Models of random spanning trees

Eric Babson* Moon Duchin† Annina Iseli‡ Pietro Poggi-Corradini§
Dylan Thurston¶ Jamie Tucker-Foltz‖

March 13, 2026

Abstract

There are numerous randomized algorithms to generate spanning trees in a given ambient graph; several target the uniform distribution on trees (UST), while in practice the fastest and most frequently used draw random weights on the edges and then employ a greedy algorithm to choose the minimum-weight spanning tree (MST). Though MST is a workhorse in applications, the mathematical properties of random MST are far less explored than those of UST. In this paper we develop tools for the quantitative study of random MST. We consider the standard case that the weights are drawn i.i.d. from a single distribution on the real numbers, as well as successive generalizations that lead to *product measures*, where the weights are independently drawn from arbitrary distributions.

1 Introduction

The theoretical computer science literature includes a wide variety of randomized algorithms for choosing a spanning tree of a graph G. First, it is natural to try to sample uniformly: We know how to count the number of spanning trees of any graph from Kirchhoff's Matrix–Tree formula [10], and from this it is possible to build an efficient uniform sampler. In the late 1980s and 1990s, Aldous [2], Broder [6], and most notably Wilson [28] devised simple, faster algorithms that build uniformly random trees via random walks on G. In a series of papers starting in the late 2010s, Anari, Liu, Oveis Gharan, Vinzant and various collaborators have had a series of results leveraging matroid theory to define a Markov chain process that they showed to be rapidly mixing, converging to the uniform distribution [3, 4].

However, a different algorithm is ubiquitous in practical applications, and it is so fundamental that it is often at the heart of a first course in discrete mathematics: assign every edge a random weight and then find a minimum spanning tree with respect to those weights, which can be computed greedily using Kruskal's algorithm [12]. Unlike the previous methods, this does not target the uniform distribution. Our motivation in this paper is to advance the systematic study of the differences between uniform spanning trees (UST) and different versions of minimum spanning trees (MST). It is already of interest simply to learn more properties of ordinary MST (with weights drawn i.i.d. from $[0, 1]$), since it is widely used but not (quantitatively) well studied.

The simplest example of a graph in which UST is different from MST is a square with a diagonal (four vertices, five edges). Here, one easily verifies that there are eight possible spanning trees; four involve the diagonal and four do not. Under UST, they are all equally likely, but under ordinary MST, the ones with the diagonal collectively have probability $8/15$. It turns out that a small modification to the edge weights will restore the greedy process to uniformity: if the outer edges have weights drawn from the same interval $[0, 1]$ (say), but the diagonal is weighted from a slightly perturbed interval $[\varepsilon, 1 + \varepsilon]$, then the minimum-weight

*Department of Mathematics, University of California, Davis, Davis, CA, USA
†Department of Computer Science and Data Science, University of Chicago, Chicago, IL, USA
‡Institut de mathématiques, École polytechnique fédérale de Lausanne (EPFL), Lausanne, Switzerland
§Department of Mathematics, Kansas State University, Manhattan, KS, USA
¶Department of Mathematics, Boston College, Chestnut Hill, MA, USA
‖Yale School of Management and Computer Science Department, Yale University, New Haven, CT, USA.
Correspondence to: j.tuckerfoltz@yale.edu

tree is uniformly distributed.[1] This is our first example of a non-i.i.d. *product measure* on the edges—that is, a choice of real-valued distribution on each edge. This example is a special form of generalized MST, that of drawing uniformly from shifted intervals.

To motivate pushing to the next level of complexity, we next consider K_4, the complete graph on four vertices. In this case, we can once again construct a product measure on the edges so that the minimum-weight tree recovers UST. However, shifted intervals are provably not enough and we must resort to more general measures. Fix a small ε and draw the edge weight on one edge from $\frac{1}{2}\delta_\varepsilon + \frac{1}{2}\delta_1$ (where δ_x denotes the atomic probability distribution supported on $\{x\}$) and on the opposite edge (the one that shares no endpoint with the first) from $\frac{1}{2}\delta_0 + \frac{1}{2}\delta_{1-\varepsilon}$. The other four are drawn uniformly from $\left[\frac{1}{4}, \frac{3}{4}\right]$. This is a more general example of a product measure, and the reader can verify that this weighting scheme makes all spanning trees equally likely.

Stepping back, we can seek to understand the space of probability distributions on minimum spanning trees by first understanding the achievable distributions of weights. The question of what distributions can be achieved on $\mathbb{R}^m$ from m independent random variables is fundamental, and one aspect of the question has received significant study for many decades: the projection onto pairwise comparisons. In 1959, Steinhaus and Trybuła began to study what they called "the paradox of three random variables," namely the fact that triples exist for which $\mathbb{P}(X > Y)$, $\mathbb{P}(Y > Z)$, $\mathbb{P}(Z > X)$ are all more than $\frac{1}{2}$ [22]. Later this would be called the problem of "intransitive dice" (see §1.3 for further discussion). For our application, we require more information than the projection onto pairwise comparisons. Since Kruskal-style selection of the minimum-weight tree is greedy, it depends only on the order of the edge weights. That is, a draw from a product measure induces a permutation via the order of the weights, and the permutation suffices to specify a choice of tree. So if we write $\mathcal{M} = \mathcal{M}_m$ for the space of non-colliding product measures on m real-valued random variables (i.e., such that the probability is zero that any two draws are equal) and $\Delta(S_m)$ for the probability distributions on permutations of m symbols, then we have defined a map $\psi\colon \mathcal{M} \to \Delta(S_m)$ and we set out to describe the image $P_m = \psi(\mathcal{M})$ of product measures. Surprisingly, though this appears to be a very natural and elementary question, we have not found this much discussed in the probability literature.

1.1 Summary of contributions

We study successive generalizations of the ordinary MST theory. Given a finite graph $G = (V, E)$ with vertex set V and edge set E, write $(X_i)_{i\in E}$ for a collection of real-valued random variables indexed by E. We will focus much of our study on random spanning trees which arise as minimum-weight trees given edge weights drawn independently from the X_i.

- *Ordinary MST* (also denoted MST_0) is a specific product measure in which there is a single measure X and all $X_i \sim X$. Without loss of generality, the weights are drawn uniformly from $[0, 1]$. (§3)

Theorems 3.4 and 3.5 give exact formulas for the MST_0 probability of any particular spanning tree in a graph, but they involve sums over permutations of edges outside and inside the tree, respectively. Though this can be improved from factorial time to exponential time, it is still more useful for insight than for practical calculation in large graphs. We show that the probability of sampling any (labeled) star in a K_n is exactly $1/(2n-3)!!$. We then develop a pair of "rotation tricks" that let us show in Theorem 3.13 that in fact stars have the highest probability weight of any labeled trees and paths have the lowest. In the process we also apply rotations to show in Theorem 3.11 that random graphs have $\mathrm{MST}_0 \neq \mathrm{UST}$.

- *Shifted-interval MST* is the class of product measures where each X_i is uniform on a unit interval $[s_i, s_i + 1]$. We briefly consider more general *connected-support* measures, where the support of each edge distribution is a connected subset of $\mathbb{R}$. (§4)

We build a parameter space for shifted-interval measures $M_{\mathsf{s}} \subset \mathcal{M}$ and show in Theorem 4.6 that shifts do not suffice to recover the uniform distribution on spanning trees for K_n, $n \geq 4$. In the process, we motivate the study of arbitrary product measures by showing some limitations of measures with connected support.

[1] The exact value of $\varepsilon \approx 0.03$ needed to recover UST has no nice closed form; it is the real root of the quintic polynomial $6x^5 - 20x^3 + 30x - 1$.

- *Arbitrary product measures* have no restrictions other than the non-colliding property (the probability of drawing two identical values is zero). (§5)

To analyze arbitrary product measures, we develop a discrete abstraction called *weighted words* that helps isolate the combinatorial aspects of the problem. We obtain Theorem 5.4, which states that every non-colliding product measure on m variables can be represented by a weighted word of finite length (bounded in terms of m). We construct universal words that suffice to hit all of P_m as their weights vary, and we employ classical quadrature schemes (from the theory of integration) to give efficient words that induce the uniform measure on permutations. Finally, we study the dimension of P_m and prove an upper bound for it in Theorem 5.13, also verifying computationally that the upper bound is an equality for $m \leq 7$.

1.2 An application of generalized MST

Some of the original motivation for this study came from shifted-interval MST because it is already in practical use in a popular graph algorithm. Random spanning trees are a key ingredient in so-called *recombination* algorithms that step from one random graph partition to another [7]. Recombination runs by iterating a merge-split procedure: two parts of a graph partition are fused, and then a random spanning tree is used to divide them in a new way. A principal application is for generating random political districting plans. When users desire to make it more likely that two nodes are placed in *different* parts of a partition, they can add a positive "surcharge" to the weight on the edge between those nodes. When a collection of contiguous nodes makes up a region that users prefer to keep in the *same* piece, the same idea can be used to surcharge the boundary edges of the region. This has the effect that minimum spanning tree is more likely to restrict to a tree on the designated region; in the bipartition step, that means the region will be kept whole or split at most once.

A primary example is keeping counties together, which is a traditional preference in forming districting plans. If within-county edges are drawn from $[0, 1]$ and between-county edges from a shifted interval $[s, s+1]$ for $0 < s \leq 1$, then for higher values of s far fewer counties will be split over the long term, a phenomenon illustrated in Figure 1. (There is no further effect for s larger than 1, because at that point the surcharged weights are necessarily larger than the $[0, 1]$ weights, which is all that is seen by Kruskal's algorithm.)

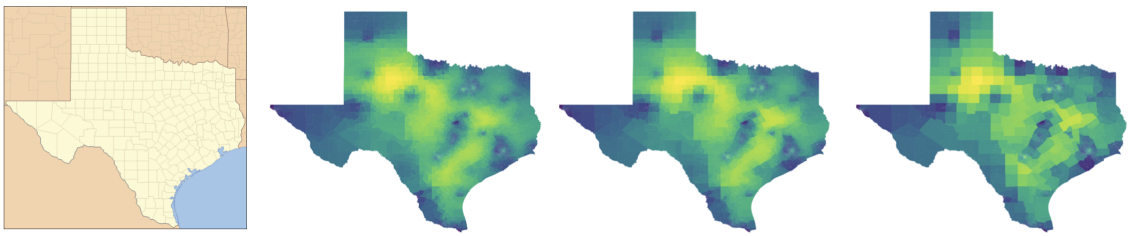

Figure 1: In [8], a "recombination" Markov chain is run to draw random partitions of Texas into 150 legislative districts. The Markov chain combines and re-splits two districts at a time by drawing and bisecting a random spanning tree. These figures show heatmaps from three different runs, with the ∼9000 precincts of Texas colored on a scale from dark blue (reassigned rarely) to yellow (reassigned frequently). When the spanning tree step uses MST weights drawn from $[0, 1]$, there is no particular relationship to county boundaries. As between-county edges are surcharged by $s = 0.1$ (middle) and then $s = 1.0$ (right), county boundaries become visible, since the steps tend to keep counties intact within the partition and so reassign whole counties at a time.

Empirically, this works very well to keep designated regions intact in a random partition process. But theoretically, it is difficult to characterize the distribution induced by the shift.

1.3 Related work

The literature giving quantitative contrasts between UST and ordinary MST is limited. The Lyons–Peres book *Probability on Trees and Networks* [14] is a valuable and thorough reference in which Chapter 11 is devoted to minimal spanning forests in the context of percolation theory. In the setting of "wired" and "free" minimal spanning forests on infinite graphs, the authors give results on degree distribution, component sizes, and the number of ends. A paper by Garban, Pete, and Schramm [9] explores the scaling limit of MST on triangular lattices in the plane. The authors show invariance under rotations, scalings and translations, but do not expect full conformal invariance (as in the case of SLE) to hold. Simultaneous recent works by Makowiec-Salvi-Sun [15] and Kúsz [13] compare MST to UST by creating Boltzmann-like distributions on spanning trees that interpolate between them.

In settings closer to ours, one of the striking quantitative comparison results known to us is in a paper by Addario-Berry, Broutin, and Reed [1], where the authors show that the expected diameter of a spanning tree of the complete graph K_n is on the order of $n^{1/3}$ under MST_0—in contrast to the classical result that the expected UST diameter is $n^{1/2}$. More recently, Tapp [24] considers MST_0 on square grid graphs, giving upper and lower bounds for the probability of sampling a given tree. He also provides conjectures about the most and least likely tree structures on grid graphs—we settle the corresponding conjecture for random spanning trees in complete graphs (Theorem 3.13).

Above, we described a question about $P_m \subset \Delta(S_m)$ that is well-studied: *Can you build n weighted dice so that each one beats the next (cyclically) with probability greater than* 50%*?* After Steinhaus-Trybuła gave a positive answer in 1959 [22], Trybuła launched a quantitative investigation in [25, 26] whose development was later surveyed by Savage in [21]. To state some results, consider the set T_m which is the projection of $P_m \subset \Delta(S_m) \subset \mathbb{R}^{m!-1}$ to the much smaller-dimensional set $[0,1]^{\binom{m}{2}}$, only keeping track of the pairwise probabilities. For example, the point $\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right) \in T_3$ is realized by standard dice, and the paradox amounts to the observation that all three coordinates can exceed $\frac{1}{2}$. The Trybuła region T_3 is shown in Figure 2.

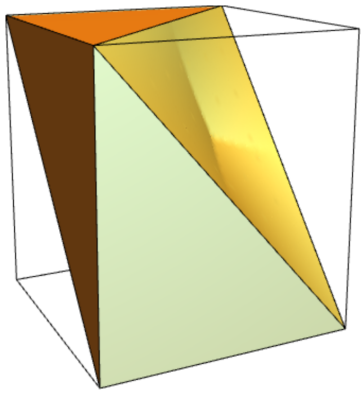

Figure 2: The Trybuła region T_3. Given independent random variables X_1, X_2, X_3 that are the components of a non-colliding product measure $\mathcal{D}$, the three coordinate axes are $x = \mathbb{P}_{\mathcal{D}}(X_1 > X_2)$, $y = \mathbb{P}_{\mathcal{D}}(X_2 > X_3)$, and $z = \mathbb{P}_{\mathcal{D}}(X_3 > X_1)$. The vertices of the cube that are hit by T_3 correspond to the pure permutations (for example, $(1,1,0)$ comes from $X_1 > X_2 > X_3$), whereas $(0,0,0)$ and $(1,1,1)$ are not hit.

On general principles, the feasible locus T_3 is contained in the convex hull of the six vertices defined by pure permutations, but T_3 is not quite convex: as formulated by Suck [23], it is the region defined by

$$\min\big(x + yz, y + xz, z + xy\big) \le 1; \qquad \min\big(\bar{x} + \bar{y}\bar{z}, \bar{y} + \bar{x}\bar{z}, \bar{z} + \bar{x}\bar{y}\big) \le 1,$$

where x, y, z are the probabilities that $X_1 > X_2$, $X_2 > X_3$, $X_3 > X_1$, respectively, and $\bar{x} = 1 - x$, $\bar{y} = 1 - y$, $\bar{z} = 1 - z$ are the complementary probabilities. There are still open questions about T_m; for instance, Komisarski only recently (2021) found the maximum extent of intransitivity [11]. Concretely, for $m \ge 3$, Komisarski maximizes p_m for which there exist dice $(X_i)_{i=1}^m$ where X_i beats X_{i+1} with probability p_m.

Much less is known about the full permutation locus P_m, even for $m = 3$. We will discuss it in §5.4.

Acknowledgements

We gratefully acknowledge the Simons-Laufer Mathematical Sciences Institute (SLMath) for hosting subsets of the authors in two recent semesters, during the thematic programs on the Analysis and Geometry of Random Spaces (AGRS) and Algorithms, Fairness, and Equity (AFE). The authors thank the National Science Foundation for partial support through grants DMS-2005512 (MD), DMS-2154032 (PPC), DMS-2110143 (DT), and DGE-1745303 (JTF), and the University of Fribourg for postdoc travel funding (AI). We also thank Peter Rock for his work on the replication repository and Nantel Bergeron, Adi Glücksam, and Russell Lyons for many extremely helpful conversations.

2 Preliminaries

We first set some notation. Recall that $\Delta(A)$ is the set of distributions on a set A; if A is a finite set, then a distribution is a non-negative function on A with total mass one. For a distribution $\mathcal{D} \in \Delta(A)$ we denote a random variable X sampled from $\mathcal{D}$ by writing $X \sim \mathcal{D}$. All graphs we consider are simple and undirected, with $V(G)$ and $E(G)$ for the vertex and edge sets, and write $n = |V|$ and $m = |E|$. We write $ST(G)$ for the set of all spanning trees of G and let $\mathrm{UST} \in \Delta(ST(G))$ denote the uniform distribution on spanning trees. In other words,

$$\forall T, T' \in ST(G): \quad \mathbb{P}_{\mathrm{UST}}(T) = \mathbb{P}_{\mathrm{UST}}(T').$$

We will also adopt the mild abuse of notation allowing us to write $e \in G$ rather than the fuller $e \in E(G)$, and $H \setminus e$ or $H \cup e$ for the subgraph of G where an edge e is removed from or added to a subgraph H. That is, we will minimize the use of set notation except where the sets themselves are being emphasized, so that H can flexibly refer to a subgraph or its edge set.

For any finite nonempty set A, we let $S(A)$ denote the set of orderings (permutations) of A, in particular writing S_n for the symmetric group on n symbols. For a graph G and a choice of distinct real-valued weights on edges, there is a unique *minimum spanning tree*, defined by the smallest possible sum of weights of included edges. Kruskal's algorithm (1956) is based on the observation that a greedy process suffices to find it: begin with the lowest-weighted edge, and successively add the other edges in order of weight, rejecting an edge only if it forms a cycle with others already selected [12]. Since this is proven to globally minimize total weight, it follows that the order of the m edge weights sufficed: the map from weights to trees factors through the permutations S_m that record the order of the weights.

A *(non-colliding) product measure* on a set S is defined using a collection of real-valued random variables $\{X_i \mid i \in S\}$ such that, for all distinct $i, j \in S$, $\mathbb{P}[X_i = X_j] = 0$; a set of independent draws gives the product measure. A product measure on m variables induces a distribution in $\Delta(S_m)$ via the order of the weights. Therefore, as described above, a product measure on the edge set E of a graph induces a distribution in $\Delta(ST(G))$.

A key tool to understand variants of MST will be the use of what we call *broken cycles*.

Definition 2.1 (Broken cycles)**.** Let T be a spanning tree of a graph G and $e \in G \setminus T$ a non-edge of the tree. Then there is a unique cycle $C_e = C_{T,e}$ in G consisting of e and the unique non-backtracking path $P_e = P_{T,e}$ in T between the endpoints of e; we call C_e the *broken cycle* for e in the graph and call P_e the *broken cycle* for e in the tree.

The relevance of broken cycles to MST is that Kruskal's algorithm rejects a proposed edge e for inclusion in a minimum spanning tree T if and only if its weight is higher than those in $P_{T,e}$.

Next, we set up notation for a key criterion about when MST induces a tree T.

Definition 2.2 (Cycle relation)**.** Given a connected graph G and a spanning tree $T \subset G$, define the *cycle relation* R_T between the edges and non-edges of T (i.e., between $E(T)$ and $E(G \setminus T)$) by

$$e_i \, R_T \, e_j \qquad \text{iff} \qquad e_i \in P_{T,e_j},$$

for $e_i \in T$, $e_j \notin T$. Sometimes it is useful to think of R_T as a subset of $E(T) \times E(G \setminus T)$ and write $(e_i, e_j) \in R_T$.

In terms of this relation, it is easy to check the following.

Proposition 2.3 (Cycles versus weights)**.** *A set of distinct weights $\{w_i\}$ on the edges of G induces T as a minimum spanning tree if and only if for all $e \in T$, $e' \notin T$ and corresponding weights w, w',*

$$e \, R_T \, e' \implies w < w'.$$

Proof. Assume first that T is the MST, $e \in T$, $e' \notin T$, and e, e' are in the same broken cycle. If $w > w'$, then we could swap e with e' and obtain a tree T' with lower total weight than T, contradicting the minimality of T. Conversely, suppose that T has the property in the statement. If any edge is rejected in the process of running Kruskal's algorithm, it must be because it completes a cycle whose other edges have lower weight. No edge of T can have this property by assumption. So T must be the tree constructed in a run of Kruskal's algorithm. □

3 Ordinary MST

3.1 Inductive formulas

Let G be a connected graph with n vertices and m edges. Given a spanning tree $T \subset G$, we wish to find $\mathbb{P}_{\mathrm{MST}}(T; G)$, the chance that T is chosen as the minimum spanning tree when MST is applied to G with i.i.d. random edge weights. We will give two different inductive formulas for this, based on two algorithms from the same paper of Kruskal [12]: the first, now known as *Kruskal's algorithm*, starts with an empty forest F and successively adds the minimal-weight remaining edge that does not create a cycle until F becomes a spanning tree. The second, now known as the *reverse-delete algorithm*, starts with the entire graph $H = G$ and successively deletes the maximal-weight edge of H that does not disconnect H until H becomes a tree.

A *spanning subgraph* $H \subset G$ is a not-necessarily-connected subgraph with the full vertex set. For $H \subset G$ a spanning subgraph, set

$$\partial H \coloneqq \{e \in G \setminus H \mid e \text{ connects different components of } H\}$$
$$H^\circ \coloneqq \{e \in H \mid e \text{ is not a separating edge of } H\}.$$

We can regard ∂H as the cut-set of H as a partition of G, and H° as the set consisting of edges lying totally within some biconnected (2-connected) component of H. Suppose that $\sigma \in \mathbb{R}^E \setminus \Delta$ is a set of distinct edge weights, i.e., avoiding the "fat diagonal" Δ, so that all $\sigma(e)$ are distinct. Let $\mathcal{F}_k$ be the family of all forests $F \subset G$ with $|F| = k$ edges, for $k = 0, \dots, n-1$. Let $\mathcal{H}_j$ be the family of all connected spanning subgraphs $H \subset G$ with $|H| = n - 1 + j$ edges, for $j = 0, \dots, m - n + 1$.

Kruskal's algorithm can now be phrased as the greedy process that, for a given σ, outputs the (nested) tuple of spanning forests $(F_k)_{k=0}^{n-1}$ with $F_k \in \mathcal{F}_k$ given by starting with the empty forest F_0 (containing all the nodes but no edges) and successively taking $F_i \coloneqq F_{i-1} \cup e$ for $e \in \partial F_k$ with minimum $\sigma(e)$. Likewise, the reverse-delete algorithm outputs nested connected subgraphs $(H_j)_{j=0}^{m-n+1}$ (generated in the reverse order), where $H_{m-n+1} = G$ and $H_{j+1} \setminus H_j$ is the edge maximizing $\sigma(e)$ for $e \in H_{j+1}^\circ$. Kruskal proves that if all edge weights are distinct, then $F_{n-1} = H_0$, and this is the unique minimum-weight spanning tree. We can introduce graph-valued random variables, X_k for the k-edge forests seen in the first process and Y_k for the $(n-1+k)$-edge spanning subgraphs seen in the second. Along the way to computing $\mathbb{P}_{\mathrm{MST}}(T)$, we will compute intermediate probabilities

$$\mathbb{P}_{\mathrm{Kru}}(F; G) \coloneqq \mathbb{P}(X_{|E(F)|} = F) \quad \text{and} \quad \mathbb{P}_{\mathrm{RD}}(H; G) \coloneqq \mathbb{P}(Y_{|E(H)|-(n-1)} = H),$$

suppressing G from the notation when it is clear.

Theorem 3.1 (Kruskal induction)**.** *For $F \subset G$ a forest consisting of $k \geq 1$ edges in a connected ambient graph G, the probability of encountering it in a run of Kruskal's algorithm can be computed by adding each of its edges "last," as follows:*

$$\mathbb{P}_{\mathrm{Kru}}(F) = \sum_{e \in F} \mathbb{P}(X_k = F \mid X_{k-1} = F \backslash e) \cdot \mathbb{P}(X_{k-1} = F \backslash e) = \sum_{e \in \mathrm{E}(F)} \frac{\mathbb{P}_{\mathrm{Kru}}(F \setminus e)}{|\partial(F \setminus e)|}.$$

The idea is to condition on the last edge e added to get F, and then notice that each such edge e is equally likely in $\partial(F \setminus e)$.

For a tree T, we can leverage this to give an inductive algorithm for computing $\mathbb{P}_{\mathrm{MST}}(T)$ with a runtime bound on the order of $n \cdot 2^n$. Namely, we compute $\mathbb{P}_{\mathrm{Kru}}(F)$ for all of the 2^{n-1} subforests $F \subsetneq T$ in order from the fewest edges to the most, using Theorem 3.1 initialized with probability 1 of encountering the empty forest. This needs less than 2^n space; in fact, we only need to remember one generation back, so this can be refined. At the stage of choosing k edges in the forest, there are k numbers to add, so the time complexity is $\sum k\binom{n-1}{k} = (n-1)2^{n-2}$. For the reverse-delete algorithm, we have an entirely analogous statement.

Theorem 3.2 (Reverse-delete induction)**.** *For $H \subsetneq G$ a connected proper spanning subgraph of G, the probability of encountering it in a run of reverse-delete can be computed by considering the removal of each of its missing edges in turn, as follows:*

$$\mathbb{P}_{\mathrm{RD}}(H) = \sum_{e \in G \setminus H} \mathbb{P}(Y_k = H \mid Y_{k+1} = H \cup e) \cdot \mathbb{P}(Y_{k+1} = H \cup e) = \sum_{e \in \mathrm{E}(G \setminus H)} \frac{\mathbb{P}_{\mathrm{RD}}(H \cup e)}{|(H \cup e)^\circ|}.$$

For the corresponding algorithm, it suffices to store all the connected supergraphs of the tree, and there are at most 2^{m-n+1} of these. For runtime, we have to add j terms in each of the $\binom{m-n+1}{j}$ terms, and the sum simplifies to $(m-n+1)2^{m-n}$.

Comparing the runtimes, we find that the Kruskal induction is more efficient for computing $\mathbb{P}_{\mathrm{MST}}(T)$ unless the graph is very sparse. In fact, in the reverse-delete case, we can often speed up this computation: frequently, the probabilities are independent of H.

Proposition 3.3. *If $H \in \mathcal{H}_j$ is biconnected, then all of its supergraphs in G are as well, so we could have deleted down to it in any order. That is, $\mathbb{P}_{\mathrm{RD}}(H) = 1/\binom{m}{m-(n-1+j)}$.*

There is a similar "dual" proposition for the Kruskal induction; instead of considering graphs for which no single edge can disconnect if removed (i.e., biconnected graphs), we would consider graphs for which no single edge could form a cycle if added.

3.2 Global formulas

The formulas above are inductive; we now write global formulas directly, first an "external" formula for $\mathbb{P}_{\mathrm{MST}}(T)$ that corresponds to the reverse-delete algorithm and then an "internal" formula that follows the steps of Kruskal.

Let $E = E(G \setminus T)$ be the external edges, i.e., the edges of G that are not in T. We first give the formula supposing we fix an order $\pi_E = (e_1, \ldots, e_{m-n+1})$ of weights on E. We will compute $\mathbb{P}_{\mathrm{MST}}(T, \pi_E)$, by which we mean the probability in an i.i.d. choice of weights σ that the external edges occur in the relative order π_E and the minimum spanning tree is T. For an edge $e \in E$, recall that $C_e \subset T \cup e$ is the corresponding broken cycle in G. Define $D_k(\pi_E) = D_k := \bigcup_{j=1}^{k} C_{e_j}$ to be the union of broken cycles up to stage k.

Theorem 3.4 (External formula)**.** *For any spanning tree $T \subset G$ and permutation π_E of its non-edges,*

$$\mathbb{P}_{\mathrm{MST}}(T, \pi_E) = \prod_{j=1}^{m-n+1} \frac{1}{|D_j|}, \qquad \text{giving} \qquad \mathbb{P}_{\mathrm{MST}}(T) = \sum_{\pi_E \in S(E(G \setminus T))} \prod_{j=1}^{m-n+1} \frac{1}{|D_j(\pi_E)|}.$$

Note that, for given weights σ inducing π_E in the expression for T, the weight $\sigma(e_j)$ must be the largest among any of the edges in the set D_j. For a single j, this happens with probability $1/|D_j|$, and the crux of the proof of Theorem 3.4 will be to see that all of these events are independent.

To connect this external formula to the inductive expression for reverse-delete, set $H_j = D_j \cup T = T \cup \{e_1, \ldots, e_j\}$. These H_j are the graphs that appear in the reverse-delete process, while on the other hand it is easy to see that $D_j = H_j^\circ$.

For the internal global formula, instead of fixing the order of the external edges we order the weights within the tree: for a spanning tree and any permutation $\pi_I = (e_j)_{j=1}^{n-1}$ of the edges in T, let $\mathbb{P}_{\mathrm{MST}}(T, \pi_I)$ be the probability for an i.i.d. choice of weights σ that the edges in T are in weight order π_I and the minimum spanning tree is T. Given such an order, let $F_k = \bigcup_{j=1}^{k} \{e_j\} \subset T$ be the forest (or partial spanning tree) constructed from the first k edges.

Theorem 3.5 (Internal formula)**.** *For a spanning tree $T \subset G$ and a permutation π_I of its edges,*

$$\mathbb{P}_{\mathrm{MST}}(T, \pi_I) = \prod_{j=0}^{n-2} \frac{1}{|\partial F_j|}, \quad \text{giving} \quad \mathbb{P}_{\mathrm{MST}}(T) = \sum_{\pi_I \in S(E(T))} \prod_{j=0}^{n-2} \frac{1}{|\partial F_j(\pi_I)|}.$$

The proof is similar: for given weights σ that are counted in $\mathbb{P}_{\mathrm{MST}}(T, \pi_I)$, the weight $\sigma(e_j)$ must be the smallest among any of the edges in the set ∂F_{j-1}. For a single j, this happens with probability $1/|\partial F_{j-1}|$, and again we must check that these are independent. We note that the internal formula appears as Proposition 11.2 of the Lyons-Peres book *Probability on Trees and Networks* [14], but without an argument for independence.

To illustrate the independence argument, we now run the proof of the internal formula on an example before remarking on how this generalizes to arbitrary $T \subset G$.

Example 3.6 (An example in K_5)**.** We compute the probability that the spanning tree T shown in Figure 3 is selected from a K_5 under ordinary MST, with the edges appearing in the indicated order $(a, c), (b, c), (d, e), (c, d)$. This happens iff

- (c, d) has the smallest weight among the 6 edges joining $\{a, b, c\}$ to $\{d, e\}$;
- (d, e) has the smallest weight among the 7 edges joining any two of $\{a, b, c\}$, $\{d\}$, and $\{e\}$;
- (b, c) has the smallest weight among the 9 edges joining any two of $\{a, c\}$, $\{b\}$, $\{d\}$, and $\{e\}$; and
- (a, c) has the smallest weight among all 10 edges.

That is, each edge highlighted in white on the right-hand side of Figure 3 must have the lowest weight among the edges in its smallest enclosing region.

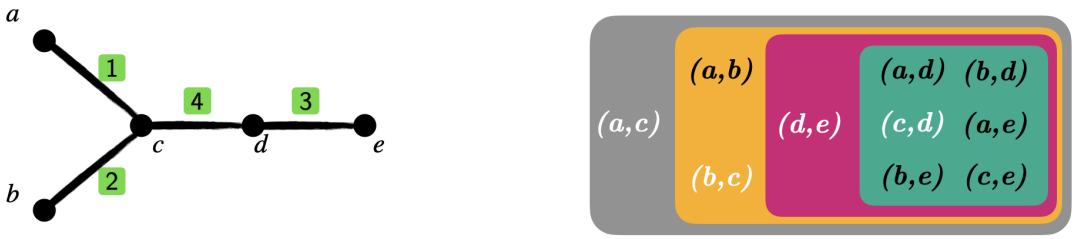

Figure 3: A tree on five vertices, regarded as belonging to an ambient K_5, will be used to illustrate the independence argument for the internal formula. We suppose the edges are added in the indicated order $(a, c), (b, c), (d, e), (c, d)$ (white text). On the right is a tiered diagram used in the argument, where the nested sets from the inside out are $\partial F_3 \subset \partial F_2 \subset \partial F_1 \subset \partial F_0$ for the partial forests F_j constructed in the course of Kruskal's algorithm.

These events are all independent of each other—for example, conditioning on (d, e) having lower weight than the six edges grouped to its right says nothing about the relative weights among those six. Thus the MST_0 probability of realizing T with this edge order is $\frac{1}{6} \cdot \frac{1}{7} \cdot \frac{1}{9} \cdot \frac{1}{10}$. To get the overall probability of this tree, we sum over all permutations on these 4 edges, performing a similar calculation.

Proof of Theorems 3.4 and 3.5. The proof of the general internal formula follows the identical structure to the example above; given a permutation of the tree edges, it is necessary and sufficient that each edge has the lowest weight in its corresponding ∂F_j. Each condition that e_j has lower weight than the ∂F_{j-1} is independent of the relative ordering of the ∂F_{j-1}. Theorem 3.4 has a similar structure, but using nested sets $D_1 \subset D_2 \subset \cdots \subset D_{m-n+1}$ instead of $\partial F_{n-2} \subset \cdots \subset \partial F_0$. □

Corollary 3.7 (Probability of stars)**.** *For any (labeled) spanning tree $T_\star$ of K_n that is a star,*

$$\mathbb{P}_{\mathrm{MST}}(T_\star) = \frac{1}{(2n-3)!!} \quad \text{while} \quad \mathbb{P}_{\mathrm{UST}}(T_\star) = \frac{1}{n^{n-2}}.$$

Proof. We apply the internal formula. Because of the symmetry in star graphs and complete graphs, it suffices to do the computation for one ordering of the edges in the star graph and multiply by $(n-1)!$. Once $j-1$ edges have been added from within the star by Kruskal's algorithm, the jth edge must be the lowest from $|\partial F_{j-1}| = \binom{n}{2} - \binom{j}{2}$ possibilities. (That is, ∂F_{j-1} is all edges of the K_n minus those between the j vertices touching the $j-1$ edges added already—these can be ignored because they've either been added already or are forbidden for completing a triangle.) To complete the calculation, we will use two easily verified facts. First, $\binom{n}{2} - \binom{j}{2} = \frac{(n+j-1)(n-j)}{2}$. Second, using double-factorial notation for the parity-restricted factorial $(2k-1)!! := (2k-1)(2k-3)\cdots 3\cdot 1$, we have the identity $(2k-1)!! = \frac{(2k-1)!}{2^{k-1}(k-1)!}$. With these, we get

$$\begin{aligned}
\mathbb{P}_{\mathrm{MST}}(T_\star) &= (n-1)! \prod_{j=1}^{n-1} \frac{1}{\binom{n}{2} - \binom{j}{2}} = (n-1)! \prod_{j=1}^{n-1} \frac{2}{(n+j-1)(n-j)} \\
&= \frac{(n-1)! \cdot 2^{n-1}}{[n(n+1)\cdots(2n-2)]\,[(n-1)(n-2)\cdots 1]} \\
&= \frac{2^{n-1}(n-1)!}{(2n-2)!} = \frac{2^{n-1}(n-1)!}{(2n-2)(2n-3)!} = \frac{2^{n-2}(n-2)!}{(2n-3)!} = \frac{1}{(2n-3)!!}.
\end{aligned}$$

□

Unfortunately, we have no similar closed-form expression for the probability of a given path. Nevertheless, we may compute these probabilities for small n using Theorem 3.5. Each probability can be thought of as the number of permutations on the $\binom{n}{2}$ edges that result in the given path being selected, divided by $\binom{n}{2}!$. Table 1 lists the first several probabilities.[2]

n	$m = \binom{n}{2}$	Numerator: A374293(n)	Denominator: $m!$
1	0	1	1
2	1	1	1
3	3	2	6
4	6	44	720
5	10	27120	3628800
6	15	882241920	1307674368000
7	21	2443792425984000	51090942171709440000

Table 1: Probabilities (as numerators and denominators) of sampling a labeled path from K_n under MST_0.

Replication code can be found in [27]. By $m = 14$, the UST probability of a path is more than three times its MST_0 probability. In the following sections, we will introduce several "rotation moves" that will allow us to derive probability inequalities between related trees. In §3.5 this will let us show rigorously what is suggested by the discussion in this section: a star is the most likely (labeled) spanning tree in a complete graph under ordinary MST, and a path is the least.

3.3 Triangle-edge rotation

We start with an edge rotation lemma that is motivated by the simple square-with-a-diagonal graph. It is not the most general rotation move we will employ, but it is useful and easy to explain. See Figure 4 for the accompanying illustration.

[2]The list of all numerators computed so far has been added to the Online Encyclopedia of Integer Sequences (OEIS) as entry A374293 [17]. We note an intriguing conjecture based on the match between this numerator sequence and a subsequence of the previously existing OEIS entry A253950: for n vertices and m edges, each numerator is equal to the number of ordered monoids of order $m-n+3$ with a neutral element that is also the top element. The pattern holds as far as the available terms allow us to compare, which is $n = 5$.

Lemma 3.8 (Triangle-edge rotation in general graphs)**.** *Suppose a graph G includes all three edges of a triangle on vertices v_1, v_2, v_3, denoted e_{12}, e_{23}, e_{13}. Suppose there are disjoint trees T_1, T_2, T_3 containing v_1, v_2, v_3, respectively, so that $S = T_1 \cup T_2 \cup T_3 \cup e_{12} \cup e_{23}$ and $S' = T_1 \cup T_2 \cup T_3 \cup e_{13} \cup e_{23}$ are spanning trees of G, related by an "edge rotation" based at v_1. Suppose the ambient graph G has at least one other edge between T_1 and T_2 and has no edges other than e_{13} between T_1 and T_3 (with no conditions on edges between T_2 and T_3). Then there is a strict inequality for ordinary MST:*

$$\mathbb{P}_{\mathrm{MST}_0}(S) > \mathbb{P}_{\mathrm{MST}_0}(S').$$

We postpone the proof of Lemma 3.8 to the end of this section.

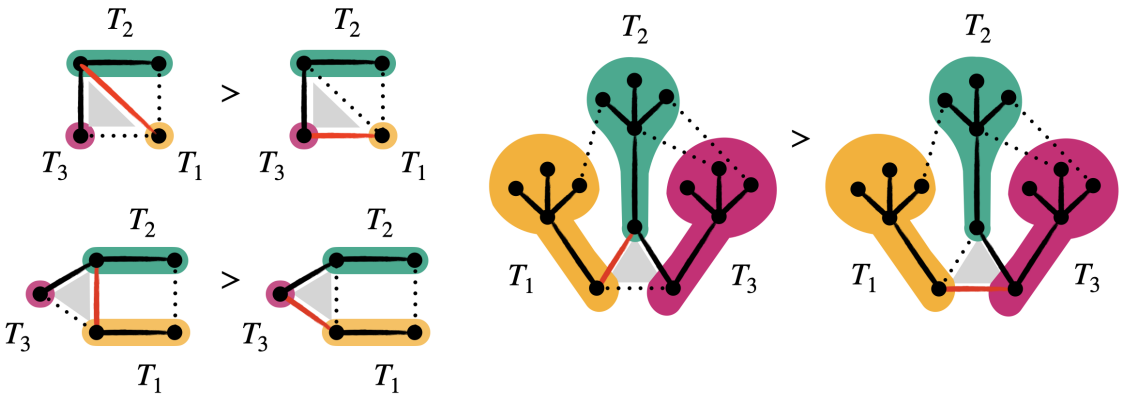

Figure 4: Three examples of triangle-edge rotation, where the ambient graphs are a square with a diagonal, a house, and a 17-edge graph. Edges in the graph but not included in the spanning trees are denoted with dotted lines. In each case, the three trees cited in the lemma are highlighted and the spanning trees S, S' differ only by the "rotation" of the red edge. In each case, the left-hand spanning tree S is strictly more likely than the right-hand spanning tree S' under ordinary MST.

Note that this triangle-edge rotation has limited usefulness for working in a complete graph, because T_1 and T_3 would have to be singletons in order to satisfy the hypothesis that there are no other edges between them than e_{13}. It is not applicable to square grid-graphs, for instance, which have no triangles. But it can be applied in many planar graphs and in random graphs, as we will see in the next section.

Recall from Proposition 2.3 that a set of weights w on $E(G)$ induces T via Kruskal's algorithm if and only if, for every $e \in T$ and $e' \notin T$, $e \, R_T \, e' \implies w(e) < w(e')$. To motivate the next definition, note that any two spanning trees of the same graph have the same number of edges $(n-1)$, and therefore the same number of non-edges $(m-n+1)$, which are in one-to-one correspondence with broken cycles. In a star graph in K_n, all broken cycles have length 3. In a path graph, they come in all lengths from 3 to n. This means in the latter, there are many more pairs e, e' in the cycle relation, which means many more weight inequalities to be satisfied. We leverage this by defining bijections of edges that increase broken-cycle lengths. The number of inequalities is twice the number of non-edges, or $n^2 - 3n - 2$, in a star, compared to $\frac{1}{6}(n^3 - 7n + 6)$ in a path. Of course there are various dependencies among these inequalities, so we will need controlled comparisons to prove probability bounds.

Definition 3.9 (Cycle-expanding)**.** Let T_1 and T_2 be two spanning trees of a graph G, with respective cycle relations R_1 and R_2. A bijection $\beta \colon E(G) \to E(G)$ is *cycle-expanding from T_1 to T_2* if $\beta(T_1) = T_2$ and $\beta(R_1) \subseteq R_2$, i.e., $e \, R_1 \, e' \implies \beta(e) \, R_2 \, \beta(e')$. Equivalently, this says that $\beta(P_{T_1,e'}) \subseteq P_{T_2,\beta(e')}$ for all $e' \notin T$—broken cycles can only get longer after applying β. The correspondence β is *strictly* cycle-expanding if the inclusion $\beta(R_1) \subsetneq R_2$ is strict, i.e., some cycles get strictly longer.

It is easy to confirm that the examples in Figure 4 are all cycle-expanding from left to right: the broken cycle lengths go from $(3, 3)$ to $(3, 4)$ for the square with a diagonal; from $(3, 4)$ to $(3, 5)$ for the house; and from $(3, 5, 6, 6)$ to $(3, 5, 6, 7)$ for the 18-edge graph. (Here the bijection exchanges e_{12} and e_{13}, fixing all other edges.)

Proposition 3.10 (Cycle expansion inequality)**.** *If there is a cycle-expanding bijection β from T to T', then the tree with the shorter cycles is more likely:*

$$\mathbb{P}_{\mathrm{MST}_0}(T) \geq \mathbb{P}_{\mathrm{MST}_0}(T').$$

Moreover, if β is strictly cycle-expanding, then the inequality is strict.

Proof. To show that T is at least as likely as T', it suffices to show that it is the minimum tree for at least as many draws of edge weights. So we will take an arbitrary set of distinct weights w that is carried by β to a permuted set of weights $\{w'_i\}_{i=1}^m$. Without loss of generality, we may order these weights so that $w'_1 < \cdots < w'_m$ and we will label the edges correspondingly as $e_1, \ldots, e_m$. By this convention, we have $w_i = w_{\beta^{-1}(i)}$ as a weight on e_i before β has been applied, noting that the w_i need not be in increasing order. We assume that the w' induce T' and we want to show that the w induce T.

Since β maps between T and T', any $e\, R_T\, e'$ is sent by β to some $\beta(e) = e_i$ and $\beta(e') = e_j$ with $e_i\, R_{T'}\, e_j$. The weights satisfy $w'_i < w'_j$ because T' is assumed minimum (appealing to Proposition 2.3). But then we are done: the weights on e and e' are precisely $w'_i < w'_j$ by construction, and we've verified that edges in T have lower weight than the non-edges that complete their broken cycle.

To handle the case of strict inequality, we will find a set of weights w_i that induces T, but so that $w_{\beta(i)}$ does not induce T'. Fix any edge $e_0 \notin T$ so that $\beta(P_{T,e_0}) \subsetneq P_{T',\beta(e_0)}$. Assign weights w_i so that all weights w_i for $e_i \in P_{T,e_0}$ are clustered near 1; the weight w_0 of our chosen edge e_0 is 2; the weights w_i for all other edges e_i of T are clustered near 3; and the weights w_i for non-edges of T are clustered near 4. Then these weights $\{w_i\}$ induce T, because Kruskal's algorithm will reject e_0, but $\beta(e_0)$ has some edge on the T' path completing its broken cycle that did not come from P, and therefore has weight more than 2. This means Kruskal will accept $\beta(e_0)$, so the weights $\{w_{\beta(i)}\}$ do not induce T'. □

Now the proof of the rotation trick is nearly immediate: we only need to show that the triangle-edge rotation is cycle-expanding.

Proof of Lemma 3.8. As before, β exchanges e_{13} and e_{23} and preserves all other edges. We must check that β is strictly cycle-expanding from S to S'. This is an easy case analysis on the non-edges $e \notin S$:

- if e is an edge of G with endpoints in a single tree T_i, then its broken cycle is within the tree before and after β and its length does not change;
- if $e = e_{13}$, its broken cycle stays at length 3;
- if e connects T_2 to T_3, its broken cycle contains e_{23} and its length is unchanged;
- if e connects T_1 to T_2, then in S' it must go the long way around the triangle rather than the short way, so its length is increased by one.

Note that there are no G-edges between T_1 and T_3 (a needed hypothesis, as these would have cycle lengths reduced by β). This completes the proof, because all cycles stay the same length or are increased by one. □

3.4 Application to random graphs

We can apply the rotation trick to derive a simple sufficient condition for MST_0 to fail to capture the uniform distribution on spanning trees of a given graph. As a consequence, for large n, we can conclude that MST_0 is different from UST on almost all graphs of order n. Recall the famous fact that $\log n/n$ is a sharp threshold for connectivity in the Erdős-Renyi random graph model $\mathcal{G}(n,p)$, where log is the natural logarithm.

Theorem 3.11 (Random graphs have $\mathrm{MST}_0 \neq \mathrm{UST}$)**.** *If $p = c \log n/n$ for any constant $c > 1$, the probability that $\mathrm{MST}_0 \neq \mathrm{UST}$ on a random $\mathcal{G}(n,p)$ graph tends to one as $n \to \infty$.*

Proof. We will use two main facts about random graphs in this probability regime: they almost surely have many triangles, and they are almost surely k-connected for every k. In particular, the threshold for almost sure k-connectivity is roughly $p = (\log n + k \log\log n)/n$, with detail about lower-order terms found in [5, §7.2]. The expected number of triangles is $\binom{n}{3}p^3$, which is on the order of $(\log n)^3$, and the variance is small

because many triangle events are independent. Consider a particular triangle T and one of its vertices v_1; the probability that v_1 has no edges to $G \setminus T$ is $(1-p)^{n-3}$, so it has at least one such edge with high probability. Putting these observations together, the probability that there exists a triangle T with v_1 connected to $G \setminus T$ tends to 1, which means the probability also tends to 1 of finding such a triangle in a 2-connected graph. But then we can apply the rotation trick (Lemma 3.8) with $T_1 = \{v_1\}$, $T_3 = \{v_3\}$, and T_2 any spanning tree of $G \setminus \{v_1, v_3\}$ (which is connected because G is 2-connected). □

3.5 MST on complete graphs via path rotation

To prove that labeled trees with the highest probability weight for MST_0 on K_n are stars and those with the lowest probability weight are paths, we will develop a more powerful rotation move. On one hand, these path rotations will be far more general than single-edge rotations, but the ambient graph must be a K_n (while Lemma 3.8 works in an arbitrary G).

3.5.1 Statement and main application

Theorem 3.12 (Path rotation in complete graphs)**.** *Let $T = L \cup P \cup R$ and $T' = L' \cup P' \cup R'$ be spanning trees of K_n obtained by rotating a path P of $\ell \geq 2$ vertices as depicted in Figure 5, where $L \cong L'$, $R \cong R'$, and both have at least one edge. Then*

$$\mathbb{P}_{\mathrm{MST}_0}(T) > \mathbb{P}_{\mathrm{MST}_0}(T').$$

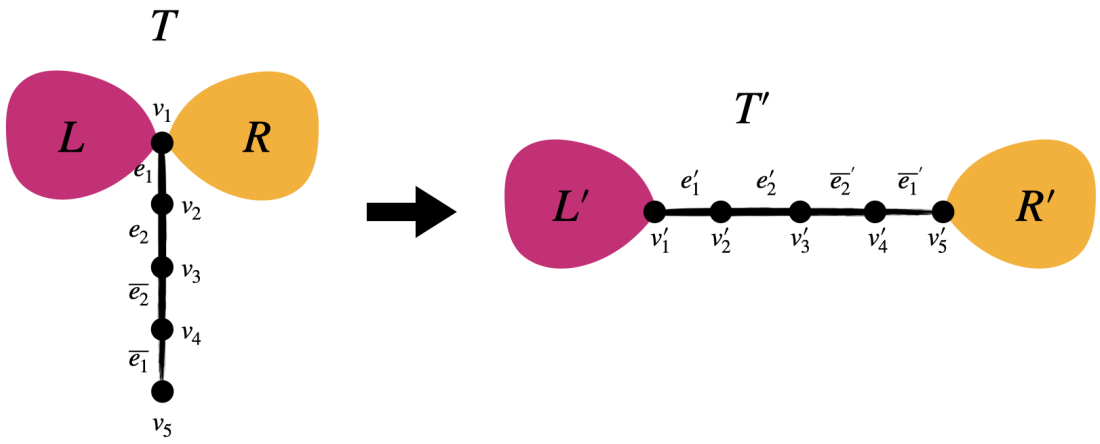

Figure 5: A path rotation operation from T to T' that rotates a path P from v_1 to v_5.

The path $P \subset T$ is on vertices $v_1, \ldots, v_\ell$. It will be useful for the later arguments in this section to label the edges of P so as to keep track of each edge and its (central) reflection. Namely, letting $r := \lfloor \ell/2 \rfloor$, we label the edges $e_1, \ldots, e_r, \overline{e_r}, \ldots, \overline{e_1}$, with $e_r = \overline{e_r}$ if ℓ is even. We will similarly use the notation $\overline{v_j} = v_{\ell-j+1}$ where it simplifies the exposition. Even though this path rotation move only applies in a K_n, we give two different names $G \cong G' \cong K_n$ to the ambient graph for clarity in the proof, and write $T, P \subset G$ and $T', P' \subset G'$, fixing this notation for the remainder of the section. Note that we do not consider v_1 to be part of L or R or v_1', v_ℓ' to be part of L' or R'; we regard v_1 as part of P and v_1', v_ℓ' as part of P'.

The proof of Theorem 3.12 requires new machinery and is postponed to the end of this section. First, we derive the main application.

Theorem 3.13 (Most and least likely trees)**.** *In a complete graph K_n with labeled vertices, let $T_\star$ and T_P be any spanning trees isomorphic to a star and a path, respectively. Then for all spanning trees T of K_n,*

$$\mathbb{P}_{\mathrm{MST}_0}(T_\star) \geq \mathbb{P}_{\mathrm{MST}_0}(T) \geq \mathbb{P}_{\mathrm{MST}_0}(T_P),$$

with equality on either side if and only if T is itself a star or path.

Proof. For any graph G, let $D(G)$ be the product of all the vertex degrees. Stars that span K_n have $D = n-1$ while spanning paths have $D = 2^{n-2}$. We will show that these are the minimum and maximum among trees, and that there are sequences of path rotation operations interpolating from stars to arbitrary trees and from arbitrary trees to paths, with each step increasing D and decreasing the probability of selection under MST_0.

If T is not a star, then it contains some path of length 3, in which each vertex has some (possibly empty) tree attached (Figure 6). This means it comes from a (reverse) path rotation from another tree involving a single edge. All the vertex degrees are the same in the two trees except at the central vertices, which changed from (a, b) to $(a + b - 1, 1)$ with $a, b \geq 2$. Since $ab > a + b - 1$ for $a, b \geq 2$, this reverse rotation operation decreases D.

If T is not a path, then it contains some vertex of degree ≥ 3, one of whose incident edges is part of a simple path ending in a leaf (Figure 6). Since its degree is at least three, its other incident edges can be separated into two non-empty trees. Rotating the path induces a graph for which P' has endpoints with vertex degrees (a, b). This means that on the original graph the corresponding vertices had degree $(a + b - 1, 1)$. The same inequality holds, so the path rotation operation increases D.

Since any non-star can have D strictly decreased while any non-path can have D strictly increased, it follows that stars and paths are extremal for D. The path rotation theorem tell us that moves that increase D decrease the MST_0 probability and vice versa, and it follows that stars and paths are also extremal for the [illegible] □

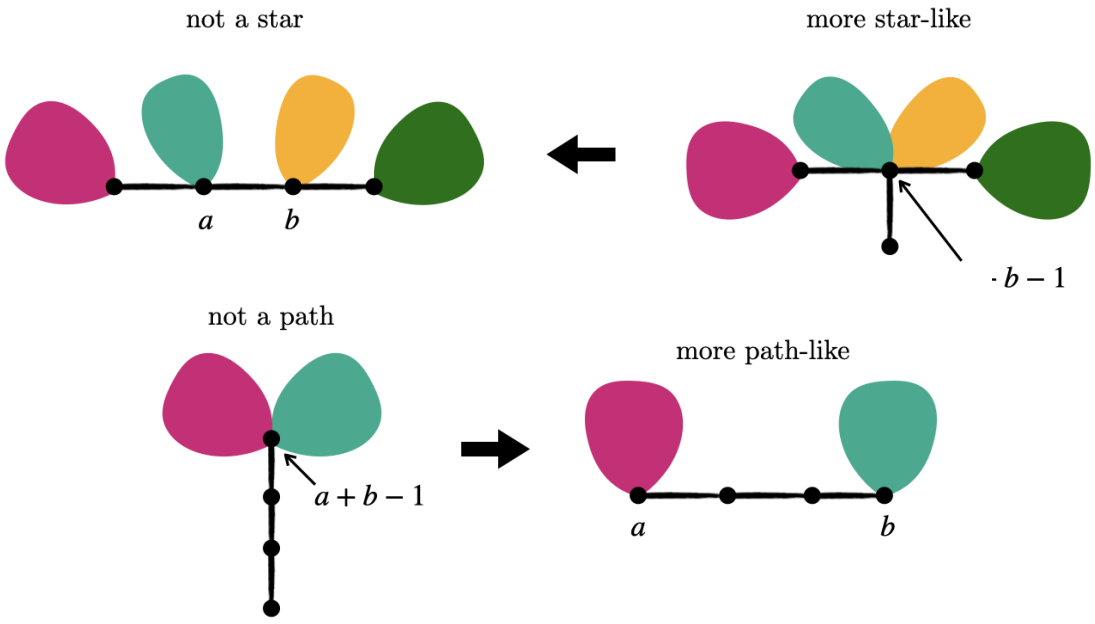

Figure 6: Moves that interpolate from arbitrary trees to stars and paths, strictly monotonic with respect to both MST_0 probability and the product D of vertex degrees.

From the proof we see that the passage from stars to arbitrary trees can be accomplished with only single-edge rotations. However, to get from an arbitrary tree to a path, longer path rotations may be required, which in turn requires a significant adaptation to the previous proof approach of constructing a cycle-expanding bijection.[3] Indeed, we have verified computationally that there is no cycle-expanding bijection (as in §3.3) from a tree with three paths of length 2 joined at a common endpoint to the path of length 6.

[3] To see what goes wrong with this approach, consider the edges in Figure 5. We could try to choose an edge to pair with $\overline{e_1}$, but choosing either e_1' or $\overline{e_1}'$ fundamentally breaks symmetry, which will cause trouble later on. For instance, suppose we choose to pair up $\overline{e_1}$ with $\overline{e_1}'$. Let $w \in R$ be a neighbor of v_1 in T, and consider the permutation where the first three edges are $\overline{e_1}$, (v_1, w), and (v_4, w). Note that this final edge is not in T, but it will still be added as it does not create a cycle. Consider the three corresponding edges in G'. We are already assuming that $\overline{e_1}$ will be sent to $\overline{e_1}' = (v_4', v_5')$. The most natural choices for the other edges are to send (v_1, w) to (v_5', w') (since (v_1, w) is an edge in R, so it should be sent to the corresponding edge in R') and (v_4, w) to (v_4', w'). These three edges *do* form a cycle, so the final edge will be rejected. Thus, we reach a state where T is no longer possible while T' is, which is precisely what we must rule out.

3.5.2 Folding a path

We introduce a new proof technique that records partial (probabilistic) information about the randomly generated edge weights. We will define an equivalence relation on weight-orderings (or permutations) σ, and a bijection β on the resulting equivalence classes, so that every equivalence class ρ induces T with probability greater than or equal to the probability than $\beta(\rho)$ induces T' (with a strict inequality for at least one ρ). To illustrate this approach with reference to Figure 5: we associate the equivalence class $\{e_1, \overline{e_1}\}$ with the equivalence class $\{e_1', \overline{e_1}'\}$ and remain agnostic to which of the two edges was actually processed first, allowing us to maintain symmetry. More generally, we will introduce several "folded" concepts here; the term refers to identifying each e_i with its reflection $\overline{e_i}$ and e_i' with $\overline{e_i}'$ in a manner that suggests folding the path in half.

Definition 3.14 (Folded permutations)**.** Using the notation introduced right after Theorem 3.12, let ι be the involution on $E(G)$ that swaps each edge of the form (v_i, v_j) with its reflection $(\overline{v_i}, \overline{v_j})$ and is otherwise the identity on $E(G)$. Likewise, ι' is defined on $E(G')$. Consider a permutation of the edges $\sigma \in S(E(G))$, thought of as a map $\sigma\colon \{1, \dots, m\} \to E(G)$. We write $E(G)/\iota$ for the set of equivalence classes under ι. Define the corresponding *folded permutation* $\sigma/\iota \in S(E(G))/\iota$ to be

$$\rho = \sigma/\iota\colon \{1, \dots, m\} \to E(G)/\iota.$$

This is no longer a bijection. The equivalence classes of edges are denoted by $[e]$ and have one or two representatives. Indeed, those with one representative are exactly the r equivalence classes for the edges of the form $(v_i, \overline{v_i})$ which are fixed by the central symmetry of the path. This shows that there are $a = \frac{1}{2}\left(\binom{\ell}{2} - r\right)$ different equivalence classes of size 2 that arise from the complete graph on the vertices of the path, and thus 2^a representatives for every folded permutation.

There is a corresponding *folded bijection* $\beta\colon E(G)/\iota \to E(G')/\iota$ that takes each $[(u, v)]$ to $[(u', v')]$ except that $[(u, v_i)] \mapsto [(u, \overline{v_i})]$ if $u \in R$ and $v_i \in P$. Looking ahead, this can be seen in Figure 7.[4] This means that from a folded permutation ρ on G we obtain a corresponding folded permutation $\rho' = \beta(\rho)$ on G'.

If we run Kruskal's algorithm on a folded permutation ρ, the outcome is no longer a single tree, but a distribution over trees—the first time we see a particular equivalent pair of edges, we choose one of them uniformly at random. The next time, we take the other. Write $\mathbb{P}_\rho(T)$ for the probability that ρ induces T, and similarly for $\mathbb{P}_{\rho'}(T')$.

With this setup, the key fact to prove Theorem 3.12 is that, for any folded permutation $\rho \in S(E(G))/\iota$ and corresponding $\rho' \coloneqq \beta \circ \rho$, we have a (weak) probability inequality $\mathbb{P}_\rho(T) \geq \mathbb{P}_{\rho'}(T')$. We actually prove a stronger statement, about the probability distribution of forests at each stage of Kruskal's algorithm, in addition to showing that the inequality is sometimes strict. To state it precisely, we build up some more terminology.

3.5.3 State vector

Fixing ρ and k, consider the probability space where we uniformly sample a random permutation σ (the order of edges in a run of Kruskal's algorithm) but condition on σ being consistent with ρ on the first k labels $\{1, \dots, k\}$, meaning that $[\sigma(i)] = \rho(i)$ for $i = 1, \dots, k$. Let $\mathrm{Kru}_k(\rho)$ be the forest that Kruskal's algorithm outputs after processing the lowest-weight k edges of σ, thought of as a forest-valued random variable. Define $\mathcal{E}_k$ to be the event that $\mathrm{Kru}_k(\rho) \subseteq T$, respectively $\mathcal{E}_k'$.

We prove the probability inequality by induction on k. To make the induction go through, we add additional hypotheses tracking the distribution of $\mathrm{Kru}_k(\rho)$ conditioned on $\mathcal{E}_k$, and of $\mathrm{Kru}_k(\rho')$ conditioned on $\mathcal{E}_k'$. These distributions are simultaneously encoded by a state vector s that records the status of the edges $e_1, \dots, e_r$ as the steps of Kruskal's algorithm are incremented. For each edge e_i and its reflection $\overline{e_i}$, it will tell us what we know about their inclusion in the tree T to that point. Formally,

$$s\colon \{1, \cdots, r\} \to \{\texttt{Neither}, \texttt{Both}, \texttt{Left}, \texttt{Right}\} \cup \mathcal{P}(\{1, \cdots, r\}),$$

where $\mathcal{P}(\{1, \cdots, r\})$ is the power set of $\{1, \cdots, r\}$.

[4] For example, the edges numbered 3 in G' form an equivalence class of size 2 and are mapped to edges $\{(v_1, v_3), (v_3, v_5)\}$ in G. On the other hand, the edge numbered 6 in G' is in an equivalence class of size 1, and is mapped to the edge in G joining the leftmost vertex of T to v_2.

Algorithm 3.1: Computes the probabilities of sampling a pair of spanning trees differing by a path rotation.

Input: A pair of spanning trees $T = L \cup P \cup R$ and $T' = L' \cup P' \cup R'$ as in Figure 5.
Output: A pair of probabilities: The respective likelihoods of sampling T and T' under MST_0.

$(p, p') \leftarrow (0, 0)$;
for $\rho \in \{\text{folded permutations}\}$ **do**
  $(q, q') \leftarrow (1, 1)$;
  $s \leftarrow (i \mapsto \texttt{Neither})$;
  for $k \in \{1, 2, \ldots, m\}$ **do**
    if $\rho(k) = [e_{\ell/2}] = \{e_{\ell/2}\}$ **then**
      $s(\ell/2) \leftarrow \texttt{Both}$;
    else if $\rho(k) = [e_i] = \{e_i, \overline{e_i}\}$ *where* $i < \frac{\ell}{2}$ **then**
      if *this is the first time we have seen this* $\rho(k)$ **then**
        $s(i) \leftarrow \{i\}$;
      else
        $s(i) \leftarrow \texttt{Both}$;
        remove all occurrences of i from sets in s;
    else if $\rho(k) = \{(u, v)\}$ for $(u, v) \notin T$ **then**
      if *the path from* u *to* v *in* T *contains an edge* e *such that* $[e] \notin \{\rho(1), \rho(2), \ldots, \rho(k-1)\}$ **then**
        $(q, q') \leftarrow (0, 0)$;
      else if $u \in V(L)$ *and* $v \in V(R)$, *or* $u \in V(R)$ *and* $v \in V(L)$ **then**
        if $s \neq [\texttt{Both}, \texttt{Both}, \ldots, \texttt{Both}]$ **then**
          $q' \leftarrow 0$;
      else if *one vertex from* $\{u, v\}$ *(without loss of generality* u*) is in* $V(L) \cup V(R)$ *and the last edge of the path from* u *to* v *in* T *is some* e_i *or* $\overline{e_i}$ **then**
        $\texttt{side} \leftarrow (\texttt{Left}$ if $\texttt{u} \in \texttt{V(L)}$, else $\texttt{Right})$;
        if *the last edge was* $\overline{e_i}$ *and there is some* $j \geq i$ *with* $s(j) \neq \texttt{Both}$ **then**
          $(q, q') \leftarrow (0, 0)$;
        $M \leftarrow \{s(j) \mid j \in \{1, 2, \ldots, i\},\ s(j) \text{ is a set}\}$;
        $N \leftarrow \{s(j) \mid j \in \{1, 2, \ldots, i\},\ s(j) \in \{\texttt{Left}, \texttt{Right}, \texttt{Neither}\}\}$;
        $(q, q') \leftarrow (q/2^{|M|}, q'/2^{|M|})$;
        if $N \not\subseteq \{\texttt{side}\}$ **then**
          $q' \leftarrow 0$;
        for $j \in \bigcup M$ **do**
          $s(j) \leftarrow \texttt{side}$;
        end
    else if $\rho(k) = [(v_{i_1}, v_{i_2})]$ *for* $1 \leq i_1 < i_2 \leq \ell$ *where* v_{i_1} *is the lower-indexed vertex of some* e_{I_1} *or* $\overline{e_{I_1}}$ *and* v_{i_2} *is the higher-indexed vertex of some* e_{I_2} *or* $\overline{e_{I_2}}$ **then**
      $M \leftarrow \{s(j) \mid \min(I_1, I_2) \leq j \leq \max(I_1, I_2),\ s(j) \text{ is a set}\}$;
      $N \leftarrow \{s(j) \mid \min(I_1, I_2) \leq j \leq \max(I_1, I_2),\ s(j) \in \{\texttt{Left}, \texttt{Right}, \texttt{Neither}\}\}$;
      if $i_1 \leq \frac{\ell}{2} \leq i_2$ *and* $s(j) \neq \texttt{Both}$ *for some* $j \geq I_1, I_2$ **then**
        $(q, q') \leftarrow (0, 0)$;
      else
        $(q, q') \leftarrow (q/2^{|M|}, q'/2^{|M|})$;
      if $N \neq \emptyset$ **then**
        if $\texttt{Neither} \in N$ *or* $\{\texttt{Left}, \texttt{Right}\} \subseteq N$ *or this is the second time we have seen* $\rho(k)$ **then**
          $(q, q') \leftarrow (0, 0)$;
        else
          $(q, q') \leftarrow (q/2, q'/2)$;
          for $j \in \bigcup M$ **do**
            $s(j) \leftarrow$ the unique element of N;
          end
      else
        for $j \in \bigcup M$ **do**
          $s(j) \leftarrow \bigcup M$;
        end
  end
  $(p, p') \leftarrow (p + q, p' + q')$;
end
return $(\frac{2^a}{m!}p, \frac{2^a}{m!}p')$ where $a = \frac{1}{2}\left(\binom{\ell}{2} - r\right)$;

The meaning of this state vector will be illustrated in several examples, but we first give a formal description via Algorithm 3.1 and Lemma 3.15. The algorithm describes how to successively update entries and track the probabilities of $\mathcal{E}_k$ and $\mathcal{E}'_k$, and the lemma proves it is correct. Algorithm 3.1 is written in the standard computer science style called *imperative programming*, with $x \leftarrow v$ denoting an update of variable x to hold value v, so that for instance $x \leftarrow x + 1$ means to increment by one. We initialize the probabilities at $(1, 1)$ and then track through a series of steps.

Lemma 3.15 (State vector and probability updates)**.** *Fix any folded permutation ρ and index $0 \le k \le m$. Consider the probability space where we uniformly sample a random permutation σ, but condition on σ being consistent with ρ on $\{1, \dots, k\}$. Then, after $\rho(k)$ has been processed in Algorithm 3.1 (i.e., after k iterations of the inner for-loop on Line 5), the values of the variables q and q' equal the respective probabilities of $\mathcal{E}_k$ and $\mathcal{E}'_k$. Furthermore, conditioned on $\mathcal{E}_k$ and $\mathcal{E}'_k$, respectively, the data in the state vector s describes the distribution of $\mathrm{Kru}_k(\rho) \cap E(T)$ and $\mathrm{Kru}_k(\rho') \cap E(T')$ as follows:*

- *For i such that $s(i) =$ `Neither`, neither edges e_i nor $\overline{e_i}$ have been added to $\mathrm{Kru}_k(\rho)$, and neither edges e'_i nor $\overline{e'_i}$ have been added to $\mathrm{Kru}_k(\rho')$.*
- *For i such that $s(i) =$ `Both`, both edges e_i and $\overline{e_i}$ have been added to $\mathrm{Kru}_k(\rho)$, and both edges e'_i and $\overline{e'_i}$ have been added to $\mathrm{Kru}_k(\rho')$.*
- *For i such that $s(i) =$ `Left`, edge e_i has been added to $\mathrm{Kru}_k(\rho)$, but not edge $\overline{e_i}$, and likewise edge e'_i has been added to $\mathrm{Kru}_k(\rho')$, but not edge $\overline{e'_i}$.*
- *For i such that $s(i) =$ `Right`, edge e_i has been added to $\mathrm{Kru}_k(\rho)$, but not edge $\overline{e_i}$, while on the other hand edge $\overline{e'_i}$ has been added to $\mathrm{Kru}_k(\rho')$, but not edge e'_i.*
- *For i such that $s(i) = S \subseteq \{1, 2, \dots, r\}$, there is a $\frac{1}{2}$ probability that all e_j for $j \in S$ have been added to $\mathrm{Kru}_k(\rho)$ but no $\overline{e_j}$ has. Otherwise, all $\overline{e_j}$ have been added to $\mathrm{Kru}_k(\rho)$ but no e_j has. Likewise, there is a $\frac{1}{2}$ probability of all e'_j but no $\overline{e_j}'$ (else all $\overline{e_j}'$ but no e'_j) were added to $\mathrm{Kru}_k(\rho')$.*

Before proving the lemma, we first illustrate the encoding and updating with an example. To understand the meaning of the entries of s, consider an example with $\ell = 14$ (so $r = 7$), where we might encounter the following state partway through the process: $s = (\texttt{Right}, \{2\}, \texttt{Neither}, \{4, 5\}, \{4, 5\}, \{6\}, \texttt{Both})$. (See Appendix B for a visualization.) This indicates that up to that point in the process, $e_1 \in T$ and $\overline{e_1}' \in T'$ rather than their reflections (i.e., the right-hand choice has been made in T'); exactly one of $\{e_2, \overline{e_2}\}$ is included and the probabilities are equal and independent of other indeterminacies to this point; neither e_3 nor $\overline{e_3}$ has been processed yet; $\{e_4, \overline{e_4}\}$ and $\{e_5, \overline{e_5}\}$ are both uncertain, with $1/2$ probability of including both e_4 and e_5 and $1/2$ of including their reflections instead and likewise e'_4, e'_5; inclusion of $\{e_6, \overline{e_6}\}$, like $\{e_2, \overline{e_2}\}$, is 50-50 with no dependencies; and the self-paired edge e_7 is included (i.e., "both" e_7 and its reflection). Whenever some $s(j)$ is a subset of $\{1, \cdots, r\}$, $s(j)$ must include j, and when its size is greater than one, as with the $\{4, 5\}$ entry here, we think of this as an *entanglement* of probabilities. In Appendix B we give an extensive discussion showing all of the ways that this state vector could be updated, which is complex enough to illustrate essentially all of the cases in Algorithm 3.1. Here, we present a simpler example with $\ell = 5$.

Example 3.16. We work through the example in Figure 7. The k values index the weight order encoded by the folded permutations ρ, ρ'. After $k = 1, 2$, we have definitely added one from each pair and nothing outside the tree, so $q = q' = 1$ give the probability that T, T' are still viable at that stage. After $k = 3$, we note two possibilities to continue: on one hand, perhaps both of the previously processed edges are chosen to the left in T' (probability $1/4$), in which case the new edge must be chosen left as well in order to be rejected. The right side is similar, giving total probability $q' = 1/4$. The situation in T is analogous, so $q = 1/4$ as well. The $k = 4, 5$ edges are added without changing s, q, q'. At $k = 6$, the only way for this edge to be rejected is if $\rho'(1)$ was on the left, which forces $\rho'(2)$ on the left as well and drops the overall probability by a factor of 2 (so $q = q' = 1/8$). Once the $k = 7$ edge is added, the $k = 8$ edge is definitely rejected in G because the other three edges on its broken cycle are already assumed present—that means that T is still possible. However, this edge is accepted in G', which kills the probability of Kruskal selecting T' (so $q = 1/8$ but $q' = 0$).

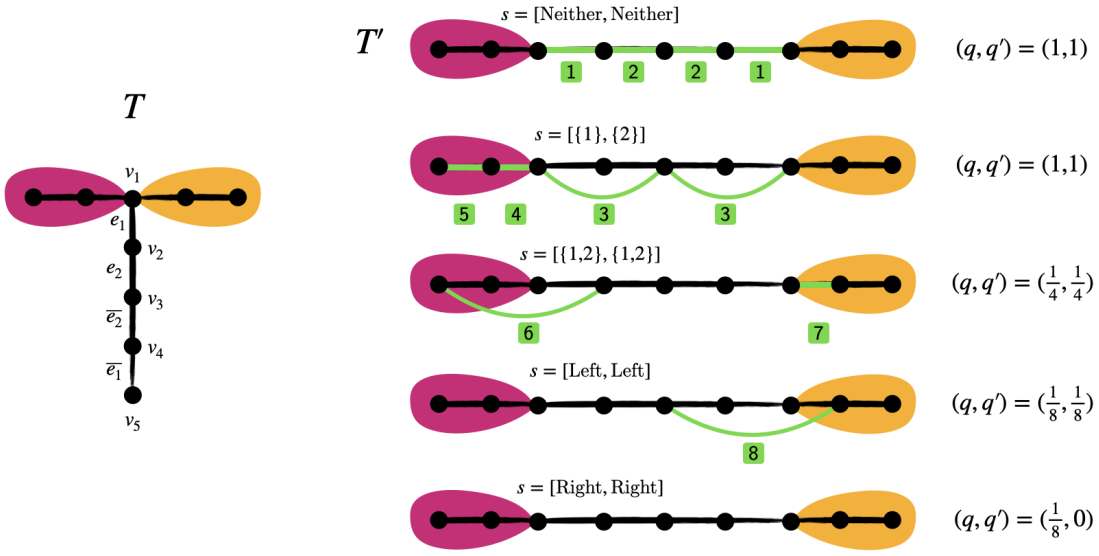

Figure 7: A simple example of processing a folded permutation; the green numbering on folded edges in T' shows the ordering from ρ' up to $k = 8$. After these eight edges have been processed, there is a $1/8$ probability that T can be chosen by Kruskal's, while T' has been ruled out.

Note the asymmetry between the meanings of the bar notations in T versus T'. In T, the e_i edges are always closer than the $\overline{e_i}$ edges to both the L and R components of the tree, while in T', the e_i' edges are closer to L and the $\overline{e_i}'$ edges are closer to R. This induces a crucial asymmetry in the statement of Lemma 3.15. In the cases where $s(i) \in \{\texttt{Left}, \texttt{Right}\}$, we can sometimes end up with one demand on the G side and two demands on the G' side, as in Example 3.16. This can lead to updating to $q' = 0$ while $q \neq 0$ (in Line 28), which gives the strict inequality we need.

3.5.4 Proof of algorithm correctness, proof of path rotation theorem

Proof of Lemma 3.15. Clearly the stated properties hold at initialization, i.e. when $k = 0$; the lemma is proved by inductively checking that q, q', and s are updated correctly in all of the cases of Algorithm 3.1. Many cases require no explanation. We work through several cases here to illustrate the trickier steps. Illustrations of these cases and more are given in Figure 11 and Table 3 in Appendix B.

First consider Line 9, processing a pair of edges inside the central path for the first time. This does not change the probabilities q or q', because those are concerned with the likelihood of Kruskal's algorithm adding any edges *outside* of T, so (q, q') are still accurate. Furthermore, it is equally likely that a uniformly random permutation σ consistent with ρ on $\{1, \ldots, k\}$ would have added edge e_i versus $\overline{e_i}$ to T, and likewise for $\beta(\sigma)$ with respect to $e_i', \overline{e_i}'$ in T'. The algorithm sets $s(i) = \{i\}$, and this checks out with the claimed meaning of $s(i) = S$. All other entries of s remain unchanged at this stage, and there is no new information about the likelihoods of various edges having been added, and thus the accuracy of s is maintained.

Next, consider the case on Line 20, where we process an edge $e \notin T$ and corresponding edge $e' \notin T'$ that both join the central path to one of the sides. For simplicity, we consider the case $i < \frac{\ell}{2}$. To update q and q', the algorithm computes the probability that all of the edges related to e by the cycle relation R_T have been added to T, and analogously, the probability that all of the edges related to e' by the cycle relation $R_{T'}$ have been added to T'. By the inductive hypothesis, the `Left` and `Right` designations in s specify knowledge that, in each tree, we have added a specific one of a given pair of edges. In T, this will always be the edge of lower index, so this will always be the edge in the cycle relation since $i \leq \frac{\ell}{2}$. However, for T', we may be on the wrong side; Line 27 catches this possibility and accordingly sets q' to zero. Otherwise, the probabilities in T and T' that all edges in the cycle relation with the new edge have been added already is just given by $\frac{1}{2}$ raised to the power of the number of independent binary events involving those edges

existing. By the inductive hypothesis, these can be calculated by taking the number of sets in s involved along the respective paths, which is precisely the size of the variable set M. Thus, q and q' are updated correctly in Line 26, maintaining their accuracy. To see that s remains correct as well, observe that the for-loop on Line 29 correctly updates the conditional probability distributions with the following fact: if the new edge was rejected, it must be that all of the edges in the cycle relation (or correlated with them through non-singleton sets in s) are on the side of the path from which the new edge came.

As a third case, we consider Line 48, which is where non-singleton sets in s may originate when the sets in M are merged. Here we have processed a pair of edges that are not in the respective trees T and T', with endpoints in the paths P and P'. The logic for why (q, q') are still accurate is similar to the previous case. To see that s is valid, observe that, conditioned on the new edge being rejected, all of the once-indeterminate pairs of edges must have now been resolved, and all to the same side (forming a cycle with the new edge), even though it is equally likely which side it was.

This kind of case analysis establishes that Algorithm 3.1 correctly processes each edge $\rho(k)$. At the final iteration $k = m$, we conclude that $q = \mathbb{P}_\rho(T)$ and $q' = \mathbb{P}_{\rho'}(T')$ for each folded permutation ρ. The final lines sum these over the folded permutations and divide by the number of folded permutations $m!/2^a$, where $a = \frac{1}{2}\left(\binom{\ell}{2} - r\right)$ as in the discussion of Definition 3.14. This amounts to averaging over all folded permutations, so we have correctly computed the probabilities $\mathbb{P}_{\mathrm{MST}_0}(T)$ and $\mathbb{P}_{\mathrm{MST}_0}(T')$. □

With this, the proof of the path rotation theorem is nearly immediate.

Proof of Theorem 3.12. To see that $\mathbb{P}_{\mathrm{MST}_0}(T) > \mathbb{P}_{\mathrm{MST}_0}(T')$, we first observe that $\mathbb{P}_\rho(T) \geq \mathbb{P}_{\rho'}(T')$ for any folded permutation ρ. This is because, whenever q is decreased in Algorithm 3.1, q' is decreased by the same factor. Furthermore, there are several cases where q' is set to zero and q is not, in which case we can get a strict inequality. Specifically, consider a folded permutation where we first add all edges in $T \setminus P$, then an edge (u, v) where $u \in L$ and $v \in R$, then the remaining edges in T. This sets $q' = 0$ on Line 19, but we will still have $q = 1$ after all k iterations. Since the overall calculation is an average over cases, this shows a strict probability inequality overall. □

4 Shifted intervals

4.1 Parametrizing shifts

Let G be a graph with m edges labeled $e_1, e_2, \ldots, e_m$. For any $\mathsf{s} = (s_1, s_2, \ldots, s_m) \in \mathbb{R}^m$, we consider the product measure where the weight on edge i is drawn uniformly from $[s_i, s_i + 1]$, denoting the induced distribution on spanning trees by $M_\mathsf{s} = M_\mathsf{s}(G) \in \Delta(ST(G))$, so that $M_0 = \mathrm{MST}_0$. In this section we seek to understand what measures on spanning trees are achievable by this *shifted-interval MST* as the s_i vary.

Consider the map $h\colon \mathbb{R}^m \to \Delta(ST(G))$ given by $\mathsf{s} \mapsto M_\mathsf{s}$. This map is highly non-injective; for instance translating each interval by the same amount has no effect on the order of the random variables, so no effect on the distribution on trees. Likewise, if the interval for one variable is entirely above the interval for a second variable, then the first is always greater than the second, no matter how large the gap. In light of this it suffices to restrict attention to those shifts s where the gaps between s_i on the number line are no greater than 1 and the sum is fixed. We can formalize this constructively.

Lemma 4.1 (Closing gaps)**.** *For every $\mathsf{s} \in \mathbb{R}^m$, there exists $\mathsf{s}' \in \mathbb{R}^m$ such that $M_\mathsf{s} = M_{\mathsf{s}'}$ and $\bigcup_{i=1}^m [s'_i, s'_i + 1]$ is connected.*

Proof. Given m real numbers $s_1, \ldots, s_m$, let σ be a permutation that sorts them into non-decreasing order, so that $r_1 \leq \cdots \leq r_m$ where $r_{\sigma(i)} = s_i$. Now consider the first i for which $r_{i+1} \geq r_i + 1$. Then the weight drawn from the $i+1$st interval must be greater than the weight drawn from the ith interval. Replacing r_{i+1} with $r_i + 1$ maintains this order. Let $t = r_i + 1 - r_{i+1} < 0$ and shift $(r_1, \ldots r_m)$ by $(0, \ldots, 0, t, t, \ldots, t)$, where the negative shift occurs in every position from $i+1$ to m. It is still the case that these r'_i are non-decreasing, and relative orders for any pair of weights are preserved whether both are below i, both are $\geq i$, or one is on each side. Thus the $s'_i = r'_{\sigma(i)}$ are guaranteed to be in the same order as before. This update can be iterated from $i = 1, \ldots, m$, ensuring that there are no gaps greater than 1 between successive r_i while maintaining the order of the variables drawn from the intervals described by s. □

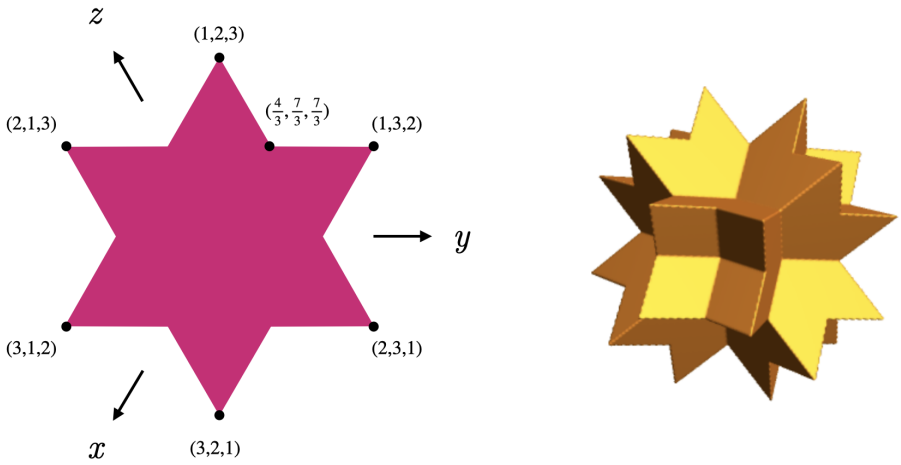

Figure 8: The shiftahedra Sh(3) and Sh(4).

Definition 4.2 (Shiftahedron)**.** For fixed m, let

$$R = \left\{(r_1, \ldots, r_m) \in \mathbb{R}^m : \sum r_i = \tbinom{m}{2} \text{ and } r_i \le r_{i+1} \le r_i + 1 \text{ for each } i\right\}.$$

Then the *shiftahedron* for m variables is defined as the orbit of R under reordering indices:

$$\mathrm{Sh} = \mathrm{Sh}(m) := S_m.R = \left\{\left(r_{\sigma(1)}, \ldots, r_{\sigma(m)}\right) : (r_1, \ldots, r_m) \in R, \sigma \in S_m\right\}.$$

A few basic properties are easily verified. Because it is a subset of the closed simplex $\sum s_i = \binom{m}{2}$ defined by non-strict inequalities, $\mathrm{Sh}(m)$ is closed and compact. It contains the extreme points $S_m.(1, 2, \ldots, m)$, and is star-shaped and symmetric with respect to the center point $\mathsf{s}_0 = (\frac{m+1}{2}, \ldots, \frac{m+1}{2})$. (Here, symmetry means that $\mathsf{s} \in \mathrm{Sh} \iff 2\mathsf{s}_0 - \mathsf{s} \in \mathrm{Sh}$.)

Proposition 4.3 (Shiftahedron parametrizes all shifts)**.** *The shiftahedron parametrizes shifted-interval MST:* $h(\mathrm{Sh}(m)) = h(\mathbb{R}^m)$.

Proof. The closing gaps lemma (Lemma 4.1) implies that $h(\mathrm{Sh}(m)) = h(\mathbb{R}^m)$ because it replaces any s with an s' giving the same distribution on spanning trees such that, after sorting, successive values differ by no more than 1. Translation suffices to fix the sum of the values, placing them in $\mathrm{Sh}(m)$. □

We note that $h\colon \mathrm{Sh}(m) \to \Delta(ST(G))$ is far from surjective, and indeed we will see below that even the much larger locus hit by all product measures does not hit every distribution on spanning trees. Furthermore, h is far from injective on the boundary of $\mathrm{Sh}(m)$. The failure of injectivity on the interior can depend on the connection topology of the graph G. In Appendix C, we study this map more closely in some special cases, and thereby confirm that shifts are enough to hit UST on a class of graphs called *theta graphs* and on a slightly generalized class we call *θ-surgery graphs*.

4.2 Shifts and complete graphs

In contrast to the simple case of theta graphs, we show that shifted-interval MST does not suffice to recover the uniform distribution on spanning trees in a complete graph K_n for $n \ge 4$. We first observe that $\mathrm{MST}_0 \ne \mathrm{UST}$ for $n \ge 4$ and $M_{\mathsf{s}} \ne \mathrm{UST}$ if the intervals $[s_i, s_i + 1]$ do not all intersect.

Lemma 4.4 (Conditions implying non-uniformity)**.** *For any graph G and any two distinct, non-separating edges e_j and e_k whose weights are given by random variables X_j and X_k, any distribution $D \in \Delta(ST(G))$ satisfying $\mathbb{P}_D(X_j < X_k) = 1$ is not uniform: $D \ne \mathrm{UST}$. In particular, for $G = K_n$, if $\bigcap_{i=1}^m [s_i, s_i + 1] = \emptyset$ then $M_{\mathsf{s}} \ne \mathrm{UST}$.*

Proof. The first part is easily seen by selecting any spanning tree T containing e_k but not e_j, which must exist if e_j is non-separating. Since this tree can't be obtained by MST selection using D weights, the distribution does not weight all spanning trees equally. The same observation carries over to the second part, letting j be the index for the smallest coordinate of s and k for the largest. The fact that not all intervals overlap implies that these intervals do not overlap, so the kth weight is guaranteed to be larger than the jth. □

Next we observe a monotonicity in probabilities as we shift a single weight.

Lemma 4.5 (Monotonicity under interval sliding)**.** *Given any $\mathsf{s} \in \mathrm{Sh}(m)$, let $\mathsf{s}(k,t)$ have all the same coordinates except for a slide in the kth coordinate: s_k is replaced by s_k+t. Then for all $j \neq k$, $\mathbb{P}_{M_{\mathsf{s}(k,t)}}(e_j \in T)$ is weakly increasing in t, and strictly increasing when the intervals overlap (that is, when $s_j - 1 < s_k + t < s_j + 1$).*

The weak increase statement holds for any graph G (not just for $G = K_n$). On the other hand, the strict increase does rely on the geometry of the graph—to see this, note that a separating edge always has probability of inclusion equal to 1.

Proof. Fix $\mathsf{s} = (s_1, \ldots, s_m) \in \mathbb{R}^m$ and $\ell \in \{1, .., m\}$, and set $M_t := M_{\mathsf{s}(\ell,t)}$. Suppose e is an edge with endpoints v, w. In this notation, we have

$$\mathbb{P}_{M_t}(e \notin T) = \sum_{\substack{\hat{T} \in ST(K_n) \\ e \notin \hat{T}}} \mathbb{P}_{M_t}(\hat{T}) = \sum_{\substack{P: v \to w \\ |P| \geq 2}} \mathbb{P}_{M_t}(P \subset T),$$

where the last sum is over paths of length ≥ 2 between the endpoints of e, so that each P completes a closed loop with e. Here, T is the tree-valued random variable, and $\hat{T}$ is an instance (a particular tree). $P \subset T$ means that the path is found within the tree. For such a path P, denote its edges by $e_{i(1)}, ..., e_{i(k)}$.

If we write $w, w_1, \ldots, w_k$ for a set of draws of weights on the edges $e, e_{i(1)}, ..., e_{i(k)}$ under s, then we can write $w_i(t) = w_i + \delta_{i\ell} t$ for corresponding weights drawn from $\mathsf{s}(\ell, t)$. Then $P \subset T$ where T is minimal for the sampled weights if and only if $w \geq \max\{w_{i(1)}(t), ..., w_{i(k)}(t)\} =: m(t)$.

So $\mathbb{P}_{M_t}(P \subset T) = \mathbb{P}(w > m(t))$. If $\ell \notin \{i(1), ..., i(k)\}$, then $\mathbb{P}_{M_t}(P \subset T)$ does not depend on t. If $\ell \in \{i(1), ..., i(k)\}$, say $\ell = i(1)$, then there are two cases.

If $w < \max\{w_2, ..., w_k\}$, the probability of the event $w > m(t)$ equals zero. On the other hand, if $w > \max\{w_2, ..., w_k\}$, then the probability of the event $w > m(t)$ equals the probability that $w > w_\ell + t$. This probability is strictly decreasing in t as long as w is in the interior of the interval from which $w_\ell(t)$ is drawn from uniformly at random, i.e. whenever $|s_1 - (a_\ell + t)| < 1$. Integrating over all samples $(w, w_{i(2)}, ..., w_{i(k)})$ yields that $\mathbb{P}_{M_t}(P \subset T)$ is decreasing in t whenever $|s_1 - (s_\ell + t)| < 1$, which finally tells us that $\mathbb{P}(e \in T)$ is increasing as desired. □

Theorem 4.6 (Shifts do not suffice)**.** *For $G = K_n$ with $n \geq 4$, there is no $\mathsf{s} \in \mathrm{Sh}(\binom{n}{2})$ for which $M_{\mathsf{s}} = \mathrm{UST}$.*

Proof. Given a shift vector s, if $\min s_i < \max s_i$, then applying interval sliding (Lemma 4.5) shows that an edge chosen from the lowest interval is more likely to be included in a tree chosen from M_{s} than edge chosen from the highest. Thus $M_{\mathsf{s}} \neq \mathrm{UST}$. On the other hand, if $s_i = s_j$ for all $1 \leq i, j \leq m$, then $M_{\mathsf{s}} = \mathrm{MST}_0 \neq \mathrm{UST}$, since stars are more likely than paths. □

4.3 A limitation of product measures with connected support

To see the limitations of product measures obtained by shifting, we next show that a shift measure has to be extremely complicated to hit given probabilities on trees; in particular, it is hard to hit the uniform distribution.

Proposition 4.7 (UST on K_n)**.** *Let $n \geq 4$, and suppose $\{X_e\}_E$ is a product measure on K_n, where each X_e is supported on a connected subset of $\mathbb{R}$. If there exist any two edges e and e' with a common endpoint for which X_e is distributed identically to $X_{e'}$, then $\{X_e\}_E$ does not induce UST on K_n.*

Proof. Since the sets $\mathrm{supp}(X_e)$ for each edge e are all connected intervals in $\mathbb{R}$, there are two possibilities for how they can intersect:

(1) There is some pair of distinct edges $e_1, e_2 \in E(K_n)$ such that $\text{supp}(X_{e_1})$ and $\text{supp}(X_{e_2})$ are disjoint intervals (possibly overlapping at endpoints), with $w_1 \leq w_2$ for all $w_1 \in \text{supp}(X_{e_1})$ and $w_2 \in \text{supp}(X_{e_2})$.

(2) The intersection $\bigcap_{e \in E} \text{supp}(X_e)$ is an interval $[a, b]$ of nonzero length (i.e., $a < b$).

In case (1), observe that it is impossible for the product measure $\{X_e\}_E$ to yield a minimum spanning tree that is a path from one endpoint of e_1 to the other, passing through e_2, because swapping out e_2 for e_1 would yield a spanning tree of lower weight. (It is indeed *strictly* lower, otherwise the product measure fails to be non-colliding.) Therefore, the product measure does not induce UST, which requires equal weight on all trees.

In case (2) we will use an edge rotation, observing that the cycle expansion inequality of Proposition 3.10 holds in a slightly more general context, beyond ordinary MST. The only property we require for the weak inequality to hold is that the cycle-expanding bijection preserve the probability distributions on the edges that are being permuted. In particular, it applies to the bijection that swaps edges e and e' since they have the same distributions by assumption. And the only property we require for the strict inequality to hold is that every permutation has nonzero probability of occurring, which follows from the condition in case (2). Thus, we may apply Lemma 3.8 with T_1 and T_3 as single vertices and T_2 some spanning tree of the rest of K_n (as in the top pair of graphs in Figure 4), with $\{e_{12}, e_{13}\} = \{e, e'\}$ to conclude that some pair of trees have unequal probabilities of being sampled. (Note that $n \geq 4$ implies there is another edge in K_n between T_1 and T_2.) □

Consider coloring the edges of a graph with a different color for distinct distributions; in this formulation, the proposition states that a product measure inducing UST would have to be a proper coloring. Note that the edge coloring problem for complete graphs requires at least $n - 1$ colors if n is even and n colors if n is odd, which means that a pattern of shifts recovering UST has to be highly nontrivial if one exists at all.

5 Arbitrary product measures

In this section, we move to the full generality of product measures, exploring the very natural question presented in the introduction: when real-valued random variables are drawn from independent measures, how likely are they to be in each possible order? That is, what distributions on permutations are induced? As noted above, the distribution on permutations is a strict generalization of the classic topic of intransitive dice, which just concerns pairwise comparisons.

5.1 Defining the permutation locus

For any integer $m \geq 1$, let $\mathcal{M}(\mathbb{R}^m)$ be the set of finite Borel probability measures on $\mathbb{R}^m$. In this section we study the set $\mathcal{M} = \mathcal{M}_m \subsetneq (\mathcal{M}(\mathbb{R}))^m \subsetneq \mathcal{M}(R^m)$ consisting of non-colliding product measures, meaning that every $\mu \in \mathcal{M}$ corresponds to m independent random variables $X_1, \ldots, X_m$ on the real line $\mathbb{R}$ with the property that $\mathbb{P}(X_i = X_j) = 0$ for all $i \neq j$. The non-colliding property is automatic when the X_i are continuous random variables, but below we will explore discrete measures with finite support, where the non-colliding property requires the supporting points to be distinct.

Let $\Delta(S_m)$ be the space of probability measures on permutations, which is a simplex with the $m!$ permutations as its extreme points. Consider the map $\psi \colon \mathcal{M} \longrightarrow \Delta(S_m)$ given by

$$\psi(\mu)(\sigma) := \mu\left(\{x \in \mathbb{R}^m : x_{\sigma_1} < \cdots < x_{\sigma_m}\}\right).$$

Non-collision implies that any draw from the X_i induces a strict order, so $\psi(\mu)$ is a probability distribution on S_m. The main goal of this section is to understand the image of ψ, denoted by $P_m := \text{Im}(\psi)$, which we call the *permutation locus* for product measures.

First, we observe that the image of ψ is a subset of the simplex $\Delta(S_m)$ that contains all $m!$ extreme points. To see this, note that you can construct a probability measure supported only on the permutation σ by having each of the random variables X_i be an atomic measure with support at one point, and arrange the support points in the order needed to get σ.

Next, we note that the image is not everything (and so, in particular, is not convex.) In the sections below, we will realize the image of ϕ as a semi-algebraic set and will set about describing bounds on its dimension—non-surjectivity follows from a dimension count. But in addition, it is easy to explicitly describe a point of $\Delta(S_3)$ (say) that is missed by product measures.

Example 5.1 (A point missed by product measures)**.** No product measure on $\mathbb{R}^3$ induces equal probability $1/3$ for each of the events $a < b < c$, $\quad b < c < a$, and $c < a < b$.

To see this, name three independent random variables a, b, and c on $\mathbb{R}$. Write Φ_a, Φ_b and Φ_c for the corresponding cumulative distribution functions.

Suppose that $\mathbb{P}(a < b < c) = \mathbb{P}(b < c < a) = \mathbb{P}(c < a < b) = \frac{1}{3}$. Then

$$0 = \mathbb{P}(b < a < c) = \int_{-\infty}^{+\infty} \int_{-\infty}^{z} \int_{-\infty}^{x} d\Phi_b(y) d\Phi_a(x) d\Phi_c(z) = \int_{-\infty}^{+\infty} \int_{-\infty}^{z} \Phi_b(x) d\Phi_a(x) d\Phi_c(z).$$

Since everything is non-negative, we get

$$\int_{-\infty}^{z} \Phi_b(x) d\Phi_a(x) = 0, \qquad \text{for } c\text{-a.e. } z \in \mathbb{R}.$$

Let $[c_1, c_2]$ be the smallest interval containing the support of $d\Phi_c$. Then $\Phi_b(x) = 0$ for a-a.e. $x \in [c_1, c_2]$. Thus, if $[b_1, b_2]$ is the smallest interval containing the support of $d\Phi_b$, we must have $c_2 \leq b_1$. By cyclicity, $\mathbb{P}(a < c < b) = \mathbb{P}(c < b < a) = 0$ implies that

$$a_1 \leq a_2 \leq c_1 \leq c_2 \leq b_1 \leq b_2 \leq a_1,$$

so $a = b = c$ all have equal one-point support, which contradicts the given description.

Certain equalities and inequalities are easily seen to be satisfied at all points in P_m. These include independence identities like $\mathbb{P}(a < b \text{ and } c < d) = \mathbb{P}(a < b) \cdot \mathbb{P}(c < d)$ and correlation inequalities like $\mathbb{P}(a < b \mid a < c) \geq \mathbb{P}(a < b)$.[5] The equalities cut down the dimension of P_m and the inequalities create its faces. A fuller study of dimension is found in Section 5.4.

In the remainder of this section, we will construct discrete measures and will use these to express arbitrary product measures and the uniform distribution in particular. Using this technology, we will bound the dimension of P_m and make a conjecture about its exact value.

5.2 Discrete product measures and word maps

Here, we develop the formalism of *weighted words* over a finite alphabet, and we will ultimately prove that these suffice to capture all non-colliding product measures. The terminology of "words" is meant to suggest parallels to combinatorial group theory, but this presentation stands alone without group-theoretical machinery. One motivation is to show that the locus of product measures P_m is semi-algebraic by describing it with *word maps*—mappings from weights to measures for fixed words.

Definition 5.2 (Word maps and weighted words)**.** Consider a finite set of symbols $\Sigma = \{a_1, \ldots, a_m\}$, which we will think of as an alphabet. A word of length r is a string of r symbols from the alphabet, $w \in \Sigma^r$. This word induces a *word map* $F_w : \mathbb{R}^r_{\geq 0} \to \Delta(S_m)$ as follows. Let $\alpha \in \mathbb{R}^r_{\geq 0}$ be a weight vector. For a symbol a_i, appearing s_i times in w, suppose the weights of these occurrences are $(r_i^1, \ldots, r_i^{s_i})$, summing to r_i. For each symbol a_i, we will select exactly one of its occurrences in the word w, selecting the kth occurrence with probability r_i^k / r_i. This makes $F_w(\alpha)$ into a probability distribution on S_m, where the probability of $\sigma \in S_m$ is given by the selected symbols coming in the order $a_{\sigma(1)} a_{\sigma(2)} \ldots a_{\sigma(m)}$. We will call the pair (w, α) a *weighted word* over the alphabet Σ, and let $\mathcal{W}_{m,r} \subseteq \mathcal{M}_m$ denote the measures on S_m given by weighted words of length $\leq r$.

[5] The latter is a simple example of a *positive correlation inequality*, which fits in the broader framework of FKG correlation inequalities from statistical physics and percolation theory.

Example 5.3. Suppose the alphabet has only two letters: $\Sigma = \{a, b\}$. Consider the weighted word given by $w = abab$ and $\alpha = (2, 1, 3, 5)$, which gives weights $(2, 3)$ to copies of a and $(1, 5)$ to copies of b. We now describe how to interpret this as a probability distribution on the strings $\{ab, ba\}$. The word $abab$ contains both ordered subsequences: ab (achieved three ways as $\underline{\mathbf{a}}\,\underline{\mathbf{b}}\,a\,b$, $\underline{\mathbf{a}}\,b\,a\,\underline{\mathbf{b}}$, or $a\,b\,\underline{\mathbf{a}}\,\underline{\mathbf{b}}$) and ba (achieved one way as $a\,\underline{\mathbf{b}}\,\underline{\mathbf{a}}\,b$). The weights tell us that the relative likelihood that the symbols are chosen in each position, so that for instance b is five times as likely to be chosen in the fourth position as in the second. We calculate that ba is chosen with probability $\frac{3}{5} \cdot \frac{1}{6} = \frac{1}{10}$, and ab is chosen the rest of the time (probability 9/10). We have therefore defined a probability distribution on the permutations of $\{a, b\}$.

We can think of the orderings ab and ba as *draws* from the weighted word (w, α).

A weighted word also has a concrete interpretation as an atomic measure, which for this example is shown in Figure 9. Fix any r points on the real line, labeled in order as in the word w, with probability weights r_i^k / r_i as described above. This describes a product measure because the random variable X_i is independent from X_j; it induces a distribution on permutations via the order of $\{X_1, \ldots, X_m\}$.

Figure 9: The measure $X = \frac{9}{10}\delta_{ab} + \frac{1}{10}\delta_{ba} \in \Delta(S_2)$ can be written as a product measure in many ways, including the one shown here. The measure X_1 is supported on $\{2, 5\}$ and the measure X_2 is supported on $\{2.8, 7\}$, with the atoms having probabilities marked above, so that $X_1 = \frac{2}{5}\delta_2 + \frac{3}{5}\delta_5$ and $X_2 = \frac{1}{6}\delta_{2.8} + \frac{5}{6}\delta_7$. The probability that $X_1 < X_2$ is therefore 9/10, so the product measure (X_1, X_2) recovers X. This corresponds to the word $abab$ weighted by $(2, 1, 3, 5)$. The positions $2, 2.8, 5, 7$ are arbitrary in this construction; only the alternation between a sites and b sites is prescribed by w.

Since we use normalized probabilities r_i^k / r_i, the weight vector is scale invariant, and we can use homogeneous coordinates $\{[r_1^1 : \cdots : r_1^{s_i}] \mid r_i^k \geq 0\} \subset \mathbb{RP}^{s_i - 1}$ for the weights, allowing us to group by symbol and rewrite the weights $(2, 1, 3, 5)$ in the example above as $([2, 3], [1, 5])$. We can use this point of view to extend beyond non-negative weights and work with complexified projective spaces like $\mathbb{PC}^{s_1} := \mathbb{CP}^{s_1 - 1}$. We can likewise do this on the codomain, thinking of the set $\Delta(S_m)$ of distributions on permutations as sitting in a projective space $\mathbb{PC}^{m!} = \mathbb{CP}^{m! - 1}$. Thus, word maps extend to algebraic maps

$$F_w : \mathbb{PC}^{s_1} \times \cdots \times \mathbb{PC}^{s_m} \to \mathbb{PC}^{m!}.$$

Theorem 5.4 (Bounded-length word maps suffice)**.** *For any m there is an N so that for any non-colliding product measure on m variables, there is a weighted word (w, α) with the same image in $\Delta(S_m)$, where w has length at most N.*

We can choose $N = m(m! + 1)$, giving $P_m = \psi(\mathcal{W}_{m,\ m(m!+1)})$.

The proof has two parts, one that represents arbitrary measures with possibly long word maps, and a lemma that reduces the length of the word representation.

Lemma 5.5 (Shortening word maps)**.** *Let (w, α) be a word map on m symbols with any number of copies of each symbol. Then there is a word map (w', α') inducing the same distribution on permutations with $s_i \leq m!$, i.e., using at most $m!$ copies of each of the m symbols.*

Proof. We describe a procedure that iteratively modifies the input word map, one index at a time, until it uses no more than $m!$ copies of each symbol. The main tool is Carathéodory's Theorem [20, Theorem 17.1], which says in particular that if a point p in $\mathbb{R}^d$ is a convex combination of an arbitrary set of points, then there is a subset of at most $d + 1$ points whose convex hull contains p.

Fix an index i. Let $q_j = r_i^j / r_i$ be the probability that we select the jth position for symbol a_i. Let $p_j(\sigma)$ be the probability that (w, α) induces $\sigma \in S_m$, conditioned on choosing the jth position for a_i. Then we have $p(\sigma) = \sum_{j=1}^{r_i} q_j p_j(\sigma)$.

Thinking of p and p_j as vectors in $\mathbb{R}^{m!}$ (where each coordinate is given by a different choice of σ), we have

$$p = \sum_{j=1}^{r_i} q_j p_j, \qquad \text{where } \sum_{j=1}^{r_i} q_j = 1 \text{ and } 0 \le q_j \le 1.$$

We see that p is contained in the convex hull of $\{p_1, p_2, \ldots, p_{r_i}\}$. Since the p_j lie in $(m!-1)$-dimensional subspace, Carathéodory's theorem implies that we can write p as a convex combination of only $m!$ of these points. Since each of the p_j amounted to putting a_i in a particular position, this reduction amounts to deleting all but $m!$ of the positions. We may repeat this process for each index i. □

Proof of Theorem 5.4. Let $X = (X_1, X_2, \ldots, X_m)$ be an arbitrary product measure with distribution $p \in \Delta(S_m)$. For each positive integer j, we will construct a discrete product measure $\hat{X}^j := (\hat{X}_1^j, \hat{X}_2^j, \ldots, \hat{X}_m^j)$ so that, for each i, the $\hat{X}_i^j$ approximate X_i as $j \to \infty$. This is subdivided in three parts: the construction, a crucial estimate, and the proof of convergence.

Part I: Construction of $\hat{X} = \hat{X}^j$. Write Φ_i for the cumulative distribution function of the variable X_i. Let the inverse $\Phi_i^{-1}\colon [0,1] \to \mathbb{R}^* = \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$ be $\Phi_i^{-1}(u) := \inf\{x \in \mathbb{R}^* : \Phi_i(x) \ge u\}$, so that whenever $0 < \Phi_i(x) < 1$, we have that $u \le \Phi_i(x)$ iff $\Phi_i^{-1}(u) \le x$. We then have the standard construction $X_i = \Phi_i^{-1}(U)$, where U is uniform on $[0,1]$.

Let j be a positive integer and split the interval $[0,1]$ uniformly into j consecutive half-open intervals:

$$[0,1] = \left[0, \tfrac{1}{j}\right) \cup \left[\tfrac{1}{j}, \tfrac{2}{j}\right) \cup \cdots \left[\tfrac{j-1}{j}, 1\right]$$

Then, sampling from U is equivalent to choosing $k \in \{0, 1, \ldots, j-1\}$ uniformly at random and then sampling from U conditioned to be in $\left[\frac{k}{j}, \frac{k+1}{j}\right)$. In other words, we have the variable decomposition $X_i = \sum_{k=0}^{j-1} X_{i,k}^j$, where $X_{i,k} := \Phi_i^{-1}\left(U \cdot \mathbb{1}_{[k/j,(k+1)/j)}\right)$.

These $X_{i,k}^j$ depend on a choice of j, but we suppress that for now for this and related objects. Then, in view of the variable decomposition above, we can think of the sample space for the product measure X as being the cube $\Omega := [0,1]^m$.

For $k = 0, 1, \ldots, j$, set $x_{i,k} := \Phi_i^{-1}\left(\frac{k}{j}\right)$ and (for $k < j$) consider the intervals $B_{i,k}$ from $x_{i,k}$ to $x_{i,k+1}$, where we include an endpoint in the interval iff X_i has a point mass there. (Thus the $B_{i,k}$ for different k may overlap at their endpoints, and may be single points.) Now we define a sequence of approximating discrete product measures $\hat{X}$ on the same sample space $\Omega = [0,1]^m$. For each $i = 1, \ldots, m$, pick (once) $u_{i,k}$ uniformly randomly in the interval $\left[\frac{k}{j}, \frac{k+1}{j}\right)$, and set $\hat{X}_i$ to be the discrete product measure equally weighted on the points $\hat{X}_{i,k} := \Phi_i^{-1}(u_{i,k})$. With probability one, $\hat{X}_{i,k} \in B_{i,k}$ and the values $\hat{X}_{i,k}$ are distinct for different i, since the product measure X was assumed non-colliding.

Part II: Estimating the error in the resulting distribution. Next, let $\hat{p} = \hat{p}^j \in \Delta(S_m)$ be the distribution induced by $\hat{X} = \hat{X}^j$ and fix a permutation $\sigma \in S_m$. We want to estimate the difference $\left|\hat{p}^j(\sigma) - p(\sigma)\right|$ in terms of the resolution j of our grid. To that end, apply the variable decomposition $X_i = \sum_{k=0}^{j-1} X_{i,k}$ to each X_i, and pick any two terms from different X_i, say $X_{1,k}$ and $X_{2,\ell}$. These two variables are supported in the intervals $B_{1,k}$ and $B_{2,\ell}$, as are the corresponding approximating variables $\hat{X}_{1,k}$ and $\hat{X}_{2,\ell}$. In particular, if $B_{1,k} \cap B_{2,\ell} = \emptyset$, then the relative order of $X_{1,k}$ and $X_{2,\ell}$ is preserved when we pass to $\hat{X}_{1,k}$ and $\hat{X}_{2,\ell}$. We will give bounds on the number of possible overlaps.

Now suppose $B_{1,k} \cap B_{2,\ell} \neq \emptyset$. We claim that if $k' < k$ and $\ell < \ell'$ then $B_{1,k'}$ and $B_{2,\ell'}$ do not intersect. Intuitively, $B_{1,k'}$ is to the left of $B_{1,k}$, which overlaps $B_{2,\ell}$, which is to the left of $B_{2,\ell'}$. More formally, we have inequalities between the endpoints of these intervals:

$$x_{1,k'+1} \le x_{1,k} \le x_{2,\ell+1} \le x_{2,\ell'}$$

where the first inequality comes from $k' < k$, the second inequality from the intersection between $B_{1,k}$ and $B_{2,\ell}$, and the third from $\ell < \ell'$. Thus we can only have an intersection between $B_{1,k'}$ and $B_{2,\ell'}$ if all three

inequalities are equalities and furthermore X_1 has a point mass at $x_{1,k}$ and X_2 has a point mass at $x_{2,\ell}$. But this contradicts the assumption that X is non-colliding.

With $k, \ell \in \{0, \dots, j-1\}$, consider the indices of the non-empty intersections $\{(k,\ell) \mid B_{1,k} \cap B_{2,\ell} \neq \emptyset\}$. By the claim above, these intersections are totally ordered by the relation $(k,\ell) < (k',\ell')$ iff $k < k'$ or $\ell < \ell'$. A maximum chain of such intersections therefore contains $2j-1$ intersections total, since the sum $k+\ell$ can take integer values between 0 and $2(j-1)$. In particular, with probability at least $1 - \frac{2j-1}{j^2}$, the variables $\hat{X}_1$ and $\hat{X}_2$ have the same relative ordering as X_1 and X_2. Since this applies for all pairs of variables, by the union bound, for all $\sigma \in S_m$, we have $\left|\hat{p}^j(\sigma) - p(\sigma)\right| < \binom{m}{2} \cdot \frac{2}{j} < \frac{m^2}{j}$ and the $\hat{p}^j$ converge to p.

Part III: Proof of convergence. Consider the sequence of discrete product measures $Y^1, Y^2, Y^3, \dots$ defined by letting Y^j be the shortening of $\hat{X}^j$ to use at most $m!$ of each symbol (using Lemma 5.5). Topologically, we may think of each Y^j as lying in a space

$$C \coloneqq F \times \mathbb{P}_1^{s_1} \times \cdots \times \mathbb{P}_m^{s_m}$$

where F is a finite set encoding the combinatorial information of the relevant word maps and the probability weights are drawn from the projective spaces, as before. Since F is finite and the projective spaces are compact, C is compact. Therefore, there is a subsequence $Y^{j_1}, Y^{j_2}, Y^{j_3}, \dots$ converging to a limit $Y^* \in C$ under the L_∞ norm. That is, Y^* is a discrete product measure where each Y_i^* is supported on a set of size at most $m!$ agreeing with each $Y_i^{j_k}$, and furthermore, for any $\varepsilon > 0$, sufficiently large k, and value v in the support, $\left|\mathbb{P}(Y_i^* = v) - \mathbb{P}(Y_i^{j_k} = v)\right| < \varepsilon$. Since Y^{j_k} induces p^{j_k}, it follows that Y^* induces a distribution p^* such that, for all $\sigma \in S_m$, $\left|p^*(\sigma) - p^{j_k}(\sigma)\right| \leq m\varepsilon$. By the triangle inequality, we have

$$|p^*(\sigma) - p(\sigma)| \leq m\varepsilon + \frac{m^2}{j_k}.$$

It follows that $p^* = p$, i.e., Y^* induces the same distribution as X. □

Corollary 5.6 (Universal words)**.** *For any word u of length at least $(m!)m(m-1)+1$ that cyclically repeats all of the letters in the same order, the collection of word maps (u, α) suffice to cover all of P_m.*

We call such a word u a *universal* word. For example, when $m = 3$, the word

$$u = abcabcabcabcabcabcabcabcabcabcabca,$$

(which consists of 12 copies of the string abc followed by the letter a) is a universal word.

Proof. From the previous theorem, any product measure can be associated to some word map on a word w with at most $m!$ copies of each symbol, for a total length of $(m!)m$. We will find this w within any given universal word u by explaining how to choose the weights. The first letter of w occurs in position m or earlier of u, and each of the remaining letters occur at most $m-1$ positions down from the previous one. Thus, after cyclically enumerating

$$m + ((m!)m - 1)(m-1) = (m!)m(m-1) + 1$$

letters of the universal word u, we have realized w as a subsequence. This implies that we can realize Y^* with the appropriate weights over u; we simply set the weights of unused letters to be zero. □

Another consequence of Theorem 5.4 is that P_m is a semi-algebraic subset of $[0,1]^{m!}$, meaning that it is described by a finite set of equalities and inequalities in the variables x_σ, as follows. For a given w, the coefficients of the x_σ are polynomial expressions in the weights α, and letting the weights vary over a universal word parametrizes the entirety of P_m. Then apply the Tarski-Seidenberg Theorem, which states that the image of a semi-algebraic set under an algebraic map is also semi-algebraic.

5.3 Word maps for the uniform distribution

Because it leverages compactness for convergence, Theorem 5.4 is non-constructive. In this section we show that the classical theory of quadrature gives an elegant construction of short words inducing the uniform distribution. To motivate this, we start with a more naive construction and observe that it is highly inefficient.

For any permutation $\sigma \in S_m$ and any word w on m symbols, we define $\sigma \circ w$ to be the word where we substitute every letter in w according to σ. We inductively define words v as follows. Let $v_{1,m}$ be a word containing each of the m letters once in arbitrary order, and for each $k \geq 2$, let

$$v_{k,m} \coloneqq \prod_{\sigma \in S_m} \sigma \circ v_{k-1,m},$$

where the product symbol refers to the operation of concatenation.

Theorem 5.7. *The words $v_{m,m}$, with equal weight on each letter, induce the uniform distribution on S_m.*

Proof. We prove the following stronger claim by induction: consider a subset $\Sigma_0 \subset \Sigma$ containing at most r of the letters from the alphabet. Restricting $v_{r,m}$ to the letters from Σ_0, we will show that the resulting word map (with equal weights) induces the uniform distribution over orderings of Σ_0.

Fix any m. The base case, $r = 1$, is immediate. For the inductive step, suppose $r \geq 2$ and the claim holds for $v_{r-1,m}$. There are two cases to consider. First suppose that all r of the letters in a draw from $v_{r,m}$ happen to be chosen from the same copy of $v_{r-1,m}$. Conditioning on this random event, each copy of $v_{r-1,m}$ is equally likely, and hence the corresponding permutation $\sigma \in S_m$ is uniformly random as well. Thus every possible ordering of the r letters is equally likely. Now suppose instead that at most $r-1$ of the r letters are chosen from any individual copy of $v_{r-1,m}$. Conditioning on this event, the inductive hypothesis says the letters are ordered uniformly at random within each copy. Thus, the r letters are randomly distributed across the different copies of $v_{r-1,m}$ and then randomly ordered wherever there is a collision. This process is clearly equivalent to uniformly randomly ordering the r letters. By induction, the claim holds for all $r \geq 1$. □

For $m = 2$ the two possible words $v_{2,2}$ arising from this construction are *abba* and *baab*. A word $v_{3,3}$ for $m = 3$, broken up into the six transformed copies of $v_{2,3}$ indexed by S_3, is:

abcacbbacbcacabcba acbabccabcbabacbca bacbcaabcacbcbacab
bcabaccbacababcacb cabcbaacbabcbcabac cbacabbcabacacbabc

The length 108 of $v_{3,3}$ is significantly longer than the universal word of length 37 given above for $m = 3$ (see below Corollary 5.6), but it has the benefit of simple weights $(1, 1, \ldots, 1)$. In general, the length coming from this construction is $m(m!)^{m-1}$, which is far greater than the $m(m!)$ upper bound established by Theorem 5.4 that would suffice to hit the uniform distribution. Closing the gap with an explicit construction is the task at hand.

A more efficient construction can be drawn from ideas of *quadrature*, meaning methods of approximating the integral of a function f on an interval by evaluating f at a small number of points.

Definition 5.8. A *quadrature scheme* is an approximation to an integral constructed by a weighted combination (with weights α_i) of function values at particular points ($0 \leq x_i \leq 1$): $\int_0^1 f(x)\, dx \approx \sum_i \alpha_i f(x_i)$. A quadrature scheme is said to *integrate polynomials of degree $\leq d$* if the approximation is an equality for those inputs.

Theorem 5.9 (Quadrature words)**.** *Suppose that $\sum_{i=1}^{r} \alpha_i \delta_{x_i}$ is a degree-k quadrature scheme with $0 \leq x_1 < x_2 < \cdots < x_r \leq 1$ and w is a uniform word in $a_1, \ldots, a_k$. Then*

$$U = w^{x_1 - 0} a_{k+1}{}^{\alpha_1} w^{x_2 - x_1} a_{k+1}{}^{\alpha_2} \ldots a_{k+1}{}^{\alpha_r} w^{1 - x_r}$$

is a uniform word in $a_1, \ldots, a_{k+1}$.

(Recall that exponents on a letter are used to indicate the weight of that letter in our word map; we abuse notation to write w^α to mean the word w with each letter individually scaled.) We defer the proof to Appendix D.

Two examples of quadrature schemes are the approximations

$$\int_0^1 f(x)\,dx \approx \frac{f(0)}{4} + \frac{3f(2/3)}{4}, \qquad \int_0^1 f(x)\,dx \approx \frac{f(0)}{6} + \frac{2f(1/2)}{3} + \frac{f(1)}{6}.$$

These are, respectively, *Gauss-Radau quadrature* with weights $(1/4, 3/4)$ and nodes $(0, 2/3)$ and *Gauss-Lobato quadrature* with weights $(1/6, 2/3, 1/6)$ and nodes $(0, 1/2, 1)$. The first integrates polynomials of degree up to 2 and the second integrates polynomials of degree up to 3.

We can now apply the theorem in the case $m = 3$. Since $w = aba$ is a word in the two variables a, b generating the uniform distribution on those letters, applying the degree-2 Gauss-Radau quadrature scheme gives

$$c^{1/4}w^{2/3}c^{3/4}w^{1/3} = c^{1/4}a^{2/3}b^{2/3}a^{2/3}c^{3/4}a^{1/3}b^{1/3}a^{1/3} \equiv ca^2b^2a^2c^3aba$$

as a uniform word on three letters, which at length 8 is highly efficient.

These constructions extend to higher degree, where Gauss-Radau has $x_1 = 0$ and integrates polynomials up to degree $2r - 2$, while Gauss-Lobatto has $x_1 = 0, x_r = 1$ and integrates polynomials up to degree $2r - 3$, handling the odd and even cases, respectively [18]. (As a note, the form of our quadrature theorem means that we will produce the shortest words when $x_1 = 0$, $x_r = 1$, or both. This makes Gauss-Radau and Gauss-Lobato slightly more efficient than the more popular quadrature scheme known as Gauss-Legendre, whose nodes are all internal.) As we show in Appendix D, this adds up as follows.

Proposition 5.10 (Word lengths from quadrature)**.** *Define $L(m)$ recursively by $L(1) = 1$ and*

$$L(m) = \begin{cases} \frac{m+1}{2} \cdot L(m-1) + \frac{m+1}{2}, & m \text{ odd} \\ \frac{m}{2} \cdot L(m-1) + \frac{m+2}{2}, & m \text{ even.} \end{cases}$$

Then there is a word map (w_m, α_m), with w_m of length $L(m)$, that generates the uniform distribution on m letters.

Clearly $L(m)$ grows factorially; Table 2 gives the first few values.

m	$L(m)$	(w_m, α_m)
1	1	a
2	3	bab
3	8	$cb^2a^2b^2c^3bab$
4	19	$dcb^2a^2b^2c^3babd^4cb^2a^2b^2c^3babd$
5	60	
6	184	
7	740	

Table 2: The first few values of $L(m)$, and corresponding uniform words w_m.

Note that varying the weights on the uniform word w_4 from this construction gives a subset of P_4 of dimension at most $3 + 7 + 3 + 2 = 15$, and we will see in the next section that $\dim P_4 = 20$, showing that these are not also universal words. However, the length of 19 significantly improves on all constructions discussed previously for uniform words. In particular, the naive construction $v_{4,4}$ from Theorem 5.7 would have required 55,296 letters.

5.4 Dimension of P_m

We may think of each point in the permutation locus P_m as a vector

$$(x_{(1,2,3,4\ldots,m)}, x_{(2,1,3,4,\ldots,m)}, x_{(2,3,1,4,\ldots,m)}, \ldots, x_{(m,m-1,m-2,m-3,\ldots,1)}) \in [0,1]^{m!}$$

where $x_{(i_1,i_2,\ldots,i_m)}$ is the probability of realizing the ordering of random variables $X_{i_1} < X_{i_2} < \cdots < X_{i_m}$. In this section, we will sometimes use numerical symbols rather than letters to denote orderings and permutations, and we will switch between horizontal and vertical notation for permutations, as needed for clean exposition. We have seen that P_m is a semi-algebraic set in $[0,1]^{m!}$. In this final section, we ask the most basic question about P_m: what is its dimension at generic points? Fuller definitions and proofs for this section appear in Appendix E; e.g., see Definition E.3 for the notion of "generic".

For instance, P_3 sits in $\Delta(S_3)$, a simplex on 6 extreme points, so the most naive upper bound on its dimension is 5. This turns out to be tight: while P_3 does not contain every point in this simplex, it is still full-dimensional. However, that is no longer true for P_4. This is due to additional constraints that hold of all distributions in P_4, such as the *independence constraints* mentioned in Section 5.1. For instance, the event $X_1 < X_2$ must be independent from the event $X_3 < X_4$:

$$\mathbb{P}([X_1 < X_2] \text{ and } [X_3 < X_4]) = \mathbb{P}(X_1 < X_2) \cdot \mathbb{P}(X_3 < X_4).$$

Translating to permutation notation, the left-hand side has six terms, while the probabilities on the right expand to twelve terms each:

$$x_{\left[\begin{smallmatrix}1\\2\\3\\4\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}1\\3\\2\\4\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}1\\3\\4\\2\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}3\\1\\2\\4\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}3\\1\\4\\2\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}3\\4\\1\\2\end{smallmatrix}\right]}$$

$$= \left(x_{\left[\begin{smallmatrix}1\\2\\3\\4\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}1\\3\\2\\4\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}1\\3\\4\\2\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}3\\1\\2\\4\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}3\\1\\4\\2\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}3\\4\\1\\2\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}1\\2\\4\\3\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}1\\4\\2\\3\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}1\\4\\3\\2\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}4\\1\\2\\3\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}4\\1\\3\\2\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}4\\3\\1\\2\end{smallmatrix}\right]} \right)$$

$$\cdot \left(x_{\left[\begin{smallmatrix}1\\2\\3\\4\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}1\\3\\2\\4\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}1\\3\\4\\2\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}3\\1\\2\\4\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}3\\1\\4\\2\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}3\\4\\1\\2\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}2\\1\\3\\4\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}2\\3\\1\\4\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}2\\3\\4\\1\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}3\\2\\1\\4\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}3\\2\\4\\1\end{smallmatrix}\right]} + x_{\left[\begin{smallmatrix}3\\4\\2\\1\end{smallmatrix}\right]} \right).$$

Since all vectors in P_4 satisfy this polynomial equation, we know that that $\dim(P_4) \leq \dim(\Delta(S_4)) - 1 = 22$. In fact, this is one of three similar equality constraints (partitioning $\{1,2,3,4\}$ into two sets of size 2), and the true dimension works out to be 20.

The main result of this section (Theorem 5.13) is a general upper bound on $\dim(P_m)$, which we conjecture to be tight (Conjecture 5.14). Our basic approach is to construct a large enough number of equality constraints that all points in P_m satisfy and show that their gradients are linearly independent at generic points. Since the constraints are all polynomials in the x_σ variables, it suffices to identify a *single* point at which the gradients of each constraint are linearly independent; at generic points, the number of independent constraints can only be larger. (See Lemma E.4.) We will thus choose to evaluate derivatives at the uniform distribution over permutations, where each $x_\sigma = \frac{1}{m!}$.

For the construction, we will not use independence constraints that directly generalize the example above; instead, we employ constraints like the following. Let A be the event $X_1 < X_2$ and B be $X_3 < X_4$. Then one identity can be nicely expressed by referring to the matrix

$$\begin{pmatrix} \mathbb{P}(A \wedge B) & \mathbb{P}(A \wedge \neg B) \\ \mathbb{P}(\neg A \wedge B) & \mathbb{P}(\neg A \wedge \neg B) \end{pmatrix}. \tag{5.1}$$

We see that the determinant of this matrix is zero because independence lets us express each conjunction as a product, so that both diagonal products are equal to $\mathbb{P}(A) \cdot \mathbb{P}(\neg A) \cdot \mathbb{P}(B) \cdot \mathbb{P}(\neg B)$. Note that the main diagonal contains terms with an even number of A, B events and the anti-diagonal contains terms with an odd number of A, B events.

This determinant identity generalizes via multilinear algebra to any number of events. If sets of events $\mathcal{E}_i^+$ for $i \in [k]$ are independent, then the product of the probabilities of all ways where an even number of the events happen equals the product of the probabilities of all ways where an odd number of the events happen. Equality still holds if you condition on another global event. In the most general setting, consider a collection of independent events $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_k$, each partitioned into disjoint sets $\mathcal{E}_i = \mathcal{E}_i^+ \sqcup \mathcal{E}_i^-$, not necessarily complementary. (In the previous example, there was no conditioning; $\mathcal{E}_1$ and $\mathcal{E}_2$ were both the entire universe,

with $\mathcal{E}_i^-$ equal to the complement of $\mathcal{E}_i^+$.) By expanding conjunctions into products and rearranging terms as before, one can prove the following even/odd constraint holds for any such collection of independent events; see Lemma E.1 for the formal proof.[6]

$$\text{(EO)} \qquad \prod_{\substack{s\colon [k]\to\{+,-\} \\ \text{such that } |s^{-1}(+)| \text{ is even}}} \mathbb{P}\left[\bigwedge_{i\in[k]} \mathcal{E}_i^{s(i)}\right] = \prod_{\substack{s\colon [k]\to\{+,-\} \\ \text{such that } |s^{-1}(+)| \text{ is odd}}} \mathbb{P}\left[\bigwedge_{i\in[k]} \mathcal{E}_i^{s(i)}\right].$$

For $m \geq 5$, we will take the gradient of Equation (EO) with respect to events involving more than just pairs of variables. For instance, in one of our constraints, we will use the following collection of events:

$$\mathcal{E}_1^+ := [X_1 < X_2 < X_3] \vee [X_3 < X_2 < X_1] \qquad \mathcal{E}_2^+ := [X_4 < X_5]$$
$$\mathcal{E}_1^- := [X_2 < X_1 < X_3] \vee [X_3 < X_1 < X_2] \qquad \mathcal{E}_2^- := [X_5 < X_4]$$

We will end up associating the corresponding EO constraint with the permutation (123)(45), just as the EO constraint from (5.1) is associated with the permutation (12)(34). For any m, we will prove that we have a linearly independent vector in the cotangent space of P_m at the uniform distribution for every permutation on $\{1, \dots m\}$ that has at least two nontrivial cycles. Counting the number of such permutations will therefore give us a lower bound on the dimension of the cotangent space, and thus an upper bound on the dimension of the tangent space, which is the dimension of P_m.

We now need to introduce just enough notation to compactly write down these cotangent vectors (i.e., gradients of these constraints at the uniform distribution). We will give the key ideas here, saving full notation and detail for Appendix E.

We will use $\Sigma \subset \{1, \dots, m\}$ for a set of symbols, and will write $O(\Sigma)$ for orderings on the symbols, such as $(1 < 3 < 2)$ as an ordering on $\{1, 2, 3\}$; we let $O(m)$ denote orderings of the full set $\{1, \dots, m\}$. Orderings are in natural bijection with permutations, but we maintain distinct notation for readability; we will treat $\mathbb{R}^{O(\Sigma)}$ as a vector space of formal combinations of orderings.[7] We think of the cotangent space of P_m as lying in $\mathbb{R}^{O(m)}$.

We define an *ordering map* $F\colon S(\Sigma) \to \mathbb{R}^{O(\Sigma)}$ carrying permutations to combinations of orderings. A formal definition is found in Definition E.2, but we give examples here that suggest the structure. We first define F on permutations which are pure cycles. For instance, $F((12)) = (1 < 2) - (2 < 1)$ and $F((123)) = (1 < 2 < 3) + (3 < 2 < 1) - (2 < 1 < 3) - (3 < 1 < 2)$. Products of disjoint cycles are combined by riffling the orderings together with a *shuffle operator* $\sqcup\!\sqcup$, so that, for example,

$$F((12)(34)) = F((12)) \sqcup\!\sqcup F((34)) = (1 < 2) \sqcup\!\sqcup (3 < 4) - (1 < 2) \sqcup\!\sqcup (4 < 3) - (2 < 1) \sqcup\!\sqcup (3 < 4) + (2 < 1) \sqcup\!\sqcup (4 < 3),$$

where we can expand

$$(1 < 2) \sqcup\!\sqcup (3 < 4) = (1 < 2 < 3 < 4) + (1 < 3 < 2 < 4) + (1 < 3 < 4 < 2) + (3 < 1 < 2 < 4) + (3 < 1 < 4 < 2) + (3 < 4 < 1 < 2),$$

and so on. For permutations $\pi \in S_m$, the vector $F(\pi)$ always has an equal number of terms with a positive sign and terms with a negative sign, because the same is true for images of cycles $F(\sigma_i)$. The general definition of the map F makes use of the formalism of Lie brackets and shuffle products, and uses a lexicographic normal form on permutations so that the map is well defined. The expressions $F(\pi)$ suggest the corresponding sets of events using indices from the σ_i.

We pull the following key statement from two works of Reutenauer and collaborators [19, Section 6.5.1] [16, Remark 6], who are working in the more general setting of free associative algebras. Our setting has a simplifying feature: we work with vectors that are homogeneous of degree 1 in each variable. This allows us to make the needed statement very succinct.

Lemma 5.11 (Lie shuffle basis)**.** *The set $\{F(\pi) : \pi \in S_m\}$ is a basis for $\mathbb{R}^{O(m)}$.*

[6] Russell Lyons has communicated to us an elegant proof of the converse: if the product-of-even probabilities equals the product-of-odd probabilities for all subsets of I, then the events $\mathcal{E}_i$ are independent. We will not apply this converse statement here.

[7] In fact, in many of the places where S_m is used in this paper, permutations are really appearing as orderings; however, we only introduce this notation here because it helps with disambiguation.

We call this basis the *Lie shuffle basis*. With this notation, we are ready to compute gradients in $\mathbb{R}^{O(m)}$.

Lemma 5.12 (Gradient expression)**.** *Let π be a permutation composed of disjoint cycles $c_1, c_2, \ldots, c_\ell$, and let $u_i = F(c_i)$. Then there are independent events $\mathcal{E}_i = \mathcal{E}_i^+ \cup \mathcal{E}_i^-$ for each $1 \le i \le \ell$ where c_i is nontrivial (has length at least 2) such that the gradient of Equation (EO) at the uniform distribution is a scalar multiple of $u_1 \sqcup u_2 \sqcup \cdots \sqcup u_k$.*

Together, these facts give us enough linearly independent cotangent vectors to get a dimension bound.

Theorem 5.13 (Dimension upper bound)**.** *Write $C(m) = \sum_{k=2}^{m} \frac{m!}{k\cdot(m-k)!}$ for the number of* pure cycles *in S_m, i.e., permutations with exactly one nontrivial cycle and all other points fixed. Then for all m, the dimension of P_m is at most $C(m)$.*

Proof. Consider the set

$$B := \{F(\pi) \mid \pi \text{ contains at least two nontrivial cycles}\}.$$

By Lemma 5.12, each element of B is a scalar multiple of a gradient of Equation (EO) at the uniform distribution for some collection of at least 2 disjoint events. By Lemma E.1, these gradients are all cotangent vectors. Since the elements of B are scalar multiples of a subset of the Lie shuffle basis, these gradient vectors are all linearly independent by Lemma 5.11. Thus, so far we have found a collection of independent cotangent vectors, one for each permutation with at least two nontrivial cycles. There is one more constraint: In $P_m \subset \Delta(S_m)$, all elements are linear combinations of permutations whose coefficients sum to one. Thus, in the tangent space, vectors can be regarded as linear combinations of permutations with coefficients summing to zero. In other words, we have the all-ones vector as an additional vector in the cotangent space, which is independent of the others. We define this vector to be F of the identity permutation. Hence, the codimension of the tangent space of P_m at the uniform distribution is at least the number of permutations that have *at least two* nontrivial cycles, plus one for the identity permutation (which has *zero* nontrivial cycles). Since this holds at the uniform distribution, Lemma E.4 lets us conclude that this holds at all points in P_m. Therefore, the dimension of P_m is at most the number of permutations with *exactly one* nontrivial cycle. □

This sequence $C(m)$ appears in OEIS as entry A006231 [17]. We note that $C(m) \sim e(m-1)!$, while $\Delta(S_m)$ itself has dimension $m! - 1$, so asymptotically this dimension tends to a fraction e/m of the ambient dimension.

Conjecture 5.14. *For all m, the dimension of P_m is exactly $C(m)$.*

We now describe our computational approach to verifying that this upper bound is tight up to $m = 7$.

Definition 5.15 (Draw matrix)**.** A word w on m symbols that uses each symbol at least once has some ordering of the m symbols as substrings; pairing the word with weights α in a word map (w, α) induces a distribution on S_m in the usual way, where the permutations are realized as draws from the word w using weights from α. The *draw matrix* M_w for a word w length r on m symbols is the $r \times m!$ matrix with columns indexed by S_m, where entry (i, σ) of M is the number of ways to draw σ as a substring of w that use the i^{th} letter of w.

For example, the draw matrix M_w for the word $w = abcabcba$ is:

	abc	*acb*	*bac*	*bca*	*cab*	*cba*
abcabcba	3	3	0	0	0	0
a**b**cabcba	2	0	1	3	0	0
ab**c**abcba	1	2	0	2	2	2
abc**a**bcba	1	1	1	1	2	0
abca**b**cba	2	1	0	1	1	1
abcab**c**ba	3	2	1	2	0	1
abcabc**b**a	0	3	0	0	1	2
abcabcb**a**	0	0	0	3	0	3

The permutation *abc* can be realized in four ways as a draw from *abcabcba*, three using the first a and one using the second a; this is reflected in the first column of M_w, in the first and fourth rows.

Lemma 5.16 (Matrix rank as a lower bound)**.** *For any word w on m symbols, the draw matrix gives a lower bound for the dimension of P_m:*

$$\dim(P_m) \geq \operatorname{rank}(M_w) - 1.$$

Proof. Define $\widetilde{P_m} := \{cx \mid c \in \mathbb{R}_{\geq 0},\ x \in P_m\}$. Clearly, the dimension of $\widetilde{P_m}$ is at most one more than the dimension of P_m; it thus suffices to show that $\dim(\widetilde{P_m}) \geq \operatorname{rank}(M)$. For this we lower-bound the dimension of the subset $S \subseteq \widetilde{P_m}$ defined as the set of non-normalized probability distributions (measures) over permutations of the m letters obtained by word maps (w, α) for *arbitrary* $\alpha \in \mathbb{R}^r_{\geq 0}$ (not necessarily summing to one). Observe that S is the image of a map f from $\mathbb{R}^r_{\geq 0}$ to $\mathbb{R}^{O(m)}_{\geq 0}$. We claim that the matrix M is the Jacobian of f evaluated at $\alpha = (1, 1, \ldots, 1)$. For any permutation σ, the map f defines

$$x_\sigma = \sum_{\substack{i_1, i_2, \ldots, i_m \in [r] \\ w_{i_1} w_{i_2} \ldots w_{i_m} = \sigma}} \alpha_{i_1} \alpha_{i_2} \ldots \alpha_{i_m}.$$

Taking the derivative of x_σ with respect to α_i, we see that the number of nonzero terms is precisely the number of subsequences of w yielding σ that use w_i, and furthermore, evaluating at $\alpha = (1, 1, \ldots, 1)$, these terms are all 1. Thus, M is the Jacobian of f, so its rank is the dimension of S at $\alpha = (1, 1, \ldots, 1)$. By Lemma E.4 this gives us the desired global lower bound on $\dim(\widetilde{P_m})$, and thus on $\dim(P_m)$ as well. □

Proposition 5.17 (Dimensions for small m)**.** $\dim(P_m) = C(m)$ *holds for* $m \leq 7$*, giving dimensions* $\dim P_m = 0, 1, 5, 20, 84, 409, 2365$ *for* $m = 1, 2, 3, 4, 5, 6, 7$*.*

For the example $w = abcabcba$ above, the reader can verify that the rank of the draw matrix is 6, implying the tight bound of $\dim(P_3) \geq 5$. Our approach for the rest of the verifications in Proposition 5.17 is simply to try sufficiently long words (roughly 10% longer than the anticipated rank of P_m) and compute ranks. Even for very long words, it is possible to very efficiently compute each entry of M using dynamic programming. However, the matrix M has thousands of rows and columns starting at $m = 7$; this is the largest size we can handle with current computational resources. Replication code can be found in [27].

6 Final remarks

We have explored some of the quantitative properties of ordinary MST, its successive extensions to shifted intervals, and then the full generality of product measures.

For ordinary MST, we have presented procedures for calculating the probability of any labeled tree in an arbitrary ambient graph G, and we have developed rotation moves that let us distinguish MST_0 from UST on random graphs and that let us show that paths and stars have extremal MST_0 probability in complete graphs. For shifted-interval MST, we have defined a combinatorially natural object called a shiftahedron as a parameter space for studying the induced distribution on permutations, and from there the distribution on trees. We included examples of distributions that are and are not realizable by shifts. Finally, for fully general product measures, we have made significant progress towards describing what distributions on permutations they induce, which enriches the well-known topic of *intransitive dice*. We established that every product measure has a corresponding word map of bounded length (associated to a discrete product measure with finite support) that induces the same probability weight on all permutations. This lets us tie the study of word maps to the theory of integration, which we can leverage to find particular word maps with nice properties. Finally, we have motivated and stated a conjecture that the dimension of the locus in $\Delta(S_m)$ hit by product measures is exactly the number of pure cycles in S_m. This is verified for $1 \leq m \leq 7$, and the upper bound is established for all m.

The results here give several ways of quantifying how ordinary MST is different from UST—for instance, the degree distribution skews to higher degrees because high-degree nodes create short broken cycles (cf. Theorem 3.4). But more than that, we build theory for the broad generalization to product measures. While the full space of distributions on S_m has dimension $m! - 1$, our results show that product measures fill out a locus whose dimension tends to $e \cdot (m-1)!$; indeed, just the part that is hit by shifted intervals has dimension at most $m - 1$ and is already flexible enough for many applications (as in §1.2).

References

[1] L. Addario-Berry, N. Broutin, and B. Reed. Critical random graphs and the structure of a minimum spanning tree. *Random Structures Algorithms*, 35(3):323–347, 2009.

[2] David J. Aldous. The random walk construction of uniform spanning trees and uniform labelled trees. *SIAM Journal of Discrete Mathematics*, 3(4):450–465, 1990.

[3] Nima Anari, Shayan Oveis Gharan, and Cynthia Vinzant. Log-concave polynomials, I: Entropy and a deterministic approximation algorithm for counting bases of matroids. *Duke Mathematical Journal*, 170(16):3459 – 3504, 2021.

[4] Nima Anari, Kuikui Liu, Shayan Oveis Gharan, and Cynthia Vinzant. Log-concave polynomials II: High-dimensional walks and an FPRAS for counting bases of a matroid. *Annals of Mathematics*, 199(1):259 – 299, 2024.

[5] Béla Bollobás. *Random Graphs*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2 edition, 2001.

[6] Andrei Z. Broder. Generating random spanning trees. In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*, pages 442–447. IEEE Computer Society, 1989.

[7] Daryl DeFord, Moon Duchin, and Justin Solomon. Recombination: A Family of Markov Chains for Redistricting. *Harvard Data Science Review*, 3(1), mar 31 2021. https://hdsr.mitpress.mit.edu/pub/1ds8ptxu.

[8] Moon Duchin and Douglas M. Spencer. Models, Race, and the Law. *The Yale Law Journal Forum*, 130:744–797, 2021.

[9] Christophe Garban, Gábor Pete, and Oded Schramm. The scaling limits of the minimal spanning tree and invasion percolation in the plane. *Ann. Probab.*, 46(6):3501–3557, 2018.

[10] G. Kirchhoff. On the solution of the equations obtained from the investigation of the linear distribution of galvanic currents. *IRE Transactions on Circuit Theory*, 5(1):4–7, 1958.

[11] Andrzej Komisarski. Nontransitive random variables and nontransitive dice. *Amer. Math. Monthly*, 128(5):423–434, 2021.

[12] Joseph B. Kruskal, Jr. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Amer. Math. Soc.*, 7:48–50, 1956.

[13] Ágnes Kúsz. The diameter of random spanning trees interpolating between the ust and the mst of the complete graph, 2024.

[14] Russell Lyons and Yuval Peres. *Probability on trees and networks*, volume 42 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, New York, 2016.

[15] Luca Makowiec, Michele Salvi, and Rongfeng Sun. Random spanning trees in random environment. arXiv preprint arXiv:2410.16830, 2024.

[16] Guy Melançon and Christophe Reutenauer. Lyndon words, free algebras and shuffles. *Canad. J. Math.*, 41(4):577–591, 1989.

[17] OEIS Foundation, Inc. The on-line encyclopedia of integer sequences, 2024. http://oeis.org/.

[18] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, USA, 3 edition, 2007.

[19] Christophe Reutenauer. *Free Lie algebras*, volume 7 of *London Mathematical Society Monographs. New Series*. The Clarendon Press, Oxford University Press, New York, 1993. Oxford Science Publications.

[20] R. Tyrrell Rockafellar. *Convex analysis*. Princeton Landmarks in Mathematics. Princeton University Press, Princeton, NJ, 1997. Reprint of the 1970 original, Princeton Paperbacks.

[21] Richard P. Savage, Jr. The paradox of nontransitive dice. *Amer. Math. Monthly*, 101(5):429–436, 1994.

[22] H. Steinhaus and S. Trybuła. On a paradox in applied probabilities. *Bulletin de l'Académie polonaise des sciences. Série des sciences mathématiques, astronomiques, et physiques*, 7(2):67–69, 1959.

[23] Reinhard Suck. Independent random utility representations. *Math. Social Sci.*, 43(3):371–389, 2002.

[24] Kristopher Tapp. On the minimum spanning tree distribution in grids. arXiv preprint arXiv:2401.17947, 2024.

[25] Stanisław Trybuła. On the paradox of three random variables. *Applicationes Mathematicae*, 4(5):321–332, 1961.

[26] Stanisław Trybuła. On the paradox of n random variables. *Zastos. Mat.*, 8:143–156, 1965.

[27] Jamie Tucker-Foltz and Peter Rock. MST Distribution. Replication repository. https://github.com/mggg/MST-Distribution.

[28] David Bruce Wilson. Generating random spanning trees more quickly than the cover time. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC, page 296–303, 1996.

A Probability and runtime

We can make a few additional observations about computing the probability of spanning trees using internal and external formulas. For complete graphs we can write $\#\partial(H)$ more simply, noting that if two connected components have a and b vertices, respectively, then there are ab missing edges between them. Since any missing edge in H is between some two of its components, we get

$$\#\partial(H) = e_2(\text{comp-size}(H))$$

where comp-size(H) is the multi-set of sizes of the connected components of H and, for a multi-set S of integers, $e_2(S)$ is the second elementary symmetric function of S:

$$e_2(x_1, \ldots, x_k) = \sum_{1 \le i < j \le k} x_i x_j.$$

Furthermore, all forests $F \subset K_n$ that are isomorphic to each other as graphs have the same probability by symmetry, so we only need to compute one number for each shape of forest.

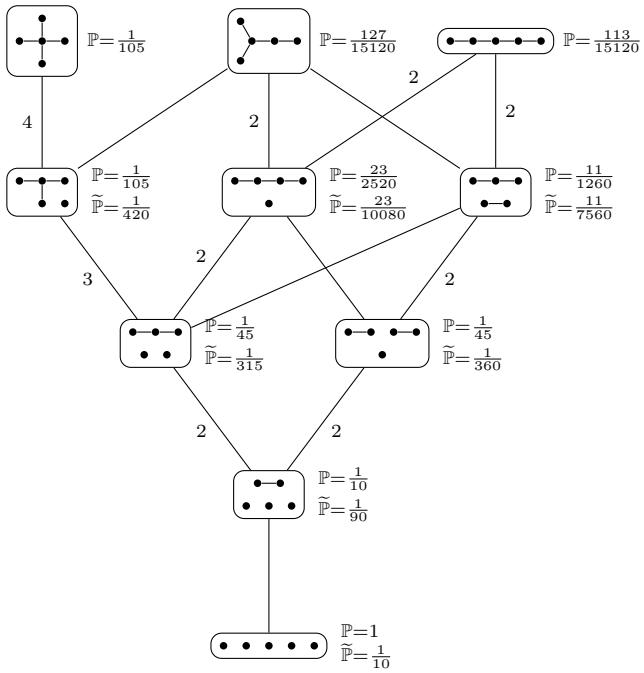

Figure 10: Probabilities for all forest shapes in K_5. Edges show inclusions $F' \subset F$ obtained by deleting one edge, with multiplicity according to the number of ways to delete the edge.

We can improve the running time of Kruskal's algorithm by combining some ingredients already assembled. Specifically, for a graph G and a forest $F \subset G$, consider the probability $\mathbb{P}(F)$ that F appears at some point in Kruskal's algorithm on G. We have an inductive formula for these probabilities. For $F \subset G$ a forest that is not a tree, define

$$\widetilde{\mathbb{P}}(F) \coloneqq \frac{\mathbb{P}(F)}{\#\partial(F)}.$$

Proposition A.1. *For a graph G and forest $F \subset G$,*

$$\mathbb{P}(F) = \sum_{e \in \text{Edges}(F)} \widetilde{\mathbb{P}}(F \setminus e).$$

This follows from the internal formula by breaking up the sum over S_{n-1} according to the last-added edge, $e_{\pi(n-1)}$. This suggests an efficient method for computing $\mathbb{P}(T)$ for a tree T: compute $\widetilde{\mathbb{P}}(F)$ for all

$F \subsetneq T$ in order from the fewest number of edges to the most, using Proposition A.1. This requires time $O(2^n)$, as opposed to time $O(n!)$ from directly applying the general internal formula.

If G is the complete graph K_n, then we can further simplify by noting that every labeling of a forest has the same probability. We can then apply Proposition A.1 on unlabeled forests as in Figure 10, at the cost of checking for graph isomorphisms.

B Extended example for path rotation algorithm

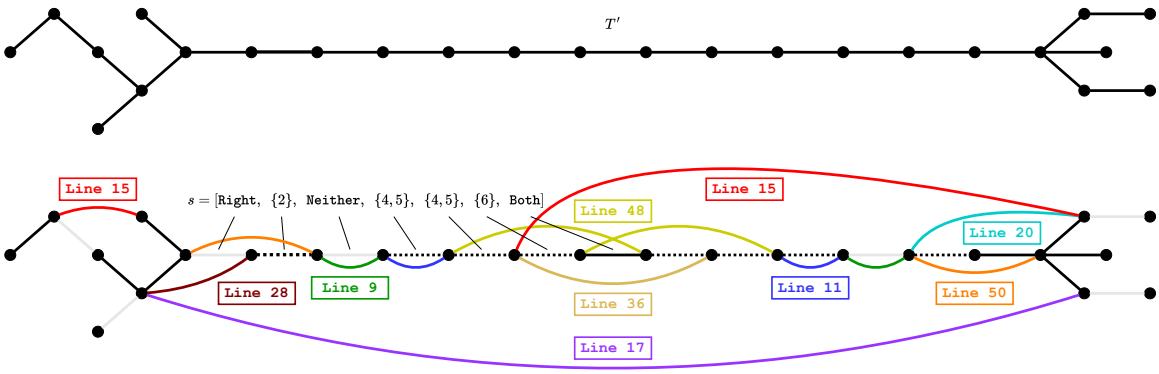

Figure 11: Above, a tree T' containing a path P' of length $\ell = 14$, giving $r = 7$ and six sets of paired edges. The state vector s has seven entries. Below, a possible state s midway through the inner for-loop of Algorithm 3.1. Solid black edges have been confirmed to be added to the tree, dashed black edges have been added with probability $\frac{1}{2}$, and gray edges have been confirmed to be excluded. The colored, curved edges are possible edges that could be processed on the next iteration. The effects of processing these edges are shown in Table 3.

After processing...	New state s, with updates colored	q	q'
(initially)	[Right, {2}, Neither, {4, 5}, {4, 5}, {6}, Both]	1/4	1/4
the pair of edges triggering Line 9	[Right, {2}, {3}, {4, 5}, {4, 5}, {6}, Both]	1/4	1/4
the pair of edges triggering Line 11	[Right, {2}, Neither, Both, {5}, {6}, Both]	1/4	1/4
either edge triggering Line 15	(does not matter)	0	0
the edge triggering Line 17	(no change)	1/4	0
the edge triggering Line 20	[Right, Right, Neither, {4, 5}, {4, 5}, {6}, Both]	1/8	1/8
the edge triggering Line 27	(no change)	1/4	0
the edge triggering Line 35	(does not matter)	0	0
the pair of edges triggering Line 48	[Right, {2}, Neither, {4, 5, 6}, {4, 5, 6}, {4, 5, 6}, Both]	1/16	1/16
the pair of edges triggering Line 44	[Right, Right, Neither, {4, 5}, {4, 5}, {6}, Both]	1/16	1/16
the pair of Line 9 edges, followed by the Line 15 edge on the top-right, which would trigger Line 22	(does not matter)	0	0

Table 3: States and conditional probabilities after processing each of the colored, curved edges for the example of the graph depicted in Figure 11. Note that, for the pairs of edges that trigger the conditions in Lines 9, 48, and 44, this would be the first time we have seen the given pair of edges; for the pair that triggers the condition in Line 11, it would be the second time.

C Hitting MST = UST with shifted intervals

C.1 Shifts and theta graphs

In the introduction, it was noted that when G is the square with a diagonal, then there exists s such that $M_{\mathsf{s}} = \text{UST}$.

In this section we prove that shifts suffice to recover the uniform distribution on a larger class of graphs called theta graphs.

Definition C.1. A *theta graph* $\theta(r, s, t)$ is formed by connecting two base vertices with disjoint paths of length r, s, and t, respectively, so that there are $m = r + s + t$ edges overall. We call these the R-path, the S-path, and the T-path, and refer to their edges as R-edges, S-edges, and T-edges; see left-hand side of Figure 12.

We can define shifted interval MST on theta graphs by using the same shifts on the edges of each type:

$$\text{Sh}(r, s, t) = \{(\alpha, \dots, \alpha, \beta, \dots, \beta, \gamma, \dots, \gamma) : (\alpha, \beta, \gamma) \in \text{Sh}(3)\}.$$

We note that every spanning tree of $\theta(r, s, t)$ must be missing exactly two edges: one edge each from two of the three paths in G. A tree of R type contains all of the R path but is missing one S-edge and one T-edge; likewise for trees of S and T type; see right-hand side of Figure 12. Given $\mathsf{s} \in \text{Sh}(r, s, t)$, let $p_R + p_S + p_T = 1$ be the probabilities that M_{s} is of type R, S, T, respectively. Any two trees of the same type occur with equal probability. Hence, every individual R-tree occurs with probability $\frac{p_R}{st}$; similarly for S-trees and T-trees. Thus the distribution is uniform if and only if $rP_R = sP_S = tP_T$, which occurs at exactly one (P_R, P_S, P_T), given by $P_R = u_R := st/(st + rt + rs)$, with similar expressions for P_S and P_T.

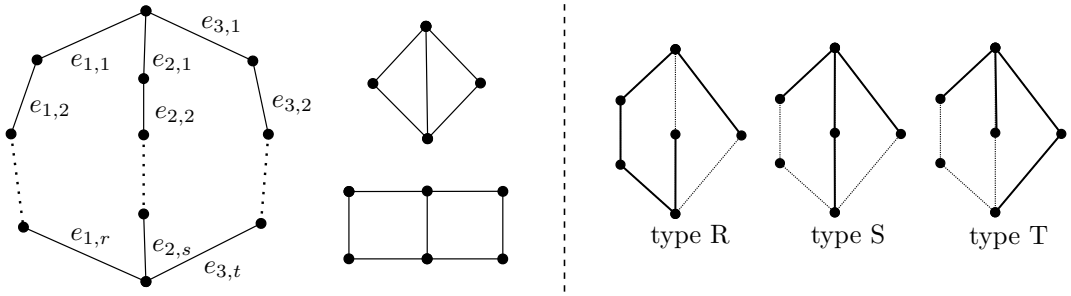

Figure 12: Left: The general theta graph $\theta(r, s, t)$ and two specific examples, $\theta(2, 1, 2)$ (the square with a diagonal) and $\theta(3, 1, 3)$. Right: Examples of trees of type R, S, and T for $\theta(3, 2, 2)$.

Theorem C.2 (UST on theta graphs)**.** *On theta graphs, the uniform distribution is achievable by shifted intervals: for any (r, s, t), there exists $\mathsf{s} \in \text{Sh}(r, s, t)$ such that $M_{\mathsf{s}} = \text{UST}$ on $\theta(r, s, t)$.*

Proof. Consider $\Delta(S_3)$, the triangle (2-simplex) parameterizing $P_R + P_S + P_T = 1$. There is a map $\text{Sh}(3) \to \Delta(S_3)$ given by the construction above, where the weight interval for R edges is $[\alpha, \alpha + 1]$ and likewise S edges and T edges have weight drawn from the β, γ intervals. As we move around the boundary of Sh(3) (shown in Figure 8), we traverse the boundary of $\Delta(S_3)$ one time. So the map is continuous and is degree 1 on the boundary (as a map $S^1 \to S^1$), which implies that it is surjective. □

In fact, we can say more. Consider the external formula from Theorem 3.4. Under MST_0, the probability of obtaining a tree of R-type is $m_R := \frac{1}{r+s} \cdot \frac{1}{r+s+t} + \frac{1}{r+t} \cdot \frac{1}{r+s+t}$, and likewise for S and T. Under the wlog assumption $r \geq s \geq t$, we have

$$u_R - m_R = \frac{rst(r^2 - st)}{(r + s)(r + t)(rs + rt + st)},$$

which is strictly positive unless $r = s = t$.

Corollary C.3. $\text{MST}_0 = \text{UST}$ *if and only if* $(r, s, t) = (r, r, r)$.

With a bit more work one can show that the map $(\alpha, \beta, \gamma) \mapsto (P_R, P_S, P_T)$ is non-singular on the interior of the region defined by $0 = \alpha \leq \beta \leq \gamma \leq 1$, which tells us that there is in fact a unique way to hit (u_R, u_S, u_T) with this class of shifts.

Finally, we note that the work in this section extends readily to *generalized theta graphs* $\theta(k; r_1, \ldots, r_k)$, which can have any number $k \geq 3$ of paths connecting a pair of base vertices; only the notation gets worse. In particular, the following hold: for a generalized theta graph $\theta(r_1, \ldots, r_k)$, UST = MST_0 if and only if $r_1 = \cdots = r_k$.

C.2 Snowmen and θ-surgery graphs

For this section, we will call a theta graph $\theta(r, s, t)$ with $(r, s, t) \neq (r, r, r)$ a *snowman*; any graph that does not contain a snowman as a subgraph is called *snowman-free*.

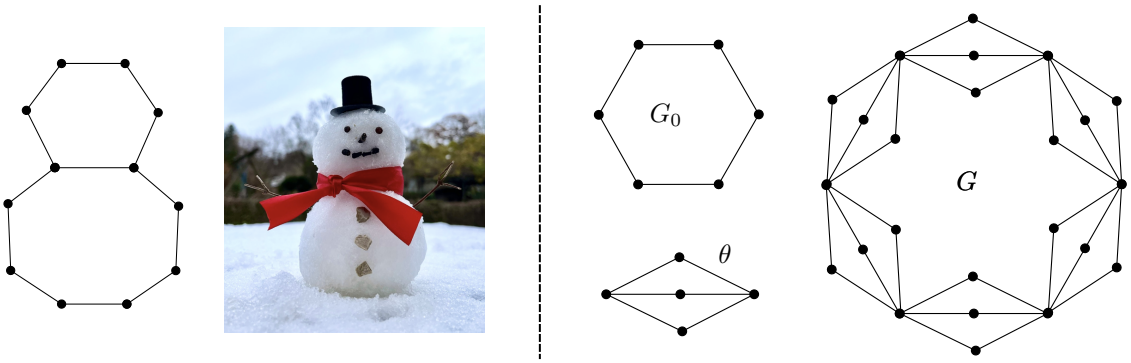

Figure 13: Left: The snowman graph $\theta(7, 1, 5)$. Right: the θ-surgery graph G built from the cycle G_0 of length 6 and the θ-graph $\theta = \theta(2, 2, 2)$ (that is, $r = 2$, $k = 3$).

Proposition C.4 (Snowman-free graphs)**.** *For a snowman-free graph G,* $\text{MST}_0 = \text{UST}$.

Proof. First, observe that every snowman-free graph G is either a tree or a finite collection of generalized theta graphs $\theta(r, r, r, ..., r)$ (for various r) that are connected by paths of edges so that no new loops arise; possibly with additional leaves. The condition *no new loops arise*, more formally, means that if we collapsed each theta graph in a snowman-free graph, the resulting graph would be a tree; see Figure 14. For a snowman-free graph G, every spanning tree must contain all edges of the connecting paths between the theta graphs as well as all leaves. So the fact that $\text{MST}_0 = \text{UST}$ on a theta graph $\theta(r, r, r)$ (Corollary C.3) implies that $\text{MST}_0 = \text{UST}$ on G. □

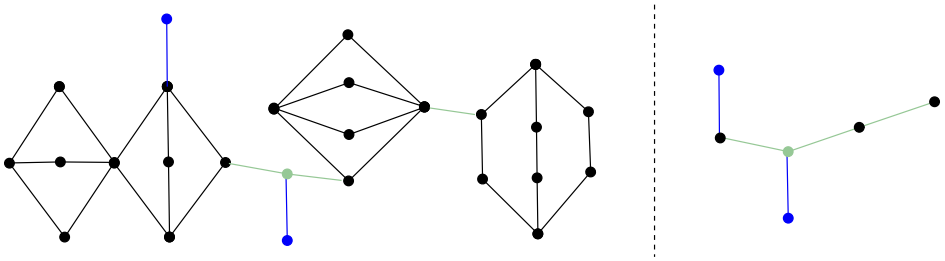

Figure 14: Condition *no new loops arise*. The left-hand side shows the snowman-free graph G consisting of four theta graphs (black), connected by paths (green), and two additional leaves (blue). The right-hand side shows the tree obtained from G by collapsing the theta graphs into points.

However, the converse of Proposition C.4 is false. To see this, consider the following construction: Let G_0 be a graph for which $\text{MST}_0 = \text{UST}$ and let $r, k \in N$. Define G to be the graph obtained from G_0 by

replacing each edge in G_0 by k paths of r edges. In other words, we replace every edge of G_0 by a $\theta(r, ..., r)$, with the two poles of the theta graph placed at the endpoints of the replaced edge. We call such a graph G a *θ-surgery graph*.

Proposition C.5. *For a θ-surgery graph G, $\mathrm{MST}_0 = \mathrm{UST}$.*

Figure 13 depicts an example of a θ-surgery graph. Notice that this graph fails to be snowman-free (for instance, it contains a $\theta(2, 10, 10)$ subgraph) but satisfies $\mathrm{MST}_0 = \mathrm{UST}$ by Proposition C.5. In an exactly similar way, we can define a *shift surgery graph* G in which the base graph G_0 and the surgery graph G_1 have the property that shifts can recover UST. This property then passes to the the compound graph G.

D Details on quadrature constructions

To prove Theorem 5.9, we first observe that a quadrature scheme of degree k gives a way to make one measure discrete out of several copies of the uniform measure.

Proposition D.1. *If $Y = \sum_{i=1}^{r} \alpha_i \delta_{x_i}$ is a degree-k quadrature scheme and* Leb *is the uniform measure on $[0, 1]$, then the measures*

$$X_i \coloneqq \begin{cases} \mathrm{Leb} & 1 \le i \le k \\ Y & i = k+1 \end{cases}$$

is a product measure giving the uniform distribution on S_{k+1}.

Proof. For $0 \le i \le k$ and $0 \le x \le 1$, let $P_j(x)$ be the probability that, in k draws from Leb, exactly j of them are less than x. This makes $P_j(x)$ a polynomial in x of degree k. The average value of P_j is $1/(k+1)$, because it is equal to the probability that, in $k+1$ draws from Leb, the last draw is in rank j. Thus, by the quadrature property, the probability that X_{k+1} appears as the jth letter in a draw from the $(X_i)_{i=1}^{k+1}$ is

$$\sum_i \alpha_i P_j(x_i) = \int_0^1 P_j(x)\, dx = \frac{1}{k+1},$$

as desired. By symmetry, the remaining letters occur in each possible order equally likely. □

Remark D.2. Since the polynomials $P_j(x)$ span the space of degree k polynomials, the converse of Proposition D.1 is also true (after suitable normalization).

To go further, we formulate a criterion to replace pieces of measures without disturbing the overall distribution on S_m.

Proposition D.3. *Suppose we are given a non-colliding product measure $X = (X_1, \ldots, X_m)$ and an open interval $I \subset \mathbb{R}$. Suppose that $X_1, \ldots, X_k$ have positive mass on I and the remainder $X_{k+1}, \ldots, X_m$ do not. For $j = 1, \ldots, k$, decompose X_j as*

$$X_j = Y_j + R_j$$

where $Y_j(I) = X_j(I) > 0$, so that R_j is the remainder. Let Y'_j for $j = 1, \ldots, k$ be any other measures with support contained in I so that

- *$Y_j(I) = Y'_j(I)$;*
- *the $(Y'_j)_{j=1}^k$ are non-colliding on I; and*
- *$Y = (Y_j)_{j=1}^k$ and $Y' = (Y'_j)_{j=1}^k$ give the same distribution on S_k.*

Then the product measure $X' = (X'_1, \ldots, X'_m)$ defined by

$$X'_j \coloneqq \begin{cases} Y'_j + R_j & j \le k \\ X_j & k < j \le m \end{cases}$$

gives the same distribution on S_m as $(X_j)_{j=1}^m$.

Proof. Consider a draw from X or X'. Some subset $T \subset \{1, \dots, k\}$ of the X_j/X'_j variables comes from Y_j/Y'_j rather than R_j. Since $Y_j(I) = Y'_j(I)$, the probability of each subset T occurring is equal. If we condition on a particular T occurring, since Y and Y' induce the same measure on S_k, they also induce the same measure on orderings of T. Putting these together we get the result. □

Proof of Theorem 5.9. Start with $k+1$ copies of Lebesgue measure Leb. By Proposition D.1, we can replace the last copy with the discrete measure $\sum_{i=1}^{r} \alpha_i \delta_{x_i}$. Set also $x_0 = 0$ and $x_{r+1} = 1$. For each non-empty interval among the $I_i = (x_i, x_{i+1})$, the word $w^{x_{i+1}-x_i}$ gives a discrete measure with the same total mass in each variable as $\mathrm{Leb}(I_i)$; rescale the domain of $w^{x_{i+1}-x_i}$ so that it lies entirely within I_i. Since w was assumed to be uniform on k letters, we thus form measures Y'_i as in the hypotheses of Proposition D.3. If we do this for each $0 \le i \le r$, we get a discrete measure corresponding to the word U in the statement. □

Proof of Proposition 5.10. Applying this inductive construction, we use the quadrature schemes discussed above.

- *Gauss-Radau* quadrature of degree $2r-2$ has r nodes and weights, and r interval widths, giving a $(2r-2, 1)$-uniform word; and
- *Gauss-Lobatto* quadrature of degree $2r-3$ has r nodes and weights and $r-1$ interval widths, giving a $(2r-3, 1)$-uniform word.

This lets us get an appropriate scheme for each parity. Defining $L(m)$ recursively by $L(1) = 1$ and

$$L(m) = \begin{cases} \frac{m-1}{2} \cdot L(m-1) + \frac{m-1}{2}, & m \text{ odd} \\ \frac{m-2}{2} \cdot L(m-1) + \frac{m}{2}, & m \text{ even}, \end{cases}$$

we have built uniform words of length $L(m)$ on m symbols for every m. □

E Details for dimension bounds

We begin with an explicit proof of the even/odd identities discussed in §5.4, which are constraints on P_m.

Lemma E.1 (General independence constraints)**.** *Fix an integer $k \ge 2$ and consider a collection of independent events events $\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_k$, each partitioned into disjoint sets $\mathcal{E}_i = \mathcal{E}_i^+ \cup \mathcal{E}_i^-$. Then Equation (EO) holds:*

$$\prod_{\substack{s\colon [k]\to\{+,-\} \\ \text{such that } |s^{-1}(+)| \text{ is even}}} \mathbb{P}\left[\bigwedge_{i\in[k]} \mathcal{E}_i^{s(i)}\right] = \prod_{\substack{s\colon [k]\to\{+,-\} \\ \text{such that } |s^{-1}(+)| \text{ is odd}}} \mathbb{P}\left[\bigwedge_{i\in[k]} \mathcal{E}_i^{s(i)}\right]. \tag{EO}$$

Proof. By independence, we may expand each conjunction as a product of probabilities across i. Switching the order of the two products, the left-hand side of (EO) then becomes

$$\prod_{i\in[k]} \prod_{\substack{s\colon [k]\to\{+,-\} \\ \text{such that } |s^{-1}(+)| \text{ is even}}} \mathbb{P}\left[\mathcal{E}_i^{s(i)}\right]$$

Next, for each fixed choice of i we may decompose the second product into two pieces, based on whether $s(i) = +$, obtaining

$$\prod_{i\in[k]} \left(\prod_{\substack{s\colon [k]\setminus\{i\}\to\{+,-\} \\ \text{such that } |s^{-1}(+)| \text{ is odd}}} \mathbb{P}\left[\mathcal{E}_i^{+}\right] \right) \left(\prod_{\substack{s\colon [k]\setminus\{i\}\to\{+,-\} \\ \text{such that } |s^{-1}(+)| \text{ is even}}} \mathbb{P}\left[\mathcal{E}_i^{-}\right] \right).$$

Observe that the probabilities do not depend on the choices of s; all that matters is the numbers of terms. And since $k \geq 2$, the two products over s each have the same number of terms, namely, 2^{k-1}. Hence, we may rewrite the expression as

$$\prod_{i \in [k]} \left(\mathbb{P}\left[\mathcal{E}_i^+\right]\right)^{(2^{k-1})} \left(\mathbb{P}\left[\mathcal{E}_i^-\right]\right)^{(2^{k-1})}.$$

We may analogously rewrite the right-hand side and obtain the exact same expression. This concludes the proof. □

For the gradient computation itself, it is helpful to use the notation of shuffle products first introduced by Eilenberg-Mac Lane in the 1950s. For two disjoint orderings u_1 and u_2 of distinct elements from $\{1, \ldots, m\}$, the *shuffle product* $u_1 \sqcup\!\sqcup u_2$ is the formal sum of all ways of interleaving the numbers, as in the example given previously:

$$(1 < 2) \sqcup\!\sqcup (3 < 4) = (1 < 2 < 3 < 4) + (1 < 3 < 2 < 4) + (3 < 1 < 2 < 4) + (3 < 1 < 4 < 2) + (3 < 4 < 1 < 2).$$

The *Lie bracket* $[u_1, u_2]$ is the concatenation of u_1 with u_2, minus the concatenation of u_2 with u_1. For example,

$$\big[(1 < 2), (3 < 4)\big] = (1 < 2 < 3 < 4) - (3 < 4 < 2 < 1).$$

We extend the shuffle product and commutators linearly, distributing over addition. We also adopt left-associated notation for nested brackets by $[u_1, u_2, u_3] \coloneqq [[u_1, u_2], u_3]$, and so on, with the convention that $[u] = u$.

Definition E.2 (Ordering map and Lie shuffle basis)**.** Define a linear map $F \colon S_m \to \mathbb{R}^{O(m)}$ as follows. Put each permutation into a lexicographic normal form where each of its cycles c_i is ordered with its lowest index first, and then the cycles are ordered by their lowest indices. We denote the elements of each cycle using superscript notation, $c_i = (c_i^1, c_i^2, \ldots, c_i^{r_i})$, where r_i is the length of the i^{th} cycle. For an individual cycle in lexicographic normal form, we define $F(c_i) \coloneqq [c_i^1, c_i^2, \ldots, c_i^{r_i}]$. Then for a general permutation, F is the shuffle product of the brackets for the constituent cycles. Then $\{F(\pi) : \pi \in S_m\}$ is called the *Lie shuffle basis*.

Throughout the rest of this section, we maintain the convention of writing cycles with the lowest index first, because it is important that a fixed element be left-most so that it is deepest in the nested brackets, which is used in the bracket arithmetic. We now prove Lemma 5.12, restated below.

Lemma 5.12 (Gradient expression)**.** *Let π be a permutation composed of disjoint cycles $c_1, c_2, \ldots, c_\ell$, and let $u_i = F(c_i)$. Then there are independent events $\mathcal{E}_i = \mathcal{E}_i^+ \cup \mathcal{E}_i^-$ for each $1 \leq i \leq \ell$ where c_i is nontrivial (has length at least 2) such that the gradient of Equation (EO) at the uniform distribution is a scalar multiple of $u_1 \sqcup\!\sqcup u_2 \sqcup\!\sqcup \cdots \sqcup\!\sqcup u_k$.*

Proof. For each nontrivial cycle c_i with length r_i, we define events $\mathcal{E}_1^{\pm}$ as follows. Consider the expansion of $F(c_i)$; each of the $2^{r_i - 1}$ terms (which are orderings of the indices appearing in the cycle) appears at most once, with coefficient $+1$ or -1. Let $\mathcal{E}_i^+$ (resp. $\mathcal{E}_i^-$) be the event that those indices occur in an ordering with coefficient $+1$ (resp. -1), so that at the uniform distribution $\mathbb{P}(\mathcal{E}_i^+) = \mathbb{P}(\mathcal{E}_i^-) =: p_i$ (with $p_i = 2^{r_i - 2}/(r_i)!$), and set $\mathcal{E}_i = \mathcal{E}_i^+ \vee \mathcal{E}_i^-$ be the union of these events.

We now look at the gradient of $Q \coloneqq \mathit{LHS} - \mathit{RHS}$ of (EO) with respect to these events, which by Lemma E.1 is a cotangent vector to P_n. For an arbitrary ordering $u \in O(m)$, the variable x_u appears on at most one side in Q. It appears at all only if it satisfies all $\mathcal{E}_i$, and which side it appears on is determined by the parity of the number of $\mathcal{E}_i^+$ it satisfies; this is the same as the sign of x_u in $F(\pi)$. The coefficient of x_u in the *gradient* of Q is determined by the other probabilities it is multiplied by on the same side of (EO). Since each such event $\bigvee \mathcal{E}_i^{s(i)}$ has the same probability $\prod p_i$, all these coefficients are the same, giving the desired scalar multiple. □

Finally, we state and prove a fact used in our arguments about the dimension of P_m, namely that it suffices to find a single point giving the right dimension bound.

Definition E.3 (Generic point)**.** A *generic point* of a simplex Δ^n is a point where the coordinates are algebraically independent. More generally, in an (irreducible) semi-algebraic set S defined over $\mathbb{Q}$, a generic point is a point that satisfies no algebraic relations with rational coefficients other than those satisfied by every element of S.

Recall that an irreducible semi-algebraic set like P_m is in general singular, but always has the structure of a stratified space as a union of strata that are smooth manifolds (with boundary) of possibly varying dimension. The *dimension* of the semi-algebraic set is the maximal dimension of a stratum, and all generic points are necessarily in this stratum.

Lemma E.4 (Dimension at generic points)**.** *Let $M(x_1, x_2, \ldots, x_m)$ be a matrix-valued function where each entry is polynomial in the m variables. If there exists a point p such that $M(p)$ has rank r, then at all generic points x, $M(x)$ has rank at least r.*

More generally, the same is true for an algebraic function M on an irreducible semi-algebraic set S.

Proof. Since $M(p)$ has rank r, we know that $M(x)$ has an $r \times r$ sub-matrix $S(x)$ that has full rank at p; i.e., $\det(S(p)) \neq 0$. Note that $\det(S(x))$ is a polynomial function of r^2 variables which is nonzero at p. Therefore, we cannot have $\det(S(x)) = 0$ at generic points, for otherwise $\det(S(x))$ would be the zero polynomial. This means that, at generic points x, $S(x)$ is invertible, implying that $M(x)$ has rank at least r.

The statement about a general semi-algebraic set S follows similarly; we give an argument that works in our case for completeness. Working in coordinates, suppose we have $S \subset \mathbb{R}^m$, and that M is a polynomial function on $\mathbb{R}^m$. Then a generic point x in S is not usually generic in $\mathbb{R}^m$; but by definition any equation like those above that is zero at x is also zero at every point p. □

We will apply this lemma to help us with both lower and upper bounds for $\dim(P_m)$. For the lower bound, we consider maps *into* P_m and for the upper bound we consider maps *from* P_m.

For the lower bound, we let M be the Jacobian of the word map from a product of simplices to P_m. The rank of M at any point gives a lower bound on the rank of M at a generic point, which is a lower bound on $\dim P_m$ (with equality at the last step for long enough word maps).

For the upper bound, we let the matrix M be gradients of independence constraints, as a function on $\mathbb{R}^{O(m)} \supset P_m$. The rank of M at the uniform distribution gives a lower bound on the rank of M at any point in P_m, as desired. (Note that the rank of M at a generic point in $\mathbb{R}^{O(m)}$ might be larger still, but that is not relevant to P_m itself.)