VaLUH: Fast Algorithms for the Configuration Model of Vertex-Labeled Undirected Hypergraphs

Maryam Abuissa
Brown University
Providence, RI, USA
maryam_abuissa@brown.edu

Matteo Riondato
Amherst College
Amherst, MA, USA
mriondato@amherst.edu

Abstract

We present VaLUH, a suite of Markov-Chain-Monte-Carlo algorithms for uniformly sampling non-degenerate, vertex-labeled, undirected hypergraphs with prescribed vertex degrees and hyperedge sizes (the hypergraph micro-canonical configuration model). One of our methods is based on stub-labeled hypergraphs, one on edge-labeled hypergraphs, and the third directly samples vertex-labeled hypergraphs. We theoretically show that our algorithms require as many or fewer steps to converge to the stationary distribution than existing ones, because they are higher in Peskun's order. We obtain this improvement by carefully defining the state space graph of the Markov chains and by optimizing the transition probabilities, using the Metropolis-Hastings approach. Our experimental evaluation on real networks shows that our methods are up to 6x faster, in number of steps and also in wall-clock time, than existing approaches, as they require less computation per step, with the direct algorithm being the fastest.

CCS Concepts

• **Theory of computation** → **Random walks and Markov chains**; **Random network models**; **Generating random combinatorial structures**; • **Mathematics of computing** → **Random graphs**; **Hypergraphs**; **Metropolis-Hastings algorithm**.

Keywords

Hypothesis Testing, Markov Chain Monte Carlo, Null Model

ACM Reference Format:

Maryam Abuissa and Matteo Riondato. 2026. VaLUH: Fast Algorithms for the Configuration Model of Vertex-Labeled Undirected Hypergraphs. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1 (KDD '26), August 09–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3770854.3780254

KDD Availability Link:

The source code and datasets for this paper has been made publicly available at https://doi.org/10.7910/DVN/QYOOXF.

1 Introduction

Hypergraphs can model more complex (i.e., more than binary) relationships between the actors in a network than dyadic (i.e., "classic", "standard") graphs [6, 9]. They can be used to study, among other applications, the insular vs. bipartisan behavior of political institutions [21, 22], the interactions of proteins [45], the evolving nature of co-authorship [36], and the spread of contagious diseases by modeling contacts between groups of people [10]. Many hypergraph analysis tasks, and algorithms for them, have been proposed in the literature [32, see also Sect. 2].

The real goal of knowledge discovery is not to study the observed dataset, but to obtain new information about the stochastic Data Generation Process (DGP) that produced it: the available data is just a noisy, random, partial representation of the DGP [44].

Statistical hypothesis testing [33] can be used to this end: the results of the analysis of the observed dataset must be compared to those obtained from a *null model*, a mathematical model that encodes all that is currently known or assumed about the DGP. The null model is a collection of all the datasets that may be generated by the DGP, and a probability distribution over this collection. For example, the micro-canonical configuration model for undirected dyadic graphs includes all and only the dyadic graphs with the same degree sequence as the observed one, and assumes each has the same probability of being generated.

By randomly drawing many datasets from the null model, and performing the desired analysis on each of them, one can generate an empirical distribution of the results over the null model, and then compare the results from the observed dataset to this distribution. An empirical p-value, obtained from such comparison, gives evidence about whether the knowledge encoded in the null model explains the observed results, or whether these results may offer new insights about the DGP.

There are two main challenges in the statistically-sound-KDD process we just described. The first is defining the null model so that it includes all and only the datasets that can be generated by the DGP and with the right generation probability. Failure to do so may invalidate the outcome of statistical hypothesis testing, i.e., to wrongly determine that the observed results give new information about the DGP. The added complexity of hypergraphs vs. dyadic graphs calls for particular care in this definition: choosing different mathematical objects to represent hypergraphs (e.g., vertex-labeled vs. edge-labeled vs. stub-labeled hypergraphs, see Sect. 3.1) leads to different null models. The second challenge is algorithmic: computing an accurate empirical p-value requires drawing thousands of random datasets from the null model, so the procedure of sampling such datasets must be as efficient as possible. While this algorithmic problem has been studied extensively on dyadic graphs [20], less work has focused on developing efficient algorithms for sampling from null models for hypergraphs.

KDD '26, Jeju Island, Republic of Korea

Contributions. We study the problem of correctly defining and efficiently sampling from the micro-canonical configuration model for undirected, non-degenerate, vertex-labeled hypergraphs [15], which includes all such hypergraphs with the same degree sequence and the same edge size distribution as an observed one, all with the same probability of being generated. Specifically:

- We give careful definitions of stub-labeled, edge-labeled, and vertex-labeled hypergraphs (Sect. 3.1). These definitions are useful beyond the scope of this work, as they clarify a source of possible confusion in the literature about the mathematical objects contained in null models for hypergraphs.
- We introduce VaLUH, a suite of efficient algorithms to draw samples from the aforementioned null model. VaLUH's methods follow the Markov-Chain-Monte-Carlo (MCMC) approach (Sect. 3.3). VaLUH-smh runs a Markov chain (MC) on the space of stub-labeled hypergraphs. It uses the Metropolis-Hastings (MH) approach to compute the acceptance probabilities so that the stationary distribution of the transformed chain on the space of vertex-labeled hypergraphs is the uniform. VaLUH-e takes a similar approach but runs the MC on the space of edge-labeled hypergraphs, leveraging the well-known mapping between them and bipartite graphs. VaLUH-d samples directly from the space of vertex-labeled hypergraphs, using carefully-tuned transition probabilities for a fast selection of the next state. The use of MH makes our design choice optimal among a large class of algorithms.
- We prove that our algorithms are higher, in Peskun's order [42] than existing algorithms [15], which implies that they have smaller or equal mixing time, converging to the stationary distribution in fewer or as many steps.
- The results of our experimental evaluation on real networks show that VaLUH's algorithms converge up to 6x faster than the state of the art, both in terms of the number of steps and in terms of wall-clock time. Thus, they can draw the desired number of samples from the null model much more quickly than existing algorithms.

2 Related Work

Battiston et al. [6], Bick et al. [9] and Lee et al. [32] survey hypergraphs models, algorithms, and applications, so here we focus on the works most related to ours.

Some works have investigated null models for restricted classes of hypergraphs, e.g., 3-uniform [23], k-uniform and simple [19], or simplicial complexes [17, 53]. Our interest is instead on the more general space of hypergraphs with any fixed vertex degree sequence and fixed hyperedge size distribution.

Another line of work proposes generative models for hypergraphs, and studies their properties, usually in expectation, or in the limit as the network grows [5, 18, 26–29, 31, 46, 47, 49, 52]. Some of these approaches can be used to generate random networks that have, *in expectation*, our desired properties, i.e., to sample (usually non-uniformly, with the goal of maximizing entropy) from the *canonical* configuration model. We focus instead on sampling *uniformly* from the *micro-canonical* configuration model, which only contains hypergraphs that have exactly the desired properties.

Miyashita et al. [39] and Nakajima et al. [40] extend the dK series of null models for dyadic graphs to edge-labeled hypergraphs. We focus instead on vertex-labeled hypergraphs. The mapping from edge-labeled to vertex-labeled hypergraphs that we discuss (Sect. 4.3) can likely be extended to null models in the dK series for vertex-labeled hypergraphs.

Abuissa et al. [2], Bradde and Bianconi [12], Chodrow and Mellor [14], and Preti et al. [43] deal with null models for annotated hypergraphs where vertices can have different roles in hyperedges (e.g., *directed* hypergraphs are annotated hypergraphs, where some of vertices in a hyperedge are the sources of the hyperedge, and the other vertices in the hyperedge are targets). Most of these works focus on edge-labeled annotated hypergraphs, sometimes implicitly, although Abuissa et al. [2] also briefly discuss vertex-labeled directed hypergraphs. The methods introduced in these works are, like ours, MCMC algorithms based on swapping vertices between pairs of hyperedges, but they are not directly applicable to the case of undirected hypergraphs. The efficient approach taken by VaLUH could likely be extended to vertex-labeled annotated hypergraphs, and is a good direction for future work.

Chodrow [15] studies different null models, including the one that is the subject of our work. He gives an MCMC algorithm based on a Markov chain on the space of stub-labeled hypergraphs, and he shows how it can be used to sample uniformly from the space of vertex-labeled hypergraphs by introducing acceptance probabilities that make the transition matrix symmetric. We show in Sect. 4.2 that these acceptance probabilities are suboptimal since our algorithms dominate Chodrow [15]'s in Peskun [42]'s order, i.e., they converge to the stationary distribution in as many or fewer steps (and often much fewer). In Appendix C we also correct some issues in the original analysis by Chodrow [15].

A shared issue in much previous work is that the exact hypergraph space being considered is not defined in a fully formal way, which at times leads to difficulties in comparing different works. One of our goals is to clear up this possible source of confusion. For example, many works use the mapping between hypergraphs and bipartite graphs without mentioning that this mapping is between *edge-labeled* hypergraphs and bipartite graphs (as also hinted by Chodrow [15, Sect. 4.1]).

3 Preliminaries

We now define the core concepts and notation. We use the usual set notation for sets, e.g., $A = \{a, b, c\}$, and a double-brace notation for multisets, e.g., $B = \{\!\{a, b, a, c\}\!\}$. Given a multiset B, we denote with $\mathsf{s}(B)$ the *set* obtained by removing repeated elements in B. For two sets A and B, $A \vartriangle B \doteq (A \cup B) \setminus (A \cap B)$ is their symmetric difference. We use $\mathbb{1}(\text{expr})$ to denote the function that has value 1 when the expression expr is true, and 0 otherwise.

3.1 Hypergraphs

We only deal with undirected hypergraphs, so we will not repeat this adjective any further. We start by defining *Vertex-Labeled* Hypergraphs (VLHs), and then discuss *edge-labeled* and *stub-labeled* hypergraphs in terms of the difference from vertex-labeled ones. Each of these three representations has its own advantages and disadvantages, and different scenarios are usually best modeled with one rather than the others.

A VLH $H \doteq (V, E)$ has a set of vertices $V \doteq \{v_1, \ldots, v_n\}$ and a multiset of hyperedges[1] $E \doteq \{\!\{e_1, \ldots, e_m\}\!\}$ where each $e_i \in E$ is a non-empty subset of V. When E contains multiple copies of the same edge, we say that H contains parallel edges. A hypergraph is said to be *degenerate* when a vertex appears multiple times in an edge, i.e., when elements of E are multisets, rather than sets. Allowing multiple copies of a vertex in a edge generalizes the concept of self-loop from dyadic graphs, but we are unaware of practical settings and applications of hypergraphs where degeneracy is needed or desirable,[2] so we only consider non-degenerate hypergraphs. The *degree* $\deg_H(v)$ of a vertex $v \in V$ in H is the number of edges of H containing v. The *size* $|e|$ of $e \in E$ is its cardinality. For any multiset A, we denote with $\mathrm{m}_A(a)$ the multiplicity of a in A, i.e., the number of times the element a appears in A.

Each vertex of V is uniquely identifiable, i.e., has a unique label, hence the term "vertex-labeled" to denote such hypergraphs. On the other hand, the edges have no label at all: each is defined purely by the set of vertices in it.

VLHs are the most studied representation of hypergraphs, in part because they are most similar to dyadic (multi-)graphs, where the same assumptions about identifiability of vertices and lack thereof for edges usually hold.

Edge-labeled hypergraphs. When unique labels are assigned to edges, or, equivalently, when E is a *sequence* $E \doteq \langle e_1, \ldots, e_m \rangle$ (so the label of an edge is its position in E), we talk about *Edge-Labeled* Hypergraphs (ELH).[3] Defining E as a sequence, rather than a multiset, is the precise difference that makes VLHs and ELHs different mathematical objects. To demonstrate this difference, consider the two ELHs $H = (\{v, w, u\}, \langle \{v, u\}, \{u, w\} \rangle)$ and $H' = (\{v, u, w\}, \langle \{u, w\}, \{v, u\} \rangle)$, which are different, since the order of their edge sequences differ.

Stub-labeled hypergraphs. Stub-labeled graphs were first introduced through the process of "stub-matching" [11] where $\deg_H(v)$ stubs of each vertex v are placed into an urn and edges are drawn by taking two stubs at a time.

Ghoshal et al. [23] and Chodrow [15] extend the concept of of *stub-labeling* to hypergraphs. In a Stub-Labeled Hypergraph (SLH) $H = (V, E)$, E is now a *set* whose elements form a partitioning of the set

$$\Sigma \doteq \bigcup_{v \in V} \left\{ v_1, \ldots, v_{\deg_H(v)} \right\}$$

where the v_i's are uniquely-identifiable copies of the vertex v known as *stubs*. Thus, the edges of H are sets of stubs, and because they form a partitioning of Σ, which is a set, they are all different, thus E is a set. As a result, SLHs are different mathematical objects than both VLHs and ELHs. To demonstrate this difference, consider two SLHs $H = (\{u, v, w\}, \{\{v_1, u_1\}, \{u_2, w_1\}\})$ and $H' = (\{u, v, w\}, \{\{v_1, u_2\}, \{u_1, w_1\}\})$, which are different, since the stubs in the edges differ. If we hid the stub labelings in both sets of edges, we would obtain the same VLH.

Curveball trades. All algorithms we describe make use of the Curveball trade [48, 51], an operation that takes two edges a and b (possibly stub-labeled) and transforms them into two edges c and d such that $|a| = |c|$ and $|b| = |d|$. The "standard" Curveball trade for vertex-labeled edges, used in algorithms that operate on VLHs and ELHs, works as follows:

(1) Let $P \doteq a \triangle b \subseteq V$. Let $q = |a \setminus b|$, and Q be a subset of P of size q chosen uniformly at random among such subsets.
(2) Add to c every vertex $v \in Q$. Add to d every vertex $w \in P \setminus Q$.
(3) Add to both c and d every vertex $v \in a \cap b$.

The stub-labeled version of the Curveball trade, called "pairwise reshuffle" by Chodrow [15, Def 11], operates on two stub-labeled edges a and b as follows [15, Lemma 1]:

(1) Let P, q, and Q as above, but computed on a_{v} and b_{v}.
(2) For every vertex $v \in Q$, add to c the only stub of v in $a \cup b$. For every vertex $w \in P \setminus Q$, add to d the only stub of w in $a \cup b$.
(3) For every vertex $v \in a_{\mathrm{v}} \cap b_{\mathrm{v}}$, select one of the two stubs of v from $a \cup b$ uniformly at random, and add it to c, then add the other stub of v to d.

3.2 Null models

A *null model* $M = (\mathcal{D}, \psi)$ is a representation of the Data Generation Process (DGP): $\mathcal{D}$ is a set of datasets that may be generated, and ψ is a distribution over $\mathcal{D}$ giving the probability of each dataset being generated. The set $\mathcal{D}$ is defined after having observed a dataset $\mathring{D}$:[4] $\mathcal{D}$ contains all and only the datasets that share some set of properties $\mathcal{P}$ with $\mathring{D}$. The goal, in choosing $\mathcal{P}$ is to capture all the assumed or available knowledge about the DGP.

The null model $M = (\mathcal{D}, \psi)$ we study is the popular *microcanonical configuration model for VLHs* [15]:[5] given an observed VLH $\mathcal{D} = \mathring{H} = (V, E)$, $\mathcal{D}$ contains all and only the VLHs $H = (V, E_H)$ such that

(1) H has the same degree sequence as $\mathring{H}$, i.e., for every $v \in V$, $\deg_H(v) = \deg_{\mathring{H}}(v)$; and
(2) H has the same edge sizes as $\mathring{H}$, i.e.,

$$\{\!\{|e| : e \in E_H\}\!\} = \{\!\{|e| : e \in E\}\!\} \enspace .$$

The distribution ψ is uniform over $\mathcal{D}$.

Using a null model. Given a null model, one can then compare a measurement $\mathrm{q}(\mathring{D})$ obtained from $\mathring{D}$ to the distribution of $\mathrm{q}(D)$ where D is sampled from $\mathcal{D}$ according to ψ. By performing a statistical hypothesis tests, we can assess whether the set of properties $\mathcal{P}$ explains the measurements obtained from $\mathring{D}$. The test is performed by computing a p-value for $\mathrm{q}(\mathring{D})$, corresponding to the probability of seeing $\mathrm{q}(D) \geq \mathrm{q}(\mathring{D})$ for D drawn from $\mathcal{D}$ according to ψ. A small p-value gives evidence that $\mathrm{q}(\mathring{D})$ is unlikely to be due to $\mathcal{P}$. Closed formulas for p-values are rarely available [33], so one has to rely

[1] In the rest of the work, we use "edge" to mean "hyperedge". When we (rarely) deal with dyadic graph edges, the context will make it clear.

[2] While dyadic graphs are hypergraphs, and there are certainly applications where allowing self-loops is reasonable, we cannot say the same for generic hypergraphs that are not dyadic.

[3] A more descriptive term would be *vertex-and-edge-labeled* hypergraphs, but since all hypergraphs have vertices with unique identifiers, using just "edge-labeled" should not cause any confusion.

[4] $\mathcal{D}$ depends on $\mathring{D}$, but we do not denote this fact in the notation to keep it light.

[5] In the rest of this paper, we refer to this model as the "VLH configuration model" for short.

on an *empirical* p-value obtained by repeatedly sampling from $\mathcal{D}$ according to ψ and using the distribution of values for q induced by the samples. The key algorithmic challenge is therefore to devise efficient procedures to draw samples from $\mathcal{D}$ according to ψ.

3.3 Markov-Chain-Monte-Carlo

The Markov-Chain-Monte-Carlo (MCMC) method is a generic approach to draw samples from a space $\mathcal{Z}$ according to some distribution ζ [38, Sect. 10.4]. The idea is to run a Markov chain (MC) on $\mathcal{Z}$ whose unique stationary distribution is ζ. By starting at any arbitrary element of $\mathcal{Z}$ and running the MC for a number of steps sufficient to (approximately) achieve stationarity, the reached state is a sample from $\mathcal{Z}$ (approximately) according to ζ. The modeling challenge requires creating such an MC, and the algorithmic one involves designing a MC that reaches stationarity as quickly as possible, where "speed" should be interpreted in two ways:

- from a theoretical point of view, one is interested in the *mixing time*, i.e., the number of *steps* needed for the chain to (approximately) reach stationarity;
- from a practical point of view, one measures the *wall-clock time*, i.e., the number of *seconds* needed to (approximately) reach stationarity.

A narrow focus on minimizing the first measure may lead in designing a chain that requires fewer steps in theory, but where each step is particularly expensive. For intuition, in a race between a mammoth with a long, slow stride, and a rabbit taking short, quick steps, the mammoth may not be the winner. Our goal in this work is designing MCMC methods for sampling from the VLH configuration model that are fast w.r.t. both types of speed.

Peskun's order. Explicit upper bounds to the mixing time of MCs are hard to derive, despite many available techniques [34]. For our purposes, we only need to be able to compare the mixing times of two MCs M_1 and M_2 over $\mathcal{Z}$ with the same stationary distribution ζ. Peskun [42]'s order allows us to perform this comparison.

Definition 3.1 (37, Def. 2.3). Let M_1 and M_2 as above, and let ξ_1 and ξ_2 be their respective transition probability distributions. M_1 *dominates* M_2 *in Peskun's order*, denoted as $\mathsf{M}_1 \succeq_{\mathrm{P}} \mathsf{M}_2$ iff, for every ordered pair $(H, K) \in \mathcal{Z} \times \mathcal{Z}$ such that $H \neq K$, it holds $\xi_1(H, K) \geq \xi_2(H, K)$.

We often abuse the notation, and use $\xi_1 \succeq_{\mathrm{P}} \xi_2$ to mean $\mathsf{M}_1 \succeq_{\mathrm{P}} \mathsf{M}_2$.

Peskun's order is a partial order on the MCs: there may be M_1 and M_2 as above for which neither $\mathsf{M}_1 \succeq_{\mathrm{P}} \mathsf{M}_2$ nor $\mathsf{M}_2 \succeq_{\mathrm{P}} \mathsf{M}_1$. In this case we say that M_1 and M_2 are *Peskun-incomparable* or, abusing notation, that their transition distributions ξ_1 and ξ_2 are Peskun-incomparable. Peskun's order and mixing time are connected.

Lemma 3.2 (37, Thm. 2 (restated)). *If $\mathsf{M}_1 \succeq_{\mathrm{P}} \mathsf{M}_2$, then the mixing time of M_1 is no larger than the mixing time of M_2.*

The Peskun-incomparability of two MCs does not imply that they have the same mixing time, just that the relation between their mixing times cannot be determined through Peskun's order.

Proposal-acceptance approaches. All the MCMC methods we discuss run MCs where the next state Z_{t+1} is chosen by first drawing a "neighbor" Z' of the current state Z_t according to a *proposal distribution* $\phi(Z_t, Z')$, and accepting Z' as Z_{t+1} by flipping a coin with heads bias $\rho(Z_t, Z')$: if the outcome is heads, then $Z_{t+1} = Z'$, otherwise $Z_{t+1} = Z_t$, i.e., the MC does not move. The proposal and acceptance distributions must be such that it is possible for the MC to move, in a sequence of steps, from any element of $\mathcal{Z}$ to any other element of $\mathcal{Z}$, i.e., that the MC is irreducible. The key questions are how to design ϕ and ρ so that the MC is irreducible and has the desired stationary distribution ζ (modeling question) and so that drawing Z' and flipping the coin can be done as quickly as possible (algorithmic question). The *Metropolis-Hastings (MH) approach* [38, Ex. 10.12] is a general recipe for ρ once ϕ has been fixed in such a way that the MC is irreducible, assuming a non-zero acceptance probability. Setting

$$\rho(Z_t, Z') \doteq \min\left\{1, \frac{\zeta(Z')}{\zeta Z_t} \frac{\phi(Z', Z_t)}{\phi(Z_t, Z')}\right\} \tag{1}$$

guarantees that the resulting MC has stationary distribution ζ. When ζ is the uniform distribution of $\mathcal{Z}$, the leftmost ratio on the r.h.s. of Eq. (1) cancels out, and the resulting transition function $\xi(Z_t, Z') \doteq \phi(Z_t, Z')\rho(Z_t, Z')$ is symmetric in its arguments.

Peskun [42, Thm. 2.2.1] shows that the MC with stationary distribution ζ and proposal distribution ϕ taking the MH approach is optimal in Peskun's order, and therefore, through Thm. 3.2 in terms of mixing time, among all MCs with the same stationary and proposal distributions following any proposal-acceptance approach (i.e., using different acceptance probabilities than the MC following MH). For this reason, all VaLUH algorithms take the MH approach.

4 Algorithms for the VLH Configuration Model

We now discuss procedures for sampling from the VLH configuration model.

4.1 Chodrow [15]'s algorithm

To warm up, we start by describing the existing SLH-based algorithm by Chodrow [15, Sect. 4], which we call sc, as some aspects are shared with the other algorithms we discuss. We introduce some technical concepts before delving into the procedure.

Let $S = (V, E_S)$ be any SLH. For any stub-labeled edge $e \in E_S$, let e_{v} be the vertex-labeled edge obtained by replacing the stubs in e with their original vertices in V. Define $\mathsf{s2v}(S) = (V, E)$ as the unique VLH where $E = \{\!\{e_{\mathrm{v}} : e \in E_S\}\!\}$. Clearly, given a VLH $H = (V, E)$, there are multiple SLHs S such that $\mathsf{s2v}(S) = H$. We denote the set of all such SLHs as $\mathsf{s2v}^{-1}(H)$.

Given an observed VLH $\mathring{H} = (V, \mathring{E})$, consider the set $\mathcal{D}$ in the VLH configuration model for $\mathring{H}$, and define the set of SLHs

$$\mathcal{D}_{\mathrm{s}} \doteq \bigcup_{H \in \mathcal{D}} \mathsf{s2v}^{-1}(H) \enspace .$$

sc runs an MC on $\mathcal{D}_{\mathrm{s}}$, which starts at any arbitrary $S_0 \in \mathsf{s2v}^{-1}(\mathring{H})$. When the MC is on state $S_t = (V, E_t)$ at time $t \geq 0$, the next state $S_{t+1} = (V, E_{t+1})$ is chosen as follows. First, the algorithm draws an unordered pair of distinct stub-labeled edges (a, b) from E_t uniformly at random among such pairs. Then, a stub-labeled Curveball trade (see Sect. 3.1) is performed on (a, b) to obtain the stub-labeled edges (c, d). Let $S' = (V, (E_t \setminus \{a, b\}) \cup \{c, d\})$. It holds $|c| = |a|$ and $|d| = |b|$, so $S' \in \mathcal{D}_{\mathrm{s}}$. S' is the candidate proposed for the next state. The probability that the neighbor S' of S_t is proposed as the next state when the MC is in S_t is[6]

$$\phi_{\mathrm{s}}(S_t, S') \doteq \binom{|E_t|}{2}^{-1} 2^{-|a_{\mathrm{v}} \cap b_{\mathrm{v}}|} \binom{|a_{\mathrm{v}} \triangle b_{\mathrm{v}}|}{|a_{\mathrm{v}} \setminus b_{\mathrm{v}}|}^{-1} (1 + \mathbb{1}(|a| = |b|)) \ . \quad (2)$$

Let $H' = (V, E') = \mathsf{s2v}(S')$ and $H_t = \mathsf{s2v}(S_t)$. If $H' = H_t$, the next state of the chain is $S_{t+1} = S'$. Otherwise, SC defines the acceptance probability [15, Eq. 4]

$$\rho_{\mathrm{SC}}(S_t, S') \doteq \frac{2^{-|a_{\mathrm{v}} \cap b_{\mathrm{v}}|}}{\mathsf{m}_{E_t}(a_{\mathrm{v}})\mathsf{m}_{E_t}(b_{\mathrm{v}})}, \quad (3)$$

and flips a coin with this heads bias. If the outcome of the coin flip is heads, then the next state of the MC is $S_{t+1} = S'$. Otherwise, the next state is $S_{t+1} = S_t$, i.e., the MC does not move. The resulting MC over $\mathcal{D}_{\mathrm{s}}$ has stationary distribution (proof in Appendix A)

$$\zeta_{\mathrm{s}}(S) = \frac{1}{\left|\mathsf{s2v}^{-1}(\mathsf{s2v}(S))\right| |\mathcal{D}|}, \quad (4)$$

which in general non-uniform over $\mathcal{D}_{\mathrm{s}}$.

The sequence $(\mathsf{s2v}(S_t))_{t \ge 0}$ is the MC on $\mathcal{D}$ that considers, for each state S_t of MC on $\mathcal{D}_{\mathrm{s}}$, the corresponding VLH $\mathsf{s2v}(S_t)$. Let $H = (V, E_H)$ and $K = (V, E_K)$ be two distinct VLHs in $\mathcal{D}$ for which there is a Curveball trade from H to K involving vertex-labeled edges a and b in E_H. Then the transition probability from H to K is [15, Proof of Thm. 2][7]

$$\xi_{\mathrm{SC}}(H, K) \doteq \binom{|E_H|}{2}^{-1} 2^{-|a \cap b|} \binom{|a \triangle b|}{|a \setminus b|}^{-1} (1 + \mathbb{1}(|a| = |b|)) \ . \quad (5)$$

Chodrow [15, Thm. 2] shows that this probability is symmetric in its arguments, so the MC has uniform stationary distribution, as can also easily be inferred from Eq. (4). We use this probability later when comparing VaLUH's algorithms and SC in terms of Peskun's order and mixing time.

4.2 VaLUH-SMH: a faster SLH-based algorithm

The acceptance probability from Eq. (3) is suboptimal: our algorithm VaLUH-SMH uses the MH approach (see Sect. 3.3) to obtain higher acceptance probabilities, which in turn leads to our MC reaching convergence more quickly. This algorithm is a "gentle introduction" to VaLUH's approach, to show how, with relatively little effort, one can do better than SC. In later sections, we describe even better algorithms for sampling from the VLH configuration model.

VaLUH-SMH proceeds as SC, using the same proposal distribution from Eq. (2), but with acceptance probability

$$\rho_{\mathrm{SMH}}(S, S') \doteq \min\left\{1, \frac{\left|\mathsf{s2v}^{-1}(\mathsf{s2v}(S))\right|}{\left|\mathsf{s2v}^{-1}(\mathsf{s2v}(S'))\right|} \frac{\phi_{\mathrm{s}}(S', S)}{\phi_{\mathrm{s}}(S, S')}\right\} \ . \quad (6)$$

This distribution is obtained using the MH approach (Eq. (1)) with proposal distribution as in Eq. (2) and stationary distribution as in Eq. (4). Thus, the stationary distribution of the MC $(\mathsf{s2v}(S_t))_{t \ge 0}$ on $\mathcal{D}$ is still uniform.

The only aspect left to discuss is how to efficiently compute this acceptance probability. To do so, we first need to study $\left|\mathsf{s2v}^{-1}(\mathsf{s2v}(S))\right|$.

[6]This formula differs slightly from the one given by Chodrow [15, Eq. 3]. As we discuss in Appendix C, there is a small mistake in the original formula.
[7]Corrected to use the proposal probability from Eq. (2).

Lemma 4.1. *Let $H = (V, E)$ be a VLH. It holds*

$$\left|\mathsf{s2v}^{-1}(H)\right| = \frac{\prod_{v \in V} \deg_H(v)!}{\prod_{e \in \mathsf{s}(E)} \mathsf{m}_E(e)!} \ . \quad (7)$$

The proof is in Appendix A. When the hypergraph is a dyadic graph, Eq. (7) matches the result obtained by Fosdick et al. [20, Eq. 3]. With this result, the second term on the r.h.s. of Eq. (6) can be greatly simplified as (derivation in Appendix A)

$$\rho_{\mathrm{SMH}}(S, S') \doteq \min\left\{1, \frac{(\mathsf{m}_E(c) + 1)(\mathsf{m}_E(d) + 1)}{\mathsf{m}_E(a)\mathsf{m}_E(b)}\right\} \ . \quad (8)$$

Thus, $\rho_{\mathrm{SMH}}(S, S')$ can be computed easily by keeping track of the multiplicities of the edges, e.g., in a hash table that is updated (in at most four entries) when the transition to S' is accepted.

Lemma 4.2. *Let t be any time step, and let $W_t = H = (V, E_H)$ be the current state of the MC $(W_t)_{t \ge 0}$ on $\mathcal{D}$ run by VaLUH-SMH. Let $K = (V, E_K) \neq H$ be a VLH in $\mathcal{D}$ for which there is a Curveball trade from H to K transforming edges a and b from E_H into edges c and d in E_K. The probability that $W_{t+1} = K$, i.e., the transition probability from H to K, is*

$$\xi_{\mathrm{SMH}}(H, K) \doteq \binom{|E_H|}{2}^{-1} \binom{|a \triangle b|}{|a \setminus b|}^{-1} (1 + \mathbb{1}(|a| = |b|)) \cdot \min\left\{\mathsf{m}_{E_H}(a)\mathsf{m}_{E_H}(b), (\mathsf{m}_{E_H}(c) + 1)(\mathsf{m}_{E_H}(d) + 1)\right\} \ . \quad (9)$$

The proof is in Appendix A. Equations (5) and (9) yield the following.

Fact 4.3. *For any H and K as above, it holds $\xi_{\mathrm{SMH}}(H, K) \ge \xi_{\mathrm{SC}}(H, K)$.*

I.e., $\xi_{\mathrm{SMH}} \succeq_{\mathrm{P}} \xi_{\mathrm{SC}}$. Theorem 3.2 gives the following result.

Lemma 4.4. *The MC on $\mathcal{D}$ run by VaLUH-SMH reaches the stationary distribution in fewer or as many steps as the one by SC.*

The relation in Fact 4.3 often holds with $>$ or even $\gg$. For H and K as above, if $|a \cap b| \ge 1$ or if $\min\left\{\mathsf{m}_{E_H}(a)\mathsf{m}_{E_H}(b), (\mathsf{m}_{E_H}(c) + 1)(\mathsf{m}_{E_H}(d) + 1)\right\} > 2$, then $\xi_{\mathrm{SMH}}(H, K) > \xi_{\mathrm{SC}}(H, K)$. These conditions are extremely mild, and satisfied (especially the first) by at least some H and K in $\mathcal{D}$ from real-world VLHs. The difference between the two transition probabilities increases as the size of the edge intersection or the multiplicities grow. Although we cannot prove that when the relation holds strictly for all such H and K then the mixing time for VaLUH-SMH is strictly smaller than the one for SC, our experimental results (Sect. 5) show that VaLUH-SMH's MC on $\mathcal{D}$ needs many fewer steps and less wall-clock time than the one by SC to converge to the stationary distribution.

4.3 VaLUH-E: mapping ELHs to bipartite graphs

Many previous works [14, 19, 39–41, 43, 46, 49] mention or leverage the mapping between bipartite graphs and hypergraphs. This mapping is really 1:1 only between ELHs and bipartite graphs, in the sense that given a ELH $H = (V, E)$, there is one and only one bipartite graph $B_H = (V, E, Z)$ where $Z \subseteq V \times E$ is the set of edges of B_H with $(v_j, e_i) \in Z$ iff $v_j \in e_j$. The vertices of H are the "left" vertices of B_H and the edges of H are the "right" vertices of B. Each vertex of B_H is given unique labels/ids, as it is normal for dyadic graphs. This 1:1 mapping does not hold, in general for SLHs and

VLHs. We use this mapping to develop VaLUH-e, our second algorithm to sample from the VLH configuration model. Before delving into the algorithm, we need to introduce some technical concepts.

Let $H = (V, E_H)$ be a VLH. Assume now to sort E_H in decreasing order by edge size,[8] ties broken lexicographically by the ids of the vertices in V, to obtain an ELH $K = (V, E_K)$. Consider now the ELH $J = (V, E_J)$ obtained by arbitrarily permuting the ordering of some of the edges of the same size in E_K (E_J is still sorted in decreasing order by edge size). As long as the permutation did not only involve identical edges, the ELHs K and J are different ELHs, but if we transform their edge sequences into multisets, we obtain H in both cases. Thus, given any ELH $Z = (V, E_Z)$ whose edge sequence E_Z is sorted in decreasing order by edge size, we define $\mathsf{e2v}(Z)$ as the VLH obtained by transforming the sequence E_Z into a multiset, i.e., by removing its ordering. We define $\mathsf{e2v}^{-1}(H)$, for any VLH H, as the set of different ELHs Z with edge sequence sorted in decreasing order according to edge size such that $H = \mathsf{e2v}(Z)$.

Given an observed VLH $\mathring{H} = (V, E_{\mathring{H}})$, and its corresponding $\mathcal{D}$, we can define the set of ELHs

$$\mathcal{D}_{\mathrm{E}} \doteq \bigcup_{H \in \mathcal{D}} \mathsf{e2v}^{-1}(H) \enspace .$$

Through the 1:1 mapping between ELHs and bipartite graphs, we can see $\mathcal{D}_{\mathrm{E}}$ as all and only the bipartite graphs with vertex sets V and E and with fixed degree sequence (as the size of a hyperedge becomes the degree of the vertex corresponding to the hyperedge in the bipartite graph). In turn, by representing each bipartite graph as its unique bi-adjacency matrix, we can see $\mathcal{D}_{\mathrm{E}}$ as all and only binary matrices with $|E|$ rows and $|V|$ columns with fixed row and column sums. Many works [8, 24, 25, 30, 48, 51] discuss how to draw a sample from this space of binary matrices. For example, an MC on $\mathcal{D}_{\mathrm{E}}$ where the next state of the chain is chosen by performing a Curveball trade [48] among two distinct rows of the matrix (i.e., between two distinct edge-labeled hyperedges) chosen uniformly among such pairs of rows, has uniform stationary distribution over $\mathcal{D}_{\mathrm{E}}$ [13]. Our goal though is to sample uniformly from the set $\mathcal{D}$ of VLHs. To achieve this goal, VaLUH-e runs an MC $(H_i)_{i\geq 0}$ on $\mathcal{D}_{\mathrm{E}}$ with stationary distribution

$$\zeta_{\mathrm{E}}(H) \doteq \frac{1}{\left|\mathsf{e2v}^{-1}(\mathsf{e2v}(H))\right||\mathcal{D}|} \enspace . \tag{10}$$

starting from any arbitrary ELH $H_0 \in \mathsf{e2v}^{-1}(\mathring{H})$. By summing $\zeta_{\mathrm{E}}(H)$ over $H \in \mathsf{e2v}^{-1}(K)$ for any VLH K, it is easy to see that the stationary probability for K of the MC $(\mathsf{e2v}^{-1}(H_i))_{i\geq 0}$ is $1/|\mathcal{D}|$, i.e., uniform over $\mathcal{D}$, as desired. A neighbor $H' = (V, E')$ is proposed from the current state $H_t = (V, E_t)$ as follows. First, VaLUH-e draws two distinct edges a and b uniformly from the collection of such unordered pair of edges. Then it performs a Curveball trade between a and b to obtain the edges c and d, and builds E' by replacing a in the sequence E_t with c, and b with d. The proposal probability of H' when the MC is in H_t is

$$\phi_{\mathrm{D}}(H_t, H') \doteq \binom{|E_t|}{2}^{-1} \binom{|a \triangle b|}{|a \setminus b|}^{-1} \enspace . \tag{11}$$

[8] This ordering is chosen w.l.o.g.

VaLUH-e uses the MH approach with proposal as in Eq. (11) and stationary distribution as in Eq. (10). The acceptance probability is

$$\rho_{\mathrm{E}}(H_t, H') \doteq \min\left\{1, \frac{\left|\mathsf{e2v}^{-1}(\mathsf{e2v}(H_t))\right|}{\left|\mathsf{e2v}^{-1}(\mathsf{e2v}(H'))\right|} \frac{\phi_{\mathrm{D}}(H', H_t)}{\phi_{\mathrm{D}}(H_t, H')}\right\} \enspace .$$

As we show in Appendix A, this expression can be simplified to

$$\rho_{\mathrm{E}}(H_t, H') \doteq \min\left\{1, \frac{(\mathsf{m}_{E_t}(c)+1)(\mathsf{m}_{E_t}(d)+1)}{\mathsf{m}_{E_t}(a)\mathsf{m}_{E_t}(b)}\right\}, \tag{12}$$

which, as discussed for Eq. (8), is easy to compute as the MC moves.

Lemma 4.5. *Let t be any time step, and let $W_t = H = (V, E_H)$ be the current state of the MC $(W_t)_{t\geq 0}$ on $\mathcal{D}$ run by VaLUH-e. Let $K = (V, E_K) \neq H$ be a VLH in $\mathcal{D}$ for which there is a Curveball trade from H to K transforming the edges a and b from E_H into edges c and d in E_K. The probability that $W_{t+1} = K$, i.e., the transition probability from H to K, is*

$$\begin{aligned}\xi_E(H,K) \doteq & \binom{|E_H|}{2}^{-1} \binom{|a \triangle b|}{|a \setminus b|}^{-1} (1 + \mathbb{1}(|a| = |b|)) \\ & \cdot \min\left\{\mathsf{m}_{E_H}(a)\mathsf{m}_{E_H}(b), (\mathsf{m}_{E_H}(c)+1)(\mathsf{m}_{E_H}(d)+1)\right\} \enspace .\end{aligned} \tag{13}$$

Fact 4.6. *For any H and K as above, it holds $\xi_{SMH}(H,K) = \xi_E(H,K)$.*

Thus, the MC on stub-labeled hypergraphs and the one on edge-labeled hypergraphs induce the same transition distribution on vertex-labeled hypergraphs, and VaLUH-smh mixes in as many steps as VaLUH-e. Why are we then introducing this algorithm? As discussed in Sect. 3.3, the *wall-clock time* needed to reach the stationary distribution is as, or for practical purposes, more important than the mixing time, which only deals with the number of steps. The results of our experimental evaluation (Sect. 5) show that VaLUH-e converges in less time than VaLUH-smh, and thus should be preferred in practice.

The following is a corollary of Fact 4.6 and Lemma 4.4.

Lemma 4.7. *The MC on $\mathcal{D}$ run by VaLUH-e reaches the stationary distribution in fewer or as many steps as the one by sc.*

Our experimental results (Sect. 5) confirm that VaLUH-e mixes faster than sc both in terms of steps and of wall-clock time.

4.4 VaLUH-d: a direct algorithm for VLHs

The MCs over $\mathcal{D}$ run by the algorithms discussed in the previous sections have the following weaknesses:

- the neighbor proposal probability is highly skewed, because the first step in choosing the neighbor to propose is sampling unordered pairs of edges from the multiset/sequence of edges: there is a much higher probability of sampling two edges with high multiplicities and the resulting Curveball trades and proposed neighbors from such edges;
- The acceptance probability is relatively low, forcing the MC to stay at in the same state at consecutive step because the proposed neighbor was not accepted.

VaLUH-d addresses both issues by:

- proposing any neighbor different than the current state with the same probability; and
- use a high acceptance probability, so the MC is rarely staying in the same state.

VaLUH-D runs an MC directly on $\mathcal{D}$, without going through SLHs or ELHs. The initial state H_0 of the MC is the observed VLH $\mathring{H}$. At step t, when the MC is in state $H_t = (V, E_t) \in \mathcal{D}$, the algorithm samples an unordered pair (a, b) of different edges from the *set* $\mathsf{s}(E_t)$. A Curveball trade (see Sect. 3.1) is then performed on a and b to obtain two edges c and d. The proposed neighbor is $H' = (V, E')$, where $E' = (E_t \setminus \{a, b\}) \cup \{c, d\}$. The probability that H' is proposed when the MC is in H_t is

$$\phi_{\mathrm{D}}(H_t, H') \doteq \binom{|\mathsf{s}(E_t)|}{2}^{-1} \binom{|a \triangle b|}{|a \setminus b|}^{-1} (1 + \mathbb{1}(|a| = |b|)) \enspace .$$

The irreducibility of the MC with this proposal probability follows from the properties of the Curveball trade [13]. VaLUH-D then uses the MH approach (Eq. (1)) with the above proposal probability and the uniform stationary distribution, obtaining the acceptance probability

$$\rho_{\mathrm{D}}(H_t, H') \doteq \min\left\{1, \frac{|\mathsf{s}(E_t)|(|\mathsf{s}(E_t)| - 1)}{|\mathsf{s}(E')|(|\mathsf{s}(E')| - 1)}\right\}, \tag{14}$$

for H' when it is proposed from H_t. It then flips a coin with this head bias. If the outcome of the coin flip is heads, the chain moves to $H_{t+1} = H'$, otherwise it stays in H_t. The use of MH ensures that the resulting stationary distribution of the MC is uniform over $\mathcal{D}$.

By sampling pairs of edges from the set $\mathsf{s}(E_t)$, rather than the multiset E, we ensure that all neighbors different than the current state have the same probability of being proposed. The acceptance probability from Eq. (14) is very close to 1, because $||\mathsf{s}(E_t)| - |\mathsf{s}(E')|| \le 2$, and optimal thanks to the use of MH. Thus, VaLUH-D addresses the weaknesses of previous algorithms.

The transition probability of moving from a VLH $H = (V, E_H)$ to a VLH $K = (V, E_K)$ for which there is a Curveball trade involving two edges a and b from $\mathsf{s}(E_H)$ is

$$\begin{aligned}\xi_{\mathrm{D}}(H, K) &\doteq \phi_{\mathrm{D}}(H, K)\rho_{\mathrm{D}}(H, K) \\ &= \binom{|\mathsf{s}(E)|}{2}^{-1} \binom{|a \triangle b|}{|a \setminus b|}^{-1} (1 + \mathbb{1}(|a| = |b|)) \\ &\quad \cdot \min\left\{1, \frac{|\mathsf{s}(E_H)|(|\mathsf{s}(E_H)| - 1)}{|\mathsf{s}(E_K)|(|\mathsf{s}(E_K)| - 1)}\right\} \enspace .\end{aligned} \tag{15}$$

Comparing Eqs. (5) and (15), we obtain the following result.

Fact 4.8. *For any H and K as above, it holds $\xi_{D}(H, K) \ge \xi_{SC}(H, K)$.*

Thus, $\xi_{\mathrm{D}} \succeq_{\mathrm{P}} \xi_{\mathrm{SC}}$, and Thm. 3.2 yields the following result.

Lemma 4.9. *The MC on $\mathcal{D}$ run by VaLUH-D reaches the stationary distribution in fewer or as many steps as the one by SC.*

The relation in Fact 4.8 often holds strictly. As we mentioned earlier, this fact does not lead to claim a strict improvement of the mixing time, but our experiments show that VaLUH-D reaches convergence much faster than SC, both in number of steps and time.

The MC on $\mathcal{D}$ run by VaLUH-D is Peskun-incomparable to the ones run by VaLUH-SMH and by VaLUH-E, as shown in Appendix A.1. However, our experimental results demonstrate that VaLUH-D is always at least as fast as other algorithms, and often much faster, both in terms of number of steps and wall-clock time.

In terms of space and time complexity, VaLUH and SC only require storing the hypergraph that is the current state of the MC (requring space $O(\sum_{e \in E} |e|)$), and the edge-multiplicity data structure ($O(|E|)$). A Curvball trade takes time $O(|V|)$ [25], and computing the acceptance probabilities and updating the data structures takes amortized time $O(1)$, for all algorithms.

5 Experimental evaluation

We report here the results of our experimental evaluation of VaLUH on real networks. The goal is to assess the behavior of our algorithms in terms of the number of steps and the wall-clock needed for the MCs on $\mathcal{D}$ to reach the stationary distribution, and to compare their behavior to that of SC.

Implementation and environment. We implemented all VaLUH algorithms and SC in Java 21, optimizing each of them as much as possible.[9] The code makes no use of parallelism to perform the Curveball trades [3] and the pairwise reshuffles. The experiments were run on identical machines with an Intel Xeon E5−2680 CPU and 256GB of RAM, running GNU/Linux 6.1.0−37.

Datasets. We use real world hypergraphs with a wide range of sizes, and from a variety of applications, from substances that make up commercial drugs to committees and bills in the house of representatives. We give their salient properties in Appendix B.1. More details about them are in [32, Suppl. Mat. Sect. 2].

5.1 Empirical convergence

Detecting convergence. Before describing the results, we discuss how we empirically detect that the distribution of the state of an MC has converged to the stationary distribution.

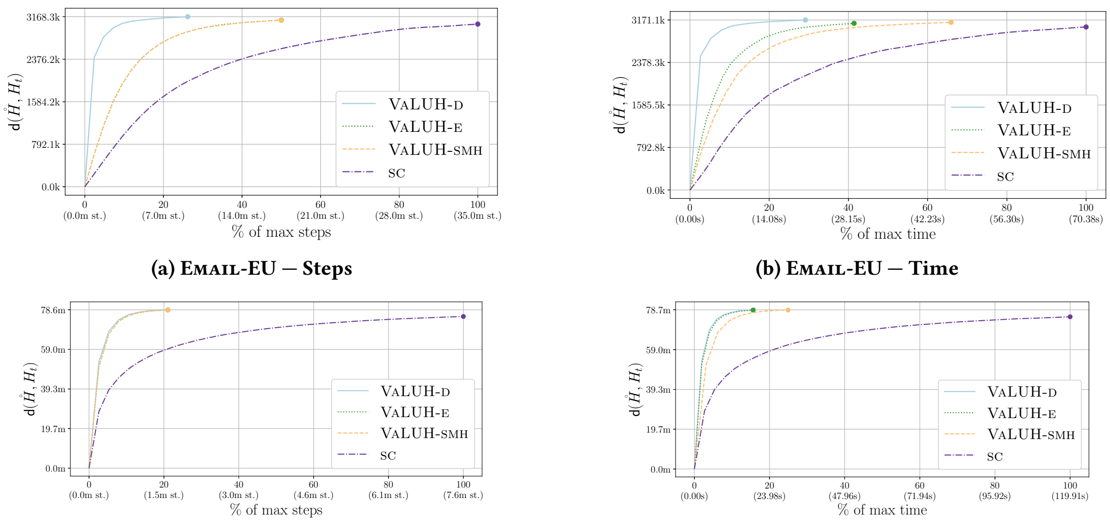
We start by defining a distance measure between VLHs. Given a VLH $H = (V, E)$, let its *clique-expansion* $\mathsf{ce}(H) = (V, E')$ be the dyadic undirected multi-graph obtained by adding to E', for each hyperedge $e \in E$, an edge (u, v) for every unordered pair of distinct vertices in $e \times e$. The adjacency matrix $\mathsf{A}(G)$ for a dyadic multi-graph $G = (V, E)$ is the square $V \times V$ matrix whose (i, j) entry represents the number of edges between i and j in G (i.e., the number of times (i, j) appears in the multiset E). The clique expansion is often used to define similarity measures between hypergraphs [50]. Given two VLHs H and K in $\mathcal{D}$, we define the distance $\mathsf{d}(H, K)$ between H and K as the L_1-norm of the difference between the multi-graph adjacency matrices of their clique expansions, i.e.,

$$\mathsf{d}(H, K) \doteq \|\mathsf{A}(\mathsf{ce}(H)) - \mathsf{A}(\mathsf{ce}(K))\|_1 \enspace . \tag{16}$$

We use this distance measure because common measures used to asses the empirical convergence of MCs such as the perturbation score [48] (used for binary matrices and bipartite graphs), do not generalize well to VLHs, because they are designed to work with matrices. Since the edge set of VLHs is not ordered, there are many vertex-edge matrices representing a VLH, differing by the ordering of the columns, and different orders would lead to drastically different perturbation scores. The perturbation score also does not generalize well to integer-valued matrices, as it does not account for the amount by which two corresponding entries in the matrices differ. The distance measure in Eq. (16) addresses both issues.

For the MCs $(H_t)_{t \ge 0}$ on $\mathcal{D}$ run by the algorithms, the value of $\mathsf{d}(\mathring{H}, H_t)$ should stabilize once the MC reaches convergence. To detect when that happens, we measure the distance $\mathsf{d}(\mathring{H}, H_t)$ from

[9] Code and datasets are available at https://doi.org/10.7910/DVN/QYOOXF.



(a) EMAIL-EU — Steps **(b) EMAIL-EU — Time**

(c) HOUSE-BILLS — Steps **(d) HOUSE-BILLS — Time**

Figure 1: Evolution of $d(\mathring{H}, H_t)$ as function of percentage of max. steps and time. Absolute steps (st) and time (s) in parentheses.

$\mathring{H}$ and the current MC state H_t every k steps for a total of s steps and consider 50 intervals of $\frac{s}{50}$ measurements each. We then compute the slope of the least-squares-fit line for these $\frac{s}{50}$ distance measurements, and say that the MC has reached convergence as soon as this slope is at most a threshold m. This approach takes into consideration the fact that the distance is not necessarily monotonic in t, especially close to convergence. The specific values of these parameters for each dataset are available in Appendix B.1. By measuring the number of steps and the cumulative wall-clock time throughout the run of the chain until convergence, we can then compare the behavior of the algorithms.

Results. Figure 1 shows, for EMAIL-EU and HOUSE-BILLS, the evolution of $d(\mathring{H}, H_t)$ as a function of the percentage of the maximum number of steps (Figs. 1a and 1c) and time (Figs. 1b and 1d), i.e., the steps and time taken by the slowest algorithm. We report the median over five runs, as there was little variance between the runs. Results for other datasets are in Appendix B and they are qualitatively similar. SC is the slowest to converge in both steps and time, on all datasets, confirming our theoretical results about it being Peskun-dominated by all VALUH's algorithms. Table 1 shows the time speedup of VALUH's algorithms w.r.t. SC. VALUH-D is usually the fastest in both steps and time, sometimes tied with the other VALUH algorithms (e.g., in Fig. 1c their three curves are indistinguishable, and in Fig. 1d, the curves of VALUH-D and VALUH-E are indistinguishable) suggesting that, despite the Peskun-incomparability of the MCs in general, on real datasets it may hold $\xi_D \succeq_P \xi_E = \xi_{SMH}$.

VALUH-E and VALUH-SMH converge in the same number of steps (their curves are indistinguishable in Figs. 1a and 1c), as predicted by Fact 4.6. VALUH-SMH is always slower in time (Figs. 1b and 1d). The reason is inherently algorithmic: pairwise reshuffles (done by VALUH-SMH and SC) must randomize the stubs in the intersection of the edges (last point in the description in Sect. 3.1), while Curveball trades (performed by VALUH-E and VALUH-D) do not, saving time. This reason also explains why the decrease in time of VALUH-D and VALUH-E vs. the SMH-based algorithms is often higher than the decrease in number of steps.

Table 1: Time speedup of VALUH algorithms vs. SC.

Dataset	VALUH-D	VALUH-E	VALUH-SMH
SUBSTANCES	3.01x	2.65x	1.89x
EMAIL-EU	3.43x	2.41x	1.52x
HOUSE-BILLS	6.32x	6.40x	4.01x
FOODWEB	1.97x	1.72x	1.08x
GEOMETRY-QUESTIONS	2.91x	3.01x	1.81x
HOUSE-COMMITTEES	3.94x	4.00x	1.93x

6 Conclusions and Future Work

We present VALUH, a suite of MCMC algorithms for uniformly sampling vertex-labeled, non-degenerate, undirected hypergraphs from the set of such hypergraphs with the same degree sequence and edge size distribution as an observed network, i.e., from the microcanonical configuration model. We show that our algorithms dominate, in Peskun's order, existing ones [15], thus VALUH requires at most as many steps to converge to the stationary distribution. The results of our experimental evaluation confirm this speedup, and also show that VALUH algorithms are faster in wall-clock time than the state of the art. Our careful definition of SLHs, ELHs, and VLHs, and our example use of each for different algorithms should be relevant to other studies of algorithms for hypergraphs. Interesting directions for future work include the derivation of explicit bounds to the mixing time of the Markov chains we consider, and extensions of our direct approach to annotated hypergraphs [14] and to more complex null models for undirected hypergraphs.

Acknowledgments

This work is supported in part by the National Science Foundation awards CAREER-2238693 (https://www.nsf.gov/awardsearch/showAward?AWD_ID=2238693) and GRFP-2439559. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the Authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Maryam Abuissa, Alexander Lee, and Matteo Riondato. 2023. ROhAN: Row-order agnostic null models for statistically-sound knowledge discovery. *Data Mining and Knowledge Discovery* 37, 4 (01 Jul 2023), 1692–1718.

[2] Maryam Abuissa, Matteo Riondato, and Eli Upfal. 2025. DiNgHy: Null Models for Non-Degenerate Directed Hypergraphs. In *ECML PKDD'25*.

[3] Daniel Allendorf, Ulrich Meyer, Manuel Penschuck, and Hung Tran. 2023. Parallel global edge switching for the uniform sampling of simple graphs with prescribed degrees. *J. Parallel and Distrib. Comput.* 174 (2023), 118–129.

[4] Ilya Amburg, Nate Veldt, and Austin R. Benson. 2022. Diverse and experienced group discovery via hypergraph clustering. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*. SIAM, 145–153.

[5] Marc Barthelemy. 2022. Class of models for random hypergraphs. *Phys. Rev. E* 106 (Dec 2022), 064310. Issue 6.

[6] Federico Battiston, Giulia Cencetti, Iacopo Iacopini, Vito Latora, Maxime Lucas, Alice Patania, Jean-Gabriel Young, and Giovanni Petri. 2020. Networks beyond pairwise interactions: Structure and dynamics. *Physics reports* 874 (2020), 1–92.

[7] Austin R. Benson, Rediet Abebe, Michael T. Schaub, Ali Jadbabaie, and Jon Kleinberg. 2018. Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences* (2018).

[8] Julian Besag and Peter Clifford. 1989. Generalized Monte Carlo significance tests. *Biometrika* 76, 4 (1989), 633–642.

[9] Christian Bick, Elizabeth Gross, Heather A. Harrington, and Michael T. Schaub. 2023. What Are Higher-Order Networks? *SIAM Rev.* 65, 3 (2023), 686–731.

[10] Jacob Charles Wright Billings, Mirko Hu, Giulia Lerda, Alexey N Medvedev, Francesco Mottes, Adrian Onicas, Andrea Santoro, and Giovanni Petri. 2019. Simplex2vec embeddings for community detection in simplicial complexes. arXiv:1906.09068 [physics.soc-ph]

[11] Béla Bollobás. 1980. A Probabilistic Proof of an Asymptotic Formula for the Number of Labelled Regular Graphs. *European Journal of Combinatorics* 1, 4 (1980), 311–316.

[12] Serena Bradde and Ginestra Bianconi. 2009. The percolation transition in correlated hypergraphs. *Journal of Statistical Mechanics: Theory and Experiment* 2009, 07 (jul 2009), P07028.

[13] Corrie J. Carstens. 2015. Proof of uniform sampling of binary matrices with fixed row sums and column sums for the fast curveball algorithm. *Physical Review E* 91, 4 (2015), 042812.

[14] Philip Chodrow and Andrew Mellor. 2020. Annotated hypergraphs: models and applications. *Applied network science* 5, Article 9 (2020).

[15] Philip S. Chodrow. 2020. Configuration models of random hypergraphs. *Journal of Complex Networks* 8, 3 (2020).

[16] Philip S Chodrow, Nate Veldt, and Austin R Benson. 2021. Hypergraph clustering: from blockmodels to modularity. *Science Advances* (2021).

[17] Owen T. Courtney and Ginestra Bianconi. 2016. Generalized network structures: The configuration model and the canonical ensemble of simplicial complexes. *Phys. Rev. E* 93 (Jun 2016), 062311. Issue 6.

[18] Manh Tuan Do, Se-eun Yoon, Bryan Hooi, and Kijung Shin. 2020. Structural Patterns and Generative Models of Real-world Hypergraphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. 176–186.

[19] Martin Dyer, Catherine Greenhill, Pieter Kleer, James Ross, and Leen Stougie. 2021. Sampling hypergraphs with given degrees. *Discrete Mathematics* 344, 11 (2021), 112566.

[20] Bailey K. Fosdick, Daniel .B Larremore, Joel Nishimura, and Johan Ugander. 2018. Configuring random graph models with fixed degree sequences. *Siam Review* 60, 2 (2018), 315–355.

[21] James H. Fowler. 2006. Connecting the Congress: A Study of Cosponsorship Networks. *Political Analysis* 14, 04 (2006), 456–487.

[22] James H. Fowler. 2006. Legislative cosponsorship networks in the US House and Senate. *Social Networks* 28, 4 (oct 2006), 454–465.

[23] Gourab Ghoshal, Vinko Zlatić, Guido Caldarelli, and M. E. J. Newman. 2009. Random hypergraphs and their applications. *Phys. Rev. E* 79 (Jun 2009), 066118. Issue 6.

[24] Aristides Gionis, Heikki Mannila, Taneli Mielikäinen, and Panayiotis Tsaparas. 2007. Assessing data mining results via swap randomization. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1, 3 (2007), 14.

[25] Karl Godard and Zachary P. Neal. 2022. fastball: A fast algorithm to randomly sample bipartite graphs with fixed degree sequences. *Journal of Complex Networks* 10, 6 (2022), cnac049.

[26] Jin-Li Guo, Xin-Yun Zhu, Qi Suo, and Jeffrey Forrest. 2016. Non-uniform evolving hypergraphs and weighted evolving hypergraphs. *Scientific reports* 6 (2016), 36648.

[27] Lilith Orion Hafner, Chase Holdener, and Nicole Eikmeier. 2022. Functional Ball Dropping: A superfast hypergraph generation scheme. In *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 4606–4615.

[28] Jonas L. Juul, Austin R. Benson, and Jon Kleinberg. 2024. Hypergraph patterns and collaboration structure. *Frontiers in Physics* 11, 1301994 (2024).

[29] Bogumił Kamiński, Valérie Poulin, Paweł Prałat, Przemysław Szufel, and François Théberge. 2019. Clustering via hypergraph modularity. *PLOS ONE* 14, 11 (nov 2019), 1–15.

[30] Ravi Kannan, Prasad Tetali, and Santosh Vempala. 1999. Simple Markov-chain algorithms for generating bipartite graphs and tournaments. *Random Structures & Algorithms* 14, 4 (1999), 293–308.

[31] Sunwoo Kim, Minyoung Choe, Jaemin Yoo, and Kijung Shin. 2023. Reciprocity in directed hypergraphs: measures, findings, and generators. *Data Mining and Knowledge Discovery* 37, 6 (Nov 2023), 2330–2388.

[32] Geon Lee, Fanchen Bu, Tina Eliassi-Rad, and Kijung Shin. 2025. A Survey on Hypergraph Mining: Patterns, Tools, and Generators. *Comput. Surveys* (Feb. 2025).

[33] Erich Leo Lehmann and Joseph P. Romano. 2022. *Testing Statistical Hypotheses* (4 ed.). Springer.

[34] David A. Levin and Yuval Peres. 2017. *Markov chains and mixing times* (2nd ed.). American Mathematical Soc.

[35] Pan Li and Olgica Milenkovic. 2017. Inhomogeneous hypergraph clustering with applications. In *Advances in Neural Information Processing Systems*. 2308–2318.

[36] Qi Luo, Zhenzhen Xie, Yu Liu, Dongxiao Yu, Xiuzhen Cheng, Xuemin Lin, and Xiaohua Jia. 2024. Sampling hypergraphs via joint unbiased random walk. *World Wide Web* 27, 2 (2024), 15.

[37] Antonietta Mira. 2001. Ordering and improving the performance of Monte Carlo Markov chains. *Statist. Sci.* 16, 4 (nov 2001), 340–350.

[38] Michael Mitzenmacher and Eli Upfal. 2005. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press.

[39] Rikuya Miyashita, Kazuki Nakajima, Mei Fukuda, and Kazuyuki Shudo. 2023. Random Hypergraph Model Preserving Two-Mode Clustering Coefficient. In *Big Data Analytics and Knowledge Discovery* 191–196.

[40] Kazuki Nakajima, Kazuyuki Shudo, and Naoki Masuda. 2022. Randomizing Hypergraphs Preserving Degree Correlation and Local Clustering. *IEEE Transactions on Network Science and Engineering* 9, 3 (May 2022), 1139–1153.

[41] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. 2001. Random graphs with arbitrary degree distributions and their applications. *Phys. Rev. E* 64 (Jul 2001), 026118. Issue 2.

[42] P. H. Peskun. 1973. Optimum Monte-Carlo Sampling Using Markov Chains. *Biometrika* 60, 3 (1973), 607–612.

[43] Giulia Preti, Adriano Fazzone, Giovanni Petri, and Gianmarco De Francisci Morales. 2024. Higher-order null models as a lens for social systems. *Physical Review X* 14, 3 (2024), 031032.

[44] Matteo Riondato. 2023. Statistically-sound Knowledge Discovery from Data. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. Society for Industrial and Applied Mathematics, 949–952.

[45] Anna Ritz, Brendan Avent, and T. M. Murali. 2017. Pathway Analysis with Signaling Hypergraphs. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 14, 5 (Sept. 2017), 1042–1055.

[46] Fabio Saracco, Giovanni Petri, Renaud Lambiotte, and Tiziano Squartini. 2024. Entropy-based random models for hypergraphs. arXiv:2207.12123 [cs.SI]

[47] Despina Stasi, Kayvan Sadeghi, Alessandro Rinaldo, Sonja Petrović, and Stephen E. Fienberg. 2014. β models for random hypergraphs with a given degree sequence. In *Proceedings of 21st International Conference on Computational Statistics*. 593–600. arXiv:1407.1004 [math.ST]

[48] Giovanni Strona, Domenico Nappo, Francesco Boccacci, Simone Fattorini, and Jesus San-Miguel-Ayanz. 2014. A fast and unbiased procedure to randomize ecological binary matrices with fixed row and column totals. *Nature communications* 5, 1 (2014), 4114.

[49] Hanlin Sun and Ginestra Bianconi. 2021. Higher-order percolation processes on multiplex hypergraphs. *Phys. Rev. E* 104 (Sep 2021), 034306. Issue 3.

[50] Amit Surana, Can Chen, and Indika Rajapakse. 2023. Hypergraph Similarity Measures. *IEEE Transactions on Network Science and Engineering* 10, 2 (2023), 658–674.

[51] Norman D. Verhelst. 2008. An efficient MCMC algorithm to sample binary matrices with fixed marginals. *Psychometrika* 73, 4 (2008), 705–728.

[52] Jian-Wei Wang, Li-Li Rong, Qiu-Hong Deng, and Ji-Yong Zhang. 2010. Evolving hypernetwork model. *The European Physical Journal B* 77 (2010), 493–498.

[53] Jean-Gabriel Young, Giovanni Petri, Francesco Vaccarino, and Alice Patania. 2017. Construction of and efficient sampling from the simplicial configuration model. *Phys. Rev. E* 96 (Sep 2017), 032312. Issue 3.

A Additional Theoretical Results

Proof of Eq. (4). The MC over $\mathcal{D}_\mathrm{s}$ run by sc has transition probability distribution $\phi_\mathrm{s}(S,S')\rho_\mathrm{sc}(S,S')$. We show that the detailed balance equations hold for this transition probability distribution and the distribution in Eq. (4), i.e., that

$$\zeta_\mathrm{s}(S)\phi_\mathrm{s}(S,S')\rho_\mathrm{sc}(S,S') = \zeta_\mathrm{s}(S')\phi_\mathrm{s}(S',S)\rho_\mathrm{sc}(S',S), \qquad (17)$$

which implies that that distribution in Eq. (4) is the stationary distribution of the MC [38, Thm. 7.10].

Assume that $\phi_\mathrm{s}(S,S') \neq 0$, otherwise Eq. (17) trivially holds, because both sides are 0, as ϕ_s is symmetric in its arguments. Then both sides are different than zero. Consider now the ratio between the sides. Many symmetric terms cancel out, giving

$$\frac{\zeta_\mathrm{s}(S)\rho_\mathrm{sc}(S,S')}{\zeta_\mathrm{s}(S')\rho_\mathrm{sc}(S',S)} = \frac{\left|\mathsf{s2v}^{-1}(\mathsf{s2v}(S'))\right|\mathsf{m}_{E'}(a_v)\mathsf{m}_{E'}(b_v)}{\left|\mathsf{s2v}^{-1}(\mathsf{s2v}(S))\right|\mathsf{m}_{E}(a_v)\mathsf{m}_{E}(b_v)}$$

This value of this expression, from Eq. (7), is 1, thus Eq. (17) holds, and the thesis follows. □

Proof of Lemma 4.1. W.l.o.g., let $V = \{v_1,\dots,v_n\}$, and let $v_{i,j}$ denote the j^{th} stub of vertex v_i, $1 \le j \le \deg_H(v_i)$.

Let $S = (V, E_S) \in \mathsf{s2v}^{-1}(H)$ be any stub-labeled hypergraph in $\mathsf{s2v}^{-1}(H)$. Let now $\vec{\sigma} = (\sigma_1,\dots,\sigma_n)$ be a vector of permutations where σ_i is a permutation of the stubs of vertex $v_i \in V$. Consider the stub-labeled hypergraph $S' = (V, E_{S'})$ obtained by permuting the stubs in the edges of S according to $\vec{\sigma}$. Clearly $S' \in \mathsf{s2v}^{-1}(H)$, as the permutations do not change the topology of the hypergraph. So for every such vector of permutations $\vec{\sigma}$ we obtain some (not necessarily unique) $S' \in \mathsf{s2v}^{-1}(H)$. Let Γ be the set of such vectors of permutations. It holds $|\Gamma| = \prod_{v\in V}\deg_H(v)!$.

When H has parallel edges, some of the vectors of permutations result in S when applied to S. For example, if e appears multiple times in E, and $\vec{\sigma}$ permutes all and only the stubs between an edge of S containing exactly stubs for the vertices in e and another edge of S also containing exactly stubs for the vertices in e, then the stub-labeled hypergraph S' obtained by applying the vector of permutations to S is such that $S' = S$. Multiple copies of e, and multiple edges in E, may be involved in a vector of permutations that result in S when applied to S. If we think of E_S as a sequence, rather than a set, these vectors of permutations can be seen as switching the order of some parallel edges in the sequence, without really changing any edge. Let $\Delta \subseteq \Gamma$ denote the set of such vectors of permutations. It holds $|\Delta| = \prod_{e\in\bar{E}}\mathsf{m}_E(e)!$, because and for each distinct edge $e \in E$, i.e., for each $e \in \bar{E}$, there are $\mathsf{m}_E(e)!$ orderings of the edges parallel to e (including e).

We can partition Γ into equivalence classes labeled by the elements of $\mathsf{s2v}^{-1}(H)$, with the class C_Q, $Q \in \mathsf{s2v}^{-1}(H)$ containing all and only the vectors of permutations that, when applied to S, result in Q. The number of elements in $\mathsf{s2v}^{-1}(H)$ is then equal to the number of classes in this partitioning of Γ.

Given any vector of permutations $\vec{\sigma}$ in C_Q, one can then obtain all other vectors in C_Q by composing $\vec{\sigma}$ with any vector of permutations in Δ, and each such composition results in a different element of C_Q. Thus, all the equivalence classes C_Q, $Q \in \mathsf{s2v}^{-1}(H)$, have the same size $|C_Q| = |\Delta|$. Thus the number of equivalence classes is $|\Gamma|/|\Delta|$, resulting in the expression in Eq. (7). □

Proof of Eq. (8). The proposal distribution function ϕ_s is symmetric in its arguments, as can be seen from Eq. (2), thus the rightmost ratio in Eq. (7) is always 1. Both $H = (V,E) = \mathsf{s2v}(S)$ and $H' = (V,E') = \mathsf{s2v}(S')$ have the same degree sequence, as they both belong to $\mathcal{D}$, so the numerators for $\left|\mathsf{s2v}^{-1}(H)\right|$ and for $\left|\mathsf{s2v}^{-1}(H')\right|$ on the r.h.s. of Eq. (7) are the same. Assume that $H \neq H'$, otherwise $\rho_\mathrm{SMH}(S,S')$ is trivially 1. We can then rewrite the second term on the r.h.s. of Eq. (6) as

$$\frac{\left|\mathsf{s2v}^{-1}(H)\right|}{\left|\mathsf{s2v}^{-1}(H')\right|}\frac{\phi_\mathrm{s}(S',S)}{\phi_\mathrm{s}(S,S')} = \frac{\prod_{e\in\mathsf{s}(E')}\mathsf{m}_{E'}(e)!}{\prod_{e\in\mathsf{s}(E)}\mathsf{m}_{E}(e)!}, \qquad (18)$$

where $\mathsf{s}(E')$ and $\mathsf{s}(E)$ are the *sets* of elements in E' and E respectively. From $H \neq H'$, and from the properties of the pairwise reshuffle, it holds $E \setminus E' = \{a,b\}$ and $E' \setminus E = \{c,d\}$, where these four vertex-labeled edges are all different. We then have $\mathsf{s}(E') \setminus \{a,b,c,d\} = \mathsf{s}(E) \setminus \{a,b,c,d\}$, and

$$\begin{aligned}
&\mathsf{m}_{E'}(a) = \mathsf{m}_E(a) - 1, \quad \mathsf{m}_{E'}(b) = \mathsf{m}_E(b) - 1,\\
&\mathsf{m}_{E'}(c) = \mathsf{m}_E(c) + 1, \quad \mathsf{m}_{E'}(d) = \mathsf{m}_E(d) + 1, \text{ and}\\
&\mathsf{m}_{E'}(e) = \mathsf{m}_E(e) \text{ for any } e \in (\mathsf{s}(E') \cup \mathsf{s}(E)) \setminus \{a,b,c,d\} \ .
\end{aligned}$$

Equation (18) can then be rewritten as

$$\begin{aligned}
&\frac{(\mathsf{m}_E(a)-1)!(\mathsf{m}_E(b)-1)!(\mathsf{m}_E(c)+1)!(\mathsf{m}_E(d)+1)!}{\mathsf{m}_E(a)!\mathsf{m}_E(b)!\mathsf{m}_E(c)!\mathsf{m}_E(d)!}\\
&= \frac{(\mathsf{m}_E(c)+1)(\mathsf{m}_E(d)+1)}{\mathsf{m}_E(a)\mathsf{m}_E(b)},
\end{aligned}$$

which is the expression in Eq. (8). □

Proving Lemma 4.2. Before proving Lemma 4.2 we need the following technical fact, shown in [15, Proof of Thm. 2].

Fact A.1. *Let $H = (V,E_H)$ and $K = (V,E_K)$ be two different VLHs for which there is a Curveball trade from H to K involving edges a and b in E_H. Let $S_H = (V, E_{S_H})$ be any SLH in $\mathsf{s2v}^{-1}(H)$. Let $\mathsf{n}(S_H,K)$ be the set of SLHs from $\mathsf{s2v}^{-1}(K)$ that can be reached by performing a pairwise reshuffle from S_H. It holds*

$$|\mathsf{n}(S_H,K)| = \mathsf{m}_{E_H}(a)\mathsf{m}_{E_K}(b)2^{|a\cap b|} \ . \qquad (19)$$

Proof of Lemma 4.2. When $W_t = H$, the underlying chain $(S_t)_{t\ge 0}$ may be in any of the SLHs from $\mathsf{s2v}^{-1}(H)$. Using the law of total probability, we can then write

$$\xi_\mathrm{SMH}(H,K) = \sum_{S_H\in\mathsf{s2v}^{-1}(H)}\left(\Pr(S_t = S_H)\sum_{S_K\in\mathsf{n}(S_H,K)}\phi_\mathrm{s}(S_H,S_K)\rho_\mathrm{SMH}(S_H,S_K)\right) \ .$$

For a fixed S_H, the value of $\phi_\mathrm{s}(S_H,S_K)\rho_\mathrm{SMH}(S_H,S_K)$ is the same for any $S_K \in \mathsf{n}(S_H,K)$. Let $Y_{S_H,K}$ be any of these. We can rewrite the r.h.s. above as

$$\sum_{S_H\in\mathsf{s2v}^{-1}(H)}\left(\Pr(S_t = S_H)\phi_\mathrm{s}(S_H,Y_{S_H,K})\rho_\mathrm{SMH}(S_H,Y_{S_H,K})|\mathsf{n}(S_H,K)|\right) \ .$$

Additionally, $\phi_\mathrm{s}(S_H,Y_{S_H,K})$, $\rho_\mathrm{SMH}(S_H,Y_{S_H,K})$ and $|\mathsf{n}(S_H,K)|$ are the same for every $S_H \in \mathsf{s2v}^{-1}(H)$. Let X_H be any of these. We can rewrite the last expression as

$$\phi_\mathrm{s}(X_H,Y_{X_H,K})\rho_\mathrm{SMH}(X_H,Y_{X_H,K})|\mathsf{n}(X_H,K)|\sum_{S_H\in\mathsf{s2v}^{-1}(H)}\Pr(S_t = S_H),$$

and the sum on the right simplifies to 1.

By plugging in the expressions from Eqs. (2), (6) and (19), we obtain the r.h.s. of Eq. (9). □

Derivation of Eq. (12). Since the proposal probability in Eq. (11) is symmetric in its arguments, the rightmost ratio cancels out. We need only to study $\left|\mathsf{e2v}^{-1}(K)\right|$ for a VLH K, in order to efficiently compute the acceptance probability. The aforementioned mapping of $\mathcal{D}_{\mathrm{E}}$ to the space of binary matrices with fixed rows and column sums allows us to leverage a result originally developed in that setting.

Lemma A.2 (1, Lemma 3 (restated)). *Let $K = (V, E)$ be a VLH. Let $T_i \doteq \{\!\{e \in E : |e| = i\}\!\}$ be the multiset of edges of size i, for $1 \leq i \leq \max_{e \in E}|e|$. Then, the number $\left|\mathsf{e2v}^{-1}(K)\right|$ of ELHs mapped to K by* e2v *is*

$$\left|\mathsf{e2v}^{-1}(K)\right| = \frac{\prod_{i=1}^{\max_{e\in E}|e|} |T_i|!}{\prod_{e \in \mathsf{s}(E)} \mathsf{m}_K(e)!} \quad . \tag{20}$$

Assume now that $\mathsf{e2v}(H_t) \neq \mathsf{e2v}(H')$, otherwise the acceptance probability is trivially 1. Using the expression in Eq. (20) and following the same argument as for the stub-labeled case (from Eq. (6) to Eq. (8)), we obtain Eq. (12).

Proving Lemma 4.5. Before proving Lemma 4.5 we need the technical fact corresponding to Fact A.1 but for the edge-labeled case.

Fact A.3. *Let $H = (V, E_H)$ and $K = (V, E_K)$ be two different VLHs for which there is a Curveball trade from H to K involving edges a and b in E_H. Let $W_H = (V, E_{W_H})$ be any ELH in $\mathsf{e2v}^{-1}(H)$. Let $\mathsf{n}(W_H, K)$ be the set of ELHs from $\mathsf{e2v}^{-1}(K)$ that can be reached by performing a curveball trade from E_H. It holds*

$$|\mathsf{n}(W_H, K)| = \mathsf{m}_{E_H}(a)\mathsf{m}_{W_K}(b)(1 + \mathbb{1}(|a| = |b|)) \quad . \tag{21}$$

Proof. Let c and d the edges in E_K that differ from E_H.

When $|a| \neq |b|$, let, w.l.o.g., $|a| = |c|$ and $|b| = |d|$. For every choice of pairs of edges in E_{W_H} identical to a and b there is a single Curveball trade transforming them to edges identical to c and d respectively, which is the only way to lead to an element of $\mathsf{e2v}^{-1}(K)$.

When $|a| = |b|$, for each every choice of pairs of edges in E_{W_H} identical to a and b, there are two Curveball trades leading to two distinct ELHs in $\mathsf{e2v}^{-1}(K)$: the one transforming a into c and b into d, and the one transforming a into d and b into c. □

Proof of Lemma 4.5. When $W_t = H$, the underlying chain $(E_t)_{t\geq 0}$ may be in any of the ELHs from $\mathsf{s2v}^{-1}(H)$. Using the law of total probability, we can then write

$$\xi_{\mathrm{E}}(H, K) = \sum_{E_H \in \mathsf{e2v}^{-1}(H)} \left(\Pr(E_t = E_H) \sum_{E_K \in \mathsf{n}(E_H, K)} \phi_{\mathrm{D}}(E_H, E_K)\rho_{\mathrm{E}}(E_H, E_K) \right) \quad .$$

For a fixed E_H, the value of $\phi_{\mathrm{D}}(E_H, E_K)\rho_{\mathrm{E}}(E_H, E_K)$ is the same for any $E_K \in \mathsf{n}(E_H, K)$. Let $Y_{E_H,K}$ be any of these. We can rewrite the r.h.s. above as

$$\sum_{E_H \in \mathsf{e2v}^{-1}(H)} \left(\Pr(E_t = E_H)\phi_{\mathrm{D}}(E_H, Y_{E_H,K})\rho_{\mathrm{E}}(E_H, Y_{E_H,K})|\mathsf{n}(E_H, K)| \right) \quad .$$

Additionally, $\phi_{\mathrm{D}}(E_H, Y_{E_H,K})$, $\rho_{\mathrm{E}}(E_H, Y_{E_H,K})$ and $|\mathsf{n}(E_H, K)|$ are the same for every $E_H \in \mathsf{e2v}^{-1}(H)$. Let X_H be any of these. We can rewrite the last expression as

$$\phi_{\mathrm{D}}(X_H, Y_{X_H,K})\rho_{\mathrm{E}}(X_H, Y_{X_H,K})|\mathsf{n}(X_H, K)| \sum_{E_H \in \mathsf{e2v}^{-1}(H)} \Pr(E_t = E_H),$$

and the sum on the right simplifies to 1.

By plugging in the expressions from Eqs. (11), (12) and (21), we obtain the r.h.s. of Eq. (13). □

A.1 Peskun-incomparability of VaLUH algorithms

The MC on $\mathcal{D}$ of VaLUH-d is Peskun-incomparable to both the one run by VaLUH-e and the one run by VaLUH-smh, as we show with two examples.

Consider the following VLHs on vertex set $V = \{a, b, c, d\}$:

- $H_1 = (V, E_1 \doteq \{\!\{\{a, b\}, \{a, b\}, \{c, d\}, \{c, d\}, \{a, c\}, \{b, d\}\}\!\})$
- $H_1' = (V, E_1' \doteq \{\!\{\{a, b\}, \{c, d\}, \{a, c\}, \{a, c\}, \{b, d\}, \{b, d\}\}\!\})$
- $H_2 = H_1{}'$
- $H_2' = (V, E_2' \doteq \{\!\{\{a, d\}, \{b, c\}, \{a, c\}, \{a, c\}, \{b, d\}, \{b, d\}\}\!\})$

We first consider the Curveball trade between H_1 and H_1' that changes the edges $\{a, b\}$ and $\{c, d\}$ to $\{a, c\}$ and $\{b, d\}$. Clearly, $|\{a, b\}| = |\{c, d\}|$. By plugging the quantities

- $|E_1| = 6$
- $|\{a, b\} \triangle \{c, d\}| = 4$
- $|\{a, b\} \setminus \{c, d\}| = 2$
- $\mathsf{m}_{H_1}(\{a, b\}) = \mathsf{m}_{H_1}(\{c, d\}) = 2$
- $\mathsf{m}_{H_1}(\{a, c\}) = \mathsf{m}_{H_1}(\{b, d\}) = 1$

into Eq. (9), we get $\xi_{\mathrm{E}}(H_1, H_1') = 4/45$.

We use the values above, as well as

$$|\mathsf{s}(E_1)| = \left|\mathsf{s}(E_1')\right| = 4$$

in Eq. (15) to get $\xi_{\mathrm{D}}(H_1, H_1') = 1/18$. Thus, $\xi_{\mathrm{E}}(H_1, H_1') > \xi_{\mathrm{D}}(H_1, H_1')$.

Now we apply the same analysis to H_2 and H_2', for the Curveball trade that changes $\{a, b\}$ and $\{c, d\}$ to $\{a, d\}$ and $\{b, c\}$. The only quantities that change from the previous case are

- $\mathsf{m}_{E_2}(\{a, b\}) = \mathsf{m}_{E_2}(\{c, d\}) = 1$
- $\mathsf{m}_{E_2}(\{a, d\}) = \mathsf{m}_{E_2}(\{b, c\}) = 0$

Using Eq. (9) and Eq. (15) gives us $\xi_{\mathrm{E}}(H_2, H_2') = 1/45$ and $\xi_{\mathrm{D}}(H_2, H_2') = 1/18$, so $\xi_{\mathrm{E}}(H_2, H_2') < \xi_{\mathrm{D}}(H_2, H_2')$.

Hence, ξ_{D} and ξ_{E} are Peskun-incomparable. Since ξ_{E} and ξ_{SMH} are the same, ξ_{D} and ξ_{SMH} are also Peskun-incomparable.

B Additional Experimental Results

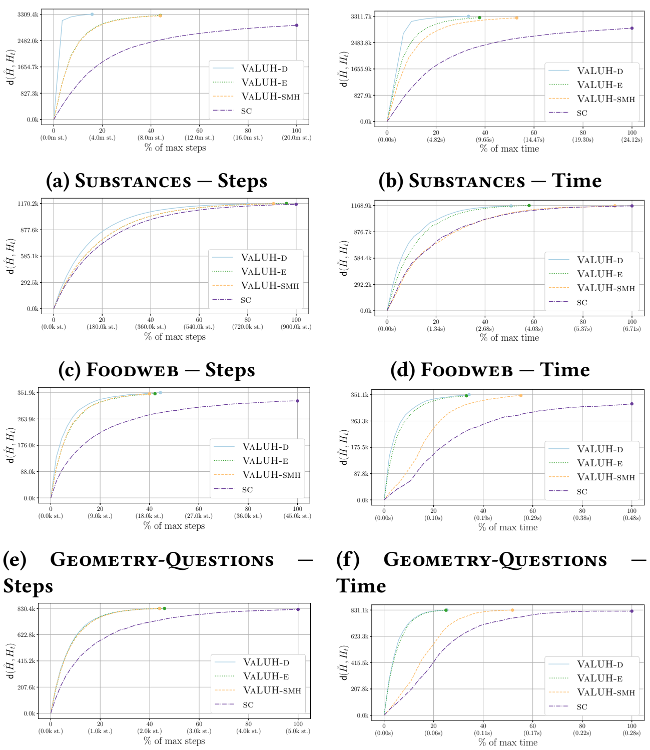
B.1 Datasets

We run our experiments on many diverse datasets. House-Bills and House-Committees [16] are both datasets related to the U.S. House of Representatives, where vertices are representatives and hyperedges are either bills or committees. Email-EU [7] records employees as nodes and emails between them as hyperedges. Substances groups substances as vertices into the drugs they make up as hyperedges. Foodweb [35] records species in the Florida Bay as vertices and groups them into hyperedges based on carbon exchange. Geometry-Questions [4] groups MathOverflow users as vertices into hyperedges based on tags of questions they had answered.

Table 2: Datasets statistics. See Sect. 5 for details.

Dataset	$\|V\|$	$\|E\|$	s	k	m
Substances	5,311	112,919	70m	700k	0.02
Email-EU	998	235,263	50m	500k	0.01
House-Bills	1,494	60,987	10m	100k	0.8
Foodweb	125	141,233	1m	100k	0.05
Geometry-Questions	580	1,193	50k	500	0.2
House-Committees	1,290	341	5k	50	8

B.1.1 Additional plots. Figure 2 shows results for the datasets not shown in Fig. 1.



(a) Substances — Steps **(b) Substances — Time**

(c) Foodweb — Steps **(d) Foodweb — Time**

(e) Geometry-Questions — Steps **(f) Geometry-Questions — Time**

(g) House-Committees — Steps **(h) House-Committees — Time**

Figure 2: Evolution of $\mathrm{d}(\mathring{H}, H_t)$ as function of percentage of max. steps and time. Absolute steps (st) and time (s) in parentheses.

C Corrections of Chodrow's results

Chodrow [15, Eq. 3] gives the following expression for the proposal probability of a SLH S' from the current state S_t of the MC on $\mathcal{D}$ (see Sect. 4.1):

$$\phi_{\mathrm{SC}}(S_t, S') \doteq \binom{|E_t|}{2}^{-1} 2^{-|a_{\mathrm{v}} \cap b_{\mathrm{v}}|} \binom{|a_{\mathrm{v}} \triangle b_{\mathrm{v}}|}{|a_{\mathrm{v}} \setminus b_{\mathrm{v}}|}^{-1} .$$

Compared to the expression in Eq. (2), it is missing the last factor $(1 + \mathbb{1}(|a| = |b|))$. This factor is needed because, when the two stub-labeled edges a and b involved in the pairwise reshuffle have the same size, there are *two* possible ways to obtain the stub-labeled edges c and d that replace a and b in the edge set of S': in one, $a = c$ and $b = d$, and in the other $c = b$ and $d = a$. When the sizes of the edges are different, there is only one way of obtaining c and d, as it will always hold $|a| = |c|$ and $|b| = |d|$, given how the pairwise reshuffled is performed.

[15, Lemma 2] also has some mistakes. Here is its statement using our notation.

Lemma C.1 (Lemma 2, 15). *Let H and H' be two VLHs in $\mathcal{D}$. For $S_1, S_2 \in \mathsf{s2v}^{-1}(H)$ and $S'_1, S'_2 \in \mathsf{s2v}^{-1}(H')$, it holds $\phi_s(S_1, S'_1) = \phi_s(S_2, S'_2)$.*

We give a counterexample to this statement. Let $V = \{a, b, c, d\}$, $H = (V, \{\{\{a, b\}, \{a, c\}, \{c, d\}\}\})$ and $H' = (V, \{\{\{a, d\}, \{a, c\}, \{c, b\}\}\})$. Consider the SLHs

- $S_1 = (V, \{\{a_1, b_1\}, \{a_2, c_1\}, \{c_2, d_1\}\}) \in \mathsf{s2v}^{-1}(H)$; and
- $S_2 = (V, \{\{a_2, b_1\}, \{a_1, c_2\}, \{c_1, d_1\}\}) \in \mathsf{s2v}^{-1}(H)$; and
- $S'_1 = (V, \{\{a_1, d_1\}, \{a_2, c_1\}, \{c_2, b_1\}\}) \in \mathsf{s2v}^{-1}(H')$.

There is a pairwise reshuffle from S_1 to S'_1 that switches the edge membership of the stubs b_1 and d_1. Thus, $\phi_s(S_1, S'_1) \neq 0$. However, there is no pairwise reshuffling between S_2 and S'_1 as they differ in all three edges, and a pairwise reshuffle modifies two or zero edges. Therefore $\phi_s(S_2, S'_1) = 0$, contradicting Lemma C.1 for $S'_2 = S'_1$.

The following revised statement corrects the above issue.

Lemma C.2. *Let $H = (V, E)$ and $H' = (V, E')$ be two VLHs in $\mathcal{D}$. For $S_1, S_2 \in \mathsf{s2v}^{-1}(H)$ and $S'_1, S'_2 \in \mathsf{s2v}^{-1}(H')$. If both $\phi_s(S_1, S'_1)$ and $\phi_s(S_2, S'_2)$ are non-zero, then they are equal.*

Proof. If both probabilities are not zero, their value, from Eq. (2), only depends on the sizes of the edges involved in the pairwise reshuffles from/to S to/from S', the size of their intersection, and whether or not they have the same size. These four quantities must be the same for both the pairwise reshuffle from S_1 to S'_1 and for the one from S_2 to S'_2. □

The proof of [15, Thm. 2] uses the wrong original version of this lemma, but it is easily fixed with our version. The statement remains correct.