FINDING TIGHT HAMILTON CYCLES IN RANDOM HYPERGRAPHS FASTER

PETER ALLEN*, CHRISTOPH KOCH**, OLAF PARCZYK†, AND YURY PERSON‡

Abstract. In an r-uniform hypergraph on n vertices a tight Hamilton cycle consists of n edges such that there exists a cyclic ordering of the vertices where the edges correspond to consecutive segments of r vertices. We provide a first deterministic polynomial time algorithm, which finds a.a.s. tight Hamilton cycles in random r-uniform hypergraphs with edge probability at least $C \log^3 n/n$.

Our result partially answers a question of Dudek and Frieze [Random Structures & Algorithms 42 (2013), 374–385] who proved that tight Hamilton cycles exists already for $p = \omega(1/n)$ for $r = 3$ and $p = (e + o(1))/n$ for $r \geq 4$ using a second moment argument. Moreover our algorithm is superior to previous results of Allen, Böttcher, Kohayakawa and Person [Random Structures & Algorithms 46 (2015), 446–465] and Nenadov and Škorić [arXiv:1601.04034] in various ways: the algorithm of Allen et al. is a randomised polynomial time algorithm working for edge probabilities $p \geq n^{-1+\varepsilon}$, while the algorithm of Nenadov and Škorić is a randomised quasipolynomial time algorithm working for edge probabilities $p \geq C \log^8 n/n$.

1. Introduction

The Hamilton Cycle Problem, i.e., deciding whether a given graph contains a Hamilton cycle, is one of the 21 classical NP-complete problems due to Karp [13]. The best currently known algorithm is due to Björklund [3]: a Monte-Carlo algorithm with worst case running time $\mathcal{O}^*(1.657^n)$,[1] without false positives and false negatives occurring only with exponentially small probability. But what about "typical" instances? In other words, when the input is a random graph sampled from some specific distribution, is there an algorithm which finds a Hamilton cycle in polynomial time with small error probabilities?

For example, let us examine the classical binomial random graph $\mathcal{G}(n, p)$: Pósa [22] and Korshunov [15, 16] proved that the hamiltonicity threshold is at $p = \Theta(\log n/n)$. Their result was improved by Komlós and Szemerédi [14] who showed that the hamiltonicity threshold coincides with the threshold for minimum degree 2, and Bollobás [4] demonstrated that this is even true for the hitting times of these two properties in the corresponding random graph process. But these results do not allow one to actually find any Hamilton cycle in polynomial time. The first polynomial time randomised algorithms for finding Hamilton cycles in $\mathcal{G}(n, p)$ are due to Angluin and Valiant [2] and Shamir [25]. Subsequently, Bollobás, Fenner and Frieze [5] developed a deterministic algorithm, whose success probability (for input sampled from $\mathcal{G}(n, p)$) matches the probability of $\mathcal{G}(n, p)$ being hamiltonian in the limit as $n \to \infty$.

Date: May 16, 2018.

* Department of Mathematics, London School of Economics, Houghton Street, London WC2A 2AE, U. K. *E-mail*: p.d.allen@lse.ac.uk .

** Department of Statistics, University of Oxford, St. Giles 24-29, Oxford OX1 3LB, U. K. *E-mail*: christoph.koch@stats.ox.ac.uk .

†Institut für Mathematik, Goethe Universität, Robert-Mayer-Str. 6-10 60325 Frankfurt am Main, Germany. *E-mail*: parczyk@math.uni-frankfurt.de .

‡Institut für Mathematik, Technische Universität Ilmenau, 98684 Ilmenau, Germany. *E-mail*: yury.person@tu-ilmenau.de .

The second author was supported by Austrian Science Fund (FWF): P26826, and European Research Council (ERC): No. 639046. The third and fourth authors were supported by DFG grant PE 2299/1-1.

[1]Writing $\mathcal{O}^*$ means we ignore polylogarithmic factors.

Turning to hypergraphs, there exist various notions of Hamilton cycles: weak Hamilton cycle, Berge Hamilton cycle, ℓ-overlapping Hamilton cycles (for $\ell \in [r-1]$). In each situation, one seeks to cyclically order the vertex set such that:

- any two consecutive vertices lie in a hyperedge (a *weak Hamilton cycle*),
- any two consecutive vertices lie in some chosen hyperedge and no hyperedge is chosen twice (a *Berge Hamilton cycle*),
- the edges are consecutive segments so that two consecutive edges intersect in exactly ℓ vertices (an *ℓ-overlapping Hamilton cycle*).

The (binomial) random r-uniform hypergraph $\mathcal{G}^{(r)}(n,p)$ defined on the vertex set $[n] := \{1,\dots,n\}$, includes each *r-set* $x \in \binom{[n]}{r}$ as an *(hyper-)edge* independently with probability $p = p(n)$. The study of Hamilton cycles in random hypergraphs was initiated more recently by Frieze in [10], who considered so-called loose cycles in 3-uniform hypergraphs (these are 1-overlapping cycles in our terminology). Dudek and Frieze [7, 8] determined, for all ℓ and r, the threshold for the appearance of an ℓ-overlapping Hamilton cycle in a random r-uniform hypergraph (most thresholds being determined exactly, some only asymptotically). However, these results were highly nonconstructive, relying either on a result of Johansson, Kahn and Vu [12] or the second moment method.

The case of weak Hamilton cycles was studied by Poole in [21], while Berge Hamilton cycles in random hypergraphs were studied by Clemens, Ehrenmüller and Person in [6], the latter one being algorithmic.

In the case $\ell = r-1$ it is customary to refer to an ℓ-overlapping cycle as a *tight* cycle. Thus, the tight r-uniform cycle on vertex set $[n]$, $n \geq r$, has edges $\{i+1, ..., i+r\}$ for all i, where we identify vertex $n+i$ with i. A general result of Friedgut [9] readily shows that the threshold for the appearance of an ℓ-overlapping cycle in $\mathcal{G}^{(r)}(n,p)$ is sharp; that is, there is some threshold function $p_0 = p_0(n)$ such that for any constant $\varepsilon > 0$ the following holds. If $p \leq (1-\varepsilon)p_0$ then $\mathcal{G}^{(r)}(n,p)$ a.a.s. does not contain the desired cycle, whereas if $p \geq (1+\varepsilon)p_0$ then it a.a.s. does contain the desired cycle. Dudek and Frieze [8] proved that for $r \geq 4$ the function $p_0(n) = e/n$ is a threshold function for containment of a tight cycle, while for $r = 3$ they showed that a.a.s. $\mathcal{G}^{(3)}(n,p)$ contains a tight Hamilton cycle for any $p = p(n) = \omega(1/n)$. An easy first moment calculation shows that if $p = p(n) \leq (1-\varepsilon)e/n$ then a.a.s. $\mathcal{G}^{(r)}(n,p)$ does not contain a tight Hamilton cycle.

1.1. **Main result.** At the end of [8], Dudek and Frieze posed the question of finding algorithmically various ℓ-overlapping Hamilton cycles at the respective thresholds. In this paper we study tight Hamilton cycles and provide a first deterministic polynomial time algorithm, which works for p only slightly above the threshold.

Theorem 1. *For each integer $r \geq 3$ there exists $C > 0$ and a deterministic polynomial time algorithm with runtime $O(n^r)$ which for any $p \geq C(\log n)^3 n^{-1}$ a.a.s. finds a tight Hamilton cycle in the random r-uniform hypergraph $\mathcal{G}^{(r)}(n,p)$.*

Prior to our work there were two algorithms known that dealt with finding tight cycles. The first algorithmic proof was given by Böttcher, Kohayakawa and the first and the fourth authors in [1], where they presented a randomised polynomial time algorithm which could find tight cycles a.a.s. at the edge probability $p \geq n^{-1+\varepsilon}$ for any fixed $\varepsilon \in (0, 1/6r)$ and running time n^{20/ε^2}. The second result is a randomised quasipolynomial time algorithm of Nenadov and Škorić [20], which works for $p \geq C(\log n)^8/n$.

Our result builds on the adaptation of the absorbing technique of Rödl, Ruciński and Szemerédi [24] to sparse random (hyper-)graphs. This technique was actually used earlier by Krivelevich in [17] in the context of random graphs. However, the first results that provided essentially optimal thresholds (for other problems) are proved in [1] mentioned above in the context of random hypergraphs and independently by Kühn and Osthus in [18], who studied the threshold for the appearance of powers of Hamilton cycles in random graphs. The probability of $p \geq C(\log n)^3 n^{-1}$ results in the use of so-called reservoir structures of polylogarithmic size, as first used by Montgomery to find spanning trees in random graphs [19], and later in [20]. Our improvements result in the combination of the two algorithmic approaches [1, 20] and in the analysis of a simpler algorithm that we provide.

Organisation. In Section 2 we provide an informal overview of our algorithm. In Section 3 we then provide two key lemmas and the proof of Theorem 1 which rests on these lemmas. In the subsequent sections we prove these main lemmas: the Connecting Lemma and the Reservoir Lemma.

2. An informal algorithm overview

2.1. **Notation and inequalities.** An *s-tuple* $(u_1, \dots, u_s)$ of vertices is an ordered set of distinct vertices. We often denote tuples by bold symbols, and occasionally also omit the brackets and write $\mathbf{u} = u_1, \dots, u_s$. Additionally, we may also use a tuple as a set and write for example, if S is a set, $S \cup \mathbf{u} := S \cup \{u_i : i \in [s]\}$. The *reverse* of the s-tuple $\mathbf{u}$ is the s-tuple $\overleftarrow{\mathbf{s}} := (u_s, \dots, u_1)$.

In an r-uniform hypergraph $\mathcal{G}$ the tuple $P = (u_1, \dots, u_\ell)$ forms a *tight path* if the set $\{u_{i+1}, \dots, u_{i+r}\}$ is an edge for every $0 \le i \le \ell - r$. For any $s \in [\ell]$ we say that P *starts* with the s-tuple $(u_1, \dots, u_s) =: \mathbf{v}$ and *ends* with the s-tuple $(u_{\ell-(s-1)}, \dots, u_\ell) =: \mathbf{w}$. We also call $\mathbf{v}$ the *start s-tuple* of P, $\mathbf{w}$ the *end s-tuple* of P, and P a $\mathbf{v} - \mathbf{w}$ path. The *interior* of P is formed by all its vertices but its start and end $(r-1)$-tuples. Note that the interior of P is not empty if and only if $\ell > 2(r-1)$.

For a binomially distributed random variable X and a constant $0 < \gamma < 1$ we will apply the following Chernoff-type bound (see, e.g., [11, Corollary 2.3])

$$\mathbb{P}\left[|X - \mathbb{E}(X)| \le \gamma \mathbb{E}(X)\right] \le 2\exp\left(-\frac{\gamma^2 \mathbb{E}(X)}{3}\right). \tag{1}$$

In addition we will make use of the following consequence of Janson's inequality (see for example [11], Theorem 2.18): Let Ω be a finite set and $\mathcal{P}$ be a family of non-empty subsets of Ω. Now consider the random experiment where each $e \in \Omega$ is chosen independently with probability p and define for each $P \in \mathcal{P}$ the indicator variable I_P that each element of P gets chosen. Set $X = \sum_{P \in \mathcal{P}} I_P$ and $\Delta = \sum_{P \ne P', P \cap P' \ne \emptyset} \mathbb{E}(I_P I_{P'})$. Then

$$\mathbb{P}[X = 0] \le \exp\left(-\frac{\mathbb{E}(X)^2}{\mathbb{E}(X) + \Delta}\right). \tag{2}$$

2.2. **Overview of the algorithm.** We start with the given sample of the random hypergraph $\mathcal{G}^{(r)}(n,p)$ and we will reveal the edges as we proceed. First, using the Reservoir Lemma (Lemma 2 below), we construct a tight path P_{res} which covers a small but bounded away from zero fraction of $[n]$, which has the *reservoir property*, namely that there is a set $R \subseteq V(P_{\text{res}})$ of size $2Cp^{-1}\log n \le 2n/\log^2 n$ such that for any $R' \subseteq R$, there is a tight path covering exactly the vertices $V(P_{\text{res}}) \setminus R'$ whose ends are the same as those of P_{res}, and this tight path can be found given P_{res} and R' in time polynomial in n a.a.s.

We now greedily extend P_{res}, choosing new vertices when possible and otherwise vertices in R. We claim that a.a.s. this strategy produces a structure P_{almost} which is almost a tight path extending P_{res} and covering $[n]$. The reason it is only 'almost' a tight path is that some vertices in R may be used twice. We denote the set of vertices used twice by R'_1. But we will succeed in covering $[n]$ with high probability. Recall that, due to the reservoir property, we can dispense with the vertices from R'_1 in the part P_{res} of the almost tight Hamilton path P_{almost}.

Finally, we apply the Connecting Lemma (Lemma 3 below) to find a tight path in $R \setminus R'_1$ joining the ends of P_{almost}, and using the reservoir property this gives the desired tight Hamilton cycle.

This approach is similar to that in [1]. The main difference is the way we prove the Reservoir Lemma (Lemma 2). In both [1] and this paper, we first construct many small, identical, vertex-disjoint *reservoir structures* (in some part of the literature, mostly in the dense case, this structure is called an absorber). A reservoir structure contains a spanning tight path, and a second tight path with the same ends which omits one *reservoir vertex*. We then use Lemma 3 to join the ends of all these reservoir structures together into the desired P_{res}. In [1], reservoir structures are of constant size (depending on the ε) and they are found by using brute-force search. This is slow, and is also the cause of the algorithm in [1] being randomised: there it is necessary to simulate exposure in rounds of the random hypergraph since the brute-force search reveals all edges. In this paper, by contrast, we construct reservoir structures by a local search procedure which is both much faster and reveals much less of the random hypergraph.

We will perform all the constructions in this paper by using local search procedures. At each step we reveal all the edges of $\mathcal{G}^{(r)}(n,p)$ which include a specified $(r-1)$-set, the *search base*. The number of such edges will always be in expectation of the order of pn, so that by Chernoff's inequality and the union bound, with high probability at every step in the algorithm the number of revealed edges is close to the expected number. Of course, what we may not do is attempt to reveal a given edge twice: we therefore keep track of an *exposure hypergraph* $\mathcal{E}$, which is the $(r-1)$-uniform hypergraph consisting of all the $(r-1)$-sets which have been used as search bases up to a given time in the algorithm. We will show that $\mathcal{E}$ remains quite sparse, which means that at each step we have almost as much freedom as at the start when no edges are exposed.

For concreteness, we use a doubly-linked list of vertices as the data structure representing a tight (almost-) path. However this choice of data structure is not critical to the paper and we will not further comment on it. The reader can easily verify that the various operations we describe can be implemented in the claimed time using this data structure. To simplify readability, we will omit in the calculations floor and ceiling signs whenever they are not crucial for the arguments.

3. Two key Lemmas and the proof of Theorem 1

3.1. **Two Key Lemmas.** Recall the definition of the *reservoir path* P_{res}. It is an r-uniform hypergraph with a special subset $R \subsetneq V(P_{\text{res}})$ and some start and end $(r-1)$-tuples $\mathbf{v}$ and $\mathbf{w}$ respectively, such that:

(1) P_{res} contains a tight path with the vertex set $V(P_{\text{res}})$ and the 'end tuples' $\mathbf{v}$ and $\mathbf{w}$, and
(2) for *any* $R' \subseteq R$, P_{res} contains a tight path with the vertex set $V(P_{\text{res}}) \setminus R'$ and the 'end tuples' $\mathbf{v}$ and $\mathbf{w}$.

We first give the lemma which constructs P_{res}. In addition to with high probability returning P_{res}, we also need to describe the likely resulting exposure hypergraph.

Lemma 2 (Reservoir Lemma). *For each $r \geq 3$ and $p \in (0,1]$ there exists $C > 0$ and a deterministic $O(n^r)$-time algorithm whose input is an n-vertex r-uniform hypergraph G and whose output is either 'Fail' or a reservoir path P_{res} with ends $\mathbf{u}$ and $\mathbf{v}$ and an $(r-1)$-uniform exposure hypergraph $\mathcal{E}$ on vertex set $V(G)$ with the following properties.*

(*i*) *All vertices of P_{res} and edges of $\mathcal{E}$ are contained in a set S of size at most $\frac{n}{4}$.*
(*ii*) *The reservoir $R \subseteq V(P_{\text{res}})$ has size $2Cp^{-1}\log n$.*
(*iii*) *There are no edges of $\mathcal{E}$ contained in $R \cup \mathbf{u} \cup \mathbf{v}$.*
(*iv*) *All r-sets in $V(G)$ which have been exposed contain at least one edge of $\mathcal{E}$.*

When G is drawn from the distribution $\mathcal{G}^{(r)}(n,p)$ and $p \geq Cn^{-1}\log^3 n$, the algorithm returns 'Fail' with probability at most n^{-2}.

Furthermore we need a lemma which allows us to connect two given tuples with a not too long path. This lemma is the engine behind the proof and behind the Reservoir Lemma.

Lemma 3 (Connecting Lemma). *For each $r \geq 3$ there exist $c, C > 0$ and a deterministic $O(n^{r-1})$-time algorithm whose input is an n-vertex r-uniform hypergraph G, a pair of distinct $(r-1)$-tuples $\mathbf{u}$ and $\mathbf{v}$, a set $S \subseteq V(G)$ and an $(r-1)$-uniform exposure hypergraph $\mathcal{E}$ on the same vertex set $V(G)$. The output of the algorithm is either 'Fail' or a tight path of length*

$o(\log n)^2$ in G whose ends are $\mathbf{u}$ and $\mathbf{v}$ and whose interior vertices are in S, and an exposure hypergraph $\mathcal{E}' \supset \mathcal{E}$. We have that all the edges $E(\mathcal{E}') \setminus E(\mathcal{E})$ are contained in $S \cup \mathbf{u} \cup \mathbf{v}$.

Suppose that G is drawn from the distribution $\mathcal{G}^{(r)}(n,p)$ with $p \geq C(\log n)^3/n$, that $\mathcal{E}$ does not contain any edges intersecting both S and $\mathbf{u} \cup \mathbf{v}$. If furthermore $|S| = Cp^{-1}\log n$ and $|e(\mathcal{E}[S])| \leq c|S|^{r-1}$ then $e(\mathcal{E}') \leq e(\mathcal{E}) + O(|S|^{r-2})$ and the algorithm returns 'Fail' with probability at most n^{-5}.

3.2. **Overview continued: more details.** We now describe the algorithm claimed by Theorem 1, which we state in a high-level overview as Algorithm 1 and explain somewhat informally some of the arguments.

Algorithm 1: Find a tight Hamilton cycle in $\mathcal{G}^{(r)}(n,p)$

1 use subroutine from Lemma 2 to either construct P_{res} (with ends $\mathbf{u}$, $\mathbf{v}$ and exposure hypergraph $\mathcal{E}$ on S) or halt with **failure**;
$L := V(G) \setminus S$;
$U := S \setminus V(P_{\text{res}})$;

2 extend P_{res} greedily from v to cover all vertices of U and using up to $n/2$ vertices of L, otherwise halt with **failure**;

3 extend P_{res} further greedily to P_{almost} by covering all vertices of L and using up to $|R|/2$ vertices of R, otherwise halt with **failure**;

4 use subroutine of Lemma 3 to connect the ends of P_{almost} using the unused at least $|R|/2$ vertices of R, otherwise halt with **failure**;

Step 1. Given G drawn from the distribution $\mathcal{G}^{(r)}(n,p)$, we begin by applying Lemma 2 to a.a.s. find a reservoir path P_{res} with ends $\mathbf{u}$ and $\mathbf{v}$ contained in a set S of size $\frac{n}{4}$. Let $L = V(G) \setminus S$, and $U = S \setminus V(P_{\text{res}})$. Recall that by Lemma 2 (i) and (iii), all edges of $\mathcal{E}$ are contained in S; and $R \cup \mathbf{u} \cup \mathbf{v}$ contains no edges of $\mathcal{E}$. By (iv) all exposed r-sets contain an edge of $\mathcal{E}$; by choosing a little carefully where to expose edges (see Step 2 below), we will not need to worry about what exactly the edges of $\mathcal{E}$ are beyond the above information.

Step 2. We extend $P_{\text{res}} := P_0$ greedily, one vertex at a time, from its end $\mathbf{u} = \mathbf{u}_0$, to cover all of U. At each step i, we simply expose the edges of G which contain the end $\mathbf{u}_{i-1}$ of P_{i-1} and whose other vertex is not in $V(P_{i-1})$, choose one of these edges e and add the vertex from $e \setminus \mathbf{u}_{i-1}$ to P_{i-1} to form P_i. The rule we use for choosing e is the following: if i is congruent to 1 or 2 modulo 3, we choose e such that $e \setminus \mathbf{u}_{i-1}$ is in L, and if i is congruent to 0 modulo 3 we choose e such that $e \setminus \mathbf{u}_{i-1}$ is in U if it is possible; if not we choose e such that $x_i := e \setminus \mathbf{u}_{i-1}$ is in L. The point of this rule is that at each step we want to choose an edge which contains at least two vertices of L, because no such r-set can contain an edge of $\mathcal{E}$ since all the edges of $\mathcal{E}$ are contained in S (Property (i)). We will see that while $U \setminus V(P_{i-1})$ is large, we always succeed in choosing a vertex in U when i is congruent to 0 modulo 3. When it becomes small we do not, but a.a.s. we succeed often enough to cover all of U while using not more than $\frac{5n}{8}$ vertices of L.

Step 3. Next, we continue the greedy extension, this time choosing a vertex in L when possible and in R when not, until we cover all of L. It follows from the first two steps and Properties (i) and (iii) that no edge of $\mathcal{E}$ is in $L \cup R$. Thus, at each step we choose from newly exposed edges and again we a.a.s. succeed in covering L using only a few vertices of R. Let the final almost-path (which uses some vertices $R_1' \subseteq R$ twice) be P_{almost}, and R_1 the subset of R consisting of vertices we did not use in the greedy extension, i.e. $R_1 = R \setminus R_1'$.

[2]We will make this more precise later. You could replace this by at most $Cn/\log\log n$.

Step 4. At last, P_{almost} covers $V(G) = L \cup U \cup V(P_{\text{res}})$. Its ends, together with the vertices of R_1, satisfy the conditions of Lemma 3, which we apply to a.a.s. complete P_{almost} to an almost-tight cycle H' in which some vertices of R_1 are used twice. The reservoir property of R now gives a tight Hamilton cycle H.

Runtime. Our applications of Lemmas 2 and 3 take time polynomial in n by the statements of those lemmas; the greedy extension procedure is trivially possible in $O(n^2)$ time (since at each extension step we just need to look at the neighbourhood of an $(r-1)$-tuple, and there are $O(n)$ steps). Finally the construction of P_{res} allows us to obtain H from H' in time $O(n^2)$: we scan through P_{res}, for each vertex r of R we scan the remainder of H' to see if it appears a second time, and if so locally reorder $V(P_{\text{res}})$ to remove r from P_{res}.

To prove Theorem 1, what remains is to justify our claims that various procedures above a.a.s. succeed.

3.3. **Proof of Theorem 1.** We choose $C \geq \max\{C_{L_2}, C_{L_3}, 10^8\}$ large enough for Lemmas 2 and 3 to hold. For this proof we do not need to know the value of c' required for Lemma 3. We suppose that n is large enough to make $\log \log n$ larger than any constant appearing in the following proof.

Constructing P_{res}. Let G be drawn from the distribution $\mathcal{G}^{(r)}(n,p)$. Lemma 2 states that with probability at least $1 - n^{-2}$, a reservoir path P_{res} in G is found in polynomial time. From this point on, at each step except the final connection, when we expose edges at an $(r-1)$-set $\mathbf{x}$, that $(r-1)$-set will be included in the path we construct. Hence in future steps we will not examine edges containing $\mathbf{x}$. Thus while we should keep updating $\mathcal{E}$, in fact we will never need to know which edges are added after generating P_{res}.

Extending P_{res} *to cover all of* U. We next aim to prove that with high probability the greedy extension of P_{res} to cover U succeeds, with at least $n/8$ vertices of L remaining uncovered at the end. Recall that we chose $|S| = \frac{n}{4}$ and thus $|L| = \frac{3n}{4}$. We choose the next vertex from L when i is congruent to 1 or 2 modulo 3 or when we fail to extend into U. At each step i where at least $n/8$ vertices of L are uncovered, we expose all the r-sets in $V(G)$ which contain the end u_{i-1} of P_{i-1} and a vertex of L. The greedy algorithm can only fail to complete step i if none of these r-sets turn out to be edges, which happens with probability at most $(1-p)^{n/8} \leq \exp\big(-\frac{pn}{8}\big) < n^{-4}$ (since the edges of the random hypergraph are independent). Taking the union bound, the greedy algorithm to cover U fails before covering $\frac{5}{8}n$ vertices of L with probability at most n^{-3}.

Similarly, for any i such that $\big|U \setminus V(P_{i-1})\big| \geq Cp^{-1}\log n$, if i is divisible by 3 the probability that no edge containing u_{i-1} and a vertex of $U \setminus V(P_{i-1})$ is in G is at most $\exp\big(-C\log n\big) < n^{-4}$. It follows that with probability at most n^{-3} the greedy algorithm chooses a vertex of L when i is divisible by 3 and $U \setminus V(P_{i-1})$ has size at least $Cp^{-1}\log n$. Let t_1 be the first time in the greedy extension procedure when $U \setminus V(P_{t_1})$ has size less than $Cp^{-1}\log n$.

It remains to show that while the last $Cp^{-1}\log n$ vertices of U are covered, at most $n/8$ vertices of L are used. We split these last $Cp^{-1}\log n$ vertices into the last $\frac{1}{2}p^{-1}$ vertices and the rest. When x vertices of U remain uncovered with $x \geq \frac{1}{2}p^{-1}$, then the probability of choosing a vertex of U for the vertex x_i extending P_{i-1} (when i is divisible by 3) is at least $1-(1-p)^x \geq \frac{1}{3}$. By Chernoff's inequality, the probability that at time $t_2 := t_1 + 6Cp^{-1}\log n$ there are more than $\frac{1}{2}p^{-1}$ vertices of U remaining uncovered is at most $\exp\big(-\frac{1}{6}Cp^{-1}\log n\big) \leq n^{-3}$. Next, we show that we cover all but at most $\log n$ vertices of U in not too much more time.

To see this, consider the following event. For $1 \leq j \leq 7n/8$ and $\log n \leq x \leq \frac{1}{2}p^{-1}$, let $A(x,j)$ be the event that we have $\big|U \setminus V(P_j)\big| = x$ and $\big|U \setminus V(P_{j-3000p^{-1}})\big| \leq 2x$. We claim that the probability for any of these events to hold is at most n^{-3}. Indeed, if for some given x and j the event $A(x,j)$ occurs, then at each of the at least $500p^{-1}$ values of i with $j - 3000p^{-1} \leq i \leq j$, an edge containing $\mathbf{u}_{i-1}$ and a vertex of U appears with probability at least $1-(1-p)^x \geq px/2$ (since $x \leq \frac{1}{2}p^{-1}$). Thus for $A(x,j)$ to hold, it is necessary that a sum of at least $500p^{-1}$ Bernoulli random variables, each with probability at least $px/2$, is at most x. Chernoff's inequality states that this probability is at most $\exp\big(-\frac{250x}{12}\big) \le n^{-5}$, and taking the union bound over all $A(x,j)$ the claim follows. Taking in particular $x = 2^{-k}n/\log n$ for $k \ge 1$ such that $2^{-k}n\log n \ge \log n$ (so $k \le \log n$) we see that with probability at least $1 - n^{-3}$, at time $t_3 := t_2 + 3000p^{-1}\log n$ there are at most $\log n$ vertices of U remaining uncovered.

While at least one vertex of U remains uncovered, the probability that when i is divisible by three we choose a vertex of U is at least p. Applying Chernoff's inequality, the probability that at time $t_4 := t_3 + 300p^{-1}\log n$ we still have not covered all of U is at most $\exp(-\frac{100\log n}{12}) \le n^{-3}$. Putting all this together, the probability that $V(P_{t_4})$ does not cover U is at most $4n^{-3}$. Since $t_1 \le 3|U|$, since $|U| \le |S| \le n/4$, and since $t_4 - t_1 \le n/16$, we conclude that with probability at least $1 - 4n^{-3}$ the greedy extension procedure indeed covers U with at least $n/8$ vertices of L left uncovered. Let t_5 be the first time at which P_{t_5} covers U.

Extending P_{res} further to P_{almost} by covering all of L. We now repeat a similar procedure to use up all of $L \setminus V(P_{t_5})$ while not using too many vertices in R. Since no edges of $\mathcal{E}$ are contained in $R \cup L$, at each time t, all the r-sets containing the end $\mathbf{u}_{t-1}$ of P_{t-1} and a vertex of $L \cup R \setminus V(P_{t-1})$ are unrevealed. In particular, provided that at each step we have $\big|R \setminus V(P_{t-1})\big| \ge \frac{1}{2}|R|$, by Chernoff's inequality with probability at least $1 - n^{-4}$ at least one edge of G is found consisting of $\mathbf{u}_{t-1}$ and a vertex of $R \setminus V(P_{t-1})$. Taking the union bound, the probability of the extension procedure failing when $\big|R \setminus V(P_{t-1})\big| \ge \frac{1}{2}|R|$ is at most n^{-3}.

As long as $\big|L \setminus V(P_{t-1})\big| \ge \frac{C}{100}p^{-1}\log n$, by Chernoff's inequality with probability at most $\exp\big(-\frac{C}{300}\log n\big) \le n^{-4}$ there is no edge of G containing $\mathbf{u}_{t-1}$ and a vertex of $L \setminus V(P_{t-1})$; in particular with probability at least $1 - n^{-3}$ the greedy extension covers all but at most $\frac{C}{100}p^{-1}\log n$ vertices of L before using any vertex of R. Let t_6 be the time at which all but at most $\frac{C}{100}p^{-1}\log n$ vertices of L are covered. Again, we now consider the time taken to cover all but $\frac{1}{2}p^{-1}$ vertices of L. At each time the probability of being able to choose a vertex of L to extend our path with is at least $\frac{1}{3}$, so that with probability at least $1 - n^3$ we cover all but at most $\frac{1}{2}p^{-1}$ vertices of L by time $t_7 \le t_6 + \frac{C}{25}p^{-1}\log n$. In particular we use at most $\frac{C}{25}p^{-1}\log n$ vertices of R in this time.

By the same analysis as before, the total time taken to go from covering all but at most $\frac{1}{2}p^{-1}$ vertices of L to covering all but at most $\log n$ vertices of L and then all vertices of L is with probability at least $1 - 2n^{-3}$ not more than $3000p^{-1}\log n + 300p^{-1}\log n$. Putting this together, provided all these good events hold we succeed in covering all but at most $\log n$ vertices of L having used at most

$$\tfrac{C}{25}p^{-1}\log n + 3300p^{-1}\log n < Cp^{-1}\log n = \tfrac{1}{2}|R|$$

vertices of R.

In sum, with probability at least $1 - n^{-2} - 8n^{-3}$, the algorithm succeeds in generating P_{almost}, where the set $R' \subseteq R$ of vertices not used in the greedy extension has size at least $\frac{1}{2}|R|$.

Connecting the end tuples of P_{almost} and getting the tight Hamilton cycle. Applying Lemma 3 to connect the end tuples of P_{almost} in a subset of R' of size $Cp^{-1}\log n$ (which is possible since R' together with the ends of P_{almost} contains no edges of $\mathcal{E}$ and since $|R'| \ge n/\log^2 n$), with probability at least $1 - n^{-4}$ we find the desired almost-tight cycle H', which gives us deterministically the desired tight Hamilton cycle H. Thus as desired the probability that our algorithm fails to find a tight Hamilton cycle is at most n^{-1}. □

4. Proof of the Connecting Lemma

In this section we prove Lemma 3 and a very similar lemma (Lemma 6) dealing with 'spike-paths' which we will require for Lemma 2. A spike-path is similar to a tight path, but after $(r-1)$-steps the direction of the last $(r-1)$-tuple is inverted.

Definition 4 (Spike path). *In an r-uniform hypergraph, a spike path of length t consists of a sequence of t pairwise disjoint $(r-1)$-tuples $\mathbf{a}_1, \dots, \mathbf{a}_t$, where $\mathbf{a}_i = (a_{i,1}, \dots, a_{i,r-1})$ for all i, with* *the property, that the edges* $\{a_{i,r-j},\ldots,a_{i,1},a_{i+1,1},\ldots,a_{i+1,j}\}$ *are present for all* $i=1,\ldots,t-1$ *and* $j=1,\ldots,r-1$. *We call* $\mathbf{a}_i$ *the* i*-th spike.*

This is the same as taking t tight paths of length $2(r-1)$, where the end $(r-1)$-tuples of path i are $\mathbf{x}_i$ and $\mathbf{y}_i$, and identifying $\overleftarrow{\mathbf{x}_i}$ with $\mathbf{y}_{i+1}$ for all $i=1,\ldots,t-1$. The proofs of Lemmas 3 and 6 are essentially identical, so we give the details of the former and then explain how to modify it to obtain the latter.

4.1. **Preliminaries.** For an $(r-1)$-tuple $\mathbf{u}$ and an integer i we define a *fan* $\mathcal{F}_i(\mathbf{u})$ in an r-uniform hypergraph $\mathcal{H}$ as a set $\{P_1,\ldots,P_s\}$ of tight paths in $\mathcal{H}$, of length i or $i+1$, starting in $\mathbf{u}$. For any set or tuple $\mathbf{a}$, let $\{P_j\}_{j\in I}$ be the subcollection of tight paths from $\mathcal{F}_i(\mathbf{u})$ in which $\mathbf{a}$ appears as a consecutive interval (in arbitrary order). The *leaves* or *ends* of $\mathcal{F}_i(\mathbf{u})$ are the ending $(r-1)$-tuples of alle the paths $P_1,\ldots,P_s$. We denote by $\mathrm{mult}(\mathbf{a})$ the number of different paths we see in $\{P_j\}_{j\in I}$ after truncating behind $\mathbf{a}$.

In any r-uniform hypergraph $H=(V,E)$ the degree of a set or tuple f of size $1\le|f|\le r-1$ is the number of edges which it is contained in, i.e.

$$\deg_H(f)=|\{e\in E: f\subseteq e\}|.$$

Given a set $S\subseteq V$, we write $\deg_H(f,S)$ for the degree into S, that is, where we count only edges e satisfying $e\setminus f\subseteq S$.

4.2. **Idea and further notation.** The basic idea is that, starting with the $\mathbf{u}$ and $\mathbf{v}$ and the empty fans $\mathcal{F}_0(\mathbf{u})$ and $\mathcal{F}_0(\mathbf{v})$, we want to fan out. That is, for each path in $\mathcal{F}_i(\mathbf{u})$ we will find a large collection of ways to extend by one vertex and all the resulting paths form $\mathcal{F}_{i+1}(\mathbf{u})$. We do this until we have fans $\mathcal{F}_t(\mathbf{u})$ and $\mathcal{F}_t(\mathbf{v})$ with

$$Q:=p^{-(r-1)/2}\log n$$

leaves each. This happens roughly when we have

$$t:=2\cdot\left\lceil\frac{\log(Q)}{\log(\log n)}\right\rceil\le(r-1)\cdot\left\lceil\frac{\log(p^{-1})}{\log(\log n)}\right\rceil+2=o(\log n).$$

A complication is that in this process we have to avoid the edges of $\mathcal{E}$ when expanding the fans. In order to make the modifications for the promised spike-path variation easy (cf. Lemma 6 below), we will do something a little more complicated. We split into expansion and continuation phases, each of length $r-1$. The first phase is an expansion phase, so when forming $\mathcal{F}_1(\mathbf{u})$, …, $\mathcal{F}_{r-1}(\mathbf{u})$ we find many ways to extend each path by one vertex and put all of them into the next fan. The second phase is a continuation phase, so when forming $\mathcal{F}_r(\mathbf{u})$, …, $\mathcal{F}_{2r-2}(\mathbf{u})$ we choose only one way to extend each path. As soon as we have a collection of paths with the desired Q leaves, we cease expanding (even if we are still in an expansion phase) and simply continue each path such that each has the same length. We construct fans from $\mathbf{v}$ similarly, and we continue construction up to $\mathcal{F}_t(\mathbf{v})$.

In the final step we find $r-1$ further edges connecting two of the leaves, giving us a tight path connecting $\mathbf{u}$ to $\mathbf{v}$. Again there is a complication here: some pairs of leaves $(\mathbf{w},\mathbf{x})$ may be *blocked* by edges of $\mathcal{E}$, meaning that inside some r consecutive vertices of the concatenation $\mathbf{w}\overleftarrow{\mathbf{x}}$ there is an edge of $\mathcal{E}$. If a pair of leaves is blocked, then trying to reveal $(r-1)$ edges connecting the pair would mean revealing an edge of the random hypergraph twice (and if a pair is not blocked then doing so does not reveal any edge twice). We need to take this into account in our analysis, and we need to construct $\mathcal{F}_t(\mathbf{v})$ carefully to avoid creating *dangerous* leaves for which a large fraction of the pairs is blocked.

To make this precise, we use the following algorithm.

The subroutine BuildFan takes as input a starting tuple, the sets in which to build a fan, and a *danger hypergraph* D which is important for the construction of the second fan: it is an $(r-1)$-uniform hypergraph which records the tuples in $S'_1,\ldots,S'_{4(r-1)}$ which we cannot easily connect to the leaves of $\mathcal{F}_t(\mathbf{u})$. The algorithm ensures that no leaf of a fan will be a dangerous tuple. Though we only need this for the leaves of the final fan, it is convenient to maintain this

Algorithm 2: Find a connecting path from $\mathbf{u}$ to $\mathbf{v}$

split S into equal parts $S_1, \ldots, S_{4(r-1)}, S'_1, \ldots, S'_{4(r-1)}$;
$\mathcal{F}_t(\mathbf{u}) :=$ BuildFan$(\mathbf{u}, S_1, \ldots, S_{4(r-1)}, \emptyset)$;
set $D := \{\mathbf{x} \in S^{r-1} : (\mathbf{w}, \mathbf{x})$ is blocked for at least $\xi' Q$ leaves $\mathbf{w}$ of $\mathcal{F}_t(\mathbf{u})\}$;
$\mathcal{F}_t(\mathbf{v}) :=$ BuildFan$(\mathbf{v}, S'_1, \ldots, S'_{4(r-1)}, D)$;
find $r-1$ edges connecting a leaf of $\mathcal{F}_t(\mathbf{u})$ to the reverse of one of $\mathcal{F}_t(\mathbf{v})$;
return tight path P connecting $\mathbf{u}$ to $\mathbf{v}$;

property throughout. For convenience, we write S_i for the set $S_{i \bmod 4(r-1)} \in \{S_1, \ldots, S_{4(r-1)}\}$ with $S_0 = S_{4(r-1)}$; the point of these sets is that we choose the ith vertex of each path in S_i, which is helpful in the analysis. Finally, we need to ensure that we always choose 'good' vertices which allow us to continue our construction and prove various probabilistic statements. To that end, we define a vertex b to be *good* with respect to an exposure hypergraph $\mathcal{E}$, a set $\mathcal{F}$ of paths with distinct ends, a danger hypergraph D and a $(r-1)$-tuple $\mathbf{a}$ if none of the following statements hold for any (possibly empty) tuple $\mathbf{c}$ whose vertices are contained in those of $\mathbf{a}$ (not necessarily in the same order).

(*i*) b appears somewhere on the unique path $P(\mathbf{a})$ ending in $\mathbf{a}$,
(*ii*) $|\mathbf{c}| \le r-2$ and $\deg_{\mathcal{E}}(\{\mathbf{c}, b\}, S) > \xi^{r-|\mathbf{c}|-1}|S|^{r-|\mathbf{c}|-2}$,
(*iii*) $\mathrm{mult}(\{\mathbf{c}, b\}) > \xi^{r-|\mathbf{c}|-1} Q \cdot |S|^{-|\mathbf{c}|-1} \cdot \log^{|\mathbf{c}|+1} n$, and
(*iv*) $|\mathbf{c}| \le r-2$ and $\deg_D(\{\mathbf{c}, b\}, S) > (\xi'|S|)^{r-|\mathbf{c}|-2}$.

Normally $\mathcal{E}$, $\mathcal{F}$ and D will be clear from the context and we will simply say good for $\mathbf{a}$. We are finally ready to give the BuildFan subroutine.

4.3. **Proof.** We set

$$\xi' = \frac{1}{100^r}\,, \quad \xi = (\xi')^r/(2r2^{20r})\,, \quad \delta = 8^r\xi + \xi'\,, \quad C = 10^{8r} \quad \text{and} \quad c = 10^{-r}\xi^r\,. \tag{3}$$

The proof amounts to showing two things. First, BuildFan is likely to succeed—that is, that it does not fail for lack of good vertices before returning a fan, that the returned fan does have size Q, and that it does not add too many tuples to $\mathcal{E}$. Second, the required extra $r-1$ edges which should connect the fans can be found.

Creating the fans. We begin by showing that the subroutine BuildFan$(\mathbf{s}, T_1, \ldots, T_{4(r-1)}, D)$ is likely to succeed, whether we choose $\mathbf{s} = \mathbf{u}$, $T_i = S_i$ and $D = \emptyset$ or we choose $\mathbf{s} = \mathbf{v}$, $T_i = S'_i$ and D as given in Algorithm 2, using the following claim.

We define L_i to be the leaves of $\mathcal{F}_i$.

Claim 5. *If step i was successful, then step $i+1$ is successful with probability at least $1 - n^{-3r}$ and the following holds throughout step $i+1$ for each $\mathbf{a} \in L_{i+1}$ and each non-empty $\mathbf{c}$ whose vertices are chosen from $\mathbf{a}$, not necessarily in the same order.*

P1 *Each path in $\mathcal{F}_i$ extends to at least one path in $\mathcal{F}_{i+1}$; if $2(r-1)\ell < i \le 2(r-1)\ell + r - 1$ and $|\mathcal{F}_{i+1}| < Q$ then each path in $\mathcal{F}_i$ extends to at least $\log n$ paths in $\mathcal{F}_{i+1}$. In both cases, all leaves are not in $\mathcal{E}$.*
P2 $e(\mathcal{E}[S]) \le c|S|^{r-1} + 20rQ$.
P3 *If $|\mathbf{c}| < r-1$ we have $\deg_{\mathcal{E}}(\mathbf{c}, S) \le \xi^{r-|\mathbf{c}|}|S|^{r-1-|\mathbf{c}|} + 1$.*
P4 *We have mult$(\mathbf{c}) \le \xi^{r-|\mathbf{c}|} Q \cdot |S|^{-|\mathbf{c}|} \cdot \log^{|\mathbf{c}|} n + 1$.*
P5 *If $1 \le |\mathbf{c}| \le r-2$ we have $\deg_D(\mathbf{c}, S) \le (\xi'|S|)^{r-|\mathbf{c}|-1}$.*

Proof of Claim 5. Observe that $\mathcal{F}_0$ trivially satisfies the conditions of Claim 5, modulo Chernoff's inequality for **P1**. Suppose that for some $0 \le i < t$, at each step $0 \le j \le i$ of Algorithm 3 the conditions of Claim 5 are satisfied. In particular, by **P4**, the ends of the paths $\mathcal{F}_i$ are distinct as for $|\mathbf{c}| = r-1$ we have $\mathrm{mult}(\mathbf{c}) < 2$, and by **P1** we have $|\mathcal{F}_i| \ge \min\left(\log^{i/2} n, Q\right)$.

Algorithm 3: BuildFan$(\mathbf{s}, T_1, \ldots, T_{4(r-1)}, D)$

$\mathcal{F}_0 := \{\mathbf{s}\}$;
foreach $i = 1, \ldots, t$ **do**
  if $i \mod 2(r-1) \in \{1, \ldots, r-1\}$ **then**
    phase =‘expand’;
  else
    phase =‘continue’;
  end
  NumPaths := $|\mathcal{F}_{i-1}|$;
  $\mathcal{F}_i := \mathcal{F}_{i-1}$;
  foreach $P \in \mathcal{F}_{i-1}$ **do**
    let the $(r-1)$-tuple $\mathbf{a}$ be the end of P ;
    reveal the edges of G containing $\mathbf{a}$ and add $\mathbf{a}$ to $\mathcal{E}$;
    let $T \subseteq T_i$ be the set of vertices b which are good for $\mathbf{a}$ and $\{\mathbf{a}, b\}$ is an edge;
    if phase =‘expand’ **then**
      Add := min $\big(\log n, Q + 1 - \text{NumPaths}\big)$;
      choose Add vertices $b_1, \ldots, b_{\text{Add}} \in T$;
      $\mathcal{F}_i := \mathcal{F}_i \cup \{(P, b_1), \ldots, (P, b_{\text{Add}})\} \setminus \{P\}$;
      NumPaths := NumPaths + Add − 1;
    else
      choose a vertex $b \in T$;
      $\mathcal{F}_i := \mathcal{F}_i \cup \{(P, b)\} \setminus \{P\}$;
    end
  end
end
return $\mathcal{F}_t$;

To begin with, we show that $\mathcal{E}$ cannot have too many edges. At each step j with $1 \le j \le i$, we add $|\mathcal{F}_{j-1}|$ edges to $\mathcal{E}$, so that we want to upper bound $\sum_{j=1}^{t} |\mathcal{F}_{j-1}|$. Definitely $\mathcal{F}_t$ has size at most Q and $\mathcal{F}_{j-4(r-1)}$ always has size less than half of $\mathcal{F}_j$, so that this sum is dominated by $4r \sum_{i=1}^{\ell} 2^i$ where $\ell = \log_2 Q$. We conclude that $\sum_{j=1}^{t} |\mathcal{F}_{j-1}| \le 8rQ$. Since we create two fans, in total we obtain the claimed bound **P2**.

We now show that, for each choice of $P \in \mathcal{F}_i$ with end $\mathbf{a}$, the total number of vertices in T_{i+1} which are not good for $\mathbf{a}$ is at most $\delta|S|$. This will allow us to prove **P1**. First, since P has at most t vertices, at most t vertices are excluded by (i).

For each $\mathbf{c}$ of size at most $r-2$ with vertices chosen from $\mathbf{a}$, there are at most $2r\xi|S|$ vertices fulfilling (ii). To see this for $|\mathbf{c}| = 0$, observe that otherwise we have $e(\mathcal{E}[S]) > 2\xi^r |S|^{r-1} > 2c|S|^{r-1}$, contradicting **P2** as $Q \le \frac{1}{C}|S|^{r-1}$. Assume that it fails for some non-empty $\mathbf{c}$. Then there are more than $2r\xi|S|$ vertices $x \in T_{i+1}$ with

$$\deg_{\mathcal{E}}(\{\mathbf{c}, x\}, S) > \xi^{r-|\mathbf{c}|-1} |S|^{r-|\mathbf{c}|-2}$$

which implies that

$$\deg_{\mathcal{E}}(\mathbf{c}, S) > 2\xi^{r-|\mathbf{c}|} |S|^{r-|\mathbf{c}|-1}$$

in contradiction to **P3**.

Furthermore there are at most $2r\xi|S|$ vertices b fulfilling (iii) for each $\mathbf{c}$. Again for $|\mathbf{c}| = 0$ it is enough to note that there are at most Q paths in total and thus there are at most

$$\frac{Q}{\xi^{r-1} Q \cdot |S|^{-1} \cdot \log n} \le \xi |S|$$

vertices b with $\mathrm{mult}(b) > \xi^{r-1}Q \cdot |S|^{-1} \cdot \log n$. Now suppose $\mathbf{c}$ is not empty. Every path in $\mathcal{F}_{i+1}$ whose end contains $\{\mathbf{c}, b\}$ was constructed by the expansion of some path in $\mathcal{F}_i$ whose end contains $\mathbf{c}$. Note that every path expands at most by a factor of $\log n$ and by **P3** there are at most $\xi^{r-|\mathbf{c}|}Q \cdot |S|^{-|\mathbf{c}|} \log^{|\mathbf{c}|} n + 1$ paths in $\mathcal{F}_i$ whose end contains $\mathbf{c}$. If this bound is less than two, then there are at most $\log n$ vertices b with $\mathrm{mult}(\{\mathbf{c}, b\}) \geq 1$. Otherwise there are at most

$$\frac{2\xi^{r-|\mathbf{c}|}Q \cdot |S|^{-|\mathbf{c}|} \log^{|\mathbf{c}|+1} n}{\xi^{r-|\mathbf{c}|-1}Q \cdot |S|^{-|\mathbf{c}|-1} \log^{|\mathbf{c}|+1} n} = 2\xi|S|$$

vertices $x \in S_i$ with $\mathrm{mult}(\{\mathbf{c}, b\}) > \xi^{r-|\mathbf{c}|-1}Q \cdot |S|^{-|\mathbf{c}|-1} \cdot \log^{|\mathbf{c}|+1} n$.

Finally, we want to show that for each $\mathbf{c}$ there are at most $\xi'|S|$ vertices b in T_i which satisfy (iv). This is trivial for $D = \emptyset$, so we may assume that D is as given in Algorithm 2.

First suppose $|\mathbf{c}| = 0$. If a vertex b satisfies (iv), then it is in $(\xi'|S|)^{r-2}$ edges of D, so if there are $\xi'|S|$ such vertices then there are at least $(\xi'|S|)^{r-1}$ edges in D using vertices of T_i (note that edges of D only intersect T_i in one vertex). In other words, the number of blocked pairs $(\mathbf{a}, \mathbf{b})$ with $\mathbf{a} \in \mathcal{F}_t(\mathbf{u})$ and $\mathbf{b} \in S^{r-1}$ is at least

$$(\xi'|S|)^{r-1} \cdot \xi' Q \geq 2r \cdot 2^{2r}\xi|S|^{(r-1)} \cdot Q$$

using our choice of parameters (3). We conclude that there is a leaf $\mathbf{a}$ of $\mathcal{F}_t(\mathbf{u})$ that is in at least $2r \cdot 2^{2r}\xi|S|^{r-1}$ blocked pairs with tuples $\mathbf{b} \in S^{r-1}$. Fix this leaf. Now **P3** holds for $\mathbf{a}$, and we will show that this gives a contradiction. Consider the following property of tuples $\mathbf{b}$. For any sets A and B with vertices in $\mathbf{a}$ and $\mathbf{b}$ respectively, if $|A| + |B| = r - 1$ then $A \cup B$ is not in $\mathcal{E}$, while if $|A| + |B| < r - 1$ then we have $\deg_{\mathcal{E}}(A \cup B, S) \leq 2\xi^{r-|A|-|B|}|S|^{r-1-|A|-|B|}$. Trivially if $\mathbf{b}$ has the property, then $(\mathbf{a}, \mathbf{b})$ is not blocked. If $\mathbf{b}$ does not have the property, then let $B_{\mathbf{b}}$ be a set of minimal size witnessing the property's failure. Since $A \notin \mathcal{E}$ by **P1**, and by **P3**, we do not have $|B_{\mathbf{b}}| = 0$.

We now count the ways to create $\mathbf{b}$ which does not have the property. We choose vertices $b_1, \ldots, b_{r-1}$ one at a time until we create a witness $B \neq \emptyset$ that $\mathbf{b}$ cannot have the property. When we come to choose b_j, we have at most $|S|$ ways to choose it without creating a witness. If we are to choose b_j which witnesses the property's failure, then there are sets A and B' contained respectively in $\mathbf{a}$ and $\{b_1, \ldots, b_{j-1}\}$ such that $(A, B' \cup \{b_j\})$ fails the property. There are at most 2^{2r} choices for A and B'. Since (A, B') does not witness the property failing, by definition for each choice of A and B' there are at most $\xi|S|$ choices of b_j. Summing up, there are at most $r \cdot 2^{2r}\xi|S|^{r-1}$ tuples $\mathbf{b}$ which do not have the property. As all blocked pairs use a tuple from this set, this is the desired contradiction.

Now suppose $\mathbf{c}$ is a tuple for which there are at least $\xi'|S|$ vertices b satisfying (iv). In other words, there are more than $\xi'|S|$ vertices $b \in T_{i+1}$ with $\deg_D(\{\mathbf{c}, b\}, S) > (\xi'|S|)^{r-|\mathbf{c}|-2}$, which implies that

$$\deg_D(\mathbf{c}, S) > (\xi'|S|)^{r-|\mathbf{c}|-1}$$

in contradiction to **P5**.

Putting all this together we conclude that there are at most $\delta|S|$ vertices b such that $\mathbf{c}$ exists satisfying any one of the conditions (i)–(iv), as desired.

Now let $\mathbf{a}$ be a leaf of $\mathcal{F}_i$. We now reveal all r-sets containing $\mathbf{a}$ which were not revealed before and which use a vertex x of T_{i+1} which is good for $\mathbf{a}$. Let X be the number of edges $\{\mathbf{a}, x\}$ which appear. Then the expected value of X is at least $p(1-\delta)|T_{i+1}| \geq \frac{C}{20r} \log n$. Applying the Chernoff bound (1) we get that $X < \frac{C}{40r} \log n$ with probability at most $2\exp(-C \log n/(240r)) \leq n^{-4r}$. Let us suppose that $X \geq \log n$. Then Algorithm 3 does not fail to create the required number of paths from $\mathbf{a}$. Taking a union bound over the at most $|S|^{r-1}t$ such events, we obtain the stated success probability of Claim 5.

It remains to prove that **P3**, **P4** and **P5** also hold in $\mathcal{F}_{i+1}(\mathbf{u})$. But this is immediate, since we avoided choosing vertices which could cause their failure. □

Taking a union bound over the $2t$ steps, we conclude that with probability at most n^{-2r} there is a failure to construct either of the desired fans $\mathcal{F}_t(\mathbf{u})$ and $\mathcal{F}_t(\mathbf{v})$.

Connecting the fans. By construction, as set up in line 6 of Algorithm 3, all leaves of $\mathcal{F}_t(\mathbf{v})$ are not edges of D and thus not dangerous. Let L be the leaves from $\mathcal{F}_t(\mathbf{u})$ and L' the leaves from $\mathcal{F}_t(\mathbf{v})$ reversed. We now want to reveal more edges to connect a leaf from L with one from L'.

For $\mathbf{a} \in L$ and $\mathbf{b} \in L'$ let P be the tight path with $r-1$ edges on the vertices $(\mathbf{a}, \mathbf{b})$. There are $|L'| \cdot (1-\xi')|L| = (1-\xi')Q^2$ many such paths P, which are not blocked, because $\mathbf{b}$ is not dangerous. Let $\mathcal{P}$ be the set of all these paths which are not blocked.

Let I_P be the indicator random variable for the event that the path P appears, which occurs with probability p^{r-1}. Further let X be the random variable counting the number of paths which we obtain and note $X = \sum_{P\in\mathcal{P}} I_P$. With Janson's inequality (2) we want to bound the probability that $X = 0$. First let us estimate the expected value of X. By the observation from above we have $\mathbb{E}(X) = |\mathcal{P}|p^{r-1} \geq (1-\xi')(cC)^{r-1}\log^{r-1} n \geq \log n$.

Now consider two distinct paths $P = (\mathbf{a}, \mathbf{b})$ and $P' = (\mathbf{a}', \mathbf{b}')$, which share at least one edge. It follows from property **P4** of Claim 5 and the quantities Q and $|S|$, that two paths are identical if they share at least $r/2$ vertices in their end tuple. Since either the start or end $r/2$-tuple of one of the $(r-1)$-tuples from P has to agree with P', we can assume without loss of generality that $\mathbf{a} = \mathbf{a}'$. Further we can assume that for some $1 \leq j < r/2$, $\mathbf{b}$ and $\mathbf{b}'$ agree on the first j entries, but not in the $(j+1)$-st. They can not share another $r/2$ or more entries as this would imply $\mathbf{b} = \mathbf{b}'$. Thus P and P' share precisely an interval of length $r-1+j$ and thus j edges. With this we can bound $\mathbb{E}(I_P I_{P'}) \leq p^{2r-2-j}$.

Let $N_{P,j}$ be the number of paths P' such that P and P' share precisely j edges. The above shows that for fixed $P = (\mathbf{a}, \mathbf{b})$, $N_{P,j}$ is at most the number of choices of leaves $\mathbf{b}' \in L'$ such that $\mathbf{b}$ and $\mathbf{b}'$ only differ in the ending $(r-1-j)$-tuple, plus the number of choices of leaves $\mathbf{a}' \in L$ such that $\mathbf{a}$ and $\mathbf{a}'$ only differ in the start $(r-1-j)$-tuple. It follows from property **P4** of Claim 5, that the start j-tuple of $\mathbf{b}'$ and the end j-tuple of $\mathbf{a}'$ are the ends of at most $\xi^{r-j}Q \cdot |S|^{-j}\log^j n + 1$ many paths. This implies that $N_{P,j} \leq Q \cdot |S|^{-j}\log^j n$, because $j < r/2$.

We can now obtain for $P, P' \in \mathcal{P}$

$$\Delta = \sum_{P\neq P', P\cap P'\neq\emptyset} \mathbb{E}(I_P I_{P'}) = \sum_{P\in\mathcal{P}} \sum_{1\leq j<r/2} \Big(\sum_{|P'\cap P|=j} \mathbb{E}(I_P I_{P'}) \Big).$$

With the above we get

$$\begin{aligned}
\Delta &\leq \sum_{P\in\mathcal{P}} \sum_{1\leq j<r/2} N_{P,j} \cdot p^{2r-2-j} \\
&\leq |\mathcal{P}|^2 p^{2r-2} \sum_{1\leq j<r/2} |\mathcal{P}|^{-1} \cdot Q \cdot |S|^{-j}\log^j n \cdot p^{-j} \\
&\leq \mathbb{E}(X)^2 \cdot 2\,Q^{-1} \sum_{1\leq j<r/2} C^{-j} \leq \mathbb{E}(X)^2 3\,C^{-1}\log^{-1} n,
\end{aligned}$$

where we used that $|S| \geq Cp^{-1}\log n$ and $Q \geq \log n$. Hence, Janson's inequality (2) implies that $\mathbb{P}(X = 0) \leq \exp(-\mathbb{E}(X)^2/(\mathbb{E}(X) + \Delta)) \leq \exp(-\frac{C}{6}\log n)$. Thus we find some connection with probability at least $1 - n^{-2r}$.

But we do not want to reveal all the $O(Q^2)$ edges for all paths from $\mathcal{P}$, since this would add way to manu edges to the exposure hypergraph $\mathcal{E}$. The above argument proves that it is very likely that the desired connecting path exists and we will argue how to find such a path in an "economic" way. We find it by the following procedure. First we reveal all the edges at each leaf in L and L'. This entails adding $2Q$ edges to $\mathcal{E}$ and if $r = 3$ then we are already done and we have added $2Q \leq |S|$ edges to $\mathcal{E}$.

For $r \geq 4$ we then construct from each leaf of L all possible tight paths in S with $\lfloor (r-2)/2 \rfloor$ edges and similarly from each leaf of L' all tight paths of length $\lfloor (r-3)/2 \rfloor$. We do this by the obvious breadth-first-search procedure, revealing at each step all edges at the end of each currently constructed path with less than $\lfloor (r-2)/2 \rfloor$ (or $\lfloor (r-3)/2 \rfloor$ respectively) edges which have not so far been revealed and adding each end to $\mathcal{E}$. Trivially, if the desired path exists then two of these constructed paths will link up, so that this procedure succeeds in finding a connecting path with probability $1 - n^{-2r}$.

The expected number of edges in S containing any given $(r-1)$-set in S is $p(|S| - r + 1)$, is between $\frac{C}{2} \log n$ and $C \log n$. Thus by Chernoff's inequality and the union bound, with probability at least $1 - n^{-3r}$ no such $(r-1)$-set is in more than $2C \log n$ edges contained in S. It follows that the number of edges we add to $\mathcal{E}$ in this procedure is with probability at least $1 - n^{-3r}$ not more than

$$2Q \sum_{i=0}^{\lfloor (r-2)/2 \rfloor} (2C \log n)^i \leq 2p^{-(r-1)/2} \log n \cdot r(2C \log n)^{(r-2)/2}$$
$$= O\left(p^{-(r-2)} \log^{r-2} n\right) = O(|S|^{r-2}),$$

for $r \geq 4$. Putting this together with property **P2** of Claim 5 we see that the final exposure graph $\mathcal{E}'$ has at most $O(|S|^{r-2})$ edges more than $\mathcal{E}$, as desired.

Probability and runtime. Altogether we have that our algorithm for the Connecting Lemma fails with probability at most $n^{-2r} + n^{-2r} + n^{-3r} \leq n^{-5}$.

We now estimate the running time of our algorithm. In total we added $O(|S|^{r-2})$ many $(r-1)$-tuples to $\mathcal{E}$. For every $(r-1)$-tuple exposed, we have to go through at most n vertices until we found all new edges. This gives at most $O(n^{r-1})$ steps. We can easily keep track of the bounds for Claim 5 and update them after each event. Since there is nothing else to take care of, we have a total number of at most $O(n^{r-1})$ steps.

4.4. **Spike path version.** The statement of the lemma is almost the same as for the tight path version, Lemma 3.

Lemma 6 (Spike path Lemma). *For each $r \geq 3$ there exist $c, C > 0$ and a deterministic $O(n^{r-1})$-time algorithm whose input is an n-vertex r-uniform hypergraph G, a pair of distinct $(r-1)$-tuples $\mathbf{u}$ and $\mathbf{v}$, a set $S \subseteq V(G)$ and a $(r-1)$-uniform exposure hypergraph $\mathcal{E}$ on the same vertex set. The output of the algorithm is either 'Fail' or a spike path of even length $o(\log n)$ in G whose ends are $\mathbf{u}$ and $\mathbf{v}$ and whose interior vertices are in S, and an exposure hypergraph $\mathcal{E}' \supset \mathcal{E}$. We have $e(\mathcal{E}') \leq e(\mathcal{E}) + O(|S|^{r-2})$ and all the edges $E(\mathcal{E}') \setminus E(\mathcal{E})$ are contained in $S \cup \mathbf{u} \cup \mathbf{v}$.*

Suppose that G is drawn from the distribution $\mathcal{G}^{(r)}(n,p)$ with $p \geq C(\log n)^3/n$, that $\mathcal{E}$ does not contain any edges intersecting both S and $\mathbf{u} \cup \mathbf{v}$. If furthermore we have $|S| = Cp^{-1} \log n$ and $|e(\mathcal{E}[S])| \leq c|S|^{r-1}$ then the algorithm returns 'Fail' with probability at most n^{-5}.

Sketch proof. We modify the proof of Lemma 3 in the following simple ways. First, we will maintain fans of spike paths rather than tight paths, and we change Algorithm 3 line 5 so that the tuple $\mathbf{a}$ to be extended is the (unique) one whose extension continues to give us a spike path. Note that whenever we have a spike path ending in $\mathbf{a}$ and we extend the spike path by adding one vertex b then the end of the new spike path is an $(r-1)$-set whose vertices are contained in $(\mathbf{a}, b)$ (though in general not the last $r-1$ vertices nor in the same order). This is all we need to make our analysis of the fan construction work; it is not necessary to change anything in this part of the proof or the constants. Second, when we come to connect fans, we let L be the reverses of the end tuples of $\mathcal{F}_t(\mathbf{u})$ and L' be the end tuples of $\mathcal{F}_t(\mathbf{v})$, and (again) look for a tight path connecting a tuple in L to one in L'. This has no effect on the proof that a connecting path from some member of L to some member of L' exists, and the result is the desired spike path. The resulting spike path is of even length as both fans have the same size. □

5. Proof of the Reservoir Lemma

5.1. **Idea.** The reservoir path P_{res} will consist of absorbing structures (each "carrying" one vertex from R). More precisely, these absorbing structures can be seen as small reservoir path with reservoir of cardinality 1. Each of these small absorbers consists of a cyclic spike path plus the reservoir vertex, where pairs of spikes are additionally connected with tight paths (cf./ Figure 1).

First we choose the reservoir set R and disjoint sets U_1, U_2 and U_3. For every vertex in R we will reveal the necessary path segment in U_1. From the endpoints of these path we fan out and also close the backbone structure of the reservoir inside U_2. Finally we use U_3 and Lemma 3 to get the missing connections in the reservoir structures and connect all structures to one path P_{res}. In each step the relevant edges of the exposure graph $\mathcal{E}$ are solely coming from the same step.

5.2. **Proof.** We arbitrarily fix the reservoir set R of size $2Cp^{-1}\log n$ and disjoint sets U_1, U_2 and U_3 of the same size such that $S = R \cup U_1 \cup U_2 \cup U_3$ is of size $\frac{n}{4}$. First we want to build the absorbing structures for every $a \in R$, which have size roughly $t^2 = o(\log^2 n)$. There is a sketch of this structure for some $a \in R$ in Figure 1.

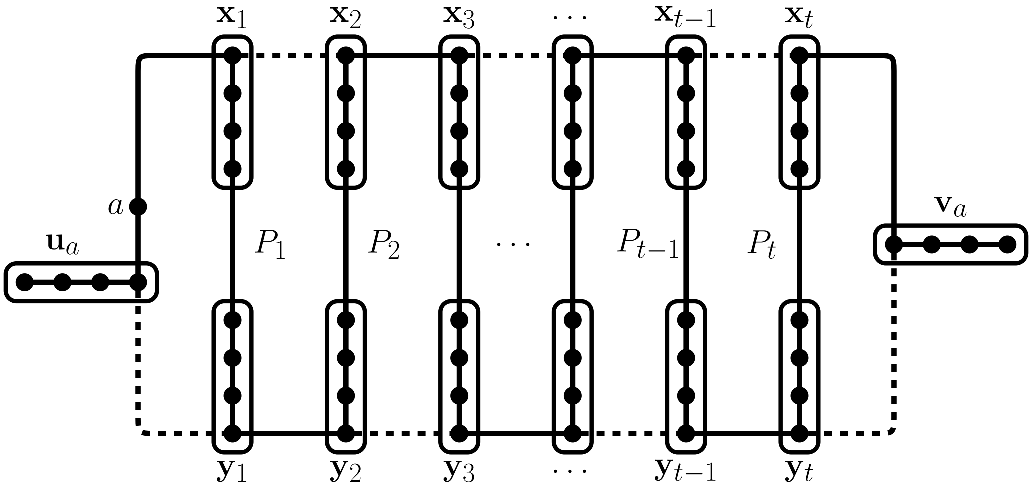

Figure 1. Illustration of the absorber for one vertex $a \in R$ and $r = 5$ with the path, which contains the vertex a.

So we fix $a \in R$. We want to construct the following tight path on $2r-1$ vertices containing a in the middle. The end tuples are $\mathbf{x}_1 = (x_1, \dots, x_{r-1})$ and $\mathbf{u}_a = (u_1, \dots, u_{r-1})$ and together with a we require that all the edges $\{x_{r-j}, \dots, x_1, a, u_1, \dots, u_{j-1}\}$ are present for $j = 1, \dots, \lfloor r$. We build this path by first choosing $x_1, \dots, x_{r-2}$ arbitrarily from U_1. Then we expose all edges containing $\{x_1, \dots, x_{r-2}, a\}$ to get x_{r-1}. We continue by exposing all edges containing the set $\{x_{r-j-1}, \dots, x_1, a, u_1, \dots, u_{j-1}\}$ to get u_j for $j = 1, \dots, \lfloor r-1$. The probability that in any of these cases we fail to find a new vertex inside a subset of U_1 of size at least $|U_1|/2$ is at most n^{-5} by Chernoff's inequality. A union bound over all r edges and over all $a \in R$ reveals that with probability at most n^{-3} we fail to construct the small starting graph for any a.

Recall that when adding edges, we always expose all edges containing one $(r-1)$-tuple and then add this to $\mathcal{E}$. All exposed $(r-1)$-tuples from this step are contained in $U_1 \cup R$ and none of them contains more than one vertex from R. Furthermore we did at most $O(|R| \cdot |U_1|) = O(n^2)$ many steps so far.

Now we want to build the absorbing structure for a. We partition each of U_2 and U_3 into parts of size $Cp^{-1}\log n$ (plus perhaps a smaller left-over set). We apply Lemma 6 to the $(r-1)$-tuples $\overleftarrow{\mathbf{x}_1}$ and $\overleftarrow{\mathbf{u}_a}$ and connect them with a spike path of even length $2t+2$ in some part of U_2, with $t = o(\log n)$. At each step we use a part of U_2 in which we have so far built the least spike paths for the application of Lemma 6, which is necessary to control the edges of $\mathcal{E}$ within this set. We use U_2 as both tuples are contained in U_1 and thus we have no problem with edges from $\mathcal{E}$ intersecting both U_2 and the end tuples. Let the spikes after $\mathbf{x}_1$ and $\mathbf{u}_a$ be called $\mathbf{x}_2, \dots, \mathbf{x}_t$ and $\mathbf{y}_1, \dots, \mathbf{y}_t$ respectively. The last remaining spike opposite of $\mathbf{u}_a$ we call $\mathbf{v}_a$. We apply the tight-path version of Lemma 3 to find paths P_i connecting the tuples $\mathbf{x}_i$ and $\mathbf{y}_i$ for $i = 1, \dots, t$ in a part of U_3. Again, we choose a part of U_3 which was used for building the least connecting paths so far. We use parts of U_3 for these connections, because all the spikes are contained in $U_1 \cup U_2$ and thus there are no edges of $\mathcal{E}$ intersecting U_3 and the spikes. This finishes the absorbing structure for a. It has end tuples $\mathbf{u}_a$ and $\mathbf{v}_a$.

To finish P_{res} we enumerate the vertices in R increasingly $a_1, \dots, a_{|R|}$. Then we use Lemma 3 repeatedly, again at each step using a part of U_3 which has been used least often previously, to connect the tuples $\mathbf{v}_{a_i}$ to $\mathbf{u}_{a_{i+1}}$ for $i = 1, \dots, |R| - 1$ with tight paths. Thus we have obtained the path P_{res} with end tuples $\mathbf{u} = \mathbf{u}_{a_1}$ and $\mathbf{v} = \mathbf{v}_{a_{|R|}}$.

The absorbing works in the following way for the structure of a single vertex $a \in R$. It relies on the fact, that the paths P_i can be traversed in both directions and that we can walk from any spike to its neighbouring spike using a tight path. The path which uses a (Figure 1) starts with $\mathbf{u}_a$, goes through a to $\mathbf{x}_1$ and then uses the path P_1 to $\mathbf{y}_1$. From there it goes via a tight path to $\mathbf{y}_2$ and uses P_2 to go back to $\mathbf{x}_2$. Going from $\mathbf{x}_i$ via path P_i to $\mathbf{y}_i$ and back from $\mathbf{y}_{i+1}$ through P_{i+1} to $\mathbf{x}_{i+1}$ for $i = 2, \dots, t-1$ the path ends up in $\mathbf{v}_a$ and uses all vertices. To avoid a (Figure 2) the path starting in $\mathbf{u}_a$ goes immediately to $\mathbf{y}_1$, then uses the path P_1 to go to $\mathbf{x_1}$. Alternating as above and traversing all the paths P_i in opposite direction we again end up in $\mathbf{v}_a$ and used all vertices but a.

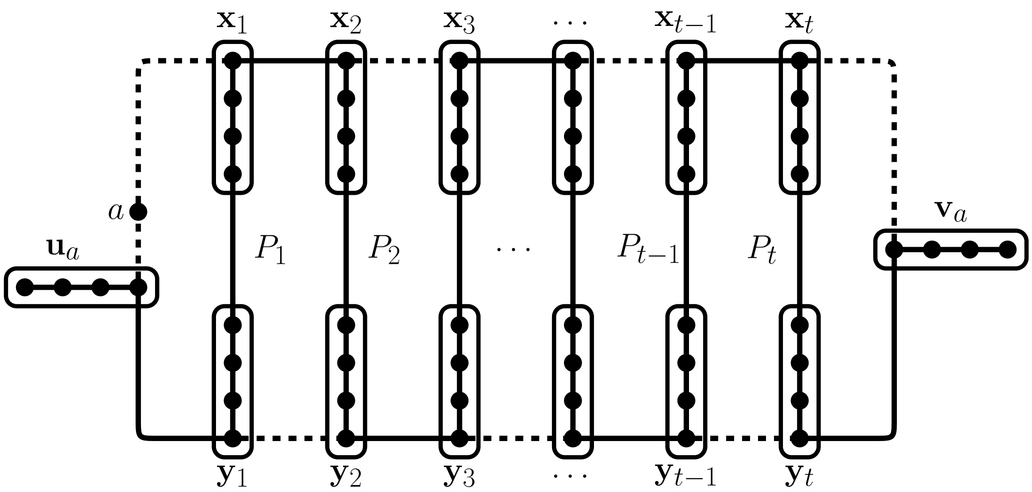

FIGURE 2. Illustration of the absorber for one vertex $a \in R$ and $r = 5$ with the path, which does not contain the vertex a.

For the proof of the lemma it remains to check that we obtain the right probability and we are indeed able to apply Lemma 3 as we described. It is immediate from the construction, that no edges of $\mathcal{E}$ are contained in $R \cup \mathbf{u} \cup \mathbf{v}$.

In total we are performing $|R|$ many connections with spike-paths and $|R| \cdot t + |R| - 1$ many connections with tight-paths. Thus altogether we have $o\left(p^{-1} \log^2 n\right)$ executions of Lemma 3 and Lemma 6. In each application we add $O\big(Cp^{-1} \log n\big)^{r-2}$ edges to $\mathcal{E}$ in some part of U_2 or U_3. Since each part initially contains no edges of $\mathcal{E}$, provided a given part has been used at most p^{-1} times the total number of edges of $\mathcal{E}$ in it is $o\big(Cp^{-1} \log n\big)^{r-1}$, and therefore we can apply Lemma 3 or 6 at least one more time with that part. Since $|U_2|$ and $|U_3|$ are of size linear in n, they each contain $\Omega\big(pn/\log n\big)$ parts. Thus we can perform in total $\Omega(n/\log n) = \Omega\big(p^{-1} \log^2 n\big)$ applications of either Lemma 3 or Lemma 6 before all parts have been used p^{-1} times and thus might acquire too many edges of $\mathcal{E}$. Since we do not need to perform that many applications, we conclude that the conditions of each of Lemma 3 and Lemma 6 are met each time we apply them.

Since the connecting lemma fails with probability at most n^{-5} the construction of this absorber fails with probability at most n^{-3}. In every connection there are at most $O(n^{r-1})$ steps performed and thus we need $o(n^{r-1}p^{-1}\log^2 n) = O(n^r)$ many steps for the construction of the absorber. □

6. Conclusion

In this paper we have improved upon the best known algorithms for finding a tight Hamilton cycle in $\mathcal{G}^{(r)}(n,p)$: we provide a deterministic algorithm with runtime $O(n^r)$ which for any edge probability $p \geq C(\log n)^3 n^{-1}$ succeeds a.a.s. While we give an affirmative answer to a question of Dudek and Frieze [8] in this regime, the question remains open for $e/n \leq p < C(\log n)^3 n^{-1}$ for $r \geq 4$, and $1/n \ll p < C(\log n)^3 n^{-1}$ for $r = 3$.

Let us now turn our attention to the closely related problem of finding the r-th power of a Hamilton cycle in the binomial random graph $\mathcal{G}(n,p)$, where $r \geq 2$. While a general result of Riordan [23] already shows that the threshold for $r \geq 3$ is given by $p = \Theta(n^{-1/r})$ (as observed in [18]), the threshold for $r = 2$ is still open, where the best known upper bound is a polylog-factor away from the first-moment lower bound $n^{-1/2}$ [20].

Since the result by Riordan is based on the second moment method it is inherently non-constructive. By contrast, the proof in [20] (for $r \geq 2$) is based on a quasi-polynomial time algorithm which for $p \geq C(\log n)^{8/r} n^{-1/r}$ finds the r-th power of an Hamilton a.a.s. in $\mathcal{G}(n,p)$, and which is very similar to their algorithm for finding tight Hamilton cycles in $\mathcal{G}^{(r)}(n,p)$. We think that our ideas are also applicable in this context and would provide an improved algorithm for finding r-th powers of Hamilton cycles in $\mathcal{G}(n,p)$, though we did not check any details.

Finally, it would be interesting to know the average case complexity of determining whether an n-vertex r-uniform hypergraph with m edges contains a tight Hamilton cycle. Our results (together with a standard link between the hypergeometric and binomial random hypergraphs) show that if $m \gg n^{r-1}\log^3 n$ then a typical such hypergraph will contain a Hamilton cycle, but the failure probability of our algorithm is not good enough to show that the average case complexity is polynomial time. For this one would need a more robust algorithm which can tolerate some 'errors' at the cost of doing extra computation to determine whether the 'error' causes Hamiltonicity to fail or not.

References

1. P. Allen, J. Böttcher, Y. Kohayakawa, and Y. Person, *Tight Hamilton cycles in random hypergraphs*, Random Structures Algorithms **46** (2015), no. 3, 446–465.
2. D. Angluin and L. G. Valiant, *Fast probabilistic algorithms for Hamiltonian circuits and matchings*, J. Comput. System Sci. **18** (1979), no. 2, 155–193.
3. Andreas Björklund, *Determinant sums for undirected Hamiltonicity*, SIAM Journal on Computing **43** (2014), no. 1, 280–299.
4. B. Bollobás, *The evolution of sparse graphs*, Graph theory and combinatorics (Cambridge, 1983), Academic Press, London, 1984, pp. 35–57. MR 777163 (86i:05119)
5. B. Bollobás, T. I. Fenner, and A. Frieze, *An algorithm for finding Hamilton paths and cycles in random graphs*, Combinatorica **7** (1987), no. 4, 327–341. MR 931191 (89h:05049)
6. D. Clemens, J. Ehrenmüller, and Y. Person, *A Dirac-type theorem for Hamilton Berge cycles in random hypergraphs.*, Discrete mathematical days. Extended abstracts of the 10th "Jornadas de matemática discreta y algorítmica" (JMDA), Barcelona, Spain, July 6–8, 2016, Amsterdam: Elsevier, 2016, pp. 181–186.
7. A. Dudek and A. Frieze, *Loose Hamilton cycles in random uniform hypergraphs*, Electron. J. Combin. **18** (2011), no. 1, Paper 48, 14. MR 2776824 (2012c:05275)
8. ——, *Tight Hamilton cycles in random uniform hypergraphs*, Random Structures Algorithms **42** (2013), no. 3, 374–385.
9. E. Friedgut, *Sharp thresholds of graph properties, and the k-sat problem*, J. Amer. Math. Soc. **12** (1999), no. 4, 1017–1054, With an appendix by Jean Bourgain.
10. A. Frieze, *Loose Hamilton cycles in random 3-uniform hypergraphs*, Electron. J. Combin. **17** (2010), no. 1, Note 28, 4. MR 2651737 (2011g:05268)

11. S. Janson, T. Łuczak, and A. Ruciński, *Random graphs*, Wiley-Interscience, New York, 2000.
12. A. Johansson, J. Kahn, and V. Vu, *Factors in random graphs*, Random Structures Algorithms **33** (2008), no. 1, 1–28. MR 2428975 (2009f:05243)
13. R. M Karp, *Reducibility among combinatorial problems*, Complexity of computer computations, Springer, 1972, pp. 85–103.
14. J. Komlós and E. Szemerédi, *Limit distribution for the existence of Hamiltonian cycles in a random graph*, Discrete Math. **43** (1983), no. 1, 55–63. MR 680304 (85g:05124)
15. A.D. Korshunov, *Solution of a problem of Erdős and Renyi on Hamiltonian cycles in nonoriented graphs.*, Sov. Math., Dokl. **17** (1976), 760–764.
16. ______, *Solution of a problem of P. Erdős and A. Renyi on Hamiltonian cycles in undirected graphs*, Metody Diskretn. Anal. **31** (1977), 17–56.
17. M. Krivelevich, *Triangle factors in random graphs*, Combinatorics, Probability and Computing **6** (1997), no. 3, 337–347.
18. D. Kühn and D. Osthus, *On Pósa's conjecture for random graphs*, SIAM Journal on Discrete Mathematics **26** (2012), no. 3, 1440–1457.
19. R. Montgomery, *Embedding bounded degree spanning trees in random graphs*, arXiv:1405.6559v2.
20. R. Nenadov and N. Škorić, *Powers of Hamilton cycles in random graphs and tight Hamilton cycles in random hypergraphs*, arXiv preprint arXiv:1601.04034 (2017).
21. D. Poole, *On weak Hamiltonicity of a random hypergraph*, arXiv:1410.7446 (2014).
22. L. Pósa, *Hamiltonian circuits in random graphs*, Discrete Math. **14** (1976), no. 4, 359–364. MR 0389666 (52 #10497)
23. O. Riordan, *Spanning subgraphs of random graphs*, Combinatorics, Probability and Computing **9** (2000), no. 2, 125148.
24. V. Rödl, A. Ruciński, and E. Szemerédi, *A Dirac-type theorem for 3-uniform hypergraphs*, Combin. Probab. Comput. **15** (2006), no. 1-2, 229–251.
25. E. Shamir, *How many random edges make a graph Hamiltonian?*, Combinatorica **3** (1983), no. 1, 123–131.