Supplementary Material for the Paper "Inference and Sampling of K_{33}-free Ising Models"

1 Technical Proofs

Lemma 1 proof. Let $E' \in \mathrm{PM}(G^*)$. Call $e \in E$ *saturated*, if it intersects an edge from $E' \cap E_I^*$. Each Fisher city is incident to an odd number of edges in $E' \cap E_I^*$. Thus, each face of G has an even number of unsaturated edges. This property is preserved, when two faces/cycles are merged into one by evaluating respective symmetric difference. Therefore, one gets that any cycle in G has an even number of unsaturated edges.

For each i define $x_i := -1^{r_i}$, where r_i is the number of unsaturated edges on the path connecting v_1 and v_i. The definition is consistent due to aforementioned cycle property. Now for each $e = \{v, w\} \in E$, $x_v = x_w$ if and only if e is saturated. To conclude, we constructed X such that $E' = M(X)$. Such X is unique, because parity of unsaturated edges on a path between v_1 and v_i uniquely determines relationship between x_1 and x_i, and x_1 is always $+1$. □

Lemma 2 proof. Let $X' = (x'_1, ..., x'_N) \in \mathcal{C}_+$, $M(X') = E'$. The statement is justified by the following chain of transitions:

$$\begin{aligned}
\mathbb{P}(M(S) = E') &= \mathbb{P}(S = X') + \mathbb{P}(S = -X') \\
&= \frac{2}{Z} \exp\left(\sum_{e=\{v,w\} \in E} J_e x'_v x'_w \right) \\
&= \frac{2}{Z} \exp\left(\sum_{e^* \in E' \cap E_I^*} 2J_{g(e^*)} - \sum_{e \in E} J_e \right) \\
&= \frac{2}{Z} \exp\left(- \sum_{e \in E} J_e \right) \prod_{e^* \in E' \cap E_I^*} c_{e^*} \\
&= \frac{2}{Z} \exp\left(- \sum_{e \in E} J_e \right) \prod_{e^* \in E'} c_{e^*} \\
&= \frac{1}{Z^*} \prod_{e^* \in E'} c_{e^*}
\end{aligned}$$

Lemma 3 proof. The Algorithm 1 reduces sampling on G to a series of samplings on $G_1, ..., G_h$.

Given the algorithm and inference formula in Lemma 3, the statement is obvious for $h = 1$. Let $h = 2$. Let v be an articulation point shared by G_1 and G_2. Denote $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$. Without loss of generality assume that v has index 1 in V, V_1 and V_2. Let $\mathcal{C}_+^i = \{+1\} \times \{-1, +1\}^{|V_i|}$. Then one derives:

$$\begin{aligned}
Z &= 2 \sum_{X \in \mathcal{C}_+} \exp\left(\sum_{e=\{v,w\} \in E} J_e x_v x_w \right) \\
&= 2 \sum_{X \in \mathcal{C}_+} \left[\exp\left(\sum_{e=\{v,w\} \in E_1} J_e x_v x_w \right) \cdot \exp\left(\sum_{e=\{v,w\} \in E_2} J_e x_v x_w \right) \right] \\
&= 2 \sum_{X_1 \in \mathcal{C}_+^1} \exp\left(\sum_{e=\{v,w\} \in E_1} J_e x_v x_w \right) \cdot \sum_{X_2 \in \mathcal{C}_+^2} \exp\left(\sum_{e=\{v,w\} \in E_2} J_e x_v x_w \right)
\end{aligned}$$

Algorithm 1 Sampling from $\mathbb{P}(S = X)$

Input: A tree of $G_1, ..., G_h$.
Draw $X_1, ..., X_h$ from ZFI models on $G_1, ..., G_h$.
ProcessComponent(1, -1).
Combine $X_1, ..., X_h$ into X.
Output: X.

Procedure ProcessComponent
Input: index i, parent index p.
v = articulation point of G_i and G_p.
if unequal spins of X_i and X_p at v **then**
    $X_i := -X_i$
for j **in** i's neighbors **do**
    if $j \neq p$ **then**
        ProcessComponent(j, i).
end Procedure

$$= \frac{1}{2} Z_1 Z_2$$

where Z_i is the PF of the ZFI model induced by G_i. As far as sampling is concerned, denote by $\mathbb{P}_i(S_i = X_i)$ a probability distribution induced by the i-th ZFI model. Then, since $\mathbb{P}_2(s_1 = x_1) = \frac{1}{2}$:

$$\begin{aligned}
\mathbb{P}(S = X) &= \frac{1}{Z} \sum_{X \in \mathcal{C}_+} \exp\left(\sum_{e=\{v,w\} \in E} J_e x_v x_w \right) \\
&= 2 \frac{1}{Z_1} \exp\left(\sum_{e=\{v,w\} \in E_1} J_e x_v x_w \right) \cdot \frac{1}{Z_2} \exp\left(\sum_{e=\{v,w\} \in E_2} J_e x_v x_w \right) \\
&= 2 \mathbb{P}_1(S_1 = X_1) \mathbb{P}_2(S_2 = X_2) \\
&= \mathbb{P}_1(S_1 = X_1) \frac{\mathbb{P}_2(S_2 = X_2)}{\mathbb{P}_2(s_1 = x_1)} \\
&= \mathbb{P}_1(S_1 = X_1) \mathbb{P}_2(S_2 = X_2 | s_1 = x_1)
\end{aligned}$$

Assume that a method for sampling S_i from $\mathbb{P}_i$ is available. Then, draw X_1 by sampling S_1 from $\mathbb{P}_1$. To sample S_2 conditional on $s_1 = x_1$ from $\mathbb{P}_2$, draw $X_2' = (x_1', ...)$ from $\mathbb{P}_2(S_2 = X_2')$. If $x_1' = x_1$, then $X_2 = X_2'$, otherwise $X_2 = -X_2'$. This is consistent with Algorithm 1.

For graphs of $h > 2$ the statement of lemma follows naturally by induction.

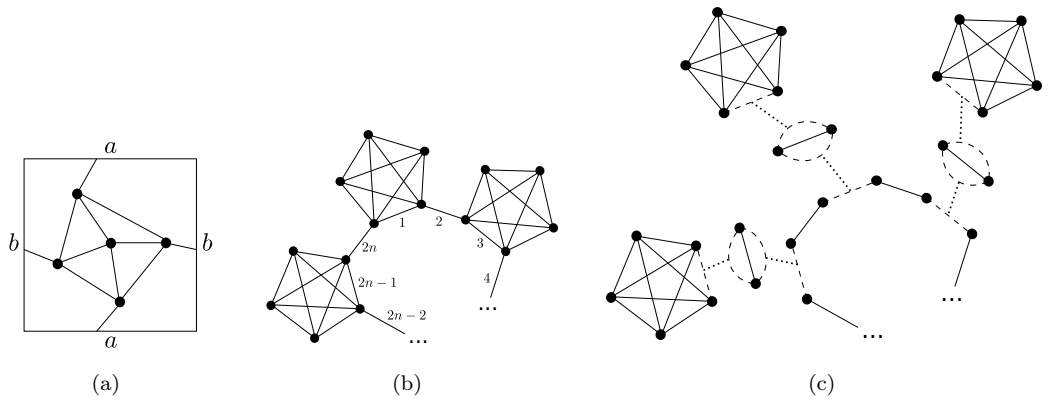
Theorem 2 proof. Since G is normal and minor-free, it holds that $|E| = O(N)$ [13]. Find all biconnected components and for each construct a triconnected component tree in $O(N + |E|) = O(N)$.

As described above, the time (number of steps) of inference or sampling is a sum of inference or sampling times of each triconnected component of G. Let the set of all G's triconnected components (that is, a union over all biconnected components) to consist of k_1 planar triconnected components of size $N_1, ..., N_{k_1}$ with $M_1^p, ..., M_{k_1}^p$ edges respectively, k_2 multiple bonds of $M_1^b, ..., M_{k_2}^b$ edges and k_3 K_5 graphs. Then the complexity of inference or sampling is $O(\sum_{i=1}^{k_1} N_i^{\frac{3}{2}} + \sum_{i=1}^{k_2} M_i^b + k_3)$.

The edges of G are partitioned among biconnected components. Inside each biconnected component apply second part of Lemma 4 to obtain that $\sum_{i=1}^{k_1} M_i^p + \sum_{i=1}^{k_2} M_i^b + 10k_3 = O(|E|) = O(N)$. This gives that $\sum_{i=1}^{k_2} M_i^b + k_3 = O(N)$ and $\sum_{i=1}^{k_1} M_i^p = O(N)$. Since triconnected components are connected graphs, we get that $N_i = O(M_i^p)$ for all $1 \leq i \leq k_1$ and hence $\sum_{i=1}^{k_1} N_i = O(N)$. From convexity of $f(x) = x^{\frac{3}{2}}$ it follows that $\sum_{i=1}^{k_1} N_i^{\frac{3}{2}} = O(N^{\frac{3}{2}})$ and finally that $O(\sum_{i=1}^{k_1} N_i^{\frac{3}{2}} + \sum_{i=1}^{k_2} M_i^b + k_3) = O(N^{\frac{3}{2}})$. □

Lemma 7 proof. A simple example illustrates that genus of a biconnected K_{33}-free graph can grow linearly with its size. First, notice that K_5 is a nonplanar graph, but it can be embedded in toroid (Fig. 1(a)), therefore genus of the graph is unity. Consider a cycle of length $2n$, enumerate edges in the order of cycle traversal from 1 to $2n$. Attach K_5 graph to each odd edge of the cycle

(see Fig. 1(b)). The resulting graph G is of size $5n$, it is biconnected and K_{33}-free (see Figure 1(c)). Remove an arbitrary even edge from the cycle. It results in a graph whose biconnected components are n K_5 graphs and n edges, so its genus is n. Since edge removal can only decrease genus, we conclude that G's genus is at least n.



(a) (b) (c)

Figure 1: (a) K_5's embedding on a toroid - glue sides with the same label together. (b) G - a "necklace" of n K_5 graphs. (c) G's triconnected components. Dashed lines are virtual edges and dotted lines identify identical virtual edges. Triconnected components consist of a cycle, triple bonds and K_5 graphs. Hence, by Lemma 6 G is K_{33}-free.

2 Counting PMs of Planar $\hat{G}$ in $O(\hat{N}^{\frac{3}{2}})$ time

This section addresses inference part of Theorem 1.

2.1 Pfaffian Orientation

Let $\hat{G}$ be an oriented graph. Its cycle of even length (built on an even number of vertices) is said to be *odd-oriented*, if, when all edges along the cycle are traversed in any direction, an odd number of edges are directed along the traversal. An orientation of $\hat{G}$ is called *Pfaffian*, if all cycles C, such that $\mathrm{PM}(\hat{G}(\hat{V} - C)) \neq \varnothing$, are odd-oriented.

We will need $\hat{G}$ to contain a Pfaffian orientation, moreover the construction is easy.

Theorem 4. *Pfaffian orientation of $\hat{G}$ can be constructed in $O(\hat{N})$.*

Proof. This theorem is proven constructively, see e.g. [15, 14], or [12], where the latter construction is based on specifics of the expanded dual graph. □

Construct a skew-symmetric sparse matrix $K \in \mathbb{R}^{\hat{N}\times\hat{N}}$ ($\rightarrow$ denotes orientation of edges):

$$K_{ij} = \begin{cases} c_e & \text{if } \{v_i, v_j\} \in \hat{E}, v_i \rightarrow v_j \\ -c_e & \text{if } \{v_i, v_j\} \in \hat{E}, v_j \rightarrow v_i \\ 0 & \text{if } \{v_i, v_j\} \notin \hat{E} \end{cases} \tag{1}$$

The next result allows to compute PF $\hat{Z}$ of PM model on $\hat{G}$ in a polynomial time.

Theorem 5. $\det K > 0$, $\hat{Z} = \sqrt{\det K}$.

Proof. See, e.g., [15] or [8]. □

2.2 Computing $\det K$

LU-decomposition of a matrix $A = LU$, found via Gaussian elimination, where L is a lower-triangular matrix with unit diagonals and U is an upper-triangular matrix, would be a standard way of computing $\det A$, which is then equal to a product of the diagonal elements of U. However,

this standard way of constructing the LU decomposition applies only if all A's leading principal submatrices are nonsingular (See e.g. [6], Section 3.5, for detailed discussions). And already the first, 1×1, leading principal submatrix of K is zero/singular.

Luckily, this difficulty can be resolved through the following construction. Take $\hat{G}$'s arbitrary perfect matching $E' \in \mathrm{PM}(\hat{G})$. In the case of a general planar graph E' can be found via e.g. Blum's algorithm [1] in $O(\sqrt{\hat{N}}|\hat{E}|) = O(\hat{N}^{\frac{3}{2}})$ time, while for graphs G^*, G_v^* and $\overline{G}_v^*$ appearing in this paper E' can be found in $O(N)$ from a spin configuration using M mapping (e.g. $E' = E_I^* = M(\{+1, ..., +1\}) \in \mathrm{PM}(G^*)$). Modify ordering of vertices, $\hat{V} = \{v_1, v_2, ..., v_{\hat{N}}\}$, so that $E' = \{\{v_1, v_2\}, ..., \{v_{\hat{N}-1}, v_{\hat{N}}\}\}$. Build K according to the definition (1). Obtain $\overline{K}$ from K by swapping column 1 with column 2, 3 with 4 and so on. This results in $\det K = |\det \overline{K}|$, where the new $\overline{K}$ is properly conditioned.

Lemma 8. *$\overline{K}$'s leading principal submatrices are nonsingular.*

Proof. The proof, presented in [15] for the case of unit weights c_e, generalizes to arbitrary positive c_e. □

Notice, that in the general case (of a matrix represented in terms of a general graph) complexity of the LU-decomposition is cubic in the size of the matrix. Fortunately, *nested dissection* technique, discussed in the following subsection, allows to reduce complexity of computing $\hat{Z}$ to $O(\hat{N}^{\frac{3}{2}})$.

2.3 Nested Dissection

The partition P_1, P_2, P_3 of set $\hat{V}$ is a *separation* of $\hat{G}$, if for any $v \in P_1, w \in P_2$ it holds that $\{v, w\} \notin \hat{E}$. We refer to P_1, P_2 as the *parts*, and to P_3 as the *separator*.

Lipton and Tarjan (LT) [10] found an $O(\hat{N})$ algorithm, which finds a separation P_1, P_2, P_3 such that $\max(|P_1|, |P_2|) \leq \frac{2}{3}\hat{N}$ and $|P_3| \leq 2^{\frac{3}{2}}\sqrt{\hat{N}}$. The LT algorithm can be used to construct the so called *nested dissection ordering* of $\hat{V}$. The ordering is built recursively, by first placing vertices of P_1, then P_2 and P_3, and finally permuting indices of P_1 and P_2 recursively according to the ordering of $\hat{G}(P_1)$ and $\hat{G}(P_2)$ (See [9] for accurate description of details, definitions and analysis of the nested dissection ordering). As shown in [9] the complexity of finding the nested dissection ordering is $O(\hat{N} \log \hat{N})$.

Let A be a $\hat{N} \times \hat{N}$ matrix with a *sparsity pattern* of $\hat{G}$. That is, A_{ij} can be nonzero only if $i = j$ or $\{v_i, v_j\} \in \hat{E}$.

Theorem 6. *[9] If $\hat{V}$ is ordered according to the nested dissection and A's leading principal submatrices are nonsingular, computing the LU-decomposition of A becomes a problem of the $O(N^{\frac{3}{2}})$ complexity.*

Notice, however, that we cannot directly apply the Theorem to $\overline{K}$, because the sparsity pattern of K is asymmetric and does not correspond, in general, to any graph.

Let $G^{**} = (V^{**}, E^{**})$ be a planar graph, obtained from $\hat{G}$, by contracting each edge in E', $|V^{**}| = |E'| = \frac{1}{2}\hat{N}$. Find and fix a nested dissection ordering over V^{**} (it takes $O(\hat{N} \log \hat{N})$ steps) and let the $\{v_1, v_2\}, \ldots, \{v_{\hat{N}-1}, v_{\hat{N}}\}$ enumeration of E' correspond to this ordering. Split K into 2×2 cells and consider the sparsity pattern of the nonzero cells. One observes that the resulting sparsity pattern coincides with the sparsity patterns of $\overline{K}$ and G^{**}. Since LU-decomposition can be stated in the 2×2 block elimination form, its complexity is reduced down to $O(\hat{N}^{\frac{3}{2}})$.

This concludes construction of an efficient inference (counting) algorithm for planar PM model.

3 Sampling PMs of Planar $\hat{G}$ in $O(\hat{N}^{\frac{3}{2}})$ time (Wilson's Algorithm)

This section addresses sampling part of Theorem 1. In this section we assume that degrees of $\hat{G}$'s vertices are upper-bounded by 3. This is true for G^*, G_v^* and $\overline{G}_v^*$ - the only PM models appearing in the paper. Any other constant substituting 3 wouldn't affect the analysis of complexity. Moreover, Wilson wilson shows that any PM model on a planar graph can be reduced to bounded-degree planar model without affecting $O(\hat{N}^{\frac{3}{2}})$ complexity.

3.1 Structure of the Algorithm

Denote a sampled PM as M, $\mathbb{P}(M) = \hat{Z}^{-1} \prod_{e \in M} c_e$. Wilson's algorithm first applies LT algorithm of [10] to find a separation P_1, P_2, P_3 of $\hat{G}$ ($\max(|P_1|, |P_2|) \leq \frac{2}{3}\hat{N}$, $|P_3| \leq 2^{\frac{3}{2}}\sqrt{\hat{N}}$). Then it iterates over $v \in P_3$ and for each v it draws an edge of M, saturating v. Then it appears that, given this intermediate result, drawing remaining edges of M may be split into two independent drawings over $\hat{G}(P_1)$ and $\hat{G}(P_2)$, respectively, and then the process is repeated recursively.

It takes $O(\hat{N}^{\frac{3}{2}})$ steps to sample edges attached to P_3 at the first step of the recursion, therefore the overall complexity of the Wilson's algorithm is also $O(\hat{N}^{\frac{3}{2}})$.

Subsection 3.2 introduces probabilities required to draw the aforementioned PM samples. Subsections 3.3 and 3.4 describe how to sample edges attached to the separator, while Subsection 3.5 focuses on describing the recursion.

3.2 Drawing Perfect Matchings

For some $Q \in \hat{E}$ consider the probability of getting Q as a subset of M:

$$\begin{aligned}\mathbb{P}(Q \subseteq M) &= \frac{1}{\hat{Z}} \sum_{\substack{M' \in \mathrm{PM}(\hat{G}) \\ Q \subseteq M'}} \left(\prod_{e \in M'} c_e \right) \\ &= \frac{1}{\hat{Z}} \left(\prod_{e \in Q} c_e \right) \cdot \sum_{M' \in \mathrm{PM}(\hat{G})} \left(\prod_{e \in M' \setminus Q} c_e \right) \end{aligned} \tag{2}$$

Let $\hat{V}_Q = \cup_{e \in Q} e$ and $\hat{G}_{\setminus Q} = \hat{G}(\hat{V} \setminus \hat{V}_Q)$. Then the set $\{M' \setminus Q \,|\, M' \in \mathrm{PM}(\hat{G})\}$ coincides with $\mathrm{PM}(\hat{G}_{\setminus Q})$. This yields the following expression

$$\mathbb{P}(Q \subseteq M) = \frac{\hat{Z}_{\setminus Q}}{\hat{Z}} \left(\prod_{e \in Q} c_e \right)$$

where

$$\hat{Z}_{\setminus Q} = \sum_{M'' \in \mathrm{PM}(\hat{G}_{\setminus Q})} \left(\prod_{e \in M''} c_e \right)$$

is a PF of the PM model on $\hat{G}_{\setminus Q}$ induced by the edge weights c_e.

For a square matrix A let $A^{r_1,\ldots,r_l}_{c_1,\ldots,c_l}$ denote the matrix obtained by deleting rows $r_1, \ldots, r_l$ and columns $c_1, \ldots, c_l$ from A. Let $[A]^{r_1,\ldots,r_l}_{c_1,\ldots,c_l}$ be obtained by leaving only rows $r_1, \ldots, r_l$ and columns $c_1, \ldots, c_l$ of A and placing them in this order.

Now let $\hat{V}_Q = \{v_{i_1}, \ldots, v_{i_r}\}, i_1 < \ldots < i_r$. A simple check demonstrates that deleting vertex from a graph preserves the Pfaffian orientation. By induction this holds for any number of vertices deleted. From that it follows that $K^{i_1,\ldots,i_r}_{i_1,\ldots,i_r}$ is a Kasteleyn matrix for $\hat{G}_{\setminus Q}$ and then

$$\hat{Z}_{\setminus Q} = \mathrm{Pf}\, K^{i_1,\ldots,i_r}_{i_1,\ldots,i_r} = \sqrt{\det K^{i_1,\ldots,i_r}_{i_1,\ldots,i_r}}$$

resulting in

$$\mathbb{P}(Q \subseteq M) = \sqrt{\frac{\det K^{i_1,\ldots,i_r}_{i_1,\ldots,i_r}}{\det K}} \cdot \left(\prod_{e \in Q} c_e \right)$$

Linear algebra transformations, described in [15], suggest that if A is non-singular, then

$$\frac{\det A^{r_1,\ldots,r_l}_{c_1,\ldots,c_l}}{\det A} = \pm \det [A^{-1}]^{c_1,\ldots,c_l}_{r_1,\ldots,r_l}$$

This observation allows us to express probability (2) as

$$\mathbb{P}(Q \subseteq M) = \sqrt{|\det [K^{-1}]^{i_1,\ldots,i_r}_{i_1,\ldots,i_r}|} \cdot \left(\prod_{e \in Q} c_e \right)$$

Now we are in the position to describe the first step of the Wilson's recursion.

3.3 Step 1: Computing Lower-Right Submatrix of $\overline{K}^{-1}$

Find a separation P_1, P_2, P_3 of $\hat{G}$. The goal is to sample an edge from every $v \in P_3$.

Let T be a set of vertices from P_3 and their neighbors, then $|T| \leq 3|P_3|$ because each vertex in $\hat{G}$ is of degree at most 3. Let $T^{**} \subseteq V^{**}$ be a set of the contracted edges (recall G^{**} definition from Subsection 2.3), containing at least one vertex from T, $|T^{**}| \leq |T|$. Then T^{**} is a separator of G^{**} such that

$$|T^{**}| \leq |T| \leq 3|P_3| \leq 3 \cdot 2^{\frac{3}{2}} \sqrt{\hat{N}} \leq 3 \cdot 2^2 \sqrt{|V^{**}|} \tag{3}$$

where one uses that, $|V^{**}| = \frac{\hat{N}}{2}$. Find a nested dissection ordering (Subsection 2.3) of V^{**} with T^{**} as a top-level separator. This is a correct nested dissection due to Eq. (3).

Utilizing this ordering, construct $\overline{K}$. Compute L and U - LU-decomposition of $\overline{K}$ ($O(\hat{N}^{\frac{3}{2}})$ time). Let $t = 2|T^{**}| \leq 3 \cdot 2^{\frac{5}{2}} \sqrt{\hat{N}}$ and let $\mathcal{I}$ be a shorthand notation for $(\hat{N} - t + 1, ..., \hat{N})$. Using L and U, find $D = [\overline{K}^{-1}]^{\mathcal{I}}_{\mathcal{I}}$, which is a lower-right $\overline{K}^{-1}$'s submatrix of size $t \times t$.

It is straightforward to observe that the i-th column of D, d_i, satisfies

$$[L]^{\mathcal{I}}_{\mathcal{I}} \times \left([U]^{\mathcal{I}}_{\mathcal{I}} \times d_i \right) = e_i,$$

where e_i is a zero vector with unity at the i-th position. Therefore constructing D is reduced to solving $2t$ triangular systems, each of size $t \times t$, resulting in $O(t^3) = O(\hat{N}^{\frac{3}{2}})$ required steps.

3.4 Step 2: Sampling Edges in the Separator

Now, progressing iteratively, one finds $v \in P_3$ which is not yet paired and draw an edge emanating from it. Suppose that the edges, $e_1 = \{v_{j_1}, v_{j_2}\}, ..., e_k = \{v_{j_{2k-1}}, v_{j_{2k}}\}$, are already sampled. We assume that by this point we have also computed LU-decomposition $A_k = [K^{-1}]^{j_1,...,j_{2k}}_{j_1,...,j_{2k}} = L_k U_k$ and we will update it to A_{k+1} when the new edge is drawn. Then

$$\mathbb{P}(e_1, ..., e_k \in M) = \sqrt{|\det A_k|} \prod_{j=1}^{k} c_{e_j} \tag{4}$$

Next we choose j_{2k+1} so that $v_{j_{2k+1}}$ is not saturated yet. We iterate over $v_{j_{2k+1}}$'s neighbors considered as candidates for becoming $v_{j_{2k+2}}$. Let v_j to become the next candidate, denote $e_{k+1} = \{v_{j_{2k+1}}, v_j\}$. For $n \in \mathbb{N}$ let $\alpha(n) = n + 1$ if n is odd and $\alpha(n) = n - 1$ if n is even. Then the identity

$$K^{-1} = [\overline{K}^{-1}]^{\alpha(1),\alpha(2),...,\alpha(\hat{N})}_{1,2,...,\hat{N}}, \tag{5}$$

follows from the definition of $\overline{K}$. One deduces from Eq. (5)

$$A_{k+1} = [K^{-1}]^{j_1,...,j_{2k+1},j}_{j_1,...,j_{2k+1},j} = [\overline{K}^{-1}]^{\alpha(j_1),...,\alpha(j_{2k+1}),\alpha(j)}_{j_1,...,j_{2k+1},j}$$

Constructing T^{**} one has $j_1, ..., j_{2k+1}, j, \alpha(j_1), ..., \alpha(j_{2k+1}), \alpha(j) > \hat{N} - t$. It means that A_{k+1} is a submatrix of D with permuted rows and columns, hence A_{k+1} is known.

We further observe that

$$A_{k+1} = \begin{bmatrix} A_k & y \\ r & d \end{bmatrix} = \begin{bmatrix} L_k & 0 \\ R & 1 \end{bmatrix} \begin{bmatrix} U_k & Y \\ 0 & z \end{bmatrix} = L_{k+1} U_{k+1}.$$

Therefore to update L_{k+1} and U_{k+1}, one just solves the triangular system of equations $RU_k = r$ and $L_k Y = y$, where $R^\top, r^\top, Y, y$ are of size $2k \times 2$ (this is done in $O(k^2)$ steps), and then compute $z = d - RY$ which is of the size 2×2, then set, $u = \det z$.

The probability to pair $v_{j_{2k+1}}$ and v_j is

$$\begin{aligned} \mathbb{P}(e_{k+1} \in M \,|\, e_1, ..., e_k \in M) &= \frac{\mathbb{P}(e_1, ..., e_{k+1} \in M)}{\mathbb{P}(e_1, ..., e_k \in M)} \\ &= \frac{\sqrt{|\det A_{k+1}|} \prod_{j=1}^{k+1} c_{e_j}}{\sqrt{|\det A_k|} \prod_{j=1}^{k} c_{e_j}} \end{aligned}$$

$$= \frac{c_{e_{k+1}} \sqrt{|u||\det A_k|}}{\sqrt{|\det A_k|}}$$
$$= c_{e_{k+1}} \sqrt{|u|}$$

Therefore maintaining U_{k+1} allows us to compute the required probability and draw a new edge from $v_{j_{2k+1}}$. By construction of $\hat{G}$, $v_{j_{2k+1}}$ has only 3 neighbors, therefore the complexity of this step is $O(\sum_{k=1}^{|P_3|} k^2) = O(\hat{N}^{\frac{3}{2}})$ because $|P_3| \leq 2^{\frac{3}{2}} \sqrt{\hat{N}}$.

3.5 Step 3: Recursion

Let $M_{sep} = \{e_1, e_2, ...\}$ be a set of edges drawn on the previous step, and $\hat{V}_{sep}$ be a set of vertices saturated by M_{sep}, $P_3 \subseteq \hat{V}_{sep}$. Given M_{sep}, the task of sampling $M \in \mathrm{PM}(\hat{G})$ such that $M_{sep} \subseteq M$ is reduced to sampling perfect matchings M_1 and M_2 over $\hat{G}(P_1 \setminus V_{sep})$ and $\hat{G}(P_2 \setminus V_{sep})$, respectively. Then $M = M_1 \cup M_2 \cup M_{sep}$ becomes the result of the perfect matching drawn from (2).

Even though only the first step of the Wilson's recursion was discussed so far, any further step in the recursion is done in exactly the same way with the only exception that vertex degrees may become less than 3, while in $\hat{G}$ they are exactly 3. Obviously, this does not change the iterative procedure and it also does not affect the complexity analysis.

4 Random Graph Generation

As our derivations cover the most general case of planar and K_{33}-free graphs, we want to test them on graphs which are as general as possible. Based on Lemma 6 (notice, that it provides necessary and sufficient conditions for a graph to be K_{33}-free) we implement a randomized construction of K_{33}-free graphs, which is assumed to cover most general K_{33}-free topologies.

Namely, one generates a set of K_5's and random planar graphs, attaching them by edges to a tree-like structure. For simplicity, we slightly relax the condition that random planar components should be triconnected (because it is not clear how to generate such graphs efficiently) and simply require the components to be biconnected. This can be interpreted as constructing T, where some neighbor planar nodes are merged (merging planar graphs results in another planar graph). We refer to such non-unique decomposition T' as *partially merged*. Inference and sampling algorithm suggested in Section 5 is applied with no changes to the partially merged decomposition. Our generation process consists of the following two steps.

1. **Planar graph generation.** This step accepts $N \geq 3$ as an input and generates a normal biconnected planar graph of size N along with its embedding on a plane. The details of the construction are as follows.

   First, a random embedded tree is drawn iteratively. We start with a single vertex, on each iteration choose a random vertex of an already "grown" tree, and add a new vertex connected only to the chosen vertex. Items I-V in Fig. 2 illustrate this step.

   Then we triangulate this tree by adding edges until the graph becomes biconnected and all faces are triangles, as in the Subsection 4.1 (VI in Figure 2). Next, to get a normal graph, we remove multiple edges possibly produced by triangulation (VII in Fig. 2). At this point the generation process is complete.

2. **K_{33}-free graph generation.** Here we take $N \geq 5$ as the input and generate a normal biconnected K_{33}-free graph G in a form of its partially merged decomposition T'. Namely, we generate a tree T' of graphs where each node is either a normal biconnected planar graph or K_5, and every two adjacent graphs share a virtual edge.

   The construction is greedy and is essentially a tree generation process from Step 1. We start with K_5 root and then iteratively create and attach new nodes. Let $N' < N$ be a size of the already generated graph, $N' = 5$ at first. Notice, that when a node of size n is generated, it contributes $n - 2$ new vertices to G.

   An elementary step of iteration here is as follows. If $N - N' \geq 3$, a coin is flipped and the type of new node is chosen - K_5 or planar. If $N - N' < 3$, K_5 cannot be added, so a planar type is chosen. If a planar node is added, its size is drawn uniformly in the range between 3

and $N - N' + 2$ and then the graph itself is drawn as described in Step 1. Then we attach a new node to a randomly chosen free edge of a randomly chosen node of T'. We repeat this process until G is of the desired size N. Fig. 3 illustrates the algorithm.

To obtain an Ising model from G, we sample pairwise interactions for each edge of G independently from $\mathcal{N}(0, 0.1^2)$.

Notice that the tractable Ising model generation procedure is designed in this section solely for the convenience of testing and it is not claimed to be sampling models of any particular practical interest (e.g. in statistical physics or computer science).

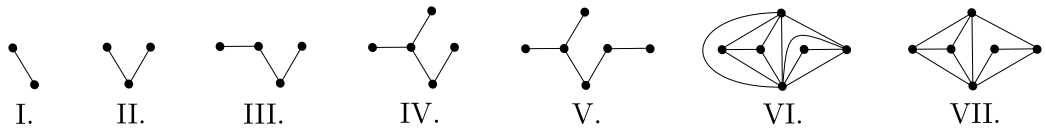


Figure 2: Steps of planar graph generation. I-V refers to random tree construction on a plane, VI is a triangulation of a tree, VII is a result after multiple edges removal.

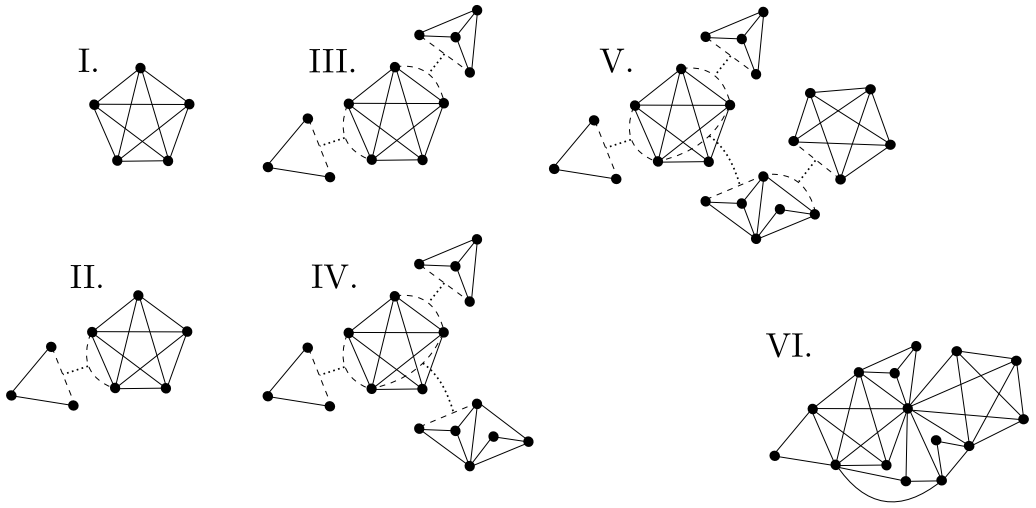


Figure 3: Generation of K_{33}-free graph G and its partially merged decomposition T'. Starting with K_5 (I), new components are generated and attached to random free edges (II-V). VI is a result graph G obtained by merging all components in T'.

5 Future Work

We conclude by discussing some future research directions:

- The class of models considered in the manuscript can be extended even further towards K_{33}-free generalizations of (a) the so-called outerplanar graphs, which can then be used for approximate inference and efficient learning in the spirit of [5] and [7] respectively; and (b) graphs embedded in the surfaces of $O(1)$ genus [11, 4, 2, 3].
- This manuscript was motivated by a larger task of using efficient inference and learning over the most general K_{33}-graphs for constructing more general (and thus, hopefully, more powerful) alternatives to traditional Neural Networks for efficient learning.

References

[1] Blum, N. A new approach to maximum matching in general graphs. In *Automata, Languages and Programming* (Berlin, Heidelberg, 1990), M. S. Paterson, Ed., Springer Berlin Heidelberg, pp. 586–597.

[2] Cimasoni, D., and Reshetikhin, N. Dimers on surface graphs and spin structures. I. *Communications in Mathematical Physics 275*, 1 (Oct 2007), 187–208.

[3] Cimasoni, D., and Reshetikhin, N. Dimers on surface graphs and spin structures. II. *Communications in Mathematical Physics 281*, 2 (Apr 2008), 445.

[4] Gallucio, A., and Loebl, M. On the theory of pfaffian orientations. I: Perfect matchings and permanents. *The Electronic Journal of Combinatorics 6*, 1 (1999), Research paper R6, 18 p.–Research paper R6, 18 p.

[5] Globerson, A., and Jaakkola, T. S. Approximate inference using planar graph decomposition. In *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. MIT Press, 2007, pp. 473–480.

[6] Horn, R. A., and Johnson, C. R. *Matrix Analysis*, 2 ed. Cambridge University Press, 2012.

[7] Johnson, J. K., Oyen, D., Chertkov, M., and Netrapalli, P. Learning planar Ising models. *J. Mach. Learn. Res. 17*, 1 (Jan. 2016), 7539–7564.

[8] Kasteleyn, P. W. Dimer statistics and phase transitions. *Journal of Mathematical Physics 4*, 2 (1963), 287–293.

[9] Lipton, R. J., Rose, D. J., and Tarjan, R. E. Generalized nested dissection. *SIAM Journal on Numerical Analysis 16*, 2 (1979), 346–358.

[10] Lipton, R. J., and Tarjan, R. E. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics 36*, 2 (1979), 177–189.

[11] Regge, T., and Zecchina, R. Combinatorial and topological approach to the 3D Ising model. *Journal of Physics A: Mathematical and General 33*, 4 (2000), 741.

[12] Schraudolph, N. N., and Kamenetsky, D. Efficient exact inference in planar Ising models. In *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2009, pp. 1417–1424.

[13] Thomason, A. The extremal function for complete minors. *J. Comb. Theory Ser. B 81*, 2 (Mar. 2001), 318–338.

[14] Vazirani, V. V. Nc algorithms for computing the number of perfect matchings in k3,3-free graphs and related problems. *Information and Computation 80*, 2 (1989), 152 – 164.

[15] Wilson, D. B. Determinant algorithms for random planar structures. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms* (Philadelphia, PA, USA, 1997), SODA '97, Society for Industrial and Applied Mathematics, pp. 258–267.