Peeling Close to the Orientability Threshold Spatial Coupling in Hashing-Based Data Structures

STEFAN WALZER, Department of Computer Science, Karlsruhe Institute of Technology, Karlsruhe, Germany

In multiple-choice data structures each element x in a set S of m *keys* is associated with a random set $e(x) \subseteq [n]$ of *buckets* with capacity $\ell \geq 1$ by hash functions. This setting is captured by the hypergraph $H = ([n], \{e(x) \mid x \in S\})$. Accommodating each key in an associated bucket amounts to finding an *ℓ-orientation* of H assigning to each hyperedge an incident vertex such that each vertex is assigned at most ℓ hyperedges. If each subhypergraph of H has minimum degree at most ℓ, then an ℓ-orientation can be found greedily and H is called *ℓ-peelable*. Peelability has a central role in invertible Bloom lookup tables and can speed up the construction of retrieval data structures, perfect hash functions, and cuckoo hash tables.

Many hypergraphs exhibit sharp density thresholds with respect to ℓ-orientability and ℓ-peelability, i.e., as the density $c = \frac{m}{n}$ grows past a critical value, the probability of these properties drops from almost 1 to almost 0. In fully random k-uniform hypergraphs the thresholds $c^*_{k,\ell}$ for ℓ-orientability significantly exceed the thresholds for ℓ-peelability. In this article, for every $k \geq 2$ and $\ell \geq 1$ with $(k, \ell) \neq (2, 1)$ and every $z > 0$, we construct a new family of random k-uniform hypergraphs with i.i.d. random hyperedges such that *both* the ℓ-peelability and the ℓ-orientability thresholds approach $c^*_{k,\ell}$ as $z \to \infty$. In particular, we achieve 1-peelability at densities arbitrarily close to 1, extending the reach of greedy algorithms.

Our construction is simple: The n vertices are linearly ordered and each hyperedge selects its k elements uniformly at random from a random range of $\frac{n}{z+1}$ consecutive vertices. We thus exploit the phenomenon of *threshold saturation via spatial coupling* discovered in the context of low-density parity-check codes. Once the connection to data structures is in plain sight, a framework by Kudekar, Richardson and Urbanke does the heavy lifting in our proof.

We demonstrate the usefulness of our construction using our hypergraphs as a drop-in replacement in a retrieval data structure by Botelho et al. This reduces memory usage from $\approx 1.23m$ bits to $\approx 1.12m$ bits (for input size m). Using $k > 3$ attains, at small sacrifices in running time, further improvements to memory usage.

CCS Concepts: • **Theory of computation → Bloom filters and hashing**; **Design and analysis of algorithms**; **Error-correcting codes**; • **Mathematics of computing → Hypergraphs**; **Random graphs**;

Additional Key Words and Phrases: Peeling Threshold, Spatial Coupling, Hashing, Random Hypergraph, Retrieval, Succinct Data Structure

ACM Reference format:
Stefan Walzer. 2025. Peeling Close to the Orientability Threshold Spatial Coupling in Hashing-Based Data Structures. *ACM Trans. Algor.* 21, 3, Article 33 (July 2025), 23 pages.
https://doi.org/10.1145/3711822

Authors' Contact Information: Stefan Walzer (corresponding author), Department of Computer Science, Karlsruhe Institute of Technology, Karlsruhe, Germany; e-mail: stefan.walzer@kit.edu.

ACM 1549-6333/2025/7-ART33
https://doi.org/10.1145/3711822

1 Introduction

Various data structures relying on the "power of multiple choices" [56] feature a set of n *buckets*, indexed by $[n] = \{1, ..., n\}$, each of capacity $\ell \geq 1$, and a set of m *keys*, each associated with $k \geq 2$ random buckets *via* hash functions. Take *cuckoo hashing* for instance. The task is to place each key into one of the k buckets associated with it such that no bucket is overloaded. This can be modeled by a hypergraph $H = (V, E)$ with vertex set $V = [n]$ representing buckets and m hyperedges of size k representing keys. A placement of keys corresponds to an *ℓ-orientation* of H [26], which is a function $o : E \to V$ with $o(e) \in e$ for all $e \in E$ and $|o^{-1}(v)| \leq \ell$ for all $v \in V$. If each key x is assigned its buckets $e(x) = \{h_1(x), h_2(x), ..., h_k(x)\} \subseteq [n]$ by k independent and fully random hash functions, then the corresponding *fully random k-uniform hypergraph* is denoted by $H^k_{n,m}$ (the issue of repeated incidences $h_i(x) = h_j(x)$ or repeated hyperedges $e(x) = e(y)$ is irrelevant for our purposes).

For a family $(H_{n,c})_{c\in\mathbb{R}_{\geq 0}, n\in\mathbb{N}}$ of random hypergraphs we say that c^* is a threshold for a property P if $c^* = \sup\{c \in \mathbb{R}_{\geq 0} \mid \lim_{n\to\infty} \Pr[H_{n,c} \in P] = 1\}$. The thresholds $c^*_{k,\ell}$ for ℓ-orientability of $(H^k_{n,cn})_{c\in\mathbb{R}^+_0, n\in\mathbb{N}}$ are known[1] for all $k \geq 2$ and $\ell \geq 1$, see [11, 15, 25–27, 30, 58]. They determine the limit $c^*_{k,\ell}/\ell$ of the memory efficiency (used space cn over allocated space ℓn) that can be reliably achieved by cuckoo hash tables.

Some applications, however, rely on the stronger hypergraph property of *ℓ-peelability*, which means that all subhypergraphs[2] of H have minimum degree at most ℓ. This often enables greedy, linear time construction algorithms. To place all keys in a cuckoo hash table, repeatedly look for a bucket that is associated with at most ℓ unplaced keys and place those keys in that bucket. The price to pay is typically reduced memory efficiency, since, at least when fully random hypergraphs are concerned, ℓ-peelability thresholds $c^{\triangle}_{k,\ell}$ fall short of ℓ-orientability thresholds $c^*_{k,\ell}$ as shown in Table 1. For a derivation of peelability thresholds, see [14, 37, 57, 59] and [51, Chapter 18].

The main contribution of this article is to propose a new kind of distribution for hyperedges, or equivalently, a new way to set up hash functions, that result in k-uniform hypergraphs with an ℓ-peelability threshold close to the ℓ-orientability threshold $c^*_{k,\ell}$ of $(H^k_{n,cn})_{c\in\mathbb{R}^+_0, n\in\mathbb{N}}$:

Theorem 1. *Let $k \geq 2$ and $\ell \geq 1$ with $k + \ell \geq 4$. For each n and $c < c^*_{k,\ell}$ there is a distribution $D^{k,\ell}_{n,c}$ on k-subsets of $[n]$ such that the random hypergraph $\hat{F}^{k,\ell}_{n,cn}$ with vertex set $[n]$ and $m = \lfloor cn \rfloor$ hyperedges independently sampled according to $D^{k,\ell}_{n,c}$ is ℓ-peelable with probability $1 - o(1)$.*

Before presenting our construction in Section 1.3, we explain the value of peelability in data structures and coding theory. In the latter field, the idea underlying our construction is known as "spatial coupling" and a suitable toolset for analyzing corresponding thresholds already exists. Sections 2 to 5 are devoted to proving our theorems, Section 6 presents experiments demonstrating the practical value of our approach.

1.1 Data Structures Benefitting from Peelability

In the following, S is always a set of $m = cn$ objects from some universe $\mathcal{U}$ and $[n]$ a set of buckets. Each $x \in S$ is associated with several buckets $e(x) := \{h_1(x), ..., h_k(x)\} \subseteq [n]$ *via* a constant[3] number $k \geq 2$ of hash functions $h_1, ..., h_k : \mathcal{U} \to [n]$ (which we assume require a negligible amount of space to store). This gives rise to a hypergraph $H = ([n], \{e(x) \mid x \in S\})$. We review data structures benefitting from ℓ-peelability of H (mostly for $\ell = 1$).

[1] They are also known to be *sharp* (cf. [28]), meaning $c^* = \inf\{c \in \mathbb{R}_{\geq 0} \mid \Pr[H_{c,n} \in P] \xrightarrow{n\to\infty} 0\}$ also holds.
[2] A subhypergraph of $H = (V, E)$ is a hypergraph $H' = (V', E')$ with $V' \subseteq V$ and $E' \subseteq E \cap 2^{V'}$.
[3] Some constructions allow $k = k(x)$ to depend on the key [47, 62].

Table 1. (Normalized) ℓ-Orientability and ℓ-Peelability Thresholds ($c^*_{k,\ell}/\ell, c^{\triangle}_{k,\ell}/\ell$) of Fully Random k-Uniform Hypergraphs, Rounded to Four Decimals

ℓ/k	2	3	4	5	6
1	(0.5000, 0.0000)	(0.9179, 0.8185)	(0.9768, 0.7723)	(0.9924, 0.7018)	(0.9974, 0.6371)
2	(0.8970, 0.8377)	(0.9882, 0.7764)	(0.9982, 0.6668)	(0.9997, 0.5789)	(1.0000, 0.5108)
3	(0.9592, 0.8582)	(0.9973, 0.7248)	(0.9998, 0.6036)	(1.0000, 0.5152)	(1.0000, 0.4496)
4	(0.9804, 0.8499)	(0.9993, 0.6867)	(1.0000, 0.5624)	(1.0000, 0.4755)	(1.0000, 0.4123)
5	(0.9896, 0.8365)	(0.9998, 0.6579)	(1.0000, 0.5331)	(1.0000, 0.4479)	(1.0000, 0.3867)
6	(0.9941, 0.8229)	(0.9999, 0.6353)	(1.0000, 0.5108)	(1.0000, 0.4272)	(1.0000, 0.3677)

The former quickly approach 1 as k or ℓ increases, the latter do not.

Cuckoo Hash Table [21, 52, 58]. A cuckoo table implements a set or dictionary data structure with key set S. Each $x \in S$ (and, possibly, associated data) should be stored in exactly one bucket $o(x)$, and each bucket can hold up to ℓ objects. To allow for constant-time lookups, we demand $o(x) \in e(x)$, which asks for an ℓ-orientation of H. If H is ℓ-peelable, a greedy construction in linear time is possible.

Otherwise, linear time constructions of ℓ-orientations are only known for the fully random hypergraph $H^k_{n,cn}$ with $c < c^*_{k,\ell}$ in the following cases. For graphs (i.e., $k = 2$) and $\ell \geq 2$, linear time algorithms are described in [11, 25]. For $\ell = 1$ and $k \geq 3$, consider [38, 39]. It is empirically plausible that random walk insertion can maintain an ℓ-orientation in a dynamic setting with expected constant time per update for any k and ℓ; a partial answer is given in [29].

Invertible Bloom Lookup Table (IBLT) [33]. Among other things, IBLTs can be used to construct error correcting codes [55] and solve the set reconciliation and straggler identification problem [23]. The data structure is inspired by Bloom Filters [4] and Bloomier Filters [12]. In IBLTs, each bucket $v \in [n]$ stores $\bigoplus_{x \in N(v)} x$, the bit-wise xor of (the bit representations of) the objects $N(v) := \{x \in S \mid v \in e(x)\}$ incident to v, as well as the degree $|N(v)|$. Note that this data structure is easy to maintain when insertions or deletions modify S, even through phases with $|S| \gg n$. Here, a ListEntries operation can be supported that recovers S if H is 1-peelable and that fails otherwise.

Retrieval [16, 18, 20, 31, 60]. An *r-bit retrieval data structure D_f* (also called *static function*) is constructed from a function $f : S \to \{0,1\}^r$. The only operation "eval" must satisfy $\text{eval}(D_f, x) = f(x)$ for all $x \in S$. The interesting setting is when D_f may only occupy $O(rm)$ bits. Note that naively storing f as a set of pairs requires $m \cdot (r + \log|\mathcal{U}|)$ bits. To save space, we exploit that the output of $\text{eval}(D_f, y)$ for $y \in \mathcal{U} \setminus S$ may yield an arbitrary element of $\{0,1\}^r$ and that membership queries "$x \in S$?" need not be supported.

The idea is to find and store values $b_1, b_2, \ldots, b_n \in \{0,1\}^r$ that satisfy the linear equations $\bigoplus_{i \in e(x)} b_i = f(x)$ for $x \in S$ (with $\oplus$ denoting bit-wise xor). When found, the sequence $(b_1, \ldots, b_n)$ is sufficient to answer eval-queries and takes up $rn = rm/c$ bits as desired.

The existence of a solution $(b_1, \ldots, b_n)$ is guaranteed if the incidence matrix of H has rank m. Actually solving the linear system may take quadratic or cubic time. If H is 1-peelable, however, then the matrix is in row echelon form up to row and column exchanges and a solution can be found in linear time.[4]

[4]The matrix being in "triangular form" motivates our use of the symbol "$c^{\triangle}$" for peeling thresholds.

Retrieval data structures are used in space efficient implementations of *filters* and *perfect hash functions* as follows:

↪ *XOR-Filters* [12, 16, 34, 68]. A filter for S with false positive rate $\varepsilon > 0$ is a randomized data structure supporting a member operation with $\text{member}(x) = 1$ for $x \in S$ and $\Pr[\text{member}(y) = 1] \leq \varepsilon$ for $y \notin S$. The best known example is the Bloom filter [4, 10, 49]. The following more space efficient alternative was appropriately dubbed "xor-filter."
We first select a random "fingerprint" function $f : \mathcal{U} \to \{0,1\}^r$. Like all hash functions, we may assume that f can be stored in $\mathcal{O}(1)$ space. However, we also store its restriction $f_S : S \to \{0,1\}^r$ to S explicitly in a retrieval data structure D_{f_S} and define $\text{member}(x) := \mathbb{1}[\text{eval}(D_{f_S}, x) = f(x)]$. For $x \in S$ we obtain $\text{member}(x) = 1$ by construction. For $x \notin S$, the independence of D_{f_S} from x and $f(x)$ guarantees that $\Pr[\text{member}(x) = 1] = 2^{-r} = \varepsilon$.

↪ *Perfect Hashing* [2, 7–9, 31, 50]. A *perfect hash function* for a key set S is an injective function $p : S \to [n]$ with n not much larger than $m = |S|$ such that p is efficient to store and evaluate. A standard construction considers $H = H^4_{n,m}$ with $c = \frac{m}{n} < c^*_{4,1} \approx 0.977$ and a 1-orientation $o : E \to V$ of H (which exists **With High Probability (whp)**). Then $p(x) := o(e(x))$ is injective. Since $o(e(x)) \in e(x) = \{h_1(x), ..., h_4(x)\}$ there is $f : S \to \{1,2,3,4\}$ such that $o(e(x)) = h_{f(x)}(x)$ for $x \in S$. Thus we only need to store f with a (two-bit) retrieval data structure (as well as $h_1, ..., h_4$) to be able to evaluate $p(x) = h_{f(x)}(x)$.

1.2 Connection to Coding Theory

Our proof of Theorem 1 imports methods from coding theory. To explain the connection to this field, we briefly introduce the **Binary Erasure Channel (BEC)** and point out the relationship to our notions on hypergraphs. This reveals how closely the task of constructing good codes aligns with the task of constructing good **Hashing-Based Data Structures (HBDS)**.

1.2.1 The BEC and ***Low-Density Parity-Check (LDPC)*** *Codes.* The BEC is a simple but important setting. We recommend [61, Chapter 3] for an excellent introduction to this subject. When a sequence $(x_1, ..., x_m) \in \{0,1\}^m$ is sent over the BEC, the receiver sees a sequence $(y_1, ..., y_m) \in \{0,1,?\}^m$ where for each $i \in [m]$ independently, the ith bit is *erased* ($y_i = ?$) with probability $\varepsilon \in [0,1]$ and unchanged ($y_i = x_i$) with probability $1 - \varepsilon$. For reliable communication over such channels, redundancy is introduced. In *linear codes*, several *parity conditions* are each specified by a set $P \subseteq [m]$ and dictate that $\bigoplus_{i \in P} x_i$ is zero. The set of admissible messages (*codewords*) then forms a linear subspace of $\{0,1\}^m$.

To relate this to hypergraphs, let V be the set of all parity conditions and let $E^+ = \{e_1, ..., e_m\}$ where $v \in e_i$ if x_i is involved in parity condition v. The incidence graph of $H^+ = (V, E^+)$ is known as the *Tanner graph* [65]. In *ldpc codes* the Tanner graph is sparse.

During transmission, bits corresponding to some set $E \subseteq E^+$ are erased, and we consider $H = (V, E)$. When decoding, we seek an assignment $x_{\text{dec}} : E \to \{0,1\}$ such that for $v \in V$ we have $\bigoplus_{e \in E, e \ni v} x_{\text{dec}}(e) = c_v$ where c_v is the parity of the successfully transmitted bits involved in parity condition v. The existence of a solution is guaranteed by construction, namely $x_{\text{dec}}(e_i) = x_i$ for $e_i \in E$. Uniqueness of the solution and thus success of the ideal *maximum a posteriori probability decoder* requires the kernel of the incidence matrix of H to be trivial—a property that implies 1-orientability. Success of the linear time *belief propagation decoder* requires 1-peelability of H. This decoder repeatedly identifies a parity condition, where all but one of the involved bits are known, and then decodes the unknown bit.

1.2.2 Good Codes vs. Good Data Structures. The goals in HBDS are sufficiently similar to those in LDPC decoding to render the techniques from LDPC codes useful in HBDS.

Hyperedge Size. The (average) hyperedge size k is, in HBDS, related to (average) query time and (average) number of cache faults per query. In LDPC codes, k is the (average) number of parity conditions relating to each message bit and contributes to overall encoding and decoding time. Thus, *small k is good.*

Density. In HBDS, a high edge density $c = |E|/|V|$ means accommodating many objects in little space (high load), while in LDPC codes it means recovering many erased bits from little redundancy (high rate). Thus, *large c is good.*

Peelability. As discussed, peelability is useful in both worlds, enabling greedy construction of data structures and greedy decoding of messages, respectively. To the author's knowledge, ℓ-peelability for $\ell > 1$ plays no role in coding theory, but the tools we import can handle this more general case nonetheless.

An aspect with imperfect alignment is *the role of randomness.* Data structures must handle *arbitrary* key sets and the role of hash functions is, in part, to make them behave like *random* keys. Unless the hash function is itself a complex data structure, it is unreasonable to assume that this produces any *useful* correlation. Therefore, the hypergraphs we deal with have independent random hyperedges (at best).

An LDPC code on the other hand is given by a fixed hypergraph H^+. Randomness is frequently involved in its construction, but we are in principle free to design H^+, for instance, we might give all vertices the same degree. Since H arises from a random ε-fraction of the hyperedges of H^+, this gives us control (proportional to ε) on H as well. In this sense the hypergraphs relevant for HBDS are a special case of those relevant for LDPC codes.

1.3 New Results

In what we call "spatially coupled hypergraphs," the vertices are linearly ordered and each hyperedge selects its k elements uniformly at random from a random range of consecutive vertices.[5] Concretely:

Definition 1 (Spatially Coupled Hypergraph). For $k \in \mathbb{N}$, $z, c \in \mathbb{R}_0^+$ and $n \in \mathbb{N}$, let $F_n = F(n, k, c, z)$ be the random k-uniform hypergraph with vertex set $V = [n]_0 = \{0, \ldots, n-1\}$ and edge set E of size $m = \lfloor cn\frac{z}{z+1} \rfloor$. Each edge $e \in E$ is independently obtained as $e = \left\{ \left\lfloor \frac{y+o_i}{z+1} n \right\rfloor \mid i \in [k] \right\}$ where $y \in [\frac{1}{2}, z + \frac{1}{2})$ and $o_1, \ldots, o_k \in [-\frac{1}{2}, \frac{1}{2}]$ are chosen uniformly at random.

We call $y \in Y := [\frac{1}{2}, z + \frac{1}{2})$ the position of $e \in E$ and $\frac{v(z+1)}{n} \in X = [0, z+1)$ the position of $v \in V$.

In Figure 1 we sketch aspects of the construction. It is possible (but rare) that incidences repeat within a hyperedge and that the same hyperedge appears several times in F_n. Note that the hyperedge density is $|E|/|V| = c\frac{z}{z+1}$ and only approaches c for large z. The main technical contribution of this article is the following Theorem.

THEOREM 2. *Let $k, \ell \in \mathbb{N}$, with $k \geq 2$ and $k + \ell \geq 4$. Then we have:*

(i) $\forall c < c^*_{k,\ell} : \forall z \in \mathbb{R}^+ : \Pr[F_n \textit{ is } \ell\textit{-peelable}] \xrightarrow{n\to\infty} 1.$

(ii) $\forall c > c^*_{k,\ell} : \exists z^* \in \mathbb{R}^+ : \forall z \geq z^* : \Pr[F_n \textit{ is} \ell\textit{-orientable}] \xrightarrow{n\to\infty} 0.$

Let us distil the main takeaways from these claims, including Theorem 1.

[5] In the precursor to this work [19] we partitioned the set of vertices into $z + k - 1$ linearly ordered segments (for $z \in \mathbb{N}$). Each hyperedge selects a random range of k consecutive segments and one random vertex from each of these segment. While this old construction has similar *empirical* properties as the updated proposal, the discreteness in the construction leads to (slightly) inferior thresholds and does not seem to admit an elegant analysis.

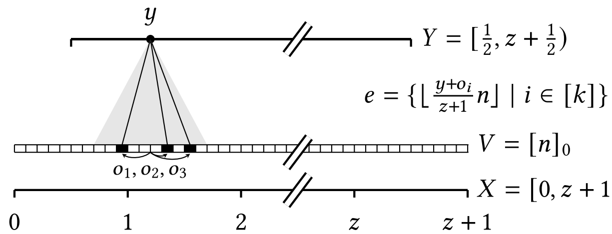

Fig. 1. In the construction from Definition 1, the vertex set $V = [n]_0$ is arranged linearly along the "coupling dimension" $X = [0, z+1)$. Thus, each vertex has a position $x \in X$. Each hyperedge e is independently obtained as follows. First, pick a random position y uniformly from $Y = [\frac{1}{2}, z + \frac{1}{2})$. Then pick the k incidences of e independently and uniformly at random from the vertices with positions in $[y - \frac{1}{2}, y + \frac{1}{2}]$.

Corollary 3. *Let $k, \ell \in \mathbb{N}$ with $k \geq 2$ and $k + \ell \geq 4$. For $z \in \mathbb{R}^+$ consider the family $(F(n, k, c, z))_{c \in \mathbb{R}_0^+, n \in \mathbb{N}}$. Let $f^{\triangle}_{k,\ell,z}$ be its threshold for ℓ-peelability and $f^*_{k,\ell,z}$ its threshold for ℓ-orientability. Then we have:*

(i) $\forall z \in \mathbb{R}^+ : f^{\triangle}_{k,\ell,z} \geq c^*_{k,\ell}$.
(ii) $\limsup_{z \to \infty} f^*_{k,\ell,z} \leq c^*_{k,\ell}$.
(iii) *Let $f^{\triangle}_{k,\ell} = \lim_{z \to \infty} f^{\triangle}_{k,\ell,z}$ and $f^*_{k,\ell} = \lim_{z \to \infty} f^*_{k,\ell,z}$. Then $f^{\triangle}_{k,\ell} = f^*_{k,\ell} = c^*_{k,\ell}$.*

Proof of Corollary 3. Claims (i) and (ii) are immediate consequences of the claims from Theorem 2. Since $f^{\triangle}_{k,\ell,z} \leq f^*_{k,\ell,z}$ we conclude (iii). □

Proof of Theorem 1. Let $c = c^*_{k,\ell} - \varepsilon$ for some $\varepsilon > 0$. We define $\hat{F}^{k,\ell}_{n,cn} := F(n, k, c\frac{z+1}{z}, z = 2\ell/\varepsilon)$. Note that this hypergraph is of the desired form with n vertices and $\lfloor cn \rfloor$ i.i.d. hyperedges.[6] Observe that $c\frac{z+1}{z} = c\frac{2\ell/\varepsilon+1}{2\ell/\varepsilon} = c(1 + \varepsilon/(2\ell)) \leq c^*_{k,\ell} - \varepsilon/2 \leq f^{\triangle}_{k,\ell,z} - \varepsilon/2$ using Corollary 3 *(i)* and the trivial bound $c^*_{k,\ell} \leq \ell$. Therefore, $\hat{F}^{k,\ell}_{n,cn}$ is ℓ-peelable with probability $1 - o(1)$ by definition of $f^{\triangle}_{k,\ell,z}$. □

Our construction is in the spirit of a technique from coding theory, see Section 1.5. Note that constructions similar to ours can already be found in [36] and [32], however, the goals of these papers are very different. Relative to these results, we can offer: *(1)* A generalization to $\ell > 1$. *(2)* A more elegant construction using the updated tools from [42] (continuous[7] coupling dimension). *(3)* A framing with data structures in mind and a demonstration of practical benefits for data structures.

1.4 Comparison with Previous Constructions of Peelable Hypergraphs

The applications in Section 1.1 require hypergraph families with i.i.d. random hyperedges, preferably with small average hyperedge size k and large ℓ-peelability threshold. In Figure 2, we compare previous constructions to our own with respect to these criteria in the most relevant case of $\ell = 1$.

The thresholds $c^{\triangle}_{k,1}$ of the fully random families $(H^k_{n,cn})_{c \in \mathbb{R}_0^+, n \in \mathbb{N}}$ for $k \geq 3$ (◆) [57] are decreasing in k and thus mostly $k = 3$ is of interest. The thresholds of a non-uniform construction (■) [47], well-known in coding theory, approach 1 for $k \to \infty$. However, the maximum hyperedge size is exponential in the average hyperedge size k, which is problematic for some applications. Further tradeoffs (●) were examined by Rink [62], for example, a hyperedge size of 3 for ≈ 89% of the

[6]We remark that the "diagonal" family $(\hat{F}^{k,\ell}_{n,cn})_{c,n}$ is mostly of theoretical interest. We do not claim that our choice of $z = 2\ell/\varepsilon$ is particularly practical. The fact that the hyperedge density affects the underlying parameter z is also inconvenient when constructing dynamic data structures.
[7]In [36] the coupling dimension is discrete. In our terms, this means that the set of admissible positions of a hyperedge is $Y \cap (\frac{1}{w}\mathbb{Z})$ for some constant $w \in \mathbb{N}$. Our construction arises for $w \to \infty$.

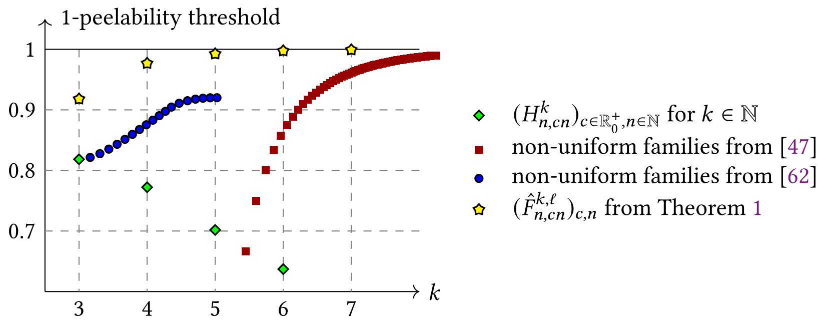

Fig. 2. Tradeoffs between hyperedge size and 1-peelability threshold. Concretely, a dot at $(k, c^{\triangle}) \in \mathbb{R}^2$ indicates a family $(H_{c,n})_{c\in\mathbb{R}^+_0,n\in\mathbb{N}}$ of random hypergraphs where $H_{c,n}$ has n vertices, $\lfloor cn \rfloor$ random independent hyperedges, expected hyperedge size k. The value $c^{\triangle}$ is the 1-peelability threshold of the family.

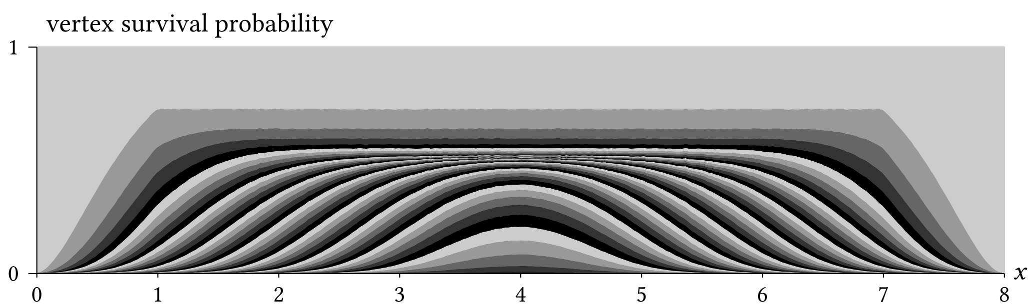

Fig. 3. The "layers" of a hypergraph sampled according to Definition 1 with parameters $k = 3$, $c = 0.85$, $z = 7$ and $m = 10^8$. It happens to be peelable in 48 rounds, that are assigned shades of gray in a round-robin fashion. For each round $r \in \{0, \dots, 48\}$ and each position $x \in X = [0, 8]$, the picture shows the fraction of vertices close to x that "survive" r rounds of the parallel peeling process, a value related to $q^{(r)}(x) \in [0, 1]$ (see Section 2).

hyperedges and a size of 21 for the rest in an otherwise fully random construction yields an average hyperedge size of ≈ 5.03 and a threshold value of ≈ 0.92. For $k \geq 3$, the threshold of the family $(\hat{F}^{k,1}_{n,cn})_{c\in[0,c^*_{k,1}),n\in\mathbb{N}}$ proposed in Theorem 1 (✩) is the 1-*orientability* threshold $c^*_{k,1}$ of $(H^k_{n,cn})_{c\in\mathbb{R}^+_0,n\in\mathbb{N}}$.

1.5 The Technique of Spatial Coupling

The hypergraph F_n arises by "spatial coupling" of $H^k_{n,cn}$ along the "coupling dimension" $X = [0, z+1)$, which roughly means that F_n inherits its *local* structure from $H^k_{n,cn}$ (cf. Facts 12 and 13) but its global structure reflects X. Consider the parallel peeling process on F_n that deletes in each of its rounds all vertices of degree at most ℓ (and incident hyperedges). Vertices with a position close to the borders 0 or $z + 1$ tend to be deleted early on, while many vertices in the denser, central parts remain. But gradually, deletions at the border "expose" vertices further on the inside and the whole hypergraph "erodes" from the outside in. The effect is shown in Figure 3. This does not happen in the more symmetric construction when X is glued into a circle, i.e., if for all $\varepsilon \in [0, 1)$ the positions ε and $z + \varepsilon$ are identified.

The authors of [41, 44] liken the phenomenon to water that is *super-cooled* to below 0℃ in a smooth container. It will not freeze unless a *nucleus* for crystallization is introduced. Once this is done, all water crystallizes quickly, starting from that nucleus. In this sense, the introduction of a nucleus raises the threshold for crystallization.

In our construction, the borders play the role of such a nucleus and raise the peeling threshold to the orientability threshold. Similarly, in coding theory the threshold for decoding with a greedy (belief propagation) decoder is raised to the threshold for decoding with the ideal (maximum *a posteriori* probability) decoder. This effect is known as *threshold saturation*.

We leave a summary of the field to the experts [42, 44]. Put briefly, the phenomenon was discovered in the form of convolutional codes [24], then rigorously explained, first in a special case [43], then more generally [44], later accounting for continuous coupling dimensions (and even multiple dimensions) [42], a form we will exploit in this article.

1.6 Outline

The proof is organized as follows. In Section 2 we idealize the peeling process by switching to a tree-like distributional limit of our hypergraphs, and capture the essential behavior of the process in terms of an operator $\hat{\mathbf{P}}$ acting on functions $q : \mathbb{R} \to [0, 1]$. In Section 3 we analyze the effect of iterated application of $\hat{\mathbf{P}}$ to functions using the rich toolbox from [42]. This is the main ingredient to proving part *(i)* of Theorem 2 in Section 4. The comparatively simple part *(ii)* is independent of these considerations and is proved in Section 5.

Finally, in Section 6 we demonstrate how using our hypergraphs can improve the performance of practical retrieval data structures.

2 The Peeling Process and Idealized Peeling Operators

This section examines how the probabilities for vertices of F_n to "survive" $r \in \mathbb{N}$ rounds of peeling change from one round to the next. In the classical setting, this could be described by a function, mapping the old survival probability to the new one [57]. In our case, however, there are distinct survival probabilities $q(x)$ depending on the position x of the vertex. Thus we need a corresponding operator $\hat{\mathbf{P}}$ that acts on such functions q.

We almost always suppress k, ℓ, c, z in notation outside of definitions. Big-O notation refers to $n \to \infty$ while k, ℓ, c, z are constant.

Consider the parallel peeling process $\mathsf{peel}(F_n, \ell)$ on $F_n = F(n, k, c, z)$. In each *round* of $\mathsf{peel}(F_n)$, all vertices of degree at most ℓ are determined and then deleted simultaneously. Deleting a vertex implicitly deletes all incident hyperedges. We also define the *r-round rooted peeling process* $\mathsf{peel}_{v,r}(F_n, \ell)$ for any vertex $v \in V$ and $r \in \mathbb{N}$. In round $1 \le r' \le r - 1$ of $\mathsf{peel}_{v,r}(F_n)$, only vertices with distance $r - r'$ from v are considered for deletion. Moreover, in round r, the root vertex v is only deleted if it has degree at most $\ell - 1$, not if it has degree ℓ.

For any vertex position $x \in X = [0, z + 1)$ and $r \in \mathbb{N}$ we let $q^{(r)}(x) = q^{(r)}(x, n, k, \ell, c, z)$ be the probability that the vertex $v = \lfloor \frac{x}{z+1} n \rfloor$ survives $\mathsf{peel}_{v,r}(F_n)$, i.e., is not deleted. It is convenient to define $q^{(0)}(x) = 1$ for all $x \in X$, i.e., every vertex survives the "0-round peeling process." To get an intuition for how $q^{(r)}(x)$ evolves with r, consider Figure 3. Even though $q^{(r)}$ is discrete in x by definition, we will later see that it has a continuous limit for $n \to \infty$.

Whether a vertex v at position x survives $\mathsf{peel}_{v,r}$ is a function of its r-neighborhood $F_n(x, r)$, i.e., the subhypergraph of F_n that can be reached from v by traversing at most r hyperedges.

It is natural to consider the distributional limit of $F_n(x, r)$ to get a grip on $q^{(r)}(x)$. In the spirit of the objective method [1], we identify a (possibly infinite) random tree T_x that captures the local characteristics of $F_n(x, r)$ for $n \to \infty$. In the following $\mathrm{Po}(\lambda)$ refers to the Poisson distribution with mean $\lambda \in \mathbb{R}^+$.

Definition 2 (Limiting Tree). Let $k \in \mathbb{N}$, $c, z \in \mathbb{R}^+$, $X = [0, z + 1)$, $Y = [\frac{1}{2}, z + \frac{1}{2})$, and $x \in X$. The random (possibly infinite) hypertree $T_x = T_x(k, c, z)$ is distributed as follows.

T_x has a root vertex $\mathrm{root}(T_x)$ at position x, which for $Y_x := [x - \frac{1}{2}, x + \frac{1}{2}] \cap Y$ has $d_x \sim \mathrm{Po}(ck|Y_x|)$ *child hyperedges* with positions uniformly distributed in Y_x.[8] Each child hyperedge at position y is incident to $k-1$ (fresh) *child vertices* of its own, each with a uniformly random position $x' \in [y - \frac{1}{2}, y + \frac{1}{2}]$. The sub-hypertree at such a child vertex at position x' is distributed recursively (and independently of its sibling-subtrees) according to $T_{x'}$.

For $x \in X$ and $r \in \mathbb{N}$, let $F_n(x, r)$ and $T_x(r)$ denote the r-neighborhoods of vertex $v = \lfloor \frac{x}{z+1} n \rfloor$ in F_n and $\mathrm{root}(T_x)$ in T_x, respectively. In the following, H is an arbitrary fixed rooted hypergraph and equality of hypergraphs indicates a root-preserving isomorphism.

Lemma 4. $\forall x \in X, r \in \mathbb{N}, H : \lim_{n \to \infty} \Pr[F_n(x, r) = H] = \Pr[T_x(r) = H]$.

Sketch of Proof. We construct for fixed r, x and H a random coupling[9] between $F_n(x, r)$ and $T_x(r)$ such that the symmetric difference between the events $\{F_n(x, r) = H\}$ and $\{T_x(r) = H\}$ has probability $o(1)$. We do so inductively, by following a sequence of events. The ith event expresses, firstly, that $F_n(x, r)$ and $T_x(r)$ agree with H concerning the first i rounds of a breadth-first search traversal and, secondly, that corresponding active vertices in $F_n(x, r)$ and $T_x(r)$ have positions with distance $O(1/n)$.

For the first step, consider the root v of $F_n(x, r)$. By construction, any hyperedge containing v must have a position $y \in [x - \frac{1}{2}, x + \frac{1}{2}]$. For $x \in [0, 1)$ or $x \in [z, z+1)$ the potential positions are further restricted by the upper and lower bounds on hyperedge positions, i.e., we have $y \in Y_x := [x - \frac{1}{2}, x + \frac{1}{2}] \cap Y$. In order for a random hyperedge e to contain v, two things have to work out:

(1) The position of e must fall within Y_x. The probability for this is $|Y_x|/|Y| = |Y_x|/z$.
(2) At least one of the k incidences of e must turn out to be to v. The probability for this is $1 - (1 - \frac{z+1}{n})^k$.

With $cn\frac{z}{z+1}$ hyperedges in total, we obtain a binomial distribution $\deg(v) \sim \mathrm{Bin}\left(cn\frac{z}{z+1}, |Y_x|/z(1 - (1 - \frac{z+1}{n})^k)\right)$. This distribution converges, for $n \to \infty$, to $\mathrm{Po}(ck|Y_x|)$, which is the distribution of $\deg(\mathrm{root}(T_x))$. The positions of the neighbors of v are uniformly distributed in the discrete set $|Y_x| \cap (\frac{z+1}{n}\mathbb{Z})$, the positions of the neighbors of $\mathrm{root}(T_x)$ uniformly in the interval $|Y_x|$. It should now be easy to see how a coupling between $F_n(x, 1)$ and $T_x(1)$ could look like.

There are three complications when continuing the argument: *(i)* The discrepancies between vertex positions of $F_n(x, r)$ and $T_x(1)$ need to be kept in check. *(ii)* $F_n(x, r)$ may contain cycles.[10] *(iii)* There are slight dependencies between vertex degrees in $F_n(x, r)$. It should be intuitively plausible that these problems vanish in the limit. We refer to [45] for a full argument showing a similar convergence and to [40] for the related technique of Poissonization. □

We now consider the idealized peeling processes $(\mathsf{peel}_{\mathrm{root}(T_x), r}(T_x))_{x \in X}$. Their survival probabilities are easier to analyze than those of $\mathsf{peel}_{v,r}(F_n)$.

[8] In other words: The positions of the child hyperedges are a Poisson point field on Y_x with intensity ck. By $|I|$ for an interval $I = [a, b]$ we mean $b - a$.
Note also that the position is now a property of a vertex, not an identifying feature. Possibly (though with probability 0) the tree T_x may contain several vertices with the same position.
[9] There is no relation to the term spatial coupling. We refer to the standard technique where several random variables are realized on the same probability space.
[10] This can actually already occur for $r = 1$.

Lemma 5. *Let $r \in \mathbb{N}_0$ be constant and $q_T^{(r)}(x) = q_T^{(r)}(x, k, \ell, c, z)$ be the probability that* $\mathrm{root}(T_x)$ *survives* $\mathsf{peel}_{\mathrm{root}(T_x),r}(T_x, \ell)$ *for $x \in X$. Then for $x \in X$*

$$q_T^{(r+1)}(x) = Q\left(ck \int_{[x-\frac{1}{2},x+\frac{1}{2}]\cap Y} \left(\int_{y-\frac{1}{2}}^{y+\frac{1}{2}} q_T^{(r)}(x')dx'\right)^{k-1} dy,\ \ \ell\right).$$

where $Q(\lambda, \ell) = 1 - \sum_{i<\ell} \frac{\lambda^i}{i!} = \Pr[\mathrm{Po}(\lambda) \geq \ell]$, the latter term slightly abusing notation.

Proof. Let $x \in X$ and $v = \mathrm{root}(T_x)$. Assume $y \in [x-\frac{1}{2}, x+\frac{1}{2}] \cap Y$ is the type of some hyperedge e incident to v. Hyperedge e survives r rounds of $\mathsf{peel}_{v,r+1}(T_x)$ if and only if all of its incident vertices survive these r rounds. Since v itself may only be deleted in round $r+1$, the relevant vertices are the $k-1$ child vertices $w_1, \ldots, w_{k-1}$ with positions uniformly distributed in $[y-\frac{1}{2}, y+\frac{1}{2}]$. Let W_i be the subtree rooted at w_i for $1 \leq i < k$. Consider the peeling process $\mathsf{peel}_{w_i,r}(W_i)$. Assume the process deletes w_i in round r, meaning w_i has degree at most $\ell - 1$ at the start of round r. Then w_i has degree at most ℓ at the start of round r in $\mathsf{peel}_{v,r+1}(T_x)$, meaning $\mathsf{peel}_{v,r+1}(T_x)$ deletes e in round r. Conversely, if none of $\mathsf{peel}_{w_1,r}(W_1), \ldots, \mathsf{peel}_{w_{k-1},r}(W_{k-1})$ delete their root vertex within r rounds, then $w_1, \ldots, w_{k-1}$ have degree at least $\ell + 1$ after round r of $\mathsf{peel}_{v,r+1}(T_x)$ and e survives round r of $\mathsf{peel}_{v,r+1}(T_x)$. Since the position of each w_i is independent and uniformly distributed in $[y-\frac{1}{2}, y+\frac{1}{2})$, the probability for e to survive is $p_y := (\int_{y-\frac{1}{2}}^{y+\frac{1}{2}} q_T^{(r)}(x')dx')^{k-1}$. Since the positions of the hyperedges incident to v are a Poisson point field on $[x-\frac{1}{2}, x+\frac{1}{2}] \cap Y$ with intensity ck, the number of incident hyperedges *surviving* round r of $\mathsf{peel}_{v,r+1}(T_x)$ has Poisson distribution with mean $\lambda := \int_{[x-\frac{1}{2},x+\frac{1}{2}]\cap Y} ckp_y dy$.

The claim now follows by observing that v survives $r+1$ rounds of $\mathsf{peel}_{v,r+1}(T_x)$ if it is incident to at least ℓ hyperedges surviving r rounds. The probability for this is $Q(\lambda, \ell)$. □

For convenience, we define the operator $\mathbf{P} = \mathbf{P}(k, \ell, c, z)$, which maps any (measurable[11]) $q : X \to [0, 1]$ to $\mathbf{P}q : X \to [0, 1]$ with

$$(\mathbf{P}q)(x) = Q\left(ck \int_{[x-\frac{1}{2},x+\frac{1}{2}]\cap Y} \left(\int_{y-\frac{1}{2}}^{y+\frac{1}{2}} q(x')dx'\right)^{k-1} dy,\ \ \ell\right).$$

Together Lemmas 4 and 5 imply that $\mathbf{P}$ can be used to approximate survival probabilities.

Corollary 6. *Let $r \in \mathbb{N}_0$ be constant. Then for all $x \in X$*

$$\mathbf{P}^r q^{(0)}(x) \overset{\text{def}}{=} \mathbf{P}^r q_T^{(0)}(x) \overset{\text{Lem 5}}{=} q_T^{(r)}(x) \overset{\text{Lem 4}}{=} q^{(r)}(x) \pm o(1).$$

To obtain *upper* bounds on survival probabilities, we may remove the awkward restriction "$\cap\, Y$" in the definition of $\mathbf{P}$. We define $\hat{\mathbf{P}} = \hat{\mathbf{P}}(k, \ell, c)$ as mapping any $q : \mathbb{R} \to [0, 1]$ to $\hat{\mathbf{P}}q : \mathbb{R} \to [0, 1]$ with

$$(\hat{\mathbf{P}}q)(x) = Q\left(ck \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} \left(\int_{y-\frac{1}{2}}^{y+\frac{1}{2}} q(x')dx'\right)^{k-1} dy,\ \ \ell\right)$$

Note that $\hat{\mathbf{P}}$ does not depend on z or n. To simplify notation, we assume that the old operator $\mathbf{P}$ also acts on functions $q : \mathbb{R} \to [0, 1]$, ignoring $q(x)$ for $x \notin X$, and producing $\mathbf{P}q : \mathbb{R} \to [0, 1]$ with $\mathbf{P}q(x) = 0$ for $x \notin X$. We also extend $q^{(0)}$ to be $\mathbb{1}[x \in X] : \mathbb{R} \to [0, 1]$, i.e., the characteristic function on X, essentially introducing vertices at positions $x \notin X$ which are, however, already deleted with probability 1 before the first round begins. Note that while $q^{(r)}(x)$ and $q_T^{(r)}(x)$ are by

[11] All functions that play a role in our analysis are measurable. We refrain from pointing this out from now on.

definition non-increasing in r, this is not the case for $(\hat{\mathbf{P}}^r q^{(0)})(x)$. For instance, $\hat{\mathbf{P}}^r q^{(0)}$ has support $(-r, z+1+r)$, which grows with r.[12] The following lemma lists a few easily verified properties of $\hat{\mathbf{P}}$. All inequalities between functions should be interpreted point-wise.

Lemma 7.

(i) $\forall q : \mathbb{R} \to [0,1] : \mathbf{P}q \le \hat{\mathbf{P}}q$.
(ii) $\mathbf{P}$ *and* $\hat{\mathbf{P}}$ *are monotonic, i.e.,* $\forall q, q' : \mathbb{R} \to [0,1] : q \le q' \Rightarrow \mathbf{P}q \le \mathbf{P}q' \wedge \hat{\mathbf{P}}q \le \hat{\mathbf{P}}q'$.
(iii) $\mathbf{P}$ *and* $\hat{\mathbf{P}}$ *are continuous, i.e., pointwise convergence of* $(q_i)_{i\in\mathbb{N}}$ *to* q^* *implies pointwise convergence of* $(\mathbf{P}q_i)_{i\in\mathbb{N}}$ *and* $(\hat{\mathbf{P}}q_i)_{i\in\mathbb{N}}$ *to* $\mathbf{P}q^*$ *and* $\hat{\mathbf{P}}q^*$*, respectively.*

3 Analysis of Iterated Peeling

The goal of this section is to prove the following proposition.

Proposition 8.

(i) *For* $c < c^*_{k,\ell}$ *and any* $z \in \mathbb{R}^+$*, we have* $(\mathbf{P}^r q_0)(x) \xrightarrow{r\to\infty} 0$ *for all* $x \in X$.
(ii) *For* $c > c^*_{k,\ell}$ *and large* z*, we have* $(\mathbf{P}^r q_0)(x) \xrightarrow{r\to\infty} q^*(x)$ *for all* $x \in X$ *and some* $q^* \ne 0$.

The intuition is that for $c > c^*_{k,\ell}$ the peeling process gets stuck, while for $c < c^*_{k,\ell}$ all vertices are eventually peeled. Conveniently, iterations such as the one given by $\mathbf{P}$ and $\hat{\mathbf{P}}$ were extensively studied in a stunning paper by Kudekar et al. [42]. For some initial function $f^{(0)} : \mathbb{R} \to [0,1]$ and non-decreasing functions $h_f, h_g : [0,1] \to [0,1]$ they study the sequence of functions

$$g^{(r)}(y) := h_g((f^{(r)} \otimes \omega)(y)) \qquad f^{(r+1)}(x) := h_f((g^{(r)} \otimes \omega)(x)) \tag{1}$$

where ω is an *averaging kernel*, i.e., an even non-negative function with integral 1 and $\otimes$ is the convolution operator. To apply the theory to our case, we use:

$$h_f(u) := Q(cku, \ell) \qquad h_g(v) := v^{k-1} \qquad \omega(x) = \mathbb{1}[|x| \le \frac{1}{2}]$$

With these substitutions the iteration (1) satisfies $\hat{\mathbf{P}}f^{(r)} = f^{(r+1)}$. If we force the functions $g^{(r)}, r \in \mathbb{N}$, to be zero outside of $Y = [\frac{1}{2}, z+\frac{1}{2})$ by replacing (1) with $g^{(r)}(y) := \min\{\mathbb{1}[y \in Y], h_g((f^{(r)} \otimes \omega)(y))\}$ we get the system with *two-sided termination*. In this case $\mathbf{P}f^{(r)} = f^{(r+1)}$. The system with *one-sided termination* is defined similarly with $Y = [\frac{1}{2}, \infty)$.

We remark that nothing in the following depends on the choice of ω.[13]

3.1 Unleashing Heavy Machinery from Coding Theory

We plan to delegate the proof of Proposition 8 to theorems from [42]. For this, we need to examine the *potential* $\phi(u,v) = \phi(h_f, h_g, u, v)$ given as:

$$\phi(u,v) = \int_0^u h_g^{-1}(u')du' + \int_0^v h_f^{-1}(v')dv' - uv \quad \text{for } 0 \le u \le h_g(1), 0 \le v \le h_f(1).$$

A visualization is given in Figure 4. Consider the equation

$$(u,v) = (h_g(v), h_f(u)). \tag{2}$$

Clearly it has the *trivial solution* $(u,v) = (0,0)$. By monotonicity of h_g and h_f, any two solutions (u_1,v_1) and (u_2,v_2) are component-wise ordered. We write $(u_1,v_1) < (u_2,v_2)$ for $u_1 < u_2 \wedge v_1 < v_2$.

[12] It is still possible to interpret $\hat{\mathbf{P}}^r q^{(0)}(x)$ as survival probabilities in more symmetric, extended versions $\hat{T}_x$ of the tree T_x, but we will not pursue this.

[13] There is a corresponding flexibility in Definition 1. Instead of a hyperedge at position y choosing its incident vertices uniformly at random from $[y-\frac{1}{2}, y+\frac{1}{2}]$, incidences can be chosen according to an almost arbitrary bounded density function that is symmetric around y. For details consider [42, Definition 2].

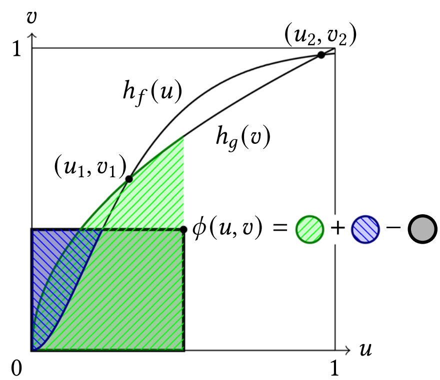

Fig. 4. A plot of the curves $u \mapsto (u, h_f(u))$ and $v \mapsto (h_g(v), v)$ for $u, v \in [0,1]$ with $k = 3$, $\ell = 2$ and $c = c^*_{k,\ell}$. The three crossing points of the curves are the solutions $(0,0)$, (u_1, v_1) and (u_2, v_2) to Equation (2). The potential $\phi(u,v)$ can be visualized as the sum of three areas as shown. The significance of the threshold $c^*_{k,\ell}$ is that the two areas enclosed by the two curves have exactly the same size, or put differently, $\phi(u_2, v_2) = 0$.

Lemma 9.

(i) *Every local minimum (u,v) of ϕ is a solution to Equation (2).*
(ii) *If Equation (2) has at least one non-trivial solution, then the smallest non-trivial solution (u_1, v_1) has potential $\phi(u_1, v_1) > 0$.*
(iii) *Equation (2) has at most two non-trivial solutions.*
(iv) *For $c = c^*_{k,\ell}$ there is a non-trivial solution (u_2, v_2) of Equation (2) with $\phi(u_2, v_2) = 0$. In this case, $(0,0)$ and (u_2, v_2) are the only minima of ϕ.*
(v) *For $c < c^*_{k,\ell}$ we have $\phi(u,v) > 0$ for $(u,v) \neq (0,0)$.*
(vi) *For $c > c^*_{k,\ell}$ Equation (2) has two non-trivial solutions $(u_1, v_1) < (u_2, v_2)$. They satisfy $\phi(u_2, v_2) < \phi(0,0) = 0 < \phi(u_1, v_1)$.*

Proof.

(i) The partial derivatives of ϕ are $\nabla\phi(u,v) = (h_g^{-1}(u) - v, h_f^{-1}(v) - u)$. Therefore, the only candidates for local minima of ϕ are the solutions to Equation (2) (it is easy to check that, except for $(u,v) = (0,0)$, there are no local minima at the borders).
(ii) Assume (u_1, v_1) is the smallest non-trivial solution to Equation (2). Considering Figure 4, we see that $|\phi(u_1, v_1)|$ is the area enclosed by $h_f(u)$ and $h_g^{-1}(u)$ for $u \in [0, u_1]$. To see that the sign of $\phi(u_1, v_1)$ is positive, observe that for small values of u we have $h_f(u) = Q(cku, \ell) = O(u^\ell)$ while $h_g^{-1}(u) = \Omega(u^{1/(k-1)})$ and thus $h_f(\varepsilon) < h_g^{-1}(\varepsilon)$ for $\varepsilon \in (0, u_1)$. This uses $\ell \geq 1$, $k \geq 2$ and $(k, \ell) \neq (2, 1)$.
(iii) By expanding h_f and h_g and substituting $\xi = ckv^{k-1}$ we get for $v \neq 0$:

$$(u,v) = (h_g(v), h_f(u)) \Rightarrow v = Q(ckv^{k-1}, \ell) \Leftrightarrow \tfrac{\xi}{ck} = Q(\xi, \ell)^{k-1} \Leftrightarrow \tfrac{\xi}{Q(\xi,\ell)^{k-1}} = ck.$$

To show that the right-most equation has at most two solutions it suffices to show that $\frac{\xi}{Q(\xi,\ell)^{k-1}}$ has at most one local extremum. If ξ is such an extremum, we get

$$\begin{aligned}\tfrac{d}{d\xi}\tfrac{\xi}{Q(\xi,\ell)^{k-1}} = 0 \Rightarrow\ & Q(\xi,\ell)^{k-1} - \xi(k-1)Q(\xi,\ell)^{k-2}Q'(\xi,\ell) = 0 \\ \Rightarrow\ & Q(\xi,\ell) - \xi(k-1)Q'(\xi,\ell) = 0 \Rightarrow \sum_{i \geq \ell} \tfrac{\xi^i}{i!} - (k-1)\xi^\ell(\ell-1)! = 0\end{aligned}$$

$$\Rightarrow \sum_{i\geq 0} \tfrac{\xi^i}{(i+\ell)!} = (k-1)(\ell-1)!$$

Since the left hand side is increasing in ξ for $\xi > 0$ while the right hand side is constant, there is exactly one solution ξ as claimed.

(iv) Recall that c occurs in the definition of h_f and note that ϕ is monotonically decreasing in c. It is easy to see that ϕ is nowhere negative for small values of c, and negative for some (u, v) if c is large. For continuity reasons and because $\phi(u, v) \geq 0$ for $u, v \in [0, \varepsilon]$ with $\varepsilon = \varepsilon(c)$ small enough (using similar arguments as in (ii)), there must be some intermediate value c where $\phi(u_2, v_2) = 0$ for a local minimum $(u_2, v_2) \neq (0, 0)$ of ϕ. By (i), (u_2, v_2) is a solution of Equation (2). By (ii) there must be a smaller solution (u_1, v_1) with $\phi(u_1, v_1) > 0$. Now by (i), and (iii), there cannot be minima of ϕ in addition to $(0, 0)$ and (u_2, v_2). The only thing left to show is $c = c^*_{k,\ell}$.

We rewrite the potential at (u_2, v_2), using Equation (2)

$$\begin{aligned}\phi(u_2, v_2) &= \int_0^{u_2} h_g^{-1}(u)du + \int_0^{v_2} h_f^{-1}(v)dv - u_2v_2\\ &= \left(u_2v_2 - \int_0^{v_2} h_g(v)dv\right) + \left(u_2v_2 - \int_0^{u_2} h_f(u)du\right) - u_2v_2\\ &= v_2h_g(v_2) - H_g(v_2) - H_f(h_g(v_2)),\end{aligned}$$

where H_g and H_f are antiderivatives of h_g and h_f, i.e.:

$$H_g(v) = \int h_g(v)dv = \tfrac{1}{k}v^k \qquad H_f(u) = \int h_f(u)du = u - \tfrac{1}{ck}\sum_{i=1}^{\ell} Q(cku, i).$$

The fact that $\int_0^\lambda Q(x, \ell)dx = \lambda - \sum_{i=1}^{\ell} Q(\lambda, i)$ can be seen by induction on ℓ. We now examine the implications of $\phi(u_2, v_2) = 0$.

In the following calculation let $\xi := ckv_2^{k-1}$ which implies $Q(\xi, \ell) = v_2$.

$$0 = \phi(u_2, v_2) = v_2h_g(v_2) - H_g(v_2) - H_f(h_g(v_2)) = v_2^k - v_2^k/k - v_2^{k-1} + \tfrac{1}{ck}\sum_{i=1}^{\ell} Q(ckv_2^{k-1}, i)$$

$$\Rightarrow 0 = \xi v_2 - \xi v_2/k - \xi + \sum_{i=1}^{\ell} Q(\xi, i) = \xi Q(\xi, \ell) - \xi Q(\xi, \ell)/k - \xi + \sum_{i=1}^{\ell} Q(\xi, i)$$

$$\Rightarrow \xi Q(\xi, \ell)/k = \xi(Q(\xi, \ell) - 1) + \sum_{i=1}^{\ell} Q(\xi, i) = -\mathrm{e}^{-\xi}\sum_{j=0}^{\ell-1}\frac{\xi^{j+1}}{j!} + \sum_{i=1}^{\ell}\left(1 - \mathrm{e}^{-\xi}\sum_{j=0}^{i-1}\frac{\xi^j}{j!}\right)$$

$$= \ell - e^{-\xi}\sum_{j=0}^{\ell-1}\left(\frac{\xi^{j+1}}{j!} + (\ell - j)\frac{\xi^j}{j!}\right) = \ell - e^{-\xi}\left(\frac{\xi^\ell}{(\ell-1)!} + \sum_{j=0}^{\ell-1}\ell\frac{\xi^j}{j!}\right)$$

$$= \ell - \ell e^{-\xi}\sum_{j=0}^{\ell}\frac{\xi^j}{j!} = \ell Q(\xi, \ell + 1)$$

$$\Rightarrow k\ell = \frac{\xi Q(\xi, \ell)}{Q(\xi, \ell + 1)}.$$

The last equation characterizes the threshold $c^*_{k,\ell}$ for ℓ-orientability of random k-uniform hypergraphs, see for instance [26]. Thus $c = c^*_{k,\ell}$ follows.

(v) We now make the dependence of $\phi_c(u,v)$ on c explicit. For monotonicity reasons we have $\phi_c(u,v) > \phi_{c'}(u,v)$ whenever $c < c'$ and $v \neq 0$. Since $\phi_{c^*_{k,\ell}}$ is positive except for its two roots at $(0,0)$ and (u_2, v_2), for $c < c^*_{k,\ell}$ the potential ϕ_c is positive except at $(0,0)$.

(vi) Since $\phi_{c^*_{k,\ell}}$ has a non-trivial root, ϕ_c attains negative values for monotonicity reasons. By (i), the potential attains its (negative) minimum at a non-trivial solution to Equation (2), and by (ii) it attains a positive value at the smallest non-trivial solution. Thus, the claim follows. □

We are now ready to prove Proposition 8 by recruiting help from [42].

Proof of Proposition 8. First note that we have $q_0 \geq \mathbf{P} q_0$ by definition, which implies $\mathbf{P}^r q_0 \geq \mathbf{P}^{r+1} q_0$ by monotonicity of $\mathbf{P}$ and induction on r. Thus, $\mathbf{P}^r q_0$ is pointwise bounded and decreasing and must converge to a limit q^*. As $\mathbf{P}$ is continuous (see Lemma 7) we have $\mathbf{P} q^* = q^*$.

(i) Let $\mathbb{1} : \mathbb{R} \to \{1\}$ be the 1-function. First note that for any $x \in X$ we have, using properties from Lemma 7 and monotonicity of h_f and h_g

$$(\mathbf{P}^r q_0)(x) \leq (\hat{\mathbf{P}}^r \mathbb{1})(x) = (h_f \circ h_g)^r(1) \xrightarrow{r\to\infty} \max\{u \in [0,1] \mid h_f(h_g(u)) = u\}. \quad (3)$$

So if the only solution of $h_f(h_g(u)) = u$ is $u = 0$, then we get $\mathbf{P}^r q_0(x) \xrightarrow{r\to\infty} 0$ from this alone. Otherwise, by Lemma 9 (iii), there are one or two non-trivial solutions, the larger one we denote by (u_2, v_2).

We now apply [42, Thm. 10][14]. It requires $\phi(u,v) > 0$ for $0 \neq (u,v) \in [0,u_2] \times [0,v_2]$, which we have shown in Lemma 9 (v). The theorem asserts pointwise convergence of $f^{(r)}$ to zero for any $f^{(0)} : \mathbb{R} \to [0, u_2]$ in the case of one-sided termination. Clearly this implies convergence to zero in the case of two-sided termination as well, i.e., $\mathbf{P}^r f^{(0)} \xrightarrow{r\to\infty} 0$. Choosing $f^{(0)} = \mathbb{1} \cdot u_2$ we get

$$\lim_{r\to\infty} (\mathbf{P}^r q_0) = \lim_{r\to\infty} \mathbf{P}^r \lim_{s\to\infty} \mathbf{P}^s q_0 \overset{(3)}{\leq} \lim_{r\to\infty} \mathbf{P}^r f^{(0)} = 0.$$

(ii) Using Lemma 9 (vi) and (iii), we know there are exactly three solutions $(0,0) < (u_1, v_1) < (u_2, v_2)$ to Equation (2) and the signs of their potentials are zero, positive and negative, respectively. This is sufficient to apply [42, Thm. 14].[15] The theorem asserts the existence of a solution $q^* : X \to [0, u_2]$ of $\mathbf{P} q^* = q^*$ with $q^*(\frac{z+1}{2}) = u_2 - \varepsilon$ for any $\varepsilon > 0$, assuming $z = z(\varepsilon)$ is large enough. By monotonicity of $\mathbf{P}$ we have $\lim_{r\to\infty} \mathbf{P}^r q_0 \geq \lim_{r\to\infty} \mathbf{P}^r q^* = q^*$. □

4 Peelability of F_n below $c^*_{k,\ell}$

We now connect the behavior of system (1) to the survival probabilities $q^{(R)}(x)$ we were originally interested in. For $c < c^*_{k,\ell}$ and any $z \in \mathbb{N}$, they can be made smaller than any $\delta > 0$ in $R = R(\delta, k, \ell, z, c)$ rounds.

Lemma 10. *If $c < c^*_{k,\ell}$ then $\forall z \in \mathbb{R}^+, \delta > 0 : \exists R, N \in \mathbb{N} : \forall n \geq N, x \in X : q^{(R)}(x) < \delta$.*

Proof. Let $z \in \mathbb{R}^+$ and $\delta > 0$ be arbitrary constants. At first, Proposition 8 *(i)* implies only pointwise convergence $\mathbf{P}^r q^{(0)}(x) \xrightarrow{r\to\infty} 0$ for all $x \in X$. However, since X is compact, since $\mathbf{P}^r q^{(0)}$ is continuous for $r > 0$ and since the all-zero limit is obviously continuous, basic calculus[16] implies

[14] Strictly speaking, the theorem requires functions h_f and h_g with $h_f(0) = h_g(0) = 0$ and $h_f(1) = h_g(1) = 1$. As the authors of [42] point out themselves, this is purely to simplify notation. We can apply the theorem to our $h_f : [0, u_2] \to [0, v_2]$ and $h_g : [0, v_2] \to [0, u_2]$ with $h_f(0) = h_g(0) = 0$ and $h_f(u_2) = v_2$, $h_g(v_2) = u_2$ after rescaling the axes so (u_2, v_2) becomes $(1,1)$. We will not do so explicitly.

[15] See previous footnote.

[16] Sometimes referred to as Dini's Theorem after Ulisse Dini (1848–1918).

uniform convergence, i.e., there is a constant R such that $\mathbf{P}^R q^{(0)}(x) \leq \delta/2$ for all $x \in X$. Therefore for $x \in X$:

$$q^{(R)}(x) \overset{\text{Cor 6}}{=} (\mathbf{P}^R q^{(0)})(x) + o(1) \leq \delta/2 + o(1) \leq \delta.$$

In the last step, we have simply choosen $N \in \mathbb{N}$ large enough. □

Lemma 4 only allows us to track $q^{(R)}$ *via* $\mathbf{P}^R q_0$ for a *constant* number of rounds R. Therefore, we need to accompany Lemma 10 with the following combinatorial argument that shows that if all but a δ-fraction of the vertices are peeled, then with high probability (whp; meaning probability $1 - o(1)$) the rest is peeled as well. Arguments such as these are standard, many similar ones can be found for instance in [26, 27, 37, 46, 48, 53, 57].

Lemma 11. *Let $c \in [0, \ell]$. There exists $\delta = \delta(k, \ell, z) > 0$ such that, whp, any subhypergraph of $F_n = F(n, k, c, z)$ induced by at most δn vertices has minimum degree at most ℓ.*

Proof. In the course of the proof, we will implicitly encounter positive upper bounds on δ in terms of k, ℓ and z. Any $\delta > 0$ small enough to respect these bounds is suitable. We consider the bad events $W_{s,t}$ that some small set $V' \subseteq [n]_0$ of size s induces t hyperedges for $1 \leq s \leq \delta n$, $\frac{(\ell+1)s}{k} \leq t \leq |E|$. If *none* of these events occur, then all such V' induce less than $(\ell + 1)|V'|/k$ hyperedges and therefore induce hypergraphs with average degree less than $\ell + 1$, so a vertex of degree at most ℓ exists in each of them.

We will show $\Pr[\bigcup_s \bigcup_t W_{s,t}] = O(1/n)$ using a first moment argument. It is convenient to assume that hyperedges are k-tuples, possibly with repetition. First note that F_n then contains three copies of the same hyperedge with probability at most $\binom{m}{3}(\frac{z+1}{n})^{-2k} = O(n^{-2k+3}) = O(n^{-1})$, so we restrict our attention to F_n with at most two copies of the same hyperedge. Given s and t there are $\binom{n}{s}$ ways to choose V'. Since there are s^k ways to form k-tuples from vertices of V' and each hyperedge occurs at most twice, there are at most $\binom{2s^k}{t}$ multisets of hyperedges that V' could induce. The probability that any given k-tuple occurs as a hyperedge is either zero if the k vertices are too far apart or at most $1 - (1 - (\frac{(z+1)}{n})^k)^{czn} \leq \frac{(z+1)^k \ell}{n^{k-1}}$. Similarly, it occurs as a duplicate hyperedge with probability at most $(\frac{(z+1)^k \ell}{n^{k-1}})^2$. Since the presence of hyperedges is negatively correlated, we may obtain an upper bound on the probability of the event that a set of hyperedges are all simultaneously present by taking the product of the events for the presence of the individual hyperedges. Thus, using constants $C, C', C'' \in \mathbb{R}^+$ (that may depend on k, ℓ and z) where precise values do not matter, we get

$$\begin{aligned}
\Pr[\bigcup_{s=1}^{\delta n} \bigcup_{t=(\ell+1)s/k}^{|E|} W_{s,t}] &\leq \sum_{s=1}^{\delta n} \sum_{t=(\ell+1)s/k}^{|E|} \Pr[W_{s,t}] \leq \sum_{s=1}^{\delta n} \sum_{t=(\ell+1)s/k}^{|E|} \binom{n}{s}\binom{2s^k}{t}\left(\frac{(z+1)^k \ell}{n^{k-1}}\right)^t \\
&\leq \sum_{s=1}^{\delta n} \sum_{t=(\ell+1)s/k}^{|E|} \left(\frac{\mathrm{e}n}{s}\right)^s \left(\frac{2\mathrm{e}(z+1)^k \ell s^k}{tn^{k-1}}\right)^t \leq \sum_{s=1}^{\delta n} \sum_{t=(\ell+1)s/k}^{|E|} \left(C\frac{n}{s}\right)^s \left(C'\frac{s^{k-1}}{n^{k-1}}\right)^t \\
&\leq 2\sum_{s=1}^{\delta n} \left(C\frac{n}{s}\right)^s \left(C'\frac{s^{k-1}}{n^{k-1}}\right)^{\lceil (\ell+1)s/k \rceil} = 2\sum_{s=1}^{\delta n} \left(C''\tfrac{s}{n}\right)^{\lceil ((k-1)(\ell+1)-k)\frac{s}{k} \rceil} \leq 2\sum_{s=1}^{\delta n} \left(C''\tfrac{s}{n}\right)^{\lceil \frac{s}{k} \rceil}.
\end{aligned}$$

To get rid of the summation over t, we assumed $(s/n)^{k-1} \leq \delta^{k-1} \leq \frac{1}{2C'}$, in the last step we used $k \geq 2$, $\ell \geq 1$, and $(k, \ell) \neq (2, 1)$. Elementary arguments show that in the resulting bound, the contribution of summands for $s \in \{1, ..., 2k\}$ is of order $O(\frac{1}{n})$, the contribution of the summands with $s \in \{2k + 1, ..., O(\log n)\}$ is of order $O(\frac{\log n}{n^2})$ (using $\frac{s}{n} \leq \frac{\log n}{n}$) and the contribution of the remaining terms with $s \geq 3\log_2 n$ is of order $O(2^{-\log_2 n}) = O(\frac{1}{n})$ (using $C''\frac{s}{n} \leq C''\delta \leq \frac{1}{2}$). This gives $\Pr[\bigcup_{s,t} W_{s,t}] = O(n^{-1})$, which proves the claim. □

We are now ready to prove the first half of Theorem 2.

Proof of Theorem 2 (i). Let $c < c^*_{k,\ell}$ and $z \in \mathbb{R}^+$. We need to show that F_n is ℓ-peelable whp.

First, let $\delta = \delta(k,\ell,z)$ be the constant from Lemma 11 and $R = R(\delta/2)$ as well as N the corresponding constants from Lemma 10.

Assuming $n \geq N$ we have $q^{(R)}(x) \leq \delta/2$ for all $x \in X$, meaning any vertex v from F_n is *not* deleted within R rounds of $\mathsf{peel}_{v,R}(F_n)$ with probability at most $\delta/2$. Since $\mathsf{peel}(F_n)$ deletes in R rounds at least the vertices that any $\mathsf{peel}_{v,R}(F_n)$ for $v \in V$ deletes in R rounds, the expected number of vertices not deleted by $\mathsf{peel}(F_n)$ within R rounds is at most $\delta n/2$.

Now standard arguments using Azuma's inequality (see, e.g., [54, Thm. 13.7]) suffice to conclude that whp at most δn vertices are not deleted by $\mathsf{peel}(F_n)$ within R rounds.

By Lemma 11, whp, neither the remaining δn vertices, nor any subset induces a hypergraph of minimum degree at least $\ell + 1$. Therefore $\mathsf{peel}(F_n)$ deletes all vertices whp. □

5 Non-Orientability of F_n above $c^*_{k,\ell}$

To show that F_n is not *ℓ-peelable* whp for $c > c^*_{k,\ell}$ we argue that F_n is even not *ℓ-orientable* whp.[17] Our proof relies on local weak convergence theory, a subject we danced around in Section 2. There are three ingredients.

Ingredient 1: Identical weak limits. For a finite graph G, let $G(\circ)$ be the random rooted graph obtained by designating a root at random. For a rooted (possibly infinite) graph T, let $T(r)$ be the r-neighborhood of the root.

Definition 3 (Random Weak Limit [46]).[18] Let $(G_n)_{n\in\mathbb{N}}$ be a sequence of (fixed) graphs and T a random (possibly infinite) rooted graph. We say that $(G_n)_{n\in\mathbb{N}}$ has *random weak limit* T if $G_n(\circ)(r)$ converges in distribution to $T(r)$ as $n \to \infty$, for all $r \in \mathbb{N}$.

For example, for $c \in \mathbb{R}^+$, $k \in \mathbb{N}$ and $n \in \mathbb{N}$, let $H_n = H^k_{n,cn}$ be the fully random k-uniform hypergraph. Let G^H_n be the incidence graph of H_n. In particular, G^H_n is bipartite with cn vertices of degree k that correspond to hyperedges in H_n and n vertices (of varying degrees) that correspond to vertices in H_n. Moreover, consider the random (possibly infinite) tree T_{vert} generated as follows. The root vertex is on level zero. A vertex v at an even level is given a random number $X_v \sim \text{Po}(c)$ of children on the next level. A vertex at an odd level is given $k-1$ children on the next level. Let further T_{edge} be the random tree with a root connected to the roots of k independently sampled copies of T_{vert}. Lastly, let T be the random tree obtained by taking a copy of T_{vert} with probability $\frac{1}{1+c}$ and a copy of T_{edge} with probability $\frac{c}{1+c}$.

The following claim is standard[19].

[17] Alternatively, one could try to base a proof on Proposition 8 (ii), possibly by going through similar motions as [57, Lemma 4]. If successful, this might give a detailed characterization of the $(\ell+1)$-*core* of F_n – the largest subhypergraph of F_n with minimum degree $\ell + 1$. Presumably, the $(\ell+1)$-core contains roughly a $q^*(x)$-fraction of the vertices with position roughly at $x \in X$. We leave this aside. Our approach has the upside of establishing a connection between orientability thresholds and peelability thresholds.

[18] The name random weak limit comes from [46]. The notion is also known as Benjamini-Schramm limit [3]. Aldous and Steele [1] call it the *standard construction.*

[19] I cannot find a crystal clear reference for this, but [45, 46] *consider* it to be standard with reference to [40]. Since the language differs significantly, I consider [45] itself to be a better reference, since an arguably more complicated case is treated in detail. The simpler case of Erdős-Renyi random graphs is handled in [6, Thm 3.23].

Fact 12. Almost surely, the sequence $(G_n^H)_{n\in\mathbb{N}}$ has random weak limit T.[20]

Now, let also $z \in \mathbb{R}^+$ and let $F_n = F(n,k,c,z)$ be the random hypergraph from Definition 1. We define $\widetilde{F}_n$ to be a "borderless" version of F_n where the vertices i and $i + \frac{nz}{z+1}$ for all $i \in [\frac{n}{z+1}]_0$ are merged, "glueing" the right-most $\frac{n}{z+1}$ vertices of F_n on top of the left-most $\frac{n}{z+1}$ vertices of F_n. Moreover, let $\widetilde{G}_n^F$ be the incidence graph of $\widetilde{F}_n$.

Techniques from [45] suffice to prove that we get the same random weak limit.

Fact 13. Almost surely, the sequence $(\widetilde{G}_n^F)_{n\in\mathbb{N}}$ has random weak limit T.[21]

Ingredient 2: Lelarge's Theorem [46]. To clarify its role in our proof, we restate a remarkable theorem due to Lelarge [46] in weaker form.

Theorem 14 Theorem 14 (Lelarge [46, Theorem 4.1]). *Let $(G_n = (A_n, B_n, E_n))_{n\in\mathbb{N}}$ be a sequence of bipartite graphs with $|E_n| = \mathcal{O}(|A_n|)$. Let further $M(G_n)$ be the maximum size of a set $E' \subseteq E_n$ with $\deg_{E'}(a) \le 1$ for $a \in A_n$ and $\deg_{E'}(b) \le \ell$ for $b \in B_n$. If $(G_n)_{n\in\mathbb{N}}$ has random weak limit T^* and T^* satisfies certain natural properties.*[22]*, then $\lim_{n\to\infty} \frac{M(G_n)}{|A_n|}$ exists and depends only on T^*.*

A graph G in the theorem should be interpreted as the incidence graph of a hypergraph H with vertex set B and hyperedge set A. Then $M(G)$ is the size of a largest set $A' \subseteq A$ such that the subhypergraph (B, A') of H is ℓ-orientable. In other words, $M(G)$ is the size of the largest *partial ℓ-orientation* of H.

Ingredient 3: Orientability-Gap above the threshold. Assume $c = c^*_{k,\ell} + \varepsilon$ for $\varepsilon > 0$. By definition, it is not the case that H_n is ℓ-orientable whp. More strongly however, it is known [26, 46] that there exists a constant $\delta = \delta(\varepsilon) > 0$ such that the largest partial ℓ-orientation of H_n has size $(1-\delta)cn + o(n)$ whp. In the terms of Theorem 14, this means $\lim_{n\to\infty} M(G_n^H)/|cn| = 1-\delta$ almost surely. We now put all three ingredients together.

Proof of Theorem 2 (ii). Let $c = c^*_{k,\ell} + \varepsilon$ and $\delta = \delta(\varepsilon)$ as above. We pick $z \ge z^* := \frac{2\ell}{\delta c}$.

Since $(G_n^H)_{n\in\mathbb{N}}$ and $(\widetilde{G}_n^F)_{n\in\mathbb{N}}$ almost surely share the random weak limit T by Facts 12 and 13, we conclude from Theorem 14 that the orientability gap carries over from H_n to $\widetilde{F}_n$, i.e., $\lim_{n\to\infty} M(\widetilde{G}_n^F)/m = 1-\delta$ almost surely, where $m = \frac{czn}{z+1}$ is the number of hyperedges in F_n and $\widetilde{F}_n$.

In particular, the size of the largest partial ℓ-orientation of $\widetilde{F}_n$ is $(1-\delta)m + o(n)$ whp. Switching from $\widetilde{F}_n$ back to F_n splits $\frac{n}{z+1}$ vertices and can increase the size of a largest partial ℓ-orientation by at most $\frac{\ell n}{z+1}$ to $(1-\delta+\frac{\ell}{c(z+1)})m + o(n) \le (1-\frac{\delta}{2})m + o(n)$ whp. Thus F_n is not ℓ-orientable whp. □

6 Experiments

To compare our new construction to other families of peelable hypergraphs, we used them to construct 1-bit retrieval data structures as explained in Section 1.1. The following peeling-based variations have been implemented.

BPZ [9]. H is a fully random 3-uniform hypergraph with a hyperedge density below the 1-peelability threshold $c^{\triangle}_{3,1} \approx 0.818$. Construction *via* peeling and eval-operations are very fast, but there is a sizeable overhead of 23%.

[20]It is easy to get confused here because we implicitly "cast" the sequence $(G_n^H)_{n\in\mathbb{N}}$ of random variables on graphs into a sequence of graphs. To reiterate: Having a certain random weak limit is a property of a *sequence of graphs* (not of distributions). The claim is that when sampling a sequence of graphs by independently sampling each element G_n^H of the sequence as explained above, then the sequence of graphs will have the property almost surely, i.e., with probability 1.

[21]Note that the limit T does not depend on z.

[22]The limit T^* must be a bipartite unimodular Galton-Watson tree, see [46] for an explanation. It is clear that $T^* = T$ has the required properties.

Table 2. Overheads and Average Running Times per Key of Various Practical Retrieval Data Structures

Approach Parameters	Overhead	Construct [μs/key]	Eval [ns]
BPZ ($c = 0.81$)	23.46%	0.40	32
LMSS ($D = 12$, $c = 0.90$)	11.12%	0.88	55
LMSS ($D = 150$, $c = 0.99$)	1.06%	0.97	64
LMSS ($D = 800$, $c = 0.997$)	0.56%	1.05	69
COUPLED ($k = 3$, $z = 120$, $c = 0.91$)	11.42%	0.27	33
COUPLED ($k = 4$, $z = 120$, $c = 0.96$)	5.04%	0.30	35
COUPLED ($k = 7$, $z = 120$, $c = 0.98$)	3.00%	0.37	40
GOV ($k = 3$, $c = 0.91$, $C = 10{,}000$)	10.24%	1.74	40
GOV ($k = 4$, $c = 0.97$, $C = 10{,}000$)	3.44%	2.78	37
2-BLOCK ($\ell = 16$, $c = 0.9995$, $C = 20{,}000$)	0.14%	5.03	37
1-BLOCK ($L = 64$, $c = 0.96$, $C = 10{,}000$)	5.43%	0.14	36

LMSS [47]. The hyperedges are distributed such that H is the 1-peelable hypergraph from [47] already mentioned in Section 1.4. To our knowledge, these hypergraphs have not been considered in the context of retrieval.

COUPLED (This Work). The hyperedges are distributed such that $H = F(n, k, c, z)$. Recall that the hyperedge density $\hat{c} = \frac{m}{n}$ is slightly smaller than the chosen parameter c.

Constructions of retrieval data structures not relying on peeling exist [18, 20, 31, 60, 68] and we have implemented three of them. To avoid high construction times for solving non-peelable linear systems they use the standard trick of splitting the input into chunks of small expected size C using an additional hash function. GOV [31] is similar to BPZ in that it uses fully random 3- or 4-uniform hypergraphs, but GOV allows for densities up to $c^*_{3,1}$ or $c^*_{4,1}$. While being more complicated, at least "2-BLOCK" from [18] undercuts the achievable overhead of the peeling-based approaches and at least "1-BLOCK" from [20] offers better construction time due to being particularly cache-efficient. Results are reported in Table 2. The used C++ code has been made available [66] and a detailed discussion is found in the author's thesis [67, Chap. 12].

Experiments were run on a Microsoft Surface Pro 6 with an Intel Core i5-8250U Processor with a maximum single-core frequency of 3.40GHz. In all cases, the data set S contains $m = 10^7$ random 64 bit integers.[23] The function $f : S \to \{0, 1\}$ is the parity of the integer.[24] Hash functions are based on `xxhash` [13] (we thereby depart from the full randomness assumption implicit in our theoretical analysis). Query times are averages obtained by performing eval for all elements of the data set once. The reported numbers are averages over 10 executions. Reported overheads are $1 - N/m$ where $N \geq n$ is the total number of bits used by the data structure, including varying amounts of metadata.

The main takeaway is that our new construction can beat the main other peeling-based approach BPZ in terms of overhead while offering similar running times. It seems likely that these improvements carry over to other data structures relying on peeling as well (see Section 1.1), for instance the space requirement of a construction of minimum perfect hash functions also suggested

[23] When using the first $m = 10^7$ URLs from the `eu-2015-host` dataset gathered by [5] with ≈80 bytes per key, we get similar results, except that running times are uniformly increased due to the higher cost for evaluating hash functions on these large keys.
[24] Note that the number and type of operation performed by construct and eval is completely independent of f. Thus f does not affect our measurements.

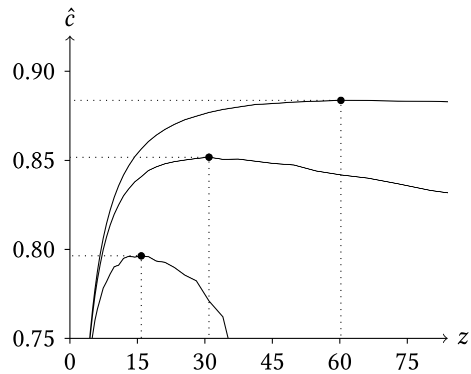

Fig. 5. Approximate hyperedge densities $\hat{c}$ such that $F(n, 3, \hat{c}\frac{z+1}{z}, z)$ is 1-peelable with probability $\frac{1}{2}$, for different choices of z. The three plot lines correspond, from bottom to top, to $n = 10^4$, $n = 10^5$, and $n = 10^6$.

in [9] drops from ≈ 2.44 bits per key to ≈ 2.18 bits per key (see [67, Theorem A3]). With LMSS it is possible to achieve smaller overheads at the expense of higher running times. Note also that the largest hyperedge size $D + 4$ of LMSS is exponential in the average hyperedge size. Therefore, the worst-case running time of eval is much larger than the reported average running time of eval.

On Choosing z. Our theoretical considerations treat z as a constant and offer no guidance in how $z = z(n)$ should be chosen in practice. But consider the following heuristic argument, suggesting that $z = \Theta(n^{1/3})$ may maximize the achievable hyperedge density $\hat{c} = \frac{m}{n} = c\frac{z}{z+1}$.

For finite z and n, two things keep us from achieving $\hat{c} = c^*_{k,1}$. *Firstly*, vertices with positions $x \in [0, 1] \cup [z, z + 1]$ at the borders have on average half the expected degree compared to other vertices. This leads to an overhead of $\varepsilon_1 = c - \hat{c} \approx 1/z$. *Secondly*, we need to choose $c = c^*_{k,1} - \varepsilon_2$ slightly smaller than $c^*_{k,1}$. To see why, imagine the peeling process working linearly through the coupling dimension X. What happens around $x \in X$ should mostly depend on the expected $\Theta(n/z)$ hyperedges with positions close to x. The standard deviation of their number is $\Theta(\sqrt{n/z})$. If the local density is correspondingly increased by $\Omega(\sqrt{n/z}/(n/z)) = \Omega(\sqrt{z/n})$, this should not lead to the local density exceeding $c^*_{k,1}$ (otherwise we might get stuck), which calls for $\varepsilon_2 = c^*_{k,1} - c \geq \Omega(\sqrt{z/n})$. Balancing ε_1 and ε_2 yields our recommendation of $z = \Theta(n^{1/3})$.

Supporting experiments are found in Figure 5. There we estimate, for $n \in \{10^4, 10^5, 10^6\}$ and various z the value $\hat{c}(n, z)$ for which $F(n, 3, \hat{c}(n, z)\frac{z+1}{z}, z)$ is 1-peelable with probability exactly 1/2. To do so, we construct 500 hypergraphs with distribution $F(n, 3, 1, z)$. Let the hyperedge set be $E = \{e_1, ..., e_m\}$. We then determine the values

$$m^* = \max_{m' \in [m]} \{H' = ([n], \{e_1, ..., e_{m'}\}) \text{ is 1-peelable}\}.$$

This can be done by a customized peeling process on H that, whenever no vertex of degree 1 exists, deletes the hyperedge with highest index. Then $m^* + 1$ is the highest index of a hyperedge that was deleted using the special rule. The median of the values $\frac{m^*}{n}$ over the 500 runs is our approximation of $\hat{c}(n, z)$. The z values maximizing $\hat{c}(10^4, z)$, $\hat{c}(10^5, z)$, and $\hat{c}(10^6, z)$ differ by roughly the factor $10^{1/3} \approx 2.15$ as predicted by the argument above. The choice of $z = 120$ for $m = 10^7$ used in Table 2 is extrapolated from these observations.

Note that the results reveal that our construction does not downscale particularly well. For $n = 10^6$ the achieved densities are still more than 3% below the optimum of $c^*_{k,\ell}$ achieved for $n \longrightarrow \infty$.[25]

7 Conclusion

We have constructed families of k-uniform random hypergraphs with i.i.d. random hyperedges and an *ℓ-peelability* threshold that approaches (for $z \longrightarrow \infty$) the *ℓ-orientability* threshold $c^*_{k,\ell}$ of fully random k-uniform hypergraphs.

We conjecture that this is best possible, i.e., no family of k-uniform random hypergraphs with i.i.d. random hyperedges has an ℓ-peelability threshold exceeding $c^*_{k,\ell}$. In fact, even achieving ℓ-orientability beyond $c^*_{k,\ell}$ seems unlikely.

We demonstrated the usefulness of our construction for HBDS using the example of retrieval data structures. The applicability of peelable hypergraphs is much wider, however, and whether using our construction yields significant improvements needs to be explored case by case. For instance, the stronger locality of the hyperedges might turn out to be advantageous in some settings while the higher number of rounds required by the (parallel) peeling process might be a problem in others.

We exploited the phenomenon of "threshold saturation *via* spatial coupling" that was discovered in coding theory and our proof borrows the powerful methods that were developed in that area. We are very pleased to so effortlessly obtain improvements in HBDS and are curious to see whether this connection might be fruitful in other ways as well.

8 Acknowledgments and Closing Remarks

This work is a comprehensive overhaul of [19], presented at ESA 2019. Back then, we were unaware of the general phenomenon of threshold saturation *via* spatial coupling [42–44] (see also [32, 63, 64, 69]). The old construction was similar, but the analysis was more *ad hoc*. The results were weaker, less general, and less elegant.

Shortly after [19] was published, Djamal Belazzougui recognized that we had stumbled upon a known phenomenon and provided very useful pointers.

Konstantinos Panagiotou helped by locating additional relevant sources.

Special thanks go to Martin Dietzfelbinger. He was a coauthor of [19] and closely followed the revision process. His valuable comments significantly improved the presentation of this work.

This article was written for SODA '21. Developments since then are not reflected here. Most notably, the spatial coupling approach was refined into *binary fuse filters* [35] and a new design for retrieval data structures has been developed that arguably outperforms all previous approaches [22].

References

[1] David Aldous and J. Michael Steele. 2004. The objective method: Probabilistic combinatorial optimization and local weak convergence. In *Probability on Discrete Structures*. Kesten Harry (Ed.), Springer, Berlin, 1–72. DOI: https://doi.org/10.1007/978-3-662-09444-0_1

[2] Djamal Belazzougui, Paolo Boldi, Giuseppe Ottaviano, Rossano Venturini, and Sebastiano Vigna. 2014. Cache-oblivious peeling of random hypergraphs. In *Proc. DCC*, 352–361. DOI: https://doi.org/10.1109/DCC.2014.48

[3] Itai Benjamini and Oded Schramm. 2011. Recurrence of distributional limits of finite planar graphs. Springer, New York, 533–545. DOI: https://doi.org/10.1007/978-1-4419-9675-6_15

[4] Burton H. Bloom. 1970. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM* 13, 7 (1970), 422–426. DOI: https://doi.org/10.1145/362686.362692

[5] Paolo Boldi, Andrea Marino, Massimo Santini, and Sebastiano Vigna. 2014. BUbiNG: Massive crawling for the masses. In *Proc. 23rd WWW '14*, 227–228. DOI: https://doi.org/10.1145/2567948.2577304

[25] One trick to mitigate this problem to some degree was suggested by Thomas Mueller Graf (personal communication) and is discussed in [67, Chap. 12.2.5].

[6] Charles Bordenave. 2016. Lecture notes on random graphs and probabilistic combinatorial optimization. Retrieved from https://www.i2m.univ-amu.fr/perso/charles.bordenave/_media/coursrg.pdf

[7] Fabiano Cupertino Botelho. 2008. *Near-Optimal Space Perfect Hashing Algorithms*. Ph.D. Dissertation, Federal University of Minas Gerais. Retrieved from http://cmph.sourceforge.net/papers/thesis.pdf

[8] Fabiano Cupertino Botelho, Rasmus Pagh, and Nivio Ziviani. 2007. Simple and space-efficient minimal perfect hash functions. In *Proc. 10th WADS*, 139–150. DOI: https://doi.org/10.1007/978-3-540-73951-7_13

[9] Fabiano Cupertino Botelho, Rasmus Pagh, and Nivio Ziviani. 2013. Practical perfect hashing in nearly optimal space. *Inf. Syst.* 38, 1 (2013), 108–131. DOI: https://doi.org/10.1016/j.is.2012.06.002

[10] Andrei Z. Broder and Michael Mitzenmacher. 2003. Network applications of bloom filters: A survey. *Internet Math* 1, 4 (2003), 485–509. DOI: https://doi.org/10.1080/15427951.2004.10129096

[11] Julie Anne Cain, Peter Sanders, and Nicholas C. Wormald. 2007. The random graph threshold for k-orientability and a fast algorithm for optimal multiple-choice allocation. In *Proc. 18th SODA*, 469–476. Retrieved from http://dl.acm.org/citation.cfm?id=1283383.1283433

[12] Denis Xavier Charles and Kumar Chellapilla. 2008. Bloomier filters: A second look. In *Proc. 16th ESA*, 259–270. DOI: https://doi.org/10.1007/978-3-540-87744-8_22

[13] Yann Collet. 2020. xxhash—Extremely fast Hash algorithm. Retrieved from https://github.com/Cyan4973/xxHash

[14] Colin Cooper. 2004. The cores of random hypergraphs with a given degree sequence. *Random Struct. Algorithms* 25, 4 (2004), 353–375. DOI: https://doi.org/10.1002/rsa.20040

[15] Martin Dietzfelbinger, Andreas Goerdt, Michael Mitzenmacher, Andrea Montanari, Rasmus Pagh, and Michael Rink. 2010. Tight thresholds for cuckoo hashing via XORSAT. In *Proc. 37th ICALP (1)*, 213–225. DOI: https://doi.org/10.1007/978-3-642-14165-2_19

[16] Martin Dietzfelbinger and Rasmus Pagh. 2008. Succinct data structures for retrieval and approximate membership (extended abstract). In *Proc. 35th ICALP (1)*, 385–396. DOI: https://doi.org/10.1007/978-3-540-70575-8_32

[17] Martin Dietzfelbinger and Michael Rink. 2012. Towards optimal degree-distributions for left-perfect matchings in random bipartite graphs. In *Proc. 7th CSR*, 99–111. DOI: https://doi.org/10.1007/978-3-642-30642-6_11

[18] Martin Dietzfelbinger and Stefan Walzer. 2019. Constant-time retrieval with $o(\log M)$ extra bits. In *Proc. 36th STACS*, 24:1–24:16. DOI: https://doi.org/10.4230/LIPIcs.STACS.2019.24

[19] Martin Dietzfelbinger and Stefan Walzer. 2019. Dense peelable random uniform hypergraphs. In *Proc. 27th ESA*, 38:1–38:16. DOI: https://doi.org/10.4230/LIPIcs.ESA.2019.38

[20] Martin Dietzfelbinger and Stefan Walzer. 2019. Efficient Gauss elimination for near-quadratic matrices with one short random block per row, with applications. In *Proc. 27th ESA*, 39:1–39:18. DOI: https://doi.org/10.4230/LIPIcs.ESA.2019.39

[21] Martin Dietzfelbinger and Christoph Weidling. 2007. Balanced allocation and dictionaries with tightly packed constant size bins. *Theor. Comput. Sci.* 380, 1–2 (2007), 47–68. DOI: https://doi.org/10.1016/j.tcs.2007.02.054

[22] Peter C. Dillinger, Lorenz Hübschle-Schneider, Peter Sanders, and Stefan Walzer. 2022. Fast succinct retrieval and approximate membership using ribbon. In *20th SEA*, 4:1–4:20. DOI: https://doi.org/10.4230/LIPICS.SEA.2022.4

[23] David Eppstein and Michael T. Goodrich. 2011. Straggler identification in round-trip data streams via Newton's identities and invertible bloom filters. *IEEE Trans. Knowl. Data Eng.* 23, 2 (2011), 297–306. DOI: https://doi.org/10.1109/TKDE.2010.132

[24] Alberto Jiménez Feltström and Kamil Sh. Zigangirov. 1999. Time-varying periodic convolutional codes with low-density parity-check matrix. *IEEE Trans. Information Theory* 45, 6 (1999), 2181–2191. DOI: https://doi.org/10.1109/18.782171

[25] Daniel Fernholz and Vijaya Ramachandran. 2007. The k-orientability thresholds for g_n, P. In *Proc. 18th SODA*, 459–468. Retrieved from http://dl.acm.org/citation.cfm?id=1283383.1283432

[26] Nikolaos Fountoulakis, Megha Khosla, and Konstantinos Panagiotou. 2016. The multiple-orientability thresholds for random hypergraphs. *Combinatorics, Probability & Computing* 25, 6 (2016), 870–908. DOI: https://doi.org/10.1017/S0963548315000334

[27] Nikolaos Fountoulakis and Konstantinos Panagiotou. 2012. Sharp load thresholds for cuckoo hashing. *Random Struct. Algorithms* 41, 3 (2012), 306–333. DOI: https://doi.org/10.1002/rsa.20426

[28] Ehud Friedgut and Jean Bourgain. 1999. Sharp thresholds of graph properties, and the k-SAT problem. *J. Am. Math. Soc.* 12, 4 (1999), 1017–1054. Retrieved from http://www.jstor.org/stable/2646096

[29] Alan Frieze and Tony Johansson. 2017. On the insertion time of random walk cuckoo hashing. In *Proc. 28th SODA*, 1497–1502. DOI: https://doi.org/10.1137/1.9781611974782.97

[30] Alan M. Frieze and Páll Melsted. 2012. Maximum matchings in random bipartite graphs and the space utilization of cuckoo Hash tables. *Random Struct. Algorithms* 41, 3 (2012), 334–364. DOI: https://doi.org/10.1002/rsa.20427

[31] Marco Genuzio, Giuseppe Ottaviano, and Sebastiano Vigna. 2016. Fast scalable construction of (minimal perfect Hash) functions. In *Proc. 15th SEA*, 339–352. DOI: https://doi.org/10.1007/978-3-319-38851-9_23

[32] Andrei Giurgiu, Nicolas Macris, and Rüdiger L. Urbanke. 2012. How to prove the maxwell conjecture via spatial coupling - a proof of concept. In *Proc. ISIT*, 458–462. DOI: https://doi.org/10.1109/ISIT.2012.6284230

[33] Michael T. Goodrich and Michael Mitzenmacher. 2011. Invertible bloom lookup tables. In *Proc. 49th Allerton*, 792–799. DOI: https://doi.org/10.1109/Allerton.2011.6120248
[34] Thomas Mueller Graf and Daniel Lemire. 2020. Xor filters. *ACM J. Exp. Algorithmics*, 25 (2020), 1–16. DOI: https://doi.org/10.1145/3376122
[35] Thomas Mueller Graf and Daniel Lemire. 2022. Binary fuse filters: Fast and smaller than xor filters. *ACM J. Exp. Algorithmics* 27 (2022), 1.5:1–1.5:15. DOI: https://doi.org/10.1145/3510449
[36] Seyed Hamed Hassani, Nicolas Macris, and Rüdiger L. Urbanke. 2013. The space of solutions of coupled XORSAT formulae. In *Proc. ISIT*, 2453–2457. DOI: https://doi.org/10.1109/ISIT.2013.6620667
[37] Svante Janson and Malwina J. Luczak. 2007. A simple solution to the k-core problem. *Random Struct. Algorithms* 30, 1–2 (2007), 50–62. DOI: https://doi.org/10.1002/rsa.20147
[38] Megha Khosla. 2013. Balls Into bins made faster. In *Proc. 21st ESA*, 601–612. DOI: https://doi.org/10.1007/978-3-642-40450-4_51
[39] Megha Khosla and Avishek Anand. 2019. A faster algorithm for cuckoo insertion and bipartite matching in large graphs. *Algorithmica* 81, 9 (2019), 3707–3724. DOI: https://doi.org/10.1007/s00453-019-00595-4
[40] Jeong Han Kim. 2006. Poisson cloning model for random graphs. In *Proc. ICM, Vol. III*, 873–898. Retrieved from https://www.mathunion.org/fileadmin/ICM/Proceedings/ICM2006.3/ICM2006.3.ocr.pdf
[41] Florent Krzakala, Marc Mézard, François Sausset, Yifan Sun, and Lenka Zdeborová. 2012. Statistical-physics-based reconstruction in compressed sensing. *Phys. Rev. X* 2 (2012), 18.
[42] S. Kudekar, T. J. Richardson, and R. L. Urbanke. 2015. Wave-like solutions of general 1-D spatially coupled systems. *IEEE Trans. Inf. Theory* 61, 8 (2015), 4117–4157. DOI: https://doi.org/10.1109/TIT.2015.2438870
[43] Shrinivas Kudekar, Tom Richardson, and Rüdiger L. Urbanke. 2010. Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC. In *Proc. ISIT*, 684–688. DOI: https://doi.org/10.1109/ISIT.2010.5513587
[44] Shrinivas Kudekar, Tom Richardson, and Rüdiger L. Urbanke. 2013. Spatially coupled ensembles universally achieve capacity under belief propagation. *IEEE Trans. Information Theory* 59, 12 (2013), 7761–7813. DOI: https://doi.org/10.1109/TIT.2013.2280915
[45] Mathieu Leconte. 2013. Double hashing thresholds via local weak convergence. In *Proc. 51st Allerton*, 131–137. DOI: https://doi.org/10.1109/Allerton.2013.6736515
[46] Marc Lelarge. 2012. A new approach to the orientation of random hypergraphs. In *Proc. 23rd SODA*, 251–264. DOI: https://doi.org/10.1137/1.9781611973099.23
[47] Michael Luby, Michael Mitzenmacher, Mohammad Amin Shokrollahi, and Daniel A. Spielman. 2001. Efficient erasure correcting codes. *IEEE Trans. Inf. Theory* 47, 2 (2001), 569–584. DOI: https://doi.org/10.1109/18.910575
[48] Tomasz Luczak. 1991. Size and connectivity of the k-core of a random graph. *Discrete Mathematics* 91, 1 (1991), 61–68. DOI: https://doi.org/10.1016/0012-365X(91)90162-U
[49] Lailong Luo, Deke Guo, Richard T. B. Ma, Ori Rottenstreich, and Xueshan Luo. 2019. Optimizing bloom filter: Challenges, solutions, and comparisons. *IEEE Communications Surveys and Tutorials* 21, 2 (2019), 1912–1949. DOI: https://doi.org/10.1109/COMST.2018.2889329
[50] Bohdan S. Majewski, Nicholas C. Wormald, George Havas, and Zbigniew J. Czech. 1996. A family of perfect hashing methods. *Comput. J.* 39, 6 (1996), 547–554. DOI: https://doi.org/10.1093/comjnl/39.6.547
[51] Marc Mezard and Andrea Montanari. 2009. *Information, Physics, and Computation.* Oxford University Press, Inc., USA.
[52] Michael Mitzenmacher. 2009. Some open questions related to cuckoo hashing. In *Proc. 17th ESA*, 1–10. DOI: https://doi.org/10.1007/978-3-642-04128-0_1
[53] Michael Mitzenmacher, Konstantinos Panagiotou, and Stefan Walzer. 2018. Load thresholds for cuckoo hashing with double hashing. In *Proc. 16th SWAT*, 29:1–29:9. DOI: https://doi.org/10.4230/LIPIcs.SWAT.2018.29
[54] Michael Mitzenmacher and Eli Upfal. 2017. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis.* Cambridge University Press, New York, NY, USA, 2nd edition.
[55] Michael Mitzenmacher and George Varghese. 2012. Biff (bloom filter) codes: Fast error correction for large data sets. In *Proc. ISIT*, 483–487. DOI: https://doi.org/10.1109/ISIT.2012.6284236
[56] Michael David Mitzenmacher. 1991. *The Power of Two Choices in Randomized Load Balancing*. Ph.D. Dissertation. Harvard University. Retrieved from https://www.eecs.harvard.edu/~michaelm/postscripts/mythesis.pdf
[57] Michael Molloy. 2005. Cores in random hypergraphs and boolean formulas. *Random Struct. Algorithms* 27, 1 (2005), 124–135. DOI: https://doi.org/10.1002/rsa.20061
[58] Rasmus Pagh and Flemming Friche Rodler. 2004. Cuckoo hashing. *J. Algorithms* 51, 2 (2004), 122–144. DOI: https://doi.org/10.1016/j.jalgor.2003.12.002
[59] Boris Pittel, Joel Spencer, and Nicholas C. Wormald. 1996. Sudden emergence of a giant k-core in a random graph. *J. Comb. Theory, Ser. B* 67, 1 (1996), 111–151. DOI: https://doi.org/10.1006/jctb.1996.0036

[60] Ely Porat. 2009. An optimal bloom filter replacement based on matrix solving. In *Proc. 4th CSR*, 263–273. DOI: https://doi.org/10.1007/978-3-642-03351-3_25
[61] Thomas J. Richardson and Rüdiger L. Urbanke. 2008. *Modern Coding Theory*. Cambridge University Press. DOI: https://doi.org/10.1017/cbo9780511791338
[62] Michael Rink. 2013. Mixed hypergraphs for linear-time construction of denser hashing-based data structures. In *Proc. 39th SOFSEM*, 356–368. DOI: https://doi.org/10.1007/978-3-642-35843-2_31
[63] Christian Schlegel and Marat V. Burnashev. 2013. Thresholds of spatially coupled systems via lyapunov's method. In *IEEE Information Theory Workshop*, 1–5. DOI: https://doi.org/10.1109/ITW.2013.6691236
[64] Keigo Takeuchi, Toshiyuki Tanaka, and Tsutomu Kawabata. 2012. A phenomenological study on threshold improvement via spatial coupling. *IEICE Trans.* 95-A, 5 (2012), 974–977. DOI: https://doi.org/10.1587/transfun.E95.A.974
[65] Robert Michael Tanner. 1981. A recursive approach to low complexity codes. *IEEE Trans. Information Theory* 27, 5 (1981), 533–547. DOI: https://doi.org/10.1109/TIT.1981.1056404
[66] Stefan Walzer. 2020. Experimental comparison of retrieval data structures. Retrieved from https://github.com/sekti/retrieval-test
[67] Stefan Walzer. 2020. *Random Hypergraphs for Hashing-Based Data Structures*. Ph.D. Dissertation. Technische Universität Ilmenau, Germany. Retrieved from https://www.db-thueringen.de/receive/dbt_mods_00047127
[68] Sean A. Weaver, Hannah J. Roberts, and Michael J. Smith. 2018. XOR-satisfiability set membership filters. In *Proc. 21st SAT*, 401–418. DOI: https://doi.org/10.1007/978-3-319-94144-8_24
[69] Arvind Yedla, Yung-Yih Jian, Phong S. Nguyen, and Henry D. Pfister. 2012. A simple proof of threshold saturation for coupled scalar recursions. In *Proc. 7th ISTC*, 51–55. DOI: https://doi.org/10.1109/ISTC.2012.6325197

Received 3 March 2021; revised 19 November 2024; accepted 18 December 2024