Set's Maximal Empty Board

Henri Abensour

March 2024

1 Introduction

Games, specifically card and board games, were a central part of my upbringing. For a period of a few years at least, it would have meant something especially disastrous had occurred if I wasn't greeted with a game against my father upon my return from school.
The games varied quite widely: we would often enter a phase of obsession, repeatedly playing one particularly addictive one.
Overall, however, the gamut included simple ones such as Connect Four, intricate ones such as Chinese checkers, and even slightly more physical ones such as the tabletop game Carrom (which I highly recommend).
I was never as fond of the traditional board games that may come to mind. Chess was the bane of my existence, despite my father and brother's prowess.
Despite my father's visible preference for, or certainly leaning towards, those historic games, he would never shy from experimenting with a new one that intrigued him. I can unquestionably attribute our vast collection of dusty board games to him.

My allure for games — including video games, which objectively consumed a major portion of my adolescence — has largely faded.
I haven't touched any in years though the nostalgia never fails to allure.
As a result, when the opportunity arose to introduce friends to a game from my youth, I seized it, only for it to spark some vivid memories.

The game may be familiar to you: *Set*.
It became fairly popular after winning a number of awards, including American Mensa's Mensa Select award in 1991. The purpose of this isn't to laud the game, but to introduce the motivation my father and I may have felt behind what follows.

A brief outline of the rules and the game-play taken from Wikipedia is as follows:
"In the game, certain combinations of three cards are said to make up a "set". For each one of the four categories of features — color, number, shape, and shading — the three cards must display that feature as either a) all the same, or b) all different. Put another way: For each feature the three cards must avoid having two cards showing one version of the feature and the remaining

card showing a different version.
For example, 3 solid red diamonds, 2 solid green squiggles, and 1 solid purple oval form a set, because the shadings of the three cards are all the same, while the numbers, the colors, and the shapes among the three cards are all different. For any set, the number of features that are constant (the same on all three cards) and the number of features that differ (different on all three cards) may break down as: all 4 features differing; or 1 feature being constant and 3 differing; or 2 constant and 2 differing; or 3 constant and 1 differing. (All 4 features being constant would imply that the three cards in the set are identical, which is impossible since no cards in the Set deck are identical.)"

Set regularly featured in our rotation of games as I was growing up.
I suspect we all enjoyed it as it was convenient, fairly short, easy to setup, and even portable.
At different times, a different member of the family would dominate. My brother and I were both naturally gifted set-identifiers.
When I must have been around eight years old, my father and I were constantly playing after school.
We would repeatedly encounter situations in games in which no one could identify a set, and as a result we would need to add three cards per the rules.
Occasionally, this would continue until we reached a total of 21 cards on the board. We therefore wondered how many possible cards could be placed without a set being produced.

Based on our experience, it appeared that that number had to be around 18. It was tremendously rare that we would need 7 rows of 3 cards to produce a single set.
One day, we therefore attempted to tackle the problem. We laid out as many cards as possible whilst ensuring no three cards formed a set.
If I recall correctly, we managed to reach 17 cards but abandoned the project very quickly.
As we were clearing the table of cards, I remember my father wondering how one could mathematically determine that maximum number of cards without a set. I too was puzzled by the problem, and sought a more general solution than testing different combinations.

And here we are now.

2 Definitions

The problem at hand is to determine the maximum number of cards such that no triplet of three cards forms a set.
To distinguish the mathematical term "set" from the nomenclature of the game, I will hereafter refer to a set in the game by the unoriginal and somewhat cumbersome name of "game-set", shortened to "gset".
I will — or I will certainly aim to — hereafter capitalise and italicise the word *Set* when referring to the game.

Definition 1 (Game-set (gset))**.** A game-set, shortened to gset, is a hand of cards that all together form a "set" in *Set*.

Throughout this work, we'll need to refer to different numbers that uniquely characterise the complexity of the version of *Set* we are playing.

Let n be the total number of cards, s the number of cards per game-set, f the number of features, all in $\mathbb{N}$.
s is therefore also the number of options within each feature (for instance, red, green, and blue for colour).
For instance, in *Set*, there are 3 cards per game-set, so $s = 3$, and 4 features (colour, number, shape, and shading), so $f = 4$.
It immediately follows that $n = s^f$.
One could imagine a game with different numbers of cards in each feature, that is, instead of blue, red, and green for the colour feature, one could add black, while not adding any shapes or shades.
Then $n = \prod_{i=1}^{f} s_i$ where s_i is the number of options in each feature ($s_i = 3, \forall i$ in the original game).
We will only briefly cover that case at the end, and until then, unless otherwise specified, we'll be tackling the case $s_i = s, \forall i$.

Different number of cards per set possible

Definition 2 (*Set* Version)**.** The *Set* Version, also known as the version or variation, is the equivalent game of *Set* with s cards per gset (not necessarily 3) and f features (not necessarily 4). Then the version under consideration can be denoted the s, f-version and one can refer to the s and f of that version.

The original *Set* is therefore the $3, 4$-version and we can immediately establish the following defining property of a gset.

Property 1. A gset in the s, f-version is a group of s cards which share the property that t features differ among the cards ($t \in \{1, \dots, f\}$).
It follows that $l = f - t \in \{0, \dots, f-1\}$ features are therefore identical amongst them.

Evidently, if $t = 0, l = f$, all s cards in the set would be identical, violating the uniqueness of the cards.

2.1 Geometric representation

The most useful and most immediate representation of all the cards in the game is an f-dimensional cubic hypergrid with s grid points per direction. Each grid point, located by its f-dimensional vector with coordinates in $\{0, \dots, s-1\}$, represents a single card (as cards are unique in the game).

Definition 3 (s, f-grid)**.** The s, f-grid is the space $\{0, 1, \dots, s-1\}^f$.

A card in the original version of *Set* would be of the form (a, b, c, d) (vectors will be represented horizontally without tedious transpose signs), in which the first component could be colour, the second number, the third shape, and the

last shading, and $a, b, c, d \in \{0, 1, 2\}$.
Formally, this grid is a rank f tensor with integer coefficients.
This yields the following useful definitions.

Definition 4 (Card)**.** A card in the s, f-version is a vector of dimensionality f with integer coordinates in $\{0, 1, \ldots, s-1\}$, that is a card $\mathbf{a}$ is in $\{0, 1, \ldots, s-1\}^f$, so in the s, f-grid.

Definition 5 (Board)**.** A board of cards is the space $\{0, 1, \ldots, s-1\}^f$ in which points associated with cards on the board are marked by a 1 and those associated with cards that are not on the board are marked by a 0.

A board of cards is therefore a grid of zeros and ones.
The deck of all the cards is simply the s, f-grid. In the vector representation, a gset can be denoted by $g = \{\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(s)}\}$.

The visual tool of the grid shines in particular when studying the original game, or any version with fixed s, as a gset is simply a line — horizontal, vertical, diagonal, and any such combinations in higher dimensions — of grid points, that is of cards.
This encapsulates property 1 about a gset for $s = 3$ or any s for that matter.

Definition 6 (Game-set line (gset line))**.** A game-set line, shortened to gset line, in the s, f-version is a line in the s, f-grid which runs past the edge of the grid and returns to the other side, parallel to

$$\sum_{j=0}^{f-1} i(j)\mathbf{e}_j \text{ where } i(j) \in \{-1, 0, 1\}, \exists j_0 \in \{0, \ldots, f-1\}, i(j_0) \neq 0$$

and $\mathbf{e}_j$ are the orthonormal unit basis vectors.

As this definition is slightly complex, and poorly-worded, let's elucidate it with an example. A gset in the original game with $t = 1, l = 3$ could be composed of three red (same colour) cards, each displaying two (same number) ovals (same shape), but each having a different shading.
The gset line in that case is simply a row (or a column).
If $t = 2$ and two features differed, the gset line would be a diagonal in a plane, that is, the diagonal of a square.
If $t = 3$ and three features differed, the gset line would be the diagonal of a cube.
This continues for higher dimensions.

But, those lines need not be the specific diagonals of, say the plane square. Take the $3, 2$-version and 3×3 grid.
Clearly the two diagonals are gset lines and therefore represent gsets. However, translating those diagonals down once and wrapping back around from the bottom to the top defines another perfectly valid gset.

The notion of "running past or wrapping around the edge" is mathematically a statement of modular equivalence. Coordinates of cards (which are in $\{0, 1, \ldots, s-1\}$) are only unique up to a constant multiple of s term.
We'll therefore write for the i-th coordinate of card $\mathbf{a}$ in the s, f-version: $a_i \mod s$.

The reason the direction of a gline takes its rather abstruse sum form in definition 6 is that only real diagonals or straight (rows, columns, etc.) directions form valid gsets. Moving to the right twice and down only once to draw a line would not constitute a gline.
The condition that not all directions of the gline vanish, that is that at least one direction be ± 1, is simply to ensure that staying in place, at a point in the grid, is not considered a line.

This association between lines and gsets is intrinsically tied to the crucial property that we note here.

Property 2 (Number of needed cards)**.** For any pair of cards, or any pair of points on the grid, there is at least one set of $s-2$ cards that forms a gset with them.

Given the peculiarity of the case of $s = 3$, we will study it in the most depth. As we'll see, versions with higher s enable one to permute lines (or, more accurately, $(f-1)$-dimensional planes) in the grid in a slightly more non-trivial way and defy the essence of the game.
For $s = 3$, any such permutations either do alter the grid but without effect or don't in any meaningful way depending on the counting scheme.

The formulation of the problem then becomes:

Problem. Find the maximal $m_{ax}(s, f) \in \mathbb{N}$ number of cards such that no set of s cards taken from a board of $m_{ax}(s, f)$ cards is a gset in the s, f-version.

Evidently, a possible approach would be to simply construct boards of m cards without a gset, until a board with $m+1$ cards necessarily contains a gset (which can be checked explicitly given the finite nature of the problem). The elegance of the proof however would reside in finding a less manual means of showing that m is indeed maximal in the sense outlined by the problem.
In either case, an algorithmic means of creating such boards will be very powerful to determine what that threshold m_{ax} is. The real challenge then remains: to demonstrate that any number of cards greater than our highest m shown necessarily includes a gset. An improved expression of the problem statement follows:

Problem. Find the minimal $m_{in}(s, f) \in \mathbb{N}$ number of cards such that any board of $m_{in}(s, f)$ cards contains at least one gset in the s, f-version.

3 Basic combinatorics for the case of 3

For $s = 3$, as in the original game, many combinatorial formulae can be derived. The number of gsets can easily be found by simply counting the number of pairs of cards, and accounting for the permutations of the order.
One immediately finds a number of game-sets of

$$L_{c3}(f) = \frac{\binom{n}{3-1}}{3} = \frac{\binom{n}{2}}{3} = \frac{3^f(3^f-1)}{2\times 3} \tag{1}$$

(the notation will become clearer later).

The number of gsets that a given card can be part of can be found by a similar method.
For the original game's case of $s = 3$, first pick any other card, so $n-1$ choices. Then only one third card completes the set. Thus the third card is fixed by the first two. But one must account for the permutations, that is, one cannot double count by picking the second card as one that has already been picked as a third card for another game-set. Therefore one must divide by 2.
This yields $\frac{n-1}{2}$ game-sets a given card can be part of. This actually holds for all f for $s = 3$.

One could define an equivalence relation between sets of $s-1$ cards whose card needed to complete their gset is the same. This approach was not explored further because it essentially equates to drawing all glines that cross a given point on the grid (so can be visualised by simply drawing lines outward from a point).

However, the issue in increasing the number of cards per game-set s — that is, increasing the number of options per feature — from 3 (as in the original game) lies here: game-sets are no longer unique.

4 Higher cases

The statement "game-sets are no longer unique" is very ambiguous and therefore will be clarified here.
By that, I meant that, for a pair of 2 cards, there can be multiple hands of $s-2$ other cards that form a game-set with the pair.
This can be trivially checked using the geometric grid approach outlined above, by, for instance in a 4×4 grid, taking a diagonal and swapping the last point with the one above it and the penultimate point with the one below it.
The choice of first studying $s = 3$ options per feature was therefore sensible.

So we are left with a choice. Either we impose a direction of propagation along the features; or we allow this breakdown of uniqueness.

Let's study the former with an example in the $4, 2$-version, which calls for 4×4 grid.
Suppose the first feature is the colour of the dot(s) on the card (red, green, blue or black) and the second feature the number of such dots (1, 2, 3 or 4).

- The first row could comprise the cards with 1, 2, 3 or 4 (in ascending order) red dots.
- The second row could comprise the cards with 1, 2, 3 or 4 (in ascending order) green dots.
- The last row would then comprise the cards with 1, 2, 3 or 4 (in ascending order) blue dots.
- The last row would then comprise the cards with 1, 2, 3 or 4 (in ascending order) black dots.

By imposing a direction of propagation, we are explicitly not allowing us to permute the way we ordered the rows (so the colours) or the columns (so the numbers).
We are specifically forbidding a certain type of link between both features.
The set of cards (red, 1), (green, 2), (blue, 3), and (black, 4) is a gset. But, the set of cards (red, 1), (green, 2), (black, 3), and (blue, 4) is not a gset because it can be rearranged to (red, 1), (green, 2), (blue, 4), and (black, 3) which is not valid.
This ... makes no sense as a choice for the game. My decision to order the columns as red, green, blue, then black was completely arbitrary and should have no sway on the gameplay.
A game with this muddled rule set would be a genuine challenge to play in practice, owing to the additional mental burden of accounting for orders of features.
It is, however, a natural generalisation of the action of counting lines in the grid.
As it's actually somewhat easier, we'll tackle the more instructive breakdown of uniqueness.

What the above demonstrates is that, in a very genuine sense, the versions with $s = 3$ are truly special.
Property 2 for $s = 3$ versions becomes:

For any pair of cards, or any pair of points on the grid, there is at least one set of $3 - 2 = 1$ card (that is, at least one card) with which it forms a set.

Evidently, there cannot be multiple such cards.
Say the pair of cards differ in t features and have $l = f - t$ features in common.
We can order the features by first listing the differing ones and then the ones in common: $(d_1, d_2, \ldots, d_t, c_1, \ldots, c_l)$ where d_i $(i \in \{1, \ldots, t\})$ is a differing one and c_i $(i \in \{1, \ldots, l\})$ is one in common.
The cards in the pair can be represented by the following vectors: $\mathbf{a}^{(1)} = (a_1^{(1)}, a_2^{(1)}, \ldots, a_t^{(1)}, a_{t+1}^{(1)}, \ldots, a_{t+l}^{(1)})$ and $\mathbf{a}^{(2)} = (a_1^{(2)}, a_2^{(2)}, \ldots, a_t^{(2)}, a_{t+1}^{(2)}, \ldots, a_{t+l}^{(2)})$.
Based on how we defined the pair, $a_i^{(1)} \neq a_i^{(2)}, \forall i \in \{1, \ldots, t\}$ and $a_i^{(1)} = a_i^{(2)}, \forall i \in \{t+1, \ldots, t+l\}$.
The third card in the gset must differ in the t features and have those l features in common by virtue of forming a gset with the pair.

Its vector representation in the grid would be

$$\mathbf{a}^{(3)} = (a_1^{(3)}, a_2^{(3)}, \ldots, a_t^{(3)}, a_{t+1}^{(3)}, \ldots, a_{t+l}^{(3)})$$
$$\text{where } a_i^{(1)} \neq a_i^{(3)} \neq a_i^{(2)}, \forall i \in \{1, \ldots, t\}$$
$$\text{and } a_i^{(1)} = a^{(3)} = a_i^{(2)} = a_i, \forall i \in \{t+1, \ldots, t+l\}.$$

As $s = 3$, there are 3 options for each feature so, for any one of the first t features in which the cards differ, that feature of the third card in the gset is constrained to one value.

To explicit the logic here, in the first feature in which they differ, there's one remaining choice for the third card, so that fixes that feature.
In the second, the same so that fixes that feature.
So on and so forth. In any feature in which they don't differ — therefore share a common value —, the third card's feature is constrained to be that common value.
So every single feature of the third card is constrained to a single value. By uniqueness, there can be only one such card.

It should be noted that, throughout this exposition, the dimensionality of the grid — that is f, the number of features — has not affected the complexity of the problem.
One may therefore wonder why $f = 4$ in the original *Set*. The likely reason is that it strikes a sensible balance among the core facets of the game's development, namely its difficulty, its portability, its aesthetic, etc.

5 A lower bound

A simple geometric application of the grid representation provides a weak lower bound for the problem, that is for $m_{ax}(s, f)$.

For example, for the $3, 2$-version, we obtain a square grid of 9 points, where rows have a constant first feature and columns a constant second feature. Diagram
In complete analogy to the description in section 4, we'll suppose the first feature is the colour of the dot(s) on the card (red, green, or blue) and the second feature the number of such dots (1, 2, or 3).

- The first row could comprise the cards with 1, 2, or 3 (in ascending order) red dots.
- The second row could comprise the cards with 1, 2, or 3 (in ascending order) green dots.
- The last row would then comprise the cards with 1, 2, or 3 (in ascending order) blue dots.

As we're discussing an $s = 3$ version, each column is therefore a gset, as is each diagonal line, allowing for continuity past the edge, for instance from the right edge to the left edge.
The total number of game-sets is simply the total number of glines, which, in the 2-dimensional 3×3 case described above, gives 12 by spanning the horizontal, vertical and diagonal lines.

The lower bound can then be immediately identified by simply excluding one option for each feature.
The reason this eliminates any possibility of forming a gset follows from the definition 1.
If we exclude one option from one feature, then any card in a hand of s cards cannot have that option for that feature. Therefore, that feature cannot differ among the cards so it must be the same for all the cards. If it weren't, our hand could for instance include two red cards and one blue card (if we eliminated green).
If we exclude one option from another feature, the same principle applies so that feature cannot differ among the cards.
Repeating this process of eliminating one option for all the features means we are left with s cards in our hand that have identical features, an impossibility unless we're treating the trivial $1, f$-version with only one card in the entire game.

In the two-dimensional 3×3 case described above, simply omitting the third colour (say green) and third number of items on a card (say 3 items) leaves you with a 2×2 grid of $2^2 = 4$ cards that cannot contain a set.
Generalising to f features with s options per feature, this provides a lower bound of $(s-1)^f$, effectively the size of the rank f tensor where each row has a length of 1 less than the original tensor.

This geometric view can be generalised to a non-cubic, but rather a rectangular prism shape, should the features not all have the same number of options $(\exists i, j \in \{1, \ldots, f\}, i \neq j, s_i \neq s_j)$.

6 The cap set

Unfortunately, my musings from my youth on the matter were not novel.
The problem at hand for $s = 3$ is effectively the cap set problem.
For *Set*, the $3, 4$-version, it was proven in 1971 that the largest group of cards that can be put together without creating a game-set is 20.
Our sense that 21 cards on the board guaranteed the presence of a gset was not unfounded.

The cap set problem, with $s = 3$, is indeed very unusual. Only cards in a gset sum to the zero vector, a mysteriously beautiful property. The treatment of $s > 3$ is far more obscure and complex, owing to points mentioned above.

Enticed by the name of this problem, we can forge on in the combinatorics of the matter.

7 General combinatorics

We're looking to count the total number of glines, so lines in the s, f-grid.
The direction are combinations of unit basis vectors.
To define a direction, we can choose only one basis vector, any pair, any triplet, etc., until we're left with choosing all of them.
As f is the dimensionality of the grid, there are f basis vectors.
The number of combinations is therefore

$$\begin{aligned} C_d(s,f) &= f + \frac{f(f-1)}{2} + \ldots + 1 = \left(1 + f + \frac{f(f-1)}{2} + \ldots + 1\right) - 1 \\ &= \sum_{k=0}^{f} \binom{f}{k} - 1 \\ &= \sum_{k=0}^{f} \binom{f}{k} 1^k \times 1^{f-k} - 1 = (1+1)^f - 1 = 2^f - 1. \end{aligned} \tag{2}$$

However, the direction given by a combination of basis vectors should account, only where necessary, for the direction along the relevant basis vectors, that is forward or backward.
If we're holding one feature constant among the gset cards, then only one feature can vary. We are therefore constrained to a horizontal or vertical line. Moving up or down that column, or moving left or right along that row, doesn't change the cards that compose the gset associated with the gline.
If both features differ among the gset cards, the line is diagonal. However, there are multiple diagonals: "/" and "\" in a 2-dimensional grid.
Likewise, there are multiple diagonals of a cube: 4 to be exact.
There are in general 2^{d-1} diagonals in a d-hypercube.
This can be seen by considering $\mathbb{R}^d$, counting the number of d-dants (quadrants, octants, etc.) and dividing by 2 as moving backwards along the same direction doesn't change the cards forming the gline.
Our number of useful directions is therefore

$$\begin{aligned} D(s,f) &= f \times 1 + \frac{f(f-1)}{2} \times 2 + \ldots + 1 \times 2^{f-1} \\ &= \frac{1}{2}\left(1 \times 2^0 + f \times 2^1 + \frac{f(f-1)}{2} \times 2^2 + \ldots + 1 \times 2^f\right) - \frac{1}{2} \\ &= \frac{1}{2} \sum_{k=0}^{f} \binom{f}{k} 2^k - \frac{1}{2} \\ &= \frac{1}{2} \sum_{k=0}^{f} \binom{f}{k} 2^k \times 1^{f-k} - \frac{1}{2} = \frac{(2+1)^f}{2} - \frac{1}{2} = \frac{3^f - 1}{2}. \end{aligned} \tag{3}$$

If the appearance of a 3 in the above formula is suspicious to you, your intuition is correct.
As we'll see Equation 3 only provides a useful counting scheme for versions with $s = 3$.
However, the logic behind this scheme provides valuable insight for versions with greater s so we'll continue down this garden path.

To find the total number of glines, we simply need to translate those directions as many times as possible until we return to our starting point.
For any diagonal line in any hyperplane, we can evidently do so s times in any of $f - 1$ orthonormal basis directions (directions given by a unit basis vector $\mathbf{e}_i$).
For any line parallel to a basis vector $\mathbf{e}_j$, the same is true as $f-1$ basis directions are not parallel to $\mathbf{e}_j$.
This almost certainly requires a visual example to be understood.

In the case of $s = f = 3$, our grid is the $3 \times 3 \times 3$ cube of points.
Take a line parallel to a basis vector ($\mathbf{e}_j$ with $j \in \{1, 2, 3\}$), so a row, a column or a "depth-column" (a row in the direction of depth, usually set to be the z-direction). Without loss of generality, say $\mathbf{e}_j = \hat{\mathbf{x}}$ ($j = 1$) so we're discussing a row.
Any translation of that line along a multiple of $\hat{\mathbf{x}}$ has no effect on the cards forming the line.
However, the line is not invariant under translation in the remaining basis vector directions. It can be translated s times — s times as coordinates are defined modulo s — in each of those directions for a total of $s^{f-1} = 3^2$ times.
Diagonal lines can be treated in a very similar fashion.
Take a diagonal line in the first face of the cube. It can be translated s times to enter each of the different faces, and s within that face without invariance, so again $s^{f-1} = 3^2$.
The diagonal lines of the entire cube are also subject to the same s^{f-1} factor.
Multiplying Equation 3 by this factor provides the total number of glines so the total number of gsets,

$$L_c(s, f) = s^{f-1}D = \frac{s^{f-1}(3^f - 1)}{2}. \tag{4}$$

Astute readers will notice that

$$L_c(3, f) = \frac{3^{f-1}(3^f - 1)}{2} = \frac{3^f(3^f - 1)}{2 \times 3} = \frac{\binom{n}{2}}{3} \tag{5}$$

as $n = s^f = 3^f$.
This is exactly our formula from Equation 1, as it should be.

However, as was alluded to earlier, this counting scheme is incorrect for $s > 3$.
It does correctly count the number of lines in the grid as it is currently laid out, but it forgets to account for the different, non-trivial permutations of the

layout.
$L_c(s, f)$ would be the correct and useful number of gsets in the s, f-version without uniqueness breakdown, as defined in section 4.

To think about those permutations, let's return to 2-dimensional grids so versions with $f = 2$.
The rows and columns can still be treated without issue so we should focus on the diagonals.
In our counting scheme leading to Equation 4, we found that there are $2^{f-1} = 2$ different diagonals ("\" and "/") and then translated both of those the appropriate number of times.
Another counting scheme makes use of our idea of permuting lines or planes.
Instead of considering different directions along which we can move for a given line, fix a set of directions for each dimension. For instance, rows will always be traversed left-to-right, columns always top-to-bottom, "depth-columns" front-to-back, etc.
Then consider permuting or reordering the labels of the features that characterise the rows, columns, etc.
For instance, in the 3×3 grid above, rather than columns being ordered as 1|2|3, swap the 2 and 3 columns to yield 1|3|2. Clearly, this doesn't change the way that the rows are traversed; and it obviously cannot in any way impact the way the columns are traversed (as they're simple being moved).
However, this permutation will impact the diagonal lines. It would assert that there is only one "diagonal direction", say "\", and that all the other diagonal lines we need to count result from arranging the columns of the $s \times s$ grid in all possible ways.
The count is correct as, by arranging the columns in different orders, one is effectively arguing that the first point/card in the "\" diagonal could be any column's card in their first row; the second point could be any of the remaining columns' card in their second row; so on until the last point in the "\" diagonal at the bottom-right edge is constrained to be one value.
There are of course $s!$ different arrangements of the columns so $s!$ different glines in that 2-dimensional grid (or 2-dimensional plane of a higher-dimensional grid for $f > 2$).
Now we can tackle higher-dimensional grids. For instance, let's think about an $f = 3$ version, requiring a cubic grid of length s.
The diagonals in the faces can be treated by simply returning to the 2-dimensional grid and then multiplying by the total number of such planes.
The diagonals of the entire cube (that is, the ones pointing along $\mathbf{d} = \hat{\mathbf{x}} + \hat{\mathbf{y}} + \hat{\mathbf{z}}$) can be counted via a slightly involved, yet straightforward generalisation of those of the 2-dimensional grid.
Let's look at our cube from above, meaning we are seeing an $s \times s$ grid.
A line parallel to our one and only diagonal direction $\mathbf{d}$ can be uniquely created by choosing our first point as any of the s^2 cards in the $s \times s$ grid; our second point as any of the $(s-1)^2$ points in the sub-grid that doesn't contain the row or column of the first point; our third point as any of the $(s-2)^2$ in the smaller

sub-grid; and so on until the last point is constrained to one value.
There are therefore $s^2 \times (s-1)^2 \times \cdots \times 1 = (s!)^2$ possible arrangements.
Generalising to higher dimensions, we expect the number of arrangements of points forming d-diagonals (the diagonals in an d-dimensional cube) to be $(s!)^{d-1}$.
To find the total number of glines (or of such diagonals), we simply need to account for the number of d-hypercubes in our s, f-grid.
1-hypercubes (so rows, columns, etc.), of which there are f, can be translated s^{f-1} times; 2-hypercubes (so faces of the cube), of which there are $\binom{f}{2}$, can be translated s^{f-2} times; 3-hypercubes (so cubes), of which there are $\binom{f}{3}$, can be translated s^{f-3} times...
Our total number of lines is therefore

$$\begin{aligned} L(s,f) &= fs^{f-1} + s!\binom{f}{2}s^{f-2} + (s!)^2\binom{f}{3}s^{f-3} + \cdots + (s!)^{f-1}\binom{f}{f}s^{f-f} \\ &= \sum_{i=1}^{f}(s!)^{i-1}\binom{f}{i}s^{f-i}. \end{aligned} \tag{6}$$

Finding the total number of gsets that include a given card should be a simple restriction of the above calculation.
Take a given card, that is a given vector in the grid, and call it **a**.
To aid the process of visualisation, let's think about the simplest card: the origin.
Via f permutations, one can always move the card under consideration to the origin, the $(0, \ldots, 0)$ vector, the upper-left corner in the most intuitive representation.
Therefore, treating the case of the card at the origin will always be sufficient.

When only one feature differs (a horizontal or vertical line), as we saw, no impact ensues.
When two features differ, swapping any two of the other rows or other columns will result in a different gset that includes the origin.
For the s, f-version, the number of ways of arranging $s-1$ rows (to exclude the origin's) is $(s-1)!$. It essentially ends up being the number of diagonals in the sub-grid of size $(s-1) \times (s-1)$.
Continuing this logic in every direction, we notice that we are effectively repeating the same counting scheme as before for the total number of lines in Equation 6.
However, now we should no longer count translations of the hypercubes, because we have fixed the origin to be in the gset. If we were to, say, translate the row that contains the origin down, it no longer would contain it and shouldn't be counted. This removes the factor s^{f-i}.
In addition, as we're now only concerned with hypercubes of length $s-1$ rather than s, the factor $(s!)^{i-1}$ will become $(s-1)!^{i-1}$.

The total number of gsets that include a given card, $g_c(s,f)$, is therefore

$$\begin{aligned} g_c(s,f) &= f + ((s-1)!)^{2-1}\binom{f}{2} + \cdots + ((s-1)!)^{f-1}\binom{f}{f} \\ &= \sum_{i=1}^{f}((s-1)!)^{i-1}\binom{f}{i} \end{aligned} \quad . \tag{7}$$

Perhaps a more straightforward means of finding the number of gsets that include a specific card is simply to make use of what we have already derived. We found in Equation 6 a formula for the total number of lines, and therefore the total number of gsets via an involved counting scheme.
Another combinatorial approach would be to count the number of gsets containing a specific card, multiply by the total number of cards, and divide appropriately to avoid overcounting gsets.
The first two factors would be $g_{\text{card}} \times s^f$.
As for the overcounting factor, each of those gsets contain $s-1$ other cards who are also attributed a factor g_{card}. That factor recounts the same gset once again for each of those $s-1$ other cards. We must therefore divide by s.
This results in

$$L(s,f) = \frac{g_c(s,f) \times s^f}{s} = g_c(s,f) \times s^{f-1}$$

$$\Rightarrow g_c(s,f) = L(s,f) \times s^{-(f-1)} = \sum_{i=1}^{f}(s!)^{i-1}\binom{f}{i}s^{1-i} = \sum_{i=1}^{f}((s-1)!)^{i-1}\binom{f}{i},$$

which just so happens to be Equation 7 (ah, the satisfaction of mathematical consistency).

It should be noted that the point concerning overcounting differs from the number of times two same cards appear in a gset together.
For instance, in the $4,2$-version, we can arrange the board as usual.

- 1, 2, 3, or 4 red dots.
- 1, 2, 3, or 4 green dots.
- 1, 2, 3, or 4 blue dots.
- 1, 2, 3, or 4 black dots.

As we need only focus on the origin (the card with 1 red dot), we can count the number of gsets that include it, distinguishing between those that differ in $t = 1$ feature or $t = 2$ features.
In the case of $t = 1$ differing feature, either colour or alternatively number differs, so only rows or columns are to be counted.

$g_1 = \{(0,0),(0,1),(0,2),(0,3)\}$ (the first row)
$g_2 = \{(0,0),(1,0),(2,0),(3,0)\}$ (the first column).

In the case of $t = 2$ differing feature, both colour and number must differ, so only diagonals (with possible permutations of rows or columns) are to be counted.

$g_3 = \{(0,0),(1,1),(2,2),(3,3)\}$ (the "\" diagonal)
$g_4 = \{(0,0),(3,1),(2,2),(1,3)\}$ (the wrapped-around "/" diagonal)
$g_5 = \{(0,0),(1,1),(3,2),(2,3)\}$
$g_6 = \{(0,0),(2,1),(1,2),(3,3)\}$
$g_7 = \{(0,0),(2,1),(3,2),(1,3)\}$
$g_8 = \{(0,0),(3,1),(1,2),(2,3)\}$

This yields the set of gsets that characterise the origin, that is $g(0,0) = \{g_1, \ldots, g_8\}$.

Ultimately, as I endeavoured to make clear in section 4, the arrangement I chose for the board was arbitrary. Choosing to place the "2" column after the "1" column, although seemingly natural, is a matter of convenience, and certainly not how the cards would actually be dealt in a real game.

The enumeration of gsets in such a manner reveals a striking beauty of the underlying structure.
Unsurprisingly the cards in the gsets with $t = 1$ appear only once: the card $(0,1)$ is only listed once in $g(0,0)$.
The cards in the gsets with $t = 2$ appear twice.
In particular, cards will appear $\frac{(s-2)^t}{2}$ times with the origin so $\frac{(s-2)^t}{2}$ times in $g(0,0)$.
This is a mere restatement of the previous derivation. The origin and the second card (say $(\underbrace{1,\ldots,1}_{t \text{ ones}}, \underbrace{0,\ldots,0}_{l=f-t \text{ zeros}})$ in the s, f-version) differ in t features and share $l = f - t$ features in common.
Together they therefore form a t-dimensional hypercube of side length $s-2$ (as two cards in the gset have already been chosen). Now remain to be counted the diagonals of that hypercube, a procedure with which we're very familiar (counting the number of t-dants and dividing by 2).
Hence follows the formula.

While the combinatorics of the game are fascinating, and very insightful on a geomeotrical level, rederiving the solution to the cap set problem seems somewhat redundant.
So, rather than test different boards of cards by hand as my father and I did years ago, perhaps let's replicate the process digitally.

8 Digital testing

My language of choice was Python due to its range of data types and built-in methods that complemented the geometric nature of my approach.
The vectors of a card was represented as a tuple.

A gset took the form of a "set" datatype of the s tuples of cards (ensuring uniqueness and eliminating order from the matter).
Many functions provided by the `itertools` and `NumPy` modules drastically simplified the process, notably when it came to representing the board as the $\{0, 1, \ldots, s-1\}^f$ space.

The generation of all gsets for the s, f-version was facilitated by the logic explained in section 7, and, in particular, the principle of permuting all possible directions which define all possible glines.
The implementation of the testing relied on simply shuffling the deck of cards, then adding cards to the board until a gset was found.
Once such a gset was found, the number of played cards was returned.
The output of running that function hundreds of times could be plotted as a histogram, which in theory would make the strict minimum number of played cards with a guaranteed gset immediately visible (as the last bin of the plot).

The efficiency was mainly limited by the procedure of determining whether a board of n played cards contained a gset yet.
The simpler option, for which I opted, was to find all combinations of s cards from the n played cards – these combinations therefore being possible gsets – and to check whether any were actually valid gsets by checking whether they were in the pre-generated list of all gsets.
Another solution would have been to adapt the geometric tools employed above, by selecting any played card and checking whether all cards in any direction were also played. The permutation of lines would still be needed and the implementation would likely become cumbersome.
Regardless, the chosen method yields a complexity dominated by the and therefore bounded above by

$$\prod_{n=s}^{\text{gset found}} \binom{n}{s} \sim (n_{\text{cap}}!)^{n_{\text{cap}}},$$

where n_{cap} is the size of the cap set for the relevant version.
Although not quite the most efficient algorithm, the code elegantly emenates the geometric and combinatorial principles presented here.

9 Faulty reasoning

I initially phrased property 2 as:

"For any set of $s-1$ cards, there is one and only card that forms a gset with them."

better formatting

which is clearly false (simply take a line with one missing card and move one point from that line). A correct statement would be

"For any set of $s-1$ cards, there is at most one card that forms a gset with them."

better formatting

The mistake stemmed from poorly generalising the $s = 3$ case.

10 Non-cubic grid

To be frank, I can't imagine playing a version of the game where different features have different numbers of options.
Gsets of different sizes would be possible and diagonal glines therefore wouldn't have a fixed length.
More importantly to game-makers and publishing houses (especially to the sales teams), the design of the cards would lose its symmetry and customers may be less enticed.