Submitted to the Annals of Applied Probability

SEQUENTIAL IMPORTANCE SAMPLING FOR ESTIMATING EXPECTATIONS OVER THE SPACE OF PERFECT MATCHINGS

BY YEGANEH ALIMOHAMMADI[1], PERSI DIACONIS[2,‡], MOHAMMAD ROGHANI[1,†] AND AMIN SABERI[1,*]

[1]*Management Science and Engineering, Stanford University, yeganeh@stanford.edu; *saberi@stanford.edu; †roghani@stanford.edu*

[2]*Department of Statistics, Stanford University, ‡diaconis@math.stanford.edu*

This paper makes three contributions to estimating the number of perfect matching in bipartite graphs. First, we prove that the popular sequential importance sampling algorithm works in polynomial time for dense bipartite graphs. More carefully, our algorithm gives a $(1-\epsilon)$-approximation for the number of perfect matchings of a λ-dense bipartite graph, using $O(n^{\frac{1-2\lambda}{8\lambda}+\epsilon^{-2}})$ samples. With size n on each side and for $\frac{1}{2} > \lambda > 0$, a λ-dense bipartite graph has all degrees greater than $(\lambda + \frac{1}{2})n$.

Second, practical applications of the algorithm requires many calls to matching algorithms. A novel preprocessing step is provided which makes significant improvements.

Third, three applications are provided. The first is for counting Latin squares, the second is a practical way of computing the greedy algorithm for a card guessing game with feedback, and the third is for stochastic block models. In all three examples, sequential importance sampling allows treating practical problems of reasonably large sizes.

1. Introduction. Given a bipartite graph $G(X, Y)$ with $|X| = |Y| = n$, a perfect matching is a subgraph of G with all vertices having degree exactly one. We study the problem of uniform sampling of perfect matchings in bipartite graphs. By classical equivalences between approximate counting and sampling [36], uniform sampling is equivalent to approximating the permanent of the adjacency matrix of G.

Computing the permanent of a general $0-1$ matrix is #P-complete [63]. A variety of algorithms have been developed to approximate the permanent. These are reviewed in Section 2.3. A highlight of this development is the Jerrum et al. [35] theorem giving a fully polynomial randomized approximation scheme (FPRAS). Alas, despite numerous attempts, they have not been very effective in practice [48, 53].

Sequential importance sampling (SIS) constructs a perfect matching sequentially by drawing each edge with weight proportional to the doubly stochastic scaling of the adjacency matrix. First suggested by [52], and actively improved and implemented by [7, 23, 53]. This seems to work well but has eluded theoretical underpinnings. The present paper corrects this.

To state the main result, the following notation is needed. For an adjacency matrix A, define its doubly-stochastic scaling Q^A, a matrix with row and column sums equal to 1, which can be written as

$$Q^A = D_\alpha A D_\beta, \qquad D_\alpha \text{ and } D_\beta \text{ are diagonal matrices.} \tag{1}$$

Since A is a strictly positive matrix, by Sinkhorn's Theorem [55], Q^A exists and it is unique.

MSC2020 subject classifications: Primary 65C05; secondary 05B15.

Keywords and phrases: Perfect Matching, Sequential Importance Sampling, Dense Bipartite Graph, KL-divergence, Latin Squares, Card Guessing Experiment, Stochastic Block Model.

Let $(M_1, X_1), \ldots, (M_N, X_N)$ be the outputs of N independent runs of Algorithm 1 with Q^A as an input, where M_i is a perfect matching and X_i is the probability of sampling M_i. To count the number of perfect matchings define the estimator as

$$I_N = \frac{1}{N} \sum_{i=1}^{N} X_i^{-1}. \tag{2}$$

Recall that $G(X, Y)$ is a λ-dense graph if $|X| = |Y| = n$ and all degrees are greater than $(\lambda + \frac{1}{2})n$.

THEOREM 1.1. *Given $\lambda \in (0, \frac{1}{2})$, let G be a λ-dense graph of size $2n$. Also, let $M(G)$ be the number of perfect matchings in G. Then there exist constant $C_\lambda, C' > 0$ such that for any given $\epsilon > 0$ and $n \geq C'$,*

$$\mathbb{E}\left(\frac{|I_N - M(G)|}{M(G)}\right) \leq 3\epsilon,$$

where $N \geq C_\lambda n^{\frac{1-2\lambda}{8\lambda} + \epsilon^{-2}}$.

The proof uses the results of Chatterjee and Diaconis [12] to bound the number of samples. In general, sequential importance sampling is used to approximate a complicated measure μ with a relatively simple measure ν. Chatterjee and Diaconis [12] characterized the necessary and sufficient number of samples by the KL-divergence from ν to μ, when $\log(d\nu/d\mu)$ is concentrated. In order to prove Theorem 1.1, we first bound the KL-divergence, and then use their result to find an upper bound on the number of samples.

Our analysis leans on the wonderful paper by Huber and Law [32]. They use a completely different approach but 'along the way' prove some estimates which are crucial to us. We hope that the approach we take will allow proofs for a wide variety of applications of sequential importance sampling. It would certainly be worthwhile to see also if the approach of Huber and Law can be adapted to some of these problem as well. It is also worth pointing out that for a fixed $\epsilon > 0$ and $\lambda < 0.1$, we get a better upper bound on the running time of our estimator compared to [32].

REMARK 1.2. Theorem 1.1 gives an upper bound on the number of samples to estimate the number of perfect matchings in G. In Corollary 3.4, we will see that the same algorithm can be used to estimate other statistics of a uniformly sampled perfect matching.

Section 4 shows that sequential importance sampling is useful in practice by implementing the algorithm for three fresh applications. The first problem is to estimate the number of Latin squares of order n. Asymptotic evaluation of this number is a long open problem. We compare three conjectured approximations using SIS.

The second problem involves a well-studied card guessing game. A deck of mn cards has n distinct values, each repeated m times. The cards are shuffled and a guessing subject tries to guess the values one at a time. After each guess the subject is told if the guess was right or wrong (not the actual value of the card if wrong). The greedy strategy – guess the most likely card each time – involves evaluating permanents. We implement SIS for this problem and give a host of examples and findings.

The third involves counting matchings in bipartite graphs generated by simple stochastic block models. Here, we investigate the benefit of using sequential importance sampling with doubly stochastic scaling (Algorithm 1 with Q^A as an input). The simulations show that with Q^A the estimator converges faster to the number of matchings in stochastic block models.

Each of the three problems begins with a separate introduction and literature review. They may be consulted now for future motivations. We have prepared a cpp code for scaled sequential importance sampling (Algorithm 1) and its implementation for examples we consider in the paper. You can access it here [1].

The rest of the paper is organized as follows. We start by going over some preliminaries in Section 2.2. Then we proceed by proving Theorem 1.1 in Section 3. Section 4 is on applications. Appendix A shows the running time of Algorithm 1 for generating one sample is $O(mn)$, where m is the number of edges in G. Moreover, it is crucial for us to use Q^A as an input of Algorithm 1. Appendix B gives an example of a class of graphs with a bounded average degree such that the original method of choosing edges uniformly at random in [21] needs an exponential number of samples, while using the doubly stochastic scaling will only need a linear number of samples. Finally, Appendix C is on computing the exact permanent of zero-blocked matrices which will be used to find the greedy strategy for card guessing games.

2. Notation and Background. In this section, we set up graph notations and give brief background on importance sampling and permanents.

2.1. *Graph Notation.* Given a graph G, let $V(G)$ be the set of vertices and $E(G)$ be the set of edges. For a node $v \in V(G)$, let $N(v)$ be the set of neighbors of v in G.

Suppose $G(X, Y)$ is a bipartite graph of size $2n$ with vertex sets $X = \{x_1, \dots, x_n\}$ and $Y = \{y_1, \dots, y_n\}$. The adjacency matrix A of a graph G is an $|X| \times |Y|$ binary matrix, where the entry A_{ij} is equal to 1 if and only if there exists an edge between x_i and y_j. Let S_n be the symmetric group of permutations on n elements. We represent a perfect matching by a permutation $\sigma \in S_n$, such that each x_i is matched to $y_{\sigma(i)}$. Then the permanent of A is defined as

$$per(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} A_{i\sigma(i)}.$$

For a graph G, let χ_G be the set of all perfect matchings in G.

2.2. *Background on Importance Sampling.* The goal of importance sampling is to estimate the expected value of a function with respect to the probability measure ν (that is hard to sample) with samples drawn from a different probability distribution μ (that is easy to sample). Assume μ and ν are two probability measures on the set χ, so that ν is absolutely continuous with respect to μ. Suppose that we want to evaluate $I(f) = \mathbb{E}_\nu(f) = \int_\chi f(x) d\nu(x)$, where f is a measurable function. Let $X_1, X_2, \dots$ be a sequence sampled from μ and $\rho = d\nu/d\mu$ be the probability density of ν with respect to μ. The importance sampling estimator of $I(f)$ is

$$I_N(f) = \frac{1}{N} \sum_{i=1}^{N} f(X_i)\rho(X_i),$$

where $\rho = d\nu/d\mu$.

In our setting, the sampling space χ is the set of perfect matchings, the target distribution ν is the uniform distribution over χ, and μ is the sampling distribution by using Algorithm 1. Note that here, $\rho = d\nu/d\mu$ is known only up to a constant. Still, we can use the estimator $I_N(f)$ to approximate the size of χ. For that purpose, we set $f(x) = |\chi|$ for all $x \in \chi$. Then for any sample X_i from the sampling distribution μ,

$$\rho(X_i)f(X_i) = \frac{1}{|\chi|\mu(X_i)} f(X_i) = \frac{1}{\mu(X_i)}.$$

Therefore, evaluating $I_N(|\chi|)$ is possible without knowing the exact value of $\rho(X)$ because,

$$I_N(|\chi|) = \frac{1}{N}\sum_{i=1}^{N}\rho(X_i)f(X_i) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\mu(X_i)}.$$

Given the estimator I_N, our goal is to bound the number of samples N. The traditional approach is to bound the variance of the estimator

$$var(I_N(f)) = \frac{1}{N}\Big(\int_{\chi} f(x)^2\rho(x)^2 d\mu(x) - I(f)^2\Big). \tag{3}$$

See for example [54, 49]. The number of samples must be greater than $var(I_N(f))/I(f)^2$ so that the standard deviation of $I_N(f)$ is bounded by $I(f)$. Here, we follow a different approach which is based on bounding the KL-divergence of the sampling distribution (μ) from the target distribution (ν), defined as $D_{KL}(\nu||\mu) = \mathbb{E}_\nu(\log\rho(Y))$. Chatterjee and Diaconis [12] found a necessary and sufficient number of samples when $\log\rho$ is concentrated with respect to ν.

THEOREM 2.1 (Theorem 1.1 in [12]). *Let $\chi, \mu, \nu, \rho, f, I(f)$ and $I_N(f)$ be as above. Let Y be an χ-valued random variable with law ν. Let $L = D_{KL}(\nu||\mu)$ be the Kullback–Leibler divergence of μ from ν, that is*

$$L = D_{KL}(\nu||\mu) = \int_{\chi}\rho(x)\log\rho(x)d\mu(x) = \int_{\chi}\log\rho(x)d\nu(x) = \mathbb{E}_\nu(\log\rho(Y)).$$

Also, let $||f||_{L^2(\nu)} := (\mathbb{E}(f(Y)^2))^{1/2} < \infty$. If $N = exp(L+t)$ for some $t \geq 0$, then

$$\mathbb{E}|I_N(f) - I(f)| \leq ||f||_{L^2(\nu)}(e^{-t/4} + 2\sqrt{\mathbb{P}_\nu(\log\rho(Y) > L + t/2)}). \tag{4}$$

Conversely, if $f = 1$, $N = exp(L - t)$ then for any $\delta \in (0,1)$,

$$\mathbb{P}_\nu(I_N(f) > 1 - \delta) \leq e^{-t/2} + \frac{1}{1-\delta}\mathbb{P}_\nu(\log(Y) < L - t/2).$$

The theorem shows that e^L samples are necessary and sufficient for keeping the expected absolute difference of the estimator and its mean small, provided $\log\rho(Y)$ is concentrated around its mean L.

For Theorem 1.1, we need to bound the number of samples required by Algorithm 1. For that purpose, we prove a logarithmic upper bound (in terms of the input size) on $L = D_{KL}(\nu||\mu)$. Then, we use the next lemma to obtain the concentration bounds needed in (4).

LEMMA 2.2. *With notation as in Theorem 2.1, we have*

$$\mathbb{P}_\nu\Big(\log\rho(Y) \geq L + t\Big) \leq \frac{L+2}{L+t}.$$

PROOF. Markov's inequality implies that

$$\mathbb{P}\Big(\log\rho(Y) > L + t\Big) \leq \frac{\mathbb{E}|\log\rho(Y)|}{L+t}.$$

Note that,

$$\mathbb{E}|\log\rho(Y)| = \mathbb{E}[\log\rho(Y)] + 2\mathbb{E}|\log\rho(Y)_-| = L + 2\mathbb{E}|\log\rho(Y)_-|,$$

where $\log \rho(Y)_- = \min(0, \log \rho(Y))$. It is enough to prove $\mathbb{E}|\log \rho(Y)_-| \le 1$. One can write

$$\begin{aligned}
\mathbb{E}|\log \rho(Y)_-| &= \int_0^\infty \mathbb{P}(\log \rho(Y) \le -x)dx \\
&= \int_0^\infty \mathbb{E}_\nu[\mathbb{1}(\rho(Y) \le e^{-x})]dx \qquad (\mathbb{1}(.) \text{ is an indicator event}) \\
&= \int_0^\infty \mathbb{E}_\mu[\rho(X)\mathbb{1}(\rho(X) \le e^{-x})]dx. \qquad (\text{change of measure})
\end{aligned}$$

To finish the proof, note that $\rho(X)\mathbb{1}(\rho(X) \le e^{-x}) \le e^{-x}$. Therefore,

$$\mathbb{E}|\log \rho(Y)_-| \le \int_0^\infty e^{-x}dx = 1.$$

□

2.3. *Background on Permanents.* The permanent of a matrix is intimately tied to the world of matchings in bipartite graphs. Fortunately, there is the bible of Lovasz and Plummer [42] which thoroughly reviews algorithms and applications to rook polynomials and other areas of combinatorics. For a wide swath of application in statistics and probability problems, see [4, 19].

Use of what amounts to sequential importance sampling to estimate permanents appears in [52]. They do not use scaling but do manage to show that the variance of a naïve SIS estimates is small for almost all bipartite graphs – of course , almost all graphs are dense. It is also shown that counting Hamiltonian circuits is a very similar problem and so should be amenable to present proof techniques. Another application of sequential importance sampling appears in [39], which uses non-uniform random permutations generated in cycle format to design an importance sampling estimator.

In [7, 31, 58], Beichl and Sullivan bring in Sinkhorn scaling. Their paper motivates this well and tries it out in a substantial real examples – dimer coverings of an $n \times n$ square lattice. They forcefully raise the need for theory. We hope that the present paper answers some of their questions. In a later work [34], they apply scaled importance sampling to counting linear extensions of a partial order.

There are host of approximation schema that give unbiased estimates of the permanents by computing the determinant of a related random matrix with elements in various fields (or even the Cayley numbers). Barvinok's fine book [5] treats some of these and [53] gives a survey, tries them out, and employs sequential importance sampling in several numerical examples. In [6], Bayati et al. took a slightly different approach to design a deterministic fully polynomial time approximation scheme for counting matchings in general graphs. They used the correlation decay, which was formerly exploited in several applications such as approximately counting the number of independent sets and colorings in some classes of graphs.

Another pocket of developments in proving things about sequential importance sampling focuses on specific graphs and gets sharp estimates for the variance and L in Theorem 2.1 above. This begins with [16], followed by [13, 21, 62]. The last is a thesis with good literature reviews. These papers illuminate the pros and cons of the weights proposed by Sinkhorn scaling. Consider G the 'Fibonacci graph' with an edge from i to i' if $|i - i'| \le 1$, for all $1 \le i, i' \le n$. This graph has F_{n+1} (Fibonacci number) perfect matchings. Consider building up a perfect matching, adding one edge at a time. Each time there are at most two choices. If the choices are made uniformly, they show that $e^{.002n}$ samples are needed. This is small for reasonable n but still exponential. They prove that if there are two choices, choosing

to transpose with probability $p = \frac{3-\sqrt{5}}{2} \sim .382$. This choice gives a sequential importance sampling algorithm which only needs a bounded number of samples, no matter what n is (see [13, 21]). What weights does Sinkhorn scaling give? We are surprised that it gives $p \sim .333$ for vertices in the middle. From [13] this choice leads to needing an exponential number of samples. The papers give several further examples with these features. Thus Sinkhorn scaling is good, but it is not perfect.

3. An SIS Algorithm for Counting the Number of Perfect Matchings. We start by formalizing the SIS algorithm to sample a perfect matching in general bipartite graphs in Section 3.1. The algorithm starts with an empty set and generates a perfect matching by adding edges sequentially. At each step, the algorithm keeps a partial matching along with the probability of generating it.

3.1. *Algorithm for General Bipartite Graphs.* Earlier authors [16, 21, 52] analyzed an importance sampling algorithm that constructs matchings sequentially by drawing edges uniformly at random. Here, we modify the algorithm by adding the possibility of drawing edges with respect to weights given by an input matrix Q. To formalize the algorithm, we need the following notation. Given a bipartite graph $G(X, Y)$ with $|X| = |Y| = n$ and a partial matching M, call an edge e *M-extendable* if there exists a perfect matching in G that contains $M \cup \{e\}$.

To sample a perfect matching, take a nonnegative matrix Q as input, and construct the matching sequentially as follows. First, draw a permutation π over the vertices in X. Then start with $M = \emptyset$, an empty matching. At step i, match the $\pi(i)^{th}$ vertex of X according to the following two criteria: 1) The pair must be M-extendable. 2) From the extendable edges, the match is chosen randomly proportional to the weights given by row $\pi(i)$ of Q. After n steps, the algorithm returns a matching M, and the probability of generating the matching $p_{\pi,Q}(M)$ with respect to π and Q. Assume that $Q_e > 0$ for all edges e, otherwise remove the edge e from graph. Then if the graph has a perfect matching, the algorithm will never fail since by condition 1 we always choose an extendable edge.

Algorithm 1: Sequential Importance Sampling of Perfect Matchings

1 **Input:** a bipartite graph $G(X, Y)$, and a nonnegative matrix $Q_{n\times n}$.
2 Draw a random permutation π on vertices in X.
3 Set $M = \emptyset$ and $p_{\pi,Q}(M) = 1$.
4 **for** *i from 1 to n* **do**
5 Find the set of extendable neighbors of $x_{\pi(i)}$ (call it $N_{\pi(i)}$).
6 Find the restriction of row $\pi(i)$ of Q to indices in $N_{\pi(i)}$ (call it $Q_{\pi(i)}[N_{\pi(i)}]$).
7 Let i^* be a random index in $N_{\pi(i)}$ drawn with respect to $Q_{\pi(i)}[N_{\pi(i)}]$.
8 $M = M \cup (x_{\pi(i)}, i^*)$.
9 $p_{\pi,Q}(M) = p_{\pi,Q}(M) \times \frac{Q_{\pi(i),i^*}}{\sum_{j\in N_{\pi(i)}} Q_{\pi(i),j}}$.
10 **end**
11 **Input:** M, $p_{\pi,Q}(M)$.

REMARK 3.1. Step 5 of the algorithm might seem challenging to implement. In Appendix A, we show a fast implementation of it using the Dulmage-Mendelsohn decomposition [42].

To count the number of perfect matchings, let $(M_1, p_{\pi_1,Q}(M_1)), \ldots, (M_N, p_{\pi_N,Q}(M_N))$ be outputs of N runs of Algorithm 1 with Q as an input. Then recall the estimator for the

number of perfect matchings (2),

$$I_N = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{p_{\pi_i,Q}(M_i)}.$$

The key observation is that I_N is an unbiased estimator on the number of perfect matchings $|\chi_G|$. This observation is a basic step that has been used in different applications of sequential importance sampling [16, 9].

PROPOSITION 3.2. *Given a bipartite graph G, assume that $Q_e > 0$ for all edges e. Define I_N as above. Then*

$$\mathbb{E}[I_N] = |\chi_G|,$$

where the expectation is over the randomness of samples generated by Algorithm 1.

PROOF. When there is no perfect matching in G, the algorithm will find an empty set of extendable edges, first time that it reaches on line 5. Then it always returns 0, which is an unbiased estimator. For the case that there exists a perfect matching in G, the result is immediate by the linearity of expectation,

$$\begin{aligned}\mathbb{E}[I_N] &= \mathbb{E}[\frac{1}{p_{\pi_1,Q}(M_1)}]\\ &= \sum_{\pi\in S_n, M\in\chi_G}\frac{1}{n!}p_{\pi,Q}(M)\frac{1}{p_{\pi,Q}(M)}\\ &= \sum_{\pi\in S_n, M\in\chi_G}\frac{1}{n!} = |\chi_G|.\end{aligned}$$

Note that the second equality holds because for any $\pi \in S_n$ and any perfect matching M, $p_{\pi,Q}(M) > 0$, since by the assumption, $Q_e > 0$ for any edge e in M. □

The above proposition proves that the algorithm can be applied to, and gives an unbiased estimator for the number of perfect matchings, for *any* bipartite graph G. In what follows, we focus on the case where G is dense and bound the number of samples, proving Theorem 1.1. Recall that for some constant $\frac{1}{2} > \lambda > 0$, the bipartite graph $G(X, Y)$ with $|X| = |Y| = n$ is called λ-dense if the degree of each vertex is at least $(\frac{1}{2} + \lambda)n$. For the rest of this section we assume that G is a λ-dense graph with the adjacency matrix A.

Also, note that the variance of the estimator, and the number of samples needed to obtain a certain degree of accuracy, depends on the input to the algorithm Q. We use the doubly stochastic scaling of the adjacency matrix Q^A, where

$$Q^A = D_\alpha A D_\beta, \qquad D_\alpha \text{ and } D_\beta \text{ are diagonal matrices.} \tag{5}$$

Finding a $(1-\epsilon)$-approximation of the doubly stochastic matrix is possible in time $\tilde{O}(m + n^{4/3})$, where m is the number of edges and $\tilde{O}$ hides a poly-logarithmic factor of n and $1/\epsilon$ [2].

Let μ be the sampling distribution resulting from Algorithm 1 with Q^A as the input. The following key lemma, which will be used in the proof of Theorem 1.1, gives an upper bound on the KL-divergence from μ to the uniform distribution.

LEMMA 3.3. *Given a λ-dense graph G, with Q^A as defined in (5), let μ be the probability distribution that a perfect matching is generated by Algorithm 1 with Q^A as an input, and ν be the uniform distribution over the set of perfect matchings in G. Then there exists a constant $N_\lambda > 0$ such that for $n \geq N_\lambda$,*

$$D_{KL}(\nu||\mu) \leq (\frac{1}{8\lambda} - \frac{1}{4}) \log n + \frac{1}{2\lambda} + \frac{1}{\lambda} \log(\frac{1}{2\lambda}).$$

We will give the proof of Lemma 3.3 in Section 3.2. Now we are ready to prove Theorem 1.1.

PROOF OF THEOREM 1.1. Let L be as defined in Theorem 2.1. Then by Lemma 3.3

$$L = D_{KL}(\nu||\mu) \leq (\frac{1}{8\lambda} - \frac{1}{4}) \log n + \frac{1}{2\lambda} + 2\log(\frac{1}{2\lambda}).$$

We apply Theorem 2.1 for $N = e^{L+t}$, which gives

$$\mathbb{E}(\frac{|I_N - M(G)|}{M(G)}) \leq e^{-t/4} + 2\sqrt{\mathbb{P}(\rho(Y) > L + t)} \leq e^{-t/4} + 2\sqrt{\frac{L+2}{L+t}},$$

where the second inequality is by Lemma 2.2. Now, letting $\epsilon^2 = \frac{L+2}{L+t}$ implies that $t = (L(1 - \epsilon) + 2)/\epsilon^2 =$, which implies, $N = e^{t+L} = C_\lambda n^{\frac{1-2\lambda}{8\lambda} + \epsilon^{-2}}$ samples are enough for

$$\mathbb{E}(\frac{|I_N - M(G)|}{M(G)}) \leq e^{-t/4} + 2\sqrt{\frac{L+2}{L+t}} \leq 3\epsilon,$$

where C_λ is a constant that is independent of n. □

As a result of Theorem 1.1, we have a theoretical upper bound on the number of samples needed by SIS to approximate the number of matchings in dense graphs. Algorithm 1 can be used beyond estimating the number of perfect matchings. Indeed Lemma 3.3 allows us to estimate the expected value of a wide range of statistics over the uniform sample of perfect matchings.

For a function f over the space of matchings, suppose that we want to estimate $I(f) = \mathbb{E}_{Y\sim\nu}(f(Y))$, where ν is the uniform distribution over prefect matchings. As before, let $(M_1, X_1), \ldots, (M_N, X_N)$ be N independent samples of Algorithm 1. Then define the estimator

$$J_N(f) = \frac{\sum_{i=1}^N f(M_i) X_i^{-1}}{\sum_{i=1}^N X_i^{-1}}.$$

Recall that $||f||_{L^2(\nu)} := \left(\mathbb{E}_{Y\sim\nu}(f(Y)^2)\right)^{1/2}$. The following gives an upper bound on the sufficient number of samples when $\frac{||f||_{L^2(\nu)}}{I(f)}$ is bounded by a constant.

COROLLARY 3.4. *Given $\lambda > 0$, let G be a λ-dense graph of size $2n$. Also let μ, ν, f, $I(f)$, $J_N(f)$, be defined as above. Suppose that there exists a constant C such that $||f||_{L^2(\nu)} \leq CI(f)$. Then given any $\epsilon > 0$ there exists a constant C_λ such that for $N \geq C_\lambda n^{\frac{1-2\lambda}{8\lambda} + \epsilon^{-4}}$,*

$$\mathbb{P}\Big(\frac{|J_N(f) - I(f)|}{I(f)} \geq C\epsilon\Big) \leq 4\epsilon.$$

PROOF. By Theorem 1.2 in [12], if we let $\epsilon' = (e^{-t/4} + 2\sqrt{\mathbb{P}(\log \rho(Y) > L + t/2)})^{1/2}$, then

$$\mathbb{P}\Big(|J_N(f) - I(f)| \geq \frac{2||f||_{L^2(\nu)}\epsilon'}{1-\epsilon'}\Big) \leq 2\epsilon'.$$

By Lemma 2.2, and for $t = 2(L+2)\epsilon^4$

$$\epsilon' \leq \Big(e^{-t/4} + 2\sqrt{\frac{L+2}{L+t/2}}\Big)^{1/2} \leq 2\epsilon.$$

Then since $||f||_{L^2(\nu)} \leq CI(f)$,

$$\mathbb{P}\Big(\frac{|J_N(f) - I(f)|}{I(f)} \geq C\epsilon\Big) \leq 4\epsilon.$$

Then Lemma 3.3 implies the desired upper bound on the number of samples, N. □

3.2. *Bounding the KL-divergence for Dense Graphs.* The purpose of this section is to bound the KL-divergence of μ from the uniform distribution ν on the set of perfect matchings. The main idea is to show that if the entries of Q are small, then $D_{KL}(\nu||\mu)$ is a convex function of the entries in Q (see Lemma 3.8). Convexity of $D_{KL}(\nu||\mu)$ enables us to use the Bregman's inequality along with the Van der Waerden lower bound on $per(A)$ to get a logarithmic upper bound on $D_{KL}(\nu||\mu)$.

Let P be the matrix of marginal probabilities, i.e., for an edge $e = (u, v)$, let $P_e = P_{uv}$ be equal to the probability that the edge e appears in a perfect matching chosen uniformly at random. Before proving the convexity of $D_{KL}(\nu||\mu)$, we need Lemmas 3.5 and 3.6 to show that Q^A_e/P_e is bounded from above by some constant c.

LEMMA 3.5 (Lemma 4.4 in [32]). *Let $G(X, Y)$ be a λ-dense bipartite graph with the adjacency matrix A, and Q^A its doubly stochastic scaling. Then for all $e \in [n] \times [n]$, $Q^A_e \leq \frac{1}{2\lambda n}$.*

LEMMA 3.6. *Let $G(X, Y)$ be a λ-dense bipartite graph and P be the matrix of matching marginals. Then there exists a constant $c_\lambda > 0$ independent from n such that for all $e \in E(G)$ we have $\frac{c_\lambda}{n} \leq P_e \leq \frac{1}{\lambda n}$.*

PROOF. To give the proof we first need the following notations. Given a graph H, let χ_H be the set of all perfect matchings in H. For any set $L \subset [n] \times [n]$, define G_L as the graph constructed by removing vertices appearing in L from G. Note that $P_e = \frac{|\chi_{G_e}|}{|\chi_G|}$, where $e = (x, y)$.

We start by proving the lower bound on P_e. Define $\Omega_e = \cup_{e' \in ([n]\setminus\{x\})\times([n]\setminus\{y\})} \chi_{G_{\{e,e'\}}}$, where the union is over all pairs of vertices in G_e. First, we prove that $2|\Omega_e| \geq |\chi_G|$ by defining a mapping $\phi : \chi_G \mapsto \Omega_e$ as follows. Suppose $M \in \chi_G$, i.e., M is a perfect matching in G. If $e \in M$, then let $\phi(M) = M \setminus (e \cup e')$, for an arbitrary $e' \in M$ such that $e' \neq e$. Otherwise $\phi(M) = M \setminus ((x, M(x)), (y, M(y))$. Now observe that at most two matchings in χ_G are mapped to the same matching in Ω_e and hence $2|\Omega_e| \geq |\chi_G|$.

Therefore, $P_e = \frac{|\chi_{G_e}|}{|\chi_G|} \geq \frac{|\chi_{G_e}|}{2|\Omega_e|}$ and to prove the statement, it is enough to prove there exists a constant $C_\lambda > 0$ such that

$$\frac{|\chi_{G_e}|}{|\Omega_e|} \geq \frac{C_\lambda}{n}. \tag{6}$$

In what remains, we prove (6). Let $H = G_e$. We can write

$$|\Omega_e| = |\cup_{(u,v)\in E(H)} \chi_{H_{(u,v)}}| + |\cup_{(u,v)\notin E(H)} \chi_{H_{(u,v)}}|.$$

The first term is equal to $(n-1)|\chi_H|$. To see this, let $\phi' : \cup_{(u,v)\in E(H)}\chi_{H_{(u,v)}} \to \chi_H$ map each element $M' \in \chi_{H_{(u,v)}}$ to $M' \cup \{(u,v)\} \in \chi_H$. Note that each matching in H is an image of $n-1$ matchings in $\cup_{(u,v)\in E(H)}\chi_{H_{(u,v)}}$, since by removing any edge from any matching in H, we get an element in $\cup_{(u,v)\in E(H)}\chi_{H_{(u,v)}}$ and there are $n-1$ edges to remove from a matching in H.

Next, we bound the second term $|\cup_{(u,v)\notin E(H)} \chi_{H_{(u,v)}}|$ with $\frac{n}{\lambda}|\chi_H|$. For that purpose, we use a many-to-many mapping between elements in $\cup_{(u,v)\notin E(H)}\chi_{H_{(u,v)}}$ and χ_H.

First, observe that for any $M \in \chi_{H_{(u,v)}}$, there are at least λn edges $(w,z) \in M$ such that $(u,z),(w,v) \in E(G)$. In other words, $(v,w),(w,z),(z,u)$ form a path of length 3 in H. To see this, note that $|N(v)| > n/2 + \lambda n$ and $|N(u)| > n/2 + \lambda n$, and every neighbor of u has a matched vertex in M. Hence, there are at least λn vertices in common between $N(v)$ and matched of vertices in $N(u)$. Now, for each such edge $(w,z) \in M$, map M to $\big(M \setminus \{(w,z)\}\big) \cup \{(w,v),(z,u)\}$, a perfect matching in H. Therefore, each element in $\cup_{(u,v)\notin E(H)}\chi_{H_{(u,v)}}$ is mapped to at least λn elements in χ_H

On the other hand, a matching $M' \in \chi_H$ is mapped to matchings of the form $M' \setminus \{(u,M'(u)),(v,M'(v))\} \cup (M'(u),M'(v))$, for all pairs (u,v) for which u and v not adjacent in H but their matches $M'(u)$ and $M'(v)$ are adjacent. There are at most n^2 candidates for such (u,v) pairs. Hence, at most n^2 elements of $\chi_{H_{(u,v)}}$ are mapped to a perfect matching in χ_H. As a result,

$$|\cup_{(u,v)\notin E(G)} \chi_{H_{(u,v)}}| \le \frac{n}{\lambda}|\chi_H|.$$

Therefore,

$$|\Omega_e| = |\cup_{(u,v)\in E(H)} \chi_{H_{(u,v)}}| + |\cup_{(u,v)\notin E(H)} \chi_{H_{(u,v)}}| \le n|\chi_H| + \frac{n}{\lambda}|\chi_H| = |\chi_{G_e}|(n + \frac{n}{\lambda}),$$

which implies the result, because

$$P_e \ge \frac{|\chi_{G_e}|}{2|\Omega_e|} \ge \frac{\lambda}{2n(\lambda+1)}.$$

It remains to prove the upper bound on P_e. Recall that $P_e = \frac{|\chi_{G_e}|}{|\chi_G|}$, where $e = (x,y)$. We prove $\lambda n \cdot |\chi_{G_e}| \le |\chi_G|$, by defining a one-to-many mapping that maps every element of χ_{G_e} to at least λn distinct elements of χ_G. Fix $M \in \chi_{G_e}$. By an argument similar to the previous bound, there are at least λn edges $(w,z) \in M$ such that $(x,z),(y,w) \in E(G)$. For each such edge $(w,z) \in M$, note that $\big(M \setminus \{(w,z)\}\big) \cup \{(y,w),(x,z)\} \in \chi_G$, so map M to $\big(M \setminus \{(w,z)\}\big) \cup \{(y,w),(x,z)\}$ for every such edge in addition to $M \cup e$. Now, we need to show that at most one element from χ_{G_e} is mapped to a matching $M' \in \chi_G$. This is because if M' contains the edge e, then M' is the image of $M' \setminus e$, and if $e \notin M'$ then it is the image of $M' \setminus \{(x,M'(x)),(y,M'(y)\}$. As a result, $\lambda n \cdot |\chi_{G_e}| \le |\chi_G|$. Therefore,

$$P_e = \frac{|\chi_{G_e}|}{|\chi_G|} \le \frac{1}{\lambda n}.$$

□

We will use the next result to relate $p_{\pi,Q}(M)$ to the matching marginals matrix P, which will be useful for bounding the KL-divergence. Define the ordering $<_\pi$ on $\{1,2,\ldots,n\}$ as $i <_\pi j$ iff $\pi(i) < \pi(j)$.

PROPOSITION 3.7. *Given a bipartite graph $G(X,Y)$ of size $2n$, let P be the matrix of matching marginals and Q be a nonnegative matrix, such that for any $e \in [n]\times[n]$ if $P_e \neq 0$, then $Q_e \neq 0$. Then for the samples generated by Algorithm 1,*

$$\mathbb{E}_{M\sim\nu,\pi\sim S_n}\Big(\log(p_{\pi,Q}(M))\Big) \geq \sum_{e\in[n]\times[n]} P_e \log(Q_e) - \sum_{i=1}^{n}\sum_{k=1}^{n} P_{ik}\mathbb{E}_{\pi\sim S_n}\log(\sum_{j\geq_\pi k} Q_{i,j}),$$

where the expectation is over ν, the uniform distribution over all perfect matchings in G, and the random choice of π.

PROOF. The proof is similar to derivation of Equation (4) in [3], except their result is for the case that $Q = P$. Let A be the adjacency matrix of G. Then for a perfect matching M note that

$$\nu(M) = \frac{1}{per(A)}.$$

For a matching M, let $M(v)$ be the vertex matched to v in M. The probability that at step i we match $x_{\pi(i)}$ to $y_{M(\pi(i))}$ is at least $\frac{Q_{\pi(i),M(\pi(i))}}{\sum_{j=i}^{n} Q_{\pi(i)M(\pi(j))}}$, because the set of extendable neighbors of $x_{\pi(i)}$ is a subset of available vertices at step i, i.e., the set $\{y_{M(\pi(i))}, y_{M(\pi(i+1))}, \ldots, y_{M(\pi(n))}\}$. Therefore,

$$p_{\pi,Q}(M) \geq \prod_{i=1}^{n} \frac{Q_{\pi(i),M(\pi(i))}}{\sum_{j=i}^{n} Q_{\pi(i)M(\pi(j))}}.$$

As a result,

(7)
$$\mathbb{E}_{M\sim\nu,\pi\sim S_n}\Big(\log(p_{\pi,Q}(M))\Big) \geq \mathbb{E}_{M\sim\nu,\pi\sim S_n}\Big(\log\prod_{i=1}^{n} \frac{Q_{\pi(i),M(\pi(i))}}{\sum_{j=i}^{n} Q_{\pi(i)M(\pi(j))}}\Big)$$

(8)
$$= \sum_{e} P_e \log(Q_e) - \sum_{i=1}^{n} \mathbb{E}_{M\sim\nu,\pi\sim S_n}\Big(\log\big(\sum_{j=i}^{n} Q_{\pi(i)M(\pi(j))}\big)\Big),$$

where in the second equality, we used the fact that each edge e appears with probability P_e in a matching drawn from ν.

Next, we use the fact that when π is a uniform random permutation, then $M\circ\pi$ is also a uniform random permutation. Therefore,

$$\sum_{i=1}^{n} \mathbb{E}_{M\sim\nu,\pi\sim S_n}\Big(\log\big(\sum_{j\geq i} Q_{\pi(i)M(\pi(j))}\big)\Big) = \sum_{i=1}^{n} \mathbb{E}_{M\sim\nu,\pi\sim S_n}\Big(\log\big(\sum_{j\geq_\pi M(i)} Q_{ij}\big)\Big).$$

Also, since each edge e is drawn with probability P_e in $M\sim\nu$,

$$\sum_{i=1}^{n} \mathbb{E}_{M\sim\nu,\pi\sim S_n}\Big(\log\big(\sum_{j\geq_\pi M(i)} Q_{ij}\big)\Big) = \sum_{i=1}^{n}\sum_{k=1}^{n} P_{ik}\mathbb{E}_{\pi\sim S_n}\Big(\log\big(\sum_{j\geq_\pi k} Q_{ij}\big)\Big),$$

which gives the statement of the result. □

Define $r_i = n\cdot\max_{1\leq j\leq n} Q^A_{ij}$, for $i\in[n]$. Also, define $\mathcal{Q}$ to be the subset of row-stochastic matrices such that for each $Q\in\mathcal{Q}$, (i) the entries of row i in Q are at most $\frac{r_i}{n}$, and (ii) each non-zero entry of Q corresponds to a non-zero entry in Q^A, i.e. if $Q_{ij}>0$ then $Q^A_{ij}>0$. The following result will find a matrix $Q^*\in\mathcal{Q}$ with (almost) equal entries in each row, such that

roughly speaking, running Algorithm 1 on Q^* would result in a larger KL-divergence than running it on Q^A. We will see that this is beneficial, because the simple structure of Q^* will allow us to prove a certain 'Bregman-like' upper bound on the KL-divergence corresponding to Q^*.

LEMMA 3.8. *Let G be a λ-dense graph of size n, and let $\mathcal{Q}$, and P be defined as above. Then for large enough n, there exists a matrix $Q^* \in \mathcal{Q}$ such that,*

1. *For each row i, all the non-zero entries (except at most one) are equal to $\frac{r_i}{n}$. Equivalently, let $\ell_1, \dots \ell_k$ be the indices of non-zero entries in the row i of Q^*, then $Q^*_{i\ell_2} = \cdots = Q^*_{i\ell_k} = \frac{r_i}{n}$.*
2. *Further,*

$$-\sum_{e\in[n]\times[n]} P_e \log(Q^*_e) + \sum_{t,k} P_{tk}\mathbb{E}_\pi \log\Big(\sum_{j\geq_\pi k} Q^*_{tj}\Big) \geq \mathbb{E}_{M\sim\nu}\Big(-\log(p_{\pi,Q^A}(M))\Big),$$

where ν is the uniform distribution on the set of perfect matchings in G.

PROOF. Fix a matrix $Q \in \mathcal{Q}$ and a row i. Also, fix all the entries of Q except Q_{ij_1} and Q_{ij_2}. Let $Q_{ij_1} + Q_{ij_2} = s$. By Proposition 3.7,

$$\mathbb{E}_{\sigma\sim\nu,\pi}(-\log(p_{\pi,Q}(\sigma))) \leq -\sum_{e\in[n]\times[n]} P_e \log(Q_e) + \sum_{t,k} P_{tk}\mathbb{E}_\pi \log(\sum_{j\geq_\pi k} Q_{tj}).$$

Define the function $g(x) = -\sum_{e\in[n]\times[n]} P_e \log(Q_e) + \sum_{t,k} P_{tk}\mathbb{E}_\pi \log(\sum_{j\geq_\pi k} Q_{tj})$, where $Q_{ij_1} = x$, $Q_{ij_2} = s - x$ and all other entries of Q are fixed. We will prove that $g(x)$ is a convex function when $x \in [0, s]$. But first, let us prove the lemma assuming that is the case.

Since g is convex, it attains its maximum either at $x = 0$ or $x = s$. Therefore for any two entries of row i of Q that are not in $\{0, \frac{r_i}{n}\}$, we can increase the value of

$$-\sum_{e\in[n]\times[n]} P_e \log(Q_e) + \sum_{t,k} P_{tk}\mathbb{E}_\pi \log\Big(\sum_{j\geq_\pi k} Q_{tj}\Big), \tag{9}$$

by making one of the entries either 0 or $\frac{r_i}{n}$ while keeping the other one in the interval $[0, \frac{r_i}{n}]$ and keeping their sum fixed. This operation does not change the row sums and therefore keeps Q within $\mathcal{Q}$. By repeating this operation we can get a matrix $Q^* \in \mathcal{Q}$ such that each entry of row i (except at most one) is in $\{0, \frac{r_i}{n}\}$, while only increasing the value of (9), hence proving the theorem.

To prove the convexity, we show the second derivative of g is positive. For simplicity, fix i and let $q_j = Q_{ij}$ and $P_j = P_{ij}$. Note that $\sum_{j\geq_\pi k} q_j$ is a constant when $j_1, j_2 \leq_\pi k$ or $j_1, j_2 \geq_\pi k$. Therefore,

$$\begin{aligned} g''(x) =& \frac{P_{j_1}}{x^2} + \frac{P_{j_2}}{(s-x)^2} \\ &- \sum_{k\neq j_1,j_2} P_k\Big(\sum_{\pi: j_1\geq_\pi k>_\pi j_2} \frac{\mathbb{P}(\pi)}{(\sum_{l\geq_\pi k} q_l)^2} + \sum_{\pi: j_2\geq_\pi k>_\pi j_1} \frac{\mathbb{P}(\pi)}{(\sum_{l\geq_\pi k} q_l)^2}\Big) \\ &- P_{j_1}\sum_{\pi: j_1>_\pi j_2} \frac{\mathbb{P}(\pi)}{(\sum_{l\geq_\pi j_1} q_l)^2} - P_{j_2}\sum_{\pi: j_2>_\pi j_1} \frac{\mathbb{P}(\pi)}{(\sum_{l\geq_\pi j_2} q_l)^2}. \end{aligned}$$

Note that here $P(\pi) = \frac{1}{n!}$ is the probability that a uniform permutation is equal to π.

Since, $x^2 < (x + \sum_{\pi: l >_\pi j_1} q_l)^2$ and $\sum_{\pi: j_1 >_\pi j_2} \mathbb{P}(\pi) = 1/2$, it is enough to prove

$$\frac{P_{j_1}}{2x^2} + \frac{P_{j_2}}{2(s-x)^2} \geq \sum_{k \neq j_1, j_2} P_k \Big(\sum_{\pi: j_1 \geq_\pi k >_\pi j_2} \frac{\mathbb{P}(\pi)}{(\sum_{l \geq_\pi k} q_l)^2} + \sum_{\pi: j_2 \geq_\pi k >_\pi j_1} \frac{\mathbb{P}(\pi)}{(\sum_{l \geq_\pi k} q_l)^2} \Big).$$

Now, since the entries of a permutation over $n-2$ entries of row i are negatively associated [37], by the Chernoff inequality (e.g., see [22])

$$\mathbb{P}_\pi \Big(\sum_{j=1}^{t} q_{\pi(j)} \leq \frac{\mathbb{E}_\pi[\sum_{j=1}^{t} q_{\pi(j)}]}{2} \Big) \leq \exp\Big(- \frac{\mathbb{E}_\pi[\sum_{j=1}^{t} q_{\pi(j)}]^2 n^2}{2t} \Big). \tag{10}$$

To compute the expectation note that for each j, $\mathbb{E}_\pi[q_{\pi(j)}] = \frac{1}{n}$ because Q is row-stochastic. Hence, by the linearity of expectation $\mathbb{E}_\pi(\sum_{j=1}^{t} q_{\pi(j)}) = \frac{t}{n}$. Further, observe that $\mathbb{P}(\pi(t) = k, j_1 <_\pi k \leq_\pi j_2) \leq \frac{t}{n(n-1)}$. Then

$$\begin{aligned} \sum_{\pi: j_1 \geq_\pi k >_\pi j_2} \frac{\mathbb{P}(\pi)}{(\sum_{l \geq_\pi k} q_l)^2} &\leq \sum_{t=1}^{\lceil \log n \rceil} \frac{t}{n(n-1)x^2} + \sum_{t=\lceil \log n \rceil + 1}^{n} \frac{t}{n(n-1)} \Big(\frac{1}{(x + \frac{t-1}{2n})^2} + \frac{e^{-t/2}}{x^2} \Big) \\ &\leq \frac{\log^2 n}{n(n-1)x^2} + 4\log(4xn+n) + \frac{2\log n}{n(n-1)x^2}, \end{aligned}$$

where in the first inequality we used (10) and the fact that $\sum_{l \geq_\pi k} q_l \geq x$ when $j_1 \geq_\pi k$. A similar inequality holds for $s - x$. Note that for large enough n, we have $2\log n \leq \log^2 n$. Therefore,

$$g''(x) \geq \frac{P_{j_1}}{2x^2} + \frac{P_{j_2}}{2(s-x)^2} - \frac{2\log^2 n}{(nx)^2} - 4\log(4xn+n) - \frac{2\log n}{(n(s-x))^2} - 4\log(4(s-x)n+n).$$

Note that for $n \geq (4r_i+1)^4$ we have

$$\frac{\log^2 n}{(nx)^2} \geq \frac{\log^2 n}{(r_i)^2} \geq 4\log(4r_i+n) \geq 4\log(4xn+n).$$

As a result,

$$g''(x) \geq \frac{P_{j_1}}{2x^2} + \frac{P_{j_2}}{2(s-x)^2} - \frac{3\log^2 n}{(nx)^2} - \frac{3\log^2 n}{(n(s-x))^2}.$$

By Lemma 3.6 and the fact that $x, s-x \in [0, \frac{r_i}{n}]$ there exists $C > 0$ such that both P_{j_1} and P_{j_2} are at least C/n. Therefore, $g''(x) \geq 0$ for $n \geq \max\left(3C^{-1}, (4r_i+1)^4\right)$, which proves the convexity of g. □

Now, we are ready to bound the KL-divergence.

PROOF OF LEMMA 3.3. Let π be the permutation that is chosen by Algorithm 1. Since ν assigns probability $\frac{1}{per(A)}$ to each perfect matching,

$$D_{KL}(\nu||\mu) = \mathbb{E}_{M \sim \mu}\Big(-\log(p_{\pi, Q^A}(M)) \Big) - \log(per(A)).$$

Let Q_i^A denote the i^{th} row of Q^A and let $\frac{r_i}{n}$ be the maximum entry of Q_i^A. First, we use Lemma 3.8 to upper bound the first term:

$$\mathbb{E}_{M\sim\mu}\Big(-\log(p_{\pi,Q^A}(M))\Big) \leq -\sum_{e\in[n]\times[n]} P_e \log(Q_e^*) + \sum_{i,k} P_{ik}\mathbb{E}_\pi \log(\sum_{j\geq_\pi k} Q_{ij}^*),$$

where in Q^* row sums are equal to 1 and each entry of Q_i^* (except at most one) is in $\{0, \frac{r_i}{n}\}$. Then, we prove

$$-\sum_{e\in[n]\times[n]} P_e \log(Q_e^*) + \sum_{i,k} P_{ik}\mathbb{E}_\pi \log(\sum_{j\geq_\pi k} Q_{ij}^*) \leq \log\prod_{i=1}^n \Big((\lfloor\frac{n}{r_i}\rfloor+1)!^{\frac{r_i}{n}} r_i^{\frac{1}{\lambda n}}\Big) \tag{11}$$

and

$$\log(per(A)) \geq \log\left(e^{-n}\sqrt{2\pi n}\prod_i \frac{n}{r_i}\right). \tag{12}$$

Assuming the above inequalities are true, we will finish the proof of the lemma. By combining inequalities (11) and (12), and using Stirling's approximation in (11),

$$\begin{aligned} D_{KL}(\nu||\mu) &\leq \log\Big(\frac{\prod_{i=1}^n \big(\sqrt{2\pi(\frac{n}{r_i}+1)}^{\frac{r_i}{n}}(\frac{n}{r_i}+1)e^{-1}r_i^{\frac{1}{\lambda n}}\big)}{(\prod_{i=1}^n \frac{n}{r_i})e^{-n}\sqrt{2\pi n}}\Big) \\ &= \sum_{i=1}^n \Big(\frac{r_i}{2n}\log(\frac{1}{r_i}+\frac{1}{n}) + \log(1+\frac{r_i}{n}) + \frac{1}{\lambda n}\log(r_i)\Big) \\ &\qquad + (\sum_{i=1}^n \frac{r_i}{2n})\log(2\pi n) - \frac{1}{2}\log(2\pi n) \\ &\leq (\frac{1}{2n}\sum_{i=1}^n r_i - \frac{1}{2})\log(2\pi n) + \frac{1}{2\lambda} + \frac{1}{\lambda}\log(\frac{1}{2\lambda}), \end{aligned}$$

where in the last inequality we used the facts that $r_i \leq (2\lambda)^{-1}$ by Lemma 3.5, and that for large enough n, we have $\log(\frac{1}{r_i}+\frac{1}{n}) \leq \log(2\lambda+\frac{1}{n}) \leq 0$, and $\log(1+\frac{r_i}{n}) \leq \frac{r_i}{n} \leq (2\lambda n)^{-1}$.

In order to bound $\sum_i r_i$, assume without loss of generality that $\beta_1 = \max_j(\beta_j)$. If A_{i1} is non-zero in row i, then Q_{i1}^A is equal to $\beta_1\alpha_i$. Furthermore, it is the highest element in row i and therefore $\frac{r_i}{n} = \beta_1\alpha_i$. Also, since Q^A is doubly stochastic, $\beta_1\sum_{i\sim 1}\alpha_i \leq 1$ where $i\sim 1$ denotes that $A_{i1}\neq 0$. Therefore, by writing the first column sum,

$$\frac{1}{n}\sum_{i\sim 1} r_i = \beta_1(\sum_{i\sim 1}\alpha_i) \leq 1.$$

For the rows $i \not\sim 1$, use Lemma 3.5 which implies $r_i \leq \frac{1}{2\lambda}$. Therefore,

$$\frac{1}{n}\sum_{i=1}^n r_i = \frac{1}{n}\sum_{i\sim 1} r_i + \frac{1}{n}\sum_{i\not\sim 1} r_i \leq 1 + n(1/2-\lambda)\frac{1}{2\lambda n} = \frac{1}{2}+\frac{1}{4\lambda},$$

which finishes the proof, since

$$\begin{aligned} D_{KL}(\nu||\mu) &\leq (\frac{1}{2n}\sum_{i=1}^n r_i - \frac{1}{2})\log(2\pi n) + \frac{1}{2\lambda} + 2\log(\frac{1}{2\lambda}) \\ &\leq (\frac{1}{8\lambda} - \frac{1}{4})\log n + \frac{1}{2\lambda} + \frac{1}{\lambda}\log(\frac{1}{2\lambda}). \end{aligned}$$

It remains to prove (12) and (11). First, we prove (12). Recall that $Q^A = D_\alpha A D_\beta$, where the diagonal entries of D_α and D_β are $\alpha_1, \dots, \alpha_n$, and $\beta_1, \dots, \beta_n$, respectively. Without loss of generality, assume there exists a matching between i of X to the vertex i of Y (otherwise we can reorder vertices). Then $\alpha_i \beta_i A_{ii} = Q^A_{ii} \le \frac{r_i}{n}$ by the definition of r_i. By this observation and the Van der Waerden inequality for Q^A (see e.g., [38]),

$$per(A) = per(D_\alpha^{-1}) \cdot per(Q^A) \cdot per(D_\beta^{-1}) \ge (\prod_i \frac{1}{\alpha_i \beta_i}) \frac{n!}{n^n} \ge e^{-n}\sqrt{2\pi n} \prod_i \frac{n}{r_i}.$$

Next, we prove (11). For an $n \times 1$ vector $\boldsymbol{c}$, let $\boldsymbol{c} \cdot Q^*$ be a matrix with each row of Q^* multiplied by a constant factor. Note that the sampling distribution of Algorithm 1 does not change, i.e., for any matching M, $p_{Q^*}(M) = p_{\boldsymbol{c} \cdot Q^*}(M)$. So, by choosing the right constants we can assume each row, Q^*_i, has $\lfloor \frac{n}{r_i} \rfloor$ entries equal to 1 and all other entries (except possibly one entry) is equal to 0. Note that this normalization does not change the left hand side of the inequality (11). In other words, following the proof of Proposition 3.7, we have that

$$\begin{aligned}
&\sum_{(i,j)\in[n]\times[n]} P_{(i,j)} \log(\boldsymbol{c}_i \cdot Q^*_{(i,j)}) - \sum_{i,k} P_{ik} \mathbb{E}_\pi \log(\sum_{j \ge_\pi k} \boldsymbol{c}_i \cdot Q^*_{ij}) \\
&\qquad = \mathbb{E}_{M\sim\nu,\pi\sim S_n} \Big(\log \prod_{i=1}^n \frac{\boldsymbol{c}_{\pi(i)} Q_{\pi(i),M(\pi(i))}}{\sum_{j=i}^n \boldsymbol{c}_{\pi(i)} Q_{\pi(i)M(\pi(j))}} \Big) \\
&\qquad = \mathbb{E}_{M\sim\nu,\pi\sim S_n} \Big(\log \prod_{i=1}^n \frac{Q_{\pi(i),M(\pi(i))}}{\sum_{j=i}^n Q_{\pi(i)M(\pi(j))}} \Big) \\
&\qquad = \sum_{e\in[n]\times[n]} P_e \log(Q^*_e) - \sum_{i,k} P_{ik} \mathbb{E}_\pi \log(\sum_{j \ge_\pi k} Q^*_{ij}).
\end{aligned}$$

So it is sufficient to show that

$$\begin{aligned}
&- \sum_{(i,j)\in[n]\times[n]} P_{(i,j)} \log(\boldsymbol{c}_i \cdot Q^*_{(i,j)}) + \sum_{i,k} P_{ik} \mathbb{E}_\pi \log(\sum_{j \ge_\pi k} \boldsymbol{c}_i \cdot Q^*_{ij}) \\
&\qquad \le \log \prod_{i=1}^n \Big((\lfloor \frac{n}{r_i} \rfloor + 1)!^{\frac{r_i}{n}} r_i^{\frac{1}{\lambda n}} \Big).
\end{aligned} \tag{13}$$

The rest is on the proof of last inequality.

By the construction of Q^* in Lemma 3.8, all entries (except at most one) in each row of $\boldsymbol{c} \cdot Q^*$ are equal to 1. The smallest non-zero entry smaller than one in row i of $\boldsymbol{c} \cdot Q^*$ is equal to $\frac{n}{r_i} - \lfloor \frac{n}{r_i} \rfloor$. So, if such an entry outside of $\{0, 1\}$ exists, i.e., if n is not divisible by r_i, then the smallest non-zero entry is at least $\frac{1}{r_i}$. Further, the matching marginal of each edge cannot be more than $\frac{1}{\lambda n}$, by Lemma 3.6. Therefore,

$$- \sum_{(i,j)\in[n]\times[n]} P_{(i,j)} \log(\boldsymbol{c}_i \cdot Q^*_{(i,j)}) \le \sum_{i=1}^n \frac{1}{\lambda n} \log(r_i).$$

Now, we need to bound the second term in the left hand side of (13). Let $\bar{Q}^*$ be a $0-1$ matrix that each of its entries are equal to 1 if and only if the same entry is non-zero in Q^*. Note that $\bar{Q}^*$ is equal to $\boldsymbol{c} \cdot Q^*$ except at most one entry per row. Then

$$\sum_{i,k} P_{ik} \mathbb{E}_\pi \log(\sum_{j \ge_\pi k} \boldsymbol{c}_i \cdot Q^*_{ij}) \le \sum_{i,k} P_{ik} \mathbb{E}_\pi \log(\sum_{j \ge_\pi k} \bar{Q}^*_{ij}).$$

The rest is an upper bound similar to Bregman-Minc's inequality [46, 10]. Note that $\sum_{j\geq_\pi k}\bar{Q}^*_{ij}$ is equal to the number of non-zero entries of row i that appear after k. There are $\lfloor\frac{n}{r_i}\rfloor+1$ non-zero entries and the probability that k appears before ℓ number of them in π is equal to $\frac{1}{\lfloor\frac{n}{r_i}\rfloor+1}$. Therefore,

$$\begin{aligned}\sum_{i,k}P_{ik}\mathbb{E}_\pi\log(\sum_{j\geq_\pi k}\bar{Q}^*_{ij}) &= \sum_{i,k}P_{ik}\frac{1}{\lfloor\frac{n}{r_i}\rfloor+1}\sum_{\ell=1}^{\lfloor\frac{n}{r_i}\rfloor+1}\log(\ell)\\ &= \sum_{i,k}P_{ik}\frac{1}{\lfloor\frac{n}{r_i}\rfloor+1}\log((\lfloor\frac{n}{r_i}\rfloor+1)!)\\ &= \sum_{i}\frac{1}{\lfloor\frac{n}{r_i}\rfloor+1}\log((\lfloor\frac{n}{r_i}\rfloor+1)!),\end{aligned}$$

where the last equality is because the sum of matching marginals over each row is equal to 1. As a result, we have proved (13). □

In Section 4, three different applications are given for SIS for sampling perfect matchings. Although some of our applications include situations where the underlying graph is not dense, we will see that SIS with doubly stochastic scaling still converges rapidly in our simulations. It remains open to give theoretical bounds, supporting the simulation results, on the convergence of SIS in graphs that are not dense.

4. Applications. Estimating the number of perfect matchings has applications in various settings. In this section, we present simulation results of three applications of SIS with doubly stochastic scaling.

We start by counting the number of Latin squares and rectangles. As will be discussed in Section 4.1, a $k\times n$ Latin rectangle corresponds to k disjoint perfect matchings in a complete bipartite graph of size n. Using SIS, we sample and estimate the number of Latin squares. Then, we use an SIS estimator as a benchmark to test three conjectures on the asymptotic number of Latin squares.

We continue by applying SIS to card guessing experiments in Section 4.2. A deck of n cards is shuffled and one has to guess the cards one by one. We will see that guessing the most likely card at each step (the greedy strategy) reduces to evaluating permanents. Therefore, we can apply SIS to find the number of correct card guesses using greedy for large decks of cards. Note that in the literature, the exact expectation of correct guesses using greedy was only known for small deck sizes (see e.g., [18, 20]).

Finally, Section 4.3 demonstrates the importance of using the doubly stochastic scaling of the adjacency matrix as the input of SIS. We compare SIS with and without doubly stochastic scaling to count the number of perfect matchings in bipartite graphs generated by stochastic block models. As we will see, using the doubly stochastic scaling of the adjacency matrix can make the standard deviation 4 times lower in some cases.

4.1. *Counting Latin Rectangles.* The first application of sequential importance sampling is for counting the number of Latin rectangles. An $n\times n$ Latin square is an $n\times n$ matrix with entries in $\{1,\dots,n\}$, such that each row and each column contains distinct integers. Let L_n be the number of $n\times n$ Latin squares. A $k\times n$ Latin rectangle is a k by n array with all rows containing $\{1,2,...,n\}$ and all columns distinct

The exact values of $L_{k,n}$ is only known for small k and n. Indeed, $L_{1,n}=n!$. Also, the number of ways to fill out the second row of a $2\times n$ Latin square is equal to the number

of derangements of $\{1,\dots,n\}$, leading to $L_{2,n} \sim \frac{(n!)^2}{e}$. A series of classical works give the asymptotics of $L_{k,n}$ when $k = o(n^{1/2})$ [26, 67, 66, 56]. Godsil and McKay [29] generalized these results to the case $k = o(n^{6/7})$, with the following asymptotics

$$L_{k,n} \sim \frac{(n!)^k([n]_k)^n}{e^{k/2}(1-\frac{k}{n})^{n/2}n^{kn}}.$$

There are Markov chains on the space of Latin squares with uniform stationary distributions [33, 51]. Alas, at this writing, there are no known bounds on the mixing time. Another method is to use divide and conquer to generate an exact uniform sample [14]. Here, we use sequential importance sampling to generate a Latin rectangle row by row. First, we describe the algorithm to sample a Latin square. Then we compare our estimator with known exact values and an earlier versions of importance sampling given by Kuznetsov [40]. Then we test three conjectured asymptotics on the number of Latin rectangles and squares. Finally, we test a conjecture by Cameron [11] on the number of odd permutation rows in a typical Latin square.

To describe the SIS algorithm, let $G(X,Y)$ be a bipartite graph, where X represents entries in a row and $Y = \{1,\dots,n\}$ represents the possible values for each entry. Then we sample a Latin rectangle row by row. Start with $G = K_{n,n}$ and repeat the following for k steps: Sample a perfect matching with sequential importance sampling and then remove its edges from G. This procedure is repeated until a Latin rectangle is obtained. In the experiment below, this is repeated $N = 10^7$ times resulting in Latin rectangles, each with a weight equal to the product of importance sampling weights for all rows. Define the estimator $L_{k,n}^{SIS}$ as the average of the N weights of $k \times n$ Latin rectangles. In the same way, we define the estimator L_n^{SIS} for the number of Latin squares.

First, we compare L_n^{SIS} with some exact values of L_n in Table 1. Note that the exact values of L_n are only known up to $n = 11$. See [57] for a comprehensive recent review on computing the value of $L_{k,n}$. To simplify the reported numbers, we divided L_n and L_n^{SIS} by $n!(n-1)!$, since there are $n!(n-1)!$ ways to generate the first row and the first column of a Latin square. As shown on the table, the relative error of SIS estimator is less than 0.1% for all $5 \le n \le 11$. Also, an early version of unscaled importance sampling with rejections has been given by Kuznetsov [40]. In Table 2, we compare our estimator with Kuznetsov's estimator.

n	Runs	$\frac{L_n^{SIS}}{n!(n-1)!}$	Confidence Interval	$\frac{L_n}{n!(n-1)!}$	% error	References
5	10^7	56.021	$(56.000, 56.041)$	56	0.0375	
6	10^7	9406.3	$(9400.9, 9411.7)$	9408	0.0181	
7	10^7	1.6945×10^7	$(1.6933 \times 10^7, 1.6958 \times 10^7)$	1.6942×10^7	0.0177	
8	10^7	5.3529×10^{11}	$(5.3475 \times 10^{11}, 5.3583 \times 10^{11})$	5.3528×10^{11}	0.0019	[47, 65]
9	10^7	3.7781×10^{17}	$(3.7729 \times 10^{17}, 3.7834 \times 10^{17})$	3.7759×10^{17}	0.0583	[47]
10	10^7	7.5876×10^{24}	$(7.5730 \times 10^{24}, 7.6024 \times 10^{24})$	7.5807×10^{24}	0.0910	[43]
11	10^7	5.3687×10^{33}	$(5.3539 \times 10^{33}, 5.3836 \times 10^{33})$	5.3639×10^{33}	0.0895	[44]

TABLE 1
Comparison of our SIS estimator (L_n^{SIS}) with the exact values of L_n for $5 \le n \le 11$.

Next we compare our method with three conjectures on the asymptotic value of $L_{k,n}$ and L_n. The first is a conjecture by Timashov [61], who used a formula by O'Neil [50] for the permanent of a random matrix with given row and column sums to guess the number of Latin

n	Runs	L_n^{SIS}	Kuznetsov's estimator	The exact value of L_n
5	10^7	1.613×10^5 $(1.613 \times 10^5, 1.614 \times 10^5)$	1.609×10^5 $(1.593 \times 10^5, 1.625 \times 10^5)$	161280
6	10^7	8.127×10^8 $(8.122 \times 10^8, 8.132 \times 10^8)$	8.135×10^8 $(8.054 \times 10^8, 8.217 \times 10^8)$	8.129×10^8
7	10^7	6.149×10^{13} $(6.145 \times 10^{13}, 6.153 \times 10^{13})$	6.149×10^{13} $(6.087 \times 10^{13}, 6.210 \times 10^{13})$	6.148×10^{13}
8	10^7	1.088×10^{20} $(1.087 \times 10^{20}, 1.089 \times 10^{20})$	1.095×10^{20} $(1.084 \times 10^{20}, 1.106 \times 10^{20})$	1.088×10^{20}
9	10^7	5.528×10^{27} $(5.520 \times 10^{27}, 5.536 \times 10^{27})$	5.531×10^{27} $(5.475 \times 10^{27}, 5.586 \times 10^{27})$	5.525×10^{27}
10	10^7	9.992×10^{36} $(9.972 \times 10^{36}, 10.011 \times 10^{36})$	9.991×10^{36} $(9.891 \times 10^{36}, 10.091 \times 10^{36})$	9.982×10^{36}
11	10^7	7.777×10^{47} $(7.755 \times 10^{47}, 7.798 \times 10^{47})$	7.777×10^{47} $(7.700 \times 10^{47}, 7.855 \times 10^{47})$	7.770×10^{47}
12	10^7	3.102×10^{60} $(3.091 \times 10^{60}, 3.114 \times 10^{60})$	3.083×10^{60} $(3.053 \times 10^{60}, 3.114 \times 10^{60})$	—
13	10^7	7.523×10^{74} $(7.480 \times 10^{74}, 7.566 \times 10^{74})$	7.427×10^{74} $(7.353 \times 10^{74}, 7.502 \times 10^{74})$	—
14	10^7	1.274×10^{91} $(1.263 \times 10^{91}, 1.285 \times 10^{91})$	1.261×10^{91} $(1.249 \times 10^{91}, 1.274 \times 10^{91})$	—
15	10^7	1.724×10^{109} $(1.702 \times 10^{109}, 1.747 \times 10^{109})$	1.728×10^{109} $(1.710 \times 10^{109}, 1.745 \times 10^{109})$	—
16	10^7	2.168×10^{129} $(2.113 \times 10^{129}, 2.224 \times 10^{129})$	2.211×10^{129} $(2.167 \times 10^{129}, 2.255 \times 10^{129})$	—
17	10^8	2.8045×10^{151} $(2.7734 \times 10^{151}, 2.8355 \times 10^{151})$	2.766×10^{151} $(2.711 \times 10^{151}, 2.821 \times 10^{151})$	—
18	10^8	4.2512×10^{175} $(4.1886 \times 10^{175}, 4.3138 \times 10^{175})$	4.163×10^{175} $(4.038 \times 10^{175}, 4.288 \times 10^{175})$	—
19	10^8	8.3851×10^{201} $(8.1913 \times 10^{201}, 8.5789 \times 10^{201})$	8.594×10^{201} $(8.250 \times 10^{201}, 8.937 \times 10^{201})$	—
20	10^8	2.4433×10^{230} $(2.3326 \times 10^{230}, 2.5541 \times 10^{230})$	2.263×10^{230} $(2.150 \times 10^{230}, 2.376 \times 10^{230})$	—

TABLE 2

Comparison of our SIS estimator with the estimator by Kuznetsov [40].

squares. Let $L_{k,n}^{Tim}$, L_n^{Tim} be the number of Latin rectangles and Latin squares, respectively, conjectured by Timashov,

$$L_{k,n}^{Tim} = \frac{(2\pi n/e)^{k/2}(1-k/n)^{n^2-nk+1/2}([n]_k)^{2n}}{n^{kn}}(1+o(1)),$$

$$L_n^{Tim} = \frac{(2\pi)^{3n/2+1}}{2} e^{-2n^2-n/2-1} n^{n^2+3n/2-1}(1+o(1)).$$

As a second conjecture, Leckey, Liebenau and Wormald [41] conjectured the following asymptotics

$$L_{k,n}^{LLW} = f(\frac{k}{n}) \frac{(n!)^k([n]_k)^{2n}}{e^{\frac{k}{2}}[n^2]_{kn}},$$

where $f(x)$ is a continuous and increasing function on $[0,1]$, with $f(0)=0$ and $f(1)=\frac{\sqrt{2\pi^3}}{e^{7/4}}$, in particular,

$$L_n^{LLW} = \frac{\sqrt{2\pi^3}}{e^{\frac{7}{4}}} \frac{(n!)^{3n}}{e^{\frac{n}{2}}(n^2)!} \approx L_n^{Tim}.$$

Actually Timashov [61] also allowed an additional constant term $C(k,n)$, and the asymptotic of L_n^{LLW} conjectures that $C(k,n)$ is of the form $f\big(k/n\big)$. As both constants are unspecified in general and the forms we have given above seem quite accurate, we will stick with them without the constant.

As the third conjecture, Eberhard, Manners and Mrazovic [24] used Maximum Entropy methods and Gibbs distributions to give the following conjecture on the number of Latin squares

$$L_n^{EMM} = \frac{(n!)^{3n}}{(n^2)!} e^{-\frac{n}{2}+\frac{5}{6}+O(\frac{1}{n})}.$$

Note the difference between the L_n^{EMM} with Timashov's conjecture for Latin squares, $\frac{L_n^{Tim}}{L_n^{EMM}} \sim \frac{\sqrt{2\pi^3}}{e^{25/12}} \approx .975$. In order to visualize the estimated number of Latin squares for large n and compare then with these conjectures, we divide the assymptotics and our estimator by the constant $c_n = \frac{(n!)^{3n}}{(n^2)!} e^{-\frac{n}{2}}$. Note that $\frac{L_n^{EMM}}{c_n} = e^{\frac{1}{2}} \approx 1.649$. As shown in Fig. 1, our estimators are in favor of Timashov's conjecture for Latin square.

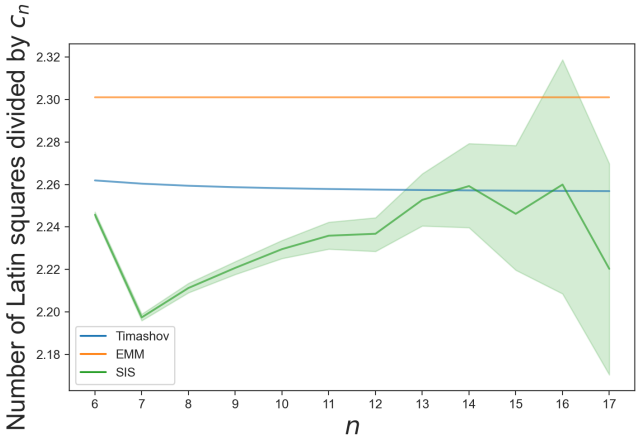

Fig 1: Comparison of SIS estimator $\frac{L_n^{SIS}}{c_n}$ with $\frac{L_n^{EMM}}{c_n}$ and $\frac{L_n^{Tim}}{c_n}$. The 95% confidence intervals are highlighted.

Next we fix $k=5$ and compare the conjectures on the number of $5\times n$ Latin rectangles. In order to simplify the plot, we divide each estimator by the factor $c_{k,n} = \frac{(n!)^k([n]_k)^{2n}}{[n^2]_{kn}}$. We know $\frac{L_{5,n}^{LLW}}{c_{5,n}} = e^{\frac{-k}{2}} \approx 0.082$, and also, $\frac{L_{k,n}^{Tim}}{L_{k,n}^{LLW}} = e^{-k/12n}$. So, the difference between conjectures decays with n. In Fig. 2, we compare our estimator L_n^{SIS} with both conjectures, and unlike the case of Latin squares, our results show a better accuracy for Leckey, Liebenau and Wormald's conjecture for Latin rectangles.

Finally, we test a conjecture by Cameron on random Latin squares. Call a row of a Latin square odd if its corresponding permutation is odd. The conjecture is as follows:

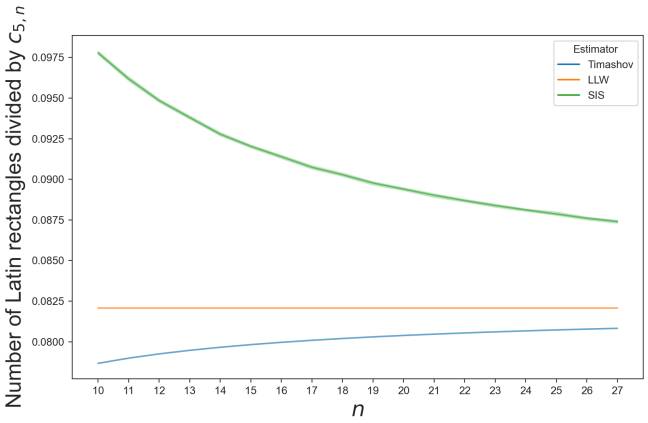

Fig 2: Comparison of $\frac{L_{5,n}^{SIS}}{c_{5,n}}$, $\frac{L_{5,n}^{Tim}}{c_{5,n}}$, and $\frac{L_{5,n}^{LLW}}{c_{5,n}}$. The 95% confidence intervals are too narrow to present, with $(0.0977, 0.0978)$ for $n = 10$ and $(0.0873, 0.0874)$ for $n = 27$.

CONJECTURE (Problem 10 in [11]). The number of odd rows of a random Latin square of order n is approximately binomial $Bin(n, \frac{1}{2})$ as $n \to \infty$.

As a special case of this conjecture, Häggkvist and Janssen [30] showed that the number of Latin squares with all even rows is exponentially small. We test the conjecture with sequential importance sampling. Note that each generated Latin square is associated with a weight which is equal to the inverse of the probability of generating that Latin square in importance sampling. Let $\hat{o}_n$ be the empirical distribution of the number of odd permutations in a Latin square of size n generated by SIS over 10^6 samples. The first moment Wasserstein distance of $\hat{o}_n$ with associated SIS weights from the distribution $Bin(n, \frac{1}{2})$ is presented for $n = 7$ to 15 in Table 3. While the distance between two distributions are small, we do not yet see a decay of the distances with our simulations for n up to 15.

n	7	8	9	10	11	12	13	14	15
$\widetilde{\mathcal{W}}$	0.0247	0.0054	0.0030	0.0043	0.0163	0.0140	0.0299	0.0191	0.0279

TABLE 3

The Wasserstein distance ($\widetilde{\mathcal{W}}$) of the number of odd rows in 10^6 weighted samples of Latin squares using SIS ($\hat{o}_n$) from $Bin(n, \frac{1}{2})$.

Of course, asymptotics and generating random Latin squares are only one aspect of the problem. The exact calculations indicate that the answers are divisible by surprisingly high powers of 2. From available data, it is hard to guess at the power or to understand why this should be (see e.g., [64] for related open problems).

4.2. *Card Guessing with Yes/No Feedback.* In this section, we apply the repeated estimation of the number of matchings in bipartite graphs to a card guessing experiment. The card guessing experiment starts with a well shuffled deck of N cards. A subject guesses the cards one at a time, sequentially, going through the deck. After each guess, the subject is told if their guess is correct or wrong.

Card guessing experiments with feedback occur in Fisher's tasting tea [28], in evaluation of randomized clinical trials [8, 25], in ESP experiments [15], and in optimal strategy in

casino card games such as blackjack or baccarat [27]. A review is in [17, 19]. The recent papers [18, 20] developed practical strategies which do perform close to optimal and gave bounds on the expected score under the optimal strategy.

Given a shuffled deck of card, how should the subject use the feedback to get a high score? Extensive numerical work in the literature suggests that greedy (guessing the most likely card given the feedback) is close to optimal. Next we will see that implementing the greedy strategy reduces to evaluating the permanent of a matrix. Then we apply SIS to estimate the expected number of correct guesses using the greedy strategy for any reasonable deck size. The exact value of greedy was only known for small decks in the literature, and the SIS estimator is close to the known exact values.

To see the relation of card guessing experiments and evaluating permanents let us fix some notation. Consider a deck of N cards, with n distinct values (say, $1, 2, \ldots, n$) with value i repeated m_i times, so $\sum_1^n m_i = N$. An example is a normal deck of cards $(N = 52)$ with 13 values each with multiplicity 4.

The deck is shuffled and a guessing subject makes sequential guesses. Consider the situation that there are a_i cards labeled i that are not guessed yet. Moreover, let b_i be the number of incorrect guesses where the subject chose card labeled i. Let $N(a, b)$ be the number of permutations of a deck of $N' = \sum_1^n a_i$ cards, where cards labeled 1 are not in first b_1 positions and cards labeled 2 are not in next b_2 positions and so on. Then $N(a, b) = Per(M_{ab})$ with M_{ab} a zero-one matrix with zero blocks of size $a_1 \times b_1, a_2 \times b_2, \ldots$ as shown in Figure 3.

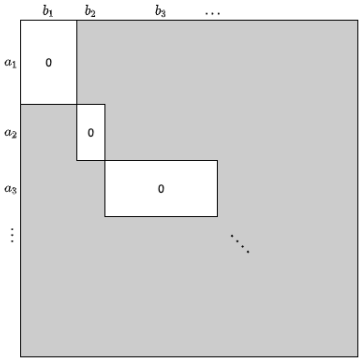

Fig 3: An example of structure of card guessing matrix (M_{ab}).

The number of consistent arrangements with card labeled i in the next position is equal $N(a^{*i}, b)$ where $a_i^{*i} = a_i - 1$ and $a_j^{*i} = a_j$ for all $j \neq i$. Thus, the chance that next guess is i equals

$$\frac{N(a^{*i}, b)}{N(a, b)}.$$

The greedy algorithm guesses i to maximize this ratio. Naively, n permanents must be evaluated at each stage. Some theory presents a simplification. In [13], the following theorem is proved.

THEOREM 4.1. *With Yes/No feedback, following an incorrect guess of i, the greedy strategy is to guess i for the next guess.*

Theorem 4.1 shows that the permanent must only be evaluated following a correct guess. Our algorithm for card guessing use the sequential importance sampling algorithm developed in the sections above to do this via Monte Carlo: fix B and generate random matchings in the appropriate bipartite graph B times, weighting each matching by its probability. Use these weighted samples to estimate the chance that the next card is i and choose the maximizing i^*.

Throughout, we simulate greedy using 'guess card i until told correct' to start. After a 'Yes' answer, sequential importance sampling runs B times as above to estimate the chance of the n possible values for the next card. The most probable is chosen and one keep guessing this value until 'Yes'. Then again begins to simulate, and repeats the procedure above. The plots below compare various values of B.

Example 1 ($m = 2$ **and varying** n)**:** We begin by discussing a long open case. A deck of size $2n$ of composition $1, 1, 2, 2, 3, 3, \ldots, n, n$. It was open until recently whether the expected number of correct guess is unbounded in n. In [18] it was shown to be bounded by 6 for all n. Figure 4 shows some numbers.

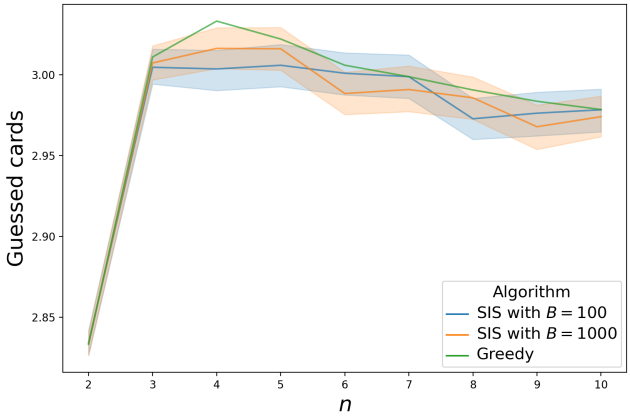

Fig 4: Comparison between different algorithms when $m = 2$. The 95% confidence intervals are highlighted.

REMARK 4.2. The figure shows estimates of the expected values with $B = 100$ and $B = 1000$. The shaded regions are pointwise 95% confidence intervals. Averages are based on 10000 repetitions. These figures suggest that the lower bound 2.91 from [20] is close to the truth.

The exact computation of general permanents is a well-known $\#P$-complete problem. However, on reflection, we have found that the permanents of matrices of the form of Figure 4 can be exactly computed; in linear time in the size of the matrix (see Appendix C). Having an exact algorithm allows us to check the numerical examples for card guessing. For small values of n the exact expectation of greedy expectations was computed by backward induction for $n = 2, 3, 4, 5$ in Table 4 and for larger n, we use our exact greedy algorithm to estimate the greedy strategy.

Example 2 ($m = n$): The classical ESP experiment had $m = n = 5$. The theorems in [18] and [20] worked for fixed m and large n or fixed n and large m. There are virtually

n	GREEDY	SIS with $B = 100$	SIS with $B = 1000$
2	2.8333	2.8343 $(2.8265, 2.8421)$	2.8338 $(2.8260, 2.8415)$
3	3.0111	3.0047 $(2.9937, 3.0158)$	3.0074 $(2.9962, 3.0186)$
4	3.0333	3.0037 $(2.9912, 3.0162)$	3.0164 $(3.0039, 3.0289)$
5	3.0433	3.0060 $(2.9930, 3.0190)$	3.0162 $(3.0031, 3.0292)$
10	2.9786 $(2.9773, 2.9799)$	2.9784 $(2.9649, 2.9918)$	2.9741 $(2.9606, 2.9876)$
20	2.9585 $(2.9571, 2.9598)$	2.9589 $(2.9454, 2.9724)$	2.9557 $(2.9423, 2.9691)$

TABLE 4
Greedy versus SIS estimator with 95% *confidence interval for* $m = 2$

no results for guesses when both m and n are large. Using SIS any reasonable deck sizes are now accessible. Fig. 5 shows results using sequential importance sampling for $m = n$ for $2 \leq m = n \leq 18$. The algorithm in Appendix C allows computing the number of correct guesses by the greedy algorithm for larger $m = n$. Note that the number of correct guesses by greedy depends on the order of the cards in the shuffled deck, and as a result we show confidence intervals both for greedy and SIS in Fig. 5.

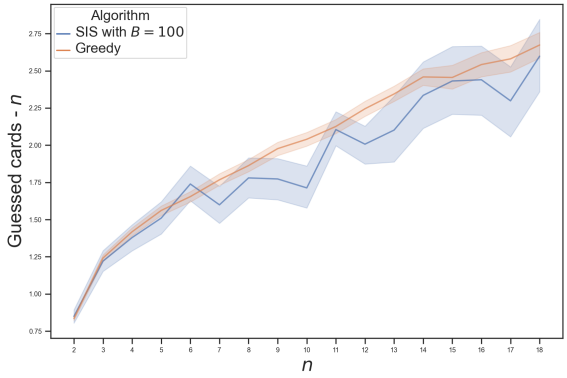

Fig 5: Comparison between different algorithms when $m = n$. The y-axis represents 'the number of guessed cards'$-n$. The 95% confidence intervals are highlighted.

REMARK 4.3. The greedy algorithm guesses at least m cards from this data, it was not clear if the excess over m even got as large as 2 (!). We now believe that for $m = n$, the experiment of the greedy strategy is $n + \sqrt{n} + o(\sqrt{n})$ (see Fig. 6).

4.3. *Stochastic Block Models.* The purpose of this section is to show the effect of doubly stochastic scaling on the convergence and variance of our estimator. The stochastic block model is a generative model for random graphs with community structures. It is widely used for statistical analysis of community structures and social networks.

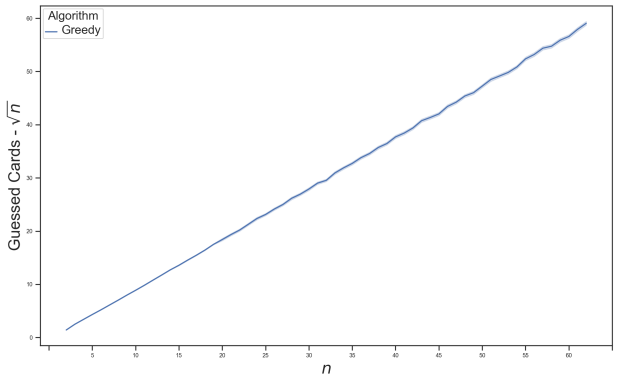

Fig 6: The plot of the 'guessed cards' $-\sqrt{n}$ when $m = n$. The 95% confidence intervals are too narrow to present, with $(2.8213, 2.8484)$ for $n = 2$ and $(66.5565, 67.2839)$ for $n = 62$.

Let n be the number of vertices and $C_1, C_2, \ldots, C_k$ be the clusters. Also, assume that P is a $k \times k$ matrix for edge probability between clusters. The probability that $u \in C_i$ is connected to $v \in C_j$ is equal to P_{ij}. We can similarly define stochastic block models for bipartite graphs. If we have k clusters in one part and r clusters in the other part, then the probability matrix will be a $k \times r$ matrix which P_{ij} shows the probability that a vertex from the cluster i in the first part is connected to a vertex from the cluster j of the second part.

We focus on the case that each part of the bipartite graph $G(X, Y)$ has only two clusters. Let X_1, X_2 be clusters in first part and Y_1, Y_2 be clusters in the other part. Hence, the matrix P will be an 2×2 matrix. Let us assume that $P_{11} = P_{12} = P_{21} = p$ and $P_{22} = q$. Intuitively, for large p and small q, most of the edges of a perfect matching would be selected from edges between X_1, and Y_2 or from edges between X_2, Y_1. In other words, If we select many edges between X_1 and Y_1, then there could be a near-perfect matching between X_2, Y_2 which has a low probability. Therefore, using the algorithm without doubly stochastic scaling of the adjacency matrix is not efficient.

In Fig. 7, we sampled one graph with the given values of p and q when $n = 20$. Then we used SIS (with/without doubly stochastic scaling) estimator to count the number of perfect matchings in the sampled graph. As shown in the figures, SIS with doubly stochastic scaling converges faster. Given that both estimators are unbiased, an important property for comparison is the standard deviation of the estimator. In Table 5, we compare the standard deviation of the SIS with doubly stochastic scaling and without doubly stochastic scaling. In this table each row correspond to one sampled graph from the stochastic block model. In some cases the standard deviation of the SIS without doubly stochastic is 10 times higher than the SIS with doubly stochastic scaling.

REMARK 4.4. Given $p, q \in [0, 1]$, it is possible to calculate the expected number of perfect matchings over all graphs drawn from doubly stochastic block model. However, the variance on the number of perfect matchings is large. Therefore, if we do not fix one sampled graph it is hard to differentiate the efficiency of one estimator over the other.

Acknowledgements. The authors thank Sourav Chatterjee, for helpful feedback on the early version of the manuscript and the proof of Lemma 2.2, and Jan Vondrák for discussions on concentration inequalities. Also, we would like to thank Nick Wormald and Fredrick Manners for bringing us up to speed about Latin squares.

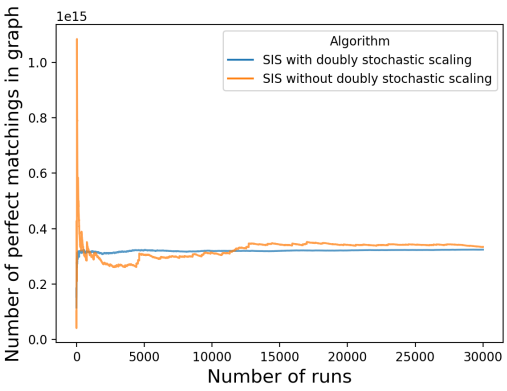

Fig 7.A: $p = 1, q = 0.1$.

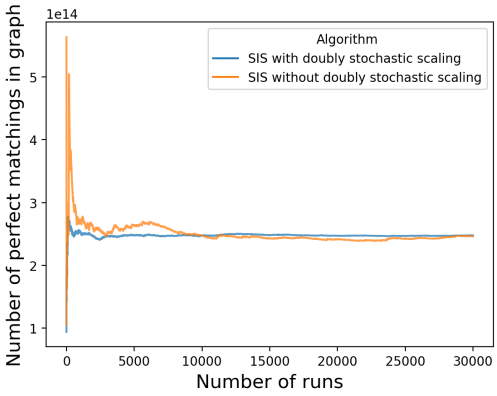

Fig 7.B: $p = 0.9, q = 0.2$.

Fig 7: Comparing the convergence of the SIS with doubly stochastic scaling and without doubly stochastic scaling for different q and p when $n = 20$. Here, the x-axis shows the number of runs of algorithm and the y-axis shows SIS estimator so far.

n	p	q	Runs	Standard deviation of SIS with doubly stochastic	Standard deviation of SIS without doubly stochastic	Mean of SIS with doubly stochastic
20	1	0.1	10^5	3.0699×10^{14}	2.2088×10^{15}	3.2606×10^{14}
20	1	0.2	10^5	1.9577×10^{15}	8.2753×10^{15}	2.3127×10^{15}
20	0.9	0.1	10^5	3.4717×10^{13}	1.9276×10^{14}	3.0495×10^{13}
20	0.9	0.2	10^5	2.5417×10^{14}	9.0992×10^{14}	2.4765×10^{14}
20	0.8	0.1	10^5	3.4055×10^{12}	1.9168×10^{13}	2.5553×10^{13}
20	0.8	0.2	10^5	3.0373×10^{13}	1.0374×10^{14}	2.6025×10^{13}

TABLE 5

Comparison of standard deviation of SIS with or without doubly stochastic scaling.

REFERENCES

[1] ALIMOHAMMADI, Y., DIACONIS, P., ROGHANI, M. and SABERI, A. (2021). Sequential Importance Sampling of Perfect Matchings– Github repository. https://github.com/mohammadroghani/SIS.

[2] ALLEN-ZHU, Z., LI, Y., DE OLIVEIRA, R. S. and WIGDERSON, A. (2017). Much Faster Algorithms for Matrix Scaling. *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)* 890-901.

[3] ANARI, N. and REZAEI, A. (2018). A Tight Analysis of Bethe Approximation for Permanent. *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)* 1434-1445.

[4] BAPAT, R. (1990). Permanents in probability and statistics. *Linear Algebra and its Applications* **127** 3–25.

[5] BARVINOK, A. (2017). *Combinatorics and Complexity of Partition Functions*, 1st ed. Springer Publishing Company, Incorporated.

[6] BAYATI, M., GAMARNIK, D., KATZ, D., NAIR, C. and TETALI, P. (2007). Simple deterministic approximation algorithms for counting matchings. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing* 122–127.

[7] BEICHL, I. and SULLIVAN, F. (1999). Approximating the Permanent via Importance Sampling with Application to the Dimer Covering Problem. *Journal of Computational Physics* **149** 128-147.

[8] BLACKWELL, D. and JR., J. L. H. (1957). Design for the Control of Selection Bias. *The Annals of Mathematical Statistics* **28** 449 – 460.

[9] BLITZSTEIN, J. and DIACONIS, P. (2010). A Sequential Importance Sampling Algorithm for Generating Random Graphs with Prescribed Degrees. *Internet Math.* **6** 489–522.

[10] BRÈGMAN, L. M. (1973). Some properties of nonnegative matrices and their permanents. In *Doklady Akademii Nauk* **211** 27–30. Russian Academy of Sciences.

[11] CAMERON, P. J. Random Latin Squares. *School of Mathematical Sciences Queen Mary and Westfield College London E1 4NS UK*.

[12] CHATTERJEE, S. and DIACONIS, P. (2018). The sample size required in importance sampling. *The Annals of Applied Probability* **28** 1099 – 1135.

[13] CHUNG, F. R. K., DIACONIS, P., GRAHAM, R. L. and MALLOWS, C. L. (1981). On the permanents of complements of the direct sum of identity matrices. *Advances in Applied Mathematics* **2** 121-137.

[14] DESALVO, S. (2017). Exact sampling algorithms for Latin squares and Sudoku matrices via probabilistic divide-and-conquer. *Algorithmica* **79** 742–762.

[15] DIACONIS, P. (1978). Statistical Problems in ESP Research. *Science* **201** 131–136.

[16] DIACONIS, P. (2018). Sequential importance sampling for estimating the number of perfect matchings in bipartite graphs : An ongoing conversation with Laci.

[17] DIACONIS, P. and GRAHAM, R. (1981). The Analysis of Sequential Experiments with Feedback to Subjects. *The Annals of Statistics* **9** 3–23.

[18] DIACONIS, P., GRAHAM, R., HE, X. and SPIRO, S. (2020). Card Guessing with Partial Feedback. *arXiv preprint arXiv:2010.05059*.

[19] DIACONIS, P., GRAHAM, R. and HOLMES, S. (1999). Statistical Problems Involving Permutations With Restricted Positions. State of the art in probability and statistics.

[20] DIACONIS, P., GRAHAM, R. and SPIRO, S. (2020). Guessing about Guessing: Practical Strategies for Card Guessing with Feedback. *arXiv preprint arXiv:2012.04019*.

[21] DIACONIS, P. and KOLESNIK, B. (2019). Randomized sequential importance sampling for estimating the number of perfect matchings in bipartite graphs. *arXiv preprint arXiv:1907.02333*.

[22] DUBHASHI, D. P. and RANJAN, D. (1996). Balls and bins: A study in negative dependence. *BRICS Report Series* **3**.

[23] DUFOSSÉ, F., KAYA, K., PANAGIOTAS, I. and UÇAR, B. (2018). Scaling matrices and counting the perfect matchings in graphs Research Report No. RR-9161, Inria Grenoble Rhône-Alpes.

[24] EBERHARD, S., MANNERS, F. and MRAZOVIC, R. (April, 2021). Oddly specific conjectures for counting latin squares. *Personal communication with Frederick Manners*.

[25] EFRON, B. (1971). Forcing a Sequential Experiment to be Balanced. *Biometrika* **58** 403–417.

[26] ERDÖS, P. and KAPLANSKY, I. (1946). The Asymptotic Number of Latin Rectangles. *American Journal of Mathematics* **68** 230–236.

[27] ETHIER, S. N. and LEVIN, D. A. (2005). On the fundamental theorem of card counting, with application to the game of trente et quarante. *Advances in Applied Probability* **37** 90–107.

[28] FISHER, R. A. (1935). *The Design of Experiments*. Oliver and Boyd, Edinburgh.

[29] GODSIL, C. D. and MCKAY, B. D. (1990). Asymptotic enumeration of Latin rectangles. *Journal of Combinatorial Theory, Series B* **48** 19-44.

[30] HÄGGKVIST, R. and JANSSEN, J. C. (1996). All-even latin squares. *Discrete Mathematics* **157** 199–206.

[31] HARRIS, D., SULLIVAN, F. and BEICHL, I. (2012). Fast Sequential Importance Sampling to Estimate the Graph Reliability Polynomial.

[32] HUBER, M. and LAW, J. (2008). Fast Approximation of the Permanent for Very Dense Problems. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '08* 681–689. Society for Industrial and Applied Mathematics, USA.

[33] JACOBSON, M. T. and MATTHEWS, P. (1996). Generating uniformly distributed random Latin squares. *Journal of Combinatorial Designs* **4** 405–437.

[34] JENSEN, A. and BEICHL, I. (2020). A Sequential Importance Sampling Algorithm for Counting Linear Extensions. *ACM J. Exp. Algorithmics* **25**.

[35] JERRUM, M., SINCLAIR, A. and VIGODA, E. (2004). A Polynomial-Time Approximation Algorithm for the Permanent of a Matrix with Nonnegative Entries. *J. ACM* **51** 671–697.

[36] JERRUM, M. R., VALIANT, L. G. and VAZIRANI, V. V. (1986). Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science* **43** 169 - 188.

[37] JOAG-DEV, K. and PROSCHAN, F. (1983). Negative association of random variables with applications. *The Annals of Statistics* 286–295.

[38] KNUTH, D. E. (1981). A Permanent Inequality. *The American Mathematical Monthly* **88** 731–740.

[39] KOU, S. C. and MCCULLAGH, P. (2009). Approximating the α-permanent. *Biometrika* **96** 635-644.

[40] KUZNETSOV, N. Y. (2009). Estimating the number of Latin rectangles by the fast simulation method. *Cybernetics and Systems Analysis* **45** 69–75.

[41] LECKEY, K., LIEBENAU, A. and WORMALD, N. (April, 2021). The asymptotic number of Latin rectangles. *Personal communication with Nick Wormald*.

[42] LOVASZ, L. L. and PLUMMER, M. D. (1986). *Matching theory*. Amsterdam ; New York : North-Holland : Elsevier Science Publishers B.V. ; New York, N.Y. : Sole distributors for the U.S.A. and Canada, Elsevier Science Pub. Co Includes indexes.

[43] MCKAY, B. D. and ROGOYSKI, E. (1995). Latin squares of order 10. *the electronic journal of combinatorics* **2** N3.
[44] MCKAY, B. D. and WANLESS, I. M. (2005). On the number of Latin squares. *Annals of combinatorics* **9** 335–344.
[45] MICALI, S. and VAZIRANI, V. V. (1980). An $O(|V||E|)$ Algoithm for Finding Maximum Matching in General Graphs. In *Proceedings of the 21st Annual Symposium on Foundations of Computer Science*. *SFCS '80* 17–27. IEEE Computer Society, USA.
[46] MINC, H. (1963). Upper bounds for permanents of $(0, 1)$-matrices. *Bulletin of the American Mathematical Society* **69** 789 – 791.
[47] MULLEN, G. and PURDY, D. (1993). Some data concerning the number of Latin rectangles. *J. Combin. Math. Combin. Comput* **13** 161–165.
[48] NEWMAN, J. E. and VARDI, M. Y. (2020). FPRAS Approximation of the Matrix Permanent in Practice.
[49] OWEN, A. and ZHOU, Y. (2000). Safe and Effective Importance Sampling. *Journal of the American Statistical Association* **95** 135–143.
[50] O'NEIL, P. E. (1970). Asymptotics in random $(0, 1)$-matrices. *Proceedings of the American Mathematical Society* **25** 290–296.
[51] PITTENGER, A. O. (1997). Mappings of latin squares. *Linear Algebra and its Applications* **261** 251-268.
[52] RASMUSSEN, L. E. (1994). Approximating the permanent: A simple approach. *Random Structures & Algorithms* **5** 349-361.
[53] SANKOWSKI, P. (2003). Alternative Algorithms for Counting All Matchings in Graphs. In *STACS 2003* (H. ALT and M. HABIB, eds.) 427–438. Springer Berlin Heidelberg, Berlin, Heidelberg.
[54] SIEGMUND, D. (1976). Importance Sampling in the Monte Carlo Study of Sequential Tests. *The Annals of Statistics* **4** 673 – 684.
[55] SINKHORN, R. (1964). A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices. *The Annals of Mathematical Statistics* **35** 876–879.
[56] STEIN, C. M. (1978). Asymptotic evaluation of the number of Latin rectangles. *Journal of Combinatorial Theory, Series A* **25** 38-49.
[57] STONES, D. (2010). The Many Formulae for the Number of Latin Rectangles. *Electr. J. Comb.* **17**.
[58] SULLIVAN, F. and BEICHL, I. (2014). Permanents, α-permanents and Sinkhorn balancing. *Computational Statistics* **29** 1793–1798.
[59] TARJAN, R. E. (1972). Depth-First Search and Linear Graph Algorithms. *SIAM J. Comput.* **1** 146-160.
[60] TASSA, T. (2012). Finding all maximally-matchable edges in a bipartite graph. *Theoretical Computer Science* **423** 50 - 58.
[61] TIMASHOV, A. (2002). On permanents of random doubly stochastic matrices and asymptotic estimates of the nom hers of Latin rectangles and Latin squares. *Discrete Mathematics and Applications* **12** 431 - 452.
[62] TSAO, A. (2020). Theoretical Analysis of Sequential Importance Sampling Algorithms for a Class of Perfect Matching Problems. *arXiv preprint arXiv:2001.02273*.
[63] VALIANT, L. G. (1979). The complexity of computing the permanent. *Theoretical Computer Science* **8** 189 - 201.
[64] WANLESS, I. (2021). Research topics on Latin squares. https://users.monash.edu.au/ iwanless/resprojhome.html.
[65] WELLS, M. B. (1967). The number of Latin squares of order eight. *Journal of Combinatorial Theory* **3** 98–99.
[66] YAMAMOTO, K. (1950). An Asymptotic Series for the Number of Three-Line Latin Rectangles. *Journal of the Mathematical Society of Japan* **1** 226 – 241.
[67] YAMAMOTO, K. (1952). On the asymptotic number of Latin rectangles. In *Japanese journal of mathematics: transactions and abstracts* **21** 113–119. The Mathematical Society of Japan.

APPENDIX A: AN IMPLEMENTATION OF ALGORITHM 1

Given a graph $G(X, Y)$ and a partial matching $\overline{M}$, recall that an edge $e \in E(G)$ is $\overline{M}$-extendable, if there exists a perfect matching that contains $\overline{M} \cup \{e\}$. We then simply say an edge is extendable if there exists a perfect matching containing it. We implement Algorithm 1 by finding extendable edges fast. The main idea presented in Section A.1 is to use Dulmage-Mendelsohn decomposition [42], to create a directed graph from G, so that an edge is extendable if and only if both of its endpoints are in the same strongly connected component.

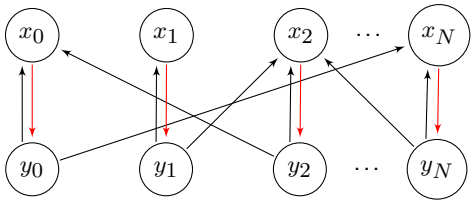

Fig 8: An example of constructing $D_G(X, Y)$, where the black edges are $E(G)$, and the red edges are in the matching M.

In addition, we need to keep track of extendable edges as the Algorithm 1 decides on adding an edge to the partial matching constructed so far. An algorithm that takes linear time to update the decomposition at each step will be shown in Section A.2.

A.1. Creating a Directed Acyclic graph and Maintaining Extendable Edges. To find all extendable edges, note that the union of any two perfect matchings of G creates a cycle decomposition on the nodes. Let M be a perfect matching. Construct a directed graph $D_G(X, Y)$ by directing all edges of $G(X, Y)$ from Y to X and adding a directed copy of M from X to Y (see Fig. 8). An edge is extendable if both of its endpoints are in the same strongly connected component.

Algorithm 2: Finding all Extendable Edges

Input: Bipartite graph $G(X, Y)$, a perfect matching M.
Construct $D_G(X, Y)$.
Find all strong connected components (SCC) of $D_G(X, Y)$.
for *any edge* $e \in E(G)$ **do**
    if *endpoints of* e *are in the same SCC* **then**
        e is extendable.
    else
        e is not extendable.
    end
end
Output: return extendable edges.

LEMMA A.1. *Given a bipartite graph $G(X, Y)$ and a perfect matching M, Algorithm 2 finds all the extendable edges in $O(n + m)$ time, where m is the number of edges, n is the number of vertices of G.*

PROOF. First, we prove the output of the algorithm is the set of all extendable edges. For that purpose, we show an edge $e = (u, v)$ is extendable if and only if both u and v are in the same strongly connected component of G. First, assume that e is an edge in some perfect matching M' (it can be equal to M). Edges of $M \cup M'$ create a directed cycle decomposition on $D_G(X, Y)$, since edges of cycles in $M \cup M'$ alternate between M and M', and M and M' are directed in opposite directions. So, in this case, both ends of e are in the same strongly connected component.

Now, assume $e = (u, v)$ and u, v appear in the same strong connected component. Then we show that e is extendable. Since u and v are in the same SCC, there exists a directed cycle $C = (u, v, a_1, a_2, \ldots, a_{2n}, u)$ in D_G. If e is a part of M we are done. So assume $e \notin M$.

Therefore, edges of C alternate between edges of M and edges not in M. By replacing edges of M that appear in the cycle C, with edges of C that are not in M, we still get a perfect matching. In other words, remove (v, a_1), $(a_3, a_4), \ldots, (a_{2n-1}, a_{2n}), (a_{2n}, u)$ from M, and add (u, v), $(a_2, a_3), \ldots, (a_{2n-2}, a_{2n-1})$ to construct M', then e is an edge of perfect matching M'.

To analyze the algorithm's running time, note that finding the strongly connected components of D_G can be done in $O(n + m)$ by [59]. □

A.2. Keeping Track of Extendable Edges . Now that we know how to find extendable edges, we can implement Step 5 of Algorithm 1. Given a graph G and one of its perfect matchings M', first construct $D_G(X, Y)$ from M' as in Section A.1. Then after each vertex is matched, we need to update $D_G(X, Y)$ so that each extendable edge of the current partial matching can be found by the SCC decomposition of $D_G(X, Y)$.

To do so, when an edge e is added to the partial matching, we consider two cases. First, if e is directed from X to Y, i.e., it is already in the perfect matching, we can only remove e from $D_G(X, Y)$ and the cycle decomposition of the rest of the graph indicate new extendable edges. In the second case, when e is directed from Y to X, find a directed cycle that it appears in (which exists since e is extendable). Note that if we reverse all edges in this cycle, we get a new graph $D_G(X, Y)$ such that e is directed from X to Y. Then as in the first case, we can remove edge e.

Algorithm 3: A Detailed Implementation of Algorithm 1

Input: a bipartite graph $G(X, Y)$, and a nonnegative matrix $Q_{n\times n}$.
Draw a random permutation π on X.
Let $M = \emptyset$, $p_{\pi,Q}(M) = 1$.
Run Algorithm 2, and let S be the SCC decomposition, and D_G be the digraph.
for *i from 1 to n* **do**
  Find the set of neighbors of $\pi(i)$ that are in the same SCC with it (call it $N_{\pi(i)}$).
  Find $Q_{\pi(i)}[N_{\pi(i)}]$, the restriction of row $\pi(i)$ of Q to indices in $N_{\pi(i)}$.
  Let i^* be the random index in $N_{\pi(i)}$ drawn with the probability proportional to $Q_{\pi(i)i^*}$.
  $M = M \cup (\pi(i), i^*)$.
  $p_{\pi,Q}(M) = p_{\pi,Q}(M) \times \frac{Q_{\pi(i),i^*}}{\sum_{j\in N_{\pi(i)}} Q_{\pi(i),j}}$.
  if *$(\pi(i), i^*)$ is directed from Y to X in D_G* **then**
    Find a directed cycle C which contains the edge $(\pi(i), i^*)$ in D_G.
    Reverse all edge of C.
  end
  Delete $\pi(i)$ and i^* from D_G.
  Find SCC decomposition of D_G.
end
Output: $M, p_{\pi,Q}(M)$.

REMARK A.2. To run Algorithm 2 in the beginning, we can find one perfect matching using the Micali-Vazirani algorithm (see e.g., [45]).

LEMMA A.3. *Suppose Q is a nonnegative matrix, and $Q_{ij} > 0$ for all $(i, j) \in E(G)$. Then Algorithm 3 always returns a perfect matching.*

PROOF. It is enough to show that for all i, $N(\pi(i))$ is the set of all extendable adjacent edges of $\pi(i)$. We prove by induction. At each step of the algorithm, all edges directed from X to Y form a perfect matching. Then the result follows by the correctness of Algorithm 2, proved in Lemma A.1.

The base case of induction is true by construction. At step i of the algorithm, when we match $\pi(i)$ to j there exists a perfect matching M' that $e = (\pi(i), j) \in M'$. Similar to the proof of Lemma A.1, we can construct matching M' so that its edges are the same with matching M, except edges that appear in cycle C. This is what happens in Algorithm 3: It removes the copy of edges in $M \cap C$ that are directed from X to Y, if $e \notin M$, and adds all edges of $M' \cap C$ from X to Y. So, at the end edges that are directed from X to Y are edges of a perfect matching that contains e, which proves the induction step. □

PROPOSITION A.4. *Let G be a bipartite graph with n vertices and m edges. Then the Algorithm 3 runs in $O(nm)$ time to sample a perfect matching.*

PROOF. First, we need to find one perfect matching to build a directed graph on top of it. Finding a perfect matching takes at most $O(n^{1/2}m)$ by using [45].

Each iteration of the algorithm requires finding a cycle. For that purpose, one can use a depth-first-search which needs $O(n+m)$ operations. Also, finding the strongly connected component decomposition can be done with $O(n+m)$ operations [59]. Therefore, Algorithm 3 can be done in $O(n(n+m))$ operations. It is worth noting that sampling from rows of Q can be done in linear time, which is bounded by the computation time of each step. □

Note that the Dulmage-Mendelsohn decomposition has been already used to find extendable edges when the partial matching is an empty-set (see [60]). However, in this section, we extended the idea to update the Dulmage-Mendelsohn decomposition and find new extendable edges after each step of the SIS algorithm without modifying the decomposition a lot.

APPENDIX B: AN EXAMPLE FOR THE NECESSITY OF THE DOUBLY STOCHASTIC SCALING

The main modification of our algorithm in comparison to the algorithm in [21] is using the doubly stochastic scaling of the adjacency matrix as the input of Algorithm 1. We show this scaling is necessary by giving an example which shows the uniform sampling of edges at each round needs an exponential number of samples while sampling according to the doubly stochastic scaling of the adjacency matrix only needs a linear number of samples.

PROPOSITION B.1. *Let $G_n(X,Y)$ be a bipartite graph with $X = \{x_0, x_1, \dots, x_n\}$ and $Y = \{y_0, y_1, \dots, y_n\}$. Assume in G_n, x_i is adjacent to y_i for all i, and x_0 and y_0 are adjacent to all vertices of the other side. Let μ be the uniform distribution over the set of perfect matchings and ν be the sampling distribution of Algorithm 1 with Q equal to an all ones matrix. If $\rho = d\nu/d\mu$ then there exist constants $\epsilon, c > 0$ such that $\mathbb{P}_\nu(\log \rho(X) > L + \epsilon n) > c$.*

Note that this result along with Theorem 2.1 shows that an exponential number of samples are needed in this class of graphs if we sample edges uniformly at random at each step of Algorithm 1. Before proving this result, we state the analogous result that shows

PROPOSITION B.2. *Let the graph G_n be as defined in Proposition B.1. Given a constant $\epsilon > 0$, there exists a constant $C > 0$ such that $Cn\epsilon^{-2}$ samples of Algorithm 1 is enough to give an $(1-\epsilon)$-approximation of the number of perfect matchings.*

PROOF OF PROPOSITION B.1. Define the following notations for perfect matchings in G,

$$M_0 = \{(x_i, y_i) : 1 \le i \le n\},$$

and for $1 \le i \le n$,

$$M_i = \{(x_0, y_i), (x_i, y_0)\} \cup \{(x_j, y_j) : j \notin \{0, i\}\}.$$

Consider M_0, and let $\pi(i)$ be the location of x_i in the permutation π. All x_i that come before x_0, have two edges to choose between, and for all nodes that come after x_0, there is only one edge left. For x_0 itself, there are $n - \pi(0) + 1$ edges available that any one of them can extend to a perfect matching. Therefore, $p_{\pi,Q}(M_0) = \frac{2^{-\pi(0)}}{n-\pi(0)+1}$, which implies

$$\mathbb{E}_\pi\big(\log_2 p_{\pi,Q}(M_0)\big) = -\sum_{i=0}^{n} \frac{1}{n+1}(i + \log_2(n+1-i)) = -\frac{n}{2} - \frac{\log_2((n+1)!)}{n+1}.$$

Now, for matching M_i with $1 \le i \le n$, consider two cases to find the sampling probability. Each vertex with $\pi(i) < \pi(0)$ has two choices for pairing, and otherwise it has only one choice. So, when $\pi(i) < \pi(0)$, $p_{\pi,Q}(M_i) = 2^{-\pi(i)}$. For the case $\pi(i) > \pi(0)$, all vertices before $\pi(0)$ have two choices, and $\pi(0)$ have $n - \pi(0) + 1$ choices. Vertices after that have only one extension to a perfect matching. Therefore, $p_{\pi,Q}(M_i) = \frac{2^{-\pi(0)}}{n-\pi(0)+1}$, which again implies,

$$\begin{aligned}
\mathbb{E}_\pi\big(\log_2 p_{\pi,Q}(M_i)\big) &= -\sum_{j=0}^{n} \Big(\frac{n+1-j}{n(n+1)}(j + \log_2(n+1-j)) + \sum_{k=0}^{j} \frac{k}{n(n+1)}\Big) \\
&= -\sum_{j=0}^{n} \Big(\frac{n+1-j}{n(n+1)}(j + \log_2(n+1-j)) + \frac{j(j+1)}{2n(n+1)}\Big) \\
&= -\sum_{j=0}^{n} \frac{j}{n} + \frac{\log_2(n+1-j)}{n} + \frac{(n+1-j)\log_2(n+1-j)}{n(n+1)}
\end{aligned}$$

The last equality shows that there exists constant $C_1, C_2 \ge 0$ such that

$$-\frac{n}{3} + C_1 \log_2 n \le \mathbb{E}_\pi\big(\log_2 p_{\pi,Q}(M_i)\big) \le -\frac{n}{3} + C_2 \log_2 n.$$

By taking expectation over the uniform distribution, μ, on all perfect matchings $M_0, M_1, \ldots, M_n$, we see that there exists constants $C_1', C_2' \ge 0$ such that

$$\frac{n}{3} + C_1' \log_2 n \le \mathbb{E}_{\mu,\pi}\big(\log_2 p_{\pi,Q}(M)\big) \le \frac{n}{3} + C_2' \log_2 n,$$

where $1/2 \ge C \ge 1/3$. Recall that $\rho(M) = \frac{1}{n p_{\pi,Q}(M)}$. Then for small enough ϵ ,

$$\mathbb{P}_\nu(\log_2 \rho(M) \ge \frac{n}{3} + \epsilon n) \ge \frac{1}{2} - \epsilon.$$

the latter inequality is true, because for all M_is, if $\pi(0) \ge n/2 + \epsilon n$ then there exists $\epsilon > 0$ such that $-\log_2(p_{\pi,Q}(M_i)) \ge n/2$. □

PROOF OF PROPOSITION B.2. The doubly stochastic scaling of the adjacency matrix, Q_A, has $1/n$ on its first row and column and $1 - \frac{1}{n}$ on all other diagonal entries and zero everywhere else.

Define M_i's as in the proof of Proposition B.1. If $\pi(0) = i$ with probability $(1 - \frac{1}{n})^i \frac{1}{n-i}$ the algorithm generates matching M_0. In this case, if we let $q = \frac{1}{n}$, then

$$\log p_{\pi,Q_A}(M_0) = i \log(1-q) + \log(q).$$

For other matchings M_i, where $i \geq 1$, with probability $(1-q)^{\min(\pi(i),\pi(0))} q/(1-qi)$ we get the matching M_i with

$$\log p_{\pi,Q_A}(M_i) = (\min(\pi(i), \pi(0)) - 1) \log(1-q) + \log(q).$$

Therefore,

$$\max_{M_i,\pi} \{-\log p_{\pi,Q_A}(M_i)\} \leq n \log(\frac{n}{n-1}) + \log(n) \leq 2 + \log n.$$

Then if we let $L = \mathbb{E}_\mu \rho(\log n)$ as in Theorem 2.1,

$$\mathbb{P}_\nu \big(\log p_{\pi,Q_A}(M_i) \geq L + \log n \big) = 0,$$

which proves the statement by Theorem 2.1. □

APPENDIX C: COMPUTING PERMANENT OF ZERO-BLOCKED MATRICES

Given sequences $a = (a_1, \ldots, a_r)$ and $b = (b_1, \ldots, b_r)$, let M_{ab} be a zero-one matrix, such that zeros forms diagonal blocks of sizes $a_1 \times b_1, a_2 \times b_2, \ldots, a_r \times b_r$ (see Figure 3). We show that it is possible to compute $per(M_{ab})$ in a linear time. As a result, one can compute the greedy strategy for card guessing, as discussed in Section 4.2.

LEMMA C.1. *Let M_{ab} be defined as above. Suppose M_{ab} is an $n \times n$ matrix. Then $per(M_{ab})$ can be computed in $O(n^2)$ time.*

PROOF. To describe the algorithm for computing $per(M_{ab})$, we need the following notations. Let $f(i,j)$ be the number of the ways that we can select j zeros in the first i blocks such that no two zeros are in the same row or same column. Since there are r blocks in total, $f(r,j)$ shows the number of the ways that we can select j zeros in M_{ab} such that no two zeros are in the same row or the same column. Then we can formulate the permanent of M_{ab} by the inclusion-exclusion principle,

$$per(M_{ab}) = \sum_{j=0}^{n} (-1)^j (n-j)! f(r,j).$$

Therefore, if we can compute $f(i,j)$ for all $0 \leq i \leq r$ and $0 \leq j \leq n$ in $O(n^2)$ time, we can compute the permanent of the matrix M_{ab} in $O(n^2)$ time.

We know that the size of the i^{th} block is $a_i \times b_i$. The number of ways to choose k zeros in this block in such a way that no two zeros are in the same row or column is $\binom{a_i}{k} \frac{b_i!}{(b_i-k)!}$. Let $q_i = \min(a_i, b_i)$. Then k can take any value in $\{0, 1, \ldots, q_i\}$. Therefore,

$$f(i,j) = \sum_{k=0}^{q_i} f(i-1, j-k) \binom{a_i}{k} \frac{b_i!}{(b_i-k)!}. \tag{14}$$

Now, we claim that we can find the values of $f(i,j)$ for all $0 \leq i \leq r, 0 \leq j \leq n$ in $O(n^2)$ time. First, note that we can compute all the values of $\binom{i}{j}$ for all $0 \leq j \leq i \leq n$ using Pascal's rule in $O(n^2)$ time. We run a procedure to find the values of $f(i,j)$ using recursive induction. First, initialize all values of $f(i,j)$ to zero except for $f(0,0) = 1$. Then iterate over $i = \{1, 2, \ldots, r\}$ and $j = \{1, 2, \ldots, n\}$ in an increasing order and update the value of

an entry, based on the previously calculated values. To compute $f(i,j)$, by (14) we need $O(\min(a_i, b_i))$ operations. Hence, for a fixed i and all $0 \leq j \leq n$, computing $f(i,j)$ takes $O(n \cdot \min(a_i, b_i))$ operations. As a result, the total number of operations for computing all values of f are

$$\sum_{i=0}^{r} O(n \cdot \min(a_i, b_i)) = O(n^2),$$

which proves the claim. Therefore, computing the permanent of M_{ab} can be done in $O(n^2)$ time. □