

BRUHAT INTERVALS THAT ARE LARGE HYPERCUBES

JORDAN ELLENBERG, NICOLAS LIBEDINSKY, DAVID PLAZA, JOSÉ SIMENTAL,
AND GEORDIE WILLIAMSON

ABSTRACT. We study the question of finding big Bruhat intervals that are poset hypercubes in the symmetric group S_n . Using permutations suggested by AlphaEvolve (an evolutionary coding agent developed by Google DeepMind), we were led to an unusual situation in which the agent produced a pattern which performed well for the n tested, and which we show works well for general n . When n is a power of 2 we exhibit a hypercube of dimension $O(n \log n)$, matching the largest possible dimension up to a constant multiple. Furthermore, we give an exact characterization of the vertices of this hypercube: they are precisely the *dyadically well-distributed* permutations—a simple digitwise property that already appeared in connection with Monte Carlo integration and mathematical finance. The maximal dimension of a Bruhat interval that is an hypercube in S_n gives a lower bound (and possibly is equal to) the maximal possible coefficient of the second-highest degree term in the Kazhdan–Lusztig R -polynomial in S_n . As a surprising consequence, we obtain a new lower bound of order $n \log n$ for the maximal number of frozen variables appearing in the cluster algebras attached to the open Richardson varieties in S_n , and a similar result for moduli spaces of embeddings of Bruhat graphs.

1. INTRODUCTION

Consider the symmetric group S_n equipped with the Bruhat order. We study the question of how large a hypercube (i.e. boolean) interval can occur inside S_n . To the best of our knowledge, this problem was first posed in Reading’s thesis [Rea02], which established lower bounds for all finite irreducible Coxeter groups. In the specific case of S_n , he showed that the largest hypercube has at least rank

$$n - 1 + \left\lfloor \frac{n - 2}{2} \right\rfloor.$$

Further works that revolve around this problem include [Ten07; HV09; Ten22; LMP21].

In this paper, we find an interval in S_n which is poset isomorphic to a hypercube whose dimension is within a constant multiple of the maximal possible size.

More precisely, our main theorem is the following (see § 2 for undefined terms):

Theorem 1.1. *Consider the symmetric group on 2^m elements, for $m \geq 1$. The set of dyadically well distributed permutations forms an interval in Bruhat order. This interval is poset isomorphic to a hypercube of dimension $m2^{m-1}$.*

By Stirling’s formula, any hypercube inside S_{2^m} is of dimension bounded by $m2^m$. Thus the dimension of our hypercube is within a constant factor of largest possible.

The interval referred to in Theorem 2.21 consists of permutations which are “dyadically well distributed,” a number-theoretic condition related to binary expansion. We remark that the definition of dyadically well distributed permutations does not involve any Lie theoretic data so the fact that this set is indeed a Bruhat interval is quite striking. To the best of the authors’ knowledge this is the first instance this phenomenon occurs in a non-trivial way.

Date: January 6, 2026.

1.1. **Background.** The existence of natural and large hypercubes in Bruhat order is surprising. Anyone who experiments with Bruhat order will notice that the majority of small intervals are small hypercubes. It is also easy to see that the interval between the identity and any Coxeter element (for example, $s_1 s_2 \dots s_{n-1}$, where s_i denotes a simple transposition) is a hypercube of dimension $n - 1$. It is perhaps surprising that there exist hypercubes of dimension larger than $n - 1$ (see Figure 1). Moreover, as the intervals become larger it becomes harder and harder to find hypercubes (see Table 1). This might lead one to guess that there are no hypercubes of superlinear dimension in Bruhat order. Our main result shows that the intuition gained from these small examples is far from the reality.

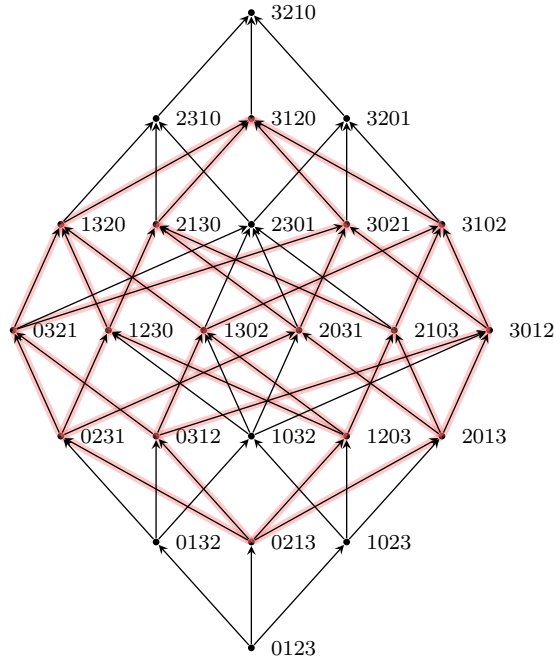


FIGURE 1. The first “unexpected” hypercube in S_4 of dimension 4. This hypercube is also the first interesting example of our construction.

A similar phenomenon occurs in cluster algebras. It was long conjectured, and recently proved, that open Richardson varieties attached to pairs of permutations admit cluster structures [Cas+25a; GLS]. A crucial invariant of a cluster variety is its number of frozen variables, which is roughly the maximal possible rank of a faithful torus action. Initially, it was thought that this number could be at most $n - 1$ in S_n , as this is the dimension of the naturally acting maximal torus in $SL(n)$. But one can find open Richardson varieties for S_n that are tori of rank greater than $n - 1$. The realization that this number of frozen variables is mysterious provided an important step in finding the cluster structure. After cluster structures on open Richardson varieties had been constructed, it was realised that the number of frozen variables is equal to the Kazhdan-Lusztig d -invariant (see § 3.3), which is readily computable, and easily seen to occasionally exceed $n - 1$. The intervals of this paper provide examples where the number of frozen variables is $n \log n$, greatly exceeding any known family of examples.

A related phenomenon occurs in studying the combinatorial invariance conjecture of Dyer and Lusztig. Here the problem is to prove that isomorphic Bruhat intervals give rise to equal Kazhdan-Lusztig polynomials. Any Bruhat interval has a natural embedding inside \mathfrak{h}^* , the dual of the Lie algebra of a maximal torus. It is well known that one can

recover the Kazhdan-Lusztig polynomial from this embedded graph. Thus, it is natural to try to prove the conjecture by relating embeddings associated to different incarnations of isomorphic intervals. Surprisingly, geometric embeddings of isomorphic intervals can have greatly differing dimensions. For example, hypercubes may appear embedded in a space of dimension much lower than their “natural” dimension. In an attempt to overcome this difficulty, Hone, Klein and the last author introduced a notion of a non-degenerate embedding, and study their moduli. In striking similarity to the cluster algebra story, the dimension in which this embedding lives is difficult to compute, appears related to the d -invariant, and may be considerably larger than $n - 1$ (the dimension of \mathfrak{h}^*). The examples of this paper provide intervals where this “natural dimension” is much higher than naively expected.

A related motivation for the consideration of hypercubes in Bruhat order is again related to the combinatorial invariance conjecture. Motivated by AI experiments, Blundell, Buesing, Davies, Veličković and the last author conjectured a new formula for Kazhdan-Lusztig polynomials based on the Bruhat graph [Dav+21; Blu+22], which (if true) would imply combinatorial invariance. This work has led to renewed interest in combinatorial invariance [BM25; GW24; BG23] and a proof for elementary intervals [BG25]. The formula relies on the choice of “hypercube decomposition” of the interval, which provides an iterative way to build Bruhat intervals from hypercubes. In [Blu+22], a simple construction proves that one can do this in $n - 1$ steps for any interval in S_n , however experimentally one can often be much more efficient. This efficiency is related to the existence of unexpectedly large families of hypercubes.

k	hypercubes	total	ratio
3	223704	241620	0.9259
4	286108	387932	0.7375
5	231484	498176	0.4647
6	111064	536860	0.2069
7	27484	502031	0.0547
8	2736	417142	0.0066
9	64	313063	0.0002
10	0	214478	0.0000
11	0	134933	0.0000
12	0	78104	0.0000

TABLE 1. Hypercubes of length k / total intervals of length k for S_7

1.2. Methodology. The large hypercube that is the subject of this paper was discovered using a novel experimental methodology. Namely: we searched for pairs of permutations in S_n , not by any form of exhaustive search (which would be impractical for any but the smallest n) but using a protocol called AlphaEvolve, recently developed by Google DeepMind. This protocol is an evolutionary algorithm in which programs evolve in time, using a large language model as the mode of reproduction, with “fitness” determined by the program’s ability to generate output that meets a user-specified mathematical objective. In this case, we were searching for pairs of permutations whose Kazhdan-Lusztig d -invariant was as large as possible.

We will describe our experimental process in some detail in Section 4, in the hope that this description will be useful for other mathematicians interested in adopting these methods.

There is already quite a bit of evidence that AlphaEvolve and related protocols can be used to generate examples of mathematical interest (see e.g. [Rom+24; Ell+25; Geo+25].) Quite naturally, a great deal of the research so far has focused on problems that have already been the subject of substantial mathematical effort; in dozens of cases, researchers have been able to use AlphaEvolve to match or incrementally improve upon the best examples in the existing literature. Typically (though not universally) the incremental improvements are difficult to interpret; for instance, the large cap set in $(\mathbb{Z}/3\mathbb{Z})^8$ found in [Rom+24] is larger than any yet discovered, but so far we have not been able to concoct a satisfying story of “why” this cap set is so large, and it is possible that no such story exists.

The present project is somewhat different. The problem of finding large d -invariants, or large hypercubes, is one of natural interest, but there is not a large existing literature about it. And the example we found using AlphaEvolve is qualitatively different from the best one we knew, having a larger asymptotic order of growth in n . What’s more, the permutations (more precisely, the program that generated the permutations) is highly interpretable; within a few days we were able to explain to ourselves why it worked, and thus to prove that the machine-generated examples for small n in fact generalized to all n . In particular, our intuition is that the large hypercube presented in this paper is an object of mathematical interest in its own right, and that there is more to learn about and from it.

ACKNOWLEDGEMENTS

This paper involves essential contributions from Adam Zsolt Wagner, who cannot be named as a coauthor for technical reasons. This material is based upon work supported by the National Science Foundation under Grant No. DMS-1929284 while JE, NL, DP, JS and GW were in residence at the Institute for Computational and Experimental Research in Mathematics in Providence, RI, during the “Categorification and Computation in Algebraic Combinatorics” semester program. JE’s work was also supported by NSF grant DMS-2301386 and a Simons Fellowship. NL was partially supported by FONDECYT-ANID grant 1230247. DP was partially supported by FONDECYT-ANID grant 1240199. JS was partially supported by UNAM’s PAPIIT Grant IA102124 and SECIHTI Project C-F-2023-G-106. GW was supported by ARC project DP230102982 and the Max Planck-Humboldt research award. The authors would also like to thank AlphaEvolve for stimulating discussions. We are grateful to Leonardo Patimo for pointing us to relevant references, and to Mario Marietti for some helpful email exchanges.

2. DYADICALLY WELL-DISTRIBUTED PERMUTATIONS

In this section we introduce the main object of study of the paper: the class of dyadically well-distributed permutations. We establish their fundamental properties and determine the cardinality of the set of such permutations. In particular, we show that this class forms a Bruhat interval isomorphic to a hypercube, precisely the large hypercube mentioned in the introduction.

2.1. Basic properties. Throughout the paper we fix a positive integer n . We denote by $[n]$ the interval with n elements $\{0, \dots, n-1\}$. We think of S_n as the set of bijections on $[n]$. Henceforth we will assume that $n = 2^m$.

Definition 2.1. For $k \geq 0$, a *basic k -interval* is an interval in $[2^m]$ of the form $[c2^k, (c+1)2^k - 1]$. Henceforth, we identify \mathbb{F}_2^m with $[2^m]$ via

$$(a_0, a_1, \dots, a_{m-1}) \mapsto \sum_{i=0}^{m-1} a_i 2^{m-1-i}.$$

Under this identification, a basic k -interval is determined by fixing the first $m - k$ bits.

For a basic k -interval S , we let P_S denote the unique binary string of length $m - k$ that appears as the first $m - k$ bits of every element of S .

We say that a permutation π is *dyadically well-distributed* if, for every basic k_1 -interval S and k_2 -interval T , with $k_1 + k_2 = m$, there exists exactly one element $i \in S$ such that $\pi(i) \in T$.

We denote by \mathcal{D}_m the set of all dyadically well-distributed permutations in S_{2^m} .

Remark 2.2. For $\pi \in S_{2^m}$ we define its permutation matrix $M_\pi = (m_{i,j})$ by $m_{i,j} = 1$ if $\pi(j) = i$ and 0 elsewhere. Identifying an element $\pi \in S_{2^m}$ with its permutation matrix $M_\pi \in \text{Mat}_{2^m \times 2^m}$, we obtain a more pictorial interpretation of dyadically well-distributed permutations. For each $k = 0, \dots, m$, tile M_π into blocks of size $2^k \times 2^{m-k}$, starting from the upper-left corner and proceeding in a left-to-right, top-to-bottom fashion (i.e., in reading order). Then $\pi \in \mathcal{D}_m$ if and only if each such block contains exactly one entry equal to 1 (see Figure 2).

When $k = 0$ and $k = m$ the above condition is automatically satisfied for any permutation. Thus, to check whether $\pi \in \mathcal{D}_m$ it is enough to check the above condition for $0 < k < m$.

Definition 2.3. By a *fundamental block* in a matrix $M \in \text{Mat}_{2^m \times 2^m}$ we will mean a $(2^k \times 2^{m-k})$ -block as considered in Remark 2.2.

Definition 2.4. We inductively define elements $x_m \in S_{2^m}$ as follows:

$$\begin{aligned} x_1 &= (0, 1) \in S_2, \\ x_2 &= (0, 2, 1, 3) \in S_4, \\ x_3 &= (0, 4, 2, 6, 1, 5, 3, 7) \in S_8, \\ &\vdots \end{aligned}$$

In other words,

$$x_{m+1} = (x_m(0), x_m(0) + 2^m, x_m(1), x_m(1) + 2^m, \dots, x_m(2^m - 1), x_m(2^m - 1) + 2^m) \in S_{2^{m+1}}.$$

Define y_m as the reverse element in one line-notation, i.e.

$$y_m = (x_m(2^m - 1), \dots, x_m(1), x_m(0)).$$

Example 2.5. Let $m = 4$. In this case we have

$$\begin{aligned} x_4 &= (0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15) \\ y_4 &= (15, 7, 11, 3, 13, 5, 9, 1, 14, 6, 10, 2, 12, 4, 8, 0). \end{aligned}$$

The permutation matrices of x_4 and y_4 are depicted in Figure 2. This figure also shows the fundamental blocks, making it evident that x_4 and y_4 are elements of \mathcal{D}_4 .

Remark 2.6. Let us explain why the permutations x_m and y_m are more natural than they may first appear. Identifying the set $[2^m]$ with \mathbb{F}_2^m as before, the permutations x_m and y_m correspond to affine maps in $\text{GL}(\mathbb{F}_2^m) \times \mathbb{F}_2^m \subset S_n$ defined by the formulae:

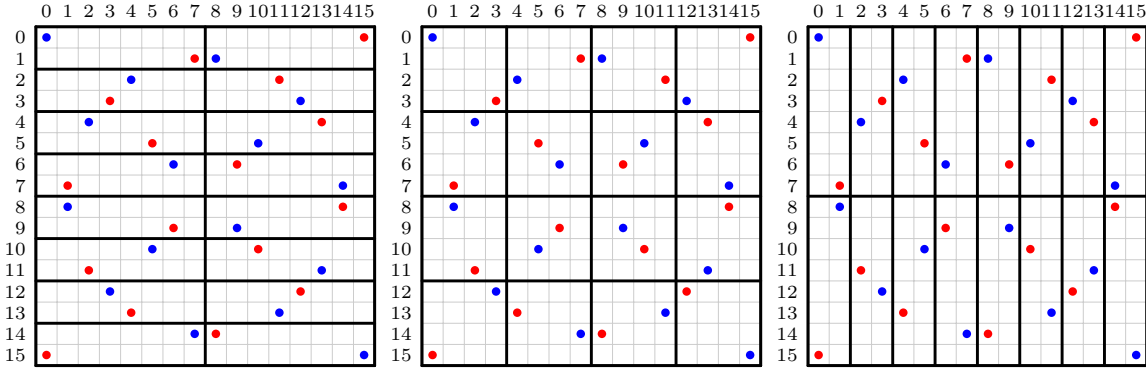
$$x_m(a_0, \dots, a_{m-1}) = (a_{m-1}, \dots, a_0) \quad \text{and} \quad y_m(a_0, \dots, a_{m-1}) = (\overline{a_{m-1}}, \dots, \overline{a_0}),$$

where $\overline{a_i} = a_i + 1$.

For $a, b \in [n]$ we denote by (a, b) the transposition that interchanges a and b .

Definition 2.7. Let $\pi = (x_1, \dots, x_n) \in S_n$. We define the *length* of π , $\ell(\pi)$, as the number of inversions of π . That is,

$$\ell(\pi) = \#\{(i, j) \in [n]^2 \mid i < j \text{ and } \pi(i) > \pi(j)\}.$$



Fundamental blocks of size $2^3 \times 2^1$ Fundamental blocks of size $2^2 \times 2^2$ Fundamental blocks of size $2^1 \times 2^3$

FIGURE 2. M_{x_4} (blue) and M_{y_4} (red) with fundamental blocks highlighted.

In the following lemma we collect some basic facts about permutations x_m and y_m .

Lemma 2.8. *Let $m \geq 1$. Then we have*

- (1) x_m and y_m are involutions.
- (2) $x_m y_m = y_m x_m = w_0$, where w_0 denotes the longest element in S_{2^m} .
- (3) $x_m, y_m \in \mathcal{D}_m$.
- (4) $\ell(x_m) = 2^{m-2}(2^m - (m+1))$ and $\ell(y_m) = 2^{m-2}(2^m + (m-1))$. Thus, $\ell(y_m) - \ell(x_m) = m2^{m-1}$.

Proof. The first two claims follow directly from Remark 2.6.

We now focus on the third claim. By Items 1 and 2 we have $y_m = x_m w_0$. Therefore, it is enough to show $x_m \in \mathcal{D}_m$.

We fix $k \in \{0, \dots, m\}$. Let S and T be basic $(m-k)$ - and k -intervals, respectively. Let $P_S \in \mathbb{F}_2^k$ and $P_T \in \mathbb{F}_2^{m-k}$ be the binary sequence associated to S and T , respectively. In order to show that x_m belongs to \mathcal{D}_m we must verify that there is a unique $i \in S$ such that $x_m(i) \in T$. It is easy to see that the unique i that satisfies both conditions is the one associated to the m -bit binary sequence $P_S P_T^{-1}$, where P_T^{-1} denotes the reverse of P_T . We have thus proved $x_m \in \mathcal{D}_m$.

It remains to prove the length formula for x_m ; the case of y_m being analogous. We recall that $\ell(x_m)$ coincides with the number of inversions, i.e. the number of pairs (i, j) with $i < j$ such that $x_m(i) > x_m(j)$.

It is immediate that $\ell(x_1) = 0$ and $\ell(x_2) = 1$. For $m \geq 3$, we claim that

$$\ell(x_m) = 2\ell(x_{m-1}) + (2^{m-1} - 1)2^{m-2}. \quad (2.1)$$

To see this, write the one-line notation of x_m as a concatenation $x_m = (a_m \mid b_m)$, where each block has length 2^{m-1} . By construction, both a_m and b_m encode copies of x_{m-1} , so together they contribute $2\ell(x_{m-1})$ inversions.

Let C_m be the number of inversions of x_m involving an entry in a_m and one in b_m . Since entries in a_m are even and those in b_m are odd, each entry $k \in a_m$ contributes exactly $k/2$ to C_m . Summing over all even numbers from 0 to $2^m - 1$, we obtain

$$C_m = (2^{m-1} - 1) \cdot 2^{m-2}.$$

This proves the recursion (2.1).

Solving the recurrence with the initial conditions $\ell(x_1) = 0$, $\ell(x_2) = 1$, we find

$$\ell(x_m) = 2^{m-2}(2^m - (m+1)),$$

as desired. \square

The Bruhat order in the symmetric group is defined as follows.

Definition 2.9. Let $\pi_1, \pi_2 \in S_n$. We write $\pi_1 \leq \pi_2$ if there exists a sequence of transpositions $t_1, \dots, t_k \in S_n$ such that

- (1) $\pi_1 t_1 \cdots t_k = \pi_2$;
- (2) $\ell(\pi_1 t_1 \cdots t_{s-1}) < \ell(\pi_1 t_1 \cdots t_s)$ for all $1 \leq s \leq k$.

Definition 2.10. Let $w \in S_n$. We define the Ehresmann matrix E_w associated to w as

$$E_w(i, j) = \#\{k \leq j \mid w(k) \geq i\}, \quad 0 \leq i, j < n.$$

That is, $E_w(i, j)$ counts how many values among $0, 1, \dots, j$ are taken by w to values greater than or equal to i . Equivalently, it counts the 1's of the permutation matrix of w lying in the southwest rectangle determined by (i, j) .

The following criterion, originally due to Ehresmann, is a useful tool for comparing two permutations in the Bruhat order. A proof (using slightly different conventions) can be found in [BB05, Theorem 2.1.5].

Lemma 2.11. *Let $x, y \in S_n$. Then, we have*

$$x \leq y \text{ in the Bruhat order} \iff E_x(i, j) \leq E_y(i, j) \text{ for all } i, j.$$

For $x, y \in S_n$ we denote by $[x, y]$ the corresponding Bruhat interval

$$[x, y] = \{z \in S_n \mid x \leq z \leq y\}.$$

The relation between dyadically well-distributed permutations and x_m and y_m is the following:

$$\mathcal{D}_m = [x_m, y_m].$$

For the moment we are in position to prove only one inclusion. The proof of the other inclusion is postponed to the next section.

Lemma 2.12. *For all $m \geq 1$ we have $[x_m, y_m] \subset \mathcal{D}_m$.*

Proof. Let $\pi \in [x_m, y_m]$ and B be a fundamental block. Let $U_\pi(B)$ denote the number of 1's inside the block B in the permutation matrix M_π . By Remark 2.2, in order to prove that $\pi \in \mathcal{D}_m$, it is enough to show that $U_\pi(B) = 1$.

Let V_1, V_2, V_3 and V_4 be the vertices of B labeled in reading order, as is shown in Figure 3. The inclusion-exclusion principle yields

$$U_\pi(B) = E_\pi(V_2) - E_\pi(V_1) - E_\pi(V_4) + E_\pi(V_3). \quad (2.2)$$

By Lemma 2.8 we know that $x_m, y_m \in \mathcal{D}_m$. In particular, via Remark 2.2, we obtain $E_{x_m}(V_i) = E_{y_m}(V_i)$ for all $1 \leq i \leq 4$. On the other hand, Lemma 2.11 yields

$$E_{x_m}(V_i) \leq E_\pi(V_i) \leq E_{y_m}(V_i),$$

for all $1 \leq i \leq 4$. We conclude that $E_{x_m}(V_i) = E_\pi(V_i)$. By combining these equalities with (2.2) we obtain $U_\pi(B) = U_{x_m}(B) = 1$, as we wanted to show. \square

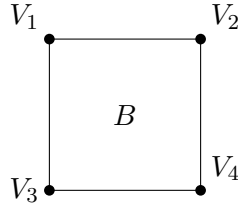


FIGURE 3. Vertices in a fundamental block

2.2. A big Bruhat hypercube. Let us introduce a special type of tiles in a $2^m \times 2^m$ -matrix.

Definition 2.13. We say that a basic k_1 -interval S and a basic k_2 -interval T are *complementary* if $k_1 + k_2 = m + 1$ and $k_1, k_2 \geq 1$. A *complementary block* is a block in a $2^m \times 2^m$ -matrix whose entries are indexed by $S \times T$, where S and T are complementary intervals. We denote by \mathcal{C}_m the set of complementary blocks.

Definition 2.14. Let $S_2 = \{e, s\}$. We define a function

$$\varphi : \mathcal{D}_m \rightarrow S_2^{\mathcal{C}_m}$$

as follows. Let S be a basic k_1 -interval and T a basic k_2 -interval such that $(S, T) \in \mathcal{C}_m$. The submatrix $M_{(S,T)}$ of M_π corresponding to (S, T) is the union of two disjoint fundamental blocks of size $2^{k_1-1} \times 2^{k_2}$. Thus, if $\pi \in \mathcal{D}_m$, there are exactly two entries equal to 1 in $M_{(S,T)}$. After deleting all zero columns and rows in $M_{(S,T)}$, we end up in one of the following cases:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

In the first case, we set $\varphi(\pi)(S, T) = e$, and in the second case we set $\varphi(\pi)(S, T) = s$.

We can see $S_2^{\mathcal{C}_m}$ as partially ordered set as follows.

Definition 2.15. Let $m \geq 1$ and $f_1, f_2 \in S_2^{\mathcal{C}_m}$. We write $f_1 \leq f_2$ if

$$f_1(P, Q) \leq f_2(P, Q)$$

for all $(P, Q) \in \mathcal{C}_m$. In other words, $f_1 \leq f_2$ if and only if for all $(P, Q) \in \mathcal{C}_m$ we have

$$f_1(P, Q) = s \implies f_2(P, Q) = s.$$

We notice that $(S_2^{\mathcal{C}_m}, \leq)$ is isomorphic as a poset to a hypercube of rank $|\mathcal{C}_m| = m2^{m-1}$. The minimal and maximal element in $(S_2^{\mathcal{C}_m}, \leq)$, which we denote by f_e and f_s , satisfy $f_e(P, Q) = e$ and $f_s(P, Q) = s$ for all $(P, Q) \in \mathcal{C}_m$.

Lemma 2.16. *The function φ is order-preserving. This is, for all $\pi_1, \pi_2 \in \mathcal{D}_m$ such that $\pi_1 \leq \pi_2$ in the Bruhat order we have $\varphi(\pi_1) \leq \varphi(\pi_2)$ in $(S_2^{\mathcal{C}_m}, \leq)$.*

Proof. We argue by contraposition. Suppose that $\varphi(\pi_1) \not\leq \varphi(\pi_2)$. Then there exists a complementary pair $(S, T) \in \mathcal{C}_m$ such that

$$\varphi(\pi_1)(S, T) = s \quad \text{and} \quad \varphi(\pi_2)(S, T) = e. \quad (2.3)$$

Divide the complementary pair (S, T) into four congruent rectangles, as illustrated in Figure 4. Let R be the bottom-left rectangle in this subdivision, and let V_1, P, V_2 , and V_3 be the vertices of R listed in reading order. Note that the points V_i are vertices of fundamental blocks. Since $\pi_1, \pi_2 \in \mathcal{D}_m$, we have $E_{\pi_1}(V_i) = E_{\pi_2}(V_i)$ for all $i \in \{1, 2, 3\}$.

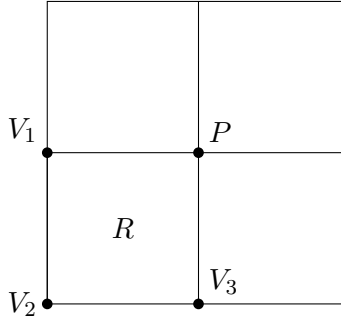


FIGURE 4. Dividing a complementary block as in the proof of Lemma 2.16

Let $X_{\pi_i}(R)$ denote the number of 1's inside R in the permutation matrix M_{π_i} . By (2.3), we have $X_{\pi_1}(R) = 1$ and $X_{\pi_2}(R) = 0$. By inclusion–exclusion, we have

$$E_{\pi_i}(P) = E_{\pi_i}(V_1) + E_{\pi_i}(V_3) - E_{\pi_i}(V_2) + X_{\pi_i}(R).$$

It follows that $E_{\pi_1}(P) - E_{\pi_2}(P) = 1$. In particular, $E_{\pi_2}(P) < E_{\pi_1}(P)$. Therefore, Lemma 2.11 yields $\pi_1 \not\leq \pi_2$, as we wanted to show. \square

Lemma 2.17. *For all $m \geq 1$ we have $\varphi(x_m) = f_e$ and $\varphi(y_m) = f_s$.*

Proof. Let us show the statement for x_m , the one for y_m is analogous. Let (S, T) be a complementary pair. We need to show that $\varphi(x_m)(S, T) = e$. Note that this means that, if $a < b \in S$ are the two elements such that $x_m(a), x_m(b) \in T$, then $x_m(a) < x_m(b)$.

We use the interpretation of x_m as an endomorphism of \mathbb{F}_2^m . Assume S is a basic k -interval, and T is a basic $m - k + 1$ -interval, with $k \geq 1$. Let $P_S \in \mathbb{F}_2^{m-k}$ and $P_T \in \mathbb{F}_2^{k-1}$ be the binary sequences associated to S and T , respectively.

If $k = 1$, then $S = \{P_S 0, P_S 1\}$ and $T = \mathbb{F}_2^m$. Then, $a = P_S 0$ and $b = P_S 1$ (since $a < b$). Thus, $x_m(a) = 0P_S^{-1}$ and $x_m(b) = 1P_S^{-1}$. It follows that $x_m(a) = P_S 0 < P_S 1 = x_m(b)$, since the natural order on the interval $[2^m]$ becomes the lexicographic order on \mathbb{F}_2^m , via our identification $[2^m] \leftrightarrow \mathbb{F}_2^m$.

We now assume $k > 1$. In this case the two elements in S mapping to T are

$$a = P_S 0 P_T^{-1} < P_S 1 P_T^{-1} = b.$$

It is clear that $x_m(a) = P_T 0 P_S^{-1} < P_T 1 P_S^{-1} = x_m(b)$ and this finishes the proof. \square

Definition 2.18. Let $m \geq 1$ and $1 \leq k \leq m$. For any k -bit binary string B we write $B\star$ for a m -bit binary string starting with B (so that \star is a binary string of length $m - k$).

Let $\pi \in \mathcal{D}_m$ and $(S, T) \in \mathcal{C}_m$. Let P_S and P_T be the binary sequences associated to S and T , respectively. Let $P_{S\star_1}, P_{S\star_2} \in \mathbb{F}_2^m$ be the two m -bit binary strings starting with P_S which are sent by π to strings starting with P_T . We define the *flip* $Fl_{S,T}(\pi)$ to be the permutation which agrees with π on all strings other than $P_{S\star_1}$ and $P_{S\star_2}$, and which sends $P_{S\star_1}$ to $\pi(P_{S\star_2})$ and $P_{S\star_2}$ to $\pi(P_{S\star_1})$. In other words, $Fl_{S,T}(\pi)$ is π multiplied on the right by the transposition $(P_{S\star_1}, P_{S\star_2})$.

In the following lemma we identify S_2 with $\mathbb{Z}/2\mathbb{Z}$, sending e to 0 and s to 1. We can thus identify $S_2^{\mathcal{C}_m}$ with the set of functions $f : \mathcal{C}_m \rightarrow \mathbb{Z}/2\mathbb{Z}$.

Lemma 2.19. *Let $\pi \in \mathcal{D}_m$ and $(S, T) \in \mathcal{C}_m$. Let $\chi_{S,T} : \mathcal{C}_m \rightarrow \mathbb{Z}/2\mathbb{Z}$ be the function sending (S, T) to 1 and every other complementary block to 0. Then, we have*

- (1) $Fl_{S,T}(\pi) \in \mathcal{D}_m$.
- (2) $\varphi(Fl_{S,T}(\pi)) = \varphi(\pi) + \chi_{S,T}$.

(3)

$$\ell(Fl_{S,T}(\pi)) = \begin{cases} \ell(\pi) + 1, & \text{if } \varphi(\pi)(S,T) = 0; \\ \ell(\pi) - 1, & \text{if } \varphi(\pi)(S,T) = 1. \end{cases} \quad (2.4)$$

Proof. Let $\sigma := Fl_{S,T}(\pi) \in \mathcal{D}_m$. By definition of the flip, the permutation matrices M_π and M_σ are identical outside the complementary pair (S,T) . Within that block, they differ by a local move

$$\begin{pmatrix} 1 & \cdots & 0 \\ \vdots & 0 & \vdots \\ 0 & \cdots & 1 \end{pmatrix} \longleftrightarrow \begin{pmatrix} 0 & \cdots & 1 \\ \vdots & 0 & \vdots \\ 1 & \cdots & 0 \end{pmatrix}. \quad (2.5)$$

We prove each item separately.

- (1) Following our established notation, we write $\pi(P_{S\star_1}) = P_{T\star_1}$ and $\pi(P_{S\star_2}) = P_{T\star_2}$ for the two 1 entries in the permutation matrix of π which are moved by the flip.

Suppose that $\sigma \notin \mathcal{D}_m$. Then there is a fundamental block (U,V) such that the equation $\sigma(P_U\star) = P_V\star'$ has no solution. Since $\pi \in \mathcal{D}_m$, the equation $\pi(P_U\star) = P_V\star'$ does have a solution, and indeed a unique solution. Since σ differs from π only on the inputs $P_{S\star_1}$ and $P_{S\star_2}$, this solution must be either $\pi(P_{S\star_1}) = P_{T\star_1}$ or $\pi(P_{S\star_2}) = P_{T\star_2}$. Without loss of generality, we assume it is the former. Then we know $P_{S\star_1}$ starts with P_U and $P_{T\star_1}$ starts with P_V .

By the definition of complementary pair and fundamental block, we know that $|P_S| + |P_T| = m - 1$ and $|P_U| + |P_V| = m$. This implies that either $|P_U| \leq |P_S|$ or $|P_V| \leq |P_T|$. In the former case, the fact that $P_{S\star_1}$ starts with P_U now implies that P_U is a prefix of P_S . In particular, $P_{S\star_2}$ starts with P_U . By the definition of the flip,

$$\sigma(P_{S\star_2}) = P_{T\star_1}$$

and since $P_{T\star_1}$ starts with P_V , this provides a solution to $\sigma(P_U\star) = P_V\star'$, contradicting our hypothesis. If on the other hand $|P_V| \leq |P_T|$, then P_V is a prefix of P_T and the equation $\sigma(P_{S\star_1}) = P_{T\star_2}$ provides a contradiction in exactly parallel fashion. We conclude that $\sigma = Fl_{S,T}(\pi)$ lies in \mathcal{D}_m , as claimed.

- (2) We now compute $\varphi(\sigma)$. We first note that

$$\varphi(\sigma)(S,T) = \varphi(\pi)(S,T) + 1,$$

because the flip switches the positions of the two 1's in the block (S,T) .

It remains to show is that the flip does not change the value of $\varphi(\pi)(S_1, T_1)$ for any other complementary pair $(S_1, T_1) \neq (S, T)$. The key point is that, by the definition of φ , the value of $\varphi(\sigma)(S_1, T_1)$ is determined by the knowledge of *either one* of the two 1's in the complementary pair (S_1, T_1) . For instance, if one of these 1's is located in the upper left quarter of (S_1, T_1) then the other 1 in (S_1, T_1) is forced to be located in the lower right quarter, and $\varphi(\sigma)(S_1, T_1) = 0$.

Now of the two 1's lying in (S_1, T_1) , at most one of them can lie in (S, T) , since $(S, T) \cap (S_1, T_1)$ is contained in a fundamental block and can thus contain at most one 1. This means that there is some 1 such that it lies in (S_1, T_1) but does *not* lie in (S, T) . Since M_π and M_σ coincide outside (S, T) we conclude that this 1 has the same location in both M_π and M_σ . Since the position of this common 1 determines the value of $\varphi(\sigma)(S_1, T_1)$ and $\varphi(\pi)(S_1, T_1)$, we conclude that the two values are equal, as claimed.

- (3) Finally, we compute $\ell(\sigma)$, this is, the number of inversions of σ . Since M_π and M_σ differ only by a local move as in (2.5), it is clear that $\ell(\sigma) = \ell(\pi) \pm 1$.

Let us be more precise. Let $i, j \in [2^m]$ be the columns where the two 1's are located. Suppose that in (2.5) the left (resp. right) matrix corresponds to a submatrix of M_π (resp. M_σ). In this case we have $\varphi(\pi)(S, T) = 0$. On the other hand, the pair (i, j) is an inversion of σ but not an inversion of π . Therefore, $\ell(\sigma) = \ell(\pi) + 1$. This proves the first case of (2.4). The other case is analogous.

Having proved the three claims the proof is complete. \square

Proposition 2.20. *The map $\varphi : \mathcal{D}_m \rightarrow S_2^{\mathcal{C}_m}$ is a poset isomorphism, i.e., $\pi_1 \leq \pi_2$ if and only if $\varphi(\pi_1) \leq \varphi(\pi_2)$.*

Proof. We split the proof in three steps.

(1) φ is injective.

Let $\pi \in \mathcal{D}_m$ and $a \in [2^m]$. We show that $\pi(a)$ can be recovered uniquely from $\varphi(\pi)$.

Let C_1 denote the column indexed by a . Consider the unique complementary $(m, 1)$ -block (S_1, T_1) containing C_1 . The value $\varphi(\pi)(S_1, T_1)$ specifies which of the two sub-columns of height 2^{m-1} cannot contain $\pi(a)$. Thus, this step *blocks* exactly 2^{m-1} possible rows in C_1 . Let C_2 be the remaining (unblocked) sub-column, which has height 2^{m-1} .

Next, consider the unique complementary $(m-1, 2)$ -block (S_2, T_2) containing C_2 . The value $\varphi(\pi)(S_2, T_2)$ again blocks half of C_2 , that is, a sub-column of 2^{m-2} rows. Let C_3 be the remaining unblocked sub-column, now of height 2^{m-2} .

Continuing in this way, at the k -th step we block 2^{m-k} rows. After m steps, the total number of blocked rows is

$$2^{m-1} + 2^{m-2} + \dots + 2^1 + 2^0 = 2^m - 1.$$

Since the entire column C_1 contains 2^m rows, blocking $2^m - 1$ of them leaves exactly one row unblocked, which must be the position of $\pi(a)$.

Therefore, $\pi(a)$ is uniquely determined by $\varphi(\pi)$ for every $a \in [2^m]$, and hence φ is injective.

(2) φ is surjective.

We use the identification $S_2 \leftrightarrow \mathbb{Z}/2\mathbb{Z}$ as in the proof of the previous lemma.

Let $f : \mathcal{C}_m \rightarrow \mathbb{Z}/2\mathbb{Z}$ and

$$\mathcal{A}_f = \{(S, T) \in \mathcal{C}_m \mid f(S, T) = 1\}.$$

We prove that any f has a pre-image by induction on $|\mathcal{A}_f|$.

If $|\mathcal{A}_f| = 0$, then $f = f_e = f_0$. Thus Lemma 2.17 gives $\varphi(x_m) = f$. This establishes the base case of the induction.

Now fix f such that $|\mathcal{A}_f| > 0$. Let $(S, T) \in \mathcal{A}_f$ and set $\mathcal{A}' = \mathcal{A}_f \setminus \{(S, T)\}$. Define $g : \mathcal{C}_m \rightarrow \mathbb{Z}/2\mathbb{Z}$ to be the function that takes the value 1 on \mathcal{A}' and 0 elsewhere. By the induction hypothesis, there exists $\pi_g \in \mathcal{D}_m$ such that $\varphi(\pi_g) = g$. It follows from Lemma 2.19 that

$$\varphi(Fl_{(S, T)}(\pi_g)) = f.$$

This completes the induction.

(3) φ is an isomorphism of posets.

Having already proved that φ is bijective, it remains to show that φ preserves the order in both directions. One direction is established in Lemma 2.16. Thus, it suffices to prove that

$$\varphi(\pi) \leq \varphi(\pi') \implies \pi \leq \pi', \quad \text{for } \pi, \pi' \in \mathcal{D}_m.$$

The key observation is that $\varphi(\pi) \leq \varphi(\pi')$ implies $\mathcal{A}_{\varphi(\pi)} \subset \mathcal{A}_{\varphi(\pi')}$. A straightforward inductive argument on $|\mathcal{A}_{\varphi(\pi')} \setminus \mathcal{A}_{\varphi(\pi)}|$, together with Items 2 and 3 in Lemma 2.19, shows that π' can be obtained from π through a sequence of length-increasing right multiplications by transpositions. Therefore, $\pi \leq \pi'$, as desired. \square

Theorem 2.21. *For all $m \geq 1$ we have:*

- (1) $\mathcal{D}_m = [x_m, y_m]$.
- (2) $[x_m, y_m] \simeq S_2^{\mathcal{C}_m}$.
- (3) *The Bruhat interval $[x_m, y_m]$ is a hypercube of rank $m2^{m-1}$.*

Proof. By Lemma 2.12 to show the first claim it is enough to prove the inclusion $\mathcal{D}_m \subset [x_m, y_m]$. Let $\pi \in \mathcal{D}_m$. We have $f_e \leq \varphi(\pi) \leq f_s$. By Lemma 2.17 and Proposition 2.20 we obtain $x_m \leq \pi \leq y_m$. This is $\pi \in [x_m, y_m]$. This finishes the proof of the first claim. The second claim follows by combining Proposition 2.20 and the first claim. Finally, the third claim is a direct consequence of the second once we recall that $|\mathcal{C}_m| = m2^{m-1}$. \square

3. CONNECTIONS, EXTENSIONS AND APPLICATIONS

In this section we point out some connections with the existing literature, explain a p -adic generalization, and give some applications to cluster algebras and moduli of embeddings of Bruhat graphs.

3.1. Low-discrepancy sequences. The sequence

$$(0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15)$$

appearing in Example 2.5 is known (after dividing by 16 and removing the initial 0) as the *van der Corput sequence* with denominator 16 in $[0, 1]$. It is a well-known example of a low-discrepancy or quasirandom sequence in the unit interval.

The dyadically well-distributed permutations also have an incarnation in the world of low-discrepancy sequences. If π is a permutation in \mathcal{D}_m , we may think of its permutation matrix as a subset of 2^m points $(i/2^m, \pi(i)/2^m)$ in the unit square. A size- 2^m subset of the square, like this one, such that each fundamental block contains exactly one of the points is called a $(0, m, 2)$ -net. This is a special case of the more general notion of (t, m, s) -net defined by Niederreiter in [Nie87]. These nets are especially well-distributed finite subsets of $[0, 1]^s$ which have proven to be widely useful for quasirandom number generation and Monte Carlo integration, for instance in mathematical finance ([BBG97].)

It is not hard to show that the set of $(0, m, 2)$ -nets (up to the obvious notion of equivalence, where a net is labeled by the set of $(1/2^m) \times (1/2^m)$ squares it intersects) is in bijection with \mathcal{D}_m . In this context, the fact that there are exactly $2^{m2^{m-1}}$ inequivalent $(0, m, 2)$ -nets (part of Proposition 2.20) is proved in [Xia02].

As in Remark 2.6, we may think of the affine linear group $\mathrm{GL}(\mathbb{F}_2^m) \times \mathbb{F}_2^m$ as a subgroup of S_{2^m} . Write B for the subgroup of upper triangular matrices in $\mathrm{GL}(\mathbb{F}_2^m)$ and G for $B \times \mathbb{F}_2^m$. Then G permutes the basic intervals in $[2^m]$, which means that \mathcal{D}_m is preserved under both left and right multiplication by G . In particular, this means that the double coset Bx_mB is contained in \mathcal{D}_m . Note that x_m , considered as an element of $\mathrm{GL}(\mathbb{F}_2^m)$, is a permutation matrix, corresponding to the reversal in S_m ; so Bx_mB is the large cell in the Bruhat decomposition of $\mathrm{GL}(\mathbb{F}_2^m)$. This very natural construction was also known in the low-discrepancy sequence setting: the $(0, m, 2)$ -nets corresponding to Bx_mB are called the *digital* $(0, m, 2)$ -nets. Indeed, one finds in that literature the condition that a matrix has all principal minors nonsingular (see for instance Corollary 4.54 of [DP10]) but the fact that this condition picks out the large Bruhat cell does not seem to have been noticed, just as

the fact that the $(0, m, 2)$ -nets form a Bruhat interval (as we show in Theorem 2.21) has not been noticed. It would be interesting to mine the literature on low-discrepancy sequences to see if there are yet more constructions there which have unexpected connections to algebraic combinatorics.

3.2. Powers of permutahedra. The arguments above generalize as follows. For each $t, m > 0$, there is a set of t -adically well-distributed permutations $\mathcal{D}_m^{(t)} \subseteq S_{t^m}$, defined by imitating Definition 2.1. Note, in particular, that $\mathcal{D}_1^{(t)} = S_t$. Further examples of these permutations are given by $x_m^{(t)}, y_m^{(t)} \in S_{t^m}$, which are the affine transformations of $(\mathbb{Z}/t\mathbb{Z})^m$ given by

$$x_m^{(t)}(a_0, \dots, a_{m-1}) = (a_{m-1}, \dots, a_0); \quad y_m^{(t)}(a_0, \dots, a_{m-1}) = (t-1-a_{m-1}, \dots, t-1-a_0).$$

Note that $x_m^{(t)}$ and $y_m^{(t)}$ are involutions, and $x_m^{(t)}y_m^{(t)} = y_m^{(t)}x_m^{(t)}$ is the longest element in S_{t^m} . Definition 2.14 generalizes to give a map $\varphi : \mathcal{D}_m^{(t)} \rightarrow S_t^{mt^{m-1}}$. Using this one can show that

$$[x_m^{(t)}, y_m^{(t)}] = \mathcal{D}_m^{(t)} \cong S_t^{mt^{m-1}},$$

where the last isomorphism is that of posets with the Bruhat order.

For example, to construct a 3-adically well-distributed permutation in S_9 we start with the following configuration:

1	1	1	τ_1
1	1	1	τ_2
1	1	1	τ_3
σ_1	σ_2	σ_3	

where the large 1's indicate that there is a unique entry equal to 1 in the corresponding 3×3 -block, and $\sigma_i, \tau_i \in S_3$ are permutations that indicate the relative positions of the 1's in the corresponding column and row, respectively. This gives us the bijection between the 3-adically well-distributed permutations in S_9 and S_3^6 . To give a more concrete example:

	1		1	$[2, 3, 1]$
1		1		$[2, 1, 3]$
		1	1	$[3, 2, 1]$
$[2, 1, 3]$	$[1, 3, 2]$	$[1, 2, 3]$		

Changing the last column from the identity $[1, 2, 3]$ to $[2, 1, 3]$ results in the following matrix.

		1			
1					
	1				
		1			
1				1	
			1		
					1
	1				

[2, 3, 1]
[2, 1, 3]
[3, 2, 1]

[2, 1, 3]
[1, 3, 2]
[2, 1, 3]

and we see that we have a covering relation in the Bruhat order.

Remark 3.1. Note that a usual Sudoku puzzle is a collection $(\sigma_1, \dots, \sigma_9)$ of 3-adically well-distributed permutations in S_9 , satisfying the condition that $\sigma_i \sigma_j^{-1}$ does not have fixed points if $i \neq j$. From this viewpoint, we can define generalized Sudoku puzzles as a collection of t^m t -adically well-distributed permutations $(\sigma_1, \dots, \sigma_{t^m})$ in S_{t^m} such that $\sigma_i \sigma_j^{-1}$ does not have fixed points if $i \neq j$. For example, the following is a 2-adic Sudoku puzzle of elements in S_8 .

1	4	5	6	2	3	8	7
2	3	8	7	5	6	1	4
5	6	1	4	8	7	2	3
8	7	3	2	1	4	5	6
4	1	6	5	3	2	7	8
3	2	7	8	6	5	4	1
6	5	4	1	7	8	3	2
7	8	2	3	4	1	6	5

3.3. Kazhdan-Lusztig d -invariant. We define the d -Kazhdan-Lusztig invariant mentioned in the introduction. It arises as the absolute value of the coefficient of the second largest power in a Kazhdan-Lusztig R -polynomial, which we recall its definition now for permutations although they can be defined for arbitrary Coxeter systems.

Definition 3.2. There is a unique family of polynomials $\{R_{u,v}(q)\}_{u,v \in S_n} \subseteq \mathbb{Z}[q]$ satisfying the following conditions:

- (1) $R_{u,v}(q) = 0$ if $u \not\leq v$;
- (2) $R_{u,v}(q) = 1$ if $u = v$;
- (3) For $1 \leq i < n$, let $s_i = (i, i + 1)$.
If $\ell(vs_i) < \ell(v)$ then

$$R_{u,v}(q) = \begin{cases} R_{us_i, vs_i}(q), & \text{if } vs_i < u, \\ qR_{us_i, vs_i}(q) + (q-1)R_{u, vs_i}(q), & \text{if } vs_i > v. \end{cases}$$

We can now define the d -invariant.

Definition 3.3. For $x, y \in S_n$ we introduce $d_{x,y} \in \mathbb{Z}$ by

$$R_{x,y}(q) = q^{\ell(y)-\ell(x)} - d_{x,y} q^{\ell(y)-\ell(x)-1} + \text{lower degree terms.}$$

We have the following recursion to compute $d_{x,y}$.

- (1) $d_{x,x} = 0$ for any $x \in W$;
- (2) for any $x \leq y$ and any simple transposition s_i such that $ys_i < y$ we have

$$d_{x,y} = \begin{cases} d_{xs_i,ys_i}, & \text{if } xs_i < x, \\ d_{x,ys_i} + 1, & \text{if } xs_i > x \text{ and } xs_i \not\leq ys_i, \\ d_{x,ys_i}, & \text{if } xs_i > x \text{ and } xs_i \leq ys_i. \end{cases} \quad (3.1)$$

Corollary 3.4. *Let $m \geq 1$. Let $x, y \in [x_m, y_m]$ be such that $x \leq y$. Then, we have*

$$R_{x,y}(q) = (q-1)^{\ell(y)-\ell(x)}. \quad (3.2)$$

In particular, $R_{x_m,y_m}(q) = (q-1)^{m2^{m-1}}$ and $d_{x_m,y_m} = m2^{m-1}$.

Proof. Theorem 2.21 implies that the Bruhat interval $[x, y]$ is a hypercube. In particular, $[x, y]$ does not contain any interval isomorphic to S_3 . Therefore, (3.2) follows by a direct application of [Bre94, Theorem 6.3]. Finally, by Lemma 2.8 we have $\ell(y_m) - \ell(x_m) = m2^{m-1}$, and the last claim follows. \square

We use the name d -invariant because of the work of Patimo [Pat21], who showed that this value is a combinatorial invariant of Bruhat intervals. Namely, it depends only on the unlabeled Hasse diagram of the interval and not on its endpoints. Building on this fact, he proved the Combinatorial Invariance Conjecture of Lusztig and Dyer for the q -coefficient of a Kazhdan–Lusztig polynomial.

The main ingredient in his proof is a combinatorial formula for computing $d_{x,y}$, namely

$$d_{x,y} = \min\{|F| \mid F \subset E_{[x,y]} \text{ and } F^\diamond = E_{[x,y]}\}, \quad (3.3)$$

where $E_{[x,y]}$ denotes the set of edges of the Hasse diagram of $[x, y]$, and F^\diamond denotes the diamond-closure of F . For the precise definition of F^\diamond we refer to [Pat21, Definition 4.5]. Roughly speaking, $F^\diamond \supset F$ is obtained as follows: whenever two adjacent edges of a diamond (i.e., a rank-two hypercube) already lie in F^\diamond , we add to F^\diamond the remaining two adjacent edges of that diamond. We repeat this process until no further edges can be added.

In general, computing $d_{x,y}$ using (3.3) is a difficult task. However, when $[x, y]$ is a hypercube, it is straightforward to verify that a minimal set F satisfying the condition in (3.3) is given by the set of edges incident to x . In this way we recover the formula $d_{x_m,y_m} = m2^{m-1}$ directly from the combinatorial definition of the d -invariant.

3.4. Cluster varieties. We say that an affine algebraic variety X is a cluster variety if its coordinate algebra $\mathbb{C}[X]$ admits the structure of a cluster algebra, cf. [FZ02]. This means that X can be covered, up to codimension 2, with algebraic tori, and each of these tori admits a coordinate system consisting of regular functions on $\mathbb{C}[X]$ (the *cluster variables*) such that the transition functions between the tori can be codified into a combinatorial rule known as *cluster mutation*. The cluster variables can be separated into *mutable* and *frozen*. The frozen variables are precisely those that are units in $\mathbb{C}[X]$, and appear as coordinates in every algebraic torus of the cluster structure on X . While an algebraic variety may admit many non-equivalent cluster structures, the number of frozen variables is an invariant of X , cf. [GLS13]. The existence of a cluster structure has many consequences for the geometry of X , for example, it implies the existence of a \mathbb{Z} -form with positive structure constants, [Gro+18].

Recently, it has been shown that open Richardson varieties in the flag variety admit cluster structures in [Cas+25a] and, independently, in [GLS]. By the main result of [Cas+25b], the cluster structures constructed in both works coincide. The work [GLS] uses Deodhar geometry [Deo85] to construct the cluster structure. In particular, the number of *mutable* variables in a seed coincides with the number of codimension-1 Deodhar strata in the Richardson variety, and the total number of cluster variables is $\dim R_{x,y}^\circ = \ell(y) - \ell(x)$.

Denoting the number of frozen variables by $f_{x,y}$ we have

$$f_{x,y} = \ell(y) - \ell(x) - \#\mathfrak{D}_{x,y}^1 \quad (3.4)$$

where $\mathfrak{D}_{x,y}^1$ is the set of codimension-1 Deodhar strata.

On the other hand, Deodhar in [Deo85, Theorem 1.3] gives an expression for the R -polynomial in terms of the Deodhar stratification:

$$R_{x,y}(q) = \sum_{\mathfrak{s}} (q-1)^{a(\mathfrak{s})} q^{b(\mathfrak{s})}.$$

where \mathfrak{s} runs over all Deodhar strata, and $a(\mathfrak{s}) + 2b(\mathfrak{s}) = \ell(y) - \ell(x)$. The quantity $b(\mathfrak{s})$ is precisely the codimension of the stratum \mathfrak{s} . There is a unique stratum \mathfrak{s}_0 with $b(\mathfrak{s}_0) = 0$. It follows that we can write

$$R_{x,y}(q) = (q-1)^{\ell(y)-\ell(x)} + \#\mathfrak{D}_{x,y}^1 (q-1)^{\ell(y)-\ell(x)-2} q + \text{lower terms}.$$

We obtain

$$R_{x,y}(q) = q^{\ell(y)-\ell(x)} - (\ell(y) - \ell(x) - \#\mathfrak{D}_{x,y}^1) q^{\ell(y)-\ell(x)-1} + \text{lower terms}. \quad (3.5)$$

Comparing (3.4) and (3.5) we conclude that $f_{x,y} = d_{x,y}$, i.e., the number of frozen variables in the Richardson variety $R_{x,y}^\circ$ is precisely the d -invariant $d_{x,y}$.

Any cluster variety admits a faithful action of an algebraic abelian group, and if the cluster variety has *really full rank*, this abelian group is a torus of rank equal to the number of frozen variables, see [LS22, Section 5] for details. In particular, open Richardson varieties have really full rank, so we have a faithful action

$$(\mathbb{C}^\times)^{d_{x,y}} \curvearrowright R_{x,y}^\circ. \quad (3.6)$$

One problem with the cluster-theoretic construction of this action is that it is non-explicit and its construction depends on intricate combinatorics (see e.g. [Kim25, Example 4.6]), so it would be desirable to have a more explicit form of it. Identifying $(\mathbb{C}^\times)^{n-1}$ with the torus of diagonal matrices in $\mathrm{PGL}(n)$, it is easy to obtain a natural action of $(\mathbb{C}^\times)^{n-1}$ on $R_{x,y}^\circ$. In many cases (for example, when $x = e$ ([Cas+24, Section 4]) or when y is a Grassmannian permutation ([GL24])), we have $d_{x,y} \leq n-1$ and the action (3.6) is a quotient of the natural $(\mathbb{C}^\times)^{n-1}$ -action. When $d_{x,y} > n-1$, it is not clear how to obtain the action of the remaining \mathbb{C}^\times -factors in (3.6). In particular, it would be very interesting to obtain an explicit action of $(\mathbb{C}^\times)^{\ell(y_n)-\ell(x_n)}$ on R_{x_n,y_n}° without using that R_{x_n,y_n}° is already isomorphic to a torus. Moreover, it would be interesting to know if the $(\mathbb{C}^\times)^{\ell(y_n)-\ell(x_n)}$ -action on the *open* Richardson variety R_{x_n,y_n}° extends to the *closed* Richardson variety \bar{R}_{x_n,y_n} . We remark that Theorem 2.21 implies, using [AB16, Corollary 3.2] and [KWY13, Corollary 1.3], that \bar{R}_{x_n,y_n} is smooth.

3.5. Moduli of Bruhat interval embeddings. The Bruhat graph of an interval $[x, y]$ has vertices consisting of the permutations in the interval and a directed edge $u \rightarrow v$ if $u \leq v$ and $u = tv$ for some reflection $t = (i, j) \in S_n$, in which case the edge is labelled by t . One can recover the unlabelled Bruhat graph of an interval from the poset [Dye91].

Bruhat graphs have natural embeddings. The symmetric group S_n acts on \mathbb{R}^n via permutation of coordinates. Let us fix $\rho = (0, 1, \dots, n-1)$ and send $u \in [x, y]$ to $u(\rho)$ and $u \rightarrow v$ to the straight line connecting $u(\rho)$ to $v(\rho)$. This embedding has a geometric meaning: it is the image of the 0- and 1-dimensional torus orbits in the closed Richardson variety under the moment map [BM01]. We will refer to this embedding as the *geometric* embedding below, and stress that it is extra information.

For several questions in combinatorics and Kazhdan-Lusztig theory, it is more natural to consider the Bruhat graph. One such example, is the combinatorial invariance conjecture

of Dyer and Lusztig, which asserts that isomorphic intervals give rise to the same Kazhdan-Lusztig polynomial. It is known [Dye91; BM01] that one can recover the Kazhdan-Lusztig polynomial from the Bruhat graph *together* with its embedding. A crucial difficulty is that although two intervals may be isomorphic, their geometric embeddings might be very different. For example, the natural ambient spaces in which the embeddings live might not even be of the same dimension. This motivated Hone, Klein and the last author to consider a general class of embeddings of Bruhat intervals, of which the geometric embeddings discussed above are an example [HKW25].

In order to explain this definition, we need to briefly recall reflection subgroups. In our setting of the symmetric group, these are subgroups generated by any subset of transpositions. Any such subgroup is isomorphic to a product of (possibly smaller) symmetric groups. We say that a reflection subgroup is of rank 2 if it is isomorphic to $S_2 \times S_2$ or S_3 . One can detect the intersections of cosets of rank 2 reflection subgroups with any Bruhat interval. These are precisely the diamond complete subgraphs isomorphic to an arrow, diamond or a full S_3 subgraph (see Figure 5). We will refer to these as *rank 2 subgraphs*.

The following definition (central to [HKW25]) attempts to abstract the properties of the embeddings of Bruhat graphs that occur in nature:

Definition 3.5. Let $x, y \in S_n$. An embedding of the Bruhat graph $\phi : [x, y] \rightarrow \mathbb{R}^m$ is *good* if:

- (1) The images of the vertices lie on a sphere $S^{m-1} \subset \mathbb{R}^m$;
- (2) For all rank 2 subgraphs, the images of its points lie on a plane.

It is easy to see that the geometric embeddings considered above satisfy these properties. However, even simple Bruhat intervals may have many more good embeddings than occur in nature. For example, only a few geometric embeddings of diamonds occur in nature, whereas the moduli space of good embeddings inside \mathbb{R}^2 is an open set inside the moduli space of 4 points on a circle $S^1 \subset \mathbb{R}^2$. More generally, in the symmetric group the angles between edges in geometric embeddings always lie in the set $\pm\pi/3, \pm\pi/2$ and $\pm 2\pi/3$, whereas no such restrictions are made for good embeddings. Good embeddings appear to provide a natural larger space in which one could attempt to relate the geometric embeddings of isomorphic intervals.

For a fixed interval $[x, y]$, it is natural to ask: what is the largest possible dimension in which this interval admits a good embedding? (We always assume embeddings are such that their vertices are not contained in an affine hyperplane.) This quantity shares at least two properties with the number of frozen variables discussed earlier: one might (falsely) guess that it is always bounded by $n - 1$; and, it is trivially bounded by $\ell(y) - \ell(x)$. It is plausible that these two numbers are in fact equal, however this is not known. The setting simplifies dramatically for hypercubes. In this case this largest possible dimension is easily seen to be the dimension of the hypercube, and the moduli of all embeddings can be understood explicitly.

4. EXPERIMENTS WITH FUNSEARCH AND ALPHAEVOLVE

The main theorems of the present paper were suggested to us by interesting examples that were discovered computationally. Since the computational methods employed here (genetic algorithms using large language models as a mode of reproduction) are novel, we will devote slightly more time than is customary to describing the course of experimentation in a fair amount of narrative detail, in the hope that this discussion will be useful to other mathematicians interested in employing these methods.

The initial problem we considered was that of finding pairs of permutations x, y in S_n whose d -invariant $d_{x,y}$ is as large as possible.

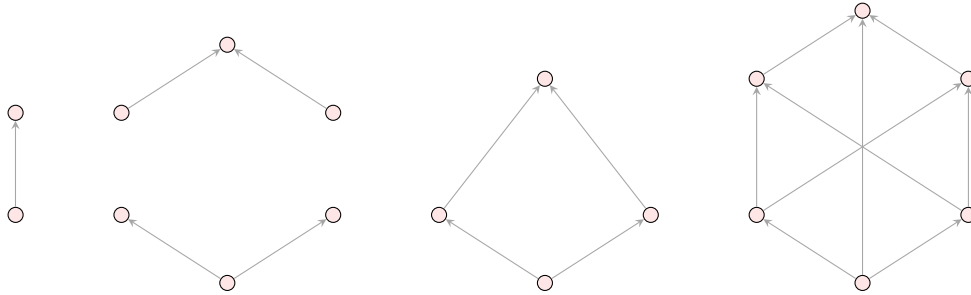


FIGURE 5. Rank two subgraphs of Bruhat intervals for symmetric groups: the arrow, incomplete diamonds, diamond and full S_3 .

This problem has two features which, together, have tended to indicate promising contexts for exploration using machine learning methods:

- The search space (pairs of permutations) has size $(n!)^2$, too large to search exhaustively for any but the very smallest n .
- The function $d_{x,y}$ to be maximized can be computed quickly and reliably using (3.1).

Other problems with these features that have been subjects for machine learning experiments of this kind include:

- (1) **The cap set problem.**
Finding large subsets of $(\mathbb{Z}/3\mathbb{Z})^n$ with no three-term arithmetic progressions.
- (2) **No-Isosceles Problem.**
Finding large subsets of the $n \times n$ grid with no isosceles triangles.
- (3) **The extremal number of C_4 .**
Finding graphs on n vertices with as many edges as possible and no 4-cycles.

Note that the search spaces for these problems have size exponential in n^2 or even doubly exponential. The search space treated here is, in some sense, relatively small among those that are too big. This may help explain the relatively successful performance observed in the experiments reported here.

Let $f : \mathbb{N} \rightarrow \mathbb{N}$ be the function given by

$$f(n) = \max\{d_{x,y} \mid x, y \in S_n\}.$$

It is quite easy to show that $f(n)$ is at least within a constant of n . For instance, we have $d_{e,w_0} = n - 1$. Exhaustive computation for $n \leq 7$ yields

n	1	2	3	4	5	6	7
$f(n)$	0	1	2	4	5	7	9

TABLE 2. Small values of $f(n)$

We found a pair of permutations in S_8 with d -invariant 12, but were not sure whether this value was maximal. Indeed, for each n we were able to construct a pair of permutations (x, y) such that $d_{x,y} = 2n - 4$ if n is even and $2n - 5$ if n is odd. These are formed by a construction that is easy to describe: for instance, when $n = 12$, a pair of permutations with $d_{x,y} = 20$ is given by

$$x = (2, 5)(4, 7)(6, 9)(8, 11) \quad \text{and} \quad y = (1, 12)(2, 4)(3, 6)(5, 8)(7, 10)(9, 11).$$

Our goal as we began our experiments was to understand whether this lower bound was best possible. As we record in 3.4, the answer turned out to be no.

4.1. FunSearch experiments. FunSearch is a protocol developed by Google DeepMind in 2023 and described in [Rom+24]. FunSearch is open-source and various versions are available. We used the implementation developed by Christoforo Fraser-Taliente, which we have found particularly mathematician-friendly, and which is publicly available at <https://github.com/kitft/funsearch>.

We recall the main idea. Our goal is to find two permutations x, y such that $d_{x,y}$ is large. FunSearch, rather than searching for such a pair directly, searches for a Python program whose output is such a pair.

The main loop in the mechanism looks something like this. Here, M and m are constants which matter only insofar as $M \gg m$.

- (1) Suppose given a large number M of Python programs.
- (2) Run them all. Discard all those whose output is not a pair of permutations. Among those that remain, choose the m programs whose output has the largest value of $d_{x,y}$.
- (3) Send various subsets of these m as input to a large language model, together with a prompt asking the model to output M programs which resemble the ones in the input.
- (4) Return to step 1.

At the initial step, the M programs can be taken to be M copies of an initial program chosen by the user. In these experiments, we typically initialized with the program that takes an integer n as input and outputs two copies of the identity permutation in S_n .

The permutation-producing programs that evolve as FunSearch runs are, for historical reasons, called **priority**. Importantly, the evaluation function (in this case, the function that computes $d_{x,y}$ given input permutations x and y) is frozen in place; we supply it to FunSearch at the beginning and it does not evolve. This gives the user quite a bit of power and flexibility in guiding the evolution of **priority** towards whatever objective is desired.

There are many ways to express a permutation in python. For these experiments, we chose to render a permutation as a list of integers between 1 and $n - 1$ inclusive; this list is interpreted by the evaluator as a product of simple reflections s_i . In particular, the identity is expressed as the empty list, so our initial program simply outputs two copies of the empty list.

One might ask how we constrain **priority** to give output of this form as it evolves under FunSearch. The answer is, we can't! Not directly, at least. What the user controls directly is the frozen evaluator, which we can design to give a low score to any **priority** whose output isn't in the desired form. In practice (in this experiment and many others) we have found that this practice is sufficient to keep most of the programs in the population producing well-formed output.

Our experiments used the prompt

```
The program is trying to learn a pair of permutations with large d-invariant.
```

```
On every iteration, improve priority_v1 over the priority_vX methods from previous
↪ iterations.
```

```
Make only small changes.
```

```
Try to make the code short.
```

```
Please do *not* use any randomness.
```

The request not to use randomness is because when a program with non-deterministic output achieves a high score on a test run, we may not be able to reproduce that performance on later runs. It's simpler to keep track of what's going on if a program that pleases the evaluator is actually deterministically good.

Our runs primarily used Mistral’s model `mistral-small-3.1-24b-instruct`, which was chosen for cheapness. In [Ell+25] we found that, in FunSearch applications, there was no consistent performance benefit to using more expensive models. In the present project we occasionally ran FunSearch with input from larger models, but anecdotally observed no improvement in performance. The total cost of the FunSearch experiments reported here was roughly 10USD in API calls to Mistral. By contrast, [Geo+25] reports that AlphaEvolve did exhibit a benefit when supplied with calls to a more expensive LLM.

In general, programs trained by FunSearch for a fixed instance or a fixed set of instances of a problem tend not generalize well to further instances. For instance, a program trained to study the No-Isosceles Problem for a 9×9 grid does not perform very well when run on a larger grid [Ell+25]. This problem is quite different. FunSearch experiments rather quickly (within a half hour) arrived at programs which gave $d_{x,y} = 2n - c$ for many n , even when trained on only one n . For example, a run for $n = 11$ yielded the following program, which produces a d -invariant of 17.

```
def priority(n: int) -> list:
    """Returns a pair of lists of integers between 1 and n-1, to be interpreted as a
    ↪ sequence of adjacent transpositions.
    n is an int. We do not constrain the length of the lists.
    """

    a = list(range(1, n-1, 2))[::-1] + list(range(2, n-1, 2))[::-1]
    b = list(range(0, n-2, 2)) + list(range(n//3, n-1, 2))[::-1]
    c = list(range(1, n-1, 3))[::-1] + list(range(2, n-1, 3))[::-1] +
    ↪ list(range(3, n-1, 3))[::-1]
    return [a, b, c]
```

One amusing aspect is that this program actually produces a list of *three* permutations, rather than two. It turns out that the evaluator as we wrote it simply discards any extra permutations passed to it as input. It is quite common in both FunSearch and AlphaEvolve that LLM-generated programs contain code irrelevant to the task.

The program above produces a pair of permutations with d -invariant $11m + 6$ for every n of the form $6m + 5$ that we tested, and we imagine it likely does so in general. This is already unusually good performance; one rarely sees programs that generalize to infinitely many problem instances.

On the other hand, when $n = 13$ the program gives a pair of permutations with d -invariant 13, a fairly poor performance. But a FunSearch run *initialized* at the above program very rapidly found a program which yielded d -invariant 21; the only change is that the definition of the second permutation changes to

```
b = list(range(0, n-1, 2)) + list(range(n//4, n-1, 2))[::-1]
```

This suggested manually checking which value of the beginning of the second range ($n // 3$ in the first program trained on $n = 11$, $n // 4$ in the second trained on $n = 13$) gives optimal performance. It turns out that in practice it seems best always to start the range at 3. This manually modified program yields a pair of permutations with d -invariant $2n - 5$ for all n .¹ This performance was equal to that of the best pair of permutations we came up with for odd n , and worse by 1 for even n . The permutations generated by this program were not identical to the ones we knew, but were in some sense “in the same spirit” – in particular, the number of involutions in both sets of permutations is linear in n , meaning that we are very close to the bottom of the Bruhat order.

¹Or so the data suggests; since this turned out not to be optimal, we did not write down a proof that this works for all n .

4.2. AlphaEvolve experiments. We then moved on to AlphaEvolve, another evolutionary protocol designed by DeepMind as a successor to FunSearch. AlphaEvolve is expected to be released as an open-source package in the near future, but for the moment we ran our trials internally at DeepMind. AlphaEvolve has already been observed to do well at example-finding in a wide variety of problems in both discrete and continuous areas of mathematics, and in some cases to find examples for a single problem instance that generalize to an infinite family of instances [Geo+25]. Customary practice in AlphaEvolve runs is to use a less terse, more directive prompt. (From [Geo+25]: “Giving AlphaEvolve an insightful piece of expert advice in the prompt almost always led to significantly better results.”) Here’s the prompt we used:

The prompt begins

```
Maximizing the d-invariant of Kazhdan-Lusztig polynomials
```

```
Act as an expert in computational algebra and combinatorial optimization. Your task is to
↪ find pairs of permutations `x` and `y` in the symmetric group  $S_n$  that have the largest
↪ possible "d-invariant", denoted  $d(x,y)$ . This invariant is a coefficient of a
↪ Kazhdan-Lusztig polynomial.
```

```
You need to produce a search function that, given an integer `n`, finds the best pair of
↪ permutations of size `n`. A permutation should be represented as a tuple of integers
↪ from 1 to `n`. For example, for  $n=3$ , a valid pair could be  $((1, 3, 2), (3, 2, 1))$ .
```

and ends

```
**Hint:** It is known that the maximum value of  $d(x,y)$  for large `n` is at least  $2n-5$ 
↪ (for odd `n`) and  $2n-4$  (for even `n`). Try to find pairs that meet or exceed this
↪ bound!
```

```
**Second Hint**: While the above prompt tells you to write a search function, what you
↪ really want is a general solution. Your program will be tested on some pretty large
↪ values of  $n$ , where search won't help you. Instead, while in the short term a search
↪ might help you, in the long term the only way you can solve this problem is by finding
↪ a general construction. This general construction can of course still contain a tiny
↪ bit of search in it, but it should contain some good structure based on ideas and
↪ understanding, it should NOT be just a general purpose search heuristic. We are here
↪ for the long term, so this is what you should focus on. Research maths is a marathon,
↪ not a sprint. Now go and find that general construction that solves the problem for all
↪  $n$ !
```

You got this! I believe in you!!!

Rather than training on a single n , we evaluated programs by computing the d -invariant of their output for all n from 10 to 50 and averaging the results. The idea is to incentivize evolution to produce programs that work for many or even all n , rather than being overfitted to the particular n trained on. This is the approach called “generalizer mode” in [Geo+25]. It’s worth noting that the “generalizer” mode of evaluation has been tried in many FunSearch experiment, with quite weak results; one apparent advantage of AlphaEvolve is it seems better able to aggregate information from combined evaluation on many problem instances.

AlphaEvolve ran for several hours, and produced programs which gave d -invariants exceeding $2n - 5$ for most n ; to the eye, the performance of these programs still seemed to be linear in n with constant 2. Letting AlphaEvolve run overnight, though, resulted in the qualitative improvement we report on in the main body of this paper. The resulting program produced d -invariants which appeared to be (and quickly proved to be) superlinear

in n . The program, its output, and the full prompt used to start the evolution process can be found at <https://github.com/JSEllenberg/Bruhat-hypercubes/tree/main>.

This run evaluated around 3000-5000 programs; its model calls were to a combination of Gemini 2.5 Flash and Gemini 2.5 Pro, and solicited from those models a number of tokens that would have cost about 70USD at public rates.²

The evolved program is rather long, and contains substantially more non-functional code than the FunSearch examples above. (Note that the instruction to "keep the code short" is no longer present in the prompt.) There is a "simulated annealing" routine which we believe is almost never actually executed, and a search over a large class of recursive constructions which in fact always ends up using one particular recursive construction.

Denoting by $\alpha(n)$ the d -invariant of the permutations in S_n outputted by the program, we observed what appears to be the relation $\alpha(2n) = 2\alpha(n) + n$, and in particular, that $\alpha(2^m) = m2^{m-1}$ in the range of values tested.

In fact, after inspection of the code we were able to come up with a simple description of what the program (or rather, the small part of the program that was actually doing the work) was doing when $n = 2^m$, and this is what gave rise to the definition of x_m and y_m appearing in the main body of the paper. In fact, x_3 and y_3 were the same permutations in S_8 we had already found in S_8 . So in an alternate universe, we might have noticed that these two permutations on 8 letters had this simple description in terms of length-3 binary expansion. But we did not in fact notice this until alerted to it by the presence of a program that computed this permutation for binary expansions of any length.

As discussed in section 3.1, these permutations of 2^m letters are already known in the world of low-discrepancy sequences. We searched the logs for relevant keywords to see whether AlphaEvolve was drawing these permutations from the existing literature, but found no references to low-discrepancy sequences. We did find some references to bit-reversal permutations, and even programs which computed x_m and y_m for $n = 2^m$; but these programs did not end up in the lineage that led to the final successful program! That's because these particular programs performed very poorly on values of n which were *not* powers of 2. This scenario illustrates the importance of thinking about the scoring function. If $s(n)$ is the d -invariant of the two permutations a program outputs in S_n , our scoring function was $\sum_{n=10}^{50} s(n)$. But since the maximal d -invariant in S_n is monotone in n , we could instead have used this knowledge and evaluated our function by $\sum_{n=10}^{50} \max_{i \leq n} s(i)$. This would have assigned a high score to the first functions to try the bit-reversal permutations for $n = 2^m$. In this case, it didn't matter, since AlphaEvolve eventually arrived at the bit-reversal permutations again, and this time with a program that performed adequately for all n in the range.

We had some prior intuition that pairs of permutations x, y with large d -invariant would have something to do with large hypercubes in the Bruhat order. Thus, it was very appealing and not completely surprising when we noticed that, for $m = 3$ and $m = 4$, the Kazhdan-Lusztig R-polynomial is a power of $(q - 1)$, which is consistent with the Bruhat interval $[x_m, y_m]$ being a hypercube (that is, equivalent as a directed graph to $(\mathbb{Z}/2\mathbb{Z})^d$ with edges corresponding to changing a single coordinate from 0 to 1.) It thus seemed likely that this interval was in fact a hypercube of dimension $m2^{m-1}$. We expected that this might be difficult to prove, since in general it is challenging even to compute the cardinality of the Bruhat interval between two permutations, let alone determine its structure. In this case, however, as we explain in the main body of the paper, we were able to find an explicit

²Of course these dollar amounts are quite unstable; we merely want to make the point that the kind of experiments we're talking about don't require computational resources that are infeasible for individual researchers.

description of the permutations lying in the Bruhat interval $[x_m, y_m]$ which made everything tractable.

It is probably not a coincidence that the three desirable phenomena

- Qualitative, not just incremental improvement over previous best known optimum for the combinatorial problem;
- Highly interpretable program;
- Output which can be proven to give good examples for all n , not just those in the training range, and which seems to be of interest for reasons beyond its good score on the trained-for objective;

all occur together in this case.

It's interesting that we found a very large hypercube while running a search optimized for a different objective, maximizing the d -invariant. It is interesting to wonder whether AlphaEvolve would have done as well if asked directly to find a Bruhat interval that was a large hypercube, rather than a large d -invariant.

REFERENCES

- [AB16] Hiraku Abe and Sara Billey. “Consequences of the Lakshmibai-Sandhya theorem: the ubiquity of permutation patterns in Schubert calculus and related geometry.” In: *Schubert calculus—Osaka 2012*. Vol. 71. Adv. Stud. Pure Math. Math. Soc. Japan, [Tokyo], 2016, pp. 1–52.
- [BG23] Grant T. Barkley and Christian Gaetz. “Combinatorial invariance for lower intervals using hypercube decompositions.” In: *Sém. Lothar. Combin.* 89B (2023), Art. 86, 8.
- [BG25] Grant T. Barkley and Christian Gaetz. “Combinatorial invariance for Kazhdan-Lusztig R -polynomials of elementary intervals.” In: *Math. Ann.* 392.3 (2025), pp. 3299–3317.
- [BB05] Anders Bjorner and Francesco Brenti. *Combinatorics of Coxeter groups*. Springer, 2005.
- [Blu+22] Charles Blundell et al. “Towards combinatorial invariance for Kazhdan-Lusztig polynomials.” In: *Represent. Theory* 26 (2022), pp. 1145–1191.
- [BBG97] Phelim Boyle, Mark Broadie, and Paul Glasserman. “Monte Carlo methods for security pricing.” In: *Journal of economic dynamics and control* 21.8-9 (1997), pp. 1267–1321.
- [BM01] Tom Braden and Robert MacPherson. “From moment graphs to intersection cohomology.” In: *Math. Ann.* 321.3 (2001), pp. 533–551.
- [Bre94] Francesco Brenti. “A combinatorial formula for Kazhdan-Lusztig polynomials.” In: *Inventiones mathematicae* 118.1 (1994), pp. 371–394.
- [BM25] Francesco Brenti and Mario Marietti. “Kazhdan-Lusztig R -polynomials, combinatorial invariance, and hypercube decompositions.” In: *Math. Z.* 309.2 (2025), Paper No. 25, 19.
- [Cas+25a] Roger Casals et al. “Cluster structures on braid varieties.” In: *J. Amer. Math. Soc.* 38.2 (2025), pp. 369–479.
- [Cas+25b] Roger Casals et al. “Comparing cluster algebras on braid varieties.” In: *arXiv preprint arXiv:2508.03816* (2025).
- [Cas+24] Marco Castronovo et al. “Cluster deep loci and mirror symmetry.” In: *arXiv preprint arXiv:2402.16970* (2024).
- [Dav+21] Alex Davies et al. “Advancing mathematics by guiding human intuition with AI.” In: *Nature* 600.7887 (2021), pp. 70–74.
- [Deo85] Vinay V. Deodhar. “On some geometric aspects of Bruhat orderings. I. A finer decomposition of Bruhat cells.” In: *Invent. Math.* 79.3 (1985), pp. 499–511.
- [DP10] Josef Dick and Friedrich Pillichshammer. *Digital nets and sequences: discrepancy theory and quasi-Monte Carlo integration*. Cambridge University Press, 2010.
- [Dye91] Matthew Dyer. “On the “Bruhat graph” of a Coxeter system.” In: *Compositio Math.* 78.2 (1991), pp. 185–191.
- [Ell+25] Jordan S Ellenberg et al. “Generative modeling for mathematical discovery.” In: *arXiv preprint arXiv:2503.11061* (2025).

- [FZ02] Sergey Fomin and Andrei Zelevinsky. “Cluster algebras. I. Foundations.” In: *J. Amer. Math. Soc.* 15.2 (2002), pp. 497–529.
- [GL24] Pavel Galashin and Thomas Lam. “Positroids, knots, and q, t -Catalan numbers.” In: *Duke Math. J.* 173.11 (2024), pp. 2117–2195.
- [GLS] Pavel Galashin, Thomas Lam, and Melissa Sherman-Bennett. “Braid variety cluster structures, II: general type.” In: *Inventiones Math.* (to appear) ().
- [GLS13] Christof Geiss, Bernard Leclerc, and Jan Schröer. “Factorial cluster algebras.” In: *Doc. Math.* 18 (2013), pp. 249–274.
- [Geo+25] Bogdan Georgiev et al. “Mathematical exploration and discovery at scale.” In: *arXiv preprint arXiv:2511.02864* (2025).
- [Gro+18] Mark Gross et al. “Canonical bases for cluster algebras.” In: *J. Amer. Math. Soc.* 31.2 (2018), pp. 497–608.
- [GW24] Maxim Gurevich and Chuijia Wang. “Parabolic recursions for Kazhdan-Lusztig polynomials and the hypercube decomposition.” In: *Selecta Math. (N.S.)* 30.5 (2024), Paper No. 81, 27.
- [HKW25] Chris Hone, Finn Klein, and Geordie Williamson. “Moduli of Embeddings of Bruhat Graphs.” Work in preparation. 2025.
- [HV09] Axel Hultman and Kathrin Vorwerk. “Pattern avoidance and Boolean elements in the Bruhat order on involutions.” In: *Journal of Algebraic Combinatorics* 30.1 (2009), pp. 87–102.
- [Kim25] Soyeon Kim. “Cluster automorphism group of braid varieties.” In: *arXiv preprint arXiv:2505.14889* (2025).
- [KWY13] Allen Knutson, Alexander Woo, and Alexander Yong. “Singularities of Richardson varieties.” In: *Math. Res. Lett.* 20.2 (2013), pp. 391–400.
- [LS22] Thomas Lam and David E. Speyer. “Cohomology of cluster varieties I: Locally acyclic case.” In: *Algebra Number Theory* 16.1 (2022), pp. 179–230.
- [LMP21] Eunjeong Lee, Mikiya Masuda, and Seonjeong Park. “Toric Bruhat interval polytopes.” In: *Journal of Combinatorial Theory, Series A* 179 (2021), p. 105387.
- [Nie87] Harald Niederreiter. “Point sets and sequences with small discrepancy.” In: *Monatshefte für Mathematik* 104.4 (1987), pp. 273–337.
- [Pat21] Leonardo Patimo. “A combinatorial formula for the coefficient of q in Kazhdan-Lusztig polynomials.” In: *Int. Math. Res. Not. IMRN* 5 (2021), pp. 3203–3223.
- [Rea02] Nathan Paul Reading. *On the structure of Bruhat order*. Thesis (Ph.D.)—University of Minnesota. ProQuest LLC, Ann Arbor, MI, 2002, p. 82.
- [Rom+24] Bernardino Romera-Paredes et al. “Mathematical discoveries from program search with large language models.” In: *Nature* 625.7995 (2024), pp. 468–475.
- [Ten07] Bridget Eileen Tenner. “Pattern avoidance and the Bruhat order.” In: *Journal of Combinatorial Theory, Series A* 114.5 (2007), pp. 888–905.
- [Ten22] Bridget Eileen Tenner. “Interval structures in the Bruhat and weak orders.” In: *Journal of Combinatorics* 13.1 (2022).
- [Xia02] Yi-Jun Xiao. “Some geometric properties of $(0, m, 2)$ -nets in base $b \geq 2$.” In: *Monte Carlo Methods Appl.* 8.1 (2002), pp. 97–106.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF WISCONSIN-MADISON, MADISON, WI USA
Email address: `ellenber@math.wisc.edu`

DEPARTAMENTO DE MATEMÁTICAS, UNIVERSIDAD DE CHILE, CHILE
Email address: `nlibedinsky@gmail.com`

INSTITUTO DE MATEMÁTICAS, UNIVERSIDAD DE TALCA, CHILE
Email address: `dplaza@utalca.cl`

INSTITUTO DE MATEMÁTICAS, UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO. MÉXICO
Email address: `simental@im.unam.mx`

SCHOOL OF MATHEMATICS AND STATISTICS, UNIVERSITY OF SYDNEY, AUSTRALIA
Email address: `g.williamson@sydney.edu.au`