For[j=1,j<= 124,j++,
     If[Mod[FieldExp[[i]]-FieldExp[[j]],5]==theDifference,AppendTo[allPairs,{i,j,
         FieldExp[[i]],FieldExp[[j]]}]]
     ]
   ]; allPairs]
OneIJPairs=Mod[FindAllPairsOfSpecificDifference[{1,0,0}][[All,{1,2}]],124]
(*Testing*)
FieldExp[[61]]

FlippedValue[t_]:=If[EvenQ[t],124-t,Mod[62-t,124]] (*turns out is the same as
    multiply with 61 then mod 124*)
FlippedPairSet[theSet_]:=Module[{newSet={}},For[i=1,i<=Length[theSet],i++,
    AppendTo[newSet,{FlippedValue[theSet[[i,1]]],FlippedValue[theSet[[i,2]]]}]
   ];newSet]
FlippedPairSet[OneIJPairs]

(*Define a function*)
FindGroupSumDifference3Coor[setOfVectors_]:=Module[{},
  differenceSumFromVector[a_,b_,c_]:=Module[{listInVar={}},ConvertTo3Var[A_]:=a^A
      [[1]] b^A[[2]] c^A[[3]] ;For[i=1,i<= Length[setOfVectors],i++,AppendTo[
      listInVar,ConvertTo3Var[setOfVectors[[i]]]]];Total[listInVar]];
  Clear[a,b,c];partOne=differenceSumFromVector[a,b,c];
  partTwo=differenceSumFromVector[a^-1,b^-1,c^-1];
  timeStart=AbsoluteTime[];
  answer =PolynomialMod[Expand[a^5 b^5 c^5differenceSumFromVector[a,b,c]*
      differenceSumFromVector[a^-1,b^-1,c^-1]],{a^5-1,b^5-1,c^5-1}];
  timeUsed=AbsoluteTime[]-timeStart;
  Print["Time Used : ", timeUsed];
  answer]

(*Testing*)
FindGroupSumDifference3Coor[valueForKVectorForm]
 FromGroupSumTo3DVectors[a_,b_,c_,q_,theSum_]:=Module[{theVectors={}},
  For[i=0,i<q,i++,
   For[j=0,j<q,j++,
    For[k=0,k<q,k++,
     If[i==0&&j==0&&k==0,
      AppendTo[theVectors,{i,j,k,Coefficient[Coefficient[Coefficient[theSum,a,0],
          b,0],c,0]}],
      If[Coefficient[theSum,a^i b^j c^k]!= 0,
       AppendTo[theVectors,{i,j,k,Coefficient[theSum,a^i b^j c^k]}]]
      ]
     ]
    ]
   ];
  theVectors]

FromGroupSumTo3DVectors[a,b,c,5,FindGroupSumDifference3Coor[valueForKVectorForm]]
CoefficientList[FindGroupSumDifference3Coor[valueForKVectorForm],{a,b,c}]
 valueForKVectorForm[[12]]={0,1,0}; (*some checking*)
valueForKVectorForm
CoefficientList[FindGroupSumDifference3Coor[valueForKVectorForm],{a,b,c
    }][[1,2,1]]
Map3DCoefficientListToPoints[coefList_ ,q_]:=Module[{thePoints={},i=0,j=0,k=0},
  For[i=0,i<q,i++,
   For[j=0,j<q,j++,
    For[k=0,k<q,k++,
     If[coefList[[i+1,j+1,k+1]]!=0,
      AppendTo[thePoints,{i,j,k,coefList[[i+1,j+1,k+1]]}]
     ]       ]      ]     ];
  thePoints]
```

```
(*This will plot the picture of K-K in 3-D. Specifically, it sees how many times
      each element t appears in K-K *)
dataToPlot=Map3DCoefficientListToPoints[CoefficientList[
      FindGroupSumDifference3Coor[valueForKVectorForm],{a,b,c}],5]
ListPointPlot3D[List/@dataToPlot[[All,{1,2,3}]],PlotStyle->({PointSize[Large],
      Blend[{{4,Darker[Green]},{6,Yellow},{8,Red}},#1]}&/@Flatten[dataToPlot[[All
      ,{4}]]])]

(*Do more random change to K visually for more insight*)
 valueForK
FlipElementsInSet[SetOfElements_]:=Module[{flippedSet={}},For[i=1,i<=Length[
      SetOfElements],i++,
    AppendTo[flippedSet,EvaluateInField[(SetOfElements[[i]])^61]]
    ];flippedSet]
FlipElementsInSet[valueForK]
ListPointPlot3D[TurnSetToVector3[FlipElementsInSet[valueForK]],Filling->Bottom,
      PlotStyle->PointSize[Large]]
```

Algorithms 7: Finding all PDSs in $\mathbb{Z}_4^3$
This is a Java code.

```
import java.util.*;
public class Main
{
    public static void check4x4x4()
    {
        long startTime = System.nanoTime();
        long stepCount = 0;

        ArrayList<String> PDS1Subgroups = new ArrayList<String>();
        ArrayList<String> PDS2DS = new ArrayList<String>();
        ArrayList<String> PDS3Others = new ArrayList<String>();

        int[][][] D = new int[4][4][4]; //this will be determined 0,1 later. 1
             means it is in D, 0 means it is not in D
        //go through loops by hand. See which depends on which
        for (int a100=0; a100<=1; a100++)
        {
            D[1][0][0] = a100;
            D[3][0][0] = a100;
            for (int a010=0; a010<=1; a010++)
            {
                D[0][1][0] = a010;
                D[0][3][0] = a010;
                for (int a110=0; a110<=1; a110++)
                {
                    D[1][1][0] = a110;
                    D[3][3][0] = a110;
                    for (int a210=0; a210<=1; a210++)
                    {
                        D[2][1][0] = a210;
                        D[2][3][0] = a210;
                        for (int a310=0; a310<=1; a310++)
                        {
                            D[3][1][0] = a310;
                            D[1][3][0] = a310;
                            for (int a120=0; a120<=1; a120++)
                            {
            D[1][2][0] = a120;
            D[3][2][0] = a120;
            //System.out.println(Arrays.deepToString(D)+"\n"); //for checking
            //do same thing for layer z=2
            for (int a102=0; a102<=1; a102++)
            {
                D[1][0][2] = a102;
                D[3][0][2] = a102;
                for (int a012=0; a012<=1; a012++)
                {
                    D[0][1][2] = a012;
                    D[0][3][2] = a012;
                    for (int a112=0; a112<=1; a112++)
                    {
                        D[1][1][2] = a112;
                        D[3][3][2] = a112;
                        for (int a212=0; a212<=1; a212++)
                        {
                            D[2][1][2] = a212;
                            D[2][3][2] = a212;
                            for (int a312=0; a312<=1; a312++)
                            {
                                D[3][1][2] = a312;
                                D[1][3][2] = a312;
                                for (int a122=0; a122<=1; a122++)
                                {
                D[1][2][2] = a122;
                D[3][2][2] = a122;
                /* //for checking
                stepCount++;
                if (stepCount%10000 == 0)
                {
                System.out.println("************************ Steps: " + stepCount
                     + "********************\n");
                System.out.println(Arrays.deepToString(D)+"\n"); //this takes
                     long to print though
                }*/
                //setting up D[a][b][1] and D[a][b][3] is more systematic: choose
                      D[a][b][1] then we will know D[3a][3b][1]
                for (int b00=0; b00<=1; b00++)
                {
                    D[0][0][1] = b00;
            for (int b01=0; b01<=1; b01++)
            {
                D[0][1][1] = b01;
                for (int b02=0; b02<=1; b02++)
                {
                    D[0][2][1] = b02;
                    for (int b03=0; b03<=1; b03++)
                    {
                        D[0][3][1] = b03;
                        for (int b10=0; b10<=1; b10++)
                        {
                            D[1][0][1] = b10;
                            for (int b11=0; b11<=1; b11++)
                            {
                                D[1][1][1] = b11;
                                for (int b12=0; b12<=1; b12++)
                                {
D[1][2][1] = b12;
    for (int b13=0; b13<=1; b13++)
    {
        D[1][3][1] = b13;
        for (int b20=0; b20<=1; b20++)
        {
            D[2][0][1] = b20;
            for (int b21=0; b21<=1; b21++)
            {
                D[2][1][1] = b21;
                for (int b22=0; b22<=1; b22++)
                {
                    D[2][2][1] = b22;
                    for (int b23=0; b23<=1; b23++)
                    {
                        D[2][3][1] = b23;
                        for (int b30=0; b30<=1; b30++)
                        {
                            D[3][0][1] = b30;
                            for (int b31=0; b31<=1; b31++)
                            {
                                D[3][1][1] = b31;
                                for (int b32=0; b32<=1; b32++)
                                {
                                    D[3][2][1] = b32;
                                    for (int b33=0; b33<=1; b33++)
                                    {
                    D[3][3][1] = b33;
                    //setting D[a][b][3]
                    for (int a=0; a<4; a++)
                    {
                        for (int b=0; b<4; b++)
                        {
                            D[a][b][3] = D[(3*a)%4][(3*b)%4][1];
                        }
                    }
                    //I can assume something upto automorphism z -> 3z
                    if (D[0][1][1]+D[1][0][1]+D[1][1][1]+D[1][2][1]+D
                         [1][3][1]+D[2][1][1] < D[0][1][3]+D[1][0][3]+D
                         [1][1][3]+D[1][2][3]+D[1][3][3]+D[2][1][3])
                    {
                        continue; //don't need to do any element of order
                              2 stuff. Skip to next set for elements of
                             order 4
                    }
                    //start dealing with elements of order 2. Will do
                         automorphism so that (2,0,0) appears first , then
                         (0,2,0) , then (0,0,2) all everything else except
                         (2,2,0)
                    for (int a200=0; a200<=1; a200++)
                    {
                        D[2][0][0] = a200;
                        for (int a020=0; a020<=a200; a020++) //<=a200 is
                             from automorphism conditions
                        {
                            D[0][2][0] = a020;
                            for (int a220=0; a220<=a020; a220++) //a020
                                 is second highest
                            {
                                D[2][2][0] = a220;
                                for (int a002=0; a002<=a020; a002++) //
                                     a020 is second highest
                                {
                                    D[0][0][2] = a002;
                                    for (int a202=0; a202<=a002; a202++)
                                         //the rest are no more than a002
```

```
                                   {
                                       D[2][0][2] = a202;
                                       for (int a022=0; a022<=a002; a022
                                            ++)
                                       {
                                           D[0][2][2] = a022;
                                           for (int a222=0; a222<=a002;
                                                a222++)
                                           {
            D[2][2][2] = a222;
            //check that there aren't too many points - no more than half
            int sizeD = 0;
            for (int a=0; a<4; a++)
            {
                for (int b=0; b<4; b++)
                {
                    for (int c=0; c<4; c++)
                    {
                        if (D[a][b][c] > 0)
                        {
                            sizeD++;
                        }
                    }
                }
            }
            if (sizeD >= 32) //if |D| is >= 32, complement has <=32
                 points, taking identity out would have <=31 points, a
                 smaller one than D
            {
                continue;
            }
            stepCount++;
            if (stepCount%100000000 == 0)
            {
                System.out.println("*********************** Steps: " +
                    stepCount + "********************\n");
                //System.out.println(Arrays.deepToString(D)+"\n"); //this
                     takes long to print though
            }

            int checkResult = checkPDS(D,sizeD);
            if ( checkResult > 0 ) //for checking
            {
                System.out.println(checkResult + " " + sizeD + " PDS
                    Found: " + Arrays.deepToString(D)+"\n");
            }

            if ( checkResult == 1 ) //found PDS
            {
                PDS1Subgroups.add(Arrays.deepToString(D));
            }
            else if ( checkResult == 2)
            {
                PDS2DS.add(Arrays.deepToString(D));
            }
            else if ( checkResult == 3)
            {
                PDS3Others.add(Arrays.deepToString(D));
            }
                                           }
                                       }
                                   }
              ....       } .... [many '}' here]
}
long estimatedTime = System.nanoTime() - startTime;
System.out.println("All execution of this method takes: " + estimatedTime
    /1000000000.000 + " seconds.");
System.out.println("The number of sets checked whether it is PDS is: " +
    stepCount);
System.out.println("The total number of PDS found is: " + (PDS1Subgroups.
    size()+PDS2DS.size()+PDS3Others.size()));
System.out.println("All PDS are broken down into difference categories:
    ");
System.out.println(PDS1Subgroups.size() + " PDS are found as type 1 - as
    subgroups");
System.out.println(PDS2DS.size() + " PDS are found as type 2 - as
    (64,28,12) Difference Set");
System.out.println(PDS3Others.size() + " PDS are found as type 3 - as
    others. They are:");
for (int i=0; i<PDS3Others.size(); i++)
{
    System.out.println(PDS3Others.get(i));
}
```

```
}
/**
 * given a set (as 0,1 matrix) in Z_4^3, determine if it is PDS. It will also
       return true if character sum has 1 value
 * (That will be DS actually, but that's interesting too, though we won't be
      getting all possible DS in this program search)
 * return 0 if it gives more than 3 character sums
 * return 1 if D is a subgroup (with identity out)
 * return 2 if D is a (64,28,12) hadamard difference set
 * return 3 otherwise (this is what is interesting)
 */
public static int checkPDS(int[][][] set, int sizeD)
{
    int[] listOfSums = {Integer.MIN_VALUE,Integer.MIN_VALUE}; //sum not
         recorded yet. These are impossible values
    //go through all characters. Sending (1,0,0),(0,1,0),(0,0,1) to  (sqrt-1)
         ^i, ^j, ^k, respectively
    for (int i=0; i<4; i++)
    {
        for (int j=0; j<4; j++)
        {
            for (int k=0; k<4; k++)
            {
                if (!(i == 0 && j == 0 && k == 0) ) //only do things when chi
                     is not principle
                {
                    int chaSum = 0; //value of chi(D)
                    for (int a=0; a<4; a++)
                    {
                        for (int b=0; b<4; b++)
                        {
                            for (int c=0; c<4; c++)
                            {
                                //look at (a,b,c) in Z_4^3
                                if (set[a][b][c] > 0) //(a,b,c) is in the set
                                {
                                    int expoOfI = (a*i+b*j+c*k)%4; //chi(a,b,
                                         c) = (sqrt-1)^(ai+bj+ck)
                                    if (expoOfI == 0)
                                    {
                                        chaSum = chaSum + 1;
                                    }
                                    else if (expoOfI == 2)
                                    {
                                        chaSum = chaSum - 1;
                                    }
                                }
                                //then we get what chi(a,b,c) is with this
                                     character now.
                            }
                        }
                    }
                    //here we get what chi(D) is (over all elements in D)
                    if (listOfSums[0] == Integer.MIN_VALUE) //no sum has been
                         recorded
                    {
                        listOfSums[0] = chaSum; //record this sum. Done for
                             this cha
                    }
                    else if (listOfSums[0] != chaSum) //listOfSums[0] is
                         recorded. If it is same as chaSum, do nothing. But
                         different means we have to record it
                    {
                        if (listOfSums[1] == Integer.MIN_VALUE) //no second
                             sum has been recorded
                        {
                            listOfSums[1] = chaSum; //record this sum. Done
                                 for this cha
                        }
                        else if (listOfSums[1] != chaSum) //listOfSums[1] is
                             recorded. If it is same as chaSum, do nothing.
                             But different means we got the thrid sum
                        {
                            return 0;
                        }
                    }
                }
            }
        }
    }
    //now it is not case 0. Have to think whether to return 1,2, or 3
    if (listOfSums[0] > listOfSums[1])
    {
            //sort the sums into increasing order
            int tmp = listOfSums[0];
            listOfSums[0] = listOfSums[1];
            listOfSums[1] = tmp;
        }
        //return 1 iff cha(D) = -1 or size of D
        if (listOfSums[0] == -1 && listOfSums[1] == sizeD)
        {
            return 1;
        }
        else if (listOfSums[0] == -4 && listOfSums[1] == 4 && sizeD == 28)
        {
            return 2;
        }
        else if (listOfSums[0] == -5 && listOfSums[1] == 3 && sizeD == 27)
           //(64,28,12) DS tossing the indentity out
        {
            return 2;
        }
        else
        {
            return 3;
        }
    }
}
```

Algorithms 8: Finding all PDSs in $\mathbb{Z}_5^3$

This is a Java code. We use Theorem 46: to find PDS is exactly the same as solving the matrix equation. Note that matrix A is the same as matrix B in Theorem 46 in Section 3. A can be computed separately (in actuality I used Mathematica to compute A and copy the array form into the Java code).

```java
import java.util.*;
/**
 *
 * @author Tao
 * @version (a version number or a date)
 */
public class Main
{
    public static final int[][] A = [Please see matrix A above];
    /**
     * The idea is to generate all possible x, which is a +-1 vector of length
         31, and check if the
     * product A.x has each row = +-5 or not.
     */
    public static void check5x5x5()
    {
        long startTime = System.nanoTime();
        long stepCount = 0;
        ArrayList<String> listOfX = new ArrayList<String>();

        int[] x = new int[31]; //will be determined +1, -1 later
        //go through loops by hand. See which depends on which
        for (int x6=0; x6<=1; x6++)
        {
            x[6] = 1 - 2*x6;
            for (int x7=0; x7<=1; x7++)
            {
                x[7] = 1 - 2*x7;
                for (int x8=0; x8<=1; x8++)
                {
                    x[8] = 1 - 2*x8;
                    for (int x9=0; x9<=1; x9++)
                    {
                        x[9] = 1 - 2*x9;
                        for (int x10=0; x10<=1; x10++)
                        {
                            x[10] = 1 - 2*x10;
                            for (int x11=0; x11<=1; x11++)
                            {
        x[11] = 1 - 2*x11;
        for (int x12=0; x12<=1; x12++)
        {
            x[12] = 1 - 2*x12;
            for (int x13=0; x13<=1; x13++)
            {
                x[13] = 1 - 2*x13;
                for (int x14=0; x14<=1; x14++)
                {
                    x[14] = 1 - 2*x14;
                    for (int x15=0; x15<=1; x15++)
                    {
                        x[15] = 1 - 2*x15;
                        for (int x16=0; x16<=1; x16++)
                        {
                            x[16] = 1 - 2*x16;
                            for (int x17=0; x17<=1; x17++)
                            {
        x[17] = 1 - 2*x17;
        for (int x18=0; x18<=1; x18++)
        {
            x[18] = 1 - 2*x18;
            for (int x19=0; x19<=1; x19++)
            {
                x[19] = 1 - 2*x19;
                for (int x20=0; x20<=1; x20++)
                {
                    x[20] = 1 - 2*x20;
                    for (int x21=0; x21<=1; x21++)
                    {
                        x[21] = 1 - 2*x21;
                        for (int x22=0; x22<=1; x22++)
                        {
                            x[22] = 1 - 2*x22;
                            for (int x23=0; x23<=1; x23++)
```

```
                      {
                          x[23] = 1 - 2*x23;
                          for (int x24=0; x24<=1; x24++)
                          {
x[24] = 1 - 2*x24;
  for (int x25=0; x25<=1; x25++)
  {
      x[25] = 1 - 2*x25;
      for (int x26=0; x26<=1; x26++)
      {
          x[26] = 1 - 2*x26;
          for (int x27=0; x27<=1; x27++)
          {
              x[27] = 1 - 2*x27;
              for (int x28=0; x28<=1; x28++)
              {
                  x[28] = 1 - 2*x28;
                  for (int x29=0; x29<=1; x29++)
                  {
                      x[29] = 1 - 2*x29;
                      for (int x30=0; x30<=1; x30++)
                      {
      x[30] = 1 - 2*x30;
      int sumLast25 = x6+x7+x8+x9+x10+x11+x12+x13+x14+x15+x16+x17+x18+x19+
          x20+x21+x22+x23+x24+x25+x26+x27+x28+x29+x30;
      if (sumLast25 != 10 && sumLast25 != 15) //this is the
          00000011111...111 row condition
      {
          continue;
      }
      /*//for checking
      stepCount++;
      if (stepCount%10000 == 0)
      {
      System.out.println("*********************** Steps: " + stepCount +
          "********************\n");
      }*/
      for (int x0=0; x0<=1; x0++)
      {
          x[0] = 1 - 2*x0;
          for (int x1=0; x1<=1; x1++)
          {
              x[1] = 1 - 2*x1;
              for (int x2=0; x2<=1; x2++)
              {
                  x[2] = 1 - 2*x2;
                  for (int x3=0; x3<=1; x3++)
                  {
                      x[3] = 1 - 2*x3;
                      for (int x4=0; x4<=1; x4++)
                      {
                          x[4] = 1 - 2*x4;
                          for (int x5=0; x5<=1; x5++)
                          {
      x[5] = 1 - 2*x5;
      //wanna delete the compliment?
      stepCount++;
      if (stepCount%10000000 == 0)
      {
          System.out.println("*********************** Steps: " + stepCount
              + "********************\n");
          //System.out.println(Arrays.deepToString(D)+"\n"); //this takes
              long to print though
      }
      //start checking if x is valid
      boolean checkResult = true;
      for (int index=0; index < A.length; index++)
      {
          int sum=0;
          for (int j=0; j<A[0].length; j++)
          {
              sum = sum + A[index][j]*x[j];
          }
          //sum is entries at row index now. Check if it's +-5 or not
          if (sum != 5 && sum != -5)
          {
              checkResult = false;
              break;
          }
      }
      if ( checkResult )
      {
          //System.out.println("PDS Found: " + Arrays.toString(x)+"\n");
```

```
                /*
                int sumOfX = sumArray(x);
                if (sumOfX != 4 && sumOfX != -4)
                {
                System.out.println("Interesting Sum: " + sumOfX + "\n");
                }*/
                listOfX.add(Arrays.toString(x));
            }
                                }
                            }
                        }
                    }
                }
            }
                                                            }
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
        long estimatedTime = System.nanoTime() - startTime;
        System.out.println("All execution of this method takes: " + estimatedTime
            /1000000000.000 + " seconds.");
        System.out.println("The number of sets checked whether it is PDS is: " +
            stepCount);
        System.out.println("The total number of PDS found is: " + listOfX.size())
            ;
        for (int i=0; i<10; i++)
        {
            //System.out.println("x" + (i+1) + "=" + listOfX.get(i) + ";");
        }
        //if want to print for some sample of PDSs
        int j=0;
        for (int i=0; i<listOfX.size(); i=i+j)
        {
            System.out.println("x[" + (i+1) + "]=" + listOfX.get(i) + ";");
            j++;
        }
    }
    /**
     * Get the sum
     */
    public static int sumArray(int[] array)
    {
        int sum = 0;
        for (int i=0; i<array.length; i++)
        {
            sum = sum + array[i];
        }
        return sum;
    }
}
```

The result of the run (not including the printing of sample PDSs at the end):

```
All execution of this method takes: 41.405464682 seconds.
The number of sets checked whether it is PDS is: 418401280
The total number of PDS found is: 94800
```

Algorithms 9: Finding the size of the stabilizer of each PDS in $\mathbb{Z}_5^3$

```
ClearAll["Global`*"];
NotebookAutoSave->True (*Does not work...*)
ListPoints=Union[Table[{1,Mod[b,5],Floor[b/5]},{b,0,24}],Table[{0,1,c},{c
     ,0,4}],{{0,0,1}}]
ListPlanes=ListPoints
 (*Field stuff*)
EvaluateInField[y_]:=PolynomialMod[PolynomialMod[y,x^3+3x+2],5];
Tra[y_]:=PolynomialMod[PolynomialMod[y+y^5+y^(25),x^3+3x+2],5];
TurnToVector3[A_]:={Coefficient[A,x,0],Coefficient[A,x,1],Coefficient[A,x,2]};
ListExpo= Table[{i,TurnToVector3[EvaluateInField[x^i]]},{i,0,123,2}];
ListSquares = Table[TurnToVector3[EvaluateInField[x^i]],{i,0,123,2}]
Length[ListSquares];


listAllExpo=Table[TurnToVector3[EvaluateInField[x^i]],{i,0,123}]
discreteLog[point_]:=Module[{answer=-1},For[i=1,i<=Length[listAllExpo],i++,If[
     listAllExpo[[i]]==point,answer=i-1;Break;]];answer];
discreteLogSet[D_]:=Module[{setLog={},i},For[i=1,i<=Length[D],i++,
    AppendTo[setLog,discreteLog[D[[i]]]];];
   setLog];
discreteLog[{0,2,0}];
discreteLogSet[listAllExpo]
(*This function get vector +-1 and convert into set D*)
convertToD[x_]:=Module[{D={}},
   For[index=1,index<=Length[x],index++,
    If[x[[index]]==1,AppendTo[D,ListPoints[[index]]];AppendTo[D,Mod[4*ListPoints
         [[index]],5]];,If[x[[index]]==-1,AppendTo[D,Mod[2*ListPoints[[index
         ]],5]];AppendTo[D,Mod[3*ListPoints[[index]],5]];]];
    ]; D];


 (*Finding automorphism group size of any set*)
(*Generate all autoporphism*)
findSizeOfAutoGroup[D_]:=Block[{},
   Print["Finding the size of autoporphism group"];
   timeStart=AbsoluteTime[];
   OneRow=Tuples[{0,1,2,3,4},3];
   count=0;
   For[i=1,i<=Length[OneRow],i++,
     For[j=1,j<=Length[OneRow],j++,
      For[k=1,k<=Length[OneRow],k++,
       (*Check if matrix has Det0*)
       MatrixAuto={OneRow[[i]],OneRow[[j]],OneRow[[k]]};
       If[Divisible[Det[MatrixAuto],5],Continue[]];
       YesIsAutomorphism=True;
       For[index=1,index<=Length[D],index++,
        If[MemberQ[D,Mod[MatrixAuto.D[[index]],5]]!=True,YesIsAutomorphism=False;
            Break];
        ];
       If[YesIsAutomorphism==True,count++;(*Print[MatrixForm[MatrixAuto]];*)];
       ]
      ]
     ]
    Print["The set tested is", D];
   Print["Size of automorphism group is: ",count];
   distinctForm=124*120*100/count;
   Print["Number of equivalent sets is: ",distinctForm];
   timeUsed=AbsoluteTime[] - timeStart;
   Print["Time used in this process is ",timeUsed];
   ];


(*Example of what PDS can be in the vector X_D form. This can be obtained from
    algorithms that find all PDS*)
 x0
     ={-1,1,-1,-1,1,1,-1,1,1,1,1,1,-1,-1,1,-1,1,-1,-1,-1,-1,-1,-1,-1,1,-1,-1,1,-1,1,-1};

x1={1,1,1,-1,1,-1,1,1,1,1,1,1,1,1,-1,-1,1,1,-1,-1,-1,-1,1,1,-1,-1,-1,1,-1,1,1};
x2={-1,1,1,-1,-1,1,1,1,1,1,1,1,1,1,-1,-1,1,1,-1,-1,-1,-1,1,-1,1,-1,1,-1,-1,1,1};
x3={1,-1,1,1,1,-1,1,1,1,1,1,1,1,1,-1,-1,1,1,-1,-1,-1,-1,1,-1,-1,1,1,-1,1,-1,1};
x4={-1,1,-1,1,-1,1,1,1,1,1,1,1,1,1,-1,-1,1,1,-1,-1,-1,-1,1,-1,-1,1,1,-1,1,-1,1};
x1001
    ={1,-1,-1,-1,1,1,1,1,1,1,1,1,-1,-1,1,-1,1,-1,1,1,-1,1,1,1,-1,-1,-1,1,-1,1,-1};

x7001
    ={-1,1,-1,-1,1,-1,1,1,1,-1,1,1,-1,-1,1,1,-1,1,-1,-1,-1,-1,1,1,-1,1,1,1,-1,1,1};

x10001
    ={-1,1,-1,1,1,1,1,1,1,-1,-1,1,-1,1,-1,-1,1,1,-1,1,1,-1,1,1,1,1,1,-1,1,-1,-1};
```

```
x11001
    ={-1,-1,1,-1,-1,1,1,1,1,-1,-1,-1,1,-1,-1,1,-1,-1,-1,-1,-1,1,-1,-1,1,-1,1,1,-1,1,-1};

x12001
    ={1,-1,1,1,1,1,1,1,-1,1,1,1,1,1,-1,1,1,1,-1,-1,-1,1,-1,1,1,-1,1,-1,-1,-1,1};
x13001
    ={-1,-1,-1,1,-1,-1,1,1,-1,1,1,1,-1,1,-1,-1,1,-1,1,-1,-1,-1,1,1,1,1,1,1,1,-1,-1};

x14001
    ={-1,1,-1,1,-1,1,1,1,-1,1,1,-1,1,-1,1,1,-1,1,1,-1,1,1,1,1,1,-1,-1,1,-1,-1,-1};

x15001
    ={-1,-1,-1,-1,1,1,1,1,-1,1,1,-1,-1,-1,1,-1,-1,-1,1,1,-1,1,1,-1,-1,-1,-1,-1,-1,1,-1};

x16001
    ={1,-1,1,-1,1,1,1,1,-1,1,-1,1,-1,1,1,-1,1,1,-1,1,1,-1,1,-1,-1,-1,1,1,1,-1,1};
x30001
    ={-1,1,-1,-1,-1,-1,1,-1,1,-1,1,1,1,1,1,-1,1,-1,-1,1,-1,1,-1,1,-1,-1,1,1,-1,1,1};

x17001
    ={1,-1,1,1,-1,-1,1,1,-1,1,-1,-1,1,-1,1,1,-1,1,-1,1,-1,-1,-1,-1,-1,-1,1,-1,-1,-1,1};

x43001
    ={1,-1,1,1,-1,1,1,-1,-1,-1,1,-1,1,1,-1,-1,-1,1,1,1,-1,-1,-1,-1,-1,-1,-1,1,1,-1,1};

x70001
    ={-1,1,-1,-1,1,1,-1,1,-1,-1,-1,-1,1,-1,1,-1,1,-1,1,1,1,1,-1,-1,1,-1,-1,-1,-1,1,-1};

x77001
    ={1,-1,1,-1,-1,-1,-1,-1,1,-1,1,1,1,1,1,-1,-1,-1,1,1,1,1,1,1,1,-1,1,-1,-1,1,-1};

x94001
    ={1,1,1,-1,-1,1,-1,-1,-1,-1,-1,-1,1,-1,1,1,1,-1,-1,-1,1,-1,1,1,-1,-1,1,-1,1,1,-1};



(*Example of how to find stabilizer size*)
findSizeOfAutoGroup[convertToD[x1001]];
findSizeOfAutoGroup[convertToD[x10001]];
findSizeOfAutoGroup[convertToD[x94001]];
findSizeOfAutoGroup[convertToD[x1]];
findSizeOfAutoGroup[convertToD[x0]];


(*The rest below are optional - added for more visualization*)
 (*Optionally, we may want to test these x vectors*)
Clear[a,b,c];
convertTo3Var[a_,b_,c_,point_]:=a^point[[1]] b^point[[2]] c^point[[3]];
convertTo3VarOnSet[a_,b_,c_,D_]:=Module[{convertedToVarD={}},For[i=1,i<=Length[D
     ],i++,AppendTo[convertedToVarD,convertTo3Var[a,b,c,D[[i]]]]];
   convertedToVarD];
convertTo3VarOnSet[a,b,c,{{1,2,3},{2,3,4},{0,0,1}}];
differenceSum[a_,b_,c_,D_]:=Total[convertTo3VarOnSet[a,b,c,D]];
findDifferenceSumFromVector[x_]:=Module[{},
  partOne=differenceSum[a,b,c,convertToD[x]];
  partTwo=differenceSum[a^-1,b^-1,c^-1,convertToD[x]];
  product=Expand[a^5 b^5 c^5*partOne*partTwo];
  PolynomialMod[product,{a^5-1,b^5-1,c^5-1}]]
(*Also, optionally we can look at its additive structure*)
 lookAdditive[xi_]:=Module[{},Print[convertToD[x0]];ListPointPlot3D[convertToD[x0
      ]]]
lookAdditive[x[1]]


(*and also, optionally we can look at its multiplicative structure*)
 lookMultiplicative[xi_]:=Module[{setLog,number,i},setLog=Sort[discreteLogSet[
      convertToD[xi]]];Print[setLog];(*Print[NumberLinePlot[setLog]];*)
  Array[set,4];
  set[1]={};set[2]={};set[3]={};set[4]={};
  For[i=1,i<=Length[setLog],i++,
   number=setLog[[i]];
   setIndex=Quotient[number,31];
   number=Mod[number,31];
   AppendTo[set[setIndex+1],number];
   ];
  NumberLinePlot[{set[1],set[2],set[3],set[4]}]
  ]
lookMultiplicative[x13501]
```