

# Optimal Hardness of Online Algorithms for Large Common Induced Subgraphs

**David Gamarnik**

MIT

**Miklós Z. Rácz**

Northwestern University

**Gabe Schoenbach**

University of Chicago

GAMARNIK@MIT.EDU

MIKLOS.RACZ@NORTHWESTERN.EDU

GABE.SCHOENBACH@UCHICAGO.EDU

**Editors:** Steve Hanneke and Tor Lattimore

## Abstract

We study the problem of efficiently finding large common induced subgraphs of two independent Erdős–Rényi random graphs  $G_1, G_2 \sim \mathbb{G}(n, 1/2)$ . Recently, [Chatterjee and Diaconis \(2023\)](#) showed that the largest common induced subgraph of  $G_1$  and  $G_2$  has size  $(4 - o(1)) \log_2 n$  with high probability. We first show that a simple greedy online algorithm finds a common induced subgraph of  $G_1$  and  $G_2$  of size  $(2 - o(1)) \log_2 n$  with high probability. Our main result shows that no online algorithm can find a common induced subgraph of  $G_1$  and  $G_2$  of size at least  $(2 + \varepsilon) \log_2 n$  with probability bounded away from 0 as  $n \rightarrow \infty$ . Together, these results provide evidence that this problem exhibits a computation-to-optimization gap. To prove the impossibility result, we show that the solution space of the problem exhibits a version of the (multi) overlap gap property (OGP), and utilize an interpolation argument recently developed by [Gamarnik, Kızıldağ, and Warnke \(2025\)](#) that connects OGP and online algorithms.

**Keywords:** Largest common induced subgraph, random optimization problems, online algorithms, computation-to-optimization gap, overlap gap property

## 1. Introduction

In the *largest common induced subgraph* (LCIS) problem, the input is a pair of graphs  $G_1$  and  $G_2$ , and the goal is to find a common induced subgraph of  $G_1$  and  $G_2$  with as many vertices as possible. A *solution*  $\sigma = (H_1, H_2)$  is any pair of induced subgraphs  $H_1 \subseteq G_1$  and  $H_2 \subseteq G_2$  such that  $H_1 \cong H_2$ , where  $\cong$  denotes graph isomorphism. The *size* of the solution  $|\sigma| := |H_1| = |H_2|$  is the number of vertices of the common subgraph and the goal is to maximize this quantity.

The LCIS problem is a fundamental optimization problem with close connections to other central problems such as graph isomorphism and graph matching, and it arises naturally in many applications, such as bioinformatics ([Ehrlich and Rarey, 2011](#)), chemoinformatics ([Schietgat et al., 2013](#)), and pattern recognition ([Conte et al., 2004](#)). It is computationally hard in the worst case, even to approximate within any reasonable factor ([Kann, 1992](#)).

In this paper we study arguably the simplest and most natural average-case version of the LCIS problem, where  $G_1, G_2 \sim \mathbb{G}(n, 1/2)$  are independent Erdős–Rényi random graphs with  $n$  vertices and edge probability  $1/2$ ; we refer to this problem as LCIS( $n$ ). Recently, [Chatterjee and Diaconis \(2023\)](#) showed that the largest solution has size  $(4 - o(1)) \log_2 n$  with high probability as  $n \rightarrow \infty$ .<sup>1,2</sup>

In this work, we initiate the study of the algorithmic side of the LCIS( $n$ ) problem. We ask:

1. In fact, they proved a much more precise two-point concentration result; however, in this paper we primarily focus on the leading order term.
2. Throughout the paper, we say that an event holds with high probability if its probability tends to 1 as  $n \rightarrow \infty$ .

*What are the limits of efficient algorithms that solve  $\text{LCIS}(n)$  with high probability?*

In particular, we focus on a broad class of online algorithms, which arise naturally for this problem. We first show that a simple greedy online algorithm finds a solution of size at least  $(2 - o(1)) \log_2 n$  with high probability. In other words, this efficient algorithm is half-optimal and serves as a natural benchmark.

**Theorem 1 (Algorithms, informal)** *There exists an efficient online algorithm that finds a solution to  $\text{LCIS}(n)$  that has size at least  $(2 - o(1)) \log_2 n$  with high probability as  $n \rightarrow \infty$ .*

Our main result shows that no online algorithm can do better than half-optimal. The class of online algorithms we consider is broad; roughly, it is defined as follows. An online algorithm for LCIS processes the two graphs  $G_1$  and  $G_2$  one vertex pair  $(u_t, v_t) \in V(G_1) \times V(G_2)$  at a time. When a vertex pair is processed, all connections of  $u_t$  and  $v_t$  to previously processed vertices are revealed (in  $G_1$  and  $G_2$ , respectively). The algorithm then has to make a decision whether or not to add a pair of vertices to the partial solution it has built so far. The main constraints are that both of the added vertices must have been already processed and *at least* one of them must be  $u_t$  or  $v_t$ .

Notably, an online algorithm does *not* have to add both  $u_t$  and  $v_t$  at the same time, and as a result, it is possible for an individual vertex to be added to the solution well after it was originally processed. This is a fundamental difference compared to online algorithms for problems involving a single graph, where usually an immediate decision must be made about a vertex as soon as it is processed.<sup>3</sup> This difference stems from the “two-dimensional” nature of the LCIS problem, with the input being two graphs. We defer formal definitions to Section 1.2 below.

**Theorem 2 (Impossibility, informal)** *Fix any  $\varepsilon > 0$ . There is no online algorithm for  $\text{LCIS}(n)$  that finds a solution of size at least  $(2 + \varepsilon) \log_2 n$  with probability bounded away from 0 as  $n \rightarrow \infty$ .*

Overall, these results combined provide evidence that the  $\text{LCIS}(n)$  problem exhibits a computation-to-optimization gap. We next turn to describing our results more formally and in more detail.

### 1.1. A half-optimal greedy online algorithm

A naive brute-force approach to finding a size- $k$  solution to  $\text{LCIS}(n)$  would require checking all pairs of size- $k$  subgraphs of  $G_1$  and  $G_2$ . For  $k = \Theta(\log n)$ , such an exhaustive approach would require superpolynomial runtime, since  $\binom{n}{k}^2 \geq (n/k)^{2k} \geq n^{\Omega(\log n)}$ .

To get a polynomial-time algorithm, a useful initial observation is that independent sets of equal size are isomorphic, so any algorithm that finds independent sets can also solve the LCIS problem (and the same applies to cliques). The largest independent set in  $G \sim \mathbb{G}(n, 1/2)$  has size  $(2 - o(1)) \log_2 n$  with high probability and a simple greedy online algorithm finds an independent set of size  $(1 + o(1)) \log_2 n$  (Grimmett and McDiarmid, 1975; Karp, 1976). So this immediately gives an efficient quarter-optimal solution to  $\text{LCIS}(n)$ : use a greedy algorithm to find independent sets of size  $(1 + o(1)) \log_2 n$  in both  $G_1$  and  $G_2$ . However, there is no known efficient algorithm

3. We note that in the problem of finding large cliques in the “one-dimensional” single graph case considered earlier in Gamarnik et al. (2025), postponing the decision about including/excluding  $u_t$  trivializes the problem. Indeed, one can delay all decisions until all but  $(2 - o(1)) \log_2 n$  vertices are revealed, find a nearly largest clique among the revealed vertices, and then include the vertices of the discovered clique in the last  $(2 - o(1)) \log_2 n$  rounds.

to find an independent set of size  $(1 + \varepsilon) \log_2 n$  in  $G \sim \mathbb{G}(n, 1/2)$  — indeed, it is conjectured that such an algorithm does not exist — so it is not immediately clear how to go beyond this.

The key idea is to tailor the greedy algorithm for finding an independent set to the LCIS problem. Recall first the algorithm for finding an independent set in a single graph  $G$  (Grimmett and McDiarmid, 1975; Karp, 1976), which processes the vertices  $u_1, \dots, u_n$  of  $G$  in an arbitrary order. It initializes the solution set  $S$  to be empty and proceeds greedily, adding  $u_t$  to  $S$  if and only if  $u_t$  is not connected to any of the vertices in  $S$ . After processing all vertices, it outputs the induced subgraph  $G[S]$ , which is an independent set.

Turning to the LCIS problem, the input is now two graphs  $G_1$  and  $G_2$ . Let  $u_1, \dots, u_n$  and  $v_1, \dots, v_n$  be arbitrary vertex orderings of  $G_1$  and  $G_2$ , respectively, and initialize the pair of induced subgraphs  $H_1$  and  $H_2$  to be empty. The greedy algorithm processes vertex pairs  $(u_t, v_t)$  one at a time and does the following at time  $t$ . First, all connections of  $u_t$  and  $v_t$  to previously processed vertices are revealed (in  $G_1$  and  $G_2$ , respectively). Then, for all  $j \leq t$ , do the following:

- if  $u_t$  and  $v_j$  can be added to the subgraphs  $H_1$  and  $H_2$ , respectively, while preserving that  $H_1 \cong H_2$ , then do so;
- otherwise, perform the same check for  $u_j$  and  $v_t$ .

Note that once a vertex pair is added to  $H_1$  and  $H_2$  at time  $t$ , the algorithm moves on to time  $t + 1$ . Once all vertices are processed, the algorithm returns  $(H_1, H_2)$  and a bijection  $\pi : V(H_1) \rightarrow V(H_2)$  that verifies  $H_1 \cong H_2$ .<sup>4</sup> See Algorithm 1 in Section 3 for a detailed description of the algorithm.

**Theorem 3 (Algorithms, formal)** *Let  $(G_1, G_2) \sim \mathbb{G}(n, 1/2)^{\otimes 2}$ , define GREEDY as in Algorithm 1, and let  $(H_1, H_2, \pi)$  denote the output of GREEDY( $G_1, G_2$ ). Then  $(H_1, H_2)$  is a solution to LCIS( $n$ ) and  $\pi$  is a graph isomorphism between  $H_1$  and  $H_2$ . Furthermore, for any  $\varepsilon > 0$  and for all  $n$  large enough, the size of  $(H_1, H_2)$  is at least  $(2 - \varepsilon) \log_2 n$  with probability at least  $1 - \exp(-n^{\varepsilon/4})$ .*

The GREEDY algorithm has runtime  $O(n^2 \log n)$  (i.e., quasilinear in the input size) with high probability in the average-case setting of Theorem 3, and  $O(n^3)$  in the worst case (see Claim 1). Also, the proof of Theorem 3 shows that the conclusion still holds for  $\varepsilon = \varepsilon_n \geq \frac{C \log \log n}{\log n}$  for large enough constant  $C$ , and in particular it shows that the output of GREEDY has size at least  $2 \log_2 n - 9 \log \log n$  with high probability (see Corollary 6).

## 1.2. The limits of online algorithms

Our impossibility result rules out a certain broad class of online algorithms, which includes the GREEDY algorithm described in Algorithm 1. Instead of receiving the input all at once, these algorithms receive bits of the input in serial and *make irrevocable decisions about the output at each step*. It is this irrevocability property that constrains online algorithms; otherwise, the serial processing of the input is moot. Unlike the closely related independent set problem on a single graph, the input to LCIS consists of *two* graphs, so there are several distinct notions of how information about the pair  $(G_1, G_2)$  should be processed by an online algorithm. What ought to be the atomic object?

4. Note that outputting a graph isomorphism  $\pi$  that verifies  $H_1 \cong H_2$  is not strictly necessary to solve the LCIS problem.

We propose the following natural definition. At a high level, at each step, an online algorithm  $\mathcal{A}$  will process a *pair* of vertices  $(v_1, v_2) \in V(G_1) \times V(G_2)$  and decide whether or not to add a (potentially distinct) pair of vertices to the partial solution, under the constraint that both of the added vertices must have been already processed, and *at least* one of them must be  $v_1$  or  $v_2$ . As a result, it is possible for an individual vertex to be added to the solution after it was processed. As highlighted previously, this is a fundamental difference compared to the definition of online algorithms for independent set (Gamarnik et al., 2025), and is due to the two-dimensional nature of the LCIS problem. Our formal definition is below.

**Definition 4 (Online algorithms for LCIS)** *The class  $\text{ONLINE}(n)$  consists of algorithms  $\mathcal{A}$  which take as input  $(G_1, G_2)$  and a random string  $\omega \in \Omega$ , where for each  $j \in \{1, 2\}$ ,  $G_j = (V_j, E_j)$  is a graph with  $V_j = [n]$ . Each  $\mathcal{A} \in \text{ONLINE}(n)$  proceeds in rounds, and at the end of round  $t \in [n]$ ,  $\mathcal{A}$  maintains the following state:*

- *The sets of vertices  $P_1^{(t)} \subseteq V_1$  and  $P_2^{(t)} \subseteq V_2$  inspected thus far.*
- *Vertex sets  $S_1^{(t)} \subseteq P_1^{(t)}$  and  $S_2^{(t)} \subseteq P_2^{(t)}$  that induce a pair of isomorphic subgraphs. We write*

$$\mathcal{A}^{(t)}(G_1, G_2, \omega) := \left( G_1 \left[ S_1^{(t)} \right], G_2 \left[ S_2^{(t)} \right] \right) \quad (1)$$

*for the (partial) solution at the end of round  $t \in [n]$ .*

*$\mathcal{A}$  initializes  $P_1^{(0)} = P_2^{(0)} = S_1^{(0)} = S_2^{(0)} = \emptyset$ . During each round  $t \in [n]$ ,  $\mathcal{A}$  proceeds as follows, where for all  $\{u, v\} \in \binom{[n]}{2}$ ,  $e_j(\{u, v\}) \in \{0, 1\}$  denotes the edge status of the vertex pair  $\{u, v\}$  in graph  $G_j$ .*

1. (Processing a new vertex pair.) *Based on  $\omega$  and all of the information revealed through the end of round  $t - 1$  (namely,  $P_1^{(t-1)}$ ,  $P_2^{(t-1)}$ ,  $G_1 \left[ P_1^{(t-1)} \right]$ ,  $G_2 \left[ P_2^{(t-1)} \right]$ ,  $S_1^{(t-1)}$ , and  $S_2^{(t-1)}$ ), for  $j \in \{1, 2\}$ , the algorithm  $\mathcal{A}$  selects  $v_j^{(t)} \in V_j \setminus P_j^{(t-1)}$  and defines  $P_j^{(t)} := P_j^{(t-1)} \cup \{v_j^{(t)}\}$ . It then reveals  $e_j(\{u, v\})$  for all not-yet-seen vertex pairs  $\{u, v\} \in \binom{P_j^{(t)}}{2}$ , for  $j \in \{1, 2\}$ .*
2. (Updating the solution.) *Based on  $\omega$  and all of the information revealed thus far,  $\mathcal{A}$  chooses how to update the vertex sets as*

$$\left( S_1^{(t)}, S_2^{(t)} \right) \in \left\{ \begin{array}{l} \left( S_1^{(t-1)}, S_2^{(t-1)} \right) \\ \left( S_1^{(t-1)} \cup \{v_1\}, S_2^{(t-1)} \cup \{v_2\} \right), \end{array} \right. \quad (2)$$

*under the constraint that  $v_j \in P_j^{(t)}$  for  $j \in \{1, 2\}$ , and either  $v_1 = v_1^{(t)}$  or  $v_2 = v_2^{(t)}$  (or both). Moreover, the algorithm  $\mathcal{A}$  must maintain that  $G_1 \left[ S_1^{(t)} \right] \cong G_2 \left[ S_2^{(t)} \right]$ .*

*We write the output of  $\mathcal{A}$  as*

$$\mathcal{A}(G_1, G_2, \omega) := \mathcal{A}^{(n)}(G_1, G_2, \omega) = \left( G_1 \left[ S_1^{(n)} \right], G_2 \left[ S_2^{(n)} \right] \right). \quad (3)$$

*We write  $|\mathcal{A}(G_1, G_2, \omega)| := \left| S_1^{(n)} \right| = \left| S_2^{(n)} \right|$  to denote the size of the output.*

Aside from the two-dimensional nature of LCIS, Definition 4 closely tracks (Gamarnik et al., 2025, Definition 2.1). One difference is that the latter definition includes algorithms that can choose a set of *exceptional edges*, which allows for better-informed decisions at each step. We believe that Definition 4 could be similarly strengthened to include exceptional information without affecting our results (assuming an appropriate bound on the amount of exceptional information), but we choose not to for simplicity.

Our main result is the following, which rules out online algorithms beyond  $2 \log_2 n$ .

**Theorem 5 (Impossibility, formal)** *Let  $(G_1, G_2) \sim \mathbb{G}(n, 1/2)^{\otimes 2}$ , let  $\mathcal{A} \in \text{ONLINE}(n)$ , and fix any  $\varepsilon > 0$ . For all  $\omega \in \Omega$  and all sufficiently large  $n$ , we have that*

$$\mathbb{P}(|\mathcal{A}(G_1, G_2, \omega)| \geq (2 + \varepsilon) \log_2 n) \leq n^{-\Omega(\log n)}. \quad (4)$$

### 1.3. Related work

**Existential results for LCIS.** The LCIS( $n$ ) problem was first studied by Chatterjee and Diaconis (2023), who showed a precise two-point concentration result: with high probability the largest common induced subgraph has size  $\lfloor x_n - \varepsilon \rfloor$  or  $\lfloor x_n + \varepsilon \rfloor$ , where  $x_n$  satisfies  $x_n = (4 - o(1)) \log_2 n$ . Subsequently, Surya, Warnke, and Zhu (2025) proved a more general result for arbitrary edge densities  $p_1, p_2 \in (0, 1)^2$  in the two graphs, with three qualitatively different regimes of  $p_1, p_2 \in (0, 1)^2$ . Diamantidis, Konstantopoulos, and Yuan (2025) used a different proof technique to show the same result, albeit for a slightly restricted set of edge densities.

**Independent set.** LCIS is closely related to the much-studied problem of finding large independent sets. In the past decade a line of work has focused on giving evidence of the computational hardness of finding large independent sets in sparse random graphs  $G \sim \mathbb{G}(n, d/n)$ . Gamarnik and Sudan (2017) showed that no local algorithms can find independent sets with size above some multiplicative factor  $\beta$  of optimal, for  $\beta = 1/2 + 1/(2\sqrt{2}) \approx 0.85$ . Rahman and Virág (2017) strengthened this result to show that local algorithms cannot produce independent sets larger than half-optimal size, which is tight. Soon after, Wein (2021) (building off of Gamarnik, Jagannath, and Wein (2024)) strengthened this result by extending the class of implicated algorithms to stable ones, which includes low-degree polynomials. Observing that a natural half-optimal approach (Grimmett and McDiarmid, 1975; Karp, 1976) is online (but not stable), Gamarnik, Kızıldağ, and Warnke (2025) proved that a natural class of online algorithms cannot beat half-optimality.

**Overlap gap properties.** All of the computational hardness results above rely on proving that the independent set problem exhibits a type of *overlap gap property* (OGP). At a high level, an OGP is a statement about the geometry of the solution space; it stipulates the non-existence of tuples of large solutions with a particular overlap pattern. Proving more sophisticated OGPs (for instance, allowing solutions to come from an “ensemble” of problem instances rather than a single one, or considering  $m$ -tuples of solutions rather than pairs) have allowed for tighter hardness results. For instance, Wein (2021) constructs an “interpolation path” of correlated random graphs  $G_1, \dots, G_T$ . The author argues by contradiction that, given a stable algorithm  $\mathcal{A}$  that outputs large solutions, one could run  $\mathcal{A}$  on each  $G_i$  and construct an  $m$ -tuple of solutions that violates an ensemble  $m$ -OGP.

Recently, Gamarnik, Kızıldağ, and Warnke (2025) introduced a novel interpolation argument, building an interpolation path of correlated inputs that are defined in reference to any algorithm  $\mathcal{A}$  in a class of online algorithms that they define. This differs from the above OGP hardness results in

that the non-existent structure is not of a tuple of solutions to the problem, but rather of a tuple of solutions *created by a particular type of algorithm*. As a result, the hardness proof is coextensive with the proof of the OGP, rather than coming afterward in, say, a proof by contradiction. This approach has several advantages, including that it establishes *strong hardness* (i.e., ruling out algorithms succeeding even with only  $o(1)$  probability), something that was elusive in OGP-style arguments prior to the work of [Huang and Sellke \(2025a\)](#). Our impossibility proof utilizes this novel interpolation argument by [Gamarnik, Kızıldağ, and Warnke \(2025\)](#); our main new ingredient is to prove an OGP for  $\text{LCIS}(n)$ .

**Computational hardness of random problems.** Our work fits into a broader landscape of using the overlap gap property to prove hardness results for random optimization problems. See [Du, Gong, and Huang \(2025\)](#); [Gamarnik, Kızıldağ, Perkins, and Xu \(2023\)](#); [Bhamidi, Gamarnik, and Gong \(2026\)](#) for further impossibility results for online algorithms. Other hardness results include [Gamarnik, Kızıldağ, Perkins, and Xu \(2022\)](#); [Huang and Sellke \(2025b,a\)](#). We note that such computation-to-optimization gaps have fundamental connections with statistical-computation gaps in planted problems which are widespread in high-dimensional learning. We do not attempt to survey the literature here — see [Gamarnik \(2021, 2025\)](#) for recent surveys and more.

#### 1.4. Open problems

We leave open several natural questions for future work.

**General dense case.** In this work we assumed that both graphs have edge density  $1/2$ . For arbitrary edge densities  $(p_1, p_2) \in (0, 1)^2$ , [Surya, Warnke, and Zhu \(2025\)](#) determined that the size of the largest common induced subgraph is roughly  $C_{p_1, p_2} \log_2 n$ , where interestingly the constant  $C_{p_1, p_2}$  takes on qualitatively different forms in three different regimes of  $(p_1, p_2) \in (0, 1)^2$ . What can we say about efficient algorithms in this more general setting? Is greedy still half-optimal, and does the impossibility proof extend naturally to this setting?

**Sparse case.** Can we find evidence of computational hardness for  $\text{LCIS}$  given  $G_1, G_2 \sim \mathbb{G}(n, d/n)$ ? Do these mirror analogous results for large independent sets in sparse graphs?

**Other classes of algorithms.** Can we rule out other classes of algorithms? For instance, it would be natural to consider online algorithms with exceptional edge-pairs, as well as stable algorithms.

**Many graphs.** What can we say about  $\text{LCIS}$  for many graphs? A quick calculation suggests that a greedy algorithm remains half-optimal; it would be interesting to explore this setting further.

**Average-case reduction from finding large independent sets.** As discussed above, the results that we obtained for  $\text{LCIS}(n)$  have similarities to those for finding large independent sets in  $G \sim \mathbb{G}(n, 1/2)$ ; in particular, a greedy algorithm is half-optimal in both settings, and conjecturally there is no efficient algorithm that does better than half-optimal. Is there a deeper connection between these two problems? In particular, is there an average-case reduction from finding large independent sets to finding large common induced subgraphs (or vice versa)?

One attempt could be to consider independent sets in  $G \sim \mathbb{G}(n^2, 1/2)$ , noting that the largest independent set in  $G$  has size roughly  $2 \log_2(n^2) = 4 \log_2 n$ , which is the same as the largest common induced subgraph in  $(G_1, G_2) \sim \mathbb{G}(n, 1/2)^{\otimes 2}$ . Moreover, the  $n^2$  vertices of  $G$  have a natural correspondence with ordered pairs of vertices  $(u, v) \in V(G_1) \times V(G_2)$ . It is not clear how to use these ideas to build a formal average-case reduction; doing so would be very interesting.

## 1.5. Notation

We write  $G \cong H$  to denote that two graphs  $G$  and  $H$  are isomorphic. For a graph  $G$  and a subset  $S \subseteq V(G)$ , we let  $G[S]$  denote the subgraph of  $G$  induced by  $S$ . We let  $[n] := \{1, \dots, n\}$ . The size  $|\sigma|$  of a solution  $\sigma = (H_1, H_2)$  to the LCIS problem is the number of vertices in each subgraph  $|\sigma| = |V(H_1)| = |V(H_2)|$ . We let  $\mathbf{1}\{\mathcal{E}\}$  denote the indicator random variable of an event  $\mathcal{E}$ . We use standard asymptotic notation throughout.

## 1.6. Outline

The rest of the paper is organized as follows. In Section 2 we describe the high level proof ideas of Theorems 3 and 5, focusing on the more involved proof of Theorem 5. Section 3 contains the analysis of the greedy algorithm and proves Theorem 3. Finally, Appendix A contains the proof of the impossibility result, Theorem 5.

## 2. Proof ideas

### 2.1. Greedy algorithm

In Section 3 we prove Theorem 3, that the algorithm GREEDY from Algorithm 1 produces half-optimal solutions to  $\text{LCIS}(n)$ . Recall that GREEDY builds up the common induced subgraph iteratively: at step  $t \in [n]$ , GREEDY processes the vertex pair  $(u_t, v_t) \in V(G_1) \times V(G_2)$ . If there exists some already-processed vertex  $v_j$  (i.e., with  $j \leq t$ ) such that  $(u_t, v_j)$  can be added to the partial solution, it does so — and similarly for any  $(u_j, v_t)$ .

Our goal is to give an upper bound on the probability that GREEDY outputs a common induced subgraph of size less than  $(2 - \varepsilon) \log_2 n$ . We observe that if this happens, then there must be a window of  $\tilde{n}$  successive steps where GREEDY does not add to the partial solution, for

$$\tilde{n} := \left\lfloor \frac{n}{2 \log_2 n} \right\rfloor.$$

In other words, over the course of the window, GREEDY considers and rejects at least  $\tilde{n}^2$  vertex pairs  $(u, v) \in V(G_1) \times V(G_2)$ . Let  $1 \leq k < (2 - \varepsilon) \log_2 n$  be the size of the partial solution at the start of the window — we will eventually take a union bound over all possible values of  $k$ . If each pair  $(u, v)$  were independent, the probability of rejecting all such pairs would be at most

$$\left(1 - \frac{1}{2^k}\right)^{\tilde{n}^2} \leq \exp\left(-\frac{\tilde{n}^2}{2^k}\right) < \exp(-\tilde{\Omega}(n^\varepsilon)),$$

which is small as desired. Of course, the vertex pairs are not all independent; we handle this via a careful conditioning argument.

### 2.2. The limits of online algorithms

In Appendix A we prove Theorem 5, that there is a vanishingly small probability that any  $\mathcal{A} \in \text{ONLINE}(n)$  can produce a solution to LCIS that is better than half-optimal. We show in Lemma 7 that it suffices to consider deterministic algorithms. At a high level, we relate the probability that  $\mathcal{A}$  finds a large solution to the probability that  $\mathcal{A}$  finds a large solution within every input from a

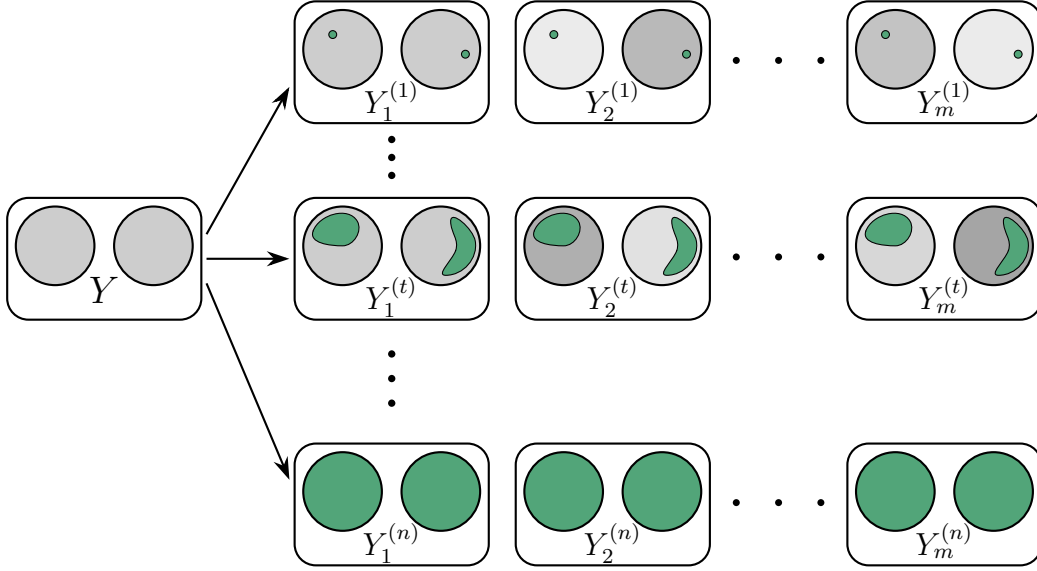


Figure 1: A family of interpolation paths based on  $\mathcal{A} \in \text{ONLINE}(n)$ ,  $Y \sim \mathbb{G}(n, 1/2)^{\otimes 2}$ , and  $m \in \mathbb{N}$ . For each  $t \in [n]$ , row  $t$  is an interpolation path  $Y_1^{(t)}, \dots, Y_m^{(t)}$ , with  $Y_1^{(t)} := Y$ . For graph  $j \in \{1, 2\}$  in each input, the green region corresponds to the vertices  $P_j^{(t)}$  that have been processed by  $\mathcal{A}(Y)$  after  $t$  steps. The edge-statuses of all pairs of nodes in the green regions are fixed for all graphs in the family. Aside from the first column (which are all  $Y$ ), the edge status of all other pairs is sampled independently from  $\text{Bern}(1/2)$ . Therefore each  $Y_i^{(t)} \sim \mathbb{G}(n, 1/2)^{\otimes 2}$ .

carefully-constructed set of correlated inputs, which will ensure that the solutions have substantial mutual overlap. Finally, we will use a combinatorial argument and the first moment method to show that such a set of solutions exists with vanishingly small probability.

In more detail, we say that a large solution  $\sigma$  to the  $\text{LCIS}(n)$  problem is one of size  $|\sigma| \geq (2 + \varepsilon) \log_2 n$ . We consider the event  $\mathcal{E}$ , that an arbitrary deterministic algorithm  $\mathcal{A}$  finds a large solution in an input  $Y = (G_1, G_2) \sim \mathbb{G}(n, 1/2)^{\otimes 2}$ . We start by relating the probability of  $\mathcal{E}$  to the probability of the event  $\mathcal{S}$ , that  $\mathcal{A}$  finds a set  $\{\sigma_1, \dots, \sigma_m\}$  of large solutions from a set of inputs called an *interpolation path* (Figure 1 and Definition 8). An interpolation path  $Y_1^{(t)}, \dots, Y_m^{(t)}$  is constructed so that on each input  $Y_i^{(t)}$ ,  $\mathcal{A}$  behaves identically for the first  $t \in [n]$  steps of the process, and thereafter independently, conditioned on the information learned in the first  $t$  steps. Moreover, the marginal distribution of each input is  $\mathbb{G}(n, 1/2)^{\otimes 2}$ .

In Lemma 11, we show that for any constant  $m \in \mathbb{N}$ , we have that

$$\mathbb{P}(\mathcal{E})^m \leq \mathbb{P}(\mathcal{S}).$$

The proof uses the conditional independence of each  $\mathcal{A}(Y_i^{(t)})$  after  $t$  steps and an application of Jensen's inequality.

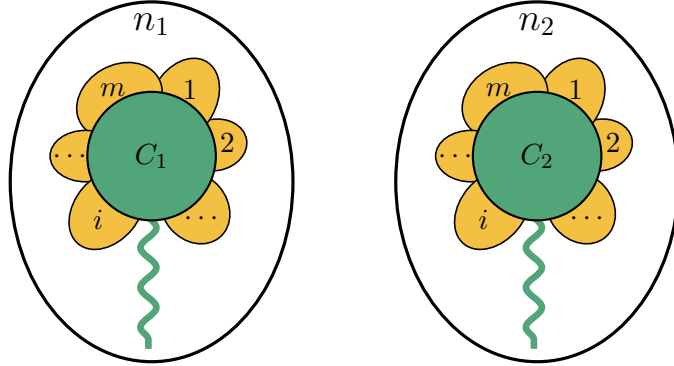


Figure 2:  $X_{m,t}$  counts the number of sets of large solutions  $\{\sigma_i\}_{i \in [m]}$  to the LCIS( $n$ ) problem with a particular overlap pattern; we visualize one such set here. Each  $\sigma_i := (H_{i,1}, H_{i,2})$  is a pair of isomorphic subgraphs of  $Y_i^{(t)} = (G_{i,1}^{(t)}, G_{i,2}^{(t)})$ , with  $|\sigma_i| \geq (2 + \varepsilon) \log_2 n$ . To see the overlap pattern, we overlay each  $V(H_{i,j})$  onto  $n_j := [n]$ , where the vertices  $P_j^{(t)}$  processed by  $\mathcal{A}$  through step  $t$  are colored green. Each  $V(H_{i,j})$  is the union of the yellow petal  $i$  in  $n_j$  and the green disc  $C_j$ , of size  $|C_j| = \lceil (2 - \varepsilon/2) \log_2 n \rceil$ . Crucially, note that  $C_j = V(H_{i,j}) \cap P_j^{(t)}$  for all  $i \in [m]$ .

For any  $m \in \mathbb{N}$  and  $t \in [n]$ , the random variable  $X_{m,t}$  counts the number of sets of large solutions from the interpolation path  $Y_1^{(t)}, \dots, Y_m^{(t)}$  that have mutual overlap of size at least  $\lceil \gamma L \rceil$  in each subgraph, for

$$\gamma := 2 - \frac{\varepsilon}{2} \quad \text{and} \quad L := \log_2 n.$$

Intuitively, these sets should look like a pair of flowers, where each flower has  $m$  petals and a central face corresponding to the overlapping portion — see Figure 2 and Definition 10.

In Lemma 12, we show that

$$\mathbb{P}(\mathcal{S}) \leq \sum_{t=1}^n \mathbb{P}(X_{m,t} \geq 1).$$

In fact, the event  $\mathcal{S}$  states that  $\mathcal{A}$  is run on each input  $Y_1^{(t)}, \dots, Y_m^{(t)}$ , for a particular  $t = \tau \in [n]$ . The proof of Lemma 12 relies on setting  $\tau$  to be the first time step at which  $\mathcal{A}(Y)$  finds a solution of size  $\lceil \gamma L \rceil$  (or  $\tau = n$  if such a solution is never found).

With this definition in hand, the proof of Lemma 12 is relatively straightforward:  $\mathcal{S}$  is the event that  $\mathcal{A}$  finds a large solution in each correlated input  $Y_i^{(\tau)}$  and we have defined  $Y_1^{(\tau)}, \dots, Y_m^{(\tau)}$  such that the first  $\lceil \gamma L \rceil$  nodes in each solution are identical. Therefore the solutions have substantial mutual overlap, so  $X_{m,\tau} \geq 1$ .

We call these  $m$ -tuples of overlapping solutions *forbidden structures*, since we show in Lemma 13 that they exist with vanishing probability as  $n \rightarrow \infty$ :

$$\sum_{t=1}^n \mathbb{P}(X_{m,t} \geq 1) \leq n^{-\log_2 n + o(\log n)}.$$

In essence, Lemma 13 says that the LCIS problem exhibits an overlap gap property — it stipulates the non-existence of certain tuples of large solutions with a particular overlap pattern. Similar to other proofs of the overlap gap property, we prove Lemma 13 by the first moment method and analyzing  $\mathbb{E}[X_{m,t}]$ . To enable our analysis, we say that each  $\sigma_i$  has size  $\alpha_i L$ , for some

$$\alpha_i \geq 2 + \varepsilon.$$

Throughout the proof we use the standard upper bound  $\binom{n}{k} \leq n^k$ . We can then upper bound the number of such forbidden structures by

$$\left[ \binom{n}{\lceil \gamma L \rceil} \prod_{i=1}^m \binom{n}{(\alpha_i - \gamma)L} \right]^2 \leq n^{2L(\gamma + \sum_{i=1}^m (\alpha_i - \gamma))}, \quad (5)$$

since the first term counts the number of choices for the set of common vertices of size  $\lceil \gamma L \rceil$  in each subgraph, and the second term counts the number of choices for the leftover  $(\alpha_i - \gamma)L$  vertices in each subgraph of each solution  $\sigma_i$ .

To analyze the probability that a given forbidden structure is in fact a set of solutions to  $\text{LCIS}(n)$ , we consider each  $\sigma_i = (H_{i,1}, H_{i,2})$ , fix a bijection  $\pi_i : V(H_{i,1}) \rightarrow V(H_{i,2})$ , and calculate the probability that  $\pi_i$  is a graph isomorphism, conditioned on the subgraph already seen by  $\mathcal{A}$  through step  $t$ . Ultimately, we take a union bound over all such permutations  $\pi_i$ , but since the subgraphs are of size  $O(\log n)$ , this contributes a multiplicative factor of only  $(\log n)^{\log n} \approx n^{\log \log n} < n^{o(L)}$ , which is negligible. Since each  $Y_i^{(t)} \sim \mathbb{G}(n, 1/2)^{\otimes 2}$ ,  $\pi_i$  preserves adjacencies independently with probability  $1/2$  for every pair of vertex pairs  $(u, u') \in \binom{V(H_{i,1})}{2}$  and  $(\pi_i(u), \pi_i(u')) \in \binom{V(H_{i,2})}{2}$ , so long as the connectivity of at least one of the pairs has not been determined by our conditioning. There are at least  $\binom{\alpha_i L}{2} - \binom{\lceil \gamma L \rceil}{2}$  such pairs (this uses that  $t \leq n - \Omega(\log n)$ ; the case when  $t$  is very close to  $n$  is simple to handle separately), so since each  $Y_i^{(t)}$  is conditionally independent, we can upper bound the probability that all  $\sigma_i$  are solutions by

$$\prod_{i=1}^m 2^{-\left(\binom{\alpha_i L}{2} - \binom{\lceil \gamma L \rceil}{2}\right)} \approx 2^{-\frac{L^2}{2} \sum_{i=1}^m (\alpha_i^2 - \gamma^2)} = n^{-\frac{L}{2} \sum_{i=1}^m (\alpha_i^2 - \gamma^2)}, \quad (6)$$

where the  $\approx$  symbol means we have dropped lower-order terms in the exponent. Leveraging the linearity of expectation together with estimates (5) and (6), we get that

$$\mathbb{E}[X_{m,t}] \leq n^{-\frac{L}{2}(-4\gamma + \sum_{i=1}^m [(\alpha_i^2 - 4\alpha_i) - (\gamma^2 - 4\gamma)])}.$$

Moreover, since  $\alpha_i^2 - 4\alpha_i \geq \varepsilon^2 - 4$  for all  $\alpha_i \geq 2 + \varepsilon$  and also  $\gamma^2 - 4\gamma = \varepsilon^2/4 - 4$ , we have that

$$-4\gamma + \sum_{i=1}^m [(\alpha_i^2 - 4\alpha_i) - (\gamma^2 - 4\gamma)] \geq -8 + m \left( \frac{3\varepsilon^2}{4} \right).$$

So by choosing  $m := \lceil 40/(3\varepsilon^2) \rceil$  we obtain  $\mathbb{E}[X_{m,t}] \leq n^{-L}$ , which proves Lemma 13. Chaining together Lemmas 11, 12, and 13, we get that

$$\mathbb{P}(\mathcal{E})^m \leq \mathbb{P}(\mathcal{S}) \leq \sum_{t=1}^n \mathbb{P}(X_{m,t} \geq 1) \leq n^{-\log_2 n + o(\log n)} \quad (7)$$

which immediately proves Theorem 5.

### 3. Analysis of the greedy algorithm

In this section we analyze the GREEDY algorithm described in Algorithm 1 and prove Theorem 3.

---

**Algorithm 1** Greedy algorithm for finding common induced subgraphs

---

**Input:** Two graphs  $G_1$  and  $G_2$ .

**Output:** Two induced subgraphs  $H_1 \subseteq G_1$  and  $H_2 \subseteq G_2$  such that  $H_1 \cong H_2$ , and a graph isomorphism  $\pi : V(H_1) \rightarrow V(H_2)$ .

**Procedure** Greedy( $G_1, G_2$ ):

```

  Let  $u_1, \dots, u_n$  be an arbitrary ordering of  $V(G_1)$ .
  Let  $v_1, \dots, v_n$  be an arbitrary ordering of  $V(G_2)$ .
  Initialize  $S_1 := \emptyset$  and  $S_2 := \emptyset$ .
  Initialize  $\pi : S_1 \rightarrow S_2$  to be the empty mapping.
  for  $i \in [n]$  do
    AddedPair := False
     $j := 1$ 
    while AddedPair == False and  $j \leq i$  do
      if  $v_j \notin S_2$  and IdenticalConnections( $u_i, v_j, S_1, S_2, \pi, G_1, G_2$ ) then
         $S_1 \leftarrow S_1 \cup \{u_i\}$ 
         $S_2 \leftarrow S_2 \cup \{v_j\}$ 
         $\pi(u_i) := v_j$ 
        AddedPair ← True
      else if  $u_j \notin S_1$  and IdenticalConnections( $u_j, v_i, S_1, S_2, \pi, G_1, G_2$ ) then
         $S_1 \leftarrow S_1 \cup \{u_j\}$ 
         $S_2 \leftarrow S_2 \cup \{v_i\}$ 
         $\pi(u_j) := v_i$ 
        AddedPair ← True
      else
         $j \leftarrow j + 1$ 
      end
    end
  end
   $H_1 := G_1[S_1]$ 
   $H_2 := G_2[S_2]$ 
  return ( $H_1, H_2, \pi$ )

```

**Procedure** IdenticalConnections( $u, v, S_1, S_2, \pi, G_1, G_2$ ):

| return  $\forall u' \in S_1 : (u, u') \in E(G_1) \iff (v, \pi(u')) \in E(G_2)$

---

**Proof** [of Theorem 3] Throughout the execution of GREEDY the induced subgraphs  $G_1[S_1]$  and  $G_2[S_2]$  are isomorphic by construction, with  $\pi : S_1 \rightarrow S_2$  being a graph isomorphism. Consequently, this remains true for the output of GREEDY too.

We now turn to the lower bound on the size of the output. We first introduce some helpful notation. Let  $S_1(t)$  denote the set  $S_1$  after GREEDY processes the vertex pair  $(u_t, v_t)$ , and define  $S_2(t)$  analogously. For  $k \in \mathbb{N}$ , let  $T_k := \min \{t : |S_1(t)| = k\}$  denote the first time when  $S_1$  has size  $k$ . For convenience we set  $T_0 := 0$ , note that  $T_1 = 1$ , and by convention we set  $T_k = \infty$  if

$|S_1(n)| < k$ . To abbreviate expressions that appear throughout the proof, set

$$\ell := \lceil (2 - \varepsilon) \log_2 n \rceil \quad \text{and} \quad \tilde{n} := \left\lfloor \frac{n}{2 \log_2 n} \right\rfloor,$$

and note that  $\ell \tilde{n} \leq n$  (assuming  $\varepsilon \geq 1/\log_2(n)$ ).

With the notation above, the size of the output of GREEDY is  $|S_1(n)|$  and our goal is to show that  $|S_1(n)| \geq (2 - \varepsilon) \log_2 n$  with high probability. Observe that  $|S_1(n)| \geq (2 - \varepsilon) \log_2 n$  if and only if  $T_\ell \leq n$ . Note also that if  $T_\ell > n$ , then there exists  $k \in \{1, \dots, \ell - 1\}$  such that  $T_k \leq k\tilde{n}$  and  $T_{k+1} > (k+1)\tilde{n}$ . Thus by a union bound we have that

$$\mathbb{P}(|S_1(n)| < (2 - \varepsilon) \log_2 n) = \mathbb{P}(T_\ell > n) \leq \sum_{k=1}^{\ell-1} \mathbb{P}(T_k \leq k\tilde{n}, T_{k+1} > (k+1)\tilde{n}). \quad (8)$$

In the remainder of the proof we fix  $k \in \{1, \dots, \ell - 1\}$  and bound the probability on the right hand side of (8). To do so, we condition on the sets  $S_1(k\tilde{n})$  and  $S_2(k\tilde{n})$ , obtaining that

$$\mathbb{P}(T_k \leq k\tilde{n}, T_{k+1} > (k+1)\tilde{n}) = \mathbb{E}[\mathbb{P}(T_k \leq k\tilde{n}, T_{k+1} > (k+1)\tilde{n} \mid S_1(k\tilde{n}), S_2(k\tilde{n}))].$$

Note that  $\mathbb{P}(T_k \leq k\tilde{n}, T_{k+1} > (k+1)\tilde{n} \mid S_1(k\tilde{n}), S_2(k\tilde{n})) = 0$  unless  $|S_1(k\tilde{n})| = |S_2(k\tilde{n})| = k$ . Thus in what follows we may and will assume that  $|S_1(k\tilde{n})| = |S_2(k\tilde{n})| = k$ .

Let  $U_k := \{u_{k\tilde{n}+1}, \dots, u_{(k+1)\tilde{n}}\}$  and  $V_k := \{v_{k\tilde{n}+1}, \dots, v_{(k+1)\tilde{n}}\}$ , and note that  $|U_k| = |V_k| = \tilde{n}$ . For  $u \in U_k$ , let  $X_{1,k}(u) \in \{0, 1\}^k$  be the random variable encoding the connectivity of  $u$  to the vertices in  $S_1(k\tilde{n})$ ; that is, the coordinates of  $X_{1,k}(u)$  are the indicator random variables  $\mathbf{1}\{(u, u') \in E(G_1)\}$  for  $u' \in S_1(k\tilde{n})$ . For  $v \in V_k$ , define  $X_{2,k}(v) \in \{0, 1\}^k$  analogously, and in particular order the coordinates in a way such that if the  $i$ -th coordinate of  $X_{1,k}(u)$  corresponds to the indicator random variable  $\mathbf{1}\{(u, u') \in E(G_1)\}$  for some  $u' \in S_1(k\tilde{n})$ , then the  $i$ -th coordinate of  $X_{2,k}(v)$  corresponds to the indicator random variable  $\mathbf{1}\{(v, \pi(u')) \in E(G_2)\}$ . With these definitions, observe that if  $T_k \leq k\tilde{n}$  and  $T_{k+1} > (k+1)\tilde{n}$ , then for every  $u \in U_k$  and for every  $v \in V_k$  we have that  $X_{1,k}(u) \neq X_{2,k}(v)$  (since if it were the case that  $X_{1,k}(u) = X_{2,k}(v)$  for some  $u \in U_k$  and  $v \in V_k$ , then GREEDY would add  $u$  to  $S_1$  and  $v$  to  $S_2$ , which contradicts  $T_{k+1} > (k+1)\tilde{n}$ ). Since  $\{X_{1,k}(u)\}_{u \in U_k}$  and  $\{X_{2,k}(v)\}_{v \in V_k}$  are all independent of  $S_1(k\tilde{n})$  and  $S_2(k\tilde{n})$ , we thus have that

$$\mathbb{P}(T_k \leq k\tilde{n}, T_{k+1} > (k+1)\tilde{n} \mid S_1(k\tilde{n}), S_2(k\tilde{n})) \leq \mathbb{P}(\forall u \in U_k \forall v \in V_k : X_{1,k}(u) \neq X_{2,k}(v)). \quad (9)$$

To bound the probability in the display above, we can first condition on  $\{X_{1,k}(u)\}_{u \in U_k}$  and then use that  $\{X_{2,k}(v)\}_{v \in V_k}$  are i.i.d. uniform to obtain that

$$\begin{aligned} & \mathbb{P}(\forall u \in U_k \forall v \in V_k : X_{1,k}(u) \neq X_{2,k}(v)) \\ &= \mathbb{E}[\mathbb{P}(\forall u \in U_k \forall v \in V_k : X_{1,k}(u) \neq X_{2,k}(v) \mid \{X_{1,k}(u)\}_{u \in U_k})] \\ &= \mathbb{E}\left[\mathbb{P}(\forall v \in V_k : X_{2,k}(v) \neq \bigwedge_{u \in U_k} X_{1,k}(u) \mid \{X_{1,k}(u)\}_{u \in U_k})^{\tilde{n}}\right] \\ &= \mathbb{E}\left[\left(1 - \frac{|\{X_{1,k}(u) : u \in U_k\}|}{2^k}\right)^{\tilde{n}}\right], \end{aligned} \quad (10)$$

where in the second-to-last line  $v$  is an arbitrary vertex in  $V_k$ . In the rest of the proof we bound the quantity in (10) in two different ways, depending on whether  $k$  is small or large.

First, assume that  $k \leq (1 - \varepsilon/3) \log_2 n$ . Note that the quantity in the expectation in (10) is at most 1, and it is equal to 0 precisely when  $|\{X_{1,k}(u) : u \in U_k\}| = 2^k$ . Consequently, we can bound (10) by the probability that  $|\{X_{1,k}(u) : u \in U_k\}| < 2^k$ , obtaining that

$$\begin{aligned} \mathbb{E} \left[ \left( 1 - \frac{|\{X_{1,k}(u) : u \in U_k\}|}{2^k} \right)^{\tilde{n}} \right] &\leq \mathbb{P} \left( |\{X_{1,k}(u) : u \in U_k\}| < 2^k \right) \\ &= \mathbb{P} \left( \exists x \in \{0, 1\}^k : \forall u \in U_k : X_{1,k}(u) \neq x \right) \\ &\leq 2^k \left( 1 - \frac{1}{2^k} \right)^{\tilde{n}} \leq 2^k \exp \left( -\frac{\tilde{n}}{2^k} \right) \\ &\leq n \cdot \exp \left( 1 - \frac{n^{\varepsilon/3}}{2 \log_2 n} \right), \end{aligned}$$

where in the last two lines we used a union bound, that  $\{X_{1,k}(u)\}_{u \in U_k}$  are i.i.d. uniform, the inequality  $1 - x \leq e^{-x}$ , and also that  $2^k \leq n^{1-\varepsilon/3}$ .

Next, assume that  $(1 - \varepsilon/3) \log_2 n \leq k \leq (2 - \varepsilon) \log_2 n$ . In this case we argue that with high probability  $|\{X_{1,k}(u) : u \in U_k\}| \geq n^{1-\varepsilon/2}$ , which allows us to bound the expectation in (10). First, we bound the probability that  $|\{X_{1,k}(u) : u \in U_k\}| < n^{1-\varepsilon/2}$  using a simple union bound as follows:

$$\begin{aligned} \mathbb{P} \left( |\{X_{1,k}(u) : u \in U_k\}| < n^{1-\varepsilon/2} \right) &\leq \mathbb{P} \left( \exists S \subseteq \{0, 1\}^k : |S| = \lfloor n^{1-\varepsilon/2} \rfloor, \forall u \in U_k : X_{1,k}(u) \in S \right) \\ &\leq \binom{2^k}{\lfloor n^{1-\varepsilon/2} \rfloor} \left( \frac{\lfloor n^{1-\varepsilon/2} \rfloor}{2^k} \right)^{\tilde{n}} \leq (2^k)^{n^{1-\varepsilon/2}} (n^{-\varepsilon/6})^{\tilde{n}} \leq n^{2n^{1-\varepsilon/2} - \varepsilon \tilde{n}/6}, \end{aligned}$$

where we used that  $n^{1-\varepsilon/3} \leq 2^k \leq n^2$ . We thus have that

$$\mathbb{E} \left[ \left( 1 - \frac{|\{X_{1,k}(u) : u \in U_k\}|}{2^k} \right)^{\tilde{n}} \right] \leq \left( 1 - \frac{n^{1-\varepsilon/2}}{2^k} \right)^{\tilde{n}} + n^{2n^{1-\varepsilon/2} - \varepsilon \tilde{n}/6},$$

and using the bound  $2^k \leq n^{2-\varepsilon}$  and the inequality  $1 - x \leq e^{-x}$  we have that

$$\left( 1 - \frac{n^{1-\varepsilon/2}}{2^k} \right)^{\tilde{n}} \leq \left( 1 - \frac{1}{n^{1-\varepsilon/2}} \right)^{\tilde{n}} \leq \exp \left( -\frac{\tilde{n}}{n^{1-\varepsilon/2}} \right) \leq \exp \left( 1 - \frac{n^{\varepsilon/2}}{2 \log_2 n} \right).$$

Finally, putting together (8), (9), (10), and the bounds on (10) discussed above, we have that

$$\begin{aligned} \mathbb{P}(|S_1(n)| < (2 - \varepsilon) \log_2(n)) &\leq (2 \log_2 n) \cdot \max \left\{ n \cdot \exp \left( 1 - \frac{n^{\varepsilon/3}}{2 \log_2 n} \right), \exp \left( 1 - \frac{n^{\varepsilon/2}}{2 \log_2 n} \right) + n^{2n^{1-\varepsilon/2} - \varepsilon \tilde{n}/6} \right\} \quad (11) \end{aligned}$$

which is at most  $\exp(-n^{\varepsilon/4})$  whenever  $\varepsilon \geq \frac{13 \log \log n}{\log n}$  and  $n$  is large enough. ■

In fact, the error probability obtained in (11) vanishes as  $n \rightarrow \infty$  whenever  $\varepsilon = \varepsilon_n \geq \frac{C \log \log n}{\log n}$  for a constant  $C > 6$ . Noting that  $6/\log(2) \approx 8.656 < 9$ , we thus obtain the following result as a corollary (noting that the constant in front of the second-order  $\log \log n$  term has not been optimized).

**Corollary 6** *Let  $(G_1, G_2) \sim \mathbb{G}(n, 1/2)^{\otimes 2}$ , define GREEDY as in Algorithm 1, let  $(H_1, H_2, \pi)$  denote the output of  $\text{GREEDY}(G_1, G_2)$ . Then  $(H_1, H_2)$  is a solution to  $\text{LCIS}(n)$  and  $\pi$  is a graph isomorphism between  $H_1$  and  $H_2$ . Furthermore, the size of  $(H_1, H_2)$  is at least  $2 \log_2 n - 9 \log \log n$  with high probability.*

For completeness we include a brief analysis of the runtime of the GREEDY algorithm.

**Claim 1 (Runtime)** *Fix  $n \in \mathbb{N}$  and let  $G_1$  and  $G_2$  be two graphs with  $n$  vertices. The GREEDY algorithm (Algorithm 1) with input  $(G_1, G_2)$  runs in  $O(n^3)$  time. In addition, if  $(G_1, G_2) \sim \mathbb{G}(n, 1/2)^{\otimes 2}$ , then with high probability  $\text{GREEDY}(G_1, G_2)$  runs in  $O(n^2 \log n)$  time.*

**Proof** The for loop goes over  $i \in [n]$ , which gives a factor of  $n$ . For any  $i$ , the body of the while loop is executed at most  $i \leq n$  times. Finally, running the IDENTICALCONNECTIONS procedure takes  $O(|S_1|)$  time, which gives a factor of  $O(n)$ , since  $|S_1| \leq n$  in the worst case. Overall, these three factors combine to give a worst-case running time of  $O(n^3)$ .

In the average-case setting with high probability the largest common subgraph has size  $O(\log n)$ . Therefore, with high probability we have  $|S_1| = O(\log n)$  throughout the execution of the algorithm, so running the IDENTICALCONNECTIONS procedure takes  $O(|S_1|) = O(\log n)$  time throughout. ■

## Acknowledgments

This project was supported in part by the NSF-funded Institute for Data, Econometrics, Algorithms and Learning (IDEAL) through the grants NSF ECCS-2216970 and ECCS-2216912. We particularly thank the IDEAL Institute and Varun Gupta for developing and running the IDEAL Summer Research Exchange, where this work started in the summer of 2025. We are grateful to Vaidehi Srinivas for insightful and inspiring discussions concerning average-case reductions between large cliques and large common induced subgraphs. We also thank anonymous reviewers for helpful feedback.

## References

- Shankar Bhamidi, David Gamarnik, and Shuyang Gong. Finding a dense submatrix of a random matrix. sharp bounds for online algorithms. *Electronic Communications in Probability*, 31:1–12, 2026.
- Sourav Chatterjee and Persi Diaconis. Isomorphisms between random graphs. *Journal of Combinatorial Theory, Series B*, 160:144–162, 2023.

- Donatello Conte, Pasquale Foggia, Carlo Sansone, and Mario Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):265–298, 2004.
- Dimitris Diamantidis, Takis Konstantopoulos, and Linglong Yuan. A combinatorial approach to phase transitions in random graph isomorphism problems. Preprint, 2025. URL <https://arxiv.org/abs/2410.00214>.
- Hang Du, Shuyang Gong, and Rundong Huang. The algorithmic phase transition of random graph alignment problem. *Probability Theory and Related Fields*, 191:1233—1288, 2025.
- Hans-Christian Ehrlich and Matthias Rarey. Maximum common subgraph isomorphism algorithms and their applications in molecular science: a review. *Wiley Interdisciplinary Reviews (WIREs): Computational Molecular Science*, 1(1):68–79, 2011.
- David Gamarnik. The overlap gap property: A topological barrier to optimizing over random structures. *Proceedings of the National Academy of Sciences*, 118(41):e2108492118, 2021.
- David Gamarnik. Turing in the Shadows of Nobel and Abel: An Algorithmic Story Behind Two Recent Prizes. *Notices of the American Mathematical Society*, 72(5):485–493, 2025.
- David Gamarnik and Madhu Sudan. Limits of local algorithms over sparse random graphs. *The Annals of Probability*, 45(4):2353 – 2376, 2017.
- David Gamarnik, Eren C. Kızıldağ, Will Perkins, and Changji Xu. Algorithms and Barriers in the Symmetric Binary Perceptron Model. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 576–587. IEEE, 2022.
- David Gamarnik, Eren C. Kızıldağ, Will Perkins, and Changji Xu. Geometric Barriers for Stable and Online Algorithms for Discrepancy Minimization. In *The Thirty Sixth Annual Conference on Learning Theory (COLT)*, volume 195, pages 3231–3263. PMLR, 2023.
- David Gamarnik, Aukosh Jagannath, and Alexander S. Wein. Hardness of Random Optimization Problems for Boolean Circuits, Low-Degree Polynomials, and Langevin Dynamics. *SIAM Journal on Computing*, 53(1):1–46, 2024.
- David Gamarnik, Eren C. Kızıldağ, and Lutz Warnke. Optimal hardness of online algorithms for large independent sets. Preprint, 2025. URL <https://arxiv.org/abs/2504.11450>.
- Geoffrey R. Grimmett and Colin J.H. McDiarmid. On colouring random graphs. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 77, pages 313–324. Cambridge University Press, 1975.
- Brice Huang and Mark Sellke. Strong Low Degree Hardness for Stable Local Optima in Spin Glasses. Preprint, 2025a. URL <https://arxiv.org/abs/2501.06427>.
- Brice Huang and Mark Sellke. Tight Lipschitz hardness for optimizing mean field spin glasses. *Communications on Pure and Applied Mathematics*, 78(1):60–119, 2025b.
- Viggo Kann. On the approximability of the maximum common subgraph problem. In *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 375–388. Springer, 1992.

Richard M. Karp. The probabilistic analysis of some combinational search algorithms. Technical Report UCB/ERL M581, Apr 1976. URL <http://www2.eecs.berkeley.edu/Pubs/TechRpts/1976/28848.html>.

Mustazee Rahman and Bálint Virág. Local algorithms for independent sets are half-optimal. *The Annals of Probability*, 45(3):1543 – 1577, 2017.

Leander Schietgat, Jan Ramon, and Maurice Bruynooghe. A polynomial-time maximum common subgraph algorithm for outerplanar graphs and its application to chemoinformatics. *Annals of Mathematics and Artificial Intelligence*, 69:343–376, 2013.

Erlang Surya, Lutz Warnke, and Emily Zhu. Isomorphisms between dense random graphs. *Combinatorica*, 45(35):1–42, 2025.

Alexander S. Wein. Optimal low-degree hardness of maximum independent set. *Mathematical Statistics and Learning*, 4(3/4):221–251, 2021.

## Appendix A. Proof of impossibility for online algorithms

In this section we prove Theorem 5. To abbreviate notation, we write  $Y := (G_1, G_2) \sim \mathbb{G}(n, 1/2)^{\otimes 2}$  for the input pair of graphs. Note that Definition 4—and hence also Theorem 5—allows randomized algorithms. However, a simple averaging-based argument shows that it suffices to consider deterministic algorithms. We state this in the following lemma.

**Lemma 7** *For any randomized algorithm  $\mathcal{A}$ , there exists a random seed  $\omega^* \in \Omega$  such that the deterministic algorithm  $\mathcal{A}^*(Y) := \mathcal{A}(Y, \omega^*)$  satisfies*

$$\mathbb{P}_{Y, \omega}(|\mathcal{A}(Y, \omega)| \geq (2 + \varepsilon) \log_2 n) \leq \mathbb{P}_Y(|\mathcal{A}^*(Y)| \geq (2 + \varepsilon) \log_2 n). \quad (12)$$

**Proof** Using the law of total expectation, we have that

$$\mathbb{P}_{Y, \omega}(|\mathcal{A}(Y, \omega)| \geq (2 + \varepsilon) \log_2 n) = \mathbb{E}_\omega[\mathbb{P}_Y(|\mathcal{A}(Y, \omega)| \geq (2 + \varepsilon) \log_2 n)].$$

By an averaging argument, there exists some  $\omega^* \in \Omega$  such that  $\mathcal{A}^*(Y) := \mathcal{A}(Y, \omega^*)$  satisfies (12). ■

Due to Lemma 7, in order to prove Theorem 5, it suffices to prove it for deterministic online algorithms. In the following we thus only consider deterministic online algorithms. In particular, this means that the only source of randomness in what follows is the input  $Y$ .

We now formally define the interpolation paths and forbidden structures that will be key to our analysis. These definitions closely follow the analogous structures in Gamarnik et al. (2025), though we take care to adapt them to the “two-dimensional” nature of the LCIS( $n$ ) problem.

**Definition 8 (Interpolation paths)** *Fix a deterministic  $\mathcal{A} \in \text{ONLINE}(n)$ ,  $m \in \mathbb{N}$ , and sample  $Y = (G_1, G_2) \sim \mathbb{G}(n, 1/2)^{\otimes 2}$ . For  $t \in [n]$ , the  $t^{\text{th}}$  interpolation path is denoted  $Y_1^{(t)}, \dots, Y_m^{(t)}$ , where  $Y_i^{(t)} = (G_{i,1}^{(t)}, G_{i,2}^{(t)})$  and each  $G_{i,j}^{(t)}$  is a graph on vertex set  $[n]$ . First, set*

$$Y_1^{(t)} := Y \quad \forall t \in [n]. \quad (13)$$

The remaining inputs are constructed based on the behavior of  $\mathcal{A}$  on input  $Y$ . Let  $P_1^{(t)}$  and  $P_2^{(t)}$  denote the vertices in  $G_1$  and  $G_2$  (respectively) processed by  $\mathcal{A}$  on input  $Y$  after the first  $t$  steps. For every  $t \in [n]$ ,  $i \in [m]$ , and  $j \in \{1, 2\}$ , let  $e_{i,j}^{(t)}(\{u, v\})$  denote the edge status of the pair  $\{u, v\}$  in  $G_{i,j}^{(t)}$ . Then, for each  $t \in [n]$ ,  $i \in \{2, \dots, m\}$ , and  $j \in \{1, 2\}$ , we set

$$e_{i,j}^{(t)}(\{u, v\}) := e_{1,j}^{(t)}(\{u, v\}) \quad \forall \{u, v\} \in \binom{P_j^{(t)}}{2}, \quad (14)$$

$$e_{i,j}^{(t)}(\{u, v\}) \stackrel{iid}{\sim} \text{Bern}(1/2) \quad \forall \{u, v\} \notin \binom{P_j^{(t)}}{2}, \quad (15)$$

where all of the random bits in (15) are independent of everything else.

See Figure 1 for an illustration of these interpolation paths.

**Remark 9** Define the family of interpolation paths  $\{Y_1^{(t)}, \dots, Y_m^{(t)}\}_{t \in [n]}$  as above. Let  $\Gamma_i^{(t)} \in \{0, 1\}^{2 \binom{P_j^{(t)}}{2}}$  encode the edge status of every pair of vertices that  $\mathcal{A}$  has inspected after  $t$  steps on input  $Y_i^{(t)}$ . Then (13) and (14) ensure that

$$\Gamma^{(t)} := \Gamma_1^{(t)} = \dots = \Gamma_m^{(t)}.$$

Since  $\mathcal{A}$  makes deterministic decisions based only on the information it has received by that point, we have that  $\mathcal{A}^{(t)}(Y_i^{(t)}) = \mathcal{A}^{(t)}(Y_{i'}^{(t)})$  for all  $i, i' \in [m]$ . Moreover, (15) ensures that the inputs  $\{Y_i^{(t)}\}_{i \in [m]}$  are mutually independent (and identically distributed), conditioned on  $\Gamma^{(t)}$ . Combining this with (13), we have that marginally

$$Y_i^{(t)} \sim \mathbb{G}(n, 1/2)^{\otimes 2} \quad \forall i \in [m], t \in [n].$$

**Definition 10 (Forbidden structures)** Fix a deterministic algorithm  $\mathcal{A} \in \text{ONLINE}(n)$ ,  $m \in \mathbb{N}$ , and  $\varepsilon > 0$ , let  $Y = (G_1, G_2) \sim \mathbb{G}(n, 1/2)^{\otimes 2}$ , and consider a family of interpolation paths  $\{Y_1^{(t)}, \dots, Y_m^{(t)}\}_{t \in [n]}$ . For any  $t \in [n]$ , let  $X_{m,t}$  count the number of  $m$ -tuples  $\{\sigma_1, \dots, \sigma_m\}$  such that for every  $i \in [m]$  the following two properties hold.

1. (Large solutions.) We have that  $\sigma_i = (H_{i,1}, H_{i,2})$  is a solution to the  $\text{LCIS}(n)$  problem on instance  $Y_i^{(t)}$ , with

$$|\sigma_i| \geq (2 + \varepsilon) \log_2 n.$$

2. (Overlap structure.) For  $j \in \{1, 2\}$  and  $i \in [m]$  we have that

$$\begin{aligned} V(H_{1,j}) \cap P_j^{(t)} &= V(H_{i,j}) \cap P_j^{(t)}, \\ |V(H_{1,j}) \cap P_j^{(t)}| &= \lceil (2 - \varepsilon/2) \log_2 n \rceil. \end{aligned}$$

See Figure 2 for an illustration of these forbidden structures. With these definitions we are now ready to prove Theorem 5.

**Proof** [of Theorem 5] Recall from Lemma 7 that we may and hence will assume that the algorithm  $\mathcal{A}$  is deterministic. Given  $\varepsilon > 0$ , we set the constant  $m$  to be

$$m := \left\lceil \frac{40}{3\varepsilon^2} \right\rceil. \quad (16)$$

For notational convenience, in all of the arguments that follow, let

$$\gamma := 2 - \frac{\varepsilon}{2} \quad \text{and} \quad L := \log_2 n.$$

Recall that the input to the algorithm  $\mathcal{A}$  is  $Y = (G_1, G_2) \sim \mathbb{G}(n, 1/2)^{\otimes 2}$ . Sample a family of interpolation paths  $\{Y_1^{(t)}, \dots, Y_m^{(t)}\}_{t \in [n]}$  based on  $\mathcal{A}$ ,  $m$ , and  $Y$  according to Definition 8. Define the stopping time  $\tau$  to be

$$\tau := \min \left\{ \min \left\{ 1 \leq t \leq n : \left| \mathcal{A}^{(t)}(Y) \right| = \lceil \gamma L \rceil \right\}, n \right\}. \quad (17)$$

In words,  $\tau$  is the first time when the algorithm  $\mathcal{A}$  on input  $Y$  reaches a (partial) solution of size *close* to half-optimal (specifically, of size at least  $(2 - \varepsilon/2) \log_2 n$ ). Next, define the events

$$\mathcal{E} := \{|\mathcal{A}(Y)| \geq (2 + \varepsilon)L\} \quad \text{and} \quad \mathcal{S} := \bigcap_{i=1}^m \left\{ \left| \mathcal{A} \left( Y_i^{(\tau)} \right) \right| \geq (2 + \varepsilon)L \right\}.$$

Observe that  $\mathcal{E}$  is the event that we care about: that algorithm  $\mathcal{A}$  on input  $Y$  finds a solution *beyond* half-optimal (specifically, of size at least  $(2 + \varepsilon) \log_2 n$ ). Furthermore,  $\mathcal{S}$  is the event that this is true for all of the inputs along the  $\tau$ -th interpolation path, namely for  $Y_1^{(\tau)}, \dots, Y_m^{(\tau)}$ .

Recall the definition of  $X_{m,t}$  from Definition 10. The core of the proof lies in the following chain of inequalities:

$$\mathbb{P}(\mathcal{E})^m \leq \mathbb{P}(\mathcal{S}) \leq \sum_{t=1}^n \mathbb{P}(X_{m,t} \geq 1) \leq n^{-\log_2 n + o(\log n)}. \quad (18)$$

We defer the proofs of these inequalities to Lemmas 11, 12, and 13, respectively, below. Theorem 5 then follows by taking an  $m$ -th root in (18), noting that  $m$  is constant (depending only on  $\varepsilon$ ).  $\blacksquare$

The rest of this section is devoted to proving the inequalities in (18). Lemma 11 is an analogue of (Gamarnik et al., 2025, Lemma 4.6); we include its proof here for completeness.

**Lemma 11** *Defining  $m$ ,  $\mathcal{E}$ , and  $\mathcal{S}$  as in the proof of Theorem 5, we have that  $\mathbb{P}(\mathcal{E})^m \leq \mathbb{P}(\mathcal{S})$ .*

**Proof** As in Remark 9, let  $\Gamma^{(t)}$  be the random variable that encodes all the information processed by  $\mathcal{A}$  through step  $t$  on input  $Y$ ; that is,  $\Gamma^{(t)}$  encodes the edge status of every pair  $\{u_1, v_1\} \in \binom{P_1^{(t)}}{2}$  in  $G_1$  and  $\{u_2, v_2\} \in \binom{P_2^{(t)}}{2}$  in  $G_2$ . Since  $\tau \in [n]$  deterministically, by the law of total probability we can write  $\mathbb{P}(\mathcal{E})$  as follows:

$$\begin{aligned} \mathbb{P}(\mathcal{E}) &= \sum_{t=1}^n \mathbb{P}(\mathcal{E} \cap \{\tau = t\}) = \sum_{t=1}^n \mathbb{E} \left[ \mathbb{P} \left( \mathcal{E} \cap \{\tau = t\} \mid \Gamma^{(t)} \right) \right] \\ &= \sum_{t=1}^n \mathbb{E} \left[ \mathbf{1}_{\{\tau = t\}} \mathbb{P} \left( \mathcal{E} \mid \Gamma^{(t)} \right) \right] = \mathbb{E} \left[ \sum_{t=1}^n \mathbf{1}_{\{\tau = t\}} \mathbb{P} \left( \mathcal{E} \mid \Gamma^{(t)} \right) \right], \end{aligned}$$

where the third equality follows because the event  $\{\tau = t\}$  is determined by  $\Gamma^{(t)}$ . Considering now the right-hand side of the desired inequality,  $\mathbb{P}(\mathcal{S})$ , the same reasoning gives that

$$\mathbb{P}(\mathcal{S}) = \mathbb{E} \left[ \sum_{t=1}^n \mathbf{1}\{\tau = t\} \mathbb{P} \left( \mathcal{E}_{1,t} \cap \dots \cap \mathcal{E}_{m,t} \mid \Gamma^{(t)} \right) \right],$$

where  $\mathcal{E}_{i,t} := \{|\mathcal{A}(Y_i^{(t)})| \geq (2 + \varepsilon) \log_2 n\}$ .

Now we use an important property of the construction of interpolation paths in Definition 8: all edges in  $Y_1^{(t)}, \dots, Y_m^{(t)}$  that are revealed after step  $t$  are i.i.d. In particular, this implies that conditioned on  $\Gamma^{(t)}$ , the events  $\mathcal{E}_{1,t}, \dots, \mathcal{E}_{m,t}$  are mutually independent (see Remark 9). Thus,

$$\mathbb{P} \left( \mathcal{E}_{1,t} \cap \dots \cap \mathcal{E}_{m,t} \mid \Gamma^{(t)} \right) = \prod_{i=1}^m \mathbb{P} \left( \mathcal{E}_{i,t} \mid \Gamma^{(t)} \right) = \mathbb{P} \left( \mathcal{E}_{1,t} \mid \Gamma^{(t)} \right)^m = \mathbb{P} \left( \mathcal{E} \mid \Gamma^{(t)} \right)^m.$$

Note that the random variable  $\mathbf{1}\{\tau = t\}$  is nonzero for exactly one term in the sum over  $t \in [n]$ . As such we have that

$$\sum_{t=1}^n \mathbf{1}\{\tau = t\} \mathbb{P} \left( \mathcal{E} \mid \Gamma^{(t)} \right)^m = \left( \sum_{t=1}^n \mathbf{1}\{\tau = t\} \mathbb{P} \left( \mathcal{E} \mid \Gamma^{(t)} \right) \right)^m.$$

Finally, since  $f(x) = x^m$  is convex for  $m \geq 1$  and  $x \geq 0$ , we can apply Jensen's inequality to get

$$\mathbb{P}(\mathcal{S})^m = \left( \mathbb{E} \left[ \sum_{t=1}^n \mathbf{1}\{\tau = t\} \mathbb{P} \left( \mathcal{E} \mid \Gamma^{(t)} \right) \right] \right)^m \leq \mathbb{E} \left[ \left( \sum_{t=1}^n \mathbf{1}\{\tau = t\} \mathbb{P} \left( \mathcal{E} \mid \Gamma^{(t)} \right) \right)^m \right] = \mathbb{P}(\mathcal{S}).$$

■

We now turn to the second inequality in (18). Lemma 12 is an analogue of (Gamarnik et al., 2025, Lemma 4.8); we include its proof here for completeness.

**Lemma 12** *Define  $\mathcal{S}$  as in the proof of Theorem 5 and  $X_{m,t}$  as in Definition 10. We have that*

$$\mathbb{P}(\mathcal{S}) \leq \sum_{t=1}^n \mathbb{P}(X_{m,t} \geq 1).$$

**Proof** As in the proof of Lemma 11, the law of total probability gives that

$$\mathbb{P}(\mathcal{S}) = \sum_{t=1}^n \mathbb{P}(\mathcal{S} \cap \{\tau = t\}),$$

so it suffices to show that for every  $t \in [n]$  the event  $\mathcal{S} \cap \{\tau = t\}$  implies  $X_{m,t} \geq 1$ . Fix any  $t \in [n]$  and suppose that  $\mathcal{S} \cap \{\tau = t\}$  occurs. For each  $i \in [m]$ , define

$$\sigma_i := \mathcal{A}(Y_i^{(\tau)}).$$

To see that  $\{\sigma_1, \dots, \sigma_m\}$  is a forbidden structure and hence  $X_{m,t} \geq 1$ , note that Property 1 of Definition 10 is directly implied by  $\mathcal{S}$  and Property 2 is implied by Remark 9 and  $\{\tau = t\}$ . ■

Finally, we turn to proving the third inequality in (18). This establishes a variant of the OGP for the LCIS( $n$ ) problem.

**Lemma 13** *Define  $X_{m,t}$  as in Definition 10. We have that*

$$\sum_{t=1}^n \mathbb{P}(X_{m,t} \geq 1) \leq n^{-\log_2 n + o(\log n)}.$$

We first deal with some simple edge cases in  $t$ . Note that if  $t < \lceil (2 - \varepsilon/2) \log_2 n \rceil$ , then  $X_{m,t} = 0$  deterministically. This is because then  $|V(H_{1,1}) \cap P_1^{(t)}| \leq |P_1^{(t)}| = t < \lceil (2 - \varepsilon/2) \log_2 n \rceil$ , and so Property 2 of Definition 10 cannot hold. Similarly, we claim that if

$$t > n - (\lceil (2 + \varepsilon) \log_2 n \rceil - \lceil (2 - \varepsilon/2) \log_2 n \rceil),$$

then  $X_{m,t} = 0$  deterministically. To see this, observe that if Property 2 of Definition 10 holds, then the solution  $\sigma_i$  at time  $n$  can have size at most  $\lceil (2 - \varepsilon/2) \log_2 n \rceil + (n - t)$ . Thus, in order for Property 1 to hold, we must have that  $\lceil (2 - \varepsilon/2) \log_2 n \rceil + (n - t) \geq (2 + \varepsilon) \log_2 n$ .

Thus in the following we focus on times  $t$  satisfying

$$\lceil (2 - \varepsilon/2) \log_2 n \rceil \leq t \leq n - (\lceil (2 + \varepsilon) \log_2 n \rceil - \lceil (2 - \varepsilon/2) \log_2 n \rceil). \quad (19)$$

For  $t \in [n]$  that does not satisfy (19), the previous paragraph shows that  $\mathbb{P}(X_{m,t} \geq 1) = 0$ .

To analyze  $X_{m,t}$ , it is useful to use a truncation argument; to this end, we introduce the following quantities.

**Definition 14** *Let  $W_{m,t}$  count the number of  $m$ -tuples  $\{\sigma_1, \dots, \sigma_m\}$  that satisfy Definition 10 and have  $\max_{i \in [m]} |\sigma_i| > 6 \log_2 n$ . Similarly, let  $Z_{m,t}$  count the number of  $m$ -tuples  $\{\sigma_1, \dots, \sigma_m\}$  that satisfy Definition 10 and have  $\max_{i \in [m]} |\sigma_i| \leq 6 \log_2 n$ .*

By definition, we have that  $X_{m,t} = W_{m,t} + Z_{m,t}$ . Therefore, we have that

$$\mathbb{P}(X_{m,t} \geq 1) \leq \mathbb{P}(W_{m,t} \geq 1) + \mathbb{P}(Z_{m,t} \geq 1), \quad (20)$$

and the following two claims bound these latter two quantities.

**Claim 2** *Define  $W_{m,t}$  as in Definition 14. For every  $t \in [n]$  we have that*

$$\mathbb{P}(W_{m,t} \geq 1) \leq mn^{-\log_2 n + o(\log n)}.$$

**Claim 3** *Define  $Z_{m,t}$  as in Definition 14. Suppose that  $t$  satisfies (19). Then we have that*

$$\mathbb{P}(Z_{m,t} \geq 1) \leq n^m n^{-\log_2 n + o(\log n)}.$$

Lemma 13 follows directly from Claims 2 and 3.

**Proof** [of Lemma 13] By (20) and Claims 2 and 3, we have that

$$\begin{aligned} \sum_{t=1}^n \mathbb{P}(X_{m,t} \geq 1) &\leq \sum_{t=1}^n \left( \mathbb{P}(W_{m,t} \geq 1) + \mathbb{P}(Z_{m,t} \geq 1) \right) \\ &\leq \sum_{t=1}^n \left( mn^{-\log_2 n + o(\log n)} + n^m n^{-\log_2 n + o(\log n)} \right) \\ &\leq 2mn^{m+1} n^{-\log_2 n + o(\log n)} \leq n^{-\log_2 n + o(\log n)}, \end{aligned}$$

where we use that  $a + b \leq 2ab$  for  $a, b \geq 1$ , and that  $m$  is a constant, so  $2mn^{m+1} \leq n^{o(\log n)}$ .  $\blacksquare$

It remains to prove the claims. Claim 2 follows from a straightforward union bound.

**Proof** [of Claim 2] Note that  $W_{m,t} \geq 1$  implies the existence of some  $Y_i^{(t)} = (G_{i,1}^{(t)}, G_{i,2}^{(t)})$  that contains a pair of subgraphs  $H_1 \subseteq G_{i,1}^{(t)}, H_2 \subseteq G_{i,2}^{(t)}$  such that  $H_1 \cong H_2$  and

$$k := |H_1| = |H_2| = \lceil 6 \log_2 n \rceil.$$

For any two subgraphs  $H_1, H_2 \sim \mathbb{G}(k, 1/2)$ , the probability that  $H_1 \cong H_2$  is at most  $(k!)2^{-\binom{k}{2}}$ . Taking a union bound over all possible choices of  $i \in [m]$  and subgraphs  $H_1, H_2$ , we see that

$$\begin{aligned} \mathbb{P}(W_{m,t} \geq 1) &\leq \sum_{i=1}^m \sum_{\substack{V(H_1) \subseteq [n], V(H_2) \subseteq [n] \\ |V(H_1)| = |V(H_2)| = k}} (k!)2^{-\binom{k}{2}} \leq m \binom{n}{k}^2 (k!)2^{-\binom{k}{2}} \\ &\leq m \left( n^2 2^{-(k-1)/2} \right)^k \leq m \left( \sqrt{2} n^{-1} \right)^k \leq mn^{-6 \log_2 n + o(\log n)}, \end{aligned}$$

where the third inequality uses that  $\binom{n}{k}^2 k! \leq n^{2k}$ , the fourth inequality uses that  $k \geq 6 \log_2 n$  and hence  $2^{-(k-1)/2} \leq \sqrt{2} n^{-3}$ , and the final inequality uses that  $k \geq 6 \log_2 n$  as well.  $\blacksquare$

The proof of Claim 3 is more involved. We use the following definition.

**Definition 15** For any  $m \in \mathbb{N}$  and  $\alpha = (\alpha_1, \dots, \alpha_m) \in [2 + \varepsilon, 6]^m$ , let  $Z_{\alpha, m, t}$  count the number of  $m$ -tuples  $\{\sigma_1, \dots, \sigma_m\}$  that satisfy Definition 10 and have  $|\sigma_i| = \alpha_i \log_2 n$  for all  $i \in [m]$ .

Note that we only consider  $\alpha$  such that  $\alpha \log_2 n \in [n]^m$ . By definition, we have that

$$Z_{m,t} = \sum_{\substack{\alpha \in [2+\varepsilon, 6]^m \\ \alpha \log_2 n \in [n]^m}} Z_{\alpha, m, t},$$

and so by a union bound we have that

$$\mathbb{P}(Z_{m,t} \geq 1) \leq \sum_{\substack{\alpha \in [2+\varepsilon, 6]^m \\ \alpha \log_2 n \in [n]^m}} \mathbb{P}(Z_{\alpha, m, t} \geq 1).$$

Let  $\alpha_{\max} := \max_{i \in [m]} \alpha_i$  and recall that  $L = \log_2 n$ . Observe that if  $t$  is such that  $n - t < \alpha_{\max} L - \lceil \gamma L \rceil$ , then  $Z_{\alpha, m, t} = 0$  deterministically. To see this, let  $i^*$  be such that  $\alpha_{i^*} = \alpha_{\max}$ . If Property 2 of Definition 10 holds, then the solution  $\sigma_{i^*}$  at time  $n$  can have size at most  $\lceil \gamma L \rceil + (n - t)$ , so in order for the solution  $\sigma_{i^*}$  at time  $n$  to have size  $\alpha_{i^*} L = \alpha_{\max} L$ , we must have that  $\lceil \gamma L \rceil + (n - t) \geq \alpha_{\max} L$ . Thus when analyzing  $Z_{\alpha, m, t}$ , we may assume that  $t$  satisfies  $\lceil \gamma L \rceil + (n - t) \geq \alpha_{\max} L$ , or in other words that

$$t \leq n - (\alpha_{\max} L - \lceil \gamma L \rceil). \quad (21)$$

Applying Markov's inequality thus yields that

$$\mathbb{P}(Z_{m,t} \geq 1) \leq \sum_{\substack{\alpha \in [2+\varepsilon, 6]^m, \alpha L \in [n]^m \\ t \leq n - (\alpha_{\max} L - \lceil \gamma L \rceil)}} \mathbb{P}(Z_{\alpha, m, t} \geq 1) \leq \sum_{\substack{\alpha \in [2+\varepsilon, 6]^m, \alpha L \in [n]^m \\ t \leq n - (\alpha_{\max} L - \lceil \gamma L \rceil)}} \mathbb{E}[Z_{\alpha, m, t}]. \quad (22)$$

We proceed to bound  $\mathbb{E}[Z_{\alpha, m, t}]$  for  $t$  such that (21) holds. Lemma 16 and Lemma 17 handle the counting and the probability estimates, respectively.

**Lemma 16** *Recall that  $\gamma = 2 - \varepsilon/2$  and  $L = \log_2 n$ . For any  $m \in \mathbb{N}$ ,  $\alpha \in [2 + \varepsilon, 6]^m$ , and  $(P_1^{(t)}, P_2^{(t)}) \subseteq [n]^2$ , the total number of  $m$ -tuples  $\{\sigma_1, \dots, \sigma_m\}$  that satisfy Property 2 in Definition 10 and have  $|\sigma_i| = \alpha_i \log_2 n$  for all  $i \in [m]$  is at most*

$$n^{2L[\gamma + \sum_{i=1}^m (\alpha_i - \gamma)]}.$$

**Proof** Recall that  $\sigma_i = (H_{i,1}, H_{i,2})$ . By Property 2 in Definition 10, for each  $i \in [m]$  and  $j \in \{1, 2\}$  we have that  $V(H_{i,j}) \cap P_j^{(t)} = V(H_{1,j}) \cap P_j^{(t)}$  and  $|V(H_{1,j}) \cap P_j^{(t)}| = \lceil \gamma L \rceil$ . To bound the number of  $m$ -tuples counted in the lemma statement, we pick any  $j \in \{1, 2\}$  and first note that the number of ways to choose the set of common vertices  $V(H_{1,j}) \cap P_j^{(t)}$  is at most

$$\binom{|P_j^{(t)}|}{\lceil \gamma L \rceil} \leq \binom{n}{\lceil \gamma L \rceil} \leq n^{\lceil \gamma L \rceil}.$$

Since the size of each  $H_{i,j}$  is  $\alpha_i L$ , the number of ways to choose the remaining vertices in each  $H_{i,j}$  is at most

$$\prod_{i=1}^m \binom{n}{\alpha_i L - \lceil \gamma L \rceil} \leq \prod_{i=1}^m n^{\alpha_i L - \lceil \gamma L \rceil} = n^{\sum_{i=1}^m (\alpha_i L - \lceil \gamma L \rceil)}.$$

As such, counting each  $j \in \{1, 2\}$  separately, we can upper bound the count by

$$n^{2\lceil \gamma L \rceil + 2 \sum_{i=1}^m (\alpha_i L - \lceil \gamma L \rceil)} \leq n^{2L[\gamma + \sum_{i=1}^m (\alpha_i - \gamma)]}$$

and the result follows. ■

**Lemma 17** *Recall that  $\gamma = 2 - \varepsilon/2$  and  $L = \log_2 n$ . Let  $m \in \mathbb{N}$  and  $\alpha \in [2 + \varepsilon, 6]^m$ , and define*

$$\Phi := \sum_{i=1}^m \left( \frac{\alpha_i^2 - \gamma^2}{2} \right).$$

*Let  $t$  be such that (21) holds. Let  $\Gamma^{(t)}$  be the random variable that encodes all the information processed by  $\mathcal{A}$  through step  $t$ . Then, for any  $m$ -tuple  $\{\sigma_1, \dots, \sigma_m\}$  that satisfies Property 2 in Definition 10 and has  $|\sigma_i| = \alpha_i L$  for all  $i \in [m]$ , we have that*

$$\mathbb{P} \left( \sigma_i \text{ is a common induced subgraph of } Y_i^{(t)} \text{ for all } i \in [m] \mid \Gamma^{(t)} \right) \leq n^{-\Phi L + C m \log L}$$

for an absolute constant  $C < \infty$ .

**Proof** First note that, as explained after Definition 15, the condition that  $t$  satisfies (21) is necessary in order for there to exist an  $m$ -tuple  $\{\sigma_1, \dots, \sigma_m\}$  that satisfies Property 2 in Definition 10 and has  $|\sigma_i| = \alpha_i L$  for all  $i \in [m]$ . (Otherwise, if  $t$  is such that (21) does not hold, then there is no such  $m$ -tuple  $\{\sigma_1, \dots, \sigma_m\}$ , and hence the claim is vacuously true.)

For  $j \in \{1, 2\}$ , define the set of common vertices

$$C_j := V(H_{1,j}) \cap P_j^{(t)}.$$

Since  $\{\sigma_1, \dots, \sigma_m\}$  satisfies Property 2 from Definition 10, we have that  $|C_j| = \lceil \gamma L \rceil$  and  $C_j \subseteq V(H_{i,j})$  for all  $i \in [m]$ . Consider a bijection  $\pi_i : V(H_{i,1}) \rightarrow V(H_{i,2})$ , for any  $i \in [m]$ . We will analyze the probability that  $\pi_i$  is a graph isomorphism, conditioned on  $\Gamma^{(t)}$ . Define

$$M_i := \{u \in C_1 : \pi_i(u) \in C_2\}$$

to be the set of common vertices in  $H_{i,1}$  that get mapped by  $\pi_i$  to common vertices in  $H_{i,2}$ . Note that  $|M_i| \leq \lceil \gamma L \rceil$ . Furthermore, define  $I_i(u, u')$  to be the indicator that  $\pi_i$  maintains the edge status of  $(u, u')$ , that is,

$$I_i(u, u') := \mathbf{1}\left\{e_{i,1}^{(n)}(\{u, u'\}) = e_{i,2}^{(n)}(\{\pi_i(u), \pi_i(u')\})\right\}.$$

Observe that  $\pi_i$  is a graph isomorphism if and only if  $I_i(u, u') = 1$  for all  $(u, u') \in \binom{V(H_{i,1})}{2}$ . Recall that  $\Gamma^{(t)}$  determines the edge statuses of all  $(u, u') \in \binom{M_i}{2}$  and their respective pairs  $(\pi_i(u), \pi_i(u'))$ , so for such pairs we have that  $\mathbb{P}(I_i(u, u') = 1 \mid \Gamma^{(t)}) \in \{0, 1\}$ . In particular, if  $\Gamma^{(t)}$  is such that  $I_i(u, u') = 0$  for some  $(u, u') \in \binom{M_i}{2}$ , then  $\mathbb{P}(\pi_i \text{ is a graph isomorphism} \mid \Gamma^{(t)}) = 0$ . However, in the argument that follows we will not use anything about  $\Gamma^{(t)}$  — merely that  $t$  satisfies (21). Observe that if  $(u, u') \in \binom{V(H_{i,1})}{2} \setminus \binom{M_i}{2}$ , then either  $(u, u') \notin \binom{P_1^{(t)}}{2}$  or  $(\pi_i(u), \pi_i(u')) \notin \binom{P_2^{(t)}}{2}$  (or both). Therefore, (15) in Definition 8 ensures that  $\{I_i(u, u') : (u, u') \in \binom{V(H_{i,1})}{2} \setminus \binom{M_i}{2}\}$  are i.i.d. Bern(1/2) random variables. Note further that<sup>5</sup>

$$\left| \binom{V(H_{i,1})}{2} \setminus \binom{M_i}{2} \right| = \binom{\alpha_i L}{2} - \binom{|M_i|}{2} \geq \binom{\alpha_i L}{2} - \binom{\lceil \gamma L \rceil}{2} \geq \frac{\alpha_i^2 - \gamma^2}{2} L^2 - 5L.$$

Thus we have that

$$\begin{aligned} \mathbb{P}\left(\pi_i \text{ is a graph isomorphism} \mid \Gamma^{(t)}\right) &\leq \prod_{(u, u') \in \binom{V(H_{i,1})}{2} \setminus \binom{M_i}{2}} \mathbb{P}\left(I_i(u, u') = 1 \mid \Gamma^{(t)}\right) \\ &= 2^{-\left(\binom{\alpha_i L}{2} - \binom{|M_i|}{2}\right)} \leq n^{-L(\alpha_i^2 - \gamma^2)/2 + 5}. \end{aligned}$$

Remark 9 implies that the event  $H_{i,1} \cong H_{i,2}$  is independent for each  $i \in [m]$ , conditioned on  $\Gamma^{(t)}$ . Therefore, taking a union bound over all bijections  $\pi_i$ , we have that

$$\begin{aligned} &\mathbb{P}\left(\sigma_i \text{ is a common induced subgraph of } Y_i^{(t)} \text{ for all } i \in [m] \mid \Gamma^{(t)}\right) \\ &\leq \prod_{i=1}^m \sum_{\pi_i} \mathbb{P}\left(\pi_i \text{ is an isomorphism} \mid \Gamma^{(t)}\right) \leq \prod_{i=1}^m n^{-L(\alpha_i^2 - \gamma^2)/2 + 5} (\alpha_i L)! \\ &\leq \prod_{i=1}^m n^{-L(\alpha_i^2 - \gamma^2)/2 + C \log L} = n^{-L(\sum_{i=1}^m (\alpha_i^2 - \gamma^2)/2) + Cm \log L} \end{aligned}$$

for an absolute constant  $C < \infty$ , where we used that  $(\alpha_i L)! \leq (6L)! \leq (6L)^{6L} \leq n^{C \log L}$ .  $\blacksquare$

5. The equalities and inequalities in this display implicitly use that  $t$  satisfies (21), which guarantees the existence of an  $m$ -tuple  $\{\sigma_1, \dots, \sigma_m\}$  with appropriate properties, in particular that  $|\sigma_i| = \alpha_i L$  and  $|M_i| \leq \lceil \gamma L \rceil$ .

**Proof** [of Claim 3] Recall that  $\gamma = 2 - \varepsilon/2$  and  $L = \log_2 n$ . For  $\alpha \in [2 + \varepsilon, 6]^m$ , define

$$f(\alpha_i, \gamma) := \left( \frac{\alpha_i^2}{2} - 2\alpha_i \right) - \left( \frac{\gamma^2}{2} - 2\gamma \right)$$

and

$$\Psi(\alpha) := -2\gamma + \sum_{i=1}^m f(\alpha_i, \gamma).$$

For any  $\alpha \in [2 + \varepsilon, 6]^m$  with  $\alpha L \in [n]^m$  and  $t$  satisfying (21), Lemmas 16 and 17 imply that

$$\begin{aligned} \mathbb{E}[Z_{\alpha, m, t}] &= \mathbb{E} \left[ \mathbb{E} \left[ Z_{\alpha, m, t} \mid \Gamma^{(t)} \right] \right] \leq \mathbb{E} \left[ n^{2L[\gamma + \sum_{i=1}^m (\alpha_i - \gamma)]} \cdot n^{-\Phi L + C m \log L} \right] \\ &= n^{-L\Psi(\alpha) + C m \log L} = n^{-L\Psi(\alpha) + o(L)}. \end{aligned}$$

Plugging this estimate into (22), we get that

$$\mathbb{P}(Z_{m, t} \geq 1) \leq \sum_{\substack{\alpha \in [2 + \varepsilon, 6]^m, \alpha L \in [n]^m \\ t \leq n - (\alpha_{\max} L - \lceil \gamma L \rceil)}} \mathbb{E}[Z_{\alpha, m, t}] \leq n^m \cdot n^{-L\Psi(\alpha) + o(L)},$$

so what remains is to show that  $\Psi(\alpha) \geq 1$  for  $\alpha \in [2 + \varepsilon, 6]^m$ .

We first note that

$$\gamma^2/2 - 2\gamma = \varepsilon^2/8 - 2.$$

Moreover, since for  $\alpha_i \geq 2 + \varepsilon$  the expression  $\alpha_i^2/2 - 2\alpha_i$  is minimized at  $\alpha_i = 2 + \varepsilon$ , we have that

$$\min_{\alpha_i \in [2 + \varepsilon, 6]} \left\{ \alpha_i^2/2 - 2\alpha_i \right\} = (2 + \varepsilon)^2/2 - 2(2 + \varepsilon) = \varepsilon^2/2 - 2.$$

Putting the previous two displays together, we thus have that

$$f(\alpha_i, \gamma) \geq \left( \frac{\varepsilon^2}{2} - 2 \right) - \left( \frac{\varepsilon^2}{8} - 2 \right) = \frac{3\varepsilon^2}{8}.$$

By our choice of  $m \geq 40/(3\varepsilon^2)$  in (16), we have that

$$\Psi(\alpha) = -2\gamma + \sum_{i=1}^m f(\alpha_i, \gamma) \geq -4 + m \left( \frac{3\varepsilon^2}{8} \right) \geq -4 + \left( \frac{40}{3\varepsilon^2} \right) \left( \frac{3\varepsilon^2}{8} \right) = 1,$$

which concludes the proof. ■