Electronic Journal of Statistics
Vol. 16 (2022) 331–407
ISSN: 1935-7524
https://doi.org/10.1214/21-EJS1940

Reconstruction of line-embeddings of graphons[*]

Jeannette Janssen

Department of Mathematics and Statistics
Dalhousie University
Halifax, NS, Canada
e-mail: Jeannette.Janssen@dal.ca

Aaron Smith

Department of Mathematics and Statistics
University of Ottawa
Ottawa, ON, Canada
e-mail: smith.aaron.matthew@gmail.com

Abstract: Consider a random graph process with n vertices corresponding to points $v_i \overset{i.i.d.}{\sim} \text{Unif}[0,1]$ embedded randomly in the interval, and where edges are inserted between v_i, v_j independently with probability given by the *graphon* $w(v_i, v_j) \in [0,1]$. Following [11], we call a graphon w *diagonally increasing* if, for each x, $w(x,y)$ decreases as y moves away from x. We call a permutation $\sigma \in S_n$ an *ordering* of these vertices if $v_{\sigma(i)} < v_{\sigma(j)}$ for all $i < j$, and ask: how can we accurately estimate σ from an observed graph? We present a randomized algorithm with output $\hat{\sigma}$ that, for a large class of graphons, achieves error $\max_{1 \leq i \leq n} |\sigma(i) - \hat{\sigma}(i)| = \tilde{O}(\sqrt{n})$ with high probability; we also show that this is the best-possible convergence rate for a large class of algorithms and proof strategies. Under an additional assumption that is satisfied by some popular graphon models, we break this "barrier" at $\sqrt{n}$ and obtain the vastly better rate $\tilde{O}(n^{\epsilon})$ for any $\epsilon > 0$. These improved seriation bounds can be combined with previous work to give more efficient and accurate algorithms for related tasks, including *estimating diagonally increasing graphons* [20, 21] and *testing whether a graphon is diagonally increasing* [11].

MSC2020 subject classifications: Primary 60-04; secondary 60B20.
Keywords and phrases: Random graph, graphon, probabilistic algorithm, R-matrix.

Received September 2020.

Contents

[*]The authors acknowledge support from the National Science and Engineering Research Council through their Discovery Grant program.

1. Introduction

In this paper, we propose and analyze new algorithms for estimating *latent vertex labellings* given an *observed random graph*. We first discuss a motivating simpler problem, often called the *seriation problem* (this dates back at least to the 1899 paper [34]; see also *e.g.* [24], [23]). The basic seriation problem considers a *Robinsonian similarity matrix*, i.e. a symmetric n by n matrix $A = [a_{i,j}]_{1 \leq i,j \leq n}$ with the property: there exists a permutation $\sigma \in S_n$ so that every row of the permuted matrix $A_\sigma = [a_{\sigma(i),\sigma(j)}]_{1 \leq i,j \leq n}$ is unimodal with the maximum occurring at the diagonal. It then asks: **how can we find such a permutation σ?**

This problem turns out to have an elegant and computationally-tractable solution. For any embedding $\phi : \{1, 2, \ldots, n\} \mapsto [0, 1]$ of the indices of A into the line segment $[0, 1]$, we define the *induced* permutation $\sigma \in S_n$ by the equation

$$\sigma(i) = \sigma(i, \phi) \equiv |\{j \in [1 : n] \,:\, \phi(j) \leq \phi(i)\}|, \tag{1.1}$$

breaking ties arbitrarily. It turns out that, under mild conditions, the correct permutation σ is of the form

$$\sigma(i) = \sigma(i, \hat{\phi}),$$

where $\hat{\phi}$ is an eigenvector of a matrix related to A [1].

This seriation problem occurs in a number of contexts, where we have some collection of objects $\{1, 2, \ldots, n\}$ which we would like to order, (*e.g.* ordering types of artifacts by their ages), and some measure $A = [a_{ij}]$ as to whether a pair of objects are similar (*e.g.* their co-occurrence in tombs). A special case is graph seriation, where the aim is to determine whether a permutation of the vertices exists so that the corresponding adjacency matrix has the Robinsonian property, and to find such a permutation.

Of course, real data is noisy. Even if the *expected value* of the similarity matrix is Robinsonian, the *observed* similarity matrix may not be and a "perfect" ordering may not exist. Our aim is to solve a very generic "noisy" version of the graph seriation problem. Following [11], we introduce noise by studying a natural generalization of the above example based on some models from Bayesian nonparametrics. Recall that a *graphon* is just a symmetric measurable function $w : [0, 1]^2 \mapsto [0, 1]$ (these were introduced in [28]; see [27] for a broader survey). Under suitable measurability conditions, a graphon defines an algorithm for sampling a random graph of any size n:

1. Sample i.i.d. sequences $\{U_i\}_{i=1}^n$, $\{U_{ij}\}_{1 \leq i < j \leq n} \sim \text{Unif}([0, 1])$.
2. Define the edge set by

$$\{(i, j) \in E\} \iff \{U_{i,j} < w(U_i, U_j)\}. \tag{1.2}$$

We write $G \sim w$ if G is a random graph obtained from graphon w in this way. In [11], the authors say that a graphon w *is diagonally increasing* if it satisfies

$$\begin{aligned} w(x, y) &\leq w(z, y) \\ w(x, z) &\geq w(x, y) \end{aligned} \tag{1.3}$$

for all $x < z < y$. Note that this implies that $w(x, \cdot)$ is unimodal with the maximum occurring at x, so this definition matches that of the Robinsonian property for matrices.

We think of the following simple and well-studied family of graphons as our prototypical examples, and will use this family as a running example in the text:

$$w(x,y) = \begin{cases} p & |x-y| < d \\ q & |x-y| \geq d. \end{cases} \tag{1.4}$$

These graphons satisfy (1.3) as long as $0 \leq q \leq p \leq 1$ (though reconstructing an ordering is clearly impossible if $p = q$ or $d \in \{0,1\}$). Very similar random graphs have been extensively studied (see *e.g.* [2, 5, 14, 36]).

Special cases of this noisy seriation problem (with points sometimes embedded on a circle instead of an interval) appear in many areas. In some instances, an approximate ordering with an error rate similar to that presented in this paper has been obtained. However, generally such results only apply to a narrow class of random graphs, and the restriction of the problem to this specific class allows for the use of highly specialized methods. A few examples beyond archaeology include genomics [2, 35], ranking [17], data visualization [32], and ecology and sociology [24].

Graphs sampled from a diagonally increasing graphon have a natural line-embedded structure, defined by the latent variables $(U_1,\ldots,U_n)$. Namely, vertex i will have a higher probability of linking to vertex j if U_j is closer to U_i. Consequently, the permutation that reveals the "almost" Robinsonian property of the adjacency matrix will be the permutation $\sigma(\cdot,(U_1,\ldots,U_n))$ induced by the values U_i. We call this permutation the *line-embedded permutation.* This raises the main question of our paper: **how accurately can we recover the line-embedded permutation based on an observed graph G?**

Remark 1.1. The embedding $\phi = (U_1,\ldots,U_n)$ and its "reverse" $1-\phi = (1-U_1,\ldots,1-U_n)$ are both equally valid line-embeddings for a sampled graph G, and it is impossible to distinguish between the line-embedded permutation $\sigma(\cdot,\phi)$ and its reverse $\sigma^{rev} = \sigma(\cdot,1-\phi)$ based on an observed graph. In informal discussion we will often implicitly mod out by this reverse operation and talk about "the unique" line-embedded permutation without further comment.

A natural idea is to apply a spectral method as used in [1] for the noiseless matrix seriation problem. A related approach was studied in the recent work [36] in the special case that w is of the form (1.4) and $d = 0.5$, $q = 0$. Theorem 3 of this paper says that, with high probability, there exists a set $I \subset \{1,2,\ldots,n\}$ of size $|I| = n(1-o(1))$ so that

$$\max_{i \in I} |\hat{\sigma}(i) - \sigma(i,(U_1,\ldots,U_n))| = \tilde{O}(n^{0.5}). \tag{1.5}$$

However, the authors indicate that their methods cannot be easily extended to other graphons or, indeed, other parameters of (1.4). It was also not clear if this bound could be improved by other methods.

Our paper introduces and analyzes two new algorithms for the noisy graph seriation problem, which approximate the line-embedded permutation for graphs sampled from a large general class of diagonally increasing graphons. The paper has two main results, which bound the error in the approximate permutation returned by our algorithms. In Theorem 1, we extend the result from (1.5) to a very general class of graphons, and obtain an error term of

$$\max_{1 \leq i \leq n} |\hat{\sigma}(i) - \sigma(i, (U_1, \ldots, U_n))| = \tilde{O}(n^{0.5}).$$

In Theorem 2, we greatly improve this bound for a slightly smaller class of graphons, showing

$$\max_{1 \leq i \leq n} |\hat{\sigma}(i) - \sigma(i, (U_1, \ldots, U_n))| = \tilde{O}(n^{\epsilon}) \tag{1.6}$$

for any fixed $\epsilon > 0$. This smaller class still contains the most popular statistical models of graphons, including those of the form (1.4) in the special case $q = 0$.

Remark 1.2 (Errors in Positions and Orderings)**.** We pause to explain why this result may be surprising. In practice, many latent-position models are analyzed by algorithms that *first* estimate all latent positions and *then* plug this estimator into a formula for a quantity of interest (in seriation this quantity of interest is the ordering, but see also [26] for applications of the same approach to other problems). This approach seems sensible under the condition that the first step is not much less accurate than the second step.

However, this condition turns out to *fail* in the context of seriation. In a fairly strong sense that we make precise in Section 6, it is not possible to reconstruct the latent positions $U_1, \ldots, U_n$ themselves with an error that is similar to the error on the positions given in (1.6). We believe that such a discrepancy between the achievable error in reconstructing *ordering* and the achievable error in reconstructing *latent positions* is important for practical algorithm development in the area, as it suggests that beginning with an embedding step could result in a statistically inefficient algorithm.

See Section 6 for further discussion and a more precise version of this heuristic.

1.1. High-level algorithm descriptions

We sketch our main algorithms and give heuristics for why they work. Our first algorithm (Algorithm 1) proceeds as follows:

1. We begin by computing a new graph $G_{\alpha}^{(2)}$ from the observed graph G, as follows:
   (a) We square the adjacency matrix; its entries represent the number of common neighbours of pairs of vertices. Since the number of common neighbours are the sum of many independent Bernouilli trials, the entries of this squared matrix concentrate around their expected values.

   (b) We convert the square matrix into a binary matrix, and thus a graph, by thresholding at some level roughly αn. We show the thresholded matrix is “almost” Robinsonian, in the sense that any violations occur in the very narrow bad region where the expected value of the squared matrix is very close to the threshold value αn.

2. We then take many small random subgraphs of $G_\alpha^{(2)}$ and attempt to order them using a deterministic algorithm. Since $G_\alpha^{(2)}$ is “almost” Robinsonian, most of these subgraphs will be exactly Robinsonian.
3. We then align our orderings of these small subgraphs. (Recall from Remark 1.1 that individual graphs can't distinguish between σ_{true} and σ_{eurt}, so our subgraph orderings will generally not all be aligned to the same one.) This turns out to be straightforward as long as there are enough small subgraphs to guarantee substantial overlap.
4. Finally, we merge all of our subgraph orderings into a large ordering, essentially by a voting procedure. All of the steps up to this point preserve quite a bit of randomness, and so we again obtain concentration bounds for the result.

Sections 3.3 and 3.4 give a more detailed sketch of our analysis of Algorithm 1, giving precise error bounds in place of informal phrases such as “almost.” We believe it is possible for the reader to have a nearly-complete understanding of the algorithm by reading these two short sections; the remainder of Section 3 is concerned with filling in unsurprising details or ruling out failure modes that “obviously” should not occur.

The second algorithm (Algorithm 6), which achieves much smaller error bounds, proceeds in a sequence of stages.

1. The process starts with a coarse ordering achieved using Algorithm 1 on a small random subgraph.
2. In each subsequent stage, we use the approximate ordering on a small graph to obtain an ordering on a slightly larger graph with a slightly smaller error. These iterative steps use the graph itself (not its square). The additional condition of Theorem 2 guarantees that for every pair of vertices i, j, there is an interval $I_{i,j} \subset [0, 1]$ so that any vertex k with $U_k \in I_{i,j}$ may link to i, but cannot link to j. Vertices of the previous iteration with U_k-values estimated to be in this interval provide a signal of the true ordering of i and j.
3. In the final stage we return the permutation based on a final ordering on the full graph.

1.2. Basic notation

We say that a sequence of events $\mathcal{A} = \{\mathcal{A}^{(n)}\}$ indexed by $n \in \mathbb{N}$ hold *with extreme probability* if

$$\mathbb{P}[\mathcal{A}^{(n)}] \geq 1 - n^{-c_1 \log(n)^{c_2}}$$

for some $c_1, c_2 > 0$ and all $n > N_0$ sufficiently large. We use the shorthand $\mathcal{A}$ *holds w.e.p.*, and note that for any fixed $C < \infty$ the intersection of n^C events that occur w.e.p. also occurs w.e.p. This property motivates our definition. In particular, we will often show that a collection of events $\mathcal{A}_{i,j}$ hold w.e.p. for all $1 \leq i, j \leq n$, then cite this to conclude that $\cap_{i,j} \mathcal{A}_{i,j}$ occurs w.e.p.

We will measure the error of a given ordering by measuring the maximum size of a 'local' neighbourhood around a vertex where errors can occur:

Definition 1.3 (Ordering Error)**.** Let $\{U_i\}_{i=1}^n$ be a collection of distinct points in $[0,1]$. Let σ_{true} be the permutation induced by $(U_1, \ldots, U_n)$ and let σ_{eurt} be its reverse. Say that a permutation $\sigma \in S_n$ is *correct* for this latent collection of points if $\sigma = \sigma_{\text{true}}$ or $\sigma = \sigma_{\text{eurt}}$, i.e. it satisfies either

$$\{\sigma(i) < \sigma(j)\} \leftrightarrow \{U_i < U_j\}$$

for all i, j, or satisfies

$$\{\sigma(i) > \sigma(j)\} \leftrightarrow \{U_i < U_j\}$$

for all i, j. Say that a permutation σ has error less than $\mathfrak{D}$ if there exists a correct ordering σ_{correct} so that, for all $1 \leq i, j \leq n$ satisfying $\sigma_{\text{correct}}(i) > \sigma_{\text{correct}}(j) + \mathfrak{D}$, we also have $\sigma(i) > \sigma(j)$.

1.3. A simple assumption

In this section, we give a simple-to-state sufficient condition for our first main result, Theorem 1. This sufficient condition is satisfied for many graphons (including those of the form (1.4)), but is far from tight; we relax these conditions in the statement of Theorem 3 in Section 3.1.

We begin by restricting our attention to the simple class of *uniformly embedded* graphons:

Definition 1.4 (Uniformly embedded graphons)**.** A graphon w is *uniformly embedded* if there exists a function $f[0,1] \to [0,1]$ so that, for all x, y,

$$w(x,y) = f(|x-y|).$$

In this context, f is called the *link probability function* of w.

We then make the following technical assumption on the link function:

Assumption 1.5. *Let w be a uniformly embedded graphon with link probability function $f : [0,1] \to [0,1]$. In addition, f is decreasing, and there exist constants $0 < d < 0.5$ and $0 \leq c < f(0)$ so that $f(z) = c$ for all $z \geq d$. Finally, there exists $\alpha \in (0,1)$ so that f satisfies*

$$\inf_{s \in [0,d]} \int_0^1 f(z) f(|s-z|) dz > \alpha > \int_0^1 f\left(\left|z - \frac{1-d'}{2}\right|\right) f\left(\left|z - \frac{1+d'}{2}\right|\right) dz \tag{1.7}$$

where $d' = \min\{0.5, 2d\}$.

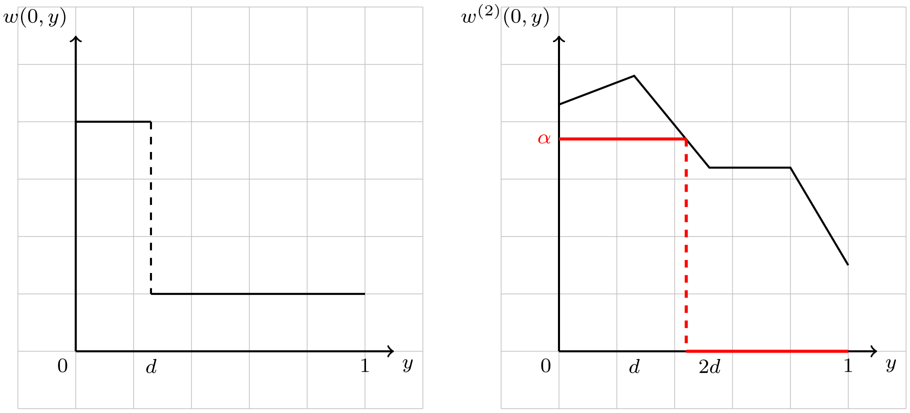

Fig 1. *This shows a sketch of $w(0,\cdot)$ and $w^{(2)}(0,\cdot)$ for a graphon of type (1.4) that satisfies Assumption 1.5. Note that $w^{(2)}$ violates the diagonally increasing condition, but $w^{(2)}_\alpha$ (shown in red) does not.*

Remark 1.6. Denote by $w^{(2)}$ the usual "square" of a graphon (see Definition 2.1). We note that the left hand side of Equation (1.7) equals $\inf_{s\in[0,d]} w^{(2)}(0,s)$, and the right hand side equals $w^{(2)}(\frac{1-d'}{2},\frac{1+d'}{2})$. Also, since $w(x,y)\geq c$ for all x,y, the lower bound on α implies that $\alpha > c^2$.

Condition (1.7) is needed to guarantee that the thresholded squared graph used in our algorithm is close to diagonally increasing. It is not generally true that the square of a diagonally increasing graphon is diagonally increasing. See Figure 1 for an illustration of a simple counterexample of type (1.4). However, even if the squared graphon itself is not diagonally increasing, we can often assure that the thresholded squared graphon is diagonally increasing for some well-chosen α. See the appendix for details on the example in Figure 1, including the choice of α.

We note that this obstacle is real (not merely an artifact of our proof technique), and other approaches to seriation have similar obstacles. For example, spectral seriation requires the Fiedler eigenvector of a matrix associated with A related to the graph to be monotone. However, the Fiedler eigenfunction associated with a graphon satisfying (1.3) is *not* always monotone - *some* additional conditions really are necessary.

Remark 1.7. Simple examples of classes of uniformly embeddded graphons for which Assumption 1.5 holds are:

(i) Any uniformly embedded graphon for which there exists a value $0<d<0.25$ so that $f(x)=0$ for all $x>d$,

(ii) Graphons of type (1.4), where $0.25\leq d<0.5$, and $d<1-p(1-3q)/(2p^2+2q^2)$.

(iii) Uniformly embedded graphons where f decreases linearly between 0 and

d from $f(0) = p$ to $f(d) = c < f(0)$, and $f(x) = c$ for all $x \geq d$, and $d \in (0.25, 0.5)$.

(iv) Uniformly embedded graphons where f is concave from 0 to $d < 0.5$ and constant from d to 1, and in addition,

$$\int_0^1 f(z)^2 dz > \int_0^1 f(|z - \frac{1}{4}|) f(|z - \frac{3}{4}|) dz.$$

We also mention one simple graphon type for which the more general Assumption 3.5 holds but which is not uniformly embedded:

(v) Graphons of the form

$$w(x,y) = \begin{cases} p & \text{if } \ell(x) \leq y \leq r(x) \\ 0 & \text{otherwise,} \end{cases}$$

where $\ell(x)$ and $r(x)$ are differentiable boundary functions that are each other's inverse, have slope bounded away from zero on the appropriate domain, and satisfy

$$0 < \inf_x (r(x) - x) \leq \sup_x (r(x) - x) < 0.25$$

Note that graphs sampled from graphons of type (v) can be interpreted as graphs where points are sampled from $[0,1]$ according to a non-uniform distribution, while links are formed according to a simple graphon of type 1.4. If this non-uniform distribution has invertible CDF F, then the associated graphon may be written as $w(x,y) = w_{unif}(F(x), F(y))$, where w_{unif} is a graphon of type 1.4.

We include these simple examples primarily as illustration. We suspect that most real-world data will be much more complicated and will not resemble graphs generated by *any* graphon with a simple formula - that is, most datasets will look quite weird and irregular in some way. We also note that most papers on "noisy" or "statistical" seriation have studied small parametric families of random graphs, but we allow a much larger (nonparametric) class of graphons. See Section 1.6.2 for further discussion.

1.4. Main results

In the next section, we will present our main new algorithm (Algorithm 1). The first of our main results is the following bound on its error:

Theorem 1 (Reconstruction for General Graphons)**.** *Fix a graphon w and constant α that satisfy Assumption 1.5. Let $G_n \sim w$ be a graph of size $n \in \mathbb{N}$. Then, when Algorithm 1 is executed with parameters as in Equation* (3.2) *on input graph G_n and value α, the output is a permutation "σ" on $\{1, 2, \ldots, n\}$ with error $\mathfrak{D}$ that satisfies*

$$\mathfrak{D} \leq \sqrt{n} \log(n)^5$$

w.e.p.

This matches (up to logarithmic terms) the rate found in [36] for the special case of graphons of the form (1.4). However, it turns out that a slightly different algorithm can give much better rates for graphons of that form, and indeed any graphons satisfying the additional assumption:

Assumption 1.8 (Sharp Boundaries)**.** *Say that a graphon w has sharp boundaries if:*

1. *There exists some $\delta > 0$ so that*
$$w(x, y) \in \{0\} \cup (\delta, 1]$$
for all $x, y \in [0, 1]$.
2. *There exists some $B > 0$ so that for all $x, y \in [0, 1]$,*
$$\mathrm{Vol}(\{z \in [0,1] \,:\, w(x,z) = 0 \neq w(y,z) \text{ or } w(y,z) = 0 \neq w(x,z)\}) \geq B|x - y|.$$

Note that this additional assumption holds for *some* examples of each type given in Remark 1.7. In particular, it holds for type (ii) when $q = 0$ and for type (iii) when $c = 0$. We don't know if this condition is close to optimal; see the discussion following Algorithm 6 for an explanation of where it is used and discussion in Section 1.6.4 for an explanation of a similar condition that turns out to be required for a closely-related problem.

We have:

Theorem 2 (Reconstruction For Graphons with Sharp Boundaries)**.** *Fix a graphon w and constant α that satisfy Assumptions 1.5 and 1.8, and let $G_n \sim w$ be a graph of size n. Fix $\epsilon > 0$. When Algorithm 6 is executed with parameters given by (4.21) on input G_n and value α, the output is a permutation "σ" on $\{1, 2, \ldots, n\}$ with error*
$$\mathfrak{D} \leq n^{\epsilon}$$
w.e.p.

Algorithm 6 consists of two parts: first a coarse ordering of a small subset of the vertices is obtained using the method from Theorem 1, then a refinement algorithm (Algorithm 7) is used to obtain an ordering of all vertices with the desired error level. These parts are independent, and the refinement algorithm could be used to improve and extend *any* coarse ordering of a subset of the vertices.

To be more precise, we will use the definition:

Definition 1.9. Fix a function $\mathfrak{F}$ which takes a finite graph as input and gives an ordering of the vertices as output. Fix a diagonally-increasing graphon w, and let $G_1, G_2, \ldots \sim w$ be a sequence of graphs on $1, 2, \ldots$ vertices. Say that the function $\mathfrak{F}$ is *efficient* for w if $\mathfrak{F}(G_n)$ has error less than $\sqrt{n}\log(n)^5$ w.e.p.

For shorthand, we write "$\mathfrak{F}$-Algorithm 6" for Algorithm 6, with step 2 replaced by the assignment:
$$\sigma_1 = \mathfrak{F}(G_1).$$

Corollary 1.10. *Fix a graphon w that satisfies Assumptions 1.8, let $\mathfrak{F}$ be efficient for w in the sense of Definition 1.9, and let $G_n \sim w$ be a graph of size n. Fix $\epsilon > 0$. When $\mathfrak{F}$-Algorithm 6 is executed with parameters given by (4.21) on input G_n and value α, the output is a permutation "σ" on $\{1, 2, \ldots, n\}$ with error*

$$\mathfrak{D} \leq n^{\epsilon}$$

w.e.p.

We can think of two situations in which Corollary 1.10 might be useful:

1. Naturally, other mathematical work may provide algorithms that are *efficient* in the sense of Assumption 1.9 (see *e.g.* [36, 33]).
2. In some situations we may use *expert knowledge*. For example, one might send a small random sample to experts who can order them by hand (likely based on features not captured by the graph). The large remainder of the data could be ordered automatically as in Algorithm 6 after Step 2. It is difficult to model expert knowledge precisely, but we know that in practice it is often very useful to be able to incorporate strong (but expensive) evidence about a (small) subsample.

Remark 1.11 (Optimistic Conjecture on Optimal Reconstruction)**.** We conjecture that the best possible reconstruction error is $O(\sqrt{n}\log(n)^C)$ in the situation of Theorem 1 and $O(\log(n)^{C'})$ in the situation of Theorem 2. However, the arguments in the present paper seem to have no hope of finding the optimal constants C, C'.

It is natural to ask when it is possible to attain a "super-small" error that is much less than $O(\sqrt{n}\log(n)^C)$. We conjecture that this can occur in many situations where Assumption 1.8 fails, but also that recovering such a super-small error would require modifications to our main refinement algorithm. We would be interested in knowing if it is possible to come close to covering all situations in which a "super-small" error occurs by using a single algorithm, but make no conjectures on this question.

Remark 1.12 (Dependence on Parameters)**.** We note that Theorems 1 and 2 have assumptions with constants c, d, δ, B that don't appear explicitly in the theorem statements. Recall that both theorems say that a certain inequality occurs *w.e.p.*- that is, it occurs for all n larger than some random integer N. The distribution of this random integer N depends on the constants c, d, δ, B. In general, N may become very large if these constants are close to the edge of their ranges (e.g. if δ is very close to 0). This dependence is essentially unavoidable. For example, when c is very close to $f(0)$, the graphon is very close to the constant graphon $w_{const}(x, y) \equiv c$. It is obviously not possible to reconstruct the ordering of vertices given observations from a constant graphon, and it is straightforward to check that one requires at least $n \gtrsim (f(0) - c)^{-0.5}$ to reliably distinguish between a graphon w and the constant graphon $w_{const}(x, y) \equiv c$.

1.5. Running time

It is straightforward to check that both algorithms have running times that are polynomial in n with parameters as stated. We have not made a serious effort to optimize the parameters for running time, and in several places we make choices in order to make our (already-long) proofs slightly shorter. With that in mind:

1. Algorithm 1 has running time that is $O(n^4 \text{polylog}(n))$, dominated by Algorithm 4. See Remark 2.2 for a quick discussion on how this may be improved to $O(n^3 \text{polylog}(n))$.
2. For the parameters appearing in Theorem 2, Algorithm 6 has running time that is $O(n^2 \text{polylog}(n))$. Furthermore, the running time is dominated by steps 3–6; these can be substantially parallelized. More generally, this $O(n^2 \text{polylog}(n))$ running time holds as long as the parameter p_1 appearing in Algorithm 6 is less than $n^{-0.5}$.

1.6. Main contributions and related work

We highlight what we view as the main contributions of this paper, in the context of the enormous literature on latent position models:

1. Many of the most-similar previous papers studied parametric families of graphons, often with few parameters (see *e.g.* [5], [36]). Often, graphs drawn from the family can be seen as slightly perturbed versions of graphs with a highly symmetric structure. The current paper studies a large non-parametric family, and importantly the algorithm does not require the user to identify a specific subfamily. This is important from a practical point of view, since simple-to-describe parametric families such as (1.4) are typically not realistic models and it is often difficult to identify a reasonable model.
   One consequence of this generality is that the conditions for our main result are longer and more complicated. This may cause our conditions to appear more technical or restrictive than previous work, even though the opposite is true. This is largely a consequence of the fact that it takes more space to write down a collection of inequalities for generic graphons than it does to write down a collection of inequalities for a small number of real-valued parameters.
2. Many previous works on latent position models proceed by first estimating latent positions, then plugging this estimate into the definition of a discrete object of interest (a permutation in the case of seriation, a community structure in the case of *e.g.* spectral clustering [26]). It is clear that estimating the full embedding is not any easier than estimating the discrete structure, since the discrete structure is a deterministic function of the true embedding. In this paper, we show that there is in fact a large gap in the difficulty of these two problems – estimating the latent positions is sometimes *much harder* than estimating the discrete structure. We hope that this observation is useful for the design of future algorithms.

In the rest of the section, we give a more detailed look at some specific parts of the literature on latent position models.

1.6.1. Consequences for efficiency of downstream tasks

We note that the problem considered in this paper has the following form: there is some "true" latent position $U_1, \ldots, U_n$ of each vertex, and we wish to estimate some explicit and reasonably nice function of these latent positions (in this case, the induced permutation as in Equation (1.1)). It is natural and very common to try to do this sort of task by the following two-step procedure: *first* finding an estimate $\hat{U}_1, \ldots, \hat{U}_n$ of the true latent positions $U_1, \ldots, U_n$, and *then* plug this estimate into the function (typical algorithms for the closely-related problem of spectral clustering are of this form; see *e.g.* the famous overview [38]).

The algorithms studied in this paper are *not* of this two-step form, and this is not an accident. In Section 6, we prove that the best estimate $\hat{U}_1, \ldots, \hat{U}_n$ of $U_1, \ldots, U_n$ will have error $\Omega(n^{-0.5} \log(n)^{-C''})$ for a broad class of models. This implies that *there is no way to recover the error bound in Theorem 2 by estimating the latent positions $\hat{U}_1, \ldots, \hat{U}_n$ and then propagating this error bound to an estimate $\hat{\sigma} = \hat{\sigma}(\hat{U}_1, \ldots, \hat{U}_n)$ of σ_{true} that depends only on these estimated latent positions.* This fact was somewhat surprising to us, as it is common to analyze the performance of algorithms on "downstream tasks" such as clustering or ordering using this sort of two-step error bound (see *e.g.* the recent paper [26], which include such a "downstream analysis" for a closely-related embedding problem). Our general observation that two-step procedures may not be optimal (even where they are common in practice) is not new - see *e.g.* [31] and the references therein.

1.6.2. Seriation

A graph for which the adjacency matrix is a Robinsonian similarity matrix is known as a *unit interval graph*. The graph seriation problem for unit interval graphs can be solved efficiently using a graph-theoretical algorithm. In particular, in [13], the author gives a linear time algorithm that returns the line-embedded permutation. The algorithm uses three sweeps of *LexBFS*, a special Breadth-First Search algorithm first proposed in [37]. Corneil's 3-sweep LexBFS algorithm is used as an important subroutine in our algorithm. The matrix seriation problem can also be solved efficiently. either with a combinatorial algorithm (see [23] for an $O(n \log n)$ algorithm) or with spectral methods, as proposed in [1].

If the matrix is not exactly Robinsonian, the seriation problem becomes intractable. Previous work on seriation-with-error considers an *optimization* problem: find a Robinson matrix and a permutation so that the ℓ_p distance between the permuted matrix and the Robinson approximation is minimized. This problem is NP-hard in general (see [3] for the case $p < \infty$ and [9] for the case $p = \infty$). In [10] the authors propose a (deterministic) approximation algorithm for the

ℓ_∞ case. If the matrix exhibits certain additional nice properties, then the seriation problem can be solved in polynomial time as a quadratic assignment problem, see [22, 6]. In [18, 25, 15], noisy seriation is presented as a convex optimization problem over a polytope related to the set of permutations. An approximate solution is reached using a relaxation of this optimization problem. Finally, [39, 19] study related problems in the special case of $\{0, 1\}$-valued matrices.

The article [16], like our work, considers a *statistical* problem where we must recover the ordering associated with some generating process. In [16], the given matrix is assumed to be a permuted version of a Robinson matrix, plus an additive error matrix of Gaussian noise. This approach falls in the general categories of shape constrained regression and latent permutation learning. In [17], the authors use spectral seriation to reconstruct a ranking from a set of pairwise comparisons.

The articles [2, 5, 14, 36] are most similar to the present work. All of them study essentially the same problem as the current paper - reconstructing an ordering based on an observed graph. The major difference is that they all consider graphons that lie in small-dimensional parametric families, all quite similar to our example (1.4). [1] In most cases, it is not clear how their algorithms or analyses could be extended to the more general setting considered in this paper. This is important in practice, since we do not expect many real-life datasets to be generated by any graphon with a simple formula such as (1.4) - real graphs are often quite messy!

The article [33] appeared after this paper and studies essentially the same problem. Like this paper, it also considers a nonparametric family of graphons. Its main result is that the popular spectral clustering algorithm gives a consistent estimate of the ordering under certain conditions. The main difference between our papers is that the approach in [33] works for a different class of graphons and achieves a slower rate of convergence.

1.6.3. Testing graphons

Our initial motivation for this problem was finding efficient tests as to whether a random graph was simulated from a graphon that could be embedded (either in a one-dimensional space as in here, or more generally). Naive approaches fall into the trap discussed in 1.6.1.

In [11] and [21], the authors construct a test of line-embeddability. Precisely, they define a function Γ^* on the space of graphs so that, for a sequence of random graphs $G_n \sim w$ sampled from a graphon w, we have $\lim_{n\to\infty} \Gamma^*(G_n) = 0$ if and only if w is a.e. diagonally increasing. Furthermore, Γ^* is obtained from a function Γ that is a *testable parameter* in the sense of [27], and thus it can be

[1] The details are somewhat different. Our points are embedded uniformly, while some of theirs are placed at positions $U_i = \frac{i}{n}$; our graphs have $\Theta(n^2)$ edges, while theirs might have fewer; etc. We suspect that these minor differences could be accounted for using the approach set forth in this paper.

efficiently estimated from small samples. Unfortunately, a problem remained: *computing* Γ^* exactly requires the computation of a vertex ordering that is "optimal," in some sense made precise in [11]. The algorithms from this paper provide an approximate solution to a very similar seriation problem, which seems to be useful in providing a rigorous and efficient method for estimating Γ^*; the details will be contained in future work.

1.6.4. Estimating graphons

Consider a graph sampled from a graphon w as in Equation (1.2) with driving randomness $\{U_i\}$, $\{U_{i,j}\}$. In the present paper, we focus on estimating the *ordering* σ for which $U_{\sigma(1)} < U_{\sigma(2)} < \ldots < U_{\sigma(n)}$. There is a larger literature on estimating the *graphon* - the full n by n matrix $\{w(U_i, U_j)\}_{i,j=1}^n$. The papers [20, 7, 29, 30, 21] all define and analyze estimators that can be used to estimate graphons (some, especially [7], also apply to much larger classes of random matrices). We summarize some main points on the relationship between these results and ours.

It is clear that good estimates of a graphon give *some* information about its ordering - if you had a perfect estimate of *e.g.* a graphon of the form (1.4), it would be trivial to recover a perfect ordering. However, this relationship is not very robust, and breaks down quickly in the presence of even small error. Due to either the weak norm [20, 7, 29, 30] or the slow convergence rate [21], existing graphon estimates can't be used to obtain good order estimates, even with unlimited computational resources.

Although our results are not equivalent, we suspect that our estimates of the *ordering* may lead to improved *graphon* estimates in some situations. The two papers with the best bounds on the convergence rates of their estimators, [20, 21], both use estimators that seem to be computationally intractable. Furthermore, both are intractable for essentially the same reason: they require the computation of an ordering of the vertices that is "optimal" in a sense that is very similar to our problem.[2] Our improved bounds on the ordering problem may lead to computationally-tractable estimators that come close to matching the efficiency of [20, 21].

Although there is no general equivalence between estimating graphons and orderings, it seems likely to us that *sufficiently good* graphon estimates could be used to obtain *consistent* order estimates, even when this "translation" is not statistically efficient. Even inefficient estimates could be used as initial steps in Algorithm 6, as discussed in and immediately before Corollary 1.10.

As one final remark, these papers indicate that the rate at which graphon estimation procedures can converge depends on the "smoothness" of the underlying graphon. This suggests that Assumption 1.8 may not be entirely an artifact of our proof technique.

[2] To be more precise: [20] asks that the vertices be partitioned into roughly $\sqrt{n}$ sets, and that these sets (rather than individual vertices) be optimally ordered. Ordering partitions is easier than ordering vertices, but in this case it doesn't seem to be *significantly* easier.

1.7. Paper guide

In Section 2, we state the main algorithms studied in this paper. In Section 3, we prove a version of Theorem 1 with weaker conditions; we defer a proof that Theorem 1 actually follows from this result until Appendix A. In Section 4 we discuss why it is possible to beat the $\tilde{O}(\sqrt{n})$ reconstruction barrier and prove Theorem 2.

In the remainder of the paper, we discuss some auxillary results that are helpful in applying and understanding our main results. In Section 5, we give simple guidelines for finding a "good" thresholding value α. In Section 6, we show that it is generally not possible to beat the $\tilde{O}(\sqrt{n})$ reconstruction barrier by using an "induced" permutation of the form (1.1).

2. Algorithms for general graphons

In this section, we state the main algorithms studied in this paper, interspersed with comments on what is "typically" happening at various important stages.

2.1. Some further notation

Much of our work will be related to taking powers of graphs and graphons, then thresholding the result:

Definition 2.1 (Powers of Graphs and Graphons)**.** For a graph G, denote by $A = A(G)$ its adjacency matrix. We will use $A^{(2)}$ to denote the square of the adjacency matrix. Entry $A^{(2)}[i,j]$ denotes the number of common neighbours of vertices i and j, i.e. the number of vertices adjacent to both i and j.

Similarly, define the "product" $w_1 \star w_2$ of two graphons w_1, w_2 by

$$(w_1 \star w_2)(x,y) = \int_u w_1(x,u)w_2(u,y)du$$

and define $w^{(2)} = w \star w$.

For any graphon w and $\alpha \in [0,1]$, define the thresholded graphon w_α by

$$w_\alpha(u,v) = \mathbf{1}_{w(u,v)\geq\alpha}.$$

For parameter $0 < \alpha < 1$ and graph $G = (V,E)$, define the "threshold-square" graph $G_\alpha^{(2)} = (V, E_\alpha)$ by

$$\{(u,v) \in E_\alpha\} \longleftrightarrow \{A^{(2)}[u,v] > \alpha\,(n-2)\}. \tag{2.1}$$

Similarly, define the thresholded graphon $w_\alpha^{(2)}$ by

$$w_\alpha^{(2)}(u,v) = \mathbf{1}_{w^{(2)}(u,v)>\alpha}.$$

Although our goal is to estimate a permutation σ, most of our calculations will be for pairwise comparison functions $F : V^2 \mapsto \{-1, 0, 1\}$. The idea is that a comparison F corresponds closely to a permutation γ if F is close to the following function F_γ:

$$F_\gamma(u,v) = \begin{cases} 1 & \text{if } \gamma(u) < \gamma(v) \\ -1 & \text{if } \gamma(u) > \gamma(v). \end{cases} \tag{2.2}$$

To go from a comparison function to a permutation, define:

$$\begin{aligned} \gamma_F(i) &= \sum_{j \in V} F_\gamma(i,j) \\ \sigma_F(i) &= |\{j \in V \,:\, \gamma_F(j) \leq \gamma_F(i)\}| \end{aligned}$$

when the values of γ_F are distinct; when there are ties, break them arbitrarily and use the above formula. We note that, for any permutation σ, we have $\sigma_{F_\sigma} = \sigma$.

We can use these formulae to move between permutations and comparison functions, and we extend notation in the obvious way. To make one frequently-used example explicit: we say that a comparison function $F : V^2 \mapsto \{-1, 0, 1\}$ has error less than $\mathfrak{D}$ if there exists a correct permutation σ_{true} so that, for all $1 \leq i, j \leq n$ satisfying $|\sigma_{\text{true}}(i) - \sigma_{\text{true}}(j)| > \mathfrak{D}$,

$$F(u,v) = 1 - 2\mathbf{1}_{\sigma_{\text{true}}(i) < \sigma_{\text{true}}(j)}.$$

We will also often define permutations on some subset $S \subset V$. We say that a permutation σ on S agrees with a permutation η on V if $F_\sigma(i,j) = F_\eta(i,j)$ for all $i, j \in S$. Abusing notation slightly, we will also say that σ agrees with η on some $D \subset S^2$ if $F_\sigma(i,j) = F_\eta(i,j)$ for all $(i,j) \in D$ (typically, D will be pairs that are sufficiently far apart in the order η).

Finally, for a vertex set $V \subset \mathbb{N}$, denote by

$$\tilde{V} = \{(u,v) \in V^2 \,:\, u < v\}$$

the list of ordered pairs of vertices. We note that a function $\tilde{F} : \tilde{V} \mapsto \{-1, 0, 1\}$ can be extended to a unique antisymmetric function $F : V^2 \mapsto \{-1, 0, 1\}$; we use this extension without comment.

2.2. Algorithms for Theorem 1

Our main algorithm, Alg. 1, uses Alg. 2 to find an initial coarse comparison F based on $G_\alpha^{(2)}$, and then "fills in" the remaining comparisons by taking larger subsamples and merging them by a voting procedure (Alg. 5). This will give a comparison F' that agrees with the line-embedded permutation σ_{true} or its reversal σ_{eurt} for all pairs sufficiently far apart, which implies that $\sigma_{F'}$ is close to either σ_{true} or σ_{eurt}.

```
Algorithm 1: Main Estimation Algorithm
parameters: Sample size $m \in \mathbb{N}$, running time $t$, threshold $\zeta$, and truncation level $\alpha$.
input: Graph $G = (V, E)$ of size $|V| = n \geq m$.
output: A full order $\sigma$ on $V$.
Compute $G_\alpha^{(2)}$ according to Equation (2.1).
Run Alg. 2 with parameters $t$, $m$, and $\zeta$. Set $F = \mathit{SparseSketch}(G_\alpha^{(2)})$.
    {With high probability, $F$ has error $\lesssim \log(n)^{-1}$.}
Run Alg. 5. Set $F' = \mathit{LocalRefinement}(G_\alpha^{(2)}, F)$.
Return $\sigma_{F'}$.
```

Most of the work is done in the function *SparseSketch* (Alg. 2). The most important step of Algorithm 2 is the repeated call to *OrderedSubsample* (Alg. 3), which samples small subgraphs of $G_\alpha^{(2)}$ and then estimates orders on these small subgraphs.

The motivation for repeated subsampling is that, with high probability, the vast majority of these subgraphs typically have a special property that $G_\alpha^{(2)}$ does not have: they will be *unit interval graphs* (see Section 3.3.1). This special property allows us to use an existing efficient algorithm for finding the line-embedded permutation, referred to here as LexBFS, that only applies to unit interval graphs.

Having used this efficient algorithm for our subgraphs, it is then necessary to stitch together their estimated orderings. The function *GlobalOrder* (Alg. 4) is needed because we expect some of the small samples to be ordered according to the true line-embedded permutation, and others according to its total reversal; they must be aligned to either all agree with the true line-embedded permutation, or all agree with its total reversal.

The number of iterations is chosen such that each pair of vertices appears together in many of the samples. After aligning all the samples, we then simply count how often each pair is ordered 'up' or 'down', and decide the final ordering by majority vote.

OrderedSubsample (Alg. 3) uses a graph algorithm to order the vertices of a small random subgraph of G.

The *LexBFS* algorithm, which quickly recognizes and orders unit interval graphs, was first described in [13]. The version of the algorithm in that paper doesn't detect if a graph is disconnected, but (since the algorithm is based on *BFS*) it can be easily modified to do so. More information about unit interval graphs and this algorithm is given below in Section 3.3.1.

GlobalOrder (Alg. 4) is used to globally align the estimated orderings for individual subgraphs.

The sets $\mathcal{L}^{(j)}$ and $\mathcal{R}^{(j)}$ capture the vertices in sample j with highest and lowest U-values, respectively. Sample size is chosen so that there is sufficient overlap between these sets between samples. If $\mathcal{L}^{(i)}$ overlaps with $\mathcal{L}^{(j)}$, then this is an indication that the samples i and j are aligned, and $H(i, j)$ is set to 1, whereas if $\mathcal{L}^{(i)}$ overlaps with $\mathcal{R}^{(j)}$, then this indicates that the samples have

Algorithm 2: *SparseSketch*

```
parameters: Stopping time $t$, size $m \in \mathbb{N}$, and alignment threshold $\zeta$.
input: "Square-thresholded" graph $H = (V, E)$ of size $|V| = n$.
output: A comparison function $F$ on $V$.
 1: Initialize the list of pairwise comparisons and number of comparisons $\mathcal{C} : \tilde{V} \mapsto \mathbb{Z}$ by
    $\mathcal{C} \equiv 0$. Also set $i = 0$.
 2: while $i < t$ do
 3:     Call Alg. 3 with parameter $m$ and input $H$. If algorithm succeeds, set
        $(\sigma, S) =$ OrderedSubsample$(H)$ and $Z = 1$; otherwise $Z = 0$.
 4:     if $Z = 1$ then
 5:         Set $i = i + 1$.
 6:         Set $(\sigma^{(i)}, S^{(i)}) = (\sigma, S)$.
 7:     end if
 8: end while
 9: Run Alg. 4 with parameter $\zeta$. Set $a =$ GlobalOrder$((\sigma^{(1)}, S^{(1)}), \ldots, (\sigma^{(t)}, S^{(t)}))$. For
    $1 \leq j \leq t$, set:
        $$\sigma^{(j)} = \begin{cases} \sigma^{(j)}, & a(j) = 1 \\ m + 1 - \sigma^{(j)}, & a(j) = -1. \end{cases}$$
10: for $j = 1, 2, \ldots, t$ do
11:     For vertices $u, v \in S^{(j)}$, update $\mathcal{C}$ by
        $$\mathcal{C}(u, v) = \begin{cases} \mathcal{C}(u, v) + 1, & \sigma^{(j)}(u) < \sigma^{(j)}(v) \\ \mathcal{C}(u, v) - 1, & \sigma^{(j)}(u) > \sigma^{(j)}(v) \end{cases}$$
12: end for
13: Return the function $F : \tilde{V} \mapsto \{-1, 0, 1\}$ given by
        $$F(u, v) = \begin{cases} -1, & \mathcal{C}(u, v) < 0, \\ 1, & \mathcal{C}(u, v) > 0, \\ 0 & \text{otherwise.} \end{cases}$$
```

Algorithm 3: *OrderedSubsample*

```
parameter: Size $m \in \mathbb{N}$.
input: "Square-thresholded" graph $H = (V, E)$ of size $|V| = n$.
output: A set $S \subset V$ of size $m$, and a permutation $\sigma$ on $S$.
 1: Sample $S \sim \mathrm{Unif}(\{T \subset V : |T| = m\})$.
 2: Apply the LexBFS algorithm to the sampled graph $H(S)$.
 3: If $H(S)$ is a connected proper interval graph, say the algorithm succeeds and return
    $S$ and a line-embedded permutation $\sigma$ of $S$.
```

opposite alignment, and $H(i, j)$ is set to -1.

When it exists, the function a can be found very quickly by a greedy search (step 9). In practical implementations, it may be better to choose an algorithm that is more robust to small errors (e.g. solve the usual convex relaxation of the problem, then round the solution to the closest value in $\{-1, 1\}$).

Remark 2.2. The cost of Algorithm 4 is dominated by the cost t^2 of constructing the t by t matrix H. We note that the choice of number of samples

Algorithm 4: *GlobalOrder*

```
parameter: Threshold $\zeta > 0$.
input: List of pairs $(\sigma^{(j)}, S^{(j)})_{j=1}^{t}$ of permutations $\sigma^{(i)}$ on sets $S^{(i)} \subset [n]$ of common size $|S^{(i)}| = m$.
output: A global alignment $a : \{1, 2, \ldots, t\} \mapsto \{-1, +1\}$.
Initialize the number of pairwise comparisons $\mathcal{N} : \tilde{V} \mapsto \{0, 1, \ldots\}$ by $\mathcal{N} \equiv 0$.
for $j \in \{1, 2, \ldots, t\}$ do
    Set
        $\mathcal{L}^{(j)} = \{u \in S^{(j)} : \sigma^{(j)}(u) < \zeta\}$
        $\mathcal{R}^{(j)} = \{u \in S^{(j)} : \sigma^{(j)}(u) > m - \zeta\}$.
end for
Initialize our estimated pairwise alignment $H : \{1, \ldots, i\}^2 \mapsto \{-1, 0, +1\}$ by $H \equiv 0$.
for $j, k \in \{1, 2, \ldots, t\}$ do
    Set
        $H(j,k) = \begin{cases} 1, & (\mathcal{L}^{(j)} \cap \mathcal{L}^{(k)}) \cup (\mathcal{R}^{(j)} \cap \mathcal{R}^{(k)}) \neq \emptyset, \text{ otherwise} \\ -1, & (\mathcal{L}^{(j)} \cap \mathcal{R}^{(k)}) \cup (\mathcal{R}^{(j)} \cap \mathcal{L}^{(k)}) \neq \emptyset, \text{ otherwise} \\ 0. & \end{cases}$
end for
Find a function $a : \{1, 2, \ldots, t\} \mapsto \{-1, 1\}$ so that, for all $i, j$ with $H(i, j) \in \{-1, 1\}$, $H(i, j) = a(i)a(j)$.
{We will show that w.e.p. such a function always exists.}
Return $a$.
```

t and their sizes m are driven by two concerns: the total number of samples mt must be at least $n^2\text{polylog}(n)$ to ensure that each pair of vertices appears together in many samples, and the size of the sample m must be small enough that the subsampled graphs have certain good properties with high probability (primarily, they must be Robinsonian).

We expect that in fact subgraphs up to size $m' = \frac{\sqrt{n}}{\text{polylog(n)}}$ would have these good properties. If true, this would allow us to replace t by $t' = O(n^{1.5}\text{polylog}(n))$, reducing the computational cost of the algorithm by a factor of n.

Finally, we describe *LocalRefinement* (Alg. 5), which refines the initial sketch provided by Algorithm 2. We use the notation $N_G(u) = \{w \in V(G) : (u, w) \in E(G)\}$ to denote the *neighbourhood* of u in G, omitting the subscript G when it is clear from context.

Algorithm 5 is based on the following observations. Recall that Algorithm 5 takes as input a "rough" ordering from some comparison F. Now consider $i, j \in V$ so that $U_i < U_j$. Then any vertex k with $U_k > U_j$ is closer to U_j than to U_i, so more likely to be a neighbour of j than of i. Thus, we expect that j will have more neighbours than i that are higher up in the ordering than j. Similarly, we expect that j will have fewer neighbours than i that are lower down in the ordering than i. Therefore, we expect to be able to order i, j by counting the number of elements of the neighbourhoods $N(i) \setminus N(j)$ and $N(j) \setminus N(i)$ that

```
Algorithm 5: LocalRefinement
input: "Square-thresholded" graph $H = (V, E)$ of size $|V| = n$, and a comparison $F$.
output: A comparison $F'$.
Initialize $F' : \tilde{V} \mapsto \{-1, 0, 1\}$ by $F' \equiv 0$.
for all $(u, v) \in \tilde{V}$ do
    Set
        $D_{u,v} = |\{x \in N(u) \setminus N(v) : F(x,v) = 1\}| - |\{x \in N(u) \setminus N(v) : F(x,v) = -1\}|$
        $D_{v,u} = |\{x \in N(v) \setminus N(u) : F(x,u) = 1\}| - |\{x \in N(v) \setminus N(u) : F(x,u) = -1\}|$
    if $|D_{u,v}| > |D_{v,u}|$ then
        Set
            $F'(u,v) = \begin{cases} 1, & D_{u,v} > 0 \\ -1, & D_{u,v} \leq 0 \end{cases}$
    else
        Set
            $F'(u,v) = \begin{cases} 1, & D_{v,u} < 0 \\ -1, & D_{v,u} \geq 0. \end{cases}$
    end if
end for
Return $F'$.
```

are higher than i, j according to the rough ordering F.

2.3. Algorithms for Theorem 2

Algorithm 1 described in the previous subsection gives an ordering with error approximately $\sqrt{n}$. Here we describe the iterative refinement algorithm, which sequentially reduces this error.

Algorithm 6 creates a coupled sequence of random graphs $G_1 \subset G_2 \subset \ldots \subset G_k = G$. Algorithm 1 is used with input G_1 to obtain an initial ordering of the vertices in G_1. Then, in each successive step, Algorithm 7 is called to replace the order F_i on G_i by the more-accurate order F_{i+1} on the larger graph G_{i+1}. The iterative step is based on G itself, not $G_\alpha^{(2)}$.

All of the work in Algorithm 6 is done in Algorithm 7, the refinement algorithm that is called in the loop. It makes use of the following consequence of Assumption 1.8: if $U_i < U_j$, then there is a region $R \subset [0, 1]$ so that any vertex with U-value in R can only be a neighbour of one of i, j. If U_i and U_j are sufficiently far apart, then R is large and will contain many vertices; these will provide the signal that indicates the true ordering of i and j. Of course, we don't know which vertices have U-values in R. But we know that the vertices that provide the signal must be at the extremes of the ordering in the neighbourhood of i or j, so we can use the ordering from the previous iteration to find these vertices. The key idea is that, in each loop, we use our current estimate of σ

```
Algorithm 6: Iterative-Improvement Estimate
  parameter: Sequence of rates $0 < p_1 < p_2 < \ldots < p_k = 1 \in [0,1]$, errors
  $d_1 > d_2 > \cdots > d_k$, truncation level $\alpha$.
  input: Graph $G = (V, E)$ of size $|V| = n$.
  output: A permutation $\sigma$ on $V$.
  Let $\{B_i\}_{i=1}^{n} \overset{i.i.d.}{\sim} \mathrm{Unif}[0,1]$.
  Run Algorithm 1 on the induced subgraph $G_1$ of $G$ with vertex set
  $V_1 = \{j \in V \,:\, B_j \leq p_1\}$, $n_1 = |V_1|$, and parameters $\alpha$ and
  $m = m(|V_1|), t = t(|V_1|)\zeta = \zeta(|V_1|)$ as in Equation (3.2). Let $\sigma_1$ be the returned
  ordering of $V_1$.
  for $i \in \{1, 2, \ldots, k-1\}$ do
      Let $G_{i+1}$ be the induced subgraph of $G$ with vertex set $V_{i+1} = \{j \in V \,:\, B_j \leq p_{i+1}\}$.
      Run Algorithm 7 with parameters $C_1 = \lceil p_i d_i n \log(n)^4 \rceil$, $C_2 = \lfloor \sqrt{d_i p_i n} \log n^6 \rfloor$,
      $C_3 = \lfloor \sqrt{d_i p_i n} \log(n)^2 \rfloor$. Let $\sigma_{i+1} = \mathit{Refine}(G_{i+1}, V_i, \sigma_i)$.
  end for
  Return $\sigma_k$.
```

to improve our estimate of R, then use our improved estimate of R to further improve our estimate of σ.

To make this precise, we introduce some definitions. Given an ordering σ of a set V, a set $S \subset V$, and a parameter $c < |S|$, we define the sets of the c elements of S with the highest and lowest rank according to σ:

$$\begin{aligned} R(S, \sigma, c) &= \{k \,:\, |\{p \in S \,:\, \sigma(k) \leq \sigma(p)\}| \leq c\}, \\ L(S, \sigma, c) &= \{k \,:\, |\{p \in S \,:\, \sigma(p) \leq \sigma(k)\}| \leq c\}. \end{aligned} \tag{2.3}$$

Using these and an estimated ordering to approximate the sets R, L, we can then estimate a new ordering of i, j by counting the number of neighbours $N(i), N(j)$ in these approximations and comparing the results. We note that this comparison, occurring in steps 4-25 of Algorithm 7, is more complicated than one might expect from this description. The additional steps allow us to "prune out" and then ignore comparisons that we are not sufficiently sure of, preventing this uncertainty from "infecting" other calculations.

3. Proof of strengthened version of Theorem 1

In this section, we will prove a slightly stronger version of Theorem 1. Since this section is fairly long and the proof does not follow the algorithm line-by-line, we give a quick guide to the subsections of this section in order. Sections 3.3 and 3.4 are particularly important, as they give the "big-picture" explanation of why the algorithm works and what needs to be checked:

3.1 We state a version of Theorem 1 that holds under weaker conditions, and state those (highly technical) conditions.

Algorithm 7: *Refine*

```
parameters: thresholds $C_1$, $C_2$, $C_3$.
input: Graph $G = (V, E)$, sets $V_1 \subset V_2 \subset V$, order $\sigma_1$ on $V_1$.
output: An order $\sigma_2$ on $V_2$.
for $i < j$ in $V_2 \setminus V_1$ do
    Set
        $R = R(i, j) = R(V_1 \cap (N(i) \cup N(j)), \sigma_1, C_1),$
        $L = L(i, j) = L(V_1 \cap (N(i) \cup N(j)), \sigma_1, C_1).$
    Initialize $F^{(2)}(i, j) = 0$.
    if $|N(j) \cap R| > |N(i) \cap R| + C_2$ or $|N(i) \cap L| > |N(j) \cap L| + C_2$ then
        Set $F^{(2)}(i, j) = 1$.
    else
        if $|N(i) \cap R| > |N(j) \cap R| + C_2$ or $|N(j) \cap L| > |N(i) \cap L| + C_2$ then
            Set $F^{(2)}(i, j) = -1$.
        end if
    end if
end for
Let $\sigma'_2$ be a permutation of $V_2 \setminus V_1$ that is compatible with $F^{(2)}$, i.e. $\sigma'_2 = \sigma_{F^{(2)}}$ as
in (2.2).
for $i$ in $V_2$ do
    Set
        $t(i) = \max\{\sigma'_2(k) \,:\, k \in N(i) \cap (V_2 \setminus V_1)\}$
        $b(i) = \min\{\sigma'_2(k) \,:\, k \in N(i) \cap (V_2 \setminus V_1)\}.$
end for
for $i \in V_1$, $j \in V_2$ do
    Initialize $F^{(2)}(i, j) = 0$.
    if $t(j) - t(i) > C_3$ or $b(j) - b(i) > C_3$ then
        Set $F^{(2)}(i, j) = 1$
    else
        if $t(i) - t(j) > C_3$ or $b(i) - b(j) > C_3$ then
            Set $F^{(2)}(i, j) = -1$
        end if
    end if
end for
Return a permutation $\sigma_2 = \sigma_{F^{(2)}}$ of $V_2$ that extends $F^{(2)}$.
```

3.2 We describe the probability space that will be used for the remainder of the proof.

3.3 We give a detailed description of several key properties of “typical” realizations of the random graph G, as well as key properties of several derived objects that appear in Algorithm 1, and explain why those properties ensure that Algorithm 1 works. This section also includes technical lemmas showing that these key properties occur with high probability.

3.4 We give a sketch of the most important parts of our proof, and how it depends on the typical properties described in the previous section.

3.5 With the preliminaries out of the way, we analyze the output of Algorithm 3, the sub-algorithm that orders small subgraphs of G.

3.6 We analyze the output of Algorithm 4, the sub-algorithm that aligns the orderings of the small subgraphs returned by Algorithm 3 (this alignment is necessary because even "good" outputs of 3 will be randomly aligned with either σ_{true} or σ_{eurt}).

3.7 We analyze the output of Algorithm 2, the sub-algorithm that calls Algorithms 3 and 4 to obtain an estimated order of the full graph G, and give bounds on the error that hold *whp* (though these bounds may still be quite large).

3.8 We complete the analysis of the remaining sub-algorithm of Algorithm 1, which refines the estimate obtained in Algorithm 2. This analysis shows that *whp* the refined ordering has error bounded as stated in Theorem 3. This completes the proof of Theorem 3.

3.1. A stronger result

Rather than proving Theorem 1 directly, we prove a related result with weaker (but more complicated) assumptions. This theorem applies to graphons in general, not just graphons with a uniform embedding. We verify that the stronger assumptions of Theorem 1 imply the more general assumptions presented in this section in Appendix A. While weakening the assumptions in this way will provide a stronger final result, our main motivation was to replace the assumptions of Theorem 1 (which are easy to read but hard to work with directly) with the weaker assumptions of Theorem 3 (which are harder to read but easier to work with in our proof).

Before we discuss the weaker assumptions, we give an alternative characterization of $\{0,1\}$-valued graphons. It is not hard to verify (see also the discussion following Definition 2.1 of [12]) that, if w is a $\{0,1\}$-valued graphon, then w is diagonally increasing if and only if there exist two non-decreasing functions $\ell : [0,1] \to [0,1]$ and $r : [0,1] \to [0,1]$ which demarkate the regions in $[0,1]^2$ where w assumes the value 1. The functions ℓ and r will be called the *boundaries* of w. Precisely, w is of the form

$$w(x,y) = \begin{cases} 1 & \text{if } y \in I(x) \\ 0 & \text{otherwise,} \end{cases}$$

where $(\ell(x), r(x)) \subset I(x) \subset [\ell(x), r(x)]$. Thus, the boundaries define w up to the values exactly on points of the form $(\ell(x), x)$ and $(x, r(x))$; of course this collection of points has measure 0, and thus does not influence the distribution of samples from w.

In this section, which leads up to the proof of Theorem 3, we will apply this characterization in particular to $w_\alpha^{(2)}$, which is almost everywhere defined by the boundaries r_α and ℓ_α defined as follows:

$$\begin{aligned} r_\alpha(x) &= \sup\{y \in [0,1] \,:\, w^{(2)}(x,y) \geq \alpha\}, \\ \ell_\alpha(x) &= \inf\{y \in [0,1] \,:\, w^{(2)}(x,y) \geq \alpha\}. \end{aligned} \tag{3.1}$$

We now state the definitions required for our weaker assumptions. For a set $S \subset \mathbb{R}^d$, denote by $\mathrm{Vol}(S)$ the d-dimensional volume S. Then define:

Definition 3.1 (Goodness)**.** Fix $0 < \alpha, \delta < 1$ and $0 < A < \infty$. Say that w is *uniformly* (A, δ)*-good at* α if it satisfies

$$\sup_{x \in [0,1]} \mathrm{Vol}(\{y \in [0,1] \,:\, |\alpha - w(x,y)| \leq \delta'\}) \leq A\delta'$$

for all $0 < \delta' < \delta$.

In other words: if w is uniformly (A, δ)-good at α, then the values of $w(x, \cdot)$ don't concentrate around α. We will not apply this assumption to w directly; we apply it to $w^{(2)}$. Next, we use a simple notion of "connectedness" for graphons:

Definition 3.2 (Connectedness of Graphons)**.** Fix $0 < \epsilon < 1$. We say that a graphon w is (ϵ, α)-connected if

$$\inf\{w(u,v) \,:\, u, v \in [0,1],\, |u - v| \leq \epsilon\} > \alpha.$$

For graphons w, to be (ϵ, α)-connected means that the region where w achieves values greater than α contains a strip of width ϵ around the diagonal. We will see later that this implies that, asymptotically almost surely, large graphs sampled from the α-thresholded version of such a graphon are connected.

The following "separation" property implies that the neighbourhoods of far-apart vertices are easy to distinguish:

Definition 3.3 (Separation of Graphons)**.** Fix $0 < \epsilon < \infty$. We say that a $\{0,1\}$-valued graphon w is ϵ-separated if, for all x, y,

$$\mathrm{Vol}(\{z \in [0,1] \,:\, |w(x,z) - w(y,z)| = 1\}) \geq \epsilon |x - y|.$$

Diagonally increasing graphs may have many indistinguishable rows - as an extreme example, the complete graph is Robinsonian! The following assumption rules out this sort of near-complete indistinguishability by requiring very different points to have disjoint neighbourhoods:

Definition 3.4 (Splitting graphons)**.** Fix $0 < \epsilon < 1$. We say that a $\{0,1\}$-valued graphon w has an ϵ-split if for all $x \leq y - (1 - \epsilon)$,

$$\{z \in [0,1] \,:\, w(x,z) = w(y,z) = 1\} = \emptyset.$$

In graphs sampled from a graphon with an ϵ-split, vertices with U_i-values that are far apart cannot have any common neighbours. This guarantees that a certain "line-embeddedness" is preserved in the graph and its samples, and a correct permutation can be extracted from the samples.

We combine these properties:

Assumptions 3.5 (Weak Assumptions)**.** *Let w be a graphon, and $\alpha, \epsilon \in (0,1)$ and $A > 0$ be constants, so that:*

1. *$w_\alpha^{(2)}$ is diagonally increasing, and*

2. $w^{(2)}$ *is uniformly* (A, ϵ)*-good at* α*, and*
3. $w^{(2)}$ *is* (ϵ, α)*-connected, and*
4. $w_\alpha^{(2)}$ *has an* ϵ*-split, and*
5. $w_\alpha^{(2)}$ *is* ϵ*-separated.*

Note that most of these assumptions are properties of $w_\alpha^{(2)}$, not $w^{(2)}$. In particular, we do not need that $w^{(2)}$ is diagonally increasing, only that the thresholded $\{0,1\}$-valued graphon $w_\alpha^{(2)}$ is. This is an important distinction: it is not true, in general, that the square of a diagonally increasing graphon is itself also diagonally increasing (see Appendix A, Remark A.8 for a simple example). Even if $w^{(2)}$ is itself not diagonally increasing, there still may be a value of α for which $w_\alpha^{(2)}$ is diagonally increasing.

Finally, we define the set of parameters for our main algorithm that gives the desired result for Theorem 1.

$$\begin{aligned} m = m(n) &\equiv \log(n)^5, \\ t = t(n) &\equiv (n\log(n))^2, \\ \zeta = \zeta(n) &\equiv 4\log(n)^4 \end{aligned} \tag{3.2}$$

We are now ready to state the stronger theorem.

Theorem 3 (Ordering on Large Sample)**.** *Fix a graphon* w *and constant* α *that satisfy Assumption 3.5. Let* $G_n \sim w$ *be a graph of size* $n \in \mathbb{N}$*. Then when Algorithm 1 is executed with parameters as in* (3.2) *on input graph* G_n *and value* α*, then the output is an ordering "*σ*" on* $\{1, 2, \ldots, n\}$ *with error that satisfies*

$$\mathfrak{D} \leq \sqrt{n}\log(n)^5$$

w.e.p.

3.2. Construction of probability spaces

We set some notation that will be used throughout the remainder of Section 3. Fix notation as in the statement of Theorem 3. Fix two i.i.d. sequences $\{U_i\}_{i\in\mathbb{N}}$, $\{U_{i,j}\}_{i<j\in\mathbb{N}}$ with uniform distribution on $[0,1]$. We will use these two sequences to couple all of the graphs described in this section as follows: for *any* graph of size $n \in \mathbb{N}$ sampled from a graphon in this section, we will *always* assume that the graph is sampled by using the elements of *this fixed* pair of sequences with indices $1 \leq i, j \leq n$, in the representation (1.2). In particular, if $G, G' \sim w$ are graphs of sizes $n < n'$, then this coupling gives a natural embedding $G \subset G'$ of the graphs. For fixed n, we denote by σ_{true} the line-embedded permutation associated with the sequence $\{U_i\}_{i=1}^n$, and denote by F_{true} the associated comparison function. Permutation σ_{eurt} is the total reversal of σ_{true}.

Denote by $H_\alpha \sim w_\alpha^{(2)}$ a "correct" thresholded graph, drawn from $w_\alpha^{(2)}$ (the thresholded square graphon as in Definition 2.1) using the same random variables $\{U_i\}$, $\{U_{i,j}\}$ as G. (Since $w_\alpha^{(2)}$ is a $\{0,1\}$-valued graphon, the variables $\{U_{i,j}\}$ are irrelevant.) Thus, for all i, j,

$$H_\alpha(i,j) = w_\alpha^{(2)}(U_i, U_j).$$

Recall the definition of the threshold-square graph $G_\alpha^{(2)}$ from Equation (2.1). For $S \subset \{1, 2, \ldots, n\}$, denote by $G(S)$ (respectively $G_\alpha^{(2)}(S)$, $H_\alpha(S)$) the induced subgraph of G (respectively $G_\alpha^{(2)}$, H_α) on vertex set S.

We define $\mathcal{F}_1(n)$ (respectively $\mathcal{F}_2(n)$) to be the σ-algebra generated by the sequence $\{U_i\}_{i=1}^n$ (respectively $\{U_{i,j}\}_{1 \le i < j \le n}$). We define $\mathcal{F}_k \equiv \mathcal{F}_k(\infty)$ for $k \in \{1, 2\}$. Finally, we set $\mathcal{F} = \mathcal{F}_1 \cap \mathcal{F}_2$ to be the σ-algebra generated by all of these random variables.

3.3. Proof sketch and main estimates for bad sets

In this section, we will show that the graphs $G, G_\alpha^{(2)}$ resulting from the random draw $\{U_i\}$, $\{U_{i,j}\}$ will have certain good properties with extreme probability, and sketch out how these properties are used to guarantee that the algorithm succeeds.

3.3.1. Unit interval graphs and the LexBFS Algorithm

The heart of Algorithm 3 is the 3-sweep LexBFS unit interval graph recognition algorithm of [13]. We recall the basic properties of unit interval graphs and this algorithm that will be used. A graph G is a unit interval graph if and only if there exists a total order of the vertices so that the adjacency matrix A of G with respect to that order is diagonally increasing (where we set the diagonal elements $A_{i,i} = 1$). We will refer to such orders as *interval orders*. If the graph is connected, then the interval order is unique, up to its total reversal and to permutation of any duplicated neighbourhoods. More precisely, for a total order of the vertices represented by a permutation σ, let A^σ be the matrix defined as:

$$A_{i,j}^\sigma = A_{\sigma^{-1}(i),\sigma^{-1}(j)}.$$

Then σ is an interval order if and only if A^σ is diagonally increasing. If G is connected, then A is irreducible. In that case, if σ and τ are both interval orders, then either $A^\tau = A^\sigma$ (which allows for the possibility that some rows are identical), or $A^\tau = A^{\sigma^{trev}}$, where σ^{trev} is the total reversal of σ.

3.3.2. Properties of latent variables

Algorithm 1 is based on the assumption that the square thresholded graph $G_\alpha^{(2)}$ has similar properties to H_α, and that the vertices in V are fairly uniformly spread over the interval $[0,1]$. This, together with our assumptions on $w^{(2)}$ and $w_\alpha^{(2)}$ will then guarantee that small samples from $G_\alpha^{(2)}$ have similar good properties. This will imply that, with high probability, the "sketch" returned by Algorithm 2 has no incorrect comparisons. In particular, we want to show that the following properties hold w.e.p.:

1. For vertices $i,j \in V$ with $|U_i - U_j| \gg \frac{1}{\sqrt{n}}$, we have $G_\alpha^{(2)}(i,j) = H_\alpha(i,j)$. That is, on a coarse scale, $G_\alpha^{(2)}$ looks like it was sampled from $w_\alpha^{(2)}$.
2. Any random uniform subsample $S \subset V$ that is independent of $\mathcal{F}$ and has $|S| \geq \log(n)^5$ will induce a connected graph $G_\alpha^{(2)}(S)$.
3. For all vertices $i,j \in V$ with $|U_i - U_j| \gg \frac{1}{\sqrt{n}}$, there are many vertices that are connected to i but not j in $G_\alpha^{(2)}$ (and vice-versa).
4. Pointwise, the empirical CDF of the collection $\{U_i\}_{i=1}^n$ is never too far from its expected value. Specifically, for all vertices $i,j \in V$ with $|U_i - U_j| \gg \frac{1}{\sqrt{n}}$, we have $|\{k \in V \,:\, U_k \in (U_i, U_j)\}| \approx n\,|U_i - U_j|$.

Note that these are all purely properties of $\{U_i\}$, $\{U_{i,j}\}$ (even property (2), which is about *typical draws of S after observing the σ-algebra $\mathcal{F}$*). When these conditions hold, we know that when i,j appear in the same subsample S of size $|S| = \log(n)^5$, the following will all occur with probability quite a bit larger than $\frac{1}{2}$:

(i) The neighbourhood of i,j in $G_\alpha^{(2)}(S)$ will agree with their "correct" neighbourhood in $H_\alpha(S)$.

(ii) The subgraph $G_\alpha^{(2)}(S)$ will be connected.

(iii) If U_i and U_j are significantly far apart, the neighbourhoods of i,j can be distinguished in $G_\alpha^{(2)}(S)$, and so in particular it is possible to compare them locally.

When *these* three events all occur for a given $i,j \in S$, we will have that $G_\alpha^{(2)}(S)$ is a connected unit interval graph whose interval orderings agree with one of the correct line-embedded permutations of $w_\alpha^{(2)}(S)$ for all vertices that are sufficiently far apart. The interval ordering can be retrieved with a standard, efficient graph-theoretic algorithm as discussed in Section 3.3.1; moreover, the algorithm will detect when $G_\alpha^{(2)}(S)$ is not a connected unit interval graph. Thus, when *(i)-(iii)* occur, we will be able to reconstruct the correct ordering of sufficiently distant pairs $i,j \in S$.

A precise statement of Conditions (1)-(4) above will be given in Definition 3.6, and Conditions (i)-(iii) will be given in Definition 3.8 in the following subsection. First we introduce the necessary notation. For fixed n and α, define the "bad set"

$$\mathcal{B}(n,\alpha) = \left\{ (x,y) \in [0,1]^2 \,:\, |w^{(2)}(x,y) - \alpha| \leq \frac{\log(n)}{\sqrt{n}} \right\}. \tag{3.3}$$

Thus, $\mathcal{B}(n,\alpha)$ is the subset of $[0,1]^2$ where $w^{(2)}$ takes value very close to α.

We define the related "bad witnesses" of a vertex $1 \leq i \leq n$ by

$$\mathcal{B}(n, i, \alpha) = \{j \in \{1, 2, \ldots, n\} \backslash \{i\} \,:\, G_\alpha^{(2)}(i, j) \neq w_\alpha^{(2)}(U_i, U_j)\}. \tag{3.4}$$

We will see that *w.e.p.* $j \notin \mathcal{B}(n, i, \alpha)$ if $(U_i, U_j) \notin \mathcal{B}(n, \alpha)$ (see Lemma B.3).

We also define the collection of "good witnesses" for a *subset* $S \subset \{1, 2, \ldots, n\}$ and *pair* of vertices $i, j \in S$ by:

$$\begin{aligned} \mathcal{W}_S(i, j) &= \{k \in S \,:\, w_\alpha^{(2)}(U_i, U_k) \neq w_\alpha^{(2)}(U_j, U_k)\} \\ \nu_S(i, j) &= |\mathcal{W}_S(i, j)|. \end{aligned}$$

When $S = \{1, 2, \ldots, n\}$, we will drop the subscript. These are the vertices that will allow us to distinguish i, j.

Finally, we define a cover of $[0, 1]$ which will be useful in showing that subgraphs of $G_\alpha^{(2)}$ are connected. Let $\epsilon > 0$ be as in Assumption 3.5. Fix $R \in \mathbb{N} \cap [\epsilon^{-1}, 2\epsilon^{-1}]$ and define the sets

$$I_k = \left[\frac{k}{4R}, \frac{(k+3)}{4R}\right] \cap [0, 1], \quad 0 \leq k \leq 4R - 3. \tag{3.5}$$

Note that these intervals form a cover of $[0, 1]$. Moreover, if $U_i \in I_k$ and $U_j \in I_k \cup I_{k+1}$, then $H_\alpha(i, j) = 1$. This implies that, for any set $S \subset V$,

$$\bigcap_{k=0}^{4R-3} \{S \cap \{i : U_i \in I_k\} \neq \emptyset\} \subset \{H_\alpha \text{ is connected}\}. \tag{3.6}$$

We now give formal definitions of the collection of bad events:

Definition 3.6 (Bad Events)**.** Let graphon w, value α, random variables $\{U_i\}$, $\{U_{i,j}\}$, and graphs $G_\alpha^{(2)}$ and H_α be as defined above. Moreover, let R, $\{I_k\}_{k=0}^{4R-3}$ be as in (3.5). We define:

$$\begin{aligned} \mathcal{A}_1 &= \cup_{1 \leq i < j \leq n} (\{(U_i, U_j) \notin \mathcal{B}(n, \alpha)\} \cap \{G_\alpha^{(2)}(i, j) \neq H_\alpha(i, j)\}) \\ \mathcal{A}_2 &= \left\{\max_{1 \leq i \leq n} |\mathcal{B}(n, i, \alpha)| > \sqrt{n}\, \log(n)^2\right\} \\ \mathcal{A}_3 &= \left\{\min_{0 \leq k \leq 4R-3} |\{i \,:\, U_i \in I_k\}| < \frac{n}{2R}\right\} \\ \mathcal{A}_4 &= \left\{\min_{1 \leq i,j \leq n \,:\, |U_i - U_j| > \frac{1}{\log(n)}} \nu(i, j) < \frac{2n}{\log(n)^2}\right\} \\ &\quad \cup \left\{\min_{1 \leq i,j \leq n \,:\, |U_i - U_j| > \frac{\log(n)^4}{\sqrt{n}}} \nu(i, j) < \sqrt{n} \log(n)^3\right\} \\ \mathcal{A}_5 &= \left\{\max_{1 \leq i < j \leq n} |\{k \,:\, U_k \in (U_i, U_j)\}| \geq n|U_i - U_j| + \sqrt{n} \log(n)\right\}. \end{aligned}$$

The event $\mathcal{A}_1^c \cap \mathcal{A}_2^c$ implies property (1) above, while $\mathcal{A}_3^c$, $\mathcal{A}_4^c$ and $\mathcal{A}_5^c$ imply properties (2), (3) and (4), respectively. The bad events mostly involve random variables that are the sum of the independent variables $\{U_i\}$,$\{U_{i,j}\}$, and by straightforward application of standard concentration inequalities we can show that *w.e.p.* none of these bad events occur. The lemmas and proofs establishing this fact can be found in the Appendix. Here, we only state their corollary:

Corollary 3.7. *Let $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4 \cup \mathcal{A}_5$. Then $\mathcal{A}^c$ holds w.e.p.*

Proof. This follows from Lemmas B.3, B.4, B.5, B.6, and B.7. □

We can therefore assume that our graph G and the derived graph $G_\alpha^{(2)}$ have the properties we need for the algorithm to succeed.

3.3.3. Properties of subsampled graphs

In Algorithm 2, small subgraphs are sampled repeatedly from the square thresholded graph $G_\alpha^{(2)}$. In this section, we show that, if $\mathcal{A}^c$ holds and thus $G_\alpha^{(2)}$ has properties as expected, with high probability these samples will have similar good properties. To this end, we define bad events for the sample, and will then proceed to bound the probability that they occur.

Let S be a uniformly chosen subset of V, of size $m = \log(n)^5$. We denote by

$$S'' = \{i \in S \,:\, \forall j \in S,\, i \notin \mathcal{B}(n, j, \alpha)\} \tag{3.7}$$

the collection of elements with no "bad" comparisons.

Choosing indices to roughly match the "global" bad events $\mathcal{A}_1, \ldots, \mathcal{A}_5$, we define:

Definition 3.8 (Bad Events for Samples)**.** Let graphon w, value α, random variables $\{U_i\}$, $\{U_{i,j}\}$, and graphs $G_\alpha^{(2)}$ and H_α be as defined above. Moreover, let R, $\{I_k\}_{k=0}^{4R-3}$ be as in (3.5). Finally, we let $S \subset V$ and $m = |S|$.

$$\begin{aligned}
\mathcal{A}_1' &= \{G_\alpha^{(2)}(S) \neq H_\alpha(S)\}\\
\mathcal{A}_3' &= \left\{\min_{0 \le k \le 4R-3} |\{i \in S \,:\, U_i \in I_k\}| < \frac{m}{4R}\right\} \cup \{|S \backslash S''| > \log(n)^3\}\\
\mathcal{A}_4' &= \left\{\min_{i,j \in S \,:\, |U_i - U_j| > \frac{1}{\log(n)}} \nu_S(i,j) < \log(n)^3\right\}\\
\mathcal{A}_5' &= \left\{\max_{i,j \in S} |\{k \,:\, U_k \in (U_i, U_j)\}| \ge m|U_i - U_j| + 2\log(n)^{3.5}\right\}.
\end{aligned}$$

We think of these events as functions of S, viewing the variables $\{U_i\}$ and $\{U_{i,j}\}$ that determine the graph G as "fixed." We have seen that $\mathcal{A}^c$ holds *w.e.p.* in Corollary 3.7. We will see here that events $\mathcal{A}_i'$ are also all unlikely. The proofs are again simple, and are deferred to Appendix B.2. We need slightly more detailed results (*e.g.* that these events remain rare *even conditional on* the occurrence of certain vertices), so we keep the lemma statements here.

Lemma 3.9. *Let S be a uniformly chosen subset of V, of size $m = \log(n)^5$. Then on the $\mathcal{F}$-measurable event $\mathcal{A}^c$,*

$$\mathbb{P}[\mathcal{A}_1' \,|\, \mathcal{F}] = O(n^{-0.49}).$$

Furthermore, for any fixed $p, q \in V(G)$ satisfying $(U_p, U_q) \notin \mathcal{B}(n, \alpha)$, we have

$$\frac{\mathbb{P}[\mathcal{A}_1' \cap \{p, q \in S\} \,|\, \mathcal{F}]}{\mathbb{P}[\{p, q \in S\} | \mathcal{F}]} = O(n^{-0.49}),$$

where again both probabilities are on $\mathcal{A}^c$.

For $0 \le k \le 4R - 3$, let $V_k = \{i \,:\, U_i \in I_k\}$.

Lemma 3.10. *Let S be a uniformly chosen subset of V, of size $m = \log(n)^5$, and let S'' be as defined in (3.7). We have $\mathbb{P}[\mathcal{A}_3' \,|\, \mathcal{F}] = n^{-\Omega(\log(n))}$ on the $\mathcal{F}$-measurable event $\mathcal{A}^c$. That is, w.e.p. for all $0 \le k \le 4R-3$, both $|S \cap V_k| \ge \frac{\log(n)^5}{4R}$ and also $|S \setminus S''| \le \log(n)^3$ on $\mathcal{A}^c$. Moreover, for any fixed $p, q \in V(G)$,*

$$\frac{\mathbb{P}[\mathcal{A}_3' \cap \{p, q \in S\} \,|\, \mathcal{F}]}{\mathbb{P}[\{p, q \in S\} | \, \mathcal{F}]} = O(n^{-\Omega(\log(n))}),$$

where again both probabilites are on the event $\mathcal{A}^c$.

We see that a number of other bad events are "almost" contained in $\mathcal{A}_3'$.

Lemma 3.11. *Let S be a uniformly chosen subset of V, of size $m = \log(n)^5$, and let S'' as defined in (3.7). For each $i \in S$, let $D_S(i) = |\{j \in S : G_\alpha^{(2)}(i, j) = 1\}|$ be the number of neighbours of i in $G_\alpha^{(2)}(S)$. Then for all n sufficiently large,*

$$\{H_\alpha(S) \text{ is not connected}\} \cup \{H_\alpha(S'') \text{ is not connected}\} \cup \left\{\min_{i \in S} D_S(i) < \frac{m}{8R}\right\} \subset \mathcal{A}_3'.$$

The next lemma shows that *w.e.p.*, for all pairs of vertices i, j that are sufficiently far apart, we can distinguish i and j since they have distinct neighbourhoods in $G_\alpha^{(2)}$.

Lemma 3.12. *Let S be a uniformly chosen subset of V, of size $m = \log(n)^5$. Then*

$$\mathbb{P}[\mathcal{A}_4' \,|\, \mathcal{F}] = n^{-\Omega(\log n)}$$

on the $\mathcal{F}$-measurable event $\mathcal{A}^c$. That is, w.e.p. for all $i, j \in S$ so that $|U_i - U_j| > \frac{1}{\log(n)}$, $\nu_S(i, j) \ge \log(n)^3$.

Similarly, it follows directly from $\mathcal{A}_5^c$ that the number of vertices between i, j in a uniformly chosen subset S' is roughly proportional to $|U_i - U_j|$:

Lemma 3.13. *On the $\mathcal{F}$-measurable event $\mathcal{A}^c$,*

$$\mathbb{P}[\mathcal{A}_5' \,|\, \mathcal{F}] = n^{-\Omega(\log n)}.$$

That is, w.e.p. for all $i, j \in S$, $|\{k \,:\, U_k \in (U_i, U_j)\}| \le m|U_i - U_j| + 2\log(n)^{3.5}$.

3.4. Proof sketch

In the next sections, we will prove the correctness of all the Algorithms that are used to prove Theorem 3. The proofs involve four major ideas, which we outline here. We will assume that $\mathcal{A}^c$ holds, so G and $G_\alpha^{(2)}$ have good properties.

- **Most of the small samples can be correctly ordered.** With probability $1 - o(1)$, the random subgraphs of size $m = \log(n)^5$ sampled in Algorithm 2 have certain good properties (*e.g.* they are connected, Robinsonian, and have many vertices with distinct neighbourhoods); when this occurs, Algorithm 2 will return an ordering that agrees with σ_{true} or σ_{eurt} for all vertex pairs sufficiently far apart.
- **The ordered small samples can be aligned.** At this point, we have many subsamples that each *individually* agree with either σ_{true} or σ_{eurt}. The next step is to "align" these subsamples by reversing some of them, so that either the *entire collection* agrees with σ_{true}, or the *entire collection* agrees with σ_{eurt}. The main observation is that it is easy to align a *particular pair* of samples if they share vertices with latent position close to either 0 or 1; it turns out to be possible to align all samples by greedily aligning pairs in this way.
- **Aligned sketches give a coarse ordering.** Once the small samples are aligned, say with σ_{true}, then a majority vote is used in Algorithm 2 to determine a *sketch*, or coarse ordering, of all the vertices, represented by a comparison F. The number of small samples taken is such that *w.e.p.* each pair of vertices is sampled many times, and in the majority of the samples containing this pair, the order agrees with σ_{true}. Thus, pairs of vertices that are sufficiently far apart will be ordered according to σ_{true} in the vast majority of the samples in which they occur.
- **Coarse ordering is refined using good witnesses.** Finally, in Algorithm 5, the comparison F returned by Alg. 2, is refined. This refinement is based on the *good witnesses* of the vertices. Since $w_\alpha^{(2)}$ is diagonally increasing, and $G_\alpha^{(2)}$ is very similar to a sample from $w_\alpha^{(2)}$, we can use these good witnesses to infer the ordering of two vertices s, t. Namely, if $\sigma_{\text{true}}(s) < \sigma_{\text{true}}(t)$, i.e. if s comes before t in the true ordering, then t will have more neighbours than s higher up in the ordering, while s will have more neighbours lower in the true ordering. Thus, by considering the vertices that are neighbours of s but not of t, or vice versa, and using the coarse ordering as a proxy for the true ordering, we can infer the correct ordering of s and t.

We note that the organization of this description doesn't line up exactly with the organization of Algorithm 1. This is somewhat unavoidable – one step of the proof may apply to many different parts of the algorithm, and the length of a part of the algorithm does not correspond closely to its importance.

3.5. Analysis of Algorithm 3

In Algorithm 3, a relatively small subgraph S of size $m = \log(n)^5$ is sampled from $G^{(2)}_\alpha$. Algorithm 3 applies a graph algorithm to determine whether the sample is a unit interval graph. If the algorithm succeeds, it will return an interval ordering. We prove in this section that, if no bad events occur, then with high probability the algorithm will succeed and give an ordering which agrees with one of the two correct line-embedded permutations of the vertices.

Throughout this section, we will assume that $\mathcal{A}^c$ holds, and study the properties of the subset S on this event. We will not condition on $(\mathcal{A}'_3 \cup \mathcal{A}'_4 \cup \mathcal{A}'_5)^c$, but we recall here that, by the lemmas in Section 3.3.3, this event holds *w.e.p.*. Since this holds *w.e.p.*, we will be able to use the following properties of S when required in our proofs, incurring only a negligible penalty in our probability estimates each time:

$$\begin{aligned}
&|S \setminus S''| \leq \log(n)^3, \text{ where } S'' \text{ as defined in (3.7)}, \\
&H_\alpha(S) \text{ and } H_\alpha(S'') \text{ are connected}, \\
&|\{j \in S : G^{(2)}_\alpha(i,j) = 1\}| \geq \frac{m}{8R} \text{ for all } i \in S, \\
&G^{(2)}_\alpha(S'') = H_\alpha(S'') \\
&\nu_S(i,j) \geq \log(n)^3 \text{ for all } i,j \in S \text{ such that } |U_i - U_j| > \frac{1}{\log(n)}, \\
&|\{k \,:\, U_i < U_k < U_j\}| < m|U_i - U_j| + 2\log(n)^{3.5} \text{ for all } i,j \in S.
\end{aligned} \tag{3.8}$$

Note that the second and third property above follow from the event $(\mathcal{A}'_3)^c$, together with Lemma 3.11. The fourth property follows from our assumption that we are on $\mathcal{A}^c$: If $\mathcal{A}^c_1$ holds, then this implies that discrepancies between $G^{(2)}_\alpha(S)$ and $H_\alpha(S)$ can only occur between vertices i,j where $i \in \mathcal{B}(n,j,\alpha)$, and, by definition, S'' does not include any such pairs.

We do not assume that we are on $(\mathcal{A}'_1)^c$, and do *not* use this event freely in the following calculations. We do this because Lemma 3.9 only implies that $(\mathcal{A}'_1)^c$ occurs with probability going to 1, but does not show it holds *w.e.p.* Algorithm 3 is called many times in Algorithm 2, and it may indeed happen that $\mathcal{A}'_1$ occurs for a small number of sketches.

We will show that the ordering returned by Algorithm 3 agrees with σ_{true} or σ_{eurt} for all pairs of vertices that are sufficiently far apart, and hence we define:

Definition 3.14. Given $S \subset V(G)$, a permutation σ of S, and a permutation τ of V, we say that σ *agrees with* τ *to precision level* d, if, for all $i,j \in S$ so that $|U_i - U_j| \geq d$, $\sigma(i) < \sigma(j)$ if and only if $\tau(i) < \tau(j)$.

Lemma 3.15. *Let* $G^{(2)}_\alpha$ *be as defined in Equation 2.1, and assume* $\mathcal{A}^c$ *holds. Let* $m = \log(n)^5$*, and let* $S \sim Unif(\{T \subset V \,:\, |T| = m\})$*. Let* $\mathcal{B}'_1$ *be the event that Algorithm 3 with input* $G^{(2)}_\alpha$ *and parameter* m *succeeds,* and furthermore *returns* *a total order σ of the sampled set S which agrees at precision level $\log(n)^{-1}$ with σ_{true} or with σ_{eurt}. Then*

$$\mathbb{P}[\mathcal{B}_1' \,|\, \mathcal{F}] = 1 - O(n^{-0.49})$$

on the $\mathcal{F}$-measurable event $\mathcal{A}^c$. Moreover, for $1 \leq p, q \leq n$ so that $|U_p - U_q| \geq \log(n)^{-1}$,

$$\frac{\mathbb{P}[\mathcal{B}_1' \cap \{p, q \in S\} | \mathcal{F}]}{\mathbb{P}[\{p, q \in S\} \,|\, \mathcal{F}]} = 1 - O(n^{-0.49}),$$

again on $\mathcal{A}^c$.

Proof. Algorithm 3 starts by sampling $S \sim Unif(\{T \subset V \,:\, |T| = m\})$. By (3.8), we have that $G_\alpha^{(2)}(S) = H_\alpha(S)$ is connected *w.e.p.*, and thus $G_\alpha^{(2)}(S)$ is a connected unit interval graph. On this event, Algorithm 3 does not fail, and returns an interval ordering σ of $G_\alpha^{(2)}(S) = H_\alpha(S)$. Since $H_\alpha(S)$ is a subgraph of the interval graph H_α, the interval ordering of $H_\alpha(S)$ agrees with σ_{true} or σ_{eurt}; assume *wlog* that it agrees with σ_{true}. This means that σ agrees with σ_{true}, except possibly for pairs of vertices with identical neighbourhoods in $G_\alpha^{(2)}(S)$.

Applying (3.8) again, *w.e.p.* for all $i, j \in S$ so that $|U_i - U_j| > \log(n)^{-1}$, we have that $\nu_S(i,j) \geq \log(n)^3$. This implies that i and j have distinct neighbourhoods in $G_\alpha^{(2)}(S)$ and thus also in $G_\alpha^{(2)}$ (see discussion in Section 3.3.1). Consequently, for pairs $i, j \in S$ with $|U_i - U_j| > \log(n)^{-1}$, we have

$$\{\sigma(i) < \sigma(j)\} \leftrightarrow \{\sigma_{\text{true}}(i) < \sigma_{\text{true}}(j)\}.$$

In other words, σ agrees with σ_{true} at precision level $\log(n)^{-1}$, completing the proof. □

Algorithm 3 is called many times in Algorithm 2, to produce a collection of sets and orderings $(\sigma^{(j)}, S^{(j)})_{j=1}^t$. Since $\mathcal{B}_1'$ may not occur *w.e.p.*, some of the set-ordering pairs in this collection may not agree with σ_{true} or σ_{eurt} at the required level of precision. We will need a minimal amount of agreement to prove that Algorithm 4, the alignment algorithm, works correctly based on *all* samples. Therefore, we now define a weaker version of agreement between orderings, and show that this weaker property holds *w.e.p.*.

Definition 3.16. Define

$$\mathcal{L} = \{i \in V \,:\, U_i < r_\alpha(\epsilon/2)\} \text{ and } \mathcal{R} = \{i \in V \,:\, U_i > \ell_\alpha(1 - \epsilon/2)\},$$

where $\epsilon > 0$ is as in Assumption 3.5. For any permutation σ of a subset $S \subset V$, let

$$\begin{aligned}\mathcal{L}(\sigma, S) &= \{i \in S \,:\, 1 \leq \sigma(i) \leq \zeta\}, \text{ where } \zeta = 4\log(n)^4, \text{ and}\\ \mathcal{R}(\sigma, S) &= \{i \in S \,:\, |S| \geq \sigma(i) \geq |S| - \zeta + 1\}.\end{aligned}$$

We say that σ *roughly agrees* with σ_{true} if $\mathcal{L}(\sigma, S) \subset \mathcal{L}$ and $\mathcal{R}(\sigma, S) \subset \mathcal{R}$. Similarly, σ *roughly agrees* with σ_{eurt} if $\mathcal{L}(\sigma, S) \subset \mathcal{R}$ and $\mathcal{R}(\sigma, S) \subset \mathcal{L}$.

Note that, by part (4) of Assumption 3.5, we have that $r_\alpha(\epsilon/2) < \ell_\alpha(1-\epsilon/2)$, and thus $\mathcal{L} \cap \mathcal{R} = \emptyset$. This implies that any permutation σ can roughly agree with at most one correct permutation.

Lemma 3.17. *Let $G_\alpha^{(2)}$ be as defined in Equation 2.1, and assume $\mathcal{A}^c$ holds. Let $m = \log(n)^5$, and let $S \sim Unif(\{T \subset V : |T| = m\})$. Let σ be the ordering returned by Algorithm 3 if the algorithm succeeds, or let σ be equal to σ_{true} restricted to S otherwise. Then*

$$\frac{\mathbb{P}[\{\sigma \text{ roughly agrees with } \sigma_{\text{true}} \text{ or } \sigma_{\text{eurt}}\} \,|\, \mathcal{F}]}{\mathbb{P}[\{\text{Algorithm 3 succeeds}\}]|\mathcal{F}} = 1 - n^{-\Omega(\log n)}$$

on the $\mathcal{F}$-measurable event $\mathcal{A}^c$. That is, if Algorithm 3 succeeds, then w.e.p. it returns a permutation that roughly agrees with σ_{true} or σ_{eurt}.

Proof. Consider a run of Algorithm 3 that succeeds; let S be the returned subset, and let σ be the returned permutation (note that σ represents an interval order of $G_\alpha^{(2)}(S)$). Let $j \in S$ be so that $\sigma(j) = 1$. By (3.8), *w.e.p.* the neighbour set of j has size

$$|\{q \in S : G_\alpha^{(2)}(j,q) = 1\}| \geq \log(n)^5/(8R) > \zeta = |\mathcal{L}(S,\sigma)|.$$

Since σ is an interval ordering of $G_\alpha^{(2)}(S)$, all neighbours of j must form an interval $\mathcal{I}$ in σ, and $\mathcal{L}(S,\sigma) \subset \mathcal{I}$. Since σ is an interval ordering, this also implies that all vertices in $\mathcal{I}$ (and thus $\mathcal{L}(S,\sigma)$) are mutually adjacent in $G_\alpha^{(2)}$.

We have shown that *w.e.p.* $\mathcal{L}(S,\sigma)$ is a clique, but have not yet shown that even *a single* element $s \in \mathcal{L}(S,\sigma)$ has a small associated latent variable U_s. Before starting the argument that this is true, we give some basic facts about S''. By (3.8), *w.e.p.* we have both that $G_\alpha^{(2)}(S'') = H_\alpha(S'')$, and that $H_\alpha(S'')$ is connected. Since H_α is a unit interval graph, so is $H_\alpha(S'')$. Since σ is an interval order of $G_\alpha^{(2)}(S)$, $\sigma|_{S''}$ is an interval order of $G_\alpha^{(2)}(S'') = H_\alpha(S'')$. Therefore, $\sigma|_{S''}$ on S'' must agree with σ_{true} or σ_{eurt}. We assume *wlog* that it agrees with σ_{true}.

The next step is to check that S'' has some intersection with the collection

$$\hat{\mathcal{L}} = \left\{ s \in S \,:\, U_s \leq \frac{1}{\log(n)} \right\} \tag{3.9}$$

of vertices with "small" latent variables. From (3.8), we have *w.e.p.* the inequalities

$$|\hat{\mathcal{L}}| \geq \frac{m}{\log(n)} - 2\log(n)^{3.5}, \quad |S \setminus S''| \leq \log(n)^3, \tag{3.10}$$

so $|\hat{\mathcal{L}} \cap S''| \geq |\hat{\mathcal{L}}| - |S \setminus S''| > 0$ for large enough n.

Having shown that $\hat{\mathcal{L}} \cap S'' \neq \emptyset$, we consider $s \in \hat{\mathcal{L}} \cap S''$ and show that it is in $\mathcal{L}(S,\sigma)$. For any $k \in S$ with $\sigma(k) < \sigma(s)$, one of the following must hold:

1. $\sigma_{\text{true}}(k) < \sigma_{\text{true}}(s)$. In this case $U_k < U_s$, so $k \in \hat{\mathcal{L}}$. From (3.10) above, we have that $|\hat{\mathcal{L}}| \leq \log(n)^4 + 2\log(n)^{3.5}$.

2. $\sigma_{\text{true}}(k) > \sigma_{\text{true}}(s)$. This means that s, k are incorrectly ordered by σ. Since $\sigma|_{S''}$ agrees with σ_{true}, we must be in one of the following subcases:
   (a) $k \in S \setminus S''$. As argued above, $|S \setminus S''| \leq \log(n)^3$.
   (b) s, k have the same neighbourhood in $H_\alpha(S'')$, and thus $\nu_S(s,k) \leq |S \setminus S''| \leq \log(n)^3$. From (3.8), this implies that $U_s < U_k \leq U_s + \log(n)^{-1}$. Again from (3.8), there are at most $m\log(n)^{-1} + 2\log(n)^{3.5} \leq 2\log(n)^4$ such vertices.

Combining these three cases, we see that there are at most $(\log(n)^4 + 2\log(n)^{3.5}) + \log(n)^3 + 2\log(n)^4$ vertices k with $\sigma(k) < \sigma(s)$. For large n, this amount is smaller than $4\log(n)^4 = \zeta$, and so $\sigma(s) \leq \zeta$. This implies $s \in \mathcal{L}(S, \sigma)$, and so we have shown that

$$\emptyset \neq \hat{\mathcal{L}} \cap S'' \subseteq \mathcal{L}(S, \sigma). \tag{3.11}$$

We are now ready to show that $\mathcal{L}(\sigma, S) \subseteq \mathcal{L}$. For n large enough, $\log(n)^{-1} < r_\alpha(\epsilon/2)$, and thus, by definition, $\hat{\mathcal{L}} \subset \mathcal{L}$. To complete the argument, let $s \in (\hat{\mathcal{L}} \cap S'') \subset \mathcal{L}(\sigma, S)$, and consider any other vertex $t \in \mathcal{L}(\sigma, S)$. As argued above, $\mathcal{L}(\sigma, S)$ is a clique in $G_\alpha^{(2)}$, so we know that s, t are connected in that graph. Since $s \in S''$ and $t \in S$, we have that $t \notin \mathcal{B}(n, s, \alpha)$. Therefore, on $\mathcal{A}_1$, $H_\alpha(s,t) = G_\alpha^{(2)}(s,t) = 1$, so

$$U_t \leq r_\alpha(U_s) \leq r_\alpha\left(\frac{1}{\log(n)}\right) < r_\alpha\left(\frac{\epsilon}{2}\right),$$

where the last inequality holds for all n sufficiently large. Therefore, $t \in \mathcal{L}$.

Collecting our assumptions throughout the proof of the theorem, we have shown that *w.e.p.* $\mathcal{L}(\sigma, S) \subset \mathcal{L}$ on $\mathcal{A}^c$. An analogous argument shows that, under the same assumptions, $\mathcal{R}(\sigma, S) \subset \mathcal{R}$. Thus, the result follows from Corollary 3.7. □

3.6. Analysis of Algorithm 4

In the previous section we showed that *w.e.p.* the output (σ, S) from Algorithm 3 roughly agrees with either σ_{true} or its total reversal, σ_{eurt}. In this section we show that the function a which is the output of Algorithm 4 correctly identifies whether the alignment is with σ_{true} or with σ_{eurt}.

Lemma 3.18. *Let Assumptions 3.5 hold. Let $t \geq (n\log(n))^2$, and let $(\sigma^{(j)}, S^{(j)})_{j=1}^t$ be the result of repeated calls to Algorithm 3, conditioned on non-failure. Define the function $a^* : \{1, \ldots, t\} \to \{-1, 1\}$ by*

$$a^*(j) = \mathbf{1}_{\{\sigma^{(j)} \text{ roughly agrees with } \sigma_{\text{true}}\}} - \mathbf{1}_{\{\sigma^{(j)} \text{ roughly agrees with } \sigma_{\text{eurt}}\}}. \tag{3.12}$$

Suppose Algorithm 4 is called with input $(\sigma^{(j)}, S^{(j)})_{j=1}^t$, and parameter $\zeta = 4\log(n)^4$. Then on the $\mathcal{F}$-measurable event $\mathcal{A}^c$,

$$\mathbb{P}[\{a = a^*\} \cup \{a = -a^*\} | \mathcal{F}] = 1 - n^{-\Omega(\log(n))}. \tag{3.13}$$

Proof. Recall that, for $1 \leq j \leq t$, $\mathcal{L}^{(j)}$ and $\mathcal{R}^{(j)}$ as computed in step 4 of Algorithm 4 correspond to the sets $\mathcal{L}(S^{(j)}, \sigma^{(j)})$ as in Definition 3.16. By Lemma 3.17, the condition

$$\bigcap_{j=1}^{t} \{\sigma^{(j)} \text{ roughly agrees with } \sigma_{\text{true}} \text{ or } \sigma_{\text{eurt}}\} \tag{3.14}$$

holds *w.e.p.* on $\mathcal{A}^c$.

By Definition (3.16), if $\sigma^{(j)}$ roughly agrees with σ_{true}, then $\mathcal{L}^{(j)} \subset \mathcal{L} = \{i : U_i < r_\alpha(\epsilon/2)\}$ and $\mathcal{R}^{(j)} \subset \mathcal{R} = \{i : U_i > \ell(\epsilon/2)\}$. Similarly, if $\sigma^{(j)}$ roughly agrees with σ_{eurt} then $\mathcal{L}^{(j)} \subset \mathcal{R}$ and $\mathcal{R}^{(j)} \subset \mathcal{L}$ (as argued, by Assumption 3.5, $\sigma^{(j)}$ can roughly agree with at most one of σ_{true}, σ_{eurt}).

Thus, if $\mathcal{L}^{(j)} \cap \mathcal{L}^{(k)} \neq \emptyset$ or $\mathcal{R}^{(j)} \cap \mathcal{R}^{(k)} \neq \emptyset$, and consequently $H(j,k)$ is set equal to 1 in step 8 of Algorithm 4, then $\sigma^{(j)}$ and $\sigma^{(k)}$ agree with the same correct permutation ($\{\sigma_{\text{true}}$ or $\sigma_{\text{eurt}}\}$), and thus $a^*(j)a^*(k) = 1$. Similarly, if $\mathcal{L}^{(j)} \cap \mathcal{R}^{(k)} \neq \emptyset$ or $\mathcal{R}^{(j)} \cap \mathcal{L}^{(k)} \neq \emptyset$, then $a^*(j)a^*(k) = -1$. Thus, condition (3.14) implies that the following holds for the function H computed in Algorithm 4 *w.e.p.*:

$$\bigcap_{1 \leq j,k, \leq t} \big(\{H(j,k) = 0\} \cup \{H(j,k) = a^*(j)a^*(k)\}\big). \tag{3.15}$$

That is, for each pair of samples j, k for which $H(j,k)$ is non-zero, H correctly identifies whether j and k are aligned with the same true ordering or with opposite orderings.

Now consider the graph $\mathcal{G}_H$ with vertex set $V_H = \{1, 2, \ldots, t\}$ and $\mathcal{G}_H(j,k) = |H(j,k)|$. That is, j and k are adjacent in $\mathcal{G}_H$ if sketch j and k could be aligned by the algorithm. In light of (3.15), to prove the result, it is enough to prove that $\mathcal{G}_H$ is connected *w.e.p.*.

As an extension of (3.9) in Lemma 3.17, define

$$S''^{(j)} = \{i \in S^{(j)} : \forall j \in S^{(j)}, i \notin \mathcal{B}(n,j,\alpha)\}, \text{ and}$$
$$\hat{\mathcal{L}}^{(j)} = \left\{ s \in S^{(j)} : U_s \leq \frac{1}{\log(n)} \right\} \text{ and } \hat{\mathcal{R}}^{(j)} = \left\{ s \in S^{(j)} : U_s \geq 1 - \frac{1}{\log(n)} \right\}.$$

By (3.11) in the proof of Lemma 3.17, we have that *w.e.p.* both

$$\emptyset \neq \hat{\mathcal{L}}^{(j)} \cap S''^{(j)} \text{ and } \hat{\mathcal{L}}^{(j)} \cap S''^{(j)} \subset \mathcal{L}^{(j)}. \tag{3.16}$$

Together with an analogous argument for $\hat{\mathcal{R}}$, we have *w.e.p.* that either

1. $P^{(j)} \equiv \mathcal{L}^{(j)} \cap \hat{\mathcal{L}}^{(j)} \neq \emptyset$ and $Q^{(j)} \equiv \mathcal{R}^{(j)} \cap \hat{\mathcal{R}}^{(j)} \neq \emptyset$, or
2. $P^{(j)} \equiv \mathcal{R}^{(j)} \cap \hat{\mathcal{L}}^{(j)} \neq \emptyset$ and $Q^{(j)} \equiv \mathcal{L}^{(j)} \cap \hat{\mathcal{R}}^{(j)} \neq \emptyset$.

Note that in the above we give a two-case definition for the sets $P^{(j)}, Q^{(j)}$; we clarify that we always choose the choice that makes both non-empty when such a choice is possible, choosing arbitrarily otherwise. By Lemma 3.17, *w.e.p.* there will be a unique choice making both non-empty.

There is an edge (i,j) in $\mathcal{G}_H$ as long as $(P^{(i)} \cup Q^{(i)}) \cap (P^{(j)} \cup Q^{(j)}) \neq \emptyset$. We make the following claim: *w.e.p.* for any $a,b \in V$ so that $U_a, U_b < \log(n)^{-1}$, there exists $1 \leq \ell \leq k$ so that $\{a,b\} \subset P^{(\ell)} \cup Q^{(\ell)}$. Before proving the claim, we show that this will complete the proof. For any $1 \leq i,j \leq k$, take $a \in (P^{(i)} \cup Q^{(i)})$ and $b \in (P^{(j)} \cup Q^{(j)})$. Let ℓ be so that $\{a,b\} \subset P^{(\ell)} \cup Q^{(\ell)}$. Then there are edges (i,ℓ) and (ℓ,j) in $\mathcal{G}_H$. This shows that $\mathcal{G}_H$ is connected.

It remains to show that our claim is true. Fix $a,b \in V$ so that $U_a, U_b < \log(n)^{-1}$. Let $\beta = \inf\{w^{(2)}(0,y) : 0 \leq y \leq \epsilon\}$. By part (3) of Assumption 3.5, $\beta > \alpha$. This implies that $\mathcal{B}(n,\alpha) \cap [0,\epsilon] = \emptyset$ for all n sufficiently large. It follows that $(U_a, U_b) \notin \mathcal{B}(n,\alpha)$, and thus $a \notin \mathcal{B}(n,b,\alpha)$ and $b \notin \mathcal{B}(n,a,\alpha)$.

To show that $a,b \in P^{(\ell)} \cup Q^{(\ell)}$, by 3.16 it suffices to show that $a,b \in S''^{(\ell)}$.

We have that

$$\mathbb{P}[\{a,b\} \subset S] \geq \left(\frac{m}{n}\right)^2.$$

Moreover, on $\mathcal{A}^c$, $|\mathcal{B}(n,a,\alpha) \cup \mathcal{B}(n,b,\alpha)| \leq 2\sqrt{n}\log(n)^2$, and thus

$$\mathbb{P}[\{a,b\} \in S'' \mid \{a,b\} \subset S] \geq 1 - 2(m-2)\frac{\log(n)^2}{\sqrt{n}} \geq \frac{1}{2},$$

on $\mathcal{A}^c$, and for n large enough. Thus the expected number of sketches for which $\{a,b\} \subset S^{(j)}$ is at least $t(m^2/(2n^2)) \geq \log(n)^{11}$. Since the sets $\{S^{(j)}\}$ are independent, by the Chernoff bound we obtain that, *w.e.p.* at least one of the sketches contains both a and b. This completes the proof. □

3.7. Analysis of Algorithm 2

We will give a bound on the output of Algorithm 2. Define the sets

$$Q_1 = \{(p,q) \in \tilde{V} \,:\, |U_p - U_q| > \log(n)^{-1}\}$$
$$Q_2 = \{(p,q) \in \tilde{V} \,:\, (U_p, U_q) \notin \mathcal{B}(n,\alpha)\}.$$

For a comparison F, define the event:

$$\mathcal{B}_2'(F) = \{\forall (p,q) \in Q_1 \cap Q_2,\, F(p,q) = F_{\mathrm{true}}(p,q)\} \tag{3.17}$$

that F gives the correct comparison for all pairs of sufficiently-distant vertices in Q_2. We have:

Lemma 3.19 (Sketch Correctness)**.** *Let $G_\alpha^{(2)}$ be as defined in Equation 2.1, and let F be the output of Algorithm 2, on input $G_\alpha^{(2)}$ and parameters $m = \log(n)^5$ and $t = (n\log(n))^2$. Then on the $\mathcal{F}$-measurable event $\mathcal{A}^c$,*

$$\mathbb{P}[\mathcal{B}_2'(F) \cup \mathcal{B}_2'(-F) \mid \mathcal{F}] = 1 - n^{-\Omega(\log(n))}.$$

Proof. Let $\{(S^{(j)},\sigma_j)\}_{j=1}^{t}$ be the sequence of subsets and preorders appearing immediately after the alignment correction in Step 12 of Algorithm 2. Let $\mathcal{C}$ be as in Algorithm 3 at the end of its run. Let $c_1 = \log(n)^7$, and let $\mathcal{N} : \tilde{V} \to \mathbb{N}$ be the function counting the number of times vertices s and t occurred together in a sample, that is, $\mathcal{N}(s,t) = |\{j : s,t \in S^{(j)}\}|$.

Next, consider a pair $(p,q) \in Q_2$, so $(U_p, U_q) \notin \mathcal{B}(n,\alpha)$, and let S be a uniformly-chosen size-m subset of $\{1,2,\dots,n\}$, as sampled by Algorithm 3 when it is called for the i'th time in Step 3 of Alg. 2. If $p,q \in S$, then $p,q \in S^{(i)}$ if Alg. 3 succeeds. By Lemma 3.15, on the event $\mathcal{A}^c$ we have

$$\mathbb{P}[p,q \in S^{(i)} \,|\, \mathcal{F}] \geq (1 - O(n^{-0.49}))\mathbb{P}[p,q \in S] \geq 0.9(m/n)^2.$$

Let $e^{(i)}(p,q) = \mathbf{1}_{\{p,q\in S^{(i)}\}}$. Then on $\mathcal{A}^c$,

$$\mathbb{E}[\mathcal{N}(p,q)|\mathcal{F}] = \mathbb{E}[\sum_{i=1}^{t} e^{(i)}(p,q)|\mathcal{F}] \geq 0.9(m/n)^2 t = 0.9\log(n)^{12} \geq 2c_1.$$

By the Chernoff bound, on $\mathcal{A}^c$ we have

$$\mathbb{P}[\mathcal{N}(p,q) < c_1 \,|\, \mathcal{F}] = n^{-\Omega(\log(n))}. \tag{3.18}$$

Let $\mathcal{B}_3'$ be the event that, when Algorithm 4 is called with input $(S^{(i)},\sigma^{(i)})_{i=1}^{t}$ in step 10 of Algorithm 2, the returned function a agrees with one of the true functions a^* or $-a^*$ as defined in the statement of Lemma 3.18. By Lemma 3.18,

$$\mathbb{P}[\mathcal{B}_3' \,|\, \mathcal{F}] = 1 - n^{-\Omega(\log(n))} \tag{3.19}$$

on $\mathcal{A}^c$.

Considering the event $\mathcal{B}_3'$ holds, we assume without loss of generality that $a = a^*$ for all arguments. This implies that, after the realignment in step 10 of Algorithm 2, $\sigma^{(i)}$ roughly agrees with σ_{true} for each $i = 1,\dots,t$. Recall the event $\mathcal{B}_1'$ as defined in Lemma 3.15; we define $\mathcal{B}_1'(i)$ to be the event that $\mathcal{B}_1'$ holds for the call to Algorithm 3 that produced the pair $(S^{(i)},\sigma^{(i)})$.

Next, fix $(p,q) \in Q_1$, so $|U_p - U_q| \geq \log(n)^{-1}$, and assume without loss of generality that $\sigma_{\text{true}}(p) < \sigma_{\text{true}}(q)$. On the event $\mathcal{B}_1'(i)$, $\sigma^{(i)}$ agrees at precision level $\log(n)^{-1}$ with σ_{true}. Defining $f^{(i)}(u,v) = e^{(i)}(p,q)\mathbf{1}_{\mathcal{B}_1'(i)}$ and applying Lemma 3.15, we have for large enough n,

$$\mathbb{P}[f^{(i)}(u,v) = 1|\mathcal{F}] \geq 0.9\mathbb{P}[e^{(i)}(u,v) = 1|\mathcal{F}]$$

on $\mathcal{A}^c$. Thus, $\mathbb{E}[\mathcal{C}(p,q)|\mathcal{F}] \geq 0.9\mathcal{N}(p,q)$ on $\mathcal{A}^c$ for all n sufficiently large. Applying the Chernoff bound and then (3.19),

$$\mathbb{P}[\mathcal{C}(p,q) > 0|\mathcal{F}] \geq \mathbb{P}[\{\mathcal{C}(p,q) < 0\} \cup (\mathcal{B}_3')^c|\mathcal{F}] - \mathbb{P}[\mathcal{B}_3^c|\mathcal{F}] = 1 - n^{-\Omega(\log(n))} \tag{3.20}$$

on $\mathcal{A}^c$. Combining Inequalities (3.18) and (3.20) and a union bound,

$$\mathbb{P}\left[\bigcup_{(p,q)\in Q_1\cap Q_2,\, U_p<U_q} (\{\mathcal{C}(p,q) \geq 0\})|\mathcal{F}\right] = n^{-\Omega(\log(n))} \tag{3.21}$$

on $\mathcal{A}^c$. This event is exactly the complement of $\mathcal{B}_2'(F)$, so this completes the proof. □

3.8. Analysis of Algorithms 5 and 1 and proof of Theorem 3

Define the set

$$Q_3 = \left\{ (i,j) \in \tilde{V} \,:\, |U_i - U_j| > \frac{\log(n)^4}{\sqrt{n}} \right\}$$

The following lemma shows that Algorithm 5, as called by Alg. 1 and run with the correct input and parameters, gives as output a comparison function that correctly orders all pairs in Q_3, i.e., all pairs that are sufficiently far apart.

Lemma 3.20 (Correct refinement)**.** *Let w be a graphon and $\alpha \in (0,1)$ be so that Assumptions 3.5 hold, and let $G \sim w$ be a graph of size n. Consider a run of Algorithm 1 with parameters as in (3.2) on input G, α; let F' be as it appears on line 3 of Algorithm 1. Then*

$$\bigcup_{(u,v)\in Q_3} \{F'(u,v) \neq F_{\text{true}}\} \subset \mathcal{A} \cup (\mathcal{B}_2(F)')^c.$$

Proof. We assume now that $\mathcal{A}^c$ and $\mathcal{B}_2'(F)$ both hold. Fix $p, q \in V$, and assume without loss of generality that $U_p < U_q$, so $F_{\text{true}}(p,q) = 1$. Let $D_{p,q}$ and $D_{q,p}$ be as computed in Alg. 5, namely

$$D_{p,q} = |\{x \in N(p) \setminus N(q) \,:\, F(x,q) = 1\}| - |\{x \in N(p) \setminus N(q) \,:\, F(x,q) = -1\}|$$
$$D_{q,p} = |\{x \in N(q) \setminus N(p) \,:\, F(x,p) = 1\}| - |\{x \in N(q) \setminus N(p) \,:\, F(x,p) = -1\}|.$$

Let $B = \mathcal{B}(n,p,\alpha) \cup \mathcal{B}(n,q,\alpha)$ and $c_2 = 5\sqrt{n}\log(n)^2$. Since $\mathcal{A}_2^c$ holds, $|B| \leq 2\sqrt{n}\log(n)^2 < c_2/2$.

We next wish to show that $F'(p,q) = 1$ for all pairs p, q with $\max(|N(p) \setminus N(q)|, |N(q) \setminus N(p)|) > 2c_2$. We begin with the case $|N(p) \setminus N(q)| > 2c_2$. In this case, $(N(p) \setminus N(q)) \setminus B \neq \emptyset$. For each $v \in (N(p) \setminus N(q)) \setminus B$, we have that $G_\alpha^{(2)}(q,v) = 0$, and, since $v \notin \mathcal{B}(n,q,\alpha)$ and we assume $\mathcal{A}^c$ holds, $H_\alpha(q,v) = 0$. Thus, by Conditions 3.5, $|U_q - U_v| \geq \epsilon \geq \log(n)^{-1}$. Therefore, since $\mathcal{B}_2'(F)$ holds, we have that $F(v,q) = F_{\text{true}}(v,q)$.

By a similar argument, $H_\alpha(p,v) = G_\alpha^{(2)}(p,v) = 1$. So v is a neighbour of p but not of q in H_α. Since H_α is diagonally increasing, this implies that$U_v < U_q$, and so $\sigma_{\text{true}}(v) < \sigma_{\text{true}}(q)$, so $F_{\text{true}}(v,q) = 1$.

In our first case, we have just shown that $F(v,q) = 1$ for all $v \in (N(p) \setminus N(q)) \setminus B$ satisfying $F_{\text{true}}(v,q) = 1$. Moreover, by our assumption $|N(p) \setminus N(q)| > 2c_2 > \frac{1}{2}c_2 \geq |B|$,

$$|\{x \in N(p) \setminus N(q) \,:\, F(x,q) = 1\}| \geq |(N(p) \setminus N(q)) \setminus B| > 2c_2 - |B|, \text{ and} \tag{3.22}$$

$$|\{x \in N(p) \setminus N(q) \,:\, F(x,q) = -1\}| \le |B| \le \frac{1}{2}c_2.$$

Therefore in this case, $D_{p,q} > c_2$ and so $F'(p,q) = 1$.

By essentially the same argument, in the case $|N(q) \setminus N(p)| > 2c_2$, we have $D_{q,p} < -c_2$. Combining these two cases, we have that

$$F'(p,q) = F_{\text{true}}(p,q),\ \forall p,q \in V \text{ s.t. } \max(|N(p) \setminus N(q)|, |N(q) \setminus N(p)|) > 2c_2. \tag{3.23}$$

Now suppose $(p,q) \in Q_3$, so $|U_p - U_q| > \frac{\log(n)^4}{\sqrt{n}}$. Since $\mathcal{A}_4^c$ holds, for all large enough n

$$|N(p) \setminus N(q)| + |N(q) \setminus N(p)| = \nu(p,q) \ge \sqrt{n}\log(n)^3 > 20\sqrt{n}\log(n)^2 = 4c_2,$$

and thus $\max(|N(p) \setminus N(q)|, |N(q) \setminus N(p)|) > 2c_2$. Applying Inequality (3.23), this implies

$$\cup_{(p,q)\in Q_3}\{F'(p,q) \neq F_{\text{true}}(p,q)\} \subset \mathcal{A} \cup (\mathcal{B}_2'(F))^c, \tag{3.24}$$

completing the proof. □

Finally, we check that $\sigma_{F'}$ is never much less accurate than F':

Lemma 3.21. *Let F be any comparison that agrees with a permutation σ_{true} at precision level d. Then $\sigma_{F'}$ agrees with σ_{true} at precision level $2d$.*

Proof. Assume *wlog* that $\sigma_{\text{true}}(i) = i$ for all i. Then for $j > i + 2d$, we have

$$\begin{aligned}\sigma_F(j) - \sigma_F(i) &= \sum_{k=1}^{n}(F(j,k) - F(i,k)) \\ &\ge 2(j-i) - |\{k \,:\, \min(|i-k|, |j-k|) \le d\}| \\ &\ge 2(j-i) - 4d > 0. \quad \square\end{aligned}$$

The proof of Theorem 3 now follows directly from the lemmas. Let σ be the result returned by Algorithm 1, and set Q_3 as above, and

$$Q_4 = \{(i,j) \in \tilde{V} \,:\, |\sigma_{\text{true}}(i) - \sigma_{\text{true}}(j)| > 2\sqrt{n}\log(n)^5\}.$$

Define:

$$\begin{aligned}\mathcal{B}_4' &= \{\forall\, i,j \in Q_3,\ \{\sigma(i) < \sigma(j)\} \longleftrightarrow \{\sigma_{\text{true}}(i) < \sigma_{\text{true}}(j)\}\} \\ \mathcal{B}_5' &= \{\forall\, i,j \in Q_4,\ \{\sigma(i) < \sigma(j)\} \longleftrightarrow \{\sigma_{\text{true}}(i) < \sigma_{\text{true}}(j)\}\}\end{aligned}$$

to be the event that σ is correct for all vertices that are sufficiently distant. We have:

Lemma 3.22 (Merge Correctness)**.** *Let w be a graphon and $\alpha \in (0,1)$ be so that Assumptions 3.5 hold, and let $G \sim (n, w)$. Let σ be the output of Algorithm 1, run with parameters as in (3.2) on input G, α. Then*

$$\mathbb{P}[\mathcal{B}_4' \cap \mathcal{B}_5'] = 1 - n^{-\Omega(\log(n))}.$$

Proof. Let F be the output of Alg. 2 as called in line 2 of Alg. 1. Let F' be the output of Alg. 5 when called in line 3 of Alg. 1. By Lemma 3.20, if $\mathcal{B}_2'(F)$ and $\mathcal{A}^c$ both hold, then F' agrees with the true ordering on Q_3. Applying Lemma 3.21, this implies

$$(\mathcal{B}_4')^c \subset \mathcal{A} \cup (\mathcal{B}_2'(F))^c.$$

Applying Corollary 3.7 and Lemma 3.19 to this inclusion gives

$$\mathbb{P}[(\mathcal{B}_4')^c] = n^{-\Omega(\log(n))}.$$

We have shown that all pairs in Q_3 are ordered correctly *w.e.p.*; now we show that this implies all pairs in Q_4 are also ordered correctly *w.e.p.*. Suppose that $(p, q) \in Q_4$ and $U_p < U_q$. By definition of Q_4,

$$|\{k : U_k \in (U_p, U_q)\}| = \sigma_{\text{true}}(q) - \sigma_{\text{true}}(p) - 1 \geq \sqrt{n} \log(n)^5.$$

If $\mathcal{A}_5^c$ holds, then $|\{k : U_k \in (U_p, U_q)\}| \leq n|U_q - U_p| + \sqrt{n}\log(n)$. Putting these two inequalities together, on $\mathcal{A}^c$ we have

$$|U_q - U_p| \geq \frac{\sqrt{n}\log(n)^5 - \sqrt{n}\log(n)}{n} \geq \frac{2\log(n)^4}{\sqrt{n}}.$$

Therefore, by Lemmas 3.20 and 3.21, each pair (p, q) is ordered correctly on $\mathcal{A}$, and so by Inequality (3.8) and Corollary 3.7,

$$\mathbb{P}[\mathcal{B}_5'] \geq \mathbb{P}[\mathcal{B}_4' \cup \mathcal{A}^c] - \mathbb{P}[\mathcal{A}] = 1 - n^{-\Omega(\log(n))}.$$

This completes the proof. □

Theorem 3 now follows immediately from Lemma 3.22.

4. Iterative error reduction

In this section, we discuss and analyze Algorithm 6, and prove Theorem 2. Throughout this section, we fix a graphon w that satisfies Assumptions 1.8 and 3.5, and use the constants introduced in those assumptions. We assume that the input graph G being analyzed is sampled from $G \sim w$ and we fix the associated notation from this choice as in the start of Section 3.2; for example, we denote by $\{U_i\}$, $\{U_{ij}\}$ the random variables used to construct G in representation (1.2). Finally, we let other notation follow as in Algorithm 6; for example, we let $\{B_j\}$ be the Bernoulli random variables introduced in step 1, and sets V_i be the sets constructed in steps 2 and 5. Let $G_i = G[V_i]$. We observe here that, conditional on $\{B_j\}$, we have $G_i \sim w$ is a random graph of size $|V_i|$ that is sampled from w.

Remarks 4.1 (Strengthening of Theorem 2)**.** *Just as Theorem 3 is a stronger version of Theorem 1, our arguments can be used to give other versions of Theorem 2. As there are two interesting versions, and both can be described entirely in terms of a small number of simple substitutions, we describe these substitutions in this remark rather than copying large amounts of nearly-identical text:*

1. ***Non-Uniform Graphons:*** *All references to Assumption 1.5 in the theorem statement and proof can be replaced by references to the (strictly weaker) Assumption 3.5.*
2. ***Other Initial Sketches:*** *Step 2 of Algorithm 6 calls Algorithm 1, but in fact any algorithm can be used for this initial sketch. The conclusions of Theorem 2 remain true as stated if all of the following substitutions are made:*
   - *Replace the call to Algorithm 1 in step 2 of Algorithm 6 by a call to another algorithm, which we call OtherSketch.*
   - *Replace Assumption 1.5 in the theorem statement with the following two assumptions: (i) part (3) of Assumption 3.5, and (ii) a new assumption, which we will call Assumption OS, that the output of OtherSketch satisfies the error bound appearing in Theorem 1.*
   - *In the proof, replace all references to Theorem 3 by references to Assumption OS.*

4.1. Algorithm 7: basic concepts

We start by discussing Algorithm 7, which builds a new ordering σ_2 on V_2 by looking at the edges in G_2 and the old ordering σ_1 on $V_1 \subset V_2$. Our main goal is to show the following: if Algorithm 7 is called with the right choice of parameters C_1, C_2, C_3, and σ_1 agrees with σ_{true} at certain precision level d_1, then σ_2 agrees with σ_{true} at a finer precision level $d_2 < d_1$. We now outline the proof and define the basic concepts. As in earlier informal descriptions, we elide the distinction between a permutation and its total reverse.

First, we need to define the boundaries where w drops to zero. Let δ be as in Assumption 1.8, and define new boundary functions $r_\delta, \ell_\delta : [0,1] \mapsto [0,1]$ by:

$$r_\delta(x) = \sup\{y \in [0,1] \,:\, w(x,y) \geq \delta\},$$
$$\ell_\delta(x) = \inf\{y \in [0,1] \,:\, w(x,y) \geq \delta\}.$$

These definitions will replace the very similar boundaries defined in (3.1), which were defined with respect to the square $w^{(2)}$. By Assumption 1.8, $w(x,y) = 0$ if $y > r_\delta(x)$ or $y < \ell_\delta(x)$.

To start the heuristic analysis, assume that σ_1 is an ordering of V_1 which agrees with σ_{true} at precision level d_1. We now show how σ_1 is used in steps 1–12 of Algorithm 7 to find an ordering σ_2 of $V_2 \setminus V_1$ which agrees with σ_{true} at smaller precision level d_2, and has certain other helpful properties to be specified later.

Fix $i, j \in V_2 \setminus V_1$ so that $|U_i - U_j| > d_2$. Assume without loss of generality that $U_i < U_j$, so we have that $\sigma_{\text{true}}(i) < \sigma_{\text{true}}(j)$. Since w is diagonally increasing,

$$\{z : w(U_i, z) = 0 \neq w(U_j, z) \text{ or } w(U_j, z) = 0 \neq w(U_i, z)\} \\ = [r_\delta(U_i), r_\delta(U_j)] \cup [\ell_\delta(U_i), \ell_\delta(U_j)].$$

From Assumption 1.8 we have that

$$\big(r_\delta(U_j) - r_\delta(U_i)\big) + \big(\ell_\delta(U_j) - \ell_\delta(U_i)\big) \geq B|U_j - U_i|.$$

Assume first that

$$r_\delta(U_j) - r_\delta(U_i) \geq \ell_\delta(U_j) - \ell_\delta(U_i), \tag{4.1}$$

and thus

$$r_\delta(U_i) < r_\delta(U_j) - \frac{B}{2}|U_i - U_j| \leq r_\delta(U_j) - d_2 \log(n)^{-1}, \tag{4.2}$$

where the second inequality holds for all n large enough that $\log(n) > \frac{2}{B}$. Let

$$I_{\text{true}}(i, j) = [r_\delta(U_j) - d_2 \log(n)^{-1}, r_\delta(U_j)]. \tag{4.3}$$

Then $w(U_i, x) = 0$, $w(U_j, x) \geq \delta$ for all $x \in I_{\text{true}}(i, j)$. Because of this "sharp" distinction between $w(U_i, x)$ and $w(U_j, x)$, the region I_{true} is very informative in distinguishing the ordering of i, j. Moreover, many points in V_1 will have latent positions in $I_{\text{true}}(i, j)$ (roughly $p_1 d_2 n \log(n)^{-1}$ points on average). Step 2 of the algorithm computes the set $R(i, j)$, defined as the C_1 neighbours of either i or j in V_1 that are ranked highest according to σ_1. The idea behind the algorithm is that $R(i, j)$ is a small set that must contain many points with latent positions in this strongly-distinguishing set $I_{\text{true}}(i, j)$; when this works, we expect the vertex with larger value U_j to have significantly more neighbours in $R(i, j)$.

The main motivation behind our definition of $R(i, j)$ is thus to include the entire "signal" in I_{true}, while including as few additional "noisy" vertices as possible. We will prove the following estimates, which make this intuition rigorous:

1. **Large Signal:** Denote by

$$\mathbf{Dist}_R(i, j) \equiv \{k \in V_1 : U_k \in I_{\text{true}}(i, j)\} \tag{4.4}$$

the set of vertices that strongly distinguish between i and j on the right. We will show that $\mathbf{Dist}_R(i, j)$ is relatively large. In particular, we show in Lemma 4.2 below that *w.e.p.* $|\mathbf{Dist}_R(i, j)| \geq p_1 d_2 n \log(n)^{-2}$.
As pointed out earlier, $\mathbf{Dist}_R(i, j) \cap N(i) = \emptyset$, so all vertices in $R(i, j) \cap \mathbf{Dist}_R(i, j)$ are neighbours of j and not of i. This leads to the definition:

$$\mathbf{Signal}_R(i, j) \equiv \mathbf{Dist}_R(i, j) \cap N(j). \tag{4.5}$$

We will show in Lemma 4.3 that, for appropriately chosen C_2, w.e.p.

$$|\mathbf{Signal}_R(i, j)| \geq 2C_2. \tag{4.6}$$

Moreover, we show in Lemma 4.4 that w.e.p.

$$\mathbf{Signal}_R(i,j) \subseteq R(i,j). \tag{4.7}$$

2. **Small Noise:** There are many vertices $k \in R(i,j)$ that are not in the "high-signal" region $\mathbf{Dist}_R(i,j)$. Due to the fact that G is sampled from a diagonally increasing graphon, which satisfies inequality (1.3), we expect to see at least as many neighbours of j as neighbours of i in $R(i,j) \setminus \mathbf{Dist}_R(i,j)$, but there may be very small (or no) signal in this discrepancy. Indeed, the expected discrepancy is *exactly* 0 for our test-case graphon (1.4). Thus, we view these edges as essentially adding noise to our comparison. We define

$$\mathbf{Noise}_R(i,j) \equiv |R(i,j) \cap N(i)| - |(R(i,j) \cap N(j)) \setminus \mathbf{Signal}_R(i,j)|. \tag{4.8}$$

When $\mathbf{Signal}_R(i,j) \subseteq R(i,j)$, then,

$$|N(j) \cap R(i,j)| - |N(i) \cap R(i,j)| = |\mathbf{Signal}_R(i,j)| - \mathbf{Noise}_R(i,j).$$

We will show in Lemma 4.5 below that, w.e.p. $\mathbf{Noise}_R(i,j) < C_2$. In particular, this noise term will always be small compared to the signal term in Inequality (4.6). When these high-probability events all occur, the function $F^{(2)}$ as defined in Steps 3–9 of Algorithm 7 agrees with σ_{true} if $|U_i - U_j| > d_2$.

Finally, L, $\mathbf{Dist}_L$, $\mathbf{Signal}_L$, $\mathbf{Noise}_L$ are the obvious analogues of R, $\mathbf{Dist}_R$, $\mathbf{Signal}_R$, $\mathbf{Noise}_R$ on the left, and the obvious analogues to the above bounds hold by the same arguments for the case where (4.1) does not hold (that is, where $r_\delta(U_j) - r_\delta(U_i) < \ell_\delta(U_j) - \ell_\delta(U_i)$). Therefore, *w.e.p.* the ordering σ_2' on $V_2 \setminus V_1$ obtained in Step 12 of the algorithm agrees with σ_{true} at precision level d_2.

The second part of the algorithm (steps 13–24) then extends this ordering to all of V_2.

(3) **Highest and lowest ranked neighbours indicate ordering** The values $t(i)$ and $b(i)$, computed in step 14, are the ranks, according to σ_2', of the top and the bottom element in $N(i) \cap (V_2 \setminus V_1)$, respectively. If $|U_i - U_j| > d_2 \log(n)^2$, then *w.e.p.* there will be many neighbours k of j with latent position U_k greater than that of any neighbour of i. Thus the top elements of $N(i)$ and $N(j)$ in $V_2 \setminus V_2$, ranked according to σ_{true}, will differ significantly in rank. Since σ_2' is a good approximation of σ_{true}, this will also be true for these same elements when ranked according to σ_2'. Thus, as will be made precise in Lemma 4.7, the ordering σ as returned by the algorithm will agree with σ_{true} at precision level $d_2 \log(n)^2$.

4.2. Algorithm 7: correctness

We are now ready to prove the statements in the previous section. We retain the same notation, and now give conditions on the parameters which we assume to hold throughout the remainder of this section:

$$\begin{aligned} 1 &\geq p_2 \geq p_1 \log(n)^3, \\ \log(n)^{-5} &\geq d_1 \geq \log(n)^{10}/(p_1 n), \\ d_2 &= \sqrt{d_1/(p_1 n)}\log(n)^{-1}, \\ C_1 &= \lceil p_1 d_1 n \log(n)^4 \rceil, \\ C_2 &= \lfloor \sqrt{p_1 d_1 n}\log(n)^6 \rfloor, \\ C_3 &= \lfloor \sqrt{p_1 d_1 n}\log(n)^2 \rfloor. \end{aligned} \tag{4.9}$$

Note that the definition of C_1, C_2, C_3 as given above agrees with the definition of these parameters in Step 5 of Algorithm 6 in its first iteration (with $i = 1$). The first three conditions are satisfied by our choice of the sequence p_i, d_i in Equation (4.21). We use the following immediate consequences of (4.9):

$$\begin{aligned} d_2 &\leq d_1 \log(n)^{-6} < d_1, \\ d_1 &\geq \frac{\log(n)^{13}}{n} \\ C_1 &\leq p_1 n \log(n)^{-1} + 1, \text{ and} \\ \sqrt{p_1 d_1 n} &\geq \log(n)^5. \end{aligned}$$

Moreover, we assume throughout that n is large enough so that $\log(n)$ exceeds any of the constants appearing in this section.

For the correctness proof of Algorithm 7, we define certain bad events on the variables U_i and $U_{i,j}$, and show that *w.e.p.* they do not occur. Let

$$\begin{aligned} \mathcal{A}_6 = \bigcup_{k,\ell \in V,\, U_k < U_\ell - \frac{1}{n}} & \Big\{ \big| |\{i \in V : U_k < U_i < U_\ell\}| - n|U_\ell - U_k| \big| \\ & > \sqrt{(U_\ell - U_k)n}\log^2(n) \Big\}. \end{aligned}$$

On the event $\mathcal{A}_6^c$, the number of vertices in each interval is concentrated around its expected value. The event is a property of variables U_i, and is closely related to the event $\mathcal{A}_5$ as given in Definition 3.6. It is a straightforward consequence of Chernoff's inequality and a union bound that $\mathcal{A}_6^c$ occurs *w.e.p.*

We will also need this kind of nice behaviour for all our sampled sets V_i. For a given p, let $S = S(p) \equiv \{i : B_i \leq p\}$ and define

$$\mathcal{A}_6'(p) = \bigcup_{k,\ell \in V,\, U_k < U_\ell - \frac{1}{np}} \Big\{ \big| |\{i \in S(p) : U_k < U_i < U_\ell\}| - np|U_\ell - U_k| \big| \tag{4.10}$$

$$> 2\sqrt{np(U_\ell - U_k)}\log(n)\Big\}.$$

By the Chernoff bound, for any given sequence $p = p(n) \in (0,1)$ satisfying $\limsup_{n\to\infty} \frac{-\log(p(n))}{\log(n)} < 1$, $(A_6'(p))^c$ holds *w.e.p.* This also means that it holds simultaneously *w.e.p.* for any sequence $p_1, p_2, \ldots, p_k$, as long as k grows at most polynomially quickly in n.

We can use $\mathcal{A}_6'$ to establish bounds on the size of $\mathbf{Dist}_R(i,j)$, where i and j are so that $U_i > U_i + d_2$. We assume without loss of generality that $r_\delta(U_j) - r_\delta(U_i) \geq \ell_\delta(U_j) - \ell_\delta(U_i)$. Applying Inequality (4.2),

$$U_j > U_i + d_2 \quad \text{and} \quad r_\delta(U_j) - r_\delta(U_i) \geq d_2 \log(n)^{-1}. \tag{4.11}$$

Lemma 4.2. *W.e.p. for all i, j satisfying (4.11),*

$$|\mathbf{Dist}_R(i,j)| \geq p_1 d_2 n \log(n)^{-2}.$$

Proof. Let i and j as stated. By definition, $|I_{true}(i,j)| = d_2 \log(n)^{-1}$, and so by the choice of parameters as given in (4.9) the set $\mathbf{Dist}_R(i,j) = \{s \in V_1 : s \in I_{true}(i,j)\}$ has expected size

$$\mathbb{E}(|\mathbf{Dist}_R(i,j)|) = np_1|I_{true}(i,j)| = p_1 d_2 n \log(n)^{-1}.$$

We also note that $\frac{1}{p_1 n} \leq d_2 \leq |U_i - U_j|$. Combining these two facts, for all $i, j \in V_2 \setminus V_1$,

$$\{|\mathbf{Dist}_R(i,j)| < p_1 d_2 n \log(n)^{-2}\} \subset \mathcal{A}_6'(p_1).$$

Since *w.e.p.* $\mathcal{A}_6'(p_1)^c$ holds, the lemma follows. □

We will next check that all moderately-long intervals contain roughly the expected number of neighbours of any given vertex (recall that $N(i)$ is the usual graph neighbourhood of i in G). More precisely, for a set $S \subset V$, an ordering σ on S, and a pair $k, \ell \in S$, define the interval $I(S,\sigma,k,\ell) = \{s \in S : \sigma(k) < \sigma(s) < \sigma(\ell)\}$ and define $\mathcal{I}(S,\sigma) = \{I(S,\sigma,k,\ell) : k,\ell \in S\}$ to be the collection of such intervals. For any vertex $i \in V$ and set $I \subset V$, let $W(i,I) = \sum_{j\in I} w(U_i, U_j) = \mathbb{E}[|N(i) \cap I|]$. For sets $S, T \subseteq V$, and ordering σ of S, define

$$\mathcal{A}_7(S,T,\sigma) = \bigcup_{I \in \mathcal{I}(S,\sigma),\, i \in T} \left\{ \big| |I \cap N(i)| - W(i,I) \big| > \sqrt{W(i,I)}\log(n) + \log(n)^2 \right\}$$

We now use this to show that $\mathbf{Signal}_R(i,j)$ is sufficiently large.

Lemma 4.3. *W.e.p. for all i, j satisfying (4.11),*

$$|\mathbf{Signal}_R(i,j)| \geq 2C_2.$$

Proof. We first show that $\mathcal{A}_7^c$ occurs *w.e.p.* for the relevant parameters. Let $\mathcal{H}$ be the σ-algebra generated by the variables $\{U_i\}$ and $\{B_i\}$, and let $\mathcal{H}(G_1)$ be the σ-algebra of the variables $\{U_i\}$ and $\{B_i\}$, and $\{U_{i,j} : i, j \in V_1\}$. Recalling that the size of the set $|I \cap N(i)|$ appearing in the definition of $\mathcal{A}_7$ can be written as a sum of independent Bernoulli random variables in these σ-algebras,[3] then applying Hoeffding's inequality and a union bound, we immediately have

$$\begin{aligned} \mathbb{P}[\mathcal{A}_7(V_1, V_2 \setminus V_1, \sigma_1) \,|\, \mathcal{H}(G_1)] &= n^{-\Omega(\log(n))}, \qquad (4.12)\\ \mathbb{P}[\mathcal{A}_7(V_2, V_2, \sigma_{\text{true}}) \cup \mathcal{A}_7(V_1, V_2, \sigma_{\text{true}}) \,|\, \mathcal{H}] &= n^{-\Omega(\log(n))}. \end{aligned}$$

By definition $\mathbf{Dist}_R(i, j)$ is an interval in $\mathcal{I}(V_1, \sigma_{\text{true}})$; also recall $|\mathbf{Signal}_R(i, j)| = |\mathbf{Dist}_R(i, j) \cap N(j)|$. By Assumption 1.8, $w(U_j, U_k) \geq \delta$ for all $k \in \mathbf{Dist}_R(i, j)$, and thus $W(j, \mathbf{Dist}_R(i, j)) \geq \delta |\mathbf{Dist}_R(i, j)|$. Also, by (4.9), $C_2/\delta \geq \log(n)^3$ for n sufficiently large. We can then directly conclude that

$$\begin{aligned} &\{|\mathbf{Signal}_R(i, j)| < 2C_2\} \qquad (4.13)\\ &\subset \{|\mathbf{Dist}_R(i, j)| < 3C_2/\delta\} \cup \{|\mathbf{Dist}_R(i, j) \cap N(j)| < (2/3)\delta |\mathbf{Dist}_R(i, j)|\}\\ &\subset \mathcal{A}'{}_6(p_1) \cup \mathcal{A}_7(V_1, V_2 \setminus V_1, \sigma_{\text{true}}). \end{aligned}$$

Since the complements of the events $\mathcal{A}_6'(p_1)$ and $\mathcal{A}_7(V_1, V_2 \setminus V_1, \sigma_{\text{true}})$ occur *w.e.p.*, the result follows. □

In addition, we show that this large signal is indeed contained in $R(i, j)$:

Lemma 4.4. *W.e.p. for all $i, j \in V_2 \setminus V_1$ satisfying (4.11),*

$$\mathbf{Signal}_R(i, j) \subset R(i, j).$$

Proof. Suppose $\mathbf{Signal}_R(i, j) = \mathbf{Dist}_R(i, j) \cap N(j) \not\subset R(i, j)$. Let

$$k = \operatorname{argmax}\{\sigma_1(s) : s \in (\mathbf{Dist}_R(i, j) \cap N(j)) \setminus R(i, j)\}.$$

That is, k is the most highly ranked vertex in the signal that is not included in $R(i, j)$.

For simplicity of notation, let $r_j = r_\delta(U_j)$. By definition, $U_k \in I_{true}$, so $U_k \geq r_j - d_2 \log(n)^{-1}$. Each $\ell \in R(i, j)$ is a neighbour of i or j, so $U_\ell < r_j$. Moreover, since $\ell \in R(i, j)$ and $k \notin R(i, j)$, it follows from the definition of $R(i, j)$ that $\sigma_1(\ell) > \sigma_1(k)$. We assumed that σ_1 agrees with σ_{true} at precision level d_1. So then either ℓ and k are correctly ordered by σ_1 (in which case $U_\ell > U_k \geq r_j - d_2 \log(n)^{-1}$), or $|U_\ell - U_k| < d_1$ (in which case $U_\ell > U_k - d_1 \geq r_j - d_2 \log(n)^{-1} - d_1$). Since $d_2 < d_1$, it follows that for each $\ell \in R(i, j)$, $r_j - 2d_1 < U_\ell < r_j$. Since we have that $R(i, j) \subset V_1$ and $C_1 \geq p_1 n d_1 \log(n)$ (by (4.9)),

$$\{\mathbf{Dist}_R(i, j) \cap N(j) \not\subset R(i, j)\} \subset \{|\{s \in V_1 : r_j - 2d_1 < U_s < r_j\}| \geq C_1\} \subset \mathcal{A}'{}_6(p_1).$$

We saw immediately following (4.10) that $(\mathcal{A}_6'(p_1))^c$ occurs *w.e.p.*, completing the proof. □

[3] Recall we have assumed that σ_1 is completely determined by $G_1 = G[V_1]$.

From this lemma and Lemma 4.3, we conclude that *w.e.p.*,

$$|\mathbf{Signal}_R(i,j) \cap R(i,j)| = |\mathbf{Signal}_R(i,j)| \geq 2C_2. \tag{4.14}$$

Finally, we show that the noise in $R(i,j)$ is smaller than this lower bound on the signal.

Lemma 4.5. *W.e.p. for all $i, j \in V_2 \setminus V_1$ satisfying* (4.11),

$$\mathbf{Noise}_R(i,j) \leq C_2.$$

Proof. We first show that *w.e.p.* $U_j < U_k$ for all $k \in R(i,j)$. By Assumption 3.5, $w^{(2)}$ is (ϵ,α)-connected, and it follows that there must be an $\epsilon' > 0$ so that $w(x,y) > 0$, and thus $w(x,y) \geq \delta$, for all x, y with $|x-y| \leq \epsilon'$. Consider the interval $J = \{s \in V_1 : U_j < U_s < U_j + \epsilon'\}$. Then for each $s \in J$, $U_s \leq r_\delta(U_j)$, so $w(U_j, U_s) \geq \delta$. If $\mathcal{A}_7(V_1, V_2 \setminus V_1, \sigma_{\text{true}})^c$ and $\mathcal{A}'_6(p_1)^c$ hold, then $|N(j) \cap J| \geq (\delta/2)|J| \geq (\delta/4)\epsilon' p_1 n > C_1$, where the last inequality holds for all n sufficiently large. It follows that *w.e.p.* j has at least C_1 neighbours with U-values greater than U_j, and so $U_j < U_k$ for all $k \in R(i,j)$ as desired.

We now proceed to bound the noise, i.e. the excess number of neighbours of i over the neighbours of j in $R(i,j)$. Since w is diagonally increasing, it follows that $W(i, R(i,j)) \leq W(j, R(i,j)) \leq |R(i,j)| = C_1$. Moreover, by definition of $R(i,j)$, there must be an interval $I \in \mathcal{I}(V_1, \sigma_1)$ so that $R(i,j) = (I \cap N(i)) \cup (I \cap N(j))$. By our choice of parameters (4.9), for all sufficiently large n

$$C_2/2 \geq \sqrt{d_1 p_1 n}\log(n)^6/2 - 1 > 2\sqrt{p_1 d_1 n}\log(n)^5 \geq \sqrt{C_1}\log(n) + \log(n)^2 \tag{4.15}$$

Let $D_i = |N(i) \cap I| - W(i,I)$ and $D_j = W(j,I) - |N(j) \cap I|$. Then

$$\mathbf{Noise}_R(i,j) \leq |N(i) \cap R(i,j)| - |N(j) \cap R(i,j)| = |N(i) \cap I| - |N(j) \cap I| \leq D_i + D_j.$$

We then have

$$\begin{aligned}
\{\mathbf{Noise}_R(i,j) > C_2\}& \\
&\subset \{D_i + D_j \geq C_2\} \\
&\subset \{D_i \geq C_2/2\} \cup \{D_j \geq C_2/2\} \\
&\subset \{D_i \geq \sqrt{C_1}\log(n) + \log(n)^2\} \cup \{D_j \geq \sqrt{C_1}\log(n) + \log(n)^2\} \\
&\subset \mathcal{A}_7(V_1, V_2 \setminus V_1, \sigma_1),
\end{aligned}$$

where (4.15) is used in the third line. The result follows from (4.12). □

This completes the proof of the "correctness" of the first step of Algorithm 7, as summarized in the next lemma. For sets $S \subseteq V$, parameter $d \in (0,1)$, and ordering σ of S, define

$$\mathcal{A}_8(S,\sigma,d) = \bigcup_{i,j \in S,\, |U_i - U_j| > d} \{\mathbf{1}_{\sigma(i)<\sigma(j)} = \mathbf{1}_{\sigma_{\text{true}}(i)>\sigma_{\text{true}}(j)}\}. \tag{4.16}$$

That is, $(\mathcal{A}_8(S,\sigma,d))^c$ holds if σ agrees with σ_{true} at precision level d.

Lemma 4.6. *Suppose parameters $C_1, C_2, p_1, p_2, d_1, d_2$ are chosen according to (4.9). Let V_1, V_2 be sampled from V at rate p_1, p_2, respectively, and assume σ_1 is an ordering of V_1, derived only from $G[V_1]$, which agrees with σ_{true} at precision level d_1. Then, for σ_2 as computed in step 12 of Algorithm 7,*

$$\mathbb{P}[\mathcal{A}_8(V_2 \setminus V_1, \sigma_2, d_2) \,|\, \mathcal{H}[G_1]] = n^{-\Omega(\log n)}.$$

That is, w.e.p. σ_2 is an ordering of $V_2 \setminus V_1$ that agrees with σ_{true} at precision level d_2.

Proof. Suppose $i, j \in V_2 \setminus V_1$ and $|U_i - U_j| > d_2$. Assume *wlog* that $U_i < U_j$. We consider first the case

$$r_\delta(U_j) - r_\delta(U_i) \geq \ell_\delta(U_j) - \ell_\delta(U_i). \tag{4.17}$$

In this case, i, j satisfy (4.11), and we conclude from (4.13), Lemma 4.4 and 4.5 that *w.e.p.*

$$|R(i,j) \cap N(j)| - |R(i,j) \cap N(i)| \geq \mathbf{Signal}_R(i,j) - \mathbf{Noise}_R(i,j) \geq C_2,$$

and thus $F(i,j)$ is set to 1 in Steps 4–9 of Algorithm 7.

If (4.17) does not hold, then there may not be a sufficiently long interval I_{true} to the right of U_j, but by Assumption 1.8 there must be a corresponding interval $I_{true} = [\ell_\delta(U_i), \ell_\delta(U_i) + d_2 \log(n)^{-1}]$ so that $w(x,i) \geq \delta$ and $w(x,j) = 0$ for all $x \in I_{true}$. By applying the arguments above to the values $1 - U_i$, we see then that *w.e.p.* $|L(i,j) \cap N(i)| - |L(i,j) \cap N(j)| \geq C_2$, and thus $F(i,j)$ is set to 1 in Steps 4–9 as above. Therefore, in both cases *w.e.p.* i and j will ordered in σ_2 according to σ_{true}. The result now follows by a union bound. □

Finally, we show that the extension of σ_2 to all of V_2 has the desired precision level.

Lemma 4.7. *Suppose p_1, p_2, d_2, C_3 are as given in (4.9), and let V_1, V_2 be sets sampled from V at rate p_1, p_2, respectively. Suppose σ_2' is an ordering of $V_2 \setminus V_1$ that agrees with σ_{true} at precision level d_2. Then*

$$\mathbb{P}[\bigcup_{i,j \in V_2,\, U_j > U_i + d_2 \log(n)^2} \{t(j) - t(i) < C_3\} \cap \{b(j) - b(i) < C_3\}] = n^{-\Omega(\log(n))}.$$

That is, w.e.p. any ordering of V_2 extending σ_2' and based on the function $F^{(2)}$ as computed in steps 16–20 of Algorithm 7 agrees with σ_{true} at precision level $d_2 \log(n)^2$.

Proof. Fix $i, j \in V_2$ satisfying $U_j \geq U_i + d_2 \log(n)^2$ and (4.11). By the same argument leading to Inequality (4.2), this gives $r_\delta(U_j) - r_\delta(U_i) > 2d_2 \log(n)$. Define

$$I_2(j) = \{s \in V_2 \setminus V_1 \,:\, U_s > r_\delta(U_j) - d_2 \log(n)\}.$$

We will first show that, for all $s \in I_2(j)$, $\sigma_2'(s) > t(i)$. Let $k \in N(i)$ be the vertex at the top of the range of $N(i)$ according to σ_2' - that is, $\sigma_2'(k) = t(i)$. Since $k \in N(i)$, we have that $U_k \leq r_\delta(U_i)$, and so for $s \in I_2(j)$

$$U_s - U_k \geq (r_\delta(U_j) - d_2 \log(n)) - r_\delta(U_i) > d_2.$$

This implies that s and k are correctly ordered by σ_2' and thus $t(i) = \sigma_2'(k) < \sigma_2'(s)$ as desired.

This correct ordering implies that $k \notin I_2(j)$, and so we see that $t(j) - t(i) \geq |I_2(j) \cap N(j)|$, so we have the containment

$$\{t(j) - t(i) < C_3\} \subset \{|I_2(j) \cap N(j)| < C_3\}. \tag{4.18}$$

From (4.9), we know that

$$C_3 \leq \sqrt{d_1 p_1 n}\log(n)^2 = d_2 p_1 n \log(n)^3 \leq d_2 p_2 n.$$

For large enough n, if $(\mathcal{A}_6'(p_2) \cup \mathcal{A}_6'(p_1))^c$ holds then

$$|I_2(j)| \geq \frac{1}{4}(d_2 \log(n))p_2 n > (2/\delta)C_3, \tag{4.19}$$

and if $\mathcal{A}_7(V_2, V_2 \setminus V_1, \sigma_{\text{true}})$ holds, then

$$|I_2 \cap N(j)| \geq \frac{1}{2}W(j, I_2(j)) \geq \frac{\delta}{2}|I_2(j)|. \tag{4.20}$$

Putting together (4.18), (4.19) and (4.20),

$$\{t(j) - t(i) < C_3\} \subset \mathcal{A}_6'(p_2) \cup \mathcal{A}_6'(p_1) \cup \mathcal{A}_7(V_2, V_2 \setminus V_1, \sigma_{\text{true}}).$$

Since we have already shown $\mathbb{P}[\mathcal{A}_6'(p_2) \cup \mathcal{A}_6'(p_1) \cup \mathcal{A}_7(V_2, V_2 \setminus V_1, \sigma_{\text{true}})] = n^{-\Omega(\log(n))}$, it holds *w.e.p.* that $t(j) - t(i) \geq C_3$, and thus $F^{(2)}(i,j)$ is set to 1 in Step 19 of Algorithm 7.

Similarly, if $U_i < U_j$ and the opposite of (4.2) holds, then $\mathbb{P}[\{b(j) - b(i) < C_3\}] \leq n^{-\Omega(\log(n))}$. Therefore, *w.e.p.* $F^{(2)}(i,j)$ is set to 1 in Step 19 of Algorithm 7. The result follows by a union bound. □

The correctness of Algorithm 7 now follows directly from the results in this section.

Lemma 4.8. *Let V_1, V_2 be sets sampled at probabilities p_1, p_2, respectively, according to the variables $\{B_i\}$, and suppose σ_1 is an ordering of V_1 that agrees with σ_{true} at precision level d_1. Suppose Algorithm 7 is executed with parameters C_1, C_2, C_3. Assume that the conditions (4.9) hold. Let $\sigma = Refine(V_1, V_2, \sigma_1)$. Then*

$$\mathbb{P}[\mathcal{A}_8(V_2, \sigma, \sqrt{d_1/(p_1 n)}\log(n))] = n^{-\Omega(\log n)}.$$

That is, w.e.p. σ is a total order of V_2 that agrees with σ_{true} at precision level $\sqrt{d_1/(p_1 n)}\log(n)$.

4.3. Proof of Theorem 2

Finally, we show the correctness of Algorithm 6 and prove Theorem 2. That is, we show that, if Algorithm 6 is executed with a particular set of parameters, and given as input a large enough graph G, then *w.e.p.* the algorithm will return a total order of V with error at most n^{ϵ}, where $\epsilon > 0$ is any desired constant error exponent. The proof will follow directly from Theorem 3 and Lemma 4.8.

First, we define the parameters. For any $0 < \epsilon < 0.5$, let integer k and sequence $\{p_i, d_i\}_{i=1}^{k}$ be as follows:

$$\begin{aligned} k &= \lfloor -\log_2(\epsilon) \rfloor + 1, \\ \beta &= \frac{\epsilon - 2^{-k}}{k}, \\ p_i &= n^{-(k-i)\beta},\ i = 1, \ldots, k, \\ d_1 &= n^{-0.5\,(1-k\beta)}, \\ d_{i+1} &= \sqrt{d_i/(p_i n)}\,\log(n),\ i = 1, \ldots, k-1. \end{aligned} \tag{4.21}$$

The definition of d_2 according to (4.21) above differs by a factor $\log^2(n)$ from the definition of d_2 as given in (4.9); this extra factor corresponds to the extra $\log(n)^2$ appearing in the statement of in Lemma 4.7, which bounds the error introduced when extending the ordering. Note that $\{p_i\}$ is an increasing sequence, with $p_k = 1$. We will use the lemmas from previous sections to show that, after each iteration of Algorithm 6, the returned ordering of the subgraph induced by V_i agrees with the true ordering at precision level d_i. The lemma below will have as corollary that $d_k \ll n^{-1+\epsilon}$, from which our result will follow.

Lemma 4.9. *For the parameters as defined in* (4.21)*, and n sufficiently large, we have that for all $1 \leq i \leq k$,*

$$d_i \leq n^{-(1-2^{-i})(1-k\beta)} \log(n).$$

In particular, there exists $0 < \epsilon' < \epsilon$ so that $d_k \leq n^{-1+\epsilon'}$.

Proof. We prove the statement by induction on i. The statement for $i = 1$ can be directly verified from the definition of d_1. Suppose then that the statement holds for $1 \leq i < k$. Then

$$\begin{aligned} d_{i+1} &= \sqrt{d_i/(p_i n)}\,\log(n) \\ &\leq n^{-0.5\,((1-2^{-i})(1-k\beta)-(k-i)\beta+1)} \log^{3/2}(n). \end{aligned}$$

Now

$$\begin{aligned} 0.5\,((1-2^{-i})(1-k\beta) - (k-i)\beta + 1) &= 0.5\,(2 - 2^{-i} - 2k\beta + i\beta + 2^{-i}k\beta) \\ &= (1 - 2^{-(i+1)} - k\beta + 2^{-(i+1)}k\beta) + 0.5i\beta \\ &\geq (1 - 2^{-(i+1)})(1 - k\beta) + 0.5\beta. \end{aligned}$$

For n large enough, $n^{-0.5\beta} \geq \log^{1/2}(n)$. This completes the proof of the first statement. To prove the second statement, observe first that

$$(1 - 2^{-k})(1 - k\beta) = (1 - 2^{-k})(1 - (\epsilon - 2^{-k}))$$

$$= 1 - \epsilon + 2^{-k}(\epsilon - 2^{-k}).$$

Let $\delta = 2^{-(k+1)}(\epsilon - 2^{-k}) \in (0, \epsilon)$ and $\epsilon' = \epsilon - \delta$. It then follows that

$$d_k \leq n^{-1+\epsilon-2\delta} \log(n) = (n^{-\delta} \log(n)) n^{-1+\epsilon'}.$$

The result then follows for n large enough so that $n^{-\delta}\log(n) \leq 1$. □

In fact, a computation very similar to that used in the above lemma shows that $\{d_i\}_{i=1}^k$ is a decreasing sequence.

In the following, we fix a value $\epsilon > 0$ and take the sequence $\{p_i, d_i\}_{i=1}^k$ as defined in (4.21). We assume the Assumptions of Theorem 2 hold, and for all $i = 1, \ldots, k$ the set $V_i = \{s : B_s \leq p_i\}$ is as in Algorithm 6.

Algorithm 6 first calls Algorithm 1 with input $G_1 = G[V_1]$, where V_1 is sampled from V at rate p_1. The algorithm returns a total order σ_1 of V_1. We will first show that, if p_1, d_1 are as defined above in (4.21), then *w.e.p.* σ_1 agrees with σ_{true} at precision level d_1.

Lemma 4.10. *W.e.p., the ordering σ_1 as returned by Algorithm 1 in Step 3 of Algorithm 6, or its total reverse σ_1^{trev}, agrees with σ_{true} at precision level d_1. That is,*

$$P[\mathcal{A}_8(V_1, \sigma_1, d_1) \cap \mathcal{A}_8(V_1, \sigma_1^{trev}, d_1)] = n^{-\Omega(\log(n))}.$$

Moreover, σ_1 only depends on $G_1 = G[V_1]$.

Proof. From (4.21), we have that $p_1 = n^{-(k-1)\beta}$ and $d_1 = n^{-0.5\,(1-k\beta)}$. Note that V_1 is a set of size $n_1 = |V_1|$ randomly sampled from $[0, 1]$. By Theorem 3, *w.e.p.* the returned ordering σ_1 agrees with σ_{true} or its reverse at precision level $n_1^{-0.5}\log(n_1)^4$. If $\mathcal{A}'_6(p_1)^c$ holds and n is large enough, then $n_1 \geq 0.5\,p_1 n$, and

$$n_1^{-0.5}\log(n_1)^4 \leq \sqrt{2} n^{-0.5(1-(k-1)\beta)} \log(n)^4 = (\sqrt{2} n^{-0.5\,\beta} \log(n)^4) d_1,$$

which is smaller than d_1 for all n sufficiently large. The result follows. □

The next lemma shows that Algorithm 7, when called by 6 to extend the ordering from V_i to V_{i+1}, improves the precision from d_i to d_{i+1}.

Lemma 4.11. *Let $1 \leq i < k$. Suppose σ_i is an ordering of V_i which agrees with σ_{true} at precision level d_i, and which depends only on $G[V_i]$. Let σ_{i+1} be the ordering returned by Algorithm 7 as it is called in Step 6 of Algorithm 6. Then w.e.p. σ_{i+1} agrees with σ_{true} at precision level d_{i+1}.*

Proof. Fix $1 \leq i < k$ and assume σ_i agrees with σ_{true} at precision level d_i. Note that the parameters C_1, C_2, C_3 as set in Step 6 of Algorithm 6 and used in the call to Algorithm 7, satisfy the conditions (4.21), with p_i and d_i taking the role of p_1 and d_1, respectively. Lemma 4.8 then shows that *w.e.p.* $\sigma_{i+1} = Refine(V_i, V_{i+1}, \sigma_i)$ agrees with σ_{true} at precision level $\sqrt{d_i/(p_i n)}\log(n) = d_{i+1}$. □

Finally, we prove Theorem 2:

Proof of Theorem 2 . By definition, $V_k = V$, and Algorithm 6 returns the ordering $\sigma = \sigma_k$ of V. By the previous lemmas and a union bound, *w.e.p.* σ will agree with σ_{true} (or its reverse) at precision level d_k. By Lemma 4.9, $d_k = n^{1-\epsilon'}$, where $0 < \epsilon' < \epsilon$.

The techniques from the previous section readily show that, *w.e.p.*, if an ordering agrees with σ_{true} at precision level d, then it has error at most $dn\log(n)$. Since $n^{-(\epsilon-\epsilon')}\log(n) < 1$ for large enough n, it follows that *w.e.p.* σ has error at most n^{ϵ}. □

5. Finding α

The first part of our results relies on knowing a good thresholding value of $\alpha \in [0,1]$. To implement the algorithm and obtain our guarantees, we need to find a value α for which Assumption 3.5 holds. Fortunately, for the general class of graphons satisfying the stronger conditions of Assumption 1.5, the set of possible values for α will contain an interval of positive measure. We now list a few simple consequences of our properties that allow us to estimate a suitable value of α from the observed graph G. It is straightforward to check that for certain classes of graphons (including *e.g.* (1.4)), these tests will be able to find a suitable value of α; in general, however, the following tests are *necessary* but not *sufficient* for the value of α to be suitable.

We begin with condition (1) of Assumption 3.5, namely the requirement that $w_\alpha^{(2)}$ is diagonally increasing. We have seen that, if $w_\alpha^{(2)}$ is diagonally increasing, then with high probability, a sample of size $\log(n)^5(n)$ from $G_\alpha^{(2)}$ will be a proper interval graph. Such samples are taken repeatedly in Algorithm 2, and these samples are then tested by a proper interval graph recognition algorithm. Thus, if Step 3 of Algorithm 2 fails more than a small percentage of the times, then it is likely that $w_\alpha^{(2)}$ is not diagonally increasing. When this occurs, the algorithm should be restarted with a new value of α.

The second condition is more straightforward: we can test whether $w^{(2)}$ is uniformly (A,δ)- good by testing if

$$\max_{v\in V}\left|\left\{w \in V \,:\, \left|\alpha - \frac{G^{(2)}(w,v)}{n-2}\right| \le \delta'\right\}\right| \le A\delta' n,$$

for several test values of $0 < \delta' < \delta$. Moreover, we can restrict the choice of δ and δ' to values in the range of $\frac{G^{(2)}}{n-2}$.

To test whether $w_\alpha^{(2)}$ is (ϵ,α)-connected, we can use the test

$$\min_{v\in V} |\{w \in V \,:\, G^{(2)}(w,v) \ge \alpha(n-2)\} > \epsilon n.$$

To test whether $w_\alpha^{(2)}$ is B-separated or has an ϵ-split is harder, since we do not have the ordering of the vertices. To test whether $w_\alpha^{(2)}$ is B-separated, we could define for any vertex v the vector

$$s_v[u] = |\{w \in V \,:\, G^{(2)}(v,w) > \alpha(n-2) > G^{(2)}(u,w) \text{ or}$$

$$G^{(2)}(v,w) < \alpha(n-2) < G^{(2)}(u,w)\}.$$

This vector should be close to the volume in the definition of B-separation. Let $\tilde{s}_v$ be the elements of s_v, sorted in increasing order. If $w_\alpha^{(2)}$ is B-separated, then we would expect the inequality

$$\tilde{s}_v[j] - \tilde{s}_v[i] \geq B(j-i)$$

to hold for typical $v \in V$ and $1 \leq i < j \leq n$ (and to be close to true for all such values).

If $w_\alpha^{(2)}$ has an ϵ-split, then pairs of vertices at extreme ends of the ordering cannot have any common neighbours. Precisely, for pairs of vertices i, j with $|U_i - U_j| \geq 1-\epsilon$, $N(i) \cap N(j) = \emptyset$. We expect there to be about $\epsilon^2 n^2/2$ such pairs. On the other hand, if $\inf_{|x|,|y|<\delta} w^{(2)}(U_i + x, U_j + y) > 0$ for some $\delta > 0$, then by the law of large numbers $N(i) \cap N(j) \neq \emptyset$ for sufficiently large n. This suggest that we should check

$$|\{(i,j) \,:\, N(i) \cap N(j) = \emptyset\}| \geq \epsilon^2 n^2/2.$$

It is easy to integrate these tests in the algorithm, so that a suitable value of α can be found in a reasonable amount of trials.

We note that we have not searched for a value of δ that satisfies Assumption 1.8, as we don't need to know δ to run Algorithm 6. In practice, we expect that it is often easy to diagnose graphons that don't satisfy Assumption 1.8 as follows. Let $F^{(2)}$ be as in step (12) of Algorithm 7 on the first time that it is called by Algorithm 6; if Assumption 1.8 is satisfied, then with extreme probability the maximum size of the set on which disagreement occurs

$$\max_{i,j \in V_2 \backslash V_1 \,:\, F^{(2)}(i,j) = -1} |\{k \in V_2 \backslash V_1 \,:\, F^{(2)}(i,k) = F^{(2)}(k,j) = 1\}|$$

will not be much larger than $d_2 n$. In other words, we can diagnose a failure of Assumption 1.8 by simply checking whether $F^{(2)}$ could plausiblY have precision better than d_2 or so.

6. Large error of embeddings

It is common to study the seriation problem using something like the following two-step procedure:

1. Compute an estimate $\hat{U} = (\hat{U}_1, \hat{U}_2, \ldots, \hat{U}_n)$ of the latent positions $U = (U_1, U_2 \ldots, U_n)$ of the vertices $1, 2, \ldots, n$, using *e.g.* a spectral embedding of the graph G.
2. Compute an estimate $\hat{\sigma}$ of σ_{true} based *only* on $\hat{U}$, typically using the formula:

$$\hat{\sigma}(i) = |\{j \in [1:n] \,:\, \hat{U}_j \leq \hat{U}_i\}|.$$

It is then straightforward to show that, if $\hat{U}$ has small error, the error of the induced ordering $\hat{\sigma}$ will inherit a similarly-small error. See *e.g.* [26] for recent work that conducts this sort of analysis.

This approach can only give a near-optimal estimate of the error of $\hat{\sigma}$ if the error of an optimal estimator $\hat{\sigma}$ is not much smaller than the error of an optimal estimator $\hat{U}$ (after appropriate scaling). We will show in this section that this is rarely the case for the seriation problem studied in this paper.

To make this precise, we need some notation that is not used in the rest of the paper. For a graphon w and sequences $u^{(1)} = \{u[j]^{(1)}\}_{j=1}^{n}$, $u^{(2)} = \{u[i,j]^{(2)}\}_{i,j=1}^{n}$ define $G = G(w; u^{(1)}, u^{(2)})$ to be the usual graph obtain from the graphon w and these sequences from the formula (1.2).

Next, for constant $\delta \in [0,1)$ and graphon w, define

$$\tilde{w}_{\delta}(x,y) = w(x^{1-\delta}, y^{1-\delta}).$$

Similarly, for any sequence $u \in \mathbb{R}^d$, define

$$\tilde{u}_{\delta} = u^{1-\delta}.$$

Notice that taking the $(1-\delta)$'th power is a monotone bijection on $[0,1]$, and in particular these transformations preserve both (i) the diagonally-increasing property of w and (ii) the ordering of u. We have the following theorem:

Theorem 4. *Fix $\epsilon > 0$. There exist constants $c = c(\epsilon), c' = c'(\epsilon), c'' = c''(\epsilon) > 0$, and couplings of two sequences $U = (U_1, \ldots, U_n), V = (V_1, \ldots, V_n) \overset{i.i.d.}{\sim} \mathrm{Unif}([0,1])$ such that, for all n sufficiently large, the following events all occur simultaneously with probability at least $(1-\epsilon)$:*

1. *For all graphons w and sequences $u^{(2)} = \{u_{i,j}^{(2)}\}_{i,j=1}^{n}$,*

$$G(w; V, u^{(2)}) = G\left(\tilde{w}_{\frac{c}{\sqrt{n}}}; U, u^{(2)}\right). \tag{6.1}$$

2. *There exists a single permutation $\sigma \in S_n$ such that*

$$U_{\sigma(1)} < U_{\sigma(2)} < \ldots < U_{\sigma(n)},\ V_{\sigma(1)} < V_{\sigma(2)} < \ldots < V_{\sigma(n)}; \tag{6.2}$$

   that is, the two sequences have the same ordering.

3. *Finally,*

$$\left| \left\{ 1 \le i \le n \,:\, |U_i - V_i| \ge \frac{c'}{\sqrt{n}} \right\} \right| \ge c'' n. \tag{6.3}$$

Proof. See Appendix C. □

We give the following informal interpretation: it is possible to couple samples $G, \tilde{G}$ from the two graphons $w, \tilde{w}_{\frac{c}{\sqrt{n}}}$ so that with high probability both (i) the graphs themselves are identical, but (ii) a positive fraction of the "true" latent vertex embeddings are at least $\frac{c'}{\sqrt{n}}$ apart. In particular, since the observed graphs

$G, \tilde{G}$ are identical, there is no way to simultaneously estimate both of "true" latent vertex embeddings without a positive fraction of errors being at least $\frac{c'}{\sqrt{n}}$.

In principle this conclusion could be avoided if *e.g.* w is assumed to belong to a class that does not include $\tilde{w}_{\delta}$, even for very small values of δ. While possible, such an assumption seems to be *extremely* strong - indeed, we have shown that sequences of samples from w, $\tilde{w}_{\frac{c}{\sqrt{n}}}$ can't be told apart by looking at their associated graphs! Furthermore, even if the *particular* perturbation $\tilde{w}_{\delta}$ can be ruled out, a quick inspection of the proof of Theorem 4 will convince the reader that similar results hold for a variety of other small perturbations of w.

Appendix A: Proof of Theorem 1

In this section we show that Theorem 3 implies Theorem 1 by checking that Assumption 1.5 implies Assumption 3.5. We will be using notation from those assumptions throughout this section, so we repeat them for ease of reference. Assumption 1.5:

Assumption A.1. *Let w be a uniformly embedded graphon with link probability function $f : [0,1] \to [0,1]$. In addition, f is decreasing, and there exist constants $0 < d < 0.5$ and $0 \leq c < f(0)$ so that $f(z) = c$ for all $z \geq d$. Finally, there exists $\alpha \in (0,1)$ so that f satisfies*

$$\inf_{s \in [0,d]} \int_0^1 f(z) f(|s-z|) dz > \alpha > \int_0^1 f\left(\left|z - \frac{1-d'}{2}\right|\right) f\left(\left|z - \frac{1+d'}{2}\right|\right) dz \tag{A.1}$$

where $d' = \min\{0.5, 2d\}$.

Assumption 3.5:

Assumptions A.2 (Weak Assumptions)**.** *Let w be a graphon, and $\alpha, \epsilon \in (0,1)$ and $A > 0$ be constants, so that:*

1. *$w_{\alpha}^{(2)}$ is diagonally increasing, and*
2. *$w^{(2)}$ is uniformly (A, ϵ)-good at α, and*
3. *$w^{(2)}$ is (ϵ, α)-connected, and*
4. *$w_{\alpha}^{(2)}$ has an ϵ-split, and*
5. *$w_{\alpha}^{(2)}$ is ϵ-separated.*

Repeating the details of these five component assumptions would take quite a bit of space; we refer to Section 3.1 for details.

A.0.1. Some basic Lemmas

Under Assumption 1.5, w is a uniformly embedded graphon, with decreasing probability function f. We also assume that, for some value $d < 0.5$, $f(x) = c$ for $x \geq d$. In the following, we assume *wlog* that d is minimal with respect to this property, so $f(x) > c$ for $x < d$. We now show that for uniformly embedded graphons, $w^{(2)}$ is continuous.

Lemma A.3. *Let w be a uniformly embedded graphon, with decreasing link probability function f. Then $w^{(2)}$ is uniformly continuous.*

Proof. Fix $0 < \epsilon < 1$, and let $\delta = (\epsilon/12)^2$. Let $D = \{x : f(x) - f(x+\delta) \geq \epsilon/6\}$ be the set of points where f makes a big jump. Since the range of f is contained in $[0,1]$, and f is decreasing, D can include at most $\frac{6}{\epsilon}$ points that are mutually at distance at least δ from each other. Thus, D is contained in the union of at most $\frac{6}{\epsilon}$ intervals of length 2δ, and thus $\mathrm{Vol}(D) \leq (6/\epsilon)(2\delta) = \epsilon/12$. Fix $0 \leq y \leq 1-\delta$, and let

$$D' = \{z < y : y - z \in D\}, \quad D'' = \{z > y : z - y - \delta \in D\}.$$

Then $\mathrm{Vol}(D')$, $\mathrm{Vol}(D'') \leq \mathrm{Vol}(D)$. Note that, for all $0 \leq x \leq y$,

$$\begin{aligned}&|w^{(2)}(x, y+\delta) - w^{(2)}(x,y)|\\ &\leq \int_0^{y-\delta} f(|z-x|)\,(f(y-z) - f(y-z+\delta))\,dz\\ &+ \int_{y+\delta}^1 f(|z-x|)\,(f(z-y-\delta) - f(z-y))\,dz + 2\delta.\end{aligned}$$

Moreover,

$$\begin{aligned}&\int_0^{y-\delta} f(|z-x|)\,(f(y-z) - f(y-z+\delta))\,dz\\ &\leq \int_{D'} f(|z-x|)\,(f(y-z) - f(y-z+\delta))\,dz + \int_{[0,y-\delta]\setminus D'} f(z-x)\frac{\epsilon}{6}dz\\ &\leq \mathrm{Vol}(D') + \left(\frac{\epsilon}{6}\right) y, \text{ and}\\ &\int_{y+\delta}^1 f(|z-x|)\,(f(z-y-\delta) - f(z-y))\,dz \leq \mathrm{Vol}(D'') + \left(\frac{\epsilon}{6}\right)(1-y).\end{aligned}$$

Therefore, $|w^{(2)}(x,y+\delta) - w^{(2)}(x,y)| \leq \mathrm{Vol}(D') + \mathrm{Vol}(D'') + \frac{\epsilon}{6} + 2\delta < \epsilon/2$.

Similarly, we can show that $|w^{(2)}(x,y-\delta) - w^{(2)}(x,y)| < \epsilon/2$. Using the fact that $w^{(2)}$ is symmetric, this shows that $|w^{(2)}(x',y') - w^{(2)}(x,y)| < \epsilon$ whenever $||(x,y) - (x',y')||_\infty < \delta$, so $w^{(2)}$ is uniformly continuous. □

Next we consider an alternative representation of diagonally increasing graphons, which will be helpful in some cases. We saw earlier (see discussion before Equation (3.1)) that, if w is a $\{0,1\}$-valued graphon, then w is diagonally increasing if and only if there exist two non-decreasing boundary functions $\ell : [0,1] \to [0,1]$ and $r : [0,1] \to [0,1]$ which demarcate the regions in $[0,1]^2$ where w assumes the value 1. Precisely, w is of the form

$$w(x,y) = \begin{cases} 1 & \text{if } y \in I(x) \\ 0 & \text{otherwise,} \end{cases} \tag{A.2}$$

where $(\ell(x), r(x)) \subset I(x) \subset [\ell(x), r(x)]$. Thus, the boundaries define w up to the values exactly on points of the form $(\ell(x), x)$ and $(x, r(x))$; of course this collection of points has measure 0, and thus does not influence the distribution of samples from w.

More generally, w is diagonally increasing if and only if, for each $s \in [0,1]$, the $\{0,1\}$-valued graphon w_s is diagonally increasing. Thus, if w is diagonally increasing, then for each $s \in [0,1]$, there exist boundaries ℓ_s and r_s that define the thresholded graphon w_s (up to a set of measure zero) as in (A.2). Let

$$I_s(x) = \{y : w_s(x,y) = 1\} = \{y : w(x,y) \geq s\} \tag{A.3}$$

as given above. This leads to an alternative representation of w:

Lemma A.4. *For any diagonally increasing graphon w,*

$$w(x,y) = \int_0^1 \mathbf{1}_{\{y \in I_s(x)\}} ds. \tag{A.4}$$

Proof. For all $0 \leq x < y \leq 1$, by definition of I_s,

$$w(x,y) = \int_0^1 \mathbf{1}_{\{s \leq w(x,y)\}} ds = \int_0^1 \mathbf{1}_{\{y \in I_s(x)\}} ds. \quad \square$$

The representation of w as given in (A.4) leads to the alternative formula of $w^{(2)}$:

$$\begin{aligned} w^{(2)}(x,y) &= \int_0^1 \left(\int_s \mathbf{1}_{\{z \in I_s(x)\}} ds \right) \left(\int_t \mathbf{1}_{\{z \in I_t(y)\}} dt \right) dz \\ &= \int_s \int_t |I_s(x) \cap I_t(y)| ds dt, \end{aligned} \tag{A.5}$$

where $I_s(x)$, $I_t(y)$ are as defined in (A.3).

In the case of graphons satisfying Assumptions 3.5, the length of the interval $I_s(x)$ is determined only by s. Precisely, for all $s, x \in [0,1]$, $(x - d_s, x + d_s) \subset I_s(x) \subset [x - d_s, x + d_s]$, where

$$d_s = \sup\{\delta \, : \, f(\delta) \geq s\}.$$

Under Conditions 1.5, we also know that $d_s \leq d$ for all $s > c$, and $d_s = 1$ otherwise.

In Lemma A.5, to follow, we use this alternative representation to show a crucial property of $w^{(2)}$. First, we make a simple observation, which will help us to conclude the second half of the lemma.

$$\begin{aligned} w^{(2)}(1-x, 1-y) &= \int_0^1 f(|1-x-z|) f(|1-y-z|) dz \\ &= \int_0^1 f(|\zeta - x|) f(|y - \zeta|) d\zeta \\ &= w^{(2)}(x,y). \end{aligned} \tag{A.6}$$

Lemma A.5. *Let w be a graphon satisfying Conditions 1.5, and let f, d, c, α be as in the statement of these conditions. Then for all $x \in [0,1]$,*

(i) $w^{(2)}(x,\cdot)$ is decreasing on $[x+d,1]$,
(ii) for all $x+d < y' < y < x+2d$,

$$(y-y') \geq w^{(2)}(x,y') - w(x,y) \geq (y-y')\frac{1}{2}\left(f(\frac{y-x}{2}) - c\right)^2. \tag{A.7}$$

In particular, $w^{(2)}(x,\cdot)$ is strictly decreasing on $(x+d, x+2d)$.
(iii) $w^{(2)}(x,\cdot)$ is increasing on $[0, x-d]$.
(iv) For all $x-2d < y < y' < x-d$,

$$(y'-y) \geq w^{(2)}(x,y') - w(x,y) \geq (y'-y)\frac{1}{2}\left(f(\frac{x-y'}{2}) - c\right)^2. \tag{A.8}$$

In particular, $w^{(2)}(x,\cdot)$ is strictly increasing on $(x-2d, x-d)$.

Moreover, since w is symmetric, equivalent statements hold for $w^{(2)}(\cdot,x)$.

Proof. Inspecting the alternative representation given in (A.5), we see that the contribution to $w^{(2)}(x,y)$ due to any particular pair of values s,t is determined by the size of the intersection $I_s(x) \cap I_t(y)$. We can derive the size of this intersection from the values of d_s, d_t. Fix $x \in [0, 1-d)$, and consider $w^{(2)}(x,y)$ as a function of $y \in [x,1]$. For $s,t \in [0,1]$, define

$$\phi_{s,t}(y) = |I_s(x) \cap I_t(y)|. \tag{A.9}$$

The set $I_s(x) \cap I_t(y)$ has measure zero if and only if $x + d_s \leq y - d_t$ (i.e. $d_s + d_t \leq y - x$). If $|I_s(x) \cap I_t(y)| > 0$, then

$$\phi_{s,t}(y) = |I_s(x) \cap I_t(y)| = \min\{x+d_s, y+d_t, 1\} - \max\{x-d_s, y-d_t, 0\}.$$

Thus $\phi_{s,t}$ is piecewise linear, with each of the linear pieces having slope in $\{-1, 0, 1\}$.

We are almost ready to prove that $w^{(2)}(x,\cdot)$ is decreasing on $[x+d,1]$, based on the following proposition:

Proposition A.6. *Fix $x \in [0,1]$. Then for all $s,t \in [0,1]$, $\phi_{s,t}$ has non-positive slope for almost every $y \in (x+d, 1]$.*

Proof. For $y \in (x,1]$, define

$$R^+(y) = \{(s,t) \,:\, y < d_t < 1-y,\, d_s > d_t + (y-x)\}.$$

Suppose that for some $s,t \in [0,1]$, y lies in the interior of a region on which $\phi_{s,t}$ has slope 1; we will check that $(s,t) \in R^+(y)$. So see this, note that, if $\phi_{s,t}$ has slope 1 at y, then the left boundary of $I_s(x) \cap I_t(y)$ cannot be given by $y - d_t$, while the right boundary must be $y + d_t$. The right boundary condition (and the fact that we are in the interior of a region with slope 1) implies that

$y + d_t < 1$ and $y + d_t < x + d_s$. This implies that $d_s > d_t + (y - x) \geq d_t$, and thus $x - d_s \leq y - d_t$. Thus we must have that the left boundary of $I_s(x) \cap I_t(y)$ equals 0, and thus $y - d_t < 0$, and so $(s, t) \in R^+(y)$.

Next, we check that, if $y > x + d$, then $R^+(y) = \emptyset$. As argued earlier, it follows from Conditions 1.5 that for all s, $d_s \leq d$ or $d_s = 1$ otherwise. Suppose then that $y - x > d$ and $(s, t) \in R^+(y)$. By definition, $d_s > d_t + (y - x) > d$ and thus $d_s = 1$. But then, $d_t < 1 - y < 1$ so $d_t \leq d \leq x + d < y$, which contradicts the condition $y > x + d$.

Combining the conclusions in these two paragraphs completes the proof. □

By Proposition A.6 if $x + d < y' < y$, then $\phi_{s,t}(y) - \phi_{s,t}(y') \leq 0$, and consequently,

$$w^{(2)}(x, y) - w^{(2)}(x, y') = \int_s \int_t \phi_{s,t}(y) - \phi_{s,t}(y') ds dt \leq 0.$$

This shows part (i) of Lemma A.5, namely that $w^{(2)}(x, \cdot)$ is decreasing on $[x + d, 1]$.

We will next prove Inequality (A.7). The lower bound follows immediately from the fact that for all s, t, $\phi_{s,t}$ has slope at least -1. To prove the upper bound, we begin by defining

$$R^-(z) = \{(s, t) \, : d_t \leq d_s \leq d, d_s + d_t \geq z - x\}$$

for all $z \geq x$. Note that $R^-(y) \subset R^-(z)$ for all $y > z$.

We now check that

$$\phi_{s,t}(y) - \phi_{s,t}(y') = -(y - y') \tag{A.10}$$

holds for all $x + d < y' < y < x + 2d$ and $(s, t) \in R^-(y)$. We will do this by showing that, for all $(s, t) \in R^-(y)$, $\phi_{s,t}$ has slope -1 at any point $z \in [y', y]$. Fix $(s, t) \in R^-(y)$ and $z \in [y', y]$. Then $(s, t) \in R^-(z)$. The condition $d_s + d_t \geq z - x$ in the definition of $R^-(z)$ guarantees that $I_s(x) \cap I_t(z)$ is non-empty. Since $d_s \geq d_t$ and $x < z$, $x - d_s < z - d_t$. Since $z > x + d \geq d \geq d_t$, $z - d_t > 0$. Thus, the left boundary of $I_s(x) \cap I_t(z)$ equals $z - d_t$. On the other hand, $z + d_t \geq z > x + d \geq x + d_s$ so the right boundary is not given by $z + d_t$. Thus, $\phi_{s,t}$ has slope -1 at z. This implies that $\phi_{s,t}$ is linear with slope -1 on the interval $[y', y]$, which implies (A.10).

Finally, we bound the size of $R^-(y)$ for $x + d < y < x + 2d$. Fix $y \in (x + d, x + 2d)$, and define $c_1 \equiv f(\frac{y-x}{2})$. In this case, $(y - x)/2 < d$ and thus $c_1 > c$. Then for all $c_1 \geq t \geq s > c$ and for all $y' \leq z \leq y$, $(y - x)/2 \leq d_t \leq d_s < d$, and thus $(s, t) \in R^-(y)$. Therefore, $|R^-(z)| \geq (c_1 - c)^2/2$.

Applying Equations (A.5) and (A.9) and then Inequality (A.10), we have for $x + d < y' < y < x + 2d$,

$$\begin{aligned} w^{(2)}(x, y) - w^{(2)}(x, y') &= \int_s \int_t (\phi_{s,t}(y) - \phi_{s,t}(y')) ds dt \qquad \text{(A.11)} \\ &\leq -(y - y')|R^-(y)| \end{aligned}$$

$$\leq -\frac{1}{2}(f(\frac{y-x}{2})-c)^2(y-y').$$

In particular, $w^{(2)}(x,\cdot)$ is strictly decreasing on $(x+d, x+2d)$. This proves the upper bound in (A.7), completing our proof of item (ii) of the lemma.

Due to the symmetry shown in (A.6) we have that $(iii),(iv)$ follow immediately from $(i),(ii)$ respectively. □

Finally, we establish a general lemmas about the behaviour of $w^{(2)}(x,y)$ when w is uniformly embedded and diagonally increasing.

Lemma A.7. *Let w be a uniformly embedded graphon, with decreasing link probability function f. Fix $\delta \in [0,\frac{1}{2}]$. Then for all $x \in [0,1-\delta]$,*

$$w^{(2)}(0,\delta) \leq w^{(2)}(x,x+\delta) \leq w^{(2)}(\frac{1-\delta}{2},\frac{1+\delta}{2}). \tag{A.12}$$

Proof. As shown in Equation (A.6), we have that $w^{(2)}(1-x,1-y) = w^{(2)}(x,y)$ for all x,y. Thus we may assume without loss of generality that $x \leq 1-(x+\delta)$, so $x \in [0,\frac{1}{2}-\frac{\delta}{2}]$. According to the definition, and making the appropriate substitutions,

$$\begin{aligned} w^{(2)}(x,x+\delta) &= \int_0^1 f(|x-z|)f(|x+\delta-z|)dz \\ &= \int_0^x f(s)f(s+\delta)ds + \int_0^\delta f(s)f(\delta-s)ds + \int_0^{1-\delta-x} f(s)f(s+\delta)ds. \end{aligned}$$

Note that the middle integral does not depend on x, while the first and last integral are taken over the same function. Since f is decreasing, so is $f(s)f(s+\delta)$ (taken as a function of s). Therefore, this expression is maximized when $x = 1-\delta-x$, so $x = \frac{1-\delta}{2}$, and minimized when $x = 0$. □

A.0.2. Assumption 1.5 implies Assumption 3.5

We start by showing that Assumption 1.5 implies that $w_\alpha^{(2)}$ is diagonally increasing.

Remark A.8. It is not true that $w^{(2)}$ is diagonally increasing for all diagonally-increasing w. We consider as simple counterexamples some graphons of the form (1.4), which are uniformly embedded with link probability function

$$f(x) = \begin{cases} p, & 0 \leq x \leq d, \\ q, & d < x \leq 1. \end{cases}$$

We consider parameters $0 \leq q < p \leq 1$, and $0 < d < \frac{1}{2}$. Then, for $y \in [0,d]$, $w^{(2)}(0,y) = q^2 + (p^2-q^2)d + q(p-q)y$. This function is increasing as y moves away from 0 and thus is further removed from the diagonal. This contradicts the defining property of diagonally increasing graphons as given in (1.3).

Even if $w^{(2)}$ itself is not diagonally increasing, there may still exist a value α so that $w^{(2)}_\alpha$ is diagonally increasing. In the above example, for $y \in [d, 2d]$, $w^{(2)}(0, y)$ is linearly decreasing with slope $-(p-q)^2$, while for $y \in [2d, 1-d]$, $w^{(2)}(0, y)$ is constant equal to $q^2 + 3dq(p-q)$. If there exists a value $w^{(2)}(0,0) > \alpha > w^{(2)}(0, 2d)$ then $w^{(2)}_\alpha(0, \cdot)$ is decreasing, even if $w^{(2)}(0, \cdot)$ is not. However, it is possible to choose values for p and q (e.g. $p = 1/3$, $q = 1/6$, $d = 0.3$) for which $w^{(2)}(0, 2d) \geq w^{(2)}(0, 0)$, and thus such a value α does not exist.

We now show that, for a graphons w satisfying our assumptions, $w^{(2)}_\alpha$ is a diagonally increasing graphon.

Lemma A.9. *Let w and α be a graphon and constant satisfying Conditions 1.5. Then $w^{(2)}_\alpha$ is diagonally increasing. Moreover, $w^{(2)}_\alpha(x, y) = 1$ if and only if $y \in [\ell_\alpha(x), r_\alpha(x)]$.*

Proof. Fix $x \in [0, 1]$. By Lemma A.3, $w^{(2)}$ is continuous, so $w^{(2)}(x, r_\alpha(x)) = \alpha$. By Assumptions 1.5 and Lemma A.7, we have that for all $0 \leq \delta \leq d$,

$$w^{(2)}(x, x+\delta) \geq w^{(2)}(0, \delta) > \alpha,$$

and thus $r_\alpha(x) > x + d$. By Lemma A.5, $w^{(2)}(x, \cdot)$ is decreasing on $[x + d, 1]$, and thus $w^{(2)}(x, y) \geq w^{(2)}(x, r_\alpha(x)) = \alpha$ for $y \in [d, r_\alpha(x)]$. Thus for $y \geq x$, $w^{(2)}(x, y) \geq \alpha$ if and only if $y \leq r_\alpha(x)$.

An analogous argument applied to $\ell_\alpha(x)$ shows that, for $y \leq x$, $w^{(2)}(x, y) \geq \alpha$ if and only if $y \geq \ell_\alpha(x)$. This completes the proof. □

The following lemma shows that $r_\alpha(x) - x$ and $x - \ell_\alpha(x)$ are bounded well away from their lower and upper bounds d and d'.

Lemma A.10. *Let w be a graphon satisfying Conditions 1.5, and let f, d, c, α be as in the statement of these conditions. Let*

$$\begin{aligned} d^- &= \inf\{r_\alpha(x) - x \,:\, x \in [0, \ell_\alpha(1)]\}, \ \textit{and} \\ d^+ &= \sup\{r_\alpha(x) - x \,:\, x \in [0, 1]\}\}. \end{aligned} \tag{A.13}$$

Then $d < d^- \leq d^+ < d'$, where $d' = \min\{2d, 0.5\}$ as in Assumption 1.5.

Proof. By our choice of α, we have that

$$r_\alpha(x) \geq \min(x + d, 1) \tag{A.14}$$

for all $x \in [0, 1]$. Let $\gamma = \inf\{w^{(2)}(0, s) \,:\, 0 \leq s \leq d\}$. By Assumption 1.5,

$$\gamma = \inf_{s\in[0,d]} w^{(2)}(0, s) = \inf_{s\in[0,d]} \int_0^1 f(z)f(|s - z|)dz > \alpha.$$

By Lemma A.7, this implies that, for all $x \in [0, 1-d]$, $w^{(2)}(x, x + d) \geq w^{(2)}(0, d) \geq \gamma > \alpha$.

By Lemma A.3, $w^{(2)}$ is uniformly continuous. Let $0 < \epsilon < \gamma - \alpha$, and let $\delta > 0$ be so that $w^{(2)}(x, y) < \epsilon$ whenever $|x - y| < \delta$. Then for all $x \in [0, 1 - d - \delta/2]$,

$$w^{(2)}(x, x + d + \delta/2) > w^{(2)}(x, x + d) + \epsilon \geq \gamma + \epsilon > \alpha.$$

Therefore, $r_\alpha(x) \geq x + d + \delta/2$ for all $x \in [0, 1 - d - \delta/2]$. Moreover, since $w^{(2)}(1 - d - \delta/2, 1) > \alpha$, $\ell_\alpha(1) < 1 - d - \delta/2$. By compactness, $d^- \geq d + \delta/2 > d$.

Let $\gamma' = w^{(2)}(\frac{1-d'}{2}, \frac{1+d'}{2})$. By Assumption 1.5,

$$\gamma' = \int_0^1 f(|z - \frac{1-d'}{2}|) f(|z - \frac{1+d'}{2}|) dz < \alpha.$$

By Lemmas A.7 and A.5, we have that for all $x \in [0, 0.5]$,

$$w^{(2)}(x, x + d') \leq w^{(2)}(\frac{1-d'}{2}, \frac{1+d'}{2}) = \gamma' < \alpha,$$

and thus $r_\alpha(x) < x + d'$. By compactness, $d^+ \leq d'$. As argued in the previous case, since $w^{(2)}(x, r_\alpha(x)) = \alpha$ and $w^{(2)}$ is continuous, we have that $d^+ < d'$. □

Lemma A.11. *Let w and α be a graphon and constant satisfying Conditions 1.5. Then there exists an $\epsilon > 0$ so that $w_\alpha^{(2)}$ is (ϵ, α)-connected.*

Proof. Define d^- as in the statement of Lemma A.10 and take $0 < \epsilon \leq d^-$. If $0 \leq x \leq y \leq 1$ and $y - x < \epsilon$, then either $x \in [0, \ell_\alpha(1)]$ and $y < x + \epsilon \leq r_\alpha(x)$, or $x > \ell_\alpha(1)$. In both cases, $w^{(2)}(x, y) > \alpha$. Since $w^{(2)}$ is continuous (Lemma A.3), the result follows. □

Lemma A.12. *Let w and α be a graphon and constant satisfying Conditions 1.5. Then there exists an $\epsilon > 0$ so that so that $w_\alpha^{(2)}$ has an ϵ-split.*

Proof. Define d^+ as defined in (A.13) and let $\epsilon = 0.5 - d^+$. Fix $x < y$ so that $y - x \geq 1 - \epsilon$. Then

$$r_\alpha(x) + \epsilon \leq x + d^+ + \epsilon \leq y - (1 - \epsilon) + d^+ + \epsilon = y - 1 + 2\epsilon + d^+ = y - d^+ \leq \ell_\alpha(y),$$

and thus $r_\alpha(x) < \ell_\alpha(y)$. Thus $w_\alpha^{(2)}$ has an ϵ-split. □

Lemma A.13. *Let w be a graphon satisfying Conditions 1.5, and let f, d, c, α be as in the statement of these conditions. Then there exist $A, \delta > 0$ so that $w^{(2)}$ is uniformly (A, δ)-good at α.*

Proof. We need to show that there exist constants $\delta, A > 0$ so that, for all $x \in [0, 1]$ and $0 < \delta' < \delta$, the size of the set where $w^{(2)}$ takes values within δ' of α is bounded by $A\delta'$.

Let d^-, d^+ be as in Equation (A.13). By our earlier assumption on the minimality of d (see beginning of this Section A.0.1), we have that $f(x) > c$ whenever $x < d$. Let

$$\beta = \frac{1}{2}(f(d^-/2) - c)^2. \tag{A.15}$$

We first show that $\beta > 0$. By Lemma A.10, $d^- < d' \leq 2d$, and thus $d^-/2 < d$. It follows from our assumptions that $f(d^-/2) > c$, so $\beta > 0$.

By Lemma A.5 and the fact that f is decreasing, for all $x \in [0,1]$, $x + d^- \leq y < x + 2d$, and all $x + d < y' < y$,

$$w^{(2)}(x,y') - w^{(2)}(x,y) \geq (y-y')\frac{1}{2}\left(f\left(\frac{y-x}{2}\right) - c\right)^2 \geq \beta(y-y').\text{(A.16)}$$

Let $\epsilon > 0$ be so that $d^- - \epsilon > d$ and $d^+ + \epsilon < d' \leq 2d$; such a value exists by Lemma A.10. Fix $0 < \epsilon' < \epsilon$. Taking $y = r_\alpha(x)$ and $y' = r_\alpha(x) - \epsilon'$ in (A.16), we have that

$$w^{(2)}(x, r_\alpha(x) - \epsilon') \geq w^{(2)}(x, r_\alpha(x)) + \beta\epsilon' = \alpha + \beta\epsilon', \tag{A.17}$$

where the equality uses the fact that $w^{(2)}$ is uniformly continuous (see Lemma A.3). Similarly, taking $y = r_\alpha(x) + \epsilon'$ and $y' = r_\alpha(x)$, we obtain that

$$w^{(2)}(x, r_\alpha(x) + \epsilon') \leq \alpha - \beta\epsilon'. \tag{A.18}$$

The above, together with the fact that $w^{(2)}$ is diagonally increasing shows that, for values y outside of the interval $(r_\alpha(x) - \epsilon, r_\alpha(x) + \epsilon)$, $w^{(2)}(x,y)$ takes values that differ from α by at least $\beta|x-y|$. This leads to our definition of δ and A for which $w^{(2)}$ is uniformly (A,δ)-good at α.

Namely, let $\delta = \epsilon\beta$ and $A = 2/\beta$. Fix any $0 < \delta' < \delta$. By setting $\epsilon' = \delta'/\beta$, we can conclude from (A.17) and (A.18) that

$$\{y \in [0,1] \,:\, |\alpha - w(x,y)| \leq \delta'\} \subseteq \{y \,:\, |r_\alpha(x) - y| < \delta'/\beta\}.$$

The conclusion follows since the superset has volume at most $2(\delta'/\beta) = A\delta'$. □

The fact that $w_\alpha^{(2)}$ is diagonally increasing implies that the boundaries r_α and ℓ_α are increasing. The following lemma establishes bounds on their slope.

Lemma A.14. *Let w be a graphon satisfying Conditions 1.5, and let f, d, c, α be as in the statement of these conditions. Then the boundary r_α is strictly increasing on the domains $[0, \ell_\alpha(1)]$, and has slope bounded away from zero.*

Precisely, there exists $\epsilon > 0$ so that for all $0 \leq x' < x < \ell_\alpha(1)$ with $x - x' < \epsilon$,

$$r_\alpha(x) - r_\alpha(x') \geq \beta(x - x'),$$

where $\beta > 0$ is as defined in (A.15).

Similarly ℓ_α is strictly increasing on $[r_\alpha(0), 1]$, and there exists $\epsilon > 0$ so that, for all $r_\alpha(0) \leq y' < y < 1$ with $y - y' < \epsilon$,

$$\ell_\alpha(y) - \ell_\alpha(y') \geq \beta(y - y').$$

Proof. We show first that ℓ_α is strictly increasing on the domain $[r_\alpha(0), 1]$. Suppose, by contradiction, that $\ell_\alpha(y_1) = \ell_\alpha(y_2) = x$ for some $r_\alpha(0) \leq y_1 < y_2$. Then $w^{(2)}(x, y_1) = w^{(2)}(x, y_2) = \alpha$, and thus $w^{(2)}(x, \cdot)$ is constant on $[y_1, y_2]$.

This contradicts Lemma A.13. A similar argument shows that r_α is strictly increasing on the domain $[0, \ell_\alpha(1)]$.

Let d^-, d^+ be as in (A.13). By Lemma A.10, we conclude that for all $x \in [0, \ell_\alpha(1)]$,

$$x + d < x + d^- \leq r_\alpha(x) \leq x + d^+ < x + d'.$$

Fix $0 \leq x' < x \leq \ell_\alpha(1)$ so that $x - x' < d^- - d$, and thus $x + d < x' + d^-$. Let $y = r_\alpha(x)$ and $y' = r_\alpha(x')$. Since r_α is strictly increasing, $y' < y$. Then

$$x + d < x' + d^- \leq y' \leq y \leq x + d^+ < x + d'.$$

Note further that $y', y \in [r_\alpha(0), 1]$, and $x = \ell_\alpha(y)$, $x' = \ell_\alpha(y')$. Lemma A.5 gives bounds on the slope of $w^{(2)}(x, \cdot)$ on the interval $[x + d, x + d']$, and the above shows that y, y' are in this interval. Therefore the bounds apply, and we have that

$$\begin{aligned} w^{(2)}(x, y') - \alpha &= w^{(2)}(x, y') - w^{(2)}(x, y) \geq \beta(y - y'), \text{ and} \\ w^{(2)}(x, y') - \alpha &= w^{(2)}(x, y') - w^{(2)}(x', y') \leq (x - x'). \end{aligned}$$

Combining these inequalities we obtain that $\ell_\alpha(y) - \ell_\alpha(y') = x - x' \geq \beta(y - y')$. This completes the proof that ℓ_α has slope at least β on $[r_\alpha(0), 1]$.

A similar argument shows that r_α is increasing with slope at least β on $[0, \ell_\alpha(1)]$. □

Lemma A.15. *Let w be a graphon satisfying Conditions 1.5, and let f, d, c, α be as in the statement of these conditions. Then there exists $B > 0$ so that $w_\alpha^{(2)}$ is B-separated.*

Proof. Fix $0 \leq x < y \leq 1$. We need to show that there exists a constant $B > 0$ so that the region where $w_\alpha^{(2)}(x, \cdot)$ and $w_\alpha^{(2)}(y, \cdot)$ take different values is bounded below by $B(y - x)$.

For any $z \in [0, 1]$, $w_\alpha^{(2)}(z, x) = 1$ and $w_\alpha^{(2)}(z, y) = 0$ precisely when $\ell_\alpha(x) \leq z < \ell_\alpha(y)$. Similarly, $w_\alpha^{(2)}(z, x) = 0$ and $w_\alpha^{(2)}(z, y) = 1$ precisely when $r_\alpha(x) \leq z < r_\alpha(y)$. We have that

$$\begin{aligned} \mathrm{Vol}(\{z \in [0,1] \,:\, |w_\alpha^{(2)}(z, x) - w_\alpha^{(2)}(z, y)| \geq 1\}) &\geq \max(\ell_\alpha(y) - \ell_\alpha(x), r_\alpha(y) - r_\alpha(x)) \\ &\geq \frac{1}{2}(\ell_\alpha(y) - \ell_\alpha(x) + r_\alpha(y) - r_\alpha(x)). \end{aligned}$$

Let d^-, d^+ be as in (A.13) and fix $0 < B \leq \min\{\beta, 2(1 - 2d^+)\}$; this is possible because $d^+ < d' \leq 0.5$. We now repeatedly apply Lemma A.14. If $y > x > r_\alpha(0)$, then $\ell_\alpha(y) - \ell_\alpha(x) \geq \beta(y - x) \geq B(y - x)$. If $x < y < \ell_\alpha(1)$, then $r_\alpha(y) - r_\alpha(x) \geq \beta(y - x) \geq B(y - x)$. If $x \leq r_\alpha(0)$ and $y \geq \ell_\alpha(1)$, then

$$\begin{aligned} \ell_\alpha(y) - \ell_\alpha(x) + r_\alpha(y) - r_\alpha(x) &= \ell_\alpha(y) - 0 + 1 - r_\alpha(x) \\ &\geq (y - d^+) + 1 - (x + d^+) \end{aligned}$$

$$
\begin{aligned}
&\geq (1 - 2d^{+}) + 1 - (2d^{+}) \\
&= 2(1 - 2d^{+}) \geq B \geq B(y - x).
\end{aligned}
$$

Therefore, $w_{\alpha}^{(2)}$ is B-separated. □

Combining the above results, we can conclude

Lemma A.16. *Let w and α be a graphon and constant satisfying Conditions 1.5. Then Assumption 3.5 also holds for w, α.*

Proof. Let $A > 0$, $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4 \in (0, 1)$ be so that w is (A, ϵ_1)-good at α, and $w_{\alpha}^{(2)}$ is ϵ_2-separated, has an ϵ_3-split, and is (ϵ_4, α)-connected. Such values exist by the earlier lemmas in this section. Taking $\epsilon = \min\{\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4\}$ completes the proof. □

We are ready to prove our the first of our main results:

Proof of Theorems 1. As noted earlier, Theorem 3 follows immediately from Lemma 3.22. Theorem 1 then follows immediately from Theorem 3 and Lemma A.16, which relates their assumptions. □

Appendix B: Bad events are rare: proofs

We give proofs deferred from Section 3.3. This appendix will use the notation from Section 3.3 freely, and in particular it will rely on the relationship between the observed random graph G_n and the driving randomness $\{U_i\}$, $\{U_{ij}\}$.

B.1. Bad events for latent variables

Section 3.3 introduces the collection of "bad" events $\mathcal{A}_1, \ldots, \mathcal{A}_5$ in Definition 3.6. We recall the definition for convenience:

Definition B.1 (Bad Events)**.** Let graphon w, value α, random variables $\{U_i\}$, $\{U_{i,j}\}$, and graphs $G_{\alpha}^{(2)}$ and H_{α} be as defined above. Moreover, let R, $\{I_k\}_{k=0}^{4R-3}$ be as in (3.5). We define:

$$
\begin{aligned}
\mathcal{A}_1 &= \cup_{1 \leq i < j \leq n} (\{(U_i, U_j) \notin \mathcal{B}(n, \alpha)\} \cap \{G_{\alpha}^{(2)}(i, j) \neq H_{\alpha}(i, j)\}) \\
\mathcal{A}_2 &= \left\{ \max_{1 \leq i \leq n} |\mathcal{B}(n, i, \alpha)| > \sqrt{n} \, \log(n)^2 \right\} \\
\mathcal{A}_3 &= \left\{ \min_{0 \leq k \leq 4R-3} |\{i \,:\, U_i \in I_k\}| < \frac{n}{2R} \right\} \\
\mathcal{A}_4 &= \left\{ \min_{1 \leq i,j \leq n \,:\, |U_i - U_j| > \frac{1}{\log(n)}} \nu(i, j) < \frac{2n}{\log(n)^2} \right\} \\
&\quad \cup \left\{ \min_{1 \leq i,j \leq n \,:\, |U_i - U_j| > \frac{\log(n)^4}{\sqrt{n}}} \nu(i, j) < \sqrt{n} \log(n)^3 \right\}
\end{aligned}
$$

$$\mathcal{A}_5 = \left\{ \max_{1 \leq i < j \leq n} |\{k \,:\, U_k \in (U_i, U_j)\}| \geq n|U_i - U_j| + \sqrt{n}\log(n) \right\}.$$

In this section we prove Corollary 3.7, which says that with high probability none of these events occur. Again, we repeat the statement for convenience:

Corollary B.2. *Let $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4 \cup \mathcal{A}_5$. Then $\mathcal{A}^c$ holds w.e.p..*

The proofs mainly follow from standard concentration bounds, applied to functions of the independent variables $\{U_i\}$ and $\{U_{i,j}\}$. See *e.g.* [4] for an overview of concentration phenomena.

We begin by checking that the "approximate" thresholded graph $G_\alpha^{(2)}$ agrees with the "correct" thresholded graph $H_\alpha \sim w_\alpha^{(2)}$ outside of the bad set defined in Equation (3.3):

Lemma B.3. *Fix $0 < \alpha < 1$ and $n \in \mathbb{N}$. Then $\mathbb{P}[\mathcal{A}_1] = n^{-\Omega(\log(n))}$. That is, w.e.p. for all i, j such that $(U_i, U_j) \notin \mathcal{B}(n, \alpha)$, $G_\alpha^{(2)}(i, j) = H_\alpha(i, j)$.*

Proof. Recall the definition of the square-threshold graph $G_\alpha^{(2)}$ from Equation (2.1). For $0 \leq a \neq b \leq n$, define $\ell_{a,b} = \mathbf{1}_{\{U_{a,b} < w(U_a, U_b)\}}$ to be the indicator function for the edge (a, b) being in G. With this notation we have

$$G^{(2)}(1,2) = \sum_{a=3}^{n} \ell_{1,a}\ell_{a,2},$$

Recall that $G_\alpha^{(2)}(1,2) = 1$ if and only the above sum is greater than $\alpha(n-2)$. Let $\mathcal{H}$ be the σ-algebra generated by (U_1, U_2); for all $a \geq 3$, $\mathbb{E}[\ell_{1,a}\ell_{a,2} \,|\, \mathcal{H}] = w^{(2)}(U_1, U_2)$. For $(U_1, U_2) \in \mathcal{B}(n, \alpha)$, the bound is clear. In the remaining case, we assume that $(U_1, U_2) \notin \mathcal{B}(n, \alpha)$, and thus $|w^{(2)}(U_1, U_2) - \alpha| > \frac{\log(n)}{\sqrt{n}}$. If $w^{(2)}(U_1, U_2) = 0$, then $G_\alpha^{(2)}(1,2) = H_\alpha(1,2) = 0$, so we also assume $w^{(2)}(U_1, U_2) > 0$. By Hoeffding's bound on the sum of independent variables,

$$\begin{aligned}
&\mathbb{P}[G_\alpha^{(2)}(1,2) \neq H_\alpha(1,2) | \mathcal{H}] \\
&\qquad \leq \mathbb{P}[| \sum_{a=3}^{n} \ell_{1,a}\ell_{a,2} - (n-2) w^{(2)}(U_1, U_2)| > \log(n)\sqrt{n} \,|\, \mathcal{H}] \\
&\qquad \leq \mathbb{P}[| \sum_{a=3}^{n} \ell_{1,a}\ell_{a,2} - \mathbb{E}[\sum_{a=3}^{n} \ell_{1,a}\ell_{a,2} \,|\, \mathcal{H}]| > \log(n)\sqrt{n-2} \,|\, \mathcal{H}] \\
&\qquad = n^{-\Omega(\log(n))},
\end{aligned}$$

Adding back in the trivial cases discounted above, we have the unconditional bound

$$\{G_\alpha^{(2)}(1,2) = H_\alpha(1,2)\} \cup \{(U_1, U_2) \in \mathcal{B}(n, \alpha)\}$$

holds *w.e.p.*. There is nothing special about the indices $(1,2)$, so we have shown that *w.e.p.*, $G_\alpha^{(2)}(i,j) = H_\alpha(i,j)$ for any choice of $1 \leq i < j \leq n$. Therefore, *w.e.p.* the same holds for all pairs i,j simultaneously. This completes the proof. □

If Assumption 3.5 (item 4) holds, then there exist $A, \delta > 0$ so that $w^{(2)}$ is uniformly (A,δ)-good at α. For n large enough, $\frac{\log(n)}{\sqrt{n}} < \delta$ and $8A \leq \log(n)$, so for all $y \in [0,1]$,

$$Vol(\{y \in [0,1] \,:\, |\alpha - w^{(2)}(x,y)| \leq \frac{\log(n)}{\sqrt{n}}\}) \leq A\frac{\log(n)}{\sqrt{n}} \leq \frac{\log(n)^2}{8\sqrt{n}}.$$

Moreover, since w is diagonally increasing, for each $x \in [0,1]$, the set $\{y \in [0,1] \,:\, |\alpha - w^{(2)}(x,y)| \leq \frac{\log(n)}{\sqrt{n}}\}$ is the union of two intervals, one each around the endpoints $\ell_\alpha(x), r_\alpha(x)$ of the interval $I_\alpha(x)$ associated to w_α in Equation (A.2). This implies that, for all $x \in [0,1]$,

$$\mathcal{B}(n,\alpha) \subset \{(x,y) : |r_\alpha(x) - y| \leq \frac{\log(n)^2}{8\sqrt{n}}\} \cup \{(x,y) : |\ell_\alpha(x) - y| \leq \frac{\log(n)^2}{8\sqrt{n}}\} \tag{B.1}$$

In the following lemma, this is used to bound the number of pairs of vertices in $\mathcal{B}(n,\alpha)$.

Lemma B.4. *Let Assumptions 3.5 hold. Then w.e.p., $\mathcal{A}_2^c$ holds. That is, w.e.p. for all i, $|\mathcal{B}(n,i,\alpha)| \leq \sqrt{n}\,\log(n)^2$.*

Proof. For all $1 \leq i,j \leq n$, let $e_{i,j} = \mathbf{1}_{\{j \in \mathcal{B}(n,i,\alpha)\}}$, $d = \frac{\log(n)^2}{8\sqrt{n}}$, and $f_{i,j} = \mathbf{1}_{\{|r_\alpha(U_i) - U_j| \leq d\} \cup \{|\ell_\alpha(U_i) - U_j| \leq d\}}$. If $\mathcal{A}_1^c$ holds, then by Lemma B.3 and by (B.1), $e_{i,j} \leq f_{i,j}$.

Fix $1 \leq i \leq n$ and let $\mathcal{H} = \sigma(U_i)$. Note that $\mathbb{E}[f_{i,j} \,|\, \mathcal{H}] \leq 4d = \frac{\log(n)^2}{2\sqrt{n}}$. By Hoeffding's bound, Equation (B.1), and Lemma B.3,

$$\begin{aligned}\mathbb{P}[\sum_{j \neq i} e_{i,j} > \sqrt{n}\log(n)^2 \,|\, \mathcal{H}] &\leq \mathbb{P}[\sum_{j \neq i} f_{i,j} > \sqrt{n}\log(n)^2 \,|\, \mathcal{H}] + \mathbb{P}[\mathcal{A}_1] \\ &= n^{-\Omega(\log(n))}\end{aligned}$$

Therefore, *w.e.p.* $|\mathcal{B}(n,i,\alpha)| \leq \sqrt{n}\,\log(n)^2$ for any particular choice of i, and thus also for all i simultaneously. □

Lemma B.5. *We have $\mathbb{P}[\mathcal{A}_3] = n^{-\Omega(\log(n))}$. That is, w.e.p. for all $0 \leq k \leq 4R-3$, $|\{i : U_i \in I_k\}| \geq \frac{n}{2R}$.*

Proof. For all $0 \leq k \leq 4R-3$ and all $0 \leq i \leq n$, let $s_{k,i} = \mathbf{1}_{\{U_i \in I_k\}}$. Then $\mathbb{E}[s_{k,i}] = \frac{3}{4R}$ and, for fixed k, the variables $s_{k,i}$, $1 \leq i \leq n$ are i.i.d. variables. Thus the result follows from Hoeffding's inequality and a union bound. □

Recall that $\nu(i,j)$ is the number of good witnesses of i and j, i.e. the number of vertices that are adjacent to precisely one of i,j in H_α, and thus can be used to distinguish i and j. Next, we show that vertices i,j with $|U_i - U_j|$ large enough have many good witnesses, and thus have substantially different neighbourhoods in H_α:

Lemma B.6. *Let Assumptions 3.5 hold. Then $\mathbb{P}[\mathcal{A}_4] = n^{-\Omega(\log(n))}$. Precisely, w.e.p. for all i,j so that $|U_i - U_j| > \frac{1}{\log(n)}$ holds, we also have $\nu(i,j) \geq \frac{2n}{\log(n)^2}$. Similarly, for all i,j so that $|U_i - U_j| > \frac{\log(n)^4}{\sqrt{n}}$, we also have $\nu(i,j) \geq \sqrt{n}\log(n)^3$.*

Proof. Fix $0 < i,j < 1$ and let $\mathcal{H}$ be the σ-algebra generated by (U_i, U_j). We consider the $\mathcal{H}$-measurable event $|U_i - U_j| > d$, where $d = d(n) = \frac{1}{\log(n)}$. Let $e_k = \mathbf{1}_{\{w_\alpha^{(2)}(U_i,U_k) \neq w_\alpha^{(2)}(U_j,U_k)\}}$ be the indicator function that vertex k distinguishes the vertices i,j. By Part 6 of Assumption 3.5, there exists $\epsilon > 0$ so that $w_\alpha^{(2)}$ is ϵ-separated. Assume that n is large enough so that $\epsilon \geq 4\log(n)^{-1}$. Then:

$$\mathbb{E}[e_k \,|\, \mathcal{H}] \geq \epsilon|U_i - U_j| > \epsilon d \geq \frac{4d}{\log(n)}.$$

Now conditional on $\mathcal{H}$, $\nu(i,j) = \sum_{k \in S\setminus\{i,j\}} e_k$ is the sum of $(n-2)$ i.i.d. Bernouilli variables. Thus, by Hoeffding's inequality,

$$\mathbb{P}[\{\sum_{k \in V(G)\setminus\{i,j\}} e_k < \frac{2dn}{\log(n)}\} \,|\, \mathcal{H}] = n^{-\Omega(\log(n))}.$$

Applying a union bound completes the proof of the first part. For the second part, the same argument applies, but with $d = \frac{\log(n)^4}{\sqrt{n}}$. □

Finally, we show that the number of vertices between i,j in the true order is roughly proportional to $|U_i - U_j|$ with high probability:

Lemma B.7. *For all i,j so that $U_i < U_j$, we have that $|\{k : U_k \in (U_i,U_j)\}| < n|U_i - U_j| + \sqrt{n}\log(n)\log(n)\}$ w.e.p.. Thus, w.e.p. $\mathcal{A}_5^c$ holds.*

Proof. This is just the rephrased version of Hoeffding's inequality. □

B.2. *Bad events for subsampled graphs*

We give the proofs from Section 3.3.3, using the notation from that section. Recall that $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4 \cup \mathcal{A}_5$, where $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4 \cup \mathcal{A}_5$ are the "bad events" described in Definition 3.6. Most of this section is concerned with analogous "bad events for samples" given in Definition 3.8, repeated here for convenience:

Definition B.8 (Bad Events for Samples)**.** Let graphon w, value α, random variables $\{U_i\}$, $\{U_{i,j}\}$, and graphs $G_\alpha^{(2)}$ and H_α be as defined above. Moreover, let R, $\{I_k\}_{k=0}^{4R-3}$ be as in (3.5). Finally, we let $S \subset V$ and $m = |S|$.

$$\begin{aligned}
\mathcal{A}_1' &= \{G_\alpha^{(2)}(S) \neq H_\alpha(S)\} \\
\mathcal{A}_3' &= \left\{ \min_{0 \le k \le 4R-3} |\{i \in S \,:\, U_i \in I_k\}| < \frac{m}{4R} \right\} \cup \{|S \backslash S''| > \log(n)^3\} \\
\mathcal{A}_4' &= \left\{ \min_{i,j \in S \,:\, |U_i - U_j| > \frac{1}{\log(n)}} \nu_S(i,j) < \log(n)^3 \right\} \\
\mathcal{A}_5' &= \left\{ \max_{i,j \in S} |\{k \,:\, U_k \in (U_i, U_j)\}| \ge m|U_i - U_j| + 2\log(n)^{3.5} \right\}.
\end{aligned}$$

We first prove Lemma 3.9, restated here for convenience:

Lemma 3.9. Let S be a uniformly chosen subset of V, of size $m = \log(n)^5$. Then on the $\mathcal{F}$-measurable event $\mathcal{A}^c$,

$$\mathbb{P}[\mathcal{A}_1' \,|\, \mathcal{F}] = O(n^{-0.49}).$$

Furthermore, for any fixed $p, q \in V(G)$ satisfying $(U_p, U_q) \notin \mathcal{B}(n, \alpha)$, we have

$$\frac{\mathbb{P}[\mathcal{A}_1' \cap \{p, q \in S\} \,|\, \mathcal{F}]}{\mathbb{P}[\{p, q \in S\} | \mathcal{F}]} = O(n^{-0.49}),$$

where again both probabilities are on $\mathcal{A}^c$.

Proof. It follows from Definition 3.4 that $G_\alpha^{(2)}(i,j) \neq H_\alpha(i,j)$ precisely when $j \in \mathcal{B}(n, i, \alpha)$. If $\mathcal{A}_2^c$ holds, then the size of $\mathcal{B}(n, i, \alpha)$ is bounded by $\sqrt{n}\log(n)^2$. Using this bound we obtain that, for a fixed $1 \le i \le n$,

$$\begin{aligned}
&\mathbb{P}[\bigcup_{j \in S \backslash \{i\}} \{G_\alpha^{(2)}(S)(i,j) \neq H_\alpha(S)(i,j)\} \,|\, \sigma(\{i \in S\}),\, \mathcal{F}] \qquad \text{(B.2)} \\
=&\mathbb{P}[\mathcal{B}(n, i, \alpha) \cap S \neq \emptyset \,|\, \sigma(\{i \in S\}),\, \mathcal{F}] \\
\le&(m-1)\left(\frac{|\mathcal{B}(n, i, \alpha)|}{n}\right) \\
\le&\frac{\log(n)^7}{\sqrt{n}},
\end{aligned}$$

where all conditional probabilities are on the event $\mathcal{A}^c$. By a union bound, $\mathbb{P}[\mathcal{A}_1' \,|\, \mathcal{F}] \le m\left(\frac{\log(n)^7}{\sqrt{n}}\right) = O(n^{-0.49})$ on $\mathcal{A}^c$.

Fix $p, q \in V(G)$, so that $(U_p, U_q) \notin \mathcal{B}(n, \alpha)$. If $(\mathcal{A}_1 \cup \mathcal{A}_2)^c$ holds, then $q \notin \mathcal{B}(n, p, \alpha)$ and $p \notin \mathcal{B}(n, q, \alpha)$. Thus, for a fixed $1 \le i \le n$, $i \notin \{p, q\}$, Equation B.2 holds for $S \setminus \{i, p, q\}$, so

$$\mathbb{P}[\cup_{j \in S \backslash \{i,p,q\}} \{G_\alpha^{(2)}(S)(i,j) \neq H_\alpha(S)(i,j)\} \,|\, \sigma(\{i \in S\}),\, \mathcal{F}] \mathbf{1}_{\{p,q \in S\}} \le \frac{\log(n)^7}{\sqrt{n}}$$

on $\mathcal{A}^c$. Again by a union bound,

$$\mathbb{P}[\mathcal{A}_1' \,|\, \mathcal{F}]\mathbf{1}_{\{p,q\in S\}} = O(n^{-0.49})$$

on $\mathcal{A}^c$, from which the result follows. □

We next prove Lemma 3.10, restated here for convenience:

Lemma 3.10. Let S be a uniformly chosen subset of V, of size $m = \log(n)^5$, and let S'' be as defined in (3.7). We have $\mathbb{P}[\mathcal{A}_3' \,|\, \mathcal{F}] = n^{-\Omega(\log(n))}$ on the $\mathcal{F}$-measurable event $\mathcal{A}^c$. That is, *w.e.p.* for all $0 \le k \le 4R-3$, both $|S \cap V_k| \ge \frac{\log(n)^5}{4R}$ and also $|S \setminus S''| \le \log(n)^3$ on $\mathcal{A}^c$. Moreover, for any fixed $p, q \in V(G)$,

$$\frac{\mathbb{P}[\mathcal{A}_3' \cap \{p,q \in S\} \,|\, \mathcal{F}]}{\mathbb{P}[\{p,q \in S\} |\, \mathcal{F}]} = O(n^{-\Omega(\log(n))}),$$

where again both probabilites are on the event $\mathcal{A}^c$.

Proof. Assume $\mathcal{A}_3^c$ holds, so for all $0 \le k \le 4R-3$, $|V_k| \ge \frac{n}{2R}$. For all $1 \le i \le n$, let $e_i = \mathbf{1}_{\{i \in S\}}$. Fix $0 \le k \le 4R-3$. Then $|\{i \in S : U_i \in I_k\}| = \sum_{\{i\in V_k\}} e_i$ and $\mathbb{E}[\sum_{\{i \in V_k\}} e_i \,|\, \mathcal{F}, \mathcal{A}^c] \ge \frac{m}{2R}$. By Azuma's inequality, we obtain that

$$\mathbb{P}[\{ \sum_{\{i\in V_k\}} e_i < \frac{m}{4R} \}] = n^{-\Omega(\log(n))}. \tag{B.3}$$

Applying a union bound, we obtain the first part of the result.

To prove the second part, note first that, for fixed $1 \le i \le n$, by definition

$$\{i \in S''\} = \{i \in S\} \cap \{i \notin \cup_{j\in S\setminus\{i\}} \mathcal{B}(n,j,\alpha)\}.$$

If $\mathcal{A}_2^c$ holds, then for all j, $|\mathcal{B}(n,j,\alpha)| \le \sqrt{n}\log(n)^2$, and thus

$$\mathbb{P}[\{i \in S''\} \,|\, \sigma(S\backslash\{i\}), \mathcal{F}] \ge 1 - |S|\frac{\log(n)^2}{\sqrt{n}} \ge 1 - \frac{2\log(n)^7}{\sqrt{n}}$$

on $\mathcal{A}^c$. Applying Azuma's inequality,

$$\mathbb{P}[\{|S''| < m - \log^3(n)\} \,|\, \mathcal{F}] = n^{-\Omega(\log(n))} \tag{B.4}$$

on $\mathcal{A}^c$. This completes the proof of the bound on $\mathbb{P}[\mathcal{A}_3'|\mathcal{F}]$.

To prove the second bound, fix $p, q \in V(G)$. An analogous argument applied to the set $S - \{p,q\}$ gives the result. □

We next prove Lemma 3.11, restated here for convenience:

Lemma 3.11. Let S be a uniformly chosen subset of V, of size $m = \log(n)^5$, and let S'' as defined in (3.7). For each $i \in S$, let $D_S(i) = |\{j \in S : G_\alpha^{(2)}(i,j) = 1\}|$ be the number of neighbours of i in $G_\alpha^{(2)}(S)$. Then for all n sufficiently large,

$$\{H_\alpha(S) \text{ is not connected}\} \cup \{H_\alpha(S'') \text{ is not connected}\} \cup \left\{ \min_{i\in S} D_S(i) < \frac{m}{8R} \right\} \subset \mathcal{A}_3'.$$

Proof. For any set $S \subset V$, if $H_\alpha(S)$ is not connected then there must be $0 \leq k \leq 4R-3$ so that $S \cap V_k = \emptyset$. Thus it follows immediately that $\{H_\alpha(S)$ is not connected $\} \subset \mathcal{A}_3'$.

Moreover, if $(\mathcal{A}_3')^c$ holds, then $|S \setminus S''| \leq \log(n)^3$ and for each $0 \leq k \leq 4R-3$,

$$|S \cap V_k| \geq \log(n)^5/(4R).$$

Thus we get that $|S'' \cap V_k| \geq |S \cap V_k| - |S/S''| \geq \log(n)^5/(8R)$ for n sufficiently large. Thus $V_k \cap S'' \neq \emptyset$ for all k, which implies that $H_\alpha(S'')$ is connected.

By the choice of R, for any two vertices $i, j \in V_k$, $H_\alpha(U_i, U_j) = 1$. Fix $i \in S$. By definition, for each $j \in S''$, $H_\alpha(i,j) = G_\alpha^{(2)}(i,j)$. Let k be so that $U_i \in I_k$. Then for all vertices $j \in S'' \cap V_k$, $G_\alpha^{(2)}(i,j) = 1$. As argued above, there are at least $\log(n)^5/(8R)$ such vertices. This shows that $\{\min_{i \in S} D_S(i) < \frac{\log(n)^5}{8R}\} \subset \mathcal{A}_3'$. □

We next prove Lemma 3.12, restated here for convenience:

Lemma 3.12. Let S be a uniformly chosen subset of V, of size $m = \log(n)^5$. Then

$$\mathbb{P}[\mathcal{A}_4' \,|\, \mathcal{F}] = n^{-\Omega(\log n)}$$

on the $\mathcal{F}$-measurable event $\mathcal{A}^c$. That is, *w.e.p.* for all $i, j \in S$ so that $|U_i - U_j| > \frac{1}{\log(n)}$, $\nu_S(i,j) \geq \log(n)^3$.

Proof. Fix $1 \leq i < j \leq n$ so that $|U_i - U_j| > \frac{1}{\log(n)}$. If $\mathcal{A}_4^c$ holds, then $\nu(i,j) \geq \frac{2n}{\log(n)^2}$. Thus $\mathbb{E}[\nu_S(i,j) \,|\, \mathcal{F}, \mathcal{A}^c] \geq 2m/\log(n)^2 = 2\log(n)^3$. The result follows from Azuma's inequality and a union bound. □

Finally, we prove Lemma 3.13, restated here for convenience:

Lemma 3.13. On the $\mathcal{F}$-measurable event $\mathcal{A}^c$,

$$\mathbb{P}[\mathcal{A}_5' \,|\, \mathcal{F}] = n^{-\Omega(\log n)}.$$

That is, *w.e.p.* for all $i, j \in S$, $|\{k : U_k \in (U_i, U_j)\}| \leq m|U_i - U_j| + 2\log(n)^{3.5}$.

Proof. Fix $1 \leq i, j \leq n$, so that $U_i < U_j$, and let $W_{i,j} = \{k : U_k \in (U_i, U_j)\}$. For each $k \in W_{i,j}$, let $e_k = \mathbf{1}_{\{k \in S\}}$. Then $|S \cap W_{i,j}| = \sum_{i \in W_{i,j}} e_k$, and $\mathbb{E}[\sum_{i \in W_{i,j}} e_k \,|\, \mathcal{F}] = (m/n)|W_{i,j}|$.

If $\mathcal{A}_5^c$ holds, then $(m/n)|W_{i,j}| \leq m|U_j - U_i| + m\frac{\log(n)}{\sqrt{n}} \leq m|U_i - U_j| + 1$ for n sufficiently large. By Azuma's inequality, on $\mathcal{A}^c$

$$\mathbb{P}[\mathcal{A}_5' \,|\, \mathcal{F}] \leq \mathbb{P}[|\sum_{i \in V_k} e_k - \mathbb{E}[\sum_{i \in V_k} e_k \,|\, \mathcal{F}]| > \log(n)^{3.4} \,|\, \mathcal{F}] = n^{-\Omega(\log(n))}.$$

The result then follows by a union bound. □

Appendix C: A simple anticoncentration lemma

We need the follow technical lemma, which follows almost immediately from Corollary 1.8 of [8]:

Lemma C.1. *Let* $U[1], U[2], \ldots \overset{i.i.d.}{\sim} \mathrm{Unif}([0,1])$*, and for* $n \in \mathbb{N}$ *and* $c > 0$ *let* $\tilde{U}_{\frac{c}{\sqrt{n}}}[i] = e^{1-\frac{c}{\sqrt{n}}} U[i]$*. Then there exists a universal constant* $A > 0$ *not depending on* c *or* n *so that*[4]

$$\|\mathcal{L}((U[1], \ldots, U[n])) - \mathcal{L}((\tilde{U}_{\frac{c}{\sqrt{n}}}[1], \tilde{U}_{\frac{c}{\sqrt{n}}}[1]))\|_{\mathrm{TV}} \leq Ac. \tag{C.1}$$

Proof. Let $V[i] = -\log(U[i])$, and note that $V[1], V[2], \ldots$ are i.i.d. random variables with exponential distribution and mean 1. Define $\tilde{V}_{\frac{c}{\sqrt{n}}}[i] = (1 - \frac{c}{\sqrt{n}})V[i]$. By Corollary 1.8 of [8], there exists a universal constant $A > 0$ such that

$$\|\mathcal{L}((V[1], \ldots, V[n])) - \mathcal{L}((\tilde{V}_{\frac{c}{\sqrt{n}}}[1], \tilde{V}_{\frac{c}{\sqrt{n}}}[1]))\|_{\mathrm{TV}} \leq Ac. \tag{C.2}$$

But the random variables appearing in Inequality (C.1) and (C.2) differ only by the monotone 1-1 transformations

$$V[i] = -\log(U[i]), \; \hat{V}_{\frac{c}{\sqrt{n}}}[i] = -\log(\hat{U}_{\frac{c}{\sqrt{n}}}[i]),$$

and so the TV distances are in fact equal. □

We give the proof of Theorem 4:

Proof of Theorem 4. Our coupling is based on the following three observations:

1. Fix $c > 0$ and define $\delta_n = \frac{c}{\sqrt{n}}$. By Lemma C.1, there exists a universal constant $0 < A < \infty$ so that

$$\|\mathcal{L}(U^{(1)}) - \mathcal{L}(\hat{U}^{(1)}_{\delta_n})\|_{\mathrm{TV}} \leq Ac. \tag{C.3}$$

2. For *any* value of δ, w and sequences $u^{(1)}, u^{(2)}$, we have

$$G(w; \hat{u}^{(1)}_{\delta}, u^{(2)}) = G(\hat{w}; u^{(1)}_{\delta}, u^{(2)}). \tag{C.4}$$

3. For *any* value of $u, \delta \in (0,1)$ we have by Taylor's theorem

$$|u - \tilde{u}_{\delta}| \geq |\delta \log(u)| - \frac{e}{2}\delta^2 \log(u)^2.$$

In particular, for δ_n as above and any $\eta > 0$, there exists a universal constant $B = B(\eta)$ so that

$$\mathbb{P}[|U[i] - \tilde{U}_{\delta_n}[i]| \geq B\frac{c}{\sqrt{n}}] \geq 1 - \eta. \tag{C.5}$$

[4] Note that the following bound refers to only the *distribution* of the vectors U, $U_{\frac{c}{\sqrt{n}}}$. In particular, it "forgets" the coupling $\tilde{U}_{\frac{c}{\sqrt{n}}}[i] = e^{1-\frac{c}{\sqrt{n}}} U[i]$ between the vectors.

Define the event $\mathcal{A}_c = \{V = \tilde{U}_{\frac{c}{\sqrt{n}}}\}$. By Equation (C.3), there is a coupling of (U, V) such that

$$\mathbb{P}[\mathcal{A}_c] \geq Ac.$$

On this event, Equation (6.1) holds from Equation (C.4), Equation (6.2) is immediate from the definition, and Inequality (6.3) follows immediately from Inequality (C.5) and Hoeffding's inequality. This completes the proof. □

References

[1] Atkins, J. E., E. G. Boman, and B. Hendrickson (1998). A spectral algorithm for seriation and the consecutive ones problem. *SIAM Journal on Computing 28*(1), 297–310. MR1630473

[2] Bagaria, V., J. Ding, D. Tse, Y. Wu, and J. Xu (2020). Hidden Hamiltonian cycle recovery via linear programming. *Operations Research 68*(1), 53–70. MR4059492

[3] Barthélemy, J.-P. and F. Brucker (2001). NP-hard approximation problems in overlapping clustering. *J. Classif. 18*, 159–183. MR1868627

[4] Boucheron, S., Lugosi, G. and Massart, P. (2013). Concentration inequalities: A nonasymptotic theory of independence. *Oxford University Press.* MR3185193

[5] Cai, T., T. Liang and A. Rakhlin (2017). On detection and structural reconstruction of small-world random networks. *IEEE Transactions on Network Science and Engineering 4*(3), 165–176. MR3696219

[6] Çela, E., V. G. Deineko, and G. J. Woeginger (2015). A new tractable case of the QAP with a Robinson matrix. In *Combinatorial Optimization and Applications*, pp. 709–720. Springer. MR3481504

[7] Chatterjee, S. (2015). Matrix estimation by universal singular value thresholding. *The Annals of Statistics 43*(1), 177–214. MR3285604

[8] Chatterjee, S. (2019). A general method for lower bounds on fluctuations of random variables. *The Annals of Probability 47*(4), 2140–2171. MR3980917

[9] Chepoi, V., B. Fichet, and M. Seston (2009). Seriation in the presence of errors: NP-hardness of ℓ_∞-fitting Robinson structures to dissimilarity matrices. *J. Classification 26*(3), 279–296. MR2587799

[10] Chepoi, V. and M. Seston (2011, Apr). Seriation in the presence of errors: A factor 16 approximation algorithm for ℓ_∞-fitting Robinson structures to distances. *Algorithmica 59*(4), 521–568. MR2771110

[11] Chuangpishit, H., M. Ghandehari, M. Hurshman, J. Janssen, and N. Kalyaniwalla (2015). Linear embeddings of graphs and graph limits. *Journal of Combinatorial Theory, Series B 113*, 162–184. MR3343752

[12] Chuangpishit, H., M. Ghandehari, and J. Janssen (2017). Uniform linear embeddings of graphons. *European Journal of Combinatorics 61*, 47–68. MR3588708

[13] Corneil, D. G. (2004). A simple 3-sweep LBFS algorithm for the recognition

of unit interval graphs. *Discrete Applied Mathematics 138*(3), 371 – 379. MR2049655

[14] Ding, J., Y. Wu, J. Xu and D. Yang (2021). Consistent recovery threshold of hidden nearest neighbor graphs. *IEEE Transactions on Information Theory 67*(8), 5211–5229. MR4306329

[15] Evangelopoulos, X., A. J. Brockmeier, T. Mu, and J. Y. Goulermas (2017). A graduated non-convexity relaxation for large scale seriation. In *Proc. SIAM 2017 Intl. Conf. Data Mining*, pp. 462–470.

[16] Flammarion, N., C. Mao, and P. Rigollet (2019, 02). Optimal rates of statistical seriation. *Bernoulli 25*(1), 623–653. MR3892331

[17] Fogel, F., A. d´Aspremont, and M. Vojnovic (2016). Spectral ranking using seriation. *J. Mach. Learn. Res. 17*, Paper No. 88, 45. MR3517111

[18] Fogel, F., R. Jenatton, F. Bach, and A. D´Aspremont (2013). Convex relaxations for permutation problems. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 26*, pp. 1016–1024. Curran Associates, Inc. MR3418224

[19] Fortin, D. and I. Tseveendorj (2015). Minimizing sign changes rowwise: Consecutive ones property and beyond. In G. O. Tost and O. Vasilieva (Eds.), *Analysis, Modelling, Optimization, and Numerical Techniques*, Cham, pp. 37–48. Springer International Publishing. MR3354166

[20] Gao, C., Y. Lu, and H. H. Zhou (2015). Rate-optimal graphon estimation. *The Annals of Statistics 43*(6), 2624–2652. MR3405606

[21] Ghandehari, M. and J. Janssen (2020). Graph sequences sampled from Robinson graphons. *arXiv preprint arXiv:*2005.05253v2.

[22] Laurent, M. and M. Seminaroti (2015). The quadratic assignment problem is easy for Robinsonian matrices with Toeplitz structure. *Operations Research Letters 43*(1), 103 – 109. MR3302558

[23] Laurent, M. and M. Seminaroti (2017). Similarity-first search: a new algorithm with application to Robinsonian matrix recognition. *SIAM J. Discrete Math. 31*(3), 1765–1800. MR3687845

[24] Liiv, I. (2010, April). Seriation and matrix reordering methods: An historical overview. *Stat. Anal. Data Min. 3*(2), 70–91. MR2608807

[25] Lim, C. H. and S. Wright (2014). Beyond the Birkhoff polytope: Convex relaxations for vector permutation problems. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 27*, pp. 2168–2176. Curran Associates, Inc.

[26] Little, A., Y. Xie, and Q. Sun (2018). An analysis of classical multidimensional scaling. *arXiv preprint arXiv:*1812.11954.

[27] Lovász, L. (2012). *Large networks and graph limits*, Volume 60 of *American Mathematical Society Colloquium Publications*. American Mathematical Society, Providence, RI. MR3012035

[28] Lovász, L. and B. Szegedy (2006). Limits of dense graph sequences. *Journal of Combinatorial Theory, Series B 96*(6), 933–957. MR2274085

[29] Mao, C., A. Pananjady, and M. J. Wainwright (2018a). Breaking the $1/\sqrt{n}$

barrier: Faster rates for permutation-based models in polynomial time. *Conference on Learning Theory (CoLT) 75*, 2037–2042.

[30] Mao, C., A. Pananjady, and M. J. Wainwright (2020). Towards optimal estimation of bivariate isotonic matrices with unknown permutations. *The Annals of Statistics 48*(6), 3183 – 3205. MR4185805

[31] Moitra, A. and G. Valiant (2010). Settling the polynomial learnability of mixtures of Gaussians. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 93–102. IEEE. MR3024779

[32] Morris, Steven A., Benyam Asnake, and Gary G. Yen (2003). Dendrogram seriation using simulated annealing. *Information Visualization 2*(2), 95–104.

[33] Natik, A. and A. Smith (2020). Consistency of spectral seriation. *arXiv preprint arXiv:*2112.04408.

[34] Petrie, W. F. (1899). Sequences in prehistoric remains. *The Journal of the Anthropological Institute of Great Britain and Ireland 29*(3/4), 295–301.

[35] Putnam, Nicholas H., Brendan L. O'Connell, Jonathan C. Stites, Brandon J. Rice, Marco Blanchette, Robert Calef, Christopher J. Troll, Andrew Fields, Paul D. Hartley, and Charles W. Sugnet (Feb. 2016). Chromosome-scale shotgun assembly using an in vitro method for long-range linkage. *Genome Research 26*(3), 342–350.

[36] Rocha, I., J. Janssen, and N. Kalyaniwalla (2018). Recovering the structure of random linear graphs. *Linear Algebra and its Applications 557*, 234–264. MR3848269

[37] Rose, D., R. Tarjan, and G. Lueker (1976). Algorithmic aspects of vertex elimination on graphs. *SIAM J. Computing 5*, 266–284. MR0408312

[38] Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing 17*(4), 395–416. MR2409803

[39] Vuokko, N. (2010). *Consecutive ones property and spectral ordering*, in *Proceedings of the SIAM International Conference on Data Mining, SDM 2010*, pp. 350–360.