Knot Probabilities in Random Diagrams

Jason Cantarella, Harrison Chapman

University of Georgia, Mathematics Department, Athens GA

Matt Mastin

MailChimp, Atlanta, GA

We consider a natural model of random knotting– choose a knot diagram at random from the finite set of diagrams with n crossings. We tabulate diagrams with 10 and fewer crossings and classify the diagrams by knot type, allowing us to compute exact probabilities for knots in this model. As expected, most diagrams with 10 and fewer crossings are unknots (about 78% of the roughly 1.6 billion 10 crossing diagrams). For these crossing numbers, the unknot fraction is mostly explained by the prevalence of "tree-like" diagrams which are unknots for any assignment of over/under information at crossings. The data shows a roughly linear relationship between the log of knot type probability and the log of the frequency rank of the knot type, analogous to Zipf's law for word frequency. The complete tabulation and all knot frequencies are included as supplementary data.

Keywords: random knots; random knot diagrams; immersion of circle in sphere; knot probabilities

1. INTRODUCTION

The study of random knots goes back to the 1960's, when polymer physicists realized that the knot type of a closed (or ring) polymer would play an important role in the statistical mechanics of the polymer [**?**]. Orlandini and Whittington [**?**] give a comprehensive survey of the development of the field in the subsequent years. Most random knot models are based on closed random walks, and there are many variants corresponding to different types of walks (lattice walks, random walks with fixed edgelengths, random walks with variable edgelengths, random walks with different types of geometric constraints such as fixed turning angles). For almost all of these models, certain phenomena have been rigorously established– for instance, the probability of knotting goes to 1 exponentially fast as the size of the walk increases[**? ? ? ? ? ?**]. However, it has been difficult to prove more informative theorems.

One way to look at the problem is these knot models are all models of random space curves, and it is quite hard to relate the three-dimensional shape of a space curve to its knot type. One can in principle express the finite-type invariants as (complicated) integrals in the spirit of [**?**], but so far computing the expected value of these integrals has been too difficult.

The point of this paper is to look at knot diagrams not as convenient combinatorial representations of space curves, but as a probability space in their own right. This is basically the same model of random knotting as [**?**] or [**?**]: the objects are equivalence classes of immersions of S^1 into

S^2 (as in Arnol'd [**?**]) paired with assignments of orientation and crossing signs. There are other quite different combinatorial models of random knotting in the literature– see [**?**], [**?**], or [**?**].

Previous authors [**?**] have sampled a slightly different version of this space by using an algorithm of Schaeffer and Zinn-Justin [**?**] to uniformly sample a space of marked link diagrams and keeping only the knots. We wanted to get exact values for probabilities and also to study the probability of very rare knots, so a sampling approach was not appropriate for our purposes. Instead, we carried out a complete enumeration of knot diagrams up to 10 crossings using the graph theory software *plantri* [**? ?**].

This paper is primarily computational. While we give a proof that our algorithm for enumerating diagrams is correct, our main contribution is the data set of diagrams itself. To generate the diagrams, we first enumerated the unoriented immersions of S^1 into S^2 in two different ways, as described below. We call these immersions knot shadows[1]. The sets of shadows generated in each way were identical, and we checked the number of shadows against counts by Arnol'd for $n \leq 5$ [**?** , page 79], sequence A008989 in the Online Encyclopedia of Integer Sequences [**?**], and later enumerations of Kapolnái et al. [**?**] and Coqueraux et al. [**?**]. We then enumerate assignments of crossing information and orientation for each shadow up to diffeomorphisms of S^2 preserving the orientation of the curve. (This is more complicated than it first seems; if the diagram has a symmetry, not all assignments of crossing information are different.) Finally, we compute the knot type for each assignment of crossing and orientation. These computations are new and fairly substantial. Enumerating the 1.6 billion 10-crossing knot diagrams and computing their HOMFLY-PT polynomial took several thousand hours of CPU time on the Amazon EC2 cloud computing service.

Figure 15 shows the relative frequency of all the knot types we observed, sorted by rank order among knots. When plotted on a log-log plot, we see that the plot is roughly linear across 9 orders of magnitude– giving some evidence for a roughly Zipfian distribution of knot types. In such a distribution, the k-th most frequent knot type would have a probability proportional to k^{-s} for some s. It would be very interesting to know what happens for large n. This distribution has been previously observed for knotting in self-avoiding polygon models by Baiesi, et al. [**?**].

How many of these diagrams represent nontrivial knots? Relatively few. Even for 10-crossing diagrams, the proportion of unknots is about 77%. For these crossing numbers, this is largely explained by the surprising frequency of "tree-like" knot shadows, studied by Aicardi [**?**], (Figure **??**) for which the knot type does not depend on the assignment of crossing information– the resulting knot is *always* the unknot. These diagrams are surprisingly common: 42.05% of 8-crossing diagrams are treelike, which explains about half of the unknots among 8-crossing diagrams (84% of 8-crossing diagrams are unknots). The remaining unknots are almost all connect sums of treelike diagrams with the unique prime 3- or 4-crossing shadow. We will show that a simple analysis (Proposition **??**) based on such diagrams gives a lower bound of 77% for the unknot fraction in 8-crossing diagrams, explaining more than 90% of all unknots in this class.

[1] Kauffman [**?**] calls these knot *universes*.

Tree-like diagrams are composite diagrams where each prime summand is one-crossing knot diagram ∞. In fact, random diagrams are in general highly composite [**?**] (Figure **??**) with many simple summands. It remains an interesting open question to try to characterize the asymptotic distribution of sizes and numbers of connect summands in a random diagram for large n.

In general, counting knot diagrams is made significantly more complicated by the existence of diagram symmetries; for instance, a correction factor due to symmetries is the main technical difference between this natural model and the model of [**?**]. Our data shows that this difficulty rapidly disappears as the number of crossings increases; the average size of the automorphism group for a random 10 crossing knot diagram, for instance, is $846929/823832 \simeq 1.028$. This means that we expect this model to behave very much like the "rooted" model of [**?**] for crossing numbers 11 and above; one of us (Chapman) has shown this is true for sufficiently many crossings [**?**].

2. DEFINITIONS

We begin with some definitions.

Definition 1. We define a *link shadow* L with n vertices to be an equivalence class of generic smooth immersions of a collection of oriented circles into S^2 with n intersections up to diffeomorphism of the sphere (which may not be orientation-preserving on either the sphere or the circles; Arnol'd [**?**] calls this equivalence "OU").

Examples of link shadows are shown in Figure 1. In our experiments, each link shadow will be represented (non-uniquely) by a combinatorial object called a PD-code.

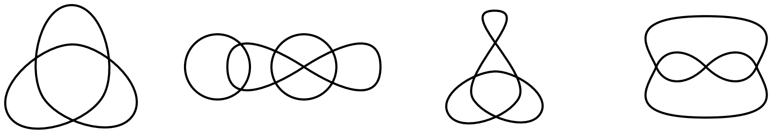


FIG. 1: Examples of link shadows.

Definition 2. A **pd-code** $\mathfrak{L}$ with n-vertices is a list of n cyclically ordered quadruples called vertices which partition the set of signed edges $\{\pm 1, \ldots, \pm 2n\}$ so that in each quadruple every pair of non-adjacent edges have opposite signs.

Two pd-codes are **pd-isomorphic** if there exists a bijection of vertices and edges which respects the partition of signed edges into vertices, including globally preserving or reversing the cyclic ordering of edges around vertices. A pd-isomorphism must take both signs of an edge $\pm i$ to both signs of another edge $\pm j$, but can take $+i$ to $-j$ and vice versa.

For example $\mathfrak{L}_0 = ((+4,-2,-5,+1),(+2,-6,-3,+5),(+6,-4,-1,+3))$ is a valid pd-code of three vertices and six edges, as each of $\pm 1,\dots,\pm 6$ occurs once, and each pair of non-adjacent labels in a single quadruple (such as $+4,-5$ and $-2,+1$ in the first quadruple) has opposite signs.

This definition is similar to the "combinatorial maps" of Walsh and Lehman [**?**]. Indeed, a pd-code $\mathfrak{L}$ describes an element σ of the permutation group on the $4n$ letters $\{\pm 1, \pm 2, \cdots, \pm 2n\}$ which is a product of n disjoint cycles of size 4, together with an implicit permutation $\tau = (+1 - 1)(+2-2)\cdots(+2n-2n)$ and a "consistent" choice of orientations on the edges. This (and other) formulations are explored by Coquereaux, et al. [**?**].

Definition 3. Given a pd-code $\mathfrak{L}$ and a signed edge e, we can define the *successor edge* $s(e)$ to be minus the edge immediately preceding e in the cyclic ordering of the quadruple where e occurs. The faces of a pd-code $\mathfrak{L}$ are the orbits of the successor map.

In the pd-code $\mathfrak{L}_0$, for example, -2 occurs in the quadruple $(+4,-2,-5,+1)$, so $s(-2) = -(+4) = -4$. Similarly, $s(s(-2)) = s(-4) = -(+6) = -6$, and $s(s(s(-2)) = s(-6) = -(+2) = -2$. Thus $(-2,-4,-6)$ is a face of $\mathfrak{L}_0$.

Proposition 4 (Mastin, [**?**])**.** *The vertices, edges, and faces of $\mathfrak{L}$ form a cell-complex structure on a 2-dimensional surface $C(\mathfrak{L})$. Given two pd-codes $\mathfrak{L}$ and $\mathfrak{L}'$, the pd-codes are pd-isomorphic $\iff$ the cell-complexes $C(\mathfrak{L})$ and $C(\mathfrak{L}')$ are isomorphic (the isomorphism may reverse orientation).*

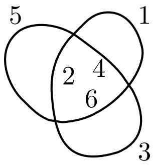
We can then define

Definition 5. The *genus g* of a pd-code $\mathfrak{L}$ is the genus of the associated cell complex $C(\mathfrak{L})$. It is given by $V - E + F = 2 - 2g$, where V, E, and F count the vertices, edges, and faces of $\mathfrak{L}$. If a pd-code has genus 0, it is *planar*.

It is another theorem of Mastin that

Proposition 6 (Mastin [**?**])**.** *There is a bijection between n-vertex link shadows and n-vertex* planar *pd-codes up to pd-isomorphism.*

The bijection itself is easy to construct: the vertices of the pd-code represent intersections of the circles, the edges represent the sections of circles between intersections (signs represent orientation along the circle), and the cyclic ordering of the vertices is the counterclockwise order of circle arcs at each intersection point. An example of a link shadow and the associated pd-code is shown in Figure 2.



$$\{[+1,-5,-2,+4],[+2,+5,-3,-6],[-1,-4,+6,+3]\}$$

FIG. 2: A three-crossing shadow and its pd-code. There is only one component. Note that we may omit directional arrows as the orientation can be inferred from the ordering of the edge labels.

It is clear that there are finitely many n-vertex planar pd-codes, and so finitely many equivalence classes of pd-codes up to isomorphism. By the proposition, there are therefore finitely many n-vertex link shadows. This will be our initial probability space.

Given a link shadow $\mathfrak{L}$ we can define a link diagram by

Definition 7. A *link diagram* is a link shadow where each intersection is decorated with over-under information for the circles meeting at the intersection. We call these intersections *crossings*. The equivalence relation for diagrams is the *diagram isomorphism*: a diffeomorphism (not necessarily orientation-preserving) of S^2 to S^2 which respects over-under information and preserves the orientation of the curves.

It is clear that there are at most $2^{\#\text{ crossings}}$ link diagrams associated to a given link shadow, but that this number will be smaller if there are nontrivial shadow automorphisms.

Definition 8. In the *random diagram model*, a random n-crossing knot is selected uniformly from the counting measure on the finite set of one-component n-crossing link diagrams.

This is the smallest probability space including all the knot diagrams that one can define; however, note that we could expand the probability space by doing things like choosing planar, rather than spherical embeddings (hence labeling a face as the exterior) or choosing a basepoint on each component. These amount to making the rules for diagram and shadow isomorphism more strict, increasing the number of equivalence classes of diagrams.

3. ENUMERATING SHADOWS

Our first goal is to enumerate the link shadows. This section describes our two enumeration algorithms. Each is built on the same computational foundation; a library which allows us to manipulate pd-codes. In this library, we provide

Definition 9. A *pdstor* is an ordered collection of pd-isomorphism types of pd-codes. We define the operation of adding a pd-code to a pdstor as adding the pd-code if it does not belong to an existing pd-isomorphism type (that is, is not pd-isomorphic to a pd-code present in the pdstor) and ignoring the pd-code otherwise.

Our implementation uses a combination of hashing and brute-force comparison to check whether a given pd-code is already pd-isomorphic to something in the database. Details of the pd-isomorphism check are provided in Appendix A for the curious reader.

We are left with the problem of creating a set of input pd-codes to the pdstor which are guaranteed to cover all n-vertex link shadows. We will then separate out the one-component knot shadows by searching the pdstor.

3.1. Dual quadrangulations and connect sums

Brinkmann et al. [**?**] provide an algorithm to enumerate all of the simple embedded planar quadrangulations of S^2, up to embedded isomorphism. A quadrangulation is a planar graph where each face has four edges, as in Figure 3.

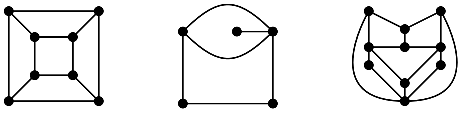


FIG. 3: This figure shows examples of quadrangulations.

The dual graph to a quadrangulation is a connected 4-regular embedded planar multigraph (graphs embedded on a surface are sometimes called “maps”), as shown at left in Figure 4. In other words, the dual graph defines a link shadow. This is almost enough to enumerate shadows. However, not every link shadow is obtained in this way: if the quadrangulation is simple, no pair of faces in the link shadow share more than one edge. As we can see in Figure 4, this property is not true for every link shadow.

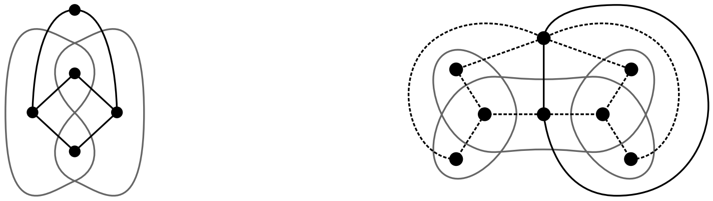


FIG. 4: This figure shows examples of quadrangulations and their dual graphs, which are link shadows. The quadrangulation at left is simple; the corresponding shadow is prime. The quadrangulation at right is non-simple; the corresponding shadow is composite.

We need a familiar idea in a slightly new guise: prime and composite shadows.

Definition 10. Every pair of faces in a *prime shadow* shares at most one edge; all other link shadows are *composite shadows*.

This is not the same thing as prime and composite links– we can assign crossings and get a prime diagram of a composite link or a composite diagram of a prime link– but the theory is quite similar. The biggest difference, as we will see below, is that while connect sum for knot types is associative, connect sum for knot shadows is not.

Definition 11. Given edges e and e' in pd-codes $\mathfrak{L}$ and $\mathfrak{L}'$, we can construct a new pd-code $\mathfrak{L}\#_{e,e'}\mathfrak{L}'$. The edges of $\mathfrak{L}\#_{e,e'}\mathfrak{L}'$ are the edges of $\mathfrak{L}$ together with the edges of $\mathfrak{L}'$. The vertices of $\mathfrak{L}\#_{e,e'}\mathfrak{L}'$ are the vertices of $\mathfrak{L}$ together with the vertices of $\mathfrak{L}'$ with one change: $+e$ and $+e'$ are swapped ($-e$ and $-e'$ stay in their original positions).

The effect of the definition is to switch the positions of the heads of the edges e and e' in the crossings where they occur. We are not trying to be too specific about notation because the edges will need to be relabeled in the new pd-code (see our code for one possible implementation). We can then prove

Proposition 12. *If $\mathfrak{L}$ and $\mathfrak{L}'$ are planar, then so is the pd-code $\mathfrak{L}\#_{e,e'}\mathfrak{L}'$.*

Proof. The new collection of vertices clearly still partitions the new collection of signed edge labels, since the same edge labels occur in vertices as in the original pd-codes $\mathfrak{L}$ and $\mathfrak{L}'$. We are swapping a pair of $+$ labels, so the rule on non-adjacent signs is still obeyed in the new pd-code. This means that we need only check the genus.

If we think of faces of a pd-code as lists of signed edges, the effect of the connect sum operation is to concatenate two pairs of these lists. As Figure 5 shows, the successor of $+e$ in $\mathfrak{L}$ is replaced by the successor of $+e'$ in $\mathfrak{L}'$, and the chain of successors continues as before until returning eventually to the successor of $+e$ in $\mathfrak{L}$. This creates a new face in $\mathfrak{L}\#_{e,e'}\mathfrak{L}'$ which is formed by merging two previous faces in $\mathfrak{L}$ and $\mathfrak{L}'$ and contains $+e$ and $+e'$. We can argue similarly that a second new face of $\mathfrak{L}\#_{e,e'}\mathfrak{L}'$ is also created, containing $-e$ and $-e'$.

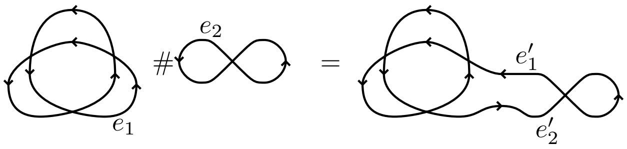


FIG. 5: The connected sum of two link shadows yields a new shadow where two pairs of faces have been merged.

Since the number of vertices and edges of $\mathfrak{L}\#_{e,e'}\mathfrak{L}'$ is the sum of the vertices and edges in $\mathfrak{L}$ and $\mathfrak{L}'$ and the number of faces is 2 less than the sum of the number of faces in $\mathfrak{L}$ and $\mathfrak{L}'$, $V-E+F=2$ for $\mathfrak{L}\#_{e,e'}\mathfrak{L}'$, so the connect sum pd-code is planar as desired. □

We now give a theorem corresponding to the prime decomposition of links [**?**]:

Proposition 13. *Every composite link shadow can be created by connect sum operations on a well-defined set of prime link shadows called the prime factors of the composite shadow.*

Proof. We proceed by induction on the number of pairs of edges shared by two faces. If no two faces share more than one edge, the number of pairs is zero and the diagram is already prime. This is the base case.

Observe that as in Figure 6, if some pair of faces shares some number of edges, the shared edges occur in precisely opposite order on both faces (if not, the diagram isn't planar). This means that our diagram is the connect sum of two subdiagrams created by cutting and splicing a pair of adjacent shared edges, as on the right side of that Figure.

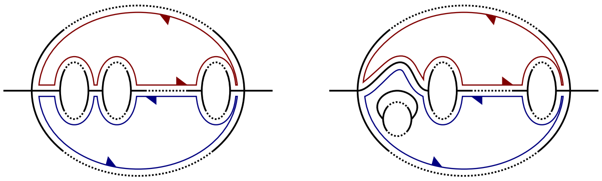


FIG. 6: Two faces which share more than one edge must create an opportunity for a cut and splice.

Since each subdiagram has fewer pairs of edges shared by two faces, by induction we may write them as connect sums of collections of prime diagrams. But we made arbitrary choices when deciding which pair of adjacent edges to cut-and-splice, and so must check that the collection of prime diagrams did not depend on these choices.

It's enough to show that the collection of subdiagrams produced by doing any two such cut-and-splices in one order is the same as the collection of subdiagrams produced by doing them in the other. But this is clear as the operations don't interfere with one another (Figure 7 illustrates the point.) □

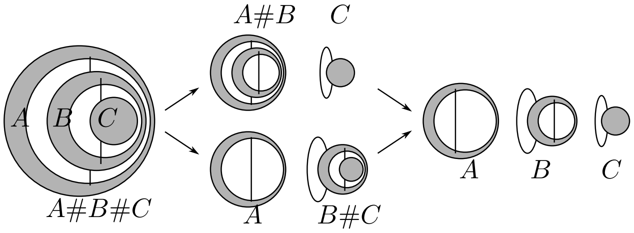


FIG. 7: Cutting and splicing at two locations can be done in either order and produce the same list of subdiagrams.

We now know that every n vertex composite shadow L is the connect sum of prime shadows $L_1, \ldots, L_k$ and that the numbers of vertices $V_1, \ldots, V_k$ form an integer partition of n. Therefore, we need to enumerate "all connect sums of all link shadows with numbers of vertices that partition n". This is not as trivial as it sounds. Writing the connect sum as $L_1\# \ldots \#L_k$ is dangerously misleading– though connect sum is associative (and even commutative) on isotopy classes of knots, the same is not true for links and for diagrams. Figure 8 gives an example of a link shadow L which is a connect sum of diagrams L_1, L_2 and L_3, which *can* be written $(L_1\#L_2)\#L_3$ but *can't* be written $L_1\#(L_2\#L_3)$.

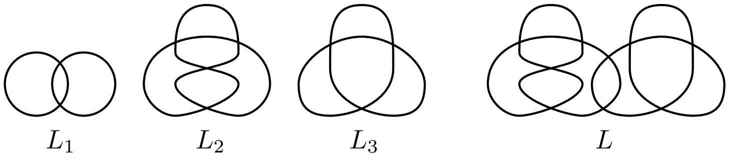


FIG. 8: L can be written as $L = (L_1\#L_2)\#L_3$ but not as $L_1\#(L_2\#L_3)$.

It is easy to see the following.

Lemma 14. *The set of prime summands $\{L_1, \ldots, L_k\}$ of a composite link shadow L can be renumbered so that $L = (((L_1\#L_2)\#L_3)\# \cdots L_k)$ and L_1 has the largest number of vertices among all the $L_1, \ldots, L_k$. The numbers of vertices $V_2, \ldots, V_k$ in the other prime summands are* not *guaranteed to be in sorted order.*

Proof. Since the overall shadow is connected, we can start connect-summing at any prime summand and build the rest of the shadow from there by undoing the cut-and-splice operations of Proposition 13. Therefore, we are free to choose L_1 to have a maximal number of vertices. □

Lemma 15. *The set $\{L_1, \ldots, L_k\}$ of the prime summands of L is a shadow-isomorphism invariant of L. In particular, the set of numbers of vertices $\{V_1, \ldots, V_k\}$ is a shadow isomorphism invariant of L.*

Proof. This follows directly from that the set $\{L_1, \ldots, L_k\}$ is well-defined. □

Now recall that a pdstor (Definition 9) is an ordered collection of canonical pd-codes which represent pd-isomorphism classes. We can define connect sum for pdstors by saying that $P_1 \# P_2$ is the pdstor constructed by adding all $\mathfrak{L}_1 \#_{e1,e2} \mathfrak{L}_2$ where e_i is an edge of some $\mathfrak{L}_\mathrm{i} \in P_i$. We now generate composite link shadows iteratively:

```
procedure BuildCompositeShadows($n$)          ▷ Build all link shadows with $\leq n$ vertices
    Use plantri to build pdstors $P_1, \ldots, P_n$ containing all oriented prime link shadows with 1 to $n$ vertices.
    for all $1 \leq k \leq n$ do
        Define an empty pdstor $P_k$.
        for all partially sorted partitions $k_1 k_2 \ldots k_\ell$ of $k$ with $k_1 \geq k_i$ do
            Build the pdstor $P_{k_1 k_2 \cdots k_\ell} = ((P_{k_1} \# P_{k_2}) \# P_{k_3}) \# \cdots P_{k_\ell}$.
            (If $((P_{k_1} \# P_{k_2}) \# P_{k_3}) \# \cdots P_{k_{\ell-1}}$ was already computed, we can reuse it.)
        end for
        for all fully sorted partitions $k_1 k_2 \ldots k_\ell$ of $k$ with $k_1 \geq k_2 \geq \cdots \geq k_\ell$ do
            Define an empty pdstor $P$.
            for all partially sorted partitions $k'_1 k'_2 \cdots k'_\ell$ with the same set of $k_i$ do
                Add all elements of $P_{k'_1 k'_2 \cdots k'_\ell}$ to $P$.
                (Some of these will be isomorphic to one another.)
            end for
            Add all elements of $P$ to $P_k$.
            (By Lemma 15, these aren't isomorphic to anything previously computed,
            so we don't need to check for isomorphism with existing elements of $P_k$.)
        end for
    end for
end procedure
```

3.2. Expansions of embedded planar simple graphs

Our next strategy will be much more complicated (and somewhat slower to run), but it serves as crucial check on the previous computation. The basic idea is to define a smaller class of graphs so that the graphs we are interested in can be obtained from the base class of graphs by various expansion moves. Lehel [**?**] gave a strategy for generating all 4-regular graphs in this way from the octahedral graph. Instead of using Lehel's strategy directly, we build on the method of Brinkmann and McKay [**? ?**] for enumerating isomorph-free embedded planar graphs; we extend their work here to generate the class of graphs that we're interested in.

We observed above that the link shadows are embedded isomorphism classes of 4-regular embedded planar multigraphs. We now define four expansion moves of embedded planar graphs with vertex degree ≤ 4 which generate embedded planar multigraphs of vertex degree ≤ 4 with the same number of vertices, but additional edges:

Definition 16. The four expansion operations that we will use are the following:

- E_1 loop insertion adds a loop edge to a vertex of degree 1 or 2, as below. Loop insertion can be performed on each side of a vertex of degree 2.

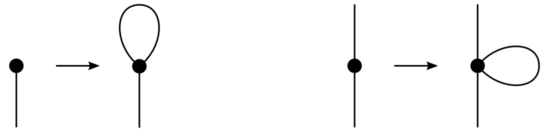


FIG. 9: Adding a loop edge to a vertex of degree 1 or 2.

- E_2 reversing edge doubling duplicates an existing edge joining vertices of degree < 4 so as to create a new bigon face. Note that the (counterclockwise) order of the two vertices is reversed on the two vertices.

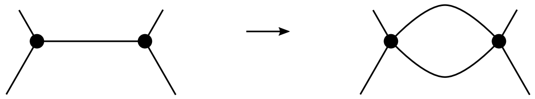


FIG. 10: Doubling an edge joining two vertices of degree < 4.

- E_3 preserving doubling also duplicates an existing edge joining vertices of degree < 4, but keeps the counterclockwise order of the edges the same on each at each of the two vertices. This sort of doubling is only available if the original edge is a cut edge of the graph.

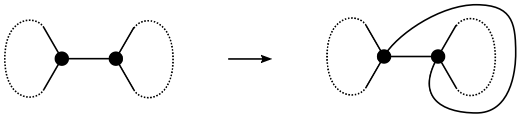


FIG. 11: Doubling a cut edge joining two vertices of degree < 4 can be done another way.

- E_4 pair insertion adds a pair of edges simultaneously, joining two vertices of degree 2 which are both on two faces of the embedding, as below.

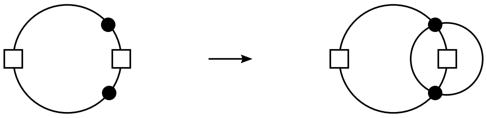


FIG. 12: Adding two new edges to join vertices of degree 2 on the same face.

We can now show

Proposition 17. *Every link shadow L can be obtained from a connected, embedded planar simple graph of vertex degree ≤ 4 G_0 by a series of E_1, E_2, E_3, and E_4 expansions.*

Equivalently, any link shadow L can be reduced to a connected embedded planar simple graph L_0 of vertex degree ≤ 4 by a series of E_1, E_2, E_3, and E_4 reductions. The embedded isomorphism type of L_0 is determined uniquely by the (unoriented) shadow isomorphism type of L (the order in which the reductions are performed doesn't matter).

An illustration of the process we describe is shown in Figure 13.

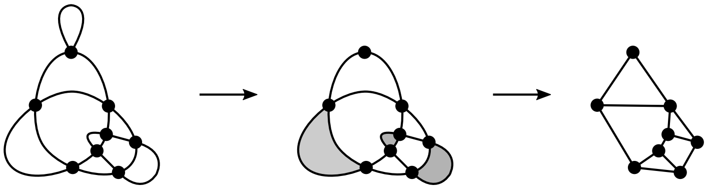


FIG. 13: Any link shadow can be reduced to a connected, embedded planar simple graph of vertex degree ≤ 4 by a series of reductions, according to Proposition 17.

The proof appears in Appendix **??**. We can now build the link shadows by applying expansions to the graphs produced by *plantri*. This is not trivial. First, not all sequences of expansion moves lead to link shadows. Second, different sequences of expansion moves may produce isomorphic link shadows. Some examples are shown in Figure 14.

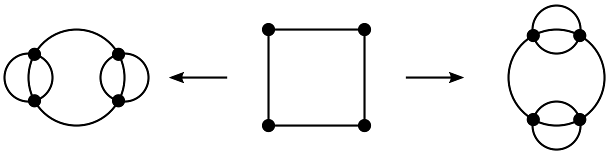


FIG. 14: Two different sequences of E_2 moves lead to isomorphic expansions of this simple graph. This means that the output of the expansion process will contain duplicate link shadows which must be detected and eliminated.

We can think of this as the problem of searching for collections of expansion moves which obey various equations, such as the fact that the total vertex degree must be 4 in any complete solution (there are less obvious equations as well– the complete system is specified in Appendix **??**).

We find all of the solutions using a standard branch-and-bound algorithm and add the results to a pdstor to eliminate isomorphic pd-codes. We can save a lot of time in this process by noting that Proposition 17 tells us that we need only check expansions of the same graph (and their reorientations) against each other for potential pd-isomorphisms.

4. COMPUTING AND VERIFYING THE RESULTS

We implemented the above algorithms in C and used them to generate a list of link shadows up to 10 crossings, each unique up to pd-isomorphism. The longest run took several days on a desktop computer. We then extracted the knot shadows, leaving the link shadows for future work. We assigned orientations and over-under information at the crossings to each shadow to make a list of diagrams containing various duplicate diagrams due to symmetry. We eliminated diagram-isomorphic duplicates from this list to arrive at a database of diagrams. Many knots were identified uniquely by their HOMFLY polynomial, which we computed using the Millett/Ewing HOMFLY-PT code [**?**]. About 6.5 million diagrams whose HOMFLY corresponded to more than one knot were classified by using Bar-Natan and Morrison's KnotTheory *Mathematica* package to compute the Kauffman polynomial and signature. Since KnotTheory gives incorrect answers for diagrams with one-component "loop" faces, we had to eliminate these faces before computation.

This part of the computation was large enough to require organization; several thousand hours of computer time were required to expand the 10 crossing shadows into diagrams and compute their HOMFLY polynomials. We processed the diagrams across 400 simultaneous Amazon EC2 cores in Oregon and Virginia (additional details available in [**?**]). The counts of shadows and diagrams are given in Table I.

Cr	Prime Shadows	Link Shadows	Knot Shadows	Knot Diagrams
3	1	7	6	36*
4	2	30	19	276*
5	3	124	76	2936*
6	9	733	376	35 872*
7	18	4586	2194	484 088*
8	62	33 373	14 614	6 967 942*
9	198	259 434*	106 421	105 555 336*
10	803	2 152 298*	823 832	1 664 142 836*

TABLE I: The number of link (including knots) and knot shadows and diagrams through 10 crossings. The unstarred numbers in the column of prime shadows are sequence A113201 in the OEIS. The unstarred numbers in the table of link shadows matche Kápolnai et al. [**?**]. The numbers in the column of knot shadows are sequence A008989 in the OEIS. They match the "UU, $g = 0$" row on page 43 of the the recent preprint of Coquereaux et al. [**?**], including the value 823 832 for Cr $=$ 10 which those authors give as "should be confirmed". The starred numbers are new.

The most important question, of course, is how the computation was checked. Our implementation was careful and involved quite a bit of internal self-checking as well as testing against *valgrind* for memory problems, and writing a suite of unit tests for the codebase. However, good programming practices can only provide a limited measure of confidence in the results, so we continued to test our work. The first and most important test was to verify that the lists of knot and link shadows obtained by connect summing and by expansions were identical.

We were also able to check against some existing enumerations. Kápolnai et al. classified spherical "multiquadrangulations" [**?**], which are the duals of our diagrams as explained in Section 3.1. Their table 2 of the count of multiquadrangulations matches our count of diagrams exactly through 8 crossings (note that the quadrangulation has two more vertices than the dual knot diagram has crossings, so their data is shifted by two). It's worth observing that these authors also use *plantri*, so their results are not completely independent from ours. Still, it provides some comfort to see that their implementation on top of *plantri* produces the same results as ours. For knot shadows in particular, Arnol'd has given counts of the number of immersions of the unoriented circle into the unoriented sphere with n crossings for n from 0 to 5 [**?** , page 79]. This is sequence A008989 in the Online Encyclopedia of Integer Sequences, with an extension to $n = 7$ credited to Guy H. Valette. We could not find a published reference for Vallette's extension of the table, but our data for knots does match A008989 including the extension. Coquereaux et al. [**?**] have recently extended Valette's count using other means (their count is independent of *plantri*) and our numbers also match those in that paper.

Looking at Table I, one is struck by how close the number of distinct knot diagrams is to the maximum number $2^{\mathrm{Cr}+1} \times (\#\text{ knot shadows})$. To take 8 crossing diagrams as an example, we would expect at most $7\,482\,368 = 14\,614 \times 2^9$, and we have 6 967 942– about 93% of the maximum possible number. By the time we reach 10 crossing diagrams, the corresponding fraction is roughly 98.6%. We have fewer distinct diagrams only because some of the underlying knot diagrams have symmetries. For instance, the trefoil diagram 3_1 has a 3-fold rotational symmetry, so the crossing sign assignments $+--$, $-+-$ and $--+$ are all the same. However, our computations reveal that such symmetries quickly become very rare as the number of crossings increases. Table II shows the mean number of automorphisms of a knot shadow. One of us (Chapman) has proved that the number of these automorphisms decreases exponentially to 1 as the number of crossings becomes large, see [**?**].

Cr	3	4	5	6	7	8	9	10
Mean Automorphisms	5	$\frac{64}{19}$	$\frac{44}{19}$	$\frac{159}{94}$	$\frac{1447}{1097}$	$\frac{8426}{7307}$	$\frac{113460}{106421}$	$\frac{846979}{823832}$
— (decimal)	5.00	3.37	2.32	1.69	1.32	1.15	1.07	1.03

TABLE II: The mean number of automorphisms decreases rapidly as the number of crossings in the diagram increases. This means that the number of knot diagrams (with crossing signs and orientations) rapidly approaches the maximum allowed by the number of knot shadows.

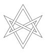
4.1. Knot Types and Inferred Counts

Counting knot types required us to be careful about knot symmetries. To review, mirroring crossings and reversing orientation yield a $\mathbb{Z}_2 \times \mathbb{Z}_2$ action on knot types. If a knot is isotopic to its image under a subgroup of this group, it is said to have a symmetry; there are five symmetry types corresponding to the five subgroups of this group: “none”, “mirror”, “reversible”, “amphichiral” (the diagonal subgroup), and “full”. We refer to a collection of knot types related by the group as a “base knot type”. For instance, the base knot type 3_1 consists of the two knot types 3_1 and 3_1^m (the mirror image of 3_1).

We were able to use the HOMFLY-PT and Kauffmann polynomials and the knot signature to classify almost all of the knots whose base type had symmetry “full” or “reversible”, since these invariants distinguish all knots with 10 crossings or less from their mirror images except the 10_{71} knot. There are 36 base types with one of the other three symmetries. These were more difficult to classify as classical invariants don’t distinguish knots from their reversals. We relied on symmetry to infer the distribution of counts:

Lemma 18. *If a base knot type K has symmetry type “amphichiral” ($K = K^{mr}$, $K^m = K^r$, but $K \neq K^r$) or “mirror” ($K = K^m$, $K^r = K^{mr}$, but $K \neq K^r$), then the number of diagrams of the* *two knot types are equal. If K has symmetry type "none" ($K \neq K^m \neq K^r \neq K^{mr}$) the number of diagrams of each of these four knot types are equal.*

Proof. In each case, there is a group action on diagrams (reverse the orientation of a diagram or take the 4-element group generated by reversing crossing signs and orientation) which converts diagrams of one knot of these knot types to a diagrams of the other types. This action extends to diagram-isomorphism classes of diagrams because the action commutes with any given diagram-isomorphism.

It is not always the case that this action is free on diagram-isomorphism classes of diagrams of any knot type. For instance, taking the mirror image of this diagram of the 7_4 knot  is equivalent to rotating it by 180 degrees (a diagram-isomorphism). This implies that 7_4 is a reversible knot, as diagram-isomorphisms are knot isotopies. But by hypothesis, our K is *not* isotopic to its images under the group action, and therefore it cannot by diagram-isomorphic either.

Since the action of the group on this set of diagram-isomorphism classes is free and exchanges the various knot types, the number of diagram-isomorphism classes of diagrams in each knot type must be the same. □

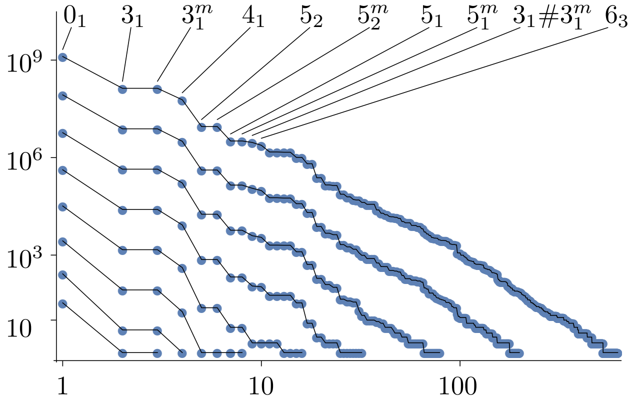
For example, using this lemma we split our original count of 5672 10-crossing diagrams of base type 8_{17} into 2836 diagrams of type 8_{17} and 2836 diagrams of type 8_{17}^r, even though we had no way of knowing which of our diagrams was assigned to each knot type.

The relative frequencies of all the knot types appear in Figure 15 in a log-log plot. The figure also shows the 10 most frequent knot types, from the unknot (the most frequent knot type in all our data) and the trefoils 3_1 and 3_1^m through the square knot $3_1\#3_1^m$ and the 6_3 knot, which is the 10th most common knot in all our data. Though we know [**?**] both that unknotted diagrams eventually become exponentially rare, and that the rank-order of the knot types must eventually change, we do not see this effect in our data; the list of 10 most common knots is the same for crossing numbers 6-10. Table III gives more detailed frequency data for the 40 most common knot types (among 10 crossing diagrams). This data does show some reordering of knot types as crossing number increases.

Crossing Number	3	4	5	6	7	8	9	10
Number of Diagrams	36	276	2936	35 872	484 088	6 967 942	105 555 336	1 664 142 836
0_1	34	265	2744	32 456	422 332	5 852 832	85 253 534	1 291 291 155
3_1	1	5	85	1466	25 432	440 570	7 696 083	135 702 456
3_1^m	1	5	85	1466	25 432	440 570	7 696 083	135 702 456
4_1	–	1	18	412	8450	165 791	3 175 612	60 146 706
5_2	–	–	1	24	730	18 075	415 290	9 025 926
5_2^m	–	–	1	24	730	18 075	415 290	9 025 926
5_1	–	–	1	6	213	5819	142 614	3 252 078
5_1^m	–	–	1	6	213	5819	142 614	3 252 078
$3_1\#3_1^m$	–	–	–	2	112	3953	113 684	2 923 783
6_3	–	–	–	2	106	3515	96 666	2 389 180
6_2	–	–	–	1	58	2027	58 354	1 493 624
6_2^m	–	–	–	1	58	2027	58 354	1 493 624
$3_1\#3_1$	–	–	–	2	58	2006	56 893	1 461 498
$3_1^m\#3_1^m$	–	–	–	2	58	2006	56 893	1 461 498
6_1^m	–	–	–	1	34	1267	38 199	1 015 996
6_1	–	–	–	1	34	1267	38 199	1 015 996
$3_1\#4_1$	–	–	–	–	8	516	20 458	648 362
$3_1^m\#4_1$	–	–	–	–	8	516	20 458	648 362
7_6^m	–	–	–	–	3	193	7608	240 121
7_6	–	–	–	–	3	193	7608	240 121
7_7	–	–	–	–	2	124	4709	144 455
7_7^m	–	–	–	–	2	124	4709	144 455
7_5^m	–	–	–	–	2	102	4244	138 467
7_5	–	–	–	–	2	102	4244	138 467
7_2^m	–	–	–	–	1	44	2103	74 739
7_2	–	–	–	–	1	44	2103	74 739
7_3	–	–	–	–	1	39	1793	62 059
7_3^m	–	–	–	–	1	39	1793	62 059
$4_1\#4_1$	–	–	–	–	–	20	1176	51 526
7_4^m	–	–	–	–	1	36	1516	49 731
7_4	–	–	–	–	1	36	1516	49 731
8_{20}	–	–	–	–	–	14	985	41 843
8_{20}^m	–	–	–	–	–	14	985	41 843
$3_1^m\#5_2^m$	–	–	–	–	–	10	784	36 548
$3_1\#5_2$	–	–	–	–	–	10	784	36 548
$3_1^m\#5_2$	–	–	–	–	–	10	784	36 544
$3_1\#5_2^m$	–	–	–	–	–	10	784	36 544
8_{21}	–	–	–	–	–	9	574	24 611
8_{21}^m	–	–	–	–	–	9	574	24 611
8_{14}	–	–	–	–	–	6	442	19 412
8_{14}^m	–	–	–	–	–	6	442	19 412
7_1^m	–	–	–	–	1	8	444	17 441

TABLE III: This table shows the number of diagrams of each knot type among all knot diagrams with between 3 and 10 crossings. The knot types shown are the most common 40 knot types among 10 crossing diagrams, and they appear in the order of their frequency among 10 crossing diagrams. This is not the same rank order for all crossing numbers– one can observe that $4_1\#4_1$ is less common than 7_4 among 8 crossing diagrams (20 diagrams versus 36 diagrams) but more common than 7_4 among 10 crossing diagrams (51 526 diagrams versus 49 731 diagrams).

More interestingly, the log-log plot of ranked knot frequencies for the 622 different possible knot types for 10 crossing diagrams is roughly linear over 9 orders of magnitude in frequency (there are about 1.6×10^9 diagrams, with the most frequent knot type (the unknot) occurring 1.2×10^9 times and the least frequent knot types (a 98-way tie among various 10-crossing knots) appearing exactly once. This is consistent with a power law: the probability of the n-th most probable knot divided by the probability of the unknot should be approximated by n^{-r}. This phenomenon was first observed in numerical simulations of random knots modeled as collapsed self-avoiding lattice polygons by Baiesi, Orlandini, and Stella [**?**], and has also been observed in samples of equilateral polygons by Cantarella and Shonkwiler [**?**]. This type of relationship was first observed by Zipf in linguistic data [**?**]. In our data, there is no sampling cutoff since we can classify every one of the diagrams, and the frequencies remain approximately linear throughout the entire dataset– we do not see the downward trend for very infrequently observed knots found in [**?**].



Cr	Unknots	(decimal)
3	$\frac{17}{18}$	0.94
4	$\frac{265}{276}$	0.96
5	$\frac{343}{367}$	0.93
6	$\frac{4057}{4484}$	0.90
7	$\frac{105\,583}{121\,022}$	0.87
8	$\frac{2\,926\,416}{3\,483\,971}$	0.84
9	$\frac{42\,626\,767}{52\,777\,668}$	0.81
10	$\frac{1\,291\,291\,155}{1\,664\,142\,836}$	0.78

FIG. 15: A log-log plot of knot frequencies in rank order for crossing numbers 3 through 10, with the 10 most common knot types identified. These knot types have the same rank ordering in all the crossing numbers computed. The data show some evidence of power-law behavior. At right, we see the unknot fraction in table form.