Algorithmic barriers to representing conditional independence

Nathanael L. Ackerman*, Jeremy Avigad†, Cameron E. Freer‡, Daniel M. Roy§ and Jason M. Rute¶
*Harvard University, Cambridge, MA, USA
†Carnegie Mellon University, Pittsburgh, PA, USA
‡Massachusetts Institute of Technology, Cambridge, MA, USA
§University of Toronto and Vector Institute, Toronto, ON, Canada
¶Penn State University, State College, PA, USA

***Abstract*—We define a representation of conditional independence in terms of products of probability kernels, and ask when such representations are computable. We pursue this question in the context of exchangeable sequences and arrays of random variables, which arise in statistical contexts. Exchangeable sequences are conditionally i.i.d. by de Finetti's theorem. Known results about the computability of de Finetti's theorem imply that these conditional independences are computable. The conditional independences underlying exchangeable arrays are characterized by the Aldous–Hoover theorem. In the special case of adjacency matrices of undirected graphs, i.e., symmetric binary arrays, this representation theorem expresses the conditional independences in terms of graphons. We prove that there exist exchangeable random graphs that can be computably sampled but whose corresponding graphons are not computable as functions or even as L^1 equivalence classes. We also give results on the approximability of graphons in certain special cases.**

1. Introduction

Independent random variables are fundamental building blocks for random structures exhibiting complex dependencies. Besides independence, the most fundamental (in)dependence is *conditional* independence. Conditional independence relationships among random variables have a critical bearing on the complexity of many computational tasks, including sampling and probabilistic inference. In particular, conditional independences among random variables imply that their joint distribution *factorizes* into a product of simpler distributions on subsets of the variables. It is this factorization that is often exploited for efficient computation. Given the importance of conditional independences and the factorizations they induce, it is worth building a formal computational model of conditional independence itself.

To that end, consider n random variables, $X_1,\dots,X_n$. Recall that $X_1,\dots,X_n$ are **independent** if the (joint probability) distribution of $(X_1,\dots,X_n)$ is equal to the product of the (marginal) distributions on X_i, for $i=1,\dots,n$, i.e., the joint distribution satisfies the factorization $\mathbf{P}[(X_1,\dots,X_m)] = \otimes_{j=1}^m \mathbf{P}[X_j]$. Introducing a random variable Y, we may now refer to the conditional distribution of X_j *given* Y. Write $T,S_1,\dots,S_n$ for the spaces in which the random variables $Y,X_1,\dots,X_n$ take values. Recall that the conditional distribution $\mathbf{P}[X_j|Y]$ is in fact a random variable of the form $g(Y)$ for some function g on T taking values in the space of *probability measures* on S_j. The function g is a **probability kernel** from T to S_j, and any probability kernel h such that $g(Y)=h(Y)$ a.s. is said to be a **version** of g. We say that $X_1,\dots,X_n$ are **conditionally independent given** Y if the factorization $\mathbf{P}[X_1,\dots,X_m|Y] = \otimes_{j=1}^n \mathbf{P}[X_j|Y]$ holds. For infinite collections $X_1,X_2,\dots$, conditional independence requires that the factorization hold only for all finite subcollections.

In light of these definitions, it is natural to express the computability of conditional independence in terms of the computability of the resulting factorizations of the joint distributions, which in turn is naturally expressed in terms of the computability of the family of probability kernels.

There is already a rich literature studying notions of computability for distributions and probability kernels on spaces endowed with notions of computability. Using these notions, we can give several natural representations for joint distributions and their factorizations. To begin, distributions can be naturally represented as functions taking representations of open sets to their probability assignment. Equivalently, distributions can be given **sampling representations** where a function $f\colon [0,1]\to S$ represents the distribution of the S-valued random variable $f(U)$, when U is a uniformly distributed element in $[0,1]$. In order to determine a concrete representation, one must fix a representation for the function f. Randomized algorithms that halt almost surely correspond to computable elements in the space of functions that are continuous on a (Lebesgue) measure one subset of $[0,1]$. Weaker representations can be obtained by representing equivalence classes of functions up to null sets. In particular, when S is a bounded subset of the reals, it is natural to represent f as a computable point in the space of L^1-integrable functions.

These representations for distributions lead to natural representation for probability kernels. For example, probability kernels from S to T can be represented as maps from S to distributions on T. Alternatively, such probability kernels can also be given **sampling representations** $f\colon [0,1]\times S\to T$ such that the distribution of $f(U,s)$ is the same as the image of the probability kernel at s. Just as above, there is flexibility

in how to represent f.

In this work, we initiate the study of conditional independence from this computational perspective. We focus on *exchangeable random structures*, a setting where there are precise characterizations of the conditional independences that hold in general.

A sequence of random variables is exchangeable when its distribution does not depend on the ordering of its elements. A well-known theorem of de Finetti shows that infinite exchangeable sequences of random variables are conditionally independent and identically distributed (i.i.d.), meaning that, by introducing a new random variable and conditioning on its value, any apparent dependence between the random variables in the exchangeable sequence is removed and all the variables have the same distribution. In terms of factorizations, exchangeability of an infinite sequence $X_1, X_2, \ldots$ is equivalent to the existence of some random variable Y and probability kernel κ such that $\mathbf{P}[X_j|Y] = \kappa(Y)$ a.s. for all $j \geq 1$. Note that the probability kernels underlying the factorization are all versions of one another.

Exchangeable sequences are models for homogeneous data sets and serve as building blocks for statistical models with more interesting dependency structures [1]. Conditional independence and exchangeability are also central to probabilistic programming [2]. Infinite exchangeable sequences arise naturally in functional probabilistic programming languages. Indeed, any finite sequence of evaluations of a closure is a prefix of some infinite exchangeable sequence. The sequence is even manifestly conditionally i.i.d.: conditioning on the closure itself, every evaluation is independent and identically distributed.

The more interesting phenomenon is the existence of (potentially stateful) probabilistic code with the property that repeated evaluations produce an exchangeable sequence yet no existing variable renders the sequence conditionally i.i.d. Exchangeability, nevertheless, licenses a programmer or compiler to commute and even prune these repeated evaluations (see, e.g., [3]). These types of transformations are central to several probabilistic programming systems, including Church [4] and Venture [5].

As described in [6], a fundamental question for probabilistic programming is whether or not support for exchangeability is in some sense necessary on the grounds of efficiency or even computability. By de Finetti's theorem, an infinite exchangeable sequence of random variables $X_1, X_2, \ldots$ admits a representation $X_j = f(\theta, \xi_j)$ a.s., for all $j \geq 1$, where $f\colon [0,1] \times [0,1] \to \mathbb{R}$ is a measurable function and $\theta, \xi_1, \xi_2, \ldots$ are i.i.d. random variables, uniformly distributed in $[0,1]$. Clearly, there are many such functions f. On the other hand, as outlined above, any way of representing f yields a way of representing the conditional independence underlying the sequence $X_1, X_2, \ldots$, and so it is natural to ask when some such f is computable. Indeed, it suffices to establish the computability of the distribution of the random probability measure $\nu = \mathbf{P}[f(\theta, \xi_1) \in \cdot|\theta]$. This question was studied in the setting of exchangeable sequences of real random variables in [7], which established the computability of the distribution of ν, and showed that it was even uniformly computable from the distribution of the exchangeable sequence, yielding an effectivization of de Finetti's theorem that acts like a program transformation.

Here we study a generalization of de Finetti's theorem to two dimensions, and in particular, the binary symmetric case of the Aldous–Hoover theorem for two-dimensional jointly exchangeable arrays. Our focus will be on the computability of various representations of the distribution of these two-dimensional arrays, with a special emphasis on the representation of graphons, which can alternatively be viewed as either sampling representations or probability kernel representations, and thus as representations of the key conditional independences underlying an exchangeable array. (The function $f(\theta, \cdot)$ is a one-dimensional analogue of a graphon. For more details on graphons and exchangeable arrays, see [8] and [9], respectively.) The situation here is more complicated, and depends on the choice of metric. Several of the standard metrics are computably equivalent to each other, but we find that one natural way of expressing the relevant measurable object — corresponding to the so-called edit distance — is not computable from the distribution of the exchangeable data itself, unlike in the one-dimensional case. This mapping is possible using the halting problem $0'$ as an oracle, and we provide an example showing that this dependence is necessary. On the other hand, for a natural subclass, the so-called random-free case, one can recover the graphon in the edit distance metric from the distribution of the array.

In the case of computable distributions on binary symmetric exchangeable arrays, all these results fall short of establishing that some graphon is computable on a measure one set of points. In fact, one cannot obtain such representations in general. We show that, in the case of computable distributions on arrays, there need not be any graphon generating the array that is almost-everywhere continuous, revealing a topological impediment to the computability of certain sampling representations.

It follows that the two-dimensional setting is fundamentally different from the one-dimensional setting. There need not be a computable representation of an exchangeable array that exposes the conditional independence inherent in the exchangeable array, even if the exchangeable array itself has some computable representation. In some special cases, we can compute the graphon outside a set of arbitrarily small measure, even though we cannot compute the graphon outside a null set. This constitutes a weaker representation of the conditional independence, which could be useful for certain approximation schemes. Our results suggest that probabilistic programming languages may need to have special support for probabilistic symmetries, such as exchangeability in the case of arrays.

We work within the standard Turing-machine-based "bit-model" [10] for computability and complexity over the reals and other metric spaces, using essentially the computable analysis formulation provided by the Type-Two Theory of Effectivity [11], [12]. (For details, see Section 4.) Uniform versions of our results could be compactly described using Weihrauch degrees [13]; however, we have chosen to give

more elementary statements in order to reach a wider audience.

2. Background and summary of main results

An (infinite) array $(X_{i,j})_{i,j\geq 1}$ of (real-valued) random variables is (jointly) **exchangeable** when, for all $n \geq 1$ and all permutations $\sigma\colon \{1,\dots,n\} \to \{1,\dots,n\}$,

$$(X_{i,j})_{1\leq i,j\leq n} \stackrel{d}{=} (X_{\sigma(i),\sigma(j)})_{1\leq i,j\leq n}.$$

A result due to Aldous [14] and Hoover [15] characterizes the distributions of all such arrays.

Theorem 2.1. *An array $(X_{i,j})_{i,j\geq 1}$ of random variables is exchangeable if and only if there exists a measurable function $f\colon [0,1]^4 \to \mathbb{R}$ such that, for all $n \geq 1$,*

$$(X_{i,j})_{1\leq i,j\leq n} \stackrel{d}{=} \big(f(U_0,U_i,U_j,V_{\{i,j\}})\big)_{1\leq i,j\leq n} \qquad (\star)$$

where $U_0, U_i, \dots, V_{\{i,j\}}$, for $i,j \geq 1$, are independent and identically distributed random variables, uniformly distributed on $[0,1]$.

We call any such function f an **Aldous–Hoover map** for the array. The Aldous–Hoover map has a close connection to the representation of the conditional independences implied by $(\star)$: the array of random variables is conditionally independent given $U_0, U_i, V_{\{i,j\}}$, for $i,j \geq 1$; and, given $U_0, U_{i'}, U_{j'}, V_{\{i',j'\}}$, the entry $X_{i',j'}$ is conditionally independent of the remainder of the array and $U_i, V_{\{i,j\}}$, for $i,j \geq 1$, where $i \neq i'$ and $j \neq j'$. Note that the U and V variables are i.i.d., and so the key to the representation of the conditional independences here is the probability kernel κ satisfying $\mathbf{P}[X_{i',j'}|U_0,U_{i'},U_{j'}] = \kappa(U_0,U_{i'},U_{j'})$ a.s. for all $i',j' \geq 1$. Therefore, any Aldous–Hoover map f is a sampling representation of κ and thus the core component of the sampling representation of the array's conditional independences.

A very basic question is the following: What is the relationship between the complexity of the distribution of the array and the function f?

One significant complication is the fact that such a function f is not unique. Indeed, for any triple of measure-preserving maps $\phi, \phi', \phi''\colon [0,1] \to [0,1]$, the function $g\colon [0,1]^4 \to \mathbb{R}$ given by $g(a,b,c,d) = f\big(\phi(a),\phi'(b),\phi'(c),\phi''(d)\big)$ is also an Aldous–Hoover map for the array. It would seem natural to define the complexity of the conditional independence in terms of the class of Aldous–Hoover maps for the array. As we will see, this adds some considerable difficulties and requires more intricate arguments.

Note that any two Aldous–Hoover maps that differ on a measure-zero set induce the same distribution on the off-diagonal array entries. To simplify this presentation, we restrict attention to the case of exchangeable arrays with zero diagonal.

We now introduce precise notions of computability and samplability from the bit-model. These will cohere with the notion of a computable pseudometric space that we describe in Section 4. Recall that a partial function $F :\subseteq \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$ is **computable** when there is an oracle Turing machine such that, for every point x in the domain of F, if the oracle tape is set to x, the machine eventually outputs every finite prefix of $F(x)$ in finite time. Formally, a **representation** of a space S is a partial surjection $\rho_S :\subseteq \{0,1\}^{\mathbb{N}} \to S$. Representations then give rise to notions of computability on S: A point x in the domain of ρ_S is a name for its image $\rho_S(x)$ in S. A point $s \in S$ is **computable** when it has a computable name. A partial multi-valued function $f :\subseteq S \rightrightarrows T$ is **computable** relative to representations ρ_S and ρ_T, respectively, when, for some computable partial function $F :\subseteq \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$, we have $\rho_T(F(x)) \in f(\rho_X(x))$ for all x in the domain of $f \circ \rho_S$. Informally, representations provide a specification of points in terms of approximations of increasing accuracy; computable functions are ones for which we can compute the output to ever greater accuracy as we obtain more accuracy on the input, potentially given the promise that the point lies in a distinguished subset.

We say the distribution of an S-valued array X is **samplable** when there exist functions $f_n\colon [0,1] \to S^{n\times n}$, for $n \geq 1$, that are (uniformly) computable (as partial functions) on some set of (Lebesgue) measure one (relative to a canonical representation of $[0,1]$ and some representation of S, and then its products canonically) such that, for every $n \geq 1$, for any uniformly distributed element U in $[0,1]$, the value $f_n(U)$ has the same distribution as the initial $n \times n$ subarray of X. Informally, the functions f_n instruct us how to generate a sequence of approximations that converge with probability one to a random variable with the same distribution as X.

In the remainder of this section, we make reference to computability on various represented spaces. Standard spaces like the real numbers have canonical representations, which in turn induce representations of the space of distributions on standard spaces. (For details, see Sections 3 and 4.) Another space that has a canonical representation is the space of equivalence classes of L^1-integrable functions from a measured space into the reals.

We can now refine our original question. If the distribution of some exchangeable array of real random variables with zero diagonal is samplable, what does this imply about the complexity of the Aldous–Hoover maps? Namely, given a samplable exchangeable array, is some Aldous–Hoover map f for the array computable? In particular, given a computable such f and $n \geq 1$, we could generate the principal $n \times n$ subarray (up to distribution) by the following **canonical sampling process**:

1) Sample $U_0, U_i, V_{\{i,j\}}$ for $1 \leq i < j \leq n$, independently from the uniform distribution on $[0,1]$.
2) Compute $X_{i,j} = f(U_0,U_i,U_j,V_{\{i,j\}})$, where f is implemented, by hypothesis, by some computable subroutine.

The syntactic structure of this canonical sampling process exposes the conditional independences, in a way that is amenable to program analysis. In fact, we will show that there may not be such a computable f, and so we will not always be able to computably generate the array in this way.

An important special case of exchangeable arrays is the class of ergodic ones; ergodicity of an exchangeable array is equivalent to one of its Aldous–Hoover maps not making use of the first input. Every symmetric binary exchangeable array with zero diagonal is the adjacency matrix of some random undirected graphs with vertex set $\mathbb{N}$ whose distribution is invariant to every permutation of the vertices. For the ergodic such arrays, Aldous–Hoover maps can be simplified to structures known as graphons. While this section's summary of the paper's main results uses the language of Aldous–Hoover maps, the remainder of the paper is formulated in terms of graphons and invariant measures on the space of undirected graphs, a setting in which some of the rigorous statements are cleaner to state.

A graphon is a symmetric measurable function $W\colon [0,1]^2 \to [0,1]$. Every graphon W induces an ergodic symmetric binary exchangeable array X with zero diagonal via the $\{0,1\}$-valued Aldous–Hoover map f satisfying $f(w,x,y,z) = 1$ if and only if $x \neq y$ and $z \leq W(x,y)$. In the other direction, any f that satisfies $W(x,y) = \int f(w,x,y,z)\,dz$ and $f(w,x,x,u) = 0$ for almost every $w,x,y,u \in [0,1]$ is an Aldous–Hoover map for X. It is straightforward to show that every Aldous–Hoover map f for X is computable as a point in L^1 if and only if the graphon W is computable with respect to the d_1 pseudometric that we define in Section 3. The key fact linking the corollaries of this section with the theorems proved in the rest of the paper is Proposition 4.5 (b), which implies that W is computable with respect to the $\delta_\square$ pseudometric if and only if the distribution of X is computable.

We now highlight the four main results of our paper.

The following key result shows that there are (zero-diagonal) exchangeable arrays that can be generated by programs but not according to the canonical sampling process, on account of no program being able to implement any of its Aldous–Hoover maps.

Corollary 2.2 (of Theorem 5.3)**.** *There exists a samplable exchangeable array with zero diagonal such that none of its Aldous–Hoover maps are computable on a measure one set.*

This result presents a clear algorithmic barrier to representing conditional independence. Unlike many results pertaining to noncomputability, this result does *not* arise from a routine coding argument involving a noncomputable set such as the halting set. The obstruction here is, in fact, topological: Any Aldous–Hoover map that is computable on a measure one subset of $[0,1]^4$ (which therefore yields a samplable array via the canonical sampling process) must be continuous on a measure one subset of $[0,1]^4$. We demonstrate that there are samplable exchangeable arrays for which every Aldous–Hoover map is discontinuous on a set of positive measure. Given our particular example of such an array, the verification that it has this property is straightforward, if technical. However, the existence of such an example is rather surprising.

Corollary 2.2 has an interesting implication for functional probabilistic programming languages, discussed in more detail in [6]. The result implies that, in order to exploit the latent conditional independence of an exchangeable array, compilers cannot rely upon that structure being apparent in the syntax of the program. In other words, an array of random variables sampled according to the "program" given by the canonical sampling process is manifestly a collection of conditionally independent random variables. However, by Corollary 2.2, not every Aldous–Hoover map is computable on a measure one set, and so there may be no such program. In order for the compiler to take advantage of the exchangeable structure, the compiler must either recognize that the array is exchangeable by some program analysis or the language must provide the user with the means to specify these symmetries.

Given Corollary 2.2, it is natural to consider weaker representations. Because every Aldous–Hoover map is bounded, it is also integrable. Results from algorithmic randomness imply that, as a point in the space of L^1-integrable functions, an Aldous–Hoover map f for an exchangeable array X is computable only if, for every $k \in \mathbb{N}$, the map f is computable on a set of measure $1 - 2^{-k}$; if the distribution of X is computable, then this condition is also sufficient (see, e.g., [16]).

This weaker notion of computability allows us to sidestep the obstruction underlying Corollary 2.2. However, a new obstruction arises, this time tied to computability rather than continuity. Using a coding argument, we establish the following lower bound.

Corollary 2.3 (of Theorem 7.1)**.** *There exists a samplable exchangeable array with zero diagonal, one of whose Aldous–Hoover maps f is continuous on a measure one set, such that every name for an Aldous–Hoover map as an L^1 function computes the halting problem.*

We are also able to prove a matching upper bound, making use of the weak regularity lemma that provides bounds on how well a small graph can approximate a large graph (in the sense of approximately preserving the distribution of its random subgraphs).

Corollary 2.4 (of Theorem 6.2)**.** *Every ergodic symmetric binary exchangeable array with zero diagonal that is samplable has some Aldous–Hoover map that is computable as a point in L^1 using the halting problem as an oracle.*

While this does provide a theoretical upper bound on the complexity of the Aldous–Hoover map, it is a noncomputable one, and hence unsatisfying in practice.

In fact, we can provide positive results (with no oracle) for an important subclass. In particular, an Aldous–Hoover map can be computed for those ergodic symmetric binary exchangeable arrays known as *simple* in the exchangeability literature [17], and which correspond to *random-free* graphons [18, §10]. These arrays are those that have an Aldous–Hoover map that depends on neither its first nor last variable.

Corollary 2.5 (of Theorem 6.4)**.** *Every ergodic symmetric binary exchangeable array with zero diagonal that is samplable and simple has some Aldous–Hoover map that is*

computable as a point in L^1.

While restricting attention to simple arrays may seem limiting, in fact their distributions can approximate those of arbitrary ergodic exchangeable arrays in a sense stronger than weak convergence [17, Theorem 2]. Therefore, Corollary 2.5 is a positive result for all exchangeable arrays arising from graphons, provided we are willing to admit some approximation error.

3. Graphons and invariant measures on graphs

In this section we provide the basic definitions of and results about the two main objects of interest in this paper, graphons and invariant measures on graphs. Let λ denote Lebesgue measure on $\mathbb{R}$ (though for notational convenience, we will often use λ to also refer to Lebesgue measure on $\mathbb{R}^2$, etc.). For $n \in \mathbb{N}$, write $[n] := \{0, \dots, n-1\}$. All logarithms, written log, will be in base 2.

3.1. Graphons

We will formulate most of our results in terms of graphons, both for concreteness and simplicity of notation. In this subsection we summarize the standard notions that we will need. For more details on graphons, their basic properties, and notation, we refer the reader to [8, Chapter 7], from which most of the definitions in this subsection are borrowed. In many cases, analogous notions and results were developed earlier in terms of exchangeable arrays; for details on this connection and the history, see [19] and [20].

Definition 3.1. A **graphon** is a symmetric measurable function $W\colon [0,1]^2 \to [0,1]$. Let $\mathcal{W}_0$ denote the set of all graphons.

Three classes of graphons play a special role in this paper.

Definition 3.2. A graphon G is **random-free** if it is $\{0,1\}$-valued a.e., in other words, when $\lambda\left(G^{-1}(\{0,1\})\right) = 1$.

Definition 3.3. Let $\mathcal{T}$ be a topology on $[0,1]$. A graphon G is **almost-everywhere (a.e.) continuous** with respect to $\mathcal{T}$ if there is a set $X \subseteq [0,1]^2$ such that $\lambda(X) = 1$ and $G|_X$ is continuous with respect to $\mathcal{T}|_X$.

Definition 3.4. A graphon G is a **step function** if there is a (finite) measurable partition P of $[0,1]$ such that for every pair of parts $p, q \in P$, the graphon G is constant on $p \times q$. Write $\mathcal{S}$ to denote the class of step functions whose underlying partition divides $[0,1]$ into some finite number of equally-sized intervals, and whose range is contained in the rationals. Write $\mathcal{S}_0$ to denote the subset of its random-free graphons. Each of $\mathcal{S}$ and $\mathcal{S}_0$ admits a straightforward computable enumeration.

One can associate to each finite graph (on $[n]$ for some $n \geq 1$) a random-free step function graphon, as we now describe, such that the space of graphons is the completion of the finite graphs (embedded this way) under an appropriate pseudometric on finite graphs.

Definition 3.5. Let G be a graph with vertex set $[n]$, for some $n \geq 1$. Define the step function graphon $W_G\colon [0,1]^2 \to \{0,1\}$ to be such that $W_G(x,y) = 1$ if and only if there is an edge between $\lfloor nx \rfloor$ and $\lfloor ny \rfloor$ in G. In other words, W_G is equal to 1 on $[i/n, (i+1)/n) \times [j/n, (j+1)/n)$ if there is an edge from i to j, and 0 otherwise. Observe that $W_G \in \mathcal{S}_0$.

We will use three key pseudometrics on the space of graphons. We begin by describing the cut norm $\|\cdot\|_\square$, which will allow us to define $d_\square$ and then $\delta_\square$, the latter of which is closely related to subsampling. The third pseudometric is d_1, more closely related to edit distance and the L^1 norm.

Definition 3.6. The **cut norm** of a symmetric measurable function $F\colon [0,1]^2 \to [-1,1]$ is defined by

$$\|F\|_\square := \sup_{S,T \subseteq [0,1]} \left| \int_{S \times T} F(x,y)\, dx\, dy \right|$$

where S and T range over measurable sets. For graphons U and W, define $d_\square(U,W) := \|U - W\|_\square$.

It is straightforward to verify that $\|\cdot\|_\square$ is a norm on $\mathcal{W}_0$, and that $d_\square$ is a pseudometric on $\mathcal{W}_0$.

As we will see, this cut norm is too coarse for many of our purposes.

Definition 3.7. The L^1 **norm** of a symmetric measurable function $F\colon [0,1]^2 \to [-1,1]$ is defined by

$$\|F\|_1 := \int |F(x,y)|\, dx\, dy.$$

For graphons U and W, define $d_1(U,W) := \|U - W\|_1$.

It is a standard fact that the L^1 norm is a norm on $\mathcal{W}_0$, and that d_1 is a pseudometric on $\mathcal{W}_0$.

While the cut norm induces a coarser topology than the L^1 norm does, they do agree on the notion of norm zero. The following easy lemma follows from the fact that if $\|W\|_\square = 0$ then $W = 0$ a.e.

Lemma 3.8. *If W is a graphon, then $\|W\|_1 = 0$ if and only if $\|W\|_\square = 0$.*

This lemma implies that the pseudometrics d_1 and $d_\square$ can be thought of as metrics on the same quotient space, namely $\mathcal{W}_0 / \{(G,H) : d_1(G,H) = 0\}$, even though the metrics they induce on this space are very different.

As we will see in §3.2, there is a standard way to associate to each graphon an invariant measure on countable graphs and given two graphons we would like to have a condition equivalent to the corresponding invariant measures being the same. However, it is easy to see, by applying a non-trivial measure-preserving transformation to any non-constant random-free graphon, that there are graphons which give rise to the same distribution but which are very far in either d_1 or $d_\square$. Hence we will need a notion of distance yielding an even coarser topology, which we now define.

Definition 3.9. Let W be a graphon and let $\varphi\colon ([0,1],\lambda) \to ([0,1],\lambda)$ be a measure-preserving map. Define W^φ to be the graphon satisfying

$$W^\varphi(x,y) = W\big(\varphi(x),\varphi(y)\big)$$

for all $x,y \in [0,1]$.

Definition 3.10. For graphons U and W, define

$$\delta_\square(U,W) := \inf_\varphi d_\square(U,W^\varphi),$$

where the infimum is taken over all measure-preserving maps of $([0,1],\lambda)$ to itself.

The following standard result will be important when we consider the relative computability of the representations of graphons induced by various metrics.

Lemma 3.11. *The set $\mathcal{S}$ is dense in $(\mathcal{W}_0,d_1)$, and its subset $\mathcal{S}_0$ is dense in $(\mathcal{W}_0,d_\square)$ and $(\mathcal{W}_0,\delta_\square)$.*

Proof. The density of step functions in d_1 is a standard measure-theoretic fact. The density of $\mathcal{S}_0$ in $\delta_\square$ follows from [8, Theorem 11.52]. This implies that

$$\big\{W^\varphi : W \in \mathcal{S}_0 \text{ and } \varphi \text{ is a measure-preserving map}\big\}$$

is dense in $d_\square$. But for every $W \in \mathcal{S}_0$, measure-preserving map φ, and $\varepsilon > 0$, there is an element $V \in \mathcal{S}_0$ such that

$$d_\square(W^\varphi, V) < \varepsilon.$$

Hence $\mathcal{S}_0$ is also dense in $d_\square$. □

We will later need the following definition.

Definition 3.12. A graphon W is **twin-free** if for each pair of distinct points $x,y \in [0,1]$, the functions $z \mapsto W(x,z)$ and $z \mapsto W(y,z)$ disagree on a set of positive Lebesgue measure.

3.2. Invariant measures on graphs

Invariant measures on graphs with underlying set $\mathbb{N}$ are the main object of study in this paper. In the probability theory literature, one often studies exchangeable arrays rather than their distributions, but here we focus on their distribution as we will be interested in the measures rather than the random variables, and so that we can avoid certain technicalities and notational difficulties.

Definition 3.13. Let $\mathcal{G} \subseteq \{0,1\}^{\mathbb{N}^2}$ denote the space of adjacency matrices of symmetric irreflexive graphs with underlying set $\mathbb{N}$. A probability measure μ on the space $\mathcal{G}$ is called an **invariant measure on graphs** if $\mu(A) = \mu(\sigma^{-1}(A))$ for all Borel $A \subseteq \mathcal{G}$ and all permutations $\sigma\colon \mathbb{N} \to \mathbb{N}$.

We will use the term *invariant measure* to refer to invariant measures on graphs.

An important subclass of the invariant measures are those that are extreme.

Definition 3.14. An invariant measure μ is **extreme** if there do not exist distinct invariant measures ν and π such that $\mu = \alpha\nu + (1-\alpha)\pi$ for some $\alpha \in (0,1)$.

In our setting, the extreme measures coincide with the ergodic ones (with respect to permutations of $\mathbb{N}$).

Graphons naturally give rise to extreme invariant measures on graphs, via the distribution of the countably infinite random graph obtained by sampling from the graphon, as we now describe.

Definition 3.15. Let W be a graphon and let S be a countable set. Let $\langle\zeta_i\rangle_{i\in S}$ be an i.i.d. collection of uniform $[0,1]$-valued random variables. Consider the random graph $\widehat{\mathcal{G}}(S,W)$ with vertex set S where for all distinct $i,j \in S$, there is an edge between i and j independently, with probability $W(\zeta_i,\zeta_j)$. For $n \geq 1$, we write $\widehat{\mathcal{G}}(n,W)$ to refer to $\widehat{\mathcal{G}}([n],W)$. When H is a finite graph, we write $\widehat{\mathcal{G}}(S,H)$ to refer to $\widehat{\mathcal{G}}(S,W_H)$. Finally, let $\mathcal{G}(S,W)$ denote the distribution of $\widehat{\mathcal{G}}(S,W)$.

For $0 < p < 1$, if W is the constant graphon $W \equiv p$, then $\mathcal{G}(\mathbb{N},W)$ is the distribution of an Erdős–Rényi random graph.

The following result is standard.

Proposition 3.16 ([8, Theorem 11.52])**.** *If W is a graphon, then $\mathcal{G}(\mathbb{N},W)$ is an extreme invariant measure on graphs.*

Conversely, every extreme invariant measure arises from a graphon.

Proposition 3.17 ([8, Theorem 11.52])**.** *If μ is an extreme invariant measure on graphs then there is some graphon W such that $\mathcal{G}(\mathbb{N},W)$ and μ are the same distribution.*

It is then natural to ask when two graphons give rise to the same invariant measure.

Proposition 3.18 ([8, Theorem 13.10])**.** *For graphons U and W, the following statements are equivalent:*

1) *$\mathcal{G}(\mathbb{N},U)$ and $\mathcal{G}(\mathbb{N},W)$ are the same distribution;*
2) *$\delta_\square(U,W) = 0$; and*
3) *there are measure-preserving maps $\varphi,\psi\colon [0,1] \to [0,1]$ such that $U^\varphi = W^\psi$ a.e.*

When any of these equivalent statements holds, we say that U and W are **weakly isomorphic**.

We now describe a natural metric on the space of invariant measures.

Definition 3.19. Let $\mathcal{E}$ be the collection of extreme invariant measures, let $\mu \in \mathcal{E}$, and let F be a finite graph on $[n]$. Define $t_{\mathrm{ind}}(F,\mu) := \mu(\{G \in \mathcal{G} : G|_{[n]} = F\})$. Fix an enumeration $\langle F_i\rangle_{i\in\mathbb{N}}$ of finite graphs with underlying set $[n]$ for some $n \geq 1$. For $\mu,\nu \in \mathcal{E}$, define

$$d_w(\mu,\nu) := \sum_{i\in\mathbb{N}} 2^{-i}\big|t_{\mathrm{ind}}(F_i,\mu) - t_{\mathrm{ind}}(F_i,\nu)\big|.$$

The following is standard.

Lemma 3.20. *The space $(\mathcal{E},d_w)$ of extreme invariant measures is a compact Polish space with the topology of weak convergence. Further,*

$$\mathcal{S}_0^* := \big\{\mathcal{G}(\mathbb{N},G) : G \text{ is a finite graph}\big\}$$

is a dense subset.

Note that $\mathcal{S}_0^*$ also admits a straightforward computable enumeration.

The previous lemma tells us that we can approximate an extreme invariant measure arbitrarily well by measures which come from sampling graphons induced by finite graphs. A natural question is whether it is possible to take an invariant measure and find a (possibly random) sequence of finite graphs whose corresponding graphons almost surely converge to the invariant measure we started with. This is possible, as the following result states.

Lemma 3.21 ([8, Corollary 11.15])**.** *Suppose U is a graphon. Then $\langle \mathcal{G}(\mathbb{N}, \widehat{\mathcal{G}}(n,U))\rangle_{n\geq 1}$ is a random sequence of extreme invariant measures that almost surely converges in $(\mathcal{E}, d_w)$ to $\mathcal{G}(\mathbb{N},U)$.*

4. Notions of computability for graphons and invariant measures on graphs

There is considerable flexibility in how the space of graphons can be represented, and each representation gives rise to a corresponding notion of computability for graphons. Several representations arise naturally in our study of graphons, and they give rise to distinct notions of computability on the same underlying space. In this section, we describe several representations for the space of graphons and the space of invariant measures on graphs, and present some of the basic relationships between them.

4.1. Computable pseudometric spaces

A focus of this paper is represented spaces that reveal the metric structure, and so we will work with a class of represented spaces known as computable pseudometric spaces, a relaxation of the notion of a computable metric space in computable analysis (see, e.g., [12]).

Definition 4.1. A **computable** (complete) pseudometric space consists of a triple $(M, d, \langle s_i\rangle_{i\in\mathbb{N}})$ such that

- (M,d) is a complete pseudometric space,
- $\langle s_i\rangle_{i\in\mathbb{N}}$ is dense in (M,d), and
- the sequence $\langle d(s_i,s_j)\rangle_{i<j\in\mathbb{N}}$ is a computable sequence of real numbers.

Definition 4.2. Suppose $(M, d, \langle s_i\rangle_{i\in\mathbb{N}})$ is a computable pseudometric space. A **rapidly converging Cauchy sequence** is a sequence $\langle s_{k_j}\rangle_{j\in\mathbb{N}}$ for which $d(s_{k_j}, s_{k_\ell}) \leq 2^{-j}$ for $j < \ell \in \mathbb{N}$.

A rapidly converging Cauchy sequence is called a **d-name** for the limiting value $\lim_{n\to\infty} s_{k_n}$. We say that $\langle s_{k_j}\rangle_{j\in\mathbb{N}}$ is **computable** in d if the sequence of natural numbers $\langle k_j\rangle_{j\in\mathbb{N}}$ is computable, and that an element $s \in M$ is **computable** if it has some d-name that is computable. (These notions relativized to an oracle are defined in the obvious way.)

The computational strength needed to produce a d-name provides a measure of the complexity of the corresponding element of the represented space. Roughly, a name for an element of the pseudometric space is a sequence of approximations that converges with rate $n \mapsto 2^{-n}$. Note that the choice of this rate is somewhat arbitrary, since given a sequence that converges with some other computable rate, one can computably "thin out" the sequence so that it converges at the rate we have chosen.

The pseudometric spaces we have considered so far can be straightforwardly made into computable pseudometric spaces using the computable enumerations of dense subsets we have identified.

Lemma 4.3. *The following are computable pseudometric spaces: $(\mathcal{W}_0, d_1, \mathcal{S})$; $(\mathcal{W}_0, d_\square, \mathcal{S}_0)$; $(\mathcal{W}_0, \delta_\square, \mathcal{S}_0)$; and $(\mathcal{E}, d_w, \mathcal{S}_0^*)$.*

The quotient (by identifying points at distance zero) of every computable pseudometric space (M,d) can be regarded as a represented space, whose names are (encodings of) d-names, and the notions of computable points and functions agree. In this paper we are interested in the relative computability of names for graphons and invariant measures considered as elements in these various computable pseudometric spaces.

4.2. Computable relationships between representations

In this section we want to consider the computable relationship between various representations for graphons and exchangeable arrays. In order to do this we need a notion of a *computable function* between two pseudometric spaces.

Definition 4.4. Suppose (M,d,S) and (N,f,T) are computable pseudometric spaces. We say a map $g\colon M \to N$ is a **computable function**, or is simply **computable**, if there is a computer program e such that whenever $K := \langle k_j\rangle_{j\in\mathbb{N}}$ is an index sequence for a d-name of an element a then $\{e\}^K$ outputs an index sequence for an f-name of $g(a)$.

Suppose $h\colon N \to M$ is a computable map. We say that a computable map $g\colon M \to N$ is a **computable equivalence witnessed by** h if $d\big(x, h(g(x))\big) = 0$ for all $x \in M$ and $f\big(y, g(h(y))\big) = 0$ for all $y \in N$. In this case we say that the spaces are **computably equivalent**.

Let $k\colon M \to N$ be a surjective function. A computable map $h\colon N \to M$ is a **computable section** for k if $f\big(y, k(h(y))\big) = 0$ for all $y \in N$.

In other words, a function is computable if there is an algorithm that takes a name in one space and computably transforms it into a name in the other. A computable equivalence provides a uniform method for transforming a name in one space to a name in the other and vice-versa. Note that a computable equivalence induces a bijection between the corresponding metric spaces obtained by taking the quotient by distance 0 on each side.

We will consider computable sections in the case where the underlying sets M and N are the same and k is the identity function. In this case, a computable section takes an f-name for a computable element of N and returns a d-name for a (possibly different) computable element of M such that

(N, f, T) cannot "distinguish" the points, in the sense that they have f-distance 0.

Consider the following notions of computability for an invariant measure μ:

(1) There is a computable d_w-name for μ.
(2) There is a graphon W with a computable $\delta_\square$-name such that $\mathcal{G}(\mathbb{N}, W) = \mu$.
(3) There is a graphon W with a computable $d_\square$-name such that $\mathcal{G}(\mathbb{N}, W) = \mu$.
(4) There is a graphon W with a computable d_1-name such that $\mathcal{G}(\mathbb{N}, W) = \mu$.

The next result establishes relationships between these four notions, which yield the implications in Corollary 4.6. In fact, as we will later see, these implications are all that are possible.

Proposition 4.5. *The following functions between pseudometric spaces are computable:*

(a) *The map* $\alpha\colon (\mathcal{E}, d_w) \to (\mathcal{W}_0, \delta_\square)$ *that takes* $\mathcal{G}(\mathbb{N}, W)$ *to* W *(up to weak isomorphism).*
(b) *The map* $\beta\colon (\mathcal{W}_0, \delta_\square) \to (\mathcal{E}, d_w)$ *that takes* W *to* $\mathcal{G}(\mathbb{N}, W)$.
(c) *The identity map* $\mathrm{id}\colon (\mathcal{W}_0, d_\square) \to (\mathcal{W}_0, \delta_\square)$.
(d) *The identity map* $\mathrm{id}\colon (\mathcal{W}_0, d_1) \to (\mathcal{W}_0, d_\square)$.

Furthermore, α *is a computable equivalence witnessed by* β*, and vice-versa, and there is a computable section of the identity map* (c)*. Finally,* (a)*,* (b)*, and* (d) *induce bijections on the corresponding metric spaces.*

Proof. The d_1-distance between two graphons is at least their $d_\square$-distance, and so any d_1-name is a $d_\square$-name. Similarly, any $d_\square$-name is a $\delta_\square$-name. Hence (c) and (d) are computable.

Now to show (b) we want to show that if $\langle W_{G_n}\rangle_{n\in\mathbb{N}} \subseteq \mathcal{S}$ is a rapidly convergent Cauchy sequence in $\mathcal{S}$ then $\langle \mathcal{G}(\mathbb{N}, W_{G_n})\rangle_{n\in\mathbb{N}}$ is a rapidly convergent sequence in $\mathcal{E}$. Given a finite graph F and graphon W, write $t_{\mathrm{ind}}(F, W)$ for $t_{\mathrm{ind}}\big(F, \mathcal{G}(\mathbb{N}, W)\big)$. By the Counting Lemma ([8, Exercise 10.30]), for any graphons U and V and finite graph F with k vertices, we have

$$\big|t_{\mathrm{ind}}(F,U) - t_{\mathrm{ind}}(F,V)\big| \le 4\tbinom{k}{2} d_\square(U,V).$$

For any measure-preserving map $\varphi\colon [0,1] \to [0,1]$, we have $t_{\mathrm{ind}}(F,V) = t_{\mathrm{ind}}(F, V^\varphi)$. Therefore

$$\big|t_{\mathrm{ind}}(F,U) - t_{\mathrm{ind}}(F,V)\big| \le 4\tbinom{k}{2} \delta_\square(U,V)$$

holds by Definition 3.10. Hence given a $\delta_\square$-name for U we can computably thin out its entries to form a d_w-name.

To show (a) we need the following Chernoff bound to obtain a $\delta_\square$-name given the distribution of an exchangeable array. As shown in [8, Lemma 10.16], for each $k \in \mathbb{N}$, with probability at least $1 - e^{-k/(2\log k)}$, we have

$$\delta_\square\big(U, W_{\widehat{\mathcal{G}}(k,U)}\big) \le \frac{22}{\sqrt{\log k}}.$$

As $\mathcal{G}(k,U)$ depends only on the distribution of the induced exchangeable array we can find an element G_k which is within $\frac{44}{\sqrt{\log k}}$ in $\delta_\square$ of $W_{\widehat{\mathcal{G}}(k,U)}$ with probability at least $1 - e^{-k/(2\log k)}$, and so in particular

$$\delta_\square(G_k, U) \le \frac{44}{\sqrt{\log k}}.$$

This therefore lets us create a sequence $\langle G_n\rangle_{k\in\mathbb{N}}$ such that $\langle W_{G_k}\rangle_{k\in\mathbb{N}}$ is a $\delta_\square$-name for any graphon with the same distribution as U.

To establish the computable equivalences in (a) and (b), we use the fact that convergence in $\delta_\square$ is equivalent to convergence of the corresponding random graphs (Theorem 3.18). This also shows that (a) and (b) induce bijections on the corresponding metric spaces.

We now show that (c) has a computable section. Assume we have a $\delta_\square$-name of U, and know a graph G_n such that

$$\delta_\square(W_{G_n}, U) < 2^{-(2^{2n}+1)}.$$

We will find a graph G_{n+1} such that

$$d_\square(W_{G_{n+1}}, W_{G_n}) \le 45 \cdot 2^{-n}$$

and

$$\delta_\square(W_{G_{n+1}}, U) \le 2^{-(2^{2(n+1)}+1)}.$$

This is enough to get a fast Cauchy sequence in $d_\square$. Find H such that

$$\delta_\square(W_H, U) < 2^{-(2^{2(n+1)}+1)}.$$

Then

$$\delta_\square(W_H, W_{G_n}) < 2^{-(2^{2n}+1)} + 2^{-(2^{2(n+1)}+1)} < 2^{-2^{2n}}.$$

There are graphs G_n' and H' both on the set $[|G_n| \cdot |H|]$ (where $|G_n|$ denotes the number of vertices of G_n, and similarly with H) such that

$$d_\square(W_{G_n'}, W_{G_n}) = d_\square(W_{H'}, W_H) = 0,$$

by taking blow-ups to a common refinement. Following the notation in [8, §8.1.3], define the quantity

$$\widehat{\delta}_\square(G_n', H') := \min_{\widehat{G'}_n, \widehat{H'}} d_\square(W_{\widehat{G'}_n}, W_{\widehat{H'}}),$$

where $\widehat{G'}_n$ ranges over the images of G_n' under permutations of $[|G_n| \cdot |H|]$, and similarly with $\widehat{H'}$. By [8, Theorem 9.29], we have

$$\widehat{\delta}_\square(G_n', H') \le \frac{45}{\sqrt{-\log \delta_\square(W_{G_n'}, W_{H'})}} < \frac{45}{\sqrt{2^{2n}}} = 45 \cdot 2^{-n}.$$

Hence there is some reordering G_{n+1} of H' such that

$$d_\square(W_{G_{n+1}}, W_{G_n'}) \le 45 \cdot 2^{-n}.$$

Because $d_\square(W_{G_n'}, W_{G_n}) = 0$, we have

$$d_\square(W_{G_{n+1}}, W_{G_n}) \le 45 \cdot 2^{-n}.$$

By definition, $\delta_\square(W_{G_{n+1}}, W_{H'}) = 0$. Because $d_\square(W_{H'}, W_H) = 0$, we therefore have

$$\delta_\square(W_{G_{n+1}}, U) = \delta_\square(W_{G_{n+1}}, H) < 2^{-(2^{2(n+1)}+1)},$$

as desired.

Finally, (d) induces a bijection on the corresponding metric spaces, as noted in Lemma 3.8. □

We have seen there is a computable equivalence between $\delta_\square$-names for a graphon and names for the corresponding invariant measure. Further, given a $\delta_\square$-name, we can computably find a $d_\square$-name for a graphon yielding the same invariant measure. We have also seen that it is possible to transform a d_1-name to a $d_\square$-name in a computable way. It is therefore natural to ask whether there is a computable equivalence from a $d_\square$-name to a d_1-name. As we will see, in general there is not. This tells us that the d_1-name for a graphon contains fundamentally more computable information than an $d_\square$-name for a graphon.

As a consequence of Proposition 4.5, we obtain the following relationships among the numbered notions appearing after Definition 4.4.

Corollary 4.6. *For an invariant measure μ, notions* (1)*,* (2) *and* (3) *are equivalent, and are all implied by notion* (4)*.*

5. Almost-everywhere continuity

In this section, we describe a random-free graphon that is computable in d_1 but not weakly isomorphic to any a.e. continuous graphon. Note that this is in contrast to the computable de Finetti theorem [7], which can be seen as saying that in a 1-dimensional analogue of this setting, the measurable object representing the sampler is a.e. computable, and in particular a.e. continuous. This provides another example of how the 2-dimensional case is considerably more complicated than the 1-dimensional case.

The notion of a.e. continuity is sensitive to the underlying topology of the space. Since a graphon is a function from $[0,1]\times[0,1]$ to $[0,1]$, it is reasonable to consider, as the topology on the domain, the product topology with respect to the usual topology on $[0,1]$. But there are situations where it is natural to consider graphons that are a.e. continuous with respect to other topologies on $[0,1]$ but are not weakly isomorphic to an a.e. continuous graphon with respect to the standard topology on $[0,1]$. We show that our result holds even in these more general situations, as long as the topology on $[0,1]$ still generates the Borel sets.

We begin by describing the construction of a random-free graphon G, which can be thought of as a symmetric measurable subset of $[0,1]^2$. We have drawn this measurable subset in Figure 1 as a black (1) and white (0) picture, with $(0,0)$ in the upper-left corner (similar to an adjacency matrix, and as is common when drawing graphons).

Construction. *First, draw a 2×2 square grid (given by products of the intervals $[0,\frac{1}{2})$ and $[\frac{1}{2},1]$ on each axis) and color the 2 squares on the diagonal black. Then on each of the 2 off-diagonal squares, draw a 4×4 square grid (similarly, from products of half-open or closed intervals) and color the 8 diagonal squares black. Then on each of the remaining 24 squares, draw an 8×8 grid and color the diagonal black. Continue in this way, filling diagonal squares within unfilled squares to obtain the graphon G.*

This clearly describes a symmetric measurable subset of $[0,1]^2$, and hence a random-free graphon. (Essentially the same graphon has independently arisen in number theory [21].)

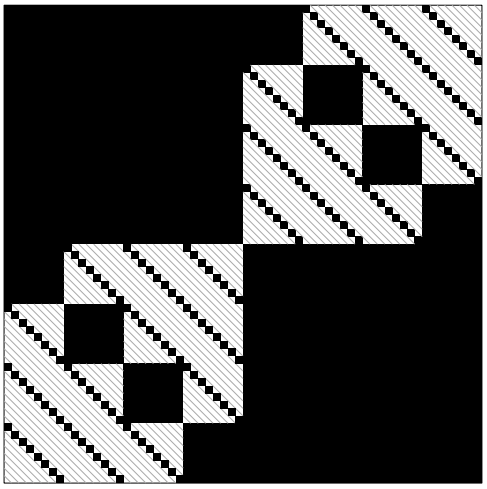

Figure 1. The graphon G, viewed as a subset of $[0,1]^2$.

The countable random graph induced by sampling from G may be thought of informally in the following way, which shows that it can be sampled in polynomial time: There is a questionnaire with an infinite list of questions indexed by positive integers. The nth question has 2^n possible answers. Each vertex corresponds to a person who has independently answered each question uniformly at random, independently from each other person. Two vertices are connected by an edge when the corresponding people agree on at least one answer to their questionnaires.

Lemma 5.1. *The graphon G constructed above is not a.e. continuous, has a computable d_1-name, and is twin-free.*

Proof. The black region $G^{-1}(\{1\})$ is clearly dense. Also the white region $G^{-1}(\{0\})$ has measure equal to $\alpha := \frac{1}{2}\cdot\frac{3}{4}\cdot\frac{7}{8}\cdots$ which is positive by the following calculation:

$$\begin{aligned}
-\log(\frac{1}{2}\cdot\frac{3}{4}\cdot\frac{7}{8}\cdots) &= -\sum_{n=1}^{\infty}\log(1-2^{-n}) \\
&= \sum_{n=1}^{\infty}\sum_{k=1}^{\infty}\frac{2^{-kn}}{k} \\
&= \sum_{k=1}^{\infty}\frac{1}{k}\sum_{n=1}^{\infty}2^{-kn} \\
&= \sum_{k=1}^{\infty}\frac{1}{k}\frac{1}{2^k-1} < \infty.
\end{aligned}$$

So this graphon itself is not a.e. continuous, since $G^{-1}(\{0\})$ is a nowhere dense set of positive measure.

But G has a computable d_1-name since one can approximate the graphon in d_1 with some initial stage of the

construction, as we now describe. Let G_n be the nth stage of the construction, and let $\beta_n := \int G_n \, d\lambda$. Then the measure of the remaining black area yet to be added to G is $(1-\alpha)-\beta_n$, a computable real that rapidly tends to 0 as $n \to \infty$.

Finally, observe that G is twin-free, as by construction, each horizontal line gives rise to a different cross-section. □

The main result of this section is that no graphon H weakly isomorphic to G is a.e. continuous (even with respect to other topologies that generate the Borel sets). The key combinatorial fact is the following.

Lemma 5.2. *Let G be the random-free graphon constructed above. Suppose $X, Y \subseteq [0,1]$ are measurable sets such that $X \times Y$ is contained in $G^{-1}(\{0\})$ up to a nullset. Then $\lambda(X \times Y) = 0$.*

Proof. For each $n \in \mathbb{N}$, let X_n (respectively Y_n) be the union of all dyadic half-open intervals of size $2^{-(2^n-1)}$ whose intersection with X (respectively Y) has positive measure. To show that $\lambda(X \times Y) = 0$, we will show by induction that $\lambda(X_n \times Y_n) \leq 4^{-n}$.

The base case is trivial as $\lambda(X_0 \times Y_0) \leq 1$. For the inductive step, consider each dyadic square $I \times J$ where $I \subseteq X_n$ and $J \subseteq Y_n$ are both of size $2^{-(2^n-1)}$. By construction, for each sub-dyadic interval $I' \subseteq I$ of size $2^{-(2^{n+1}-1)}$, there is a corresponding dyadic interval $J' \subseteq J$ of the same size such that $I' \times J'$ is a black square. If I' is disjoint from X (up to a nullset), then $I' \subseteq X_n \setminus X_{n+1}$. Otherwise, J' is disjoint from Y (up to a nullset), and $J' \subseteq Y_n \setminus Y_{n+1}$; for if not, then the black square $I' \times J'$ intersects $X \times Y$ outside a nullset, which cannot happen, since $X \times Y \subseteq G^{-1}(\{0\})$ is white. After considering all such sub-dyadic intervals I' we have that

$$\lambda(X_{n+1} \cap I) + \lambda(Y_{n+1} \cap J) \leq \frac{\lambda(I) + \lambda(J)}{2} = \lambda(I).$$

By the arithmetic–geometric mean inequality,

$$\begin{aligned}\lambda\big((X_{n+1} \times Y_{n+1}) \cap (I \times J)\big) &= \lambda(X_{n+1} \cap I) \cdot \lambda(Y_{n+1} \cap J) \\ &\leq \left(\frac{\lambda(X_{n+1} \cap I) + \lambda(Y_{n+1} \cap J)}{2}\right)^2 \\ &\leq \left(\frac{\lambda(I)}{2}\right)^2 = \frac{\lambda(I)^2}{4}.\end{aligned}$$

Summing up over all such $I \times J$ and using the induction hypothesis we have

$$\lambda(X_{n+1} \times Y_{n+1}) \leq \frac{\lambda(X_n \times Y_n)}{4} \leq 4^{-(n+1)}.$$

Therefore $\lambda(X \times Y) = 0$. □

We may now prove the main result about G.

Theorem 5.3. *Let G be the random-free graphon (which has a computable d_1-name) constructed above. Let H be a graphon weakly isomorphic to G, and let $\mathcal{T}$ be some topology on $[0,1]$ every open set of which is a standard Borel set. Then H is not a.e. continuous with respect to $\mathcal{T} \times \mathcal{T}$.*

Proof. Because G is twin-free and weakly isomorphic to H, by [18, Theorem 8.6 (vi)] there is a measure-preserving map $\psi\colon [0,1] \to [0,1]$ such that $H = G^\psi$ a.e. Hence H must be random-free as well.

Now assume, towards a contradiction, that the map H is a.e. continuous with respect to $\mathcal{T} \times \mathcal{T}$. Because ψ is measure-preserving,

$$\lambda\big(G^{-1}(\{0\})\big) = \lambda\big(H^{-1}(\{0\})\big).$$

Define $A \subseteq [0,1]^2$ to be the set $H^{-1}(\{0\})$.

Since H is a.e. continuous with respect to $\mathcal{T} \times \mathcal{T}$, we have that A is a λ-continuity set, and therefore its interior (in the product topology $\mathcal{T} \times \mathcal{T}$) is standard Borel and has the same (positive) measure as A. Hence there is some open set $B \times C \subseteq A$ where B and C are $\mathcal{T}$-open sets of $[0,1]$ (and hence are standard Borel sets) that have positive measure.

Let $\lambda(\cdot|B)$ denote Lebesgue measure conditioned on B, that is

$$\lambda(A|B) = \frac{\lambda(A \cap B)}{\lambda(B)}.$$

Both $\lambda(\cdot|B)$ and $\lambda(\cdot|C)$ are well-defined since B and C have positive measure. Now, let μ_B and μ_C denote the pushforward measures on $[0,1]$ of $\lambda(\cdot|B)$ and $\lambda(\cdot|C)$ along the map ψ. That is, $\mu_B(S) = \lambda(\psi \in S|B)$ for all measurable $S \subseteq [0,1]$, and likewise with C. We claim that μ_B and μ_C are absolutely continuous with respect to λ. Indeed, if $\lambda(S) = 0$, then

$$\mu_B(S) = \lambda(\psi \in S|B) \leq \frac{\lambda(\psi \in S)}{\lambda(B)} = \frac{\lambda(S)}{\lambda(B)} = 0.$$

Therefore, the supports of μ_B and μ_A have positive λ-measure.

Because ψ is a measure-preserving map, we have

$$\begin{aligned}(\mu_B \otimes \mu_C)\{G = 0\} &= \frac{1}{\lambda(A)\lambda(B)} \int_C \int_B \mathbf{1}_{\{G^\psi = 0\}} \, d\lambda \, d\lambda \\ &= \frac{1}{\lambda(A)\lambda(B)} \int_C \int_B \mathbf{1}_{\{H = 0\}} \, d\lambda \, d\lambda \\ &= 1,\end{aligned}$$

where the last equality follows from the fact that $H = 0$ on $B \times C$.

Now let X be the support of μ_B and Y be the support of μ_C. Then $X \times Y$ is contained in $G^{-1}(\{0\})$ up to a nullset. By Lemma 5.2, we have $\lambda(X \times Y) = 0$, and so one of X and Y has measure 0, a contradiction to the fact that μ_B and μ_C are absolutely continuous probability measures. Hence H is not a.e. continuous with respect to $\mathcal{T} \times \mathcal{T}$. □

6. $d_\square$-names vs. d_1-names: Upper bound

In Section 4 we introduced four representations for the space of graphons and showed that the first three are equivalent. In this section we characterize the relative computability of the fourth representation (i.e., d_1-names for graphons). Our theorems involve encodings of the halting set, and so we must introduce a few pieces of notation. For $e \in \mathbb{N}$, let $\{e\}$ denote the partial computable function $\mathbb{N} \to \mathbb{N}$ given by computer program e. For $n \in \mathbb{N}$ write $\{e\}(n)\downarrow$ to

denote that this program halts on input n, and $\{e\}(n)\uparrow$ to denote that it does not halt. For $s \in \mathbb{N}$ write $\{e\}_s(n)\downarrow$ to denote that this program has halted after at most s steps, and $\{e\}(n)_s\uparrow$ to denote that it has not yet halted after s steps. Write $0' := \{e \in \mathbb{N} : \{e\}(0)\downarrow\}$ to denote the halting set.

Recall that a d_1-name for a point is already a $d_\square$-name for that same point. In this section we establish that the halting problem $0'$ suffices as an oracle to computably transform a computable $d_\square$-name to a d_1-name. Further, in the random-free case, this oracle is not needed. In the next section we show that this is tight, in the sense that the use of $0'$ is necessary in general.

For $n \in \mathbb{N}$, let $\mathcal{P}_n$ denote the equipartition of $[0,1]$ into 2^n-many intervals of width 2^{-n}. For a graphon U we write $U_{\mathcal{P}_n}$ (as in [8, §7.1]) to denote the step function graphon $\mathbb{E}[U \mid \mathcal{P}_n \times \mathcal{P}_n]$, i.e., the conditional expectation of the function U averaged on this $2^n \times 2^n$ square grid.

Proposition 6.1. *Let U be a graphon computable in $d_\square$. Then $\langle U_{\mathcal{P}_n}\rangle_{n\in\mathbb{N}}$ is a uniformly d_1-computable sequence of graphons that converges in d_1 to U.*

Proof. Suppose $\langle V_k\rangle_{k\in\mathbb{N}}$ is a $d_\square$-name for U. The conditional expectation operator with respect to $\mathcal{P}_n \times \mathcal{P}_n$ is contractive with respect to the cut norm [8, Proposition 14.13]. Hence $d_\square(U_{\mathcal{P}_n}, (V_k)_{\mathcal{P}_n}) \le d_\square(U, V_k)$ for all $k, n \in \mathbb{N}$. We can therefore compute a $d_\square$-name for $U_{\mathcal{P}_n}$ uniformly in n.

However, because $U_{\mathcal{P}_n}$ is a $(\mathcal{P}_n \times \mathcal{P}_n)$-measurable step function, from a $d_\square$-name for $U_{\mathcal{P}_n}$ we can uniformly in n compute a d_1-name for $U_{\mathcal{P}_n}$. Therefore, from $\langle V_k\rangle_{k\in\mathbb{N}}$ we can uniformly in n compute a d_1-name for $U_{\mathcal{P}_n}$. In particular, $\langle U_{\mathcal{P}_n}\rangle_{n\in\mathbb{N}}$ is a uniformly d_1-computable sequence of graphons which converges in d_1 to U. □

Although $\langle U_{\mathcal{P}_n}\rangle_{n\in\mathbb{N}}$ converges in d_1 to U, it need not converge quickly. Its Turing jump allows us to identify a rapidly converging subsequence, yielding the next theorem. As we will see in Section 7, the Turing jump is necessary.

Theorem 6.2. *Let U be a graphon. Then from the Turing jump of any $d_\square$-name of U, we can compute a d_1-name for it. In particular, if U has a computable $d_\square$-name, then it has a $0'$-computable d_1-name.*

Proof. Let X be some $d_\square$-name for U. Then by Proposition 6.1, using X, uniformly in n we can compute a d_1-name for $U_{\mathcal{P}_n}$. Since $\langle U_{\mathcal{P}_n}\rangle_{n\in\mathbb{N}}$ converges in d_1, its limit is computable in d_1 from the Turing jump of any sequence of d_1-names for $\langle U_{\mathcal{P}_n}\rangle_{n\in\mathbb{N}}$. Hence we can compute a d_1-name for U from the Turing jump of X. □

It is natural to consider the case of random-free graphons, especially since the ability to flip between greyscale regions and black-and-white ones will be key to the lower-bound proof in Section 7.

In fact, in the random-free case, computable convergence in $d_\square$ is equivalent to computable convergence in d_1, as we now show.

Lemma 6.3. *If $\langle U_n\rangle_{n\in\mathbb{N}}$ is a uniformly d_1-computable sequence of graphons that converges to a random-free graphon U, then U has a computable d_1-name.*

Proof. The key idea is that since U is 0–1-valued a.e., the value $d_1(U_n, U)$ is computable. This is because given a square in $\mathcal{P}_n \times \mathcal{P}_n$, the Lebesgue measure of those points (x,y) in the square such that $U(x,y) = 1$ is equal to the value that U_n takes there, and this suffices to compute the L_1 norm of the difference on that square. For example, for the constant function U_0, if $U_0 \equiv p$, then

$$\begin{aligned} d_1(U_0, U) &= \lambda\{U = 0\} \cdot p + \lambda\{U = 1\} \cdot (1-p) \\ &= (1-p)p + p(1-p) = 2p(1-p). \end{aligned}$$

Since $\langle U_n\rangle_{n\in\mathbb{N}}$ is a d_1-name for U, and since we can compute each quantity $d_1(U_n, U)$, we may find a subsequence $\langle U_{k_n}\rangle_{n\in\mathbb{N}}$ such that $d_1(U_{k_n}, U) < 2^{-n}$. From this sequence $\langle U_{k_n}\rangle_{n\in\mathbb{N}}$ we can find a computable d_1-name for U. □

This implies that there is a computable procedure for translating d_1-names to $d_\square$-names in the case of a random-free graphon.

Theorem 6.4. *Let U be a random-free graphon computable in $d_\square$. Then U is computable in d_1.*

Proof. Since U is computable in $d_\square$, by Proposition 6.1 the sequence $\langle U_{\mathcal{P}_n}\rangle_{n\in\mathbb{N}}$, which converges to U in d_1, is a uniformly d_1-computable sequence of graphons. Then, because U is random-free, we can compute a d_1-name for the limit of this sequence by Lemma 6.3. □

We have just seen that for random-free graphons, unlike the general case, d_1 and $d_\square$-names can be computably transformed into each other. One might therefore wonder whether one can computably determine that a graphon is random-free. In fact, it is not possible to computably recognize when a graphon is random-free, as we now demonstrate. On the other hand $0'$ does allow us to recognize this.

Proposition 6.5. *The collection of d_1-names for random-free graphons is a Π^0_1 class. Further, there is a uniformly d_1-computable sequence of graphons $\{V_e\}_{e\in\mathbb{N}}$ such that V_e is not random-free if and only if $e \in 0'$.*

Proof. Recall that $W(1-W) \ge 0$ a.e. for a graphon W, as it takes values in $[0,1]$. Also, W is random-free if and only if $\int W(1-W)\,d\lambda = 0$. (See, e.g., [18, Lemma 10.4].) Therefore, given a d_1-name for a graphon, we may compute $\int W(1-W)\,d\lambda$, and hence by noticing when this quantity is positive, we may enumerate the d_1-names of the non-random-free graphons. Hence the d_1-names of random-free graphons form a Π^0_1 class.

For $s, e \in \mathbb{N}$, let U^s_e be the constant function 2^{-s} if $\{e\}_s(0)\uparrow$, and U^s_e be the constant function 2^{-k} if $k \le s$ and minimal with $\{e\}_k(0)\downarrow$. Observe that for each $e \in \mathbb{N}$, the sequence $\langle U^s_e\rangle_{s\in\mathbb{N}}$ is a computable d_1-name for a graphon V_e that is not random-free if and only if $e \in 0'$. □

In particular, there is no computer program that, given a d_1-name of a graphon, correctly asserts whether or not the graphon is random-free.

Having shown that every graphon with a computable $d_\square$-name has a d_1-name that is $0'$-computable, one may ask if this is tight, i.e., if $0'$ is necessary. We have just seen that this is not tight in the random-free case, and so any witness to the necessity of $0'$ must not be random-free. Next, in Section 7, we provide such an example.

7. $d_\square$-names vs. d_1-names: Lower bound

We have just seen that using $0'$ we can compute a d_1-name of a graphon given a computable $d_\square$-name for it. We now show that this is tight in the sense that there is a graphon that is computable in $d_\square$ such that $0'$ is computable from any d_1-name for a graphon weakly isomorphic to it. Furthermore, we may take this graphon to be a.e. continuous.

Theorem 7.1. *There is an a.e. continuous graphon Z that is computable in $d_\square$, such that if V is weakly isomorphic to Z then any d_1-name for V computes the halting problem $0'$.*

Proof. For each $n \in \mathbb{N}$, define the open interval $A_n := (1-2^{-n}, 1-2^{-(n+1)})$. The graphon Z will take the value zero outside the block-diagonal $\bigcup_{n\in\mathbb{N}}(A_n \times A_n)$. Also, for each $n \in \mathbb{N}$, let $\ell_n := 1-2^{-(2n+1)}$, let $r_n := 1-2^{-(2n+2)}$, and let $m_n := \frac{\ell_n + r_n}{2}$, so that

$$0 < \ell_0 < m_0 < r_0 < \ell_1 < m_1 < r_1 < \ldots < 1.$$

Define the constant graphon $H^* \equiv \frac{1}{2}$. Because H^* is computable in $d_\square$, for each $s \in \mathbb{N}$ we can computably find a random-free step function graphon $G^*_s \in \mathcal{S}$ such that $\|H^* - G^*_s\|_\square < 2^{-s}$. For each $e \in \mathbb{N}$, let $\iota_e \colon [0,1] \to [\ell_e, r_e]$ be the unique increasing linear bijection. Note that $\iota_e(\frac{1}{2}) = m_e$. Let $G_{e,s} := \iota_e \circ G^*_s$ and $H_e := \iota_e \circ H^*$. Observe that

$$\|H_e - G_{e,s}\|_\square < \|H^* - G^*_s\|_\square < 2^{-s}.$$

Now for $s \in \mathbb{N}$, define Z_s to be the graphon that is 0 outside of $\bigcup_{e\le s}(A_e \times A_e)$ and for each $e \le s$ is equal to the scaling to fit $A_e \times A_e$ of the following graphon $K_{e,s}$ on $[0,1]\times[0,1]$:

$$K_{e,s} := \begin{cases} H_e & \text{if } \{e\}_s(0)\uparrow, \text{ and} \\ G_{e,t} & \text{if } t \le s \text{ is minimal such that } \{e\}_t(0)\downarrow. \end{cases}$$

Note that

$$\begin{aligned} \|Z_s - Z_{s+1}\|_\square &\le 2^{-s} + \sum_{n>s} \lambda(A_n \times A_n) \\ &< 2^{-s} + 2^{-(s+1)} \cdot 2^{-(s+1)} \\ &< 2^{-s+1}. \end{aligned}$$

Clearly the graphons $K_{e,s}$ are uniformly computable in $d_\square$, and so the sequence $\langle Z_s \rangle_{s\in\mathbb{N}}$ is a computable $d_\square$-name. For $x, y \in [0,1]$, define

$$Z(x,y) = \lim_{s\to\infty} Z_s(x,y)$$

when it is defined, and 0 otherwise. Note that Z is a limit of the sequence $\langle Z_s \rangle_{s\in\mathbb{N}}$, and it is a.e. continuous, as it is piecewise constant (i.e., a step function with countably many steps).

Note that for each $n \in \mathbb{N}$, we have

$$\lambda(Z^{-1}\{\ell_n, m_n, r_n\}) = \lambda(A_n \times A_n) = 2^{-2(n+1)}$$

by construction, as each $K_{n,s}$ only takes values among ℓ_n, m_n, or r_n. For any graphon W define the set

$$X_W := \{x \in [0,1] : \lambda(W^{-1}(x)) > 0\}.$$

Note that for any V weakly isomorphic to Z, by Theorem 3.18 condition (3) we have $X_V = X_Z$. Further, as

$$\{\ell_n, m_n, r_n\} \cap \{\ell_p, m_p, r_p\} = \emptyset$$

for $p \ne n$, from a d_1-name for V we can compute the countable discrete set X_Z. But $m_e \in X$ if and only if $\{e\}(0)\uparrow$, and so X_Z computes $0'$. □

We note that by taking the direct sum of the graphon G constructed in Section 5 with the graphon Z that we have just constructed, we may obtain a $0'$-computable d_1-name for a graphon for which no weakly isomorphic graphon is a.e. continuous and from which any d_1-name computes $0'$. Namely, consider Z scaled by $\frac{1}{2}$ placed on $[0,\frac{1}{2}]^2$, along with G scaled by $\frac{1}{2}$ placed on $[\frac{1}{2},1]^2$.

Acknowledgments

The authors would like to thank Jan Reimann and Ted Slaman for helpful discussions, Rachel Paiste and Feras Saad for comments on a draft, and an anonymous referee for a suggestion that improved the proof of Proposition 6.1. Work on this publication was made possible through the support of United States Air Force Office of Scientific Research (AFOSR) Contract FA9550-15-1-0074. J.A. was supported by AFOSR Contracts FA9550-15-1-0053 and FA9550-18-1-0120, and National Science Foundation (NSF) grant DMS-1615444. C.E.F. was supported by United States Air Force (USAF) and the Defense Advanced Research Projects Agency (DARPA) Contracts FA8750-14-C-0001 and FA8750-14-2-0004, Army Research Office (ARO) grant W911NF-13-1-0212, Office of Naval Research (ONR) grant N00014-13-1-0333, NSF grants DMS-0800198 and DMS-0901020, a grant from the John Templeton Foundation, and by Google, the MIT CSAIL Systems That Learn Initiative, and an anonymous philanthropic gift. D.M.R. was supported by a Newton International Fellowship, Emmanuel Research Fellowship, NSERC Discovery Grant, Connaught Award, and Google Faculty Research Award. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force, Army, Navy, DARPA, or the John Templeton Foundation.

References

[1] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*, 3rd ed., ser. Texts in Statistical Science Series. CRC Press, Boca Raton, FL, 2014.

[2] A. D. Gordon, T. A. Henzinger, A. V. Nori, and S. K. Rajamani, "Probabilistic programming," in *Proc. Future of Software Eng. (FOSE 2014)*. New York, NY, USA: ACM, 2014, pp. 167–181.

[3] S. Staton, D. Stein, H. Yang, N. L. Ackerman, C. E. Freer, and D. M. Roy, "The Beta-Bernoulli process and algebraic effects," in *45th Int. Colloq. Automata, Languages, and Programming (ICALP 2018)*, ser. LIPIcs, vol. 107. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018, pp. 141:1–141:15.

[4] N. D. Goodman, V. K. Mansinghka, D. M. Roy, K. Bonawitz, and J. B. Tenenbaum, "Church: A language for generative models," in *Proc. 24th Conf. Uncertainty in Artificial Intelligence (UAI 2008)*. Corvalis, Oregon: AUAI Press, 2008, pp. 220–229.

[5] V. Mansinghka, D. Selsam, and Y. Perov, "Venture: a higher-order probabilistic programming platform with programmable inference," 2014, arXiv eprint 1404.0099.

[6] N. L. Ackerman, J. Avigad, C. E. Freer, D. M. Roy, and J. M. Rute, "On computable representations of exchangeable data," Workshop on Probabilistic Programming Semantics (PPS 2017), 2017.

[7] C. E. Freer and D. M. Roy, "Computable de Finetti measures," *Ann. Pure Appl. Logic*, vol. 163, no. 5, pp. 530–546, 2012.

[8] L. Lovász, *Large networks and graph limits*, ser. Amer. Math. Soc. Colloq. Publ. Amer. Math. Soc., Providence, RI, 2012, vol. 60.

[9] O. Kallenberg, *Probabilistic symmetries and invariance principles*, ser. Probability and its Applications. New York: Springer, 2005.

[10] M. Braverman and S. Cook, "Computing over the reals: foundations for scientific computing," *Notices Amer. Math. Soc.*, vol. 53, no. 3, pp. 318–329, 2006.

[11] K. Weihrauch, *Computable analysis*, ser. Texts in Theoretical Computer Science. Berlin: Springer, 2000.

[12] M. Hoyrup and C. Rojas, "Computability of probability measures and Martin-Löf randomness over metric spaces," *Inform. and Comput.*, vol. 207, no. 7, pp. 830–847, 2009.

[13] V. Brattka and G. Gherardi, "Effective choice and boundedness principles in computable analysis," *Bull. Symbolic Logic*, vol. 17, no. 1, pp. 73–117, 2011.

[14] D. J. Aldous, "Representations for partially exchangeable arrays of random variables," *J. Multivariate Analysis*, vol. 11, no. 4, pp. 581–598, 1981.

[15] D. N. Hoover, "Relations on probability spaces and arrays of random variables," 1979, preprint, Institute for Advanced Study, Princeton, NJ.

[16] K. Miyabe, "L^1-computability, layerwise computability and Solovay reducibility," *Computability*, vol. 2, no. 1, pp. 15–29, 2013.

[17] O. Kallenberg, "Multivariate sampling and the estimation problem for exchangeable arrays," *J. Theoret. Probab.*, vol. 12, no. 3, pp. 859–883, 1999.

[18] S. Janson, *Graphons, cut norm and distance, couplings and rearrangements*, ser. New York J. Math. Monographs. Albany, NY: Univ. at Albany, State Univ. of New York, 2013, vol. 4.

[19] T. Austin, "On exchangeable random variables and the statistics of large graphs and hypergraphs," *Probab. Surv.*, vol. 5, pp. 80–145, 2008.

[20] P. Diaconis and S. Janson, "Graph limits and exchangeable random graphs," *Rend. Mat. Appl. (7)*, vol. 28, no. 1, pp. 33–61, 2008.

[21] S. Martineau, "On coprime percolation, the visibility graphon, and the local limit of the GCD profile," 2018, arXiv eprint arXiv:1804.06486.