Kronecker Generative Models for Power-Law Patterns in Real-World Hypergraphs

Minyoung Choe
KAIST
Seoul, Republic of Korea
minyoung.choe@kaist.ac.kr

Jihoon Ko
KAIST
Seoul, Republic of Korea
jihoonko@kaist.ac.kr

Taehyung Kwon
KAIST
Seoul, Republic of Korea
taehyung.kwon@kaist.ac.kr

Kijung Shin
KAIST
Seoul, Republic of Korea
kijungs@kaist.ac.kr

Christos Faloutsos
Carnegie Mellon University
Pittsburgh, PA, USA
christos@cs.cmu.edu

Abstract

Do real-world hypergraphs obey any patterns? Are power laws fundamental in hypergraphs as they are in real-world graphs? What generator can reproduce these patterns? A hypergraph is a generalization of a conventional graph, and it consists of nodes and hyperedges, with each hyperedge joining any number of nodes. Hypergraphs are adept at representing group interactions where two or more entities interact simultaneously, such as collaborative research and group discussions. In a wide range of real-world hypergraphs, we discover power-law or log-logistic distributions in eight structural properties. To simulate these observed patterns, we introduce HyRec, a tractable and realistic generative model leveraging the Kronecker product. We mathematically demonstrate that HyRec accurately reproduces both the patterns we observed and typical evolutionary trends found in real-world hypergraphs. To fit the parameters of HyRec to large-scale hypergraphs, we design SingFit, a fast and space-efficient algorithm successfully applied to eleven real-world hypergraphs with up to *one million* nodes and hyperedges. This paper makes the following contributions: (a) *Discoveries*: we identify multiple patterns that real-world hypergraphs obey, (b) *Model*: we propose HyRec, a tractable and realistic model capable of reproducing real-world hypergraphs efficiently (spec., with fewer than 1,000 parameters) with the support of SingFit, and (c) *Proofs*: we prove that HyRec adheres to these patterns.

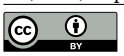
CCS Concepts

• **Information systems** → **Data mining**.

Keywords

Hypergraph, Structural pattern, Generative model, Kronecker graph

ACM Reference Format:
Minyoung Choe, Jihoon Ko, Taehyung Kwon, Kijung Shin, and Christos Faloutsos. 2025. Kronecker Generative Models for Power-Law Patterns in Real-World Hypergraphs. In *Proceedings of the ACM Web Conference 2025 (WWW '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3696410.3714893



WWW '25, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1274-6/25/04
https://doi.org/10.1145/3696410.3714893

1 Introduction

In the real world, group interactions, such as collaborative research, co-purchases of items, and group discussions on online platforms, are prevalent. These are well-represented by *hypergraphs*, where each hyperedge indicates a group interaction as a subset of nodes of any size. Hypergraphs extend conventional graphs, overcoming their limitation of exclusively modeling pairwise interactions.

What patterns or "laws" shape the structure of real-world hypergraphs? While power laws are fundamental in real-world graphs [15, 19], are they also prevalent in hypergraphs? To answer this, we analyze eight power-law properties across eleven real-world hypergraphs. First, we confirm the presence of power laws in previously identified heavy-tailed distributions, specifically for node pair degrees and hyperedge intersection sizes in hypergraphs [13, 24, 27], by applying linear regression fitting on a log-log scale. We also reveal that the slopes of these linear regressions are consistent within the same domains. Then, we find that the distributions of node degrees and hyperedge sizes follow log-logistic distributions, which are mathematically closely related to power laws. Lastly, we uncover new power-law patterns related to clustering coefficients, density, and overlap[27] in hypergraphs.

What mechanisms underlie the complex structures of real-world hypergraphs, and how can we effectively model them? Inspired by the Kronecker graph model [29], a successful generative model for conventional graphs, we introduce HyRec, a new generative model for hypergraphs. In essence, it yields an incidence matrix, indicating which nodes belong to which hyperedges, through the Kronecker power of a small-sized initiator matrix. We mathematically prove that HyRec yields five structural properties following multinomial distributions, which can mimic power-law and log-logistic distributions [4, 10, 29], and simulates evolutionary patterns of real-world hypergraphs, such as densification and shrinking diameters [24].

Since HyRec generates hypergraphs using Kronecker products of the initiator matrix, finding the most suitable initiator matrix to accurately reflect a specific real-world hypergraph is critical for the model's success. This poses significant challenges, including (C1) identifying node correspondences, (C2) ensuring differentiable generation, and (C3) maintaining computational cost feasible. To address these challenges, we propose SingFit, a fast and space-efficient fitting algorithm for HyRec. SingFit (S1) circumvents the node correspondence issue by aligning singular values, instead of hyperedge occurrences, (S2) employs Gumbel-Softmax [20] for

Table 1: Comparison of Hypergraph Generative Models. Our HyRec is the *only* model that matches all the specs. HyperCL is labeled as '?' since only analysis of node degrees and hyperedge sizes is available for it.

Capability	No Size Limit	Theoretical Analysis	Not Requiring Detailed Statistics	Extra-polation
HyperCL [27]	✔	?		
HyperFF [24]	✔		✔	✔
HyperPA [13]				✔
HyperLAP [27]	✔			
THera [21]	✔			✔
HyRec	✔	✔	✔	✔

continuous approximation of sampling to ensure differentiability, and (S3) leverages Kronecker product properties by dividing the full-matrix sampling into smaller matrices, reducing both space and time requirements.

In various real-world hypergraphs, HyRec, facilitated by SingFit, demonstrates its efficacy in two practical scenarios: (1) **Fitting**: generating hypergraphs that closely replicate real-world hypergraphs with minimal parameters, and (2) **Extrapolation**: forecasting their future growth, offering insights into potential evolutionary trends. Our contributions are as follows:

- **Discoveries**: We find out that real-world hypergraphs exhibit power-law or log-logistic distributions in various properties.
- **Model**: We propose HyRec, a tractable generative model that accurately replicates real-world hypergraph properties with a small number of parameters.
- **Proofs**: We mathematically prove that HyRec is able to replicate the discovered realistic power-law and log-logistic distributions (see Theorem 1 and 2 in Section 5).

For **reproducibility**, our code and data are available at **https://github.com/young917/HyRec**.

2 Related Work

Kronecker Models for Graphs and Their Fitting Algorithms. The Kronecker graph model [29] is recognized for its simplicity and theoretical depth, offering a lens to understand real network dynamics with a minimal set of parameters. It adeptly mimics real-world network characteristics, such as heavy-tailed degree, eigenvalue, and eigenvector distributions, making it a standard for benchmarks like Graph500 [36]. Due to its tractability, extensive research has explored its theoretical properties, including degree distributions [39], isolated nodes, triangles [37, 39], and network connectivity [33]. Applied to bipartite graphs, the Kronecker graph model reveals patterns like scaling laws for edge clustering coefficients [42], which are relevant to our study given the bipartite nature of hypergraphs in linking nodes to hyperedges. However, our study prioritizes unique attributes of hypergraphs, such as hyperedge intersection, extending beyond the insights from bipartite graph analysis.

Fitting Kronecker graph models involves strategies like maximum likelihood [29], which aligns edge occurrences with adjacency matrices, and method-of-moments estimators [17, 35], which match structural counts, such as the counts of triangles and stars. HyperKron [14] aims to match the triangle (or motif) count in graphs using 3D tensor Kronecker products. While it can be interpreted as modeling hyperedge occurrences in uniform hypergraphs, where all hyperedges have a size of three, it is not directly suitable for real-world hypergraphs characterized by diverse hyperedge sizes.

Properties and Generators of Real-World Hypergraphs. Real-world hypergraphs exhibit non-trivial structural and temporal patterns. Structural properties include heavy-tailed distributions in node degrees [13], hyperedge sizes [24], intersection sizes [24], singular values of incidence matrices [24], and node-pair degrees [27]. Moreover, hyperedges in real-world hypergraphs overlap more significantly [27] with higher transitivity [21], compared to those in random counterparts. Dynamics of time-evolving hypergraphs, including diminishing overlaps, densification, and shrinking diameters [24], have also been explored. Prior studies have examined dynamics regarding repetition [3], recency [3], burstiness [5], persistence [5, 9], ego-network structures [11], and triadic closures among nodes or hyperedges [2, 28]. For a comprehensive overview of patterns in real-world hypergraphs, see the survey [26].

Generative models aim to replicate these structural and dynamic properties. They have focused on reproducing node subset connectivity [13], modularity [16], heavy-tailed distributions [24], hyperedge overlaps [27], and transitivity [21]. Most of them, however, require detailed hypergraph statistics, such as hyperedge size distributions, as inputs for accurate reproduction (see Table 1).

3 Preliminaries

3.1 Hypergraph and Incidence Matrix

A **hypergraph** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of nodes $\mathcal{V} = \{v_1, \ldots, v_N\}$ and a set of hyperedges $\mathcal{E} = \{e_1, \ldots, e_M\} \subseteq 2^{\mathcal{V}}$. Each hyperedge $e \in \mathcal{E}$ is a non-empty subset of $\mathcal{V}$. The **degree** of a node v, denoted by d_v, is defined as the number of hyperedges containing v. A hypergraph can also be expressed by an **incidence matrix** $I(\mathcal{G}) \in \{0,1\}^{N \times M}$, where each (i,j)-th entry $g_{i,j}$ of $I(\mathcal{G})$ is 1 if and only if the hyperedge e_j contains the node v_i. A **path** in a hypergraph is defined as a sequence of hyperedges $(e_{p_1}, \cdots, e_{p_L})$ where $e_{p_i} \cap e_{p_{i+1}} \neq \emptyset$ for every $i \in \{1, \ldots, L-1\}$. The **distance** between two nodes (v_i, v_j) is defined as the length of a shortest path $(e_{p_1}, \cdots, e_{p_L})$ where $v_i \in e_{p_1}$ and $v_j \in e_{p_L}$. The **diameter** of the hypergraph is the maximum distance between any pairs of nodes; the **effective diameter** [31] is the minimum distance within which 90% or more of node pairs are reachable.

3.2 Kronecker Product and Power

Given two matrices $\mathbf{A} \in \mathbb{R}^{N \times M}$ and $\mathbf{B} \in \mathbb{R}^{P \times Q}$, the kronecker product $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{NP \times MQ}$ is a matrix formed by multiplying $\mathbf{B}$ by each element of $\mathbf{A}$, i.e.,

$$\mathbf{A} \otimes \mathbf{B} := \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1M}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{N1}\mathbf{B} & \cdots & a_{NM}\mathbf{B} \end{bmatrix}.$$

We define the K-th Kronecker power of $\mathbf{A}$ as $\mathbf{A}^{[K]} = \mathbf{A}^{[K-1]} \otimes \mathbf{A}$ and $\mathbf{A}^{[1]} = \mathbf{A}$.

3.3 Power-law and Log-logistic Distributions

Power-law distributions are frequently observed in various fields [15, 32]. Mathematically, a quantity x follows a power-law distribution if its probability distribution function is of the form $p(x) \approx x^{-\alpha}$, where α is a constant. The **log-logistic distribution** [6] occurs

when the logarithm of a random variable ($\log x$) follows a logistic distribution. A key characteristic of log-logistic distributions is that the odds ratios [12][1] derived from them follow power-law distributions, linking the two distributions.

4 Discoveries

In this section, we discover eight patterns across 11 real-world hypergraphs from six distinct dataset domains: emails, contacts, drugs (NDC), tags, threads, and co-authorship (described in Appendix A). We first uncover (**D1**) power-law distributions in node pair degrees, intersection sizes, and singular values of incidence matrices, with a focus on distribution slopes. Then, we reveal that (**D2**) node degrees and hyperedge sizes exhibit log-logistic distributions. Lastly, we discover (**D3**) additional power-law patterns in clustering coefficients, density, and overlapness in egonets.

4.1 D1: Power-law Distributions

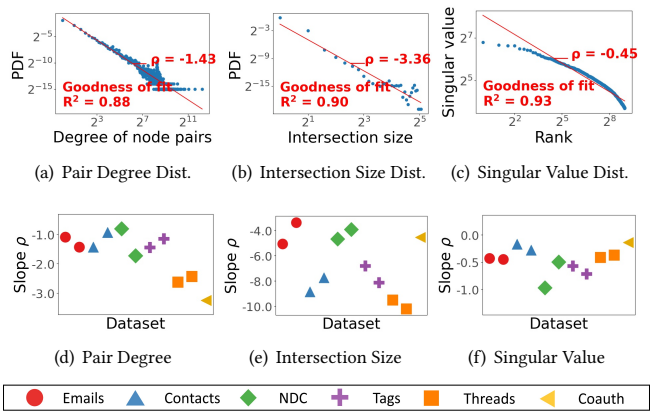
Prior studies have shown that node pair degrees [27], hyperedge intersection sizes [24], and singular values [24] exhibit heavy-tailed distributions. The **degree of a node pair** i, j is defined as $d^{(2)}(i,j) := |e \in \mathcal{E} : \{i,j\} \subseteq e|$, i.e., the number of hyperedges containing that pair. **Intersection size** measures the overlap between hyperedges e_i, e_j as $|e_i \cap e_j|$, while **singular values** are derived from the hypergraph's incidence matrix (see Section 3.1).[2]

We extend these findings to eleven real-world hypergraphs, including those with duplicated hyperedges, offering a complementary perspective to prior analyses focused on distinct hyperedges [24, 27]. Consistent with prior work, we evaluate the fits of heavy-tailed distributions, using power-law and log-normal distributions,[3] by comparing their log-likelihood ratios (LRs) against those of exponential distributions. Table 2 shows that the LRs are significantly greater than zero, indicating that power-law or log-normal distributions fit better than exponential ones. To further validate power-law characteristics, we evaluate the quality of linear regression fits (R^2 scores) on a log-log scale, commonly used to test power-law properties [12, 15]. Table 2 and Figure 1 show high R^2 scores near 1 (i.e., good linear regression fitting on a log-log scale), confirming power-law behavior. Similar slopes across hypergraphs within the same domain suggest domain-based similarities.

4.2 D2: Log-logistic Distributions

Recent studies reveal that both node degrees and hyperedge sizes exhibit heavy-tailed distributions rather than exponential ones [24]. However, they overlook the observed flatness at lower degrees or sizes, deviating from perfect power-law distributions. Our analysis of real-world hypergraphs suggests that, more precisely, both **node degrees** and **hyperedge sizes** follow *log-logistic* distributions.

Based on the relation between power-law and log-logistic distributions as discussed in Section 3.3, we investigate the odds ratios for node degrees and hyperedge sizes. Through linear regression on a log-log scale of these odds ratios, we find that the R^2 scores, indicative of fit quality, are close to 1. These power-law-like distributions of odd ratios, in turn, imply that both node degrees and hyperedge sizes adhere to log-logistic distributions. Table 2 shows R^2 scores above 0.8 across all real-world hypergraphs (see Figure 2), with similar slopes within dataset domains.



(a) Pair Degree Dist. (b) Intersection Size Dist. (c) Singular Value Dist.

(d) Pair Degree (e) Intersection Size (f) Singular Value

Figure 1: Discovery D1: Real-world Hypergraphs Follow Power-law Distributions. (a)-(c): The distributions of node pair degrees, intersection sizes, and singular values from the email-Eu dataset fit well with linear regressions on a log-log scale, indicated by R^2 scores close to 1. (d)-(f): The slopes tend to be similar within the same dataset domain.

4.3 D3: Additional Power-law Patterns

We present three new power-law patterns in egonets within real-world hypergraphs. An **egonet** for a central node v is the set of hyperedges containing v, i.e., $\tilde{\mathcal{E}}_{\{v\}} := \{e \in \mathcal{E} : v \in e\}$.

Clustering Coefficients. We investigate the count of intersecting hyperedge pairs relative to the central node's degree in egonets, focusing on pairs that share nodes beyond the central node. Specifically, we compute $|\{\{e_i, e_j\} : (e_i \cap e_j) \setminus \{v\} \neq \emptyset \wedge i \neq j \wedge e_i \in \tilde{\mathcal{E}}_{\{v\}} \wedge e_j \in \tilde{\mathcal{E}}_{\{v\}}\}|$ divided by d_v. Since this is related to the clustering coefficient in bipartite graphs [38, 42, 44], reflecting 4-cycle statistics for nodes, we refer to this as the **clustering coefficient** (clustering coef.). As shown in Table 2 and Figure 2, the distribution of intersecting hyperedge pairs relative to the central node's degree follows a power-law pattern. It fits well with linear regression on a log-log scale with high R^2 scores close to 1.0 and slopes around 2.0, indicating a consistent intersection rate across egonets.

Density and Overlapness of Egonets. We further examine the relationship between the numbers of hyperedges and nodes within egonets. As illustrated in Table 2 and Figure 2, this relationship fits well with a linear regression model in a log-log scale, exhibiting a slope greater than 1. Given that the **density** [18, 27] of an egonet is defined as the ratio of the number of hyperedges to the number of nodes (i.e., $|\tilde{\mathcal{E}}_{\{v\}}|/|\bigcup_{e \in \tilde{\mathcal{E}}_{\{v\}}} e|$), our findings suggest the egonet density tends to grow with more nodes. Similarly, the sum of hyperedge sizes in relation to the node count within egonets also reveals a power-law pattern; and the **overlapness** of egonets [27], defined

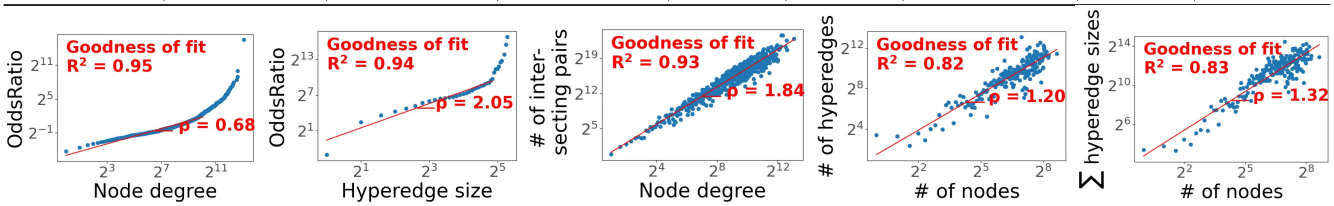
[1] The odds ratio is defined as $\text{OddRatio}(x) := \frac{\text{CDF}(x)}{1-\text{CDF}(x)}$, where CDF is the cumulative distribution function.

[2] Given the rapid decay of singular value distributions at the tail, we focus on top 50% singular values. For larger datasets, spec., tags, threads, and co-authorship, we use top 1,000, 1,000, and 500 singular values, respectively.

[3] Power-law and log-normal distributions are both common types of heavy-tailed distributions [24, 27]. We use a power-law distribution for node pair degrees and log-normal distributions for intersection sizes and singular values.

Table 2: Discoveries D1-D3: Evaluation of Power-law Fitness. For each dataset, we report the goodness of fit to power laws using the R^2 score of linear regression on a log-log scale, with the slope of the regression line. For the probability distributions regarding D1, we also compute log-likelihood ratios (LR) comparing fits to power-law or log-normal distributions against fits to exponential distributions. R^2 scores over 0.8, slopes with p-values under 0.05, and positive LRs are highlighted in bold.

Dataset	D1. Power-law Dist.									D2. Log-logistic Dist.				D3. Power-law Patterns					
	Pairdegree			Intersection			Singular Value			Degree		Size		Clustering Coef.		Density		Overlapness	
	R^2	Slope	LR	R^2	Slope	LR	R^2	Slope	LR	R^2	Slope	R^2	Slope	R^2	Slope	R^2	Slope	R^2	Slope
email-Enron	**0.8**	**-1.1**	**13**	**0.9**	-5.1	**22**	**1.0**	**-0.4**	**44**	**1.0**	**1.3**	**1.0**	**3.6**	**1.0**	**2.0**	0.1	**0.5**	0.2	**0.6**
email-Eu	**0.9**	**-1.4**	**1**	**0.9**	**-3.4**	**212**	**0.9**	**-0.4**	**158**	**1.0**	**0.7**	**0.9**	2.0	**0.9**	**1.8**	**0.8**	**1.2**	**0.8**	**1.3**
contact-primary	**0.9**	**-1.4**	**2**	**0.9**	-8.8	**8**	**0.9**	**-0.2**	**107**	**1.0**	**2.6**	**1.0**	9.7	**0.8**	**1.9**	**0.8**	**1.0**	**0.8**	**1.0**
contact-high	**0.8**	**-0.9**	**24**	**0.9**	-7.7	**20**	**0.9**	**-0.3**	**120**	**1.0**	**1.5**	**0.9**	10.0	**0.9**	**2.1**	**0.8**	**1.5**	**0.8**	**1.5**
NDC-classes	0.6	**-0.8**	**6**	**0.8**	**-4.7**	**6**	**0.9**	**-1.0**	**26**	**1.0**	**0.7**	**1.0**	**3.1**	**0.9**	**1.9**	0.2	**0.6**	0.6	**1.0**
NDC-substances	**0.8**	**-1.7**	**6**	**1.0**	-3.9	**40**	**1.0**	**-0.5**	**144**	**1.0**	**0.9**	**0.9**	1.8	0.6	1.6	0.5	0.8	**0.9**	**1.2**
tags-ubuntu	**0.8**	**-1.4**	**35**	**0.9**	-6.8	**17**	**1.0**	**-0.6**	**184**	**1.0**	**0.9**	**1.0**	**2.2**	**0.9**	**1.8**	**1.0**	**1.5**	**1.0**	**1.5**
tags-math	**0.8**	**-1.1**	**36**	**0.9**	-8.1	**5**	**1.0**	**-0.7**	**176**	**1.0**	**0.8**	**1.0**	2.4	**1.0**	**1.8**	**0.9**	**1.8**	**0.9**	**1.7**
threads-ubuntu	**0.9**	**-2.6**	**3**	**1.0**	-9.5	**3**	**1.0**	**-0.4**	**148**	**1.0**	**1.1**	**1.0**	**5.1**	**0.8**	**1.4**	**1.0**	**1.0**	**1.0**	**1.1**
threads-math	**0.9**	**-2.4**	**15**	**1.0**	-10.2	**16**	**1.0**	**-0.4**	**241**	**1.0**	**1.0**	**1.0**	**4.9**	**0.9**	**1.6**	**1.0**	**1.0**	**1.0**	**1.1**
coauth-geology	**1.0**	**-3.2**	**54**	**1.0**	-4.5	**13**	**1.0**	**-0.1**	**165**	**0.9**	**1.9**	**1.0**	**3.1**	**0.9**	**1.7**	**0.9**	**1.0**	**1.0**	1.1



(a) Degree (b) Size (c) Clustering Coef. (d) Desity (e) Overlapness

Figure 2: Discoveries D2 and D3: Real-world Hypergraphs Follow Log-logistic Distributions and Exhibit Power-law Patterns. (a)-(b): The odds ratios in relation to node degrees and hyperedge sizes are linear on a log-log scale, indicated by R^2 scores over 0.9. (c)-(e): The distributions of clustering coefficients, egonet density, and overlapness are also well-fitted by linear regression on a log-log scale. These distributions are from the email-Eu dataset.

as the ratio of the sum of hyperedge sizes to the node count (i.e., $(\sum_{e\in\tilde{\mathcal{E}}_{\{v\}}}|e|)/|\bigcup_{e\in\tilde{\mathcal{E}}_{\{v\}}}e|$), tends to increase with more nodes.

5 Proposed Hypergraph Generator

Inspired by the previously described power-law patterns in real-world hypergraphs and the Kronecker graph model [29], we introduce **HyRec**, a tractable and realistic hypergraph generative model that produces multiple power-law patterns.

5.1 Description of HyRec

We define the Kronecker product of two hypergraphs as the Kronecker product of their incidence matrices (see Section 3). For hypergraphs $\mathcal{G}$ and $\mathcal{H}$ with incidence matrices $I(\mathcal{G})$ and $I(\mathcal{H})$, their Kronecker product $\mathcal{G}\otimes\mathcal{H}$ is the hypergraph with incidence matrix $I(\mathcal{G})\otimes I(\mathcal{H})$. Based on this, we present HyRec, a hypergraph model using the Kronecker product. Given an initiator hypergraph $\mathcal{G}$ and order K, HyRec$(\mathcal{G},K) := \mathcal{G}^{[K]}$ is the hypergraph with $I(\mathcal{G})^{[K]}$, i.e., the K-th Kronecker power of $I(\mathcal{G})$, as its incident matrix.

5.2 Theoretical Characteristics of HyRec

In this section, we derive several theoretical characteristics of HyRec, including multinomial distributions across various structural measures and evolutionary patterns that mirror those in real-world hypergraphs. This tractability is valuable, allowing for easier analysis and a better understanding of HyRec's behavior. It also facilitates parameter fitting (see Section 6.2).

Structural Patterns. We prove that HyRec creates hypergraphs with several statistics following multinomial distributions. As discussed in Appendix B.2, multinomial distributions resemble log-logistic and power-law distributions, which are commonly observed in real-world hypergraphs (refer to Section 4).

Theorem 1. *HyRec$(\mathcal{G},K)$ has multinomial distributions of (1) degrees, (2) hyperedge sizes, (3) pair degrees, and (4) intersection sizes.*
Proof. Refer to Appendix B.3 for the proof. ■

Theorem 2. *In HyRec$(\mathcal{G},K)$, both singular values and singular vectors of its incidence matrix follow multinomial distributions.*
Proof. Refer to Appendix B.4 for the proof. ■

Evolutionary Patterns. We prove the evolutionary patterns of HyRec, focusing on changes in density and (effective) diameter as the exponent K of the Kronecker power increases, which can be considered as the hypergraph's growth over time.

Theorem 3. *In HyRec$(\mathcal{G},K)$ where $I(\mathcal{G})\in\{0,1\}^{N_1\times M_1}$ exhibits a $1:\frac{\log M_1}{\log N_1}$ power-law relationship between the number of nodes and the number of hyperedges as K increases.*
Proof. Refer to Appendix B.5 for the proof. ■

Theorem 4. *If the initiator hypergraph $\mathcal{G}$ has a diameter D, the diameter of HyRec$(\mathcal{G},K)$ is exactly D.*
Proof. Refer to Appendix B.6 for the proof. ■

Theorem 5. *If the initiator hypergraph $\mathcal{G}$ has a diameter D, then the effective diameter of HyRec$(\mathcal{G},K)$ converges to D as K increases.*
Proof. Refer to Appendix B.7 for the proof. ■

5.3 Stochastic HyRec: Stochastic Version

We have so far applied the Kronecker power approach to a binary initiator matrix, which always produces the same hypergraph, limiting variability, a key property for (hyper)graph models in tasks like statistical testing [35]. To address this, we introduce a stochastic version of HyRec. Starting with an initiator matrix $\Theta \in [0,1]^{N_1 \times M_1}$, we compute $\Theta^{[K]}$, where each (i,j)-th entry represents the probability of the i-th node being part of the j-th hyperedge. We then sample a hypergraph $\tilde{\mathcal{G}}$ by independently performing Bernoulli trials on each entry of $\Theta^{[K]}$, generating a binary incidence matrix $I(\tilde{\mathcal{G}})$ of $\tilde{\mathcal{G}}$. In Online Appendix [8], we illustrate how different initiator matrices, including those modeling community and core-fringe structures, lead to hypergraphs with diverse properties.

6 SingFit: Fitting to Real-World Hypergraphs

In the preceding analysis, we demonstrate that HyRec can replicate many properties of real-world hypergraphs. *But how can we generate a Kronecker hypergraph that closely resembles a specific real-world hypergraph?* Specifically, how can we identify an initiator matrix (i.e., Θ) that captures the underlying mechanisms? To answer this, we explore fitting the initiator matrix to the target hypergraph. Throughout this section, we focus on the stochastic version of HyRec, referred to simply as HyRec.

6.1 Challenges in Fitting HyRec

Fitting an initiator matrix poses the following challenges:

C1. Computational Cost of Alignment. Identifying correspondences between nodes or hyperedges of input and generated hypergraphs requires considering all possible ($|\mathcal{V}|! \times |\mathcal{E}|!$) permutations of nodes and hyperedges. Thus, directly aligning incidence matrices faces computational challenges.

C2. Non-Differentiability of Generation. The stochastic version of HyRec uses probability matrices (i,e., $\Theta^{[K]}$) to generate hypergraphs, independently drawing each entry to form binary matrices (i.e., $I(\tilde{\mathcal{G}})$). This process causes a discrepancy between the characteristics (e.g., singular values) of probability matrices and those of binarized matrices. Additionally, the non-differentiable nature of the sampling process poses challenges for parameter fitting.

C3. Density of Probability Matrix. Naively sampling a hypergraph $\tilde{\mathcal{G}}$ from the probability matrix $\Theta^{[K]}$ leads to high computational and memory overhead, as Bernoulli sampling must be applied to every possible connection between nodes and hyperedges, resulting in a complexity of $O(|\mathcal{V}| \cdot |\mathcal{E}|)$.

6.2 Strategies for Overcoming Fitting Challenges with SingFit

We propose SingFit, an initiator-matrix fitting algorithm that addresses the above challenges with three solutions.

S1: Singular-Value Matching. To avoid high alignment costs (Challenge **C1**), we aim to match statistics that do not require aligning nodes and hyperedges between input and generated hypergraphs. One of the statistics is the incidence-matrix singular values of the input hypergraph $\mathcal{G}$ and the generated hypergraph $\tilde{\mathcal{G}}$ (singular values are invariant to the row and column orders), quantified by the following loss function:

$$\mathcal{L}_\sigma = \sum_{i=1}^{|\sigma(\mathcal{G})|} \left(\sigma(\mathcal{G})_i - \sigma(\tilde{\mathbf{G}})_i\right)^2 / |\sigma(\mathcal{G})| \tag{1}$$

Here, $\sigma(\cdot)$ is a function that computes the singular values of the incident matrix of a given hypergraph, sorted in descending order. The symbol $|\sigma(\mathcal{G})|$ represents the number of singular values, i.e., the rank of the incident matrix of $\mathcal{G}$.

To further improve the model's ability, we include additional loss terms $\mathcal{L}_d$ and $\mathcal{L}_s$ to learn the distributions of node degrees and hyperedge sizes. These terms are computed similarly to $\mathcal{L}_\sigma$ by replacing $\sigma(\cdot)$ in Eq. (1) with $d(\cdot)$ and $s(\cdot)$, which denote the node degrees and hyperedge sizes, respectively, sorted in descending order. $|d(\mathcal{G})|$ and $|s(\mathcal{G})|$ represent the numbers of nodes and hyperedges respectively. Note that the values are sorted and remain independent of node and hyperedge alignments. The final loss function is: $\mathcal{L} = \mathcal{L}_\sigma + \lambda_d \mathcal{L}_d + \lambda_s \mathcal{L}_s$, where λ_d and λ_s are positive weights controlling the influence of $\mathcal{L}_d$ and $\mathcal{L}_s$ respectively. Our preliminary experiments reveal that these additional terms are especially effective for hypergraphs with small hyperedges, including those from the contact or tag domains.

S2: Differentiable Sampling with Gumbel-Softmax. To address the discrepancy between the probability matrix $\Theta^{[K]}$ and the binary incidence matrix $I(\tilde{\mathcal{G}})$, which is generated after fitting (Challenge **C2**), we propose using Gumbel-Softmax [20]. This simulates the generation process during fitting, aligning the model's output with the target binary structure while remaining differentiable.

Formally, a differentiable binary matrix $\hat{X}$ is derived from a probability matrix X of the same size as follows:

$$\tilde{p}_{i,j} = \frac{\exp\left(\frac{\log(X_{i,j}) + g_{i,j}^{(1)}}{\tau}\right)}{\exp\left(\frac{\log(1-X_{i,j}) + g_{i,j}^{(0)}}{\tau}\right) + \exp\left(\frac{\log(X_{i,j}) + g_{i,j}^{(1)}}{\tau}\right)}, \tag{2}$$

$$\hat{X}_{i,j} = p_{\text{hard}} + \tilde{p}_{i,j} - \mathbf{sg}\left(\tilde{p}_{i,j}\right), \tag{3}$$

where $p_{\text{hard}} \sim Bernoulli(\tilde{p}_{i,j})$. Here, $\tilde{p}_{i,j}$ represents the probability of sampling the (i,j)-th element, while $g_{i,j}^{(0)}$ and $g_{i,j}^{(1)}$ are independent samples from the $Gumbel(0,1)$ distribution. The softmax temperature τ controls how close $\tilde{p}_{i,j}$ is to binary values. The stop gradient operator $\mathbf{sg}(\cdot)$ ensures $\hat{X}_{i,j}$ is binary during the forward pass, while allowing gradients to flow through $\tilde{p}_{i,j}$ via $\nabla_X \hat{X}_{i,j} = \nabla_X \tilde{p}_{i,j}$. Thus this solves Challenge **C2**.

S3: Acceleration Using Kronecker Product Properties. To address the high computational costs of handling the probability matrix $\Theta^{[K]}$ (Challenge **C3**), we leverage Kronecker product properties of distributions of singular values, node degrees and hyperedge sizes. These structural properties of Kronecker hypergraphs can be expressed as Kronecker powers of the corresponding properties of the initiator hypergraph, as demonstrated in the proofs of Theorem 1 (see Appendix B.3) and Theorem 2 (see Appendix B.4). This allows us to approximate these properties sampled from $\Theta^{[K]}$ using unit sampling, where we decompose the Kronecker power to the K into L units with smaller exponents and compute the properties sampled from these smaller units.

Algorithm 1: Initiator Fitting of HyRec: SingFit

Input : (1) Incidence matrix of hypergraph $\mathcal{G} \in \mathbb{R}^{N \times M}$,
(2) Size of the initiator matrix N_1 and M_1,
(3) Number of units L,
(4) Number of iterations E
Output : Initiator $\Theta \in \mathbb{R}^{N_1 \times M_1}$

$K = \lceil \max(log_{N_1} N, log_{M_1} M) \rceil$
$\mathbf{S} \leftarrow \left(\left\lfloor \frac{K+(L-1)}{L} \right\rfloor, \left\lfloor \frac{K+(L-2)}{L} \right\rfloor, \cdots, \lfloor \frac{K}{L} \rfloor \right)$ ▷ $\sum_{l=1}^{L} \left\lfloor \frac{K+(L-l)}{L} \right\rfloor = K$
$\tilde{\sigma} \leftarrow \mathbf{I}^{1 \times 1}$
for each index $(i, j) \in \{1, \cdots, N_1\} \times \{1, \cdots, M_1\}$ **do**
  $\Theta_{i,j} \leftarrow softplus(\Theta_{i,j}^{init})$ where $\Theta_{i,j}^{init} \sim \mathcal{U}(0, 1)$
for each epoch $e = 1, \cdots, E$ **do**
  for each unit $l = 1, \cdots, L$ **do**
    for each index $(i, j) \in \{1, \cdots, N_1^{S_l}\} \times \{1, \cdots, M_1^{S_l}\}$ **do**
      Compute $\tilde{p}_{i,j}$ from $\Theta_{i,j}^{[\mathbf{S}_l]}$ according to Eq. (2)
      $p_{\text{hard}} \sim Bernoulli(\tilde{p}_{i,j})$
      $\hat{\Theta}_{i,j}^{[\mathbf{S}_l]} \leftarrow p_{\text{hard}} + \tilde{p}_{i,j} - \mathbf{sg}(\tilde{p}_{i,j})$ ▷ Eq. (3)
    $\tilde{\sigma} \leftarrow \tilde{\sigma} \otimes \sigma(\hat{\Theta}^{[\mathbf{S}_l]})$ ▷ $\sigma(\cdot)$ computes singular values
    $\tilde{d} \leftarrow \tilde{d} \otimes d(\hat{\Theta}^{[\mathbf{S}_l]})$ ▷ $d(\cdot)$ computes node degrees
    $\tilde{s} \leftarrow \tilde{s} \otimes s(\hat{\Theta}^{[\mathbf{S}_l]})$ ▷ $s(\cdot)$ computes hyperedge sizes
  Update Θ by $\nabla_\Theta \mathcal{L}_\sigma + \lambda_d \cdot \nabla_\Theta \mathcal{L}_d + \lambda_s \cdot \nabla_\Theta \mathcal{L}_s$ ▷ Eq. (1)
return Θ

While this unit-sampling approach applies to all three properties, for clarity, we describe it formally using singular values as an example. When $I(\mathcal{G})$ can be decomposed using SVD as $I(\mathcal{G}) = \mathbf{U}_1 \Sigma_1 \mathbf{V}_1^\top$, the singular values Σ_K of $I(\mathcal{G})^{[K]}$, are expressed as:

$$\begin{aligned}
\Sigma_K &= \underbrace{(\Sigma_1 \otimes \Sigma_1 \otimes \cdots \otimes \Sigma_1)}_{K \text{ times}} = \Sigma_1^{[K]} \quad (\because \textit{Appendix B.4}) \\
&= \underbrace{(\Sigma_1 \otimes \cdots \otimes \Sigma_1)}_{\mathbf{S}_1 \text{ times}} \otimes \underbrace{(\Sigma_1 \otimes \cdots \otimes \Sigma_1)}_{\mathbf{S}_2 \text{ times}} \otimes \cdots \otimes \underbrace{(\Sigma_1 \otimes \cdots \otimes \Sigma_1)}_{\mathbf{S}_L \text{ times}} \\
&= \Sigma_1^{[\mathbf{S}_1]} \otimes \Sigma_1^{[\mathbf{S}_2]} \otimes \cdots \otimes \Sigma_1^{[\mathbf{S}_L]} \text{ (where } \textstyle\sum_{i=1}^{L} \mathbf{S}_i = K) \\
&= \Sigma_{\mathbf{S}_1} \otimes \Sigma_{\mathbf{S}_2} \otimes \cdots \otimes \Sigma_{\mathbf{S}_L}.
\end{aligned}$$

Here, L is the number of units each with size $\mathbf{S}_i$. Thus, we approximate the singular values Σ_K sampled from $\Theta^{[K]}$ by calculating the singular values $\Sigma_{\mathbf{S}_i}$ sampled from smaller $\Theta^{[\mathbf{S}_i]}$ and then compute their Kronecker products. If $\mathbf{S}_i = K/L$ for all i, this substantially reduces the computational complexity from $O(\min(N_1^2 M_1, N_1 M_1^2)^K)$ to $O(L \cdot \min(N_1^2 M_1, N_1 M_1^2)^{K/L})$ (refer to Section 7.4 for empirical results and Appendix C for complexity analysis). The same approach can be applied to compute node degrees and hyperedge sizes; see our finding in the proof of Theorem 1 (Appendix B.3).

6.3 Description of SingFit

Algorithms 1 and 2 outline the fitting and generation procedures, respectively. The process involves sampling (i.e., binarizing) unit matrices across L iterations, where $L = 1$ represents the naive approach using the full probabilistic matrix. When fitting the initiator, we employ the Gumbel-Softmax technique for sampling, which enables gradient calculation. Singular values are computed from these sampled matrices and combined using Kronecker products

Algorithm 2: Hypergraph Generation in HyRec

Input : (1) Initiator matrix $\Theta \in \mathbb{R}^{N_1 \times M_1}$,
(2) order K,
(3) Number of units L
Output : A generated hypergraph $\tilde{\mathcal{G}}$

$\mathbf{S} \leftarrow \left(\left\lfloor \frac{K+(L-1)}{L} \right\rfloor, \left\lfloor \frac{K+(L-2)}{L} \right\rfloor, \cdots, \lfloor \frac{K}{L} \rfloor \right)$
for each unit $l = 1, \cdots, L$ **do**
  for each index $(i, j) \in \{1, \cdots, N_1^{S_l}\} \times \{1, \cdots, M_1^{S_l}\}$ **do**
    $\hat{\Theta}_{i,j}^{[\mathbf{S}_l]} \sim Bernoulli(\Theta_{i,j}^{[\mathbf{S}_l]})$
  $\tilde{\mathcal{G}} \leftarrow \tilde{\mathcal{G}} \otimes \hat{\Theta}^{[\mathbf{S}_l]}$
return $\tilde{\mathcal{G}}$

to produce the complete singular value set. Node degrees and hyperedge sizes are computed similarly. For the generation phase, when the sampled unit incidence matrices are sparse, we focus on '1' entries to improve computational and memory efficiency. Refer to Section 7.4 for efficiency-related experiments.

7 Experimental Results

In this section, we review our experiments, whose results demonstrate the effectiveness of HyRec.

7.1 Experimental Settings

Competitors. We consider 5 baseline generators: **HyperCL** [27], **HyperFF** [23], **HyperPA** [13], **HyperLAP** [27], and **THera** [21], with their features summarized in Table 1. Further details on their settings can be found in Appendix D.

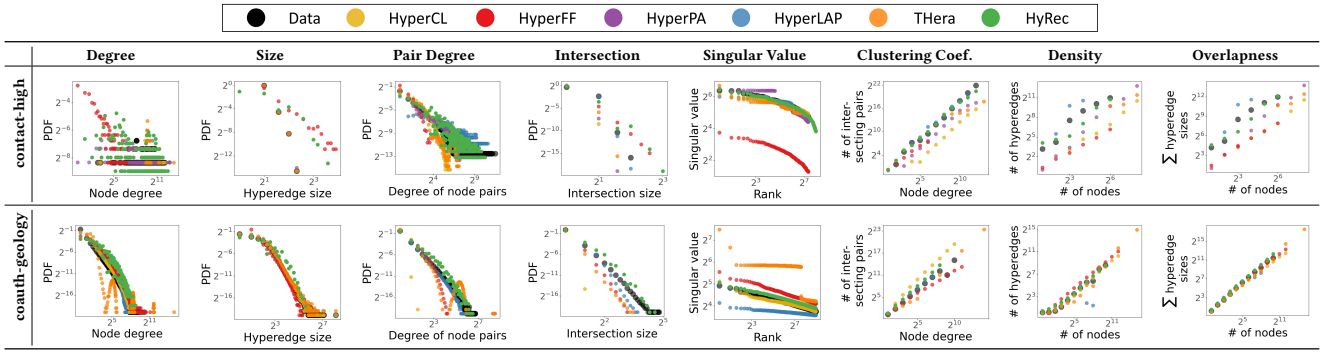
Evaluation. We evaluated HyRec's performance in replicating **9 real-world patterns** (spec., those discussed in Section 4 and the effective diameters of hypergraphs [24]). We consider the **11 real-world hypergraphs** from six domains (email, contact, drug, tags, threads, and co-authorship) described in Appendix A. We evaluate the goodness of fit using the Kolmogorov-Smirnov (*D-statistic*) for the probability density distributions of degree, size, pair degree, and intersection size; Root Mean Square Error (RMSE) for singular values, clustering coefficients, density, and overlapness[4]; and relative difference for the effective diameter which is a scalar value. Hypergraphs are generated once per model, per parameter set in the search space, and per dataset. We rank each generator's fit for each pattern and average these rankings across the 11 datasets.

Parameter Settings. In all experiments, we use at least two units (i.e., $L \geq 2$) and maintain the same number of units (L) for both training and generation. The initial matrix size ($N_1 \times M_1$) is determined using the logarithm of the original incidence matrix with base K, where the order K is chosen between 2 and 50 to minimize the size difference between the generated and target hypergraphs. Refer to Appendix D for further details and the parameter settings of the competitors.

[4]When computing RMSE of y values (e.g., singular values or intersecting pairs in egonets), we consider only the intersection of the x values (e.g., ranks or central node's degree in egonets) from the generator outputs and the ground-truth dataset. For clustering coefficients, density, and overlapness, this process is applied after a logarithmic binning of x values.

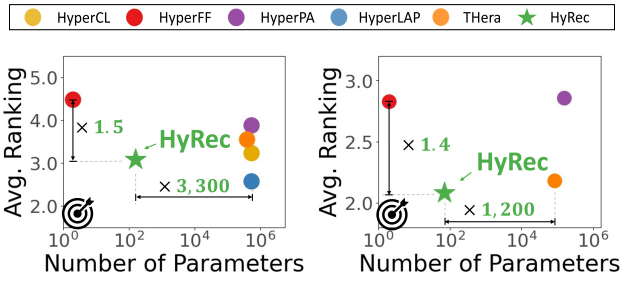
Table 3: HyRec Fits Real-World Hypergraphs.

(a) **HyRec obeys the patterns in Section 4.** HyRec (green) closely resembles the properties of real-world hypergraphs (black).



(b) **HyRec is accurate and parsimonious.** Across eleven datasets, HyRec ranks within the top three for most properties and ranks second on average. It has the second-fewest input parameters, following HyperFF, which ranks last. **Note that the top-performing model requires three orders of magnitude more parameters than HyRec.** The best, second-best, and third-best performance are highlighted in blue, green, and yellow, respectively.

	# Input Parameters		Average Ranking (Across 11 Hypergraph Datasets)									
	Min	Max	Degree	Size	Pair Deg.	Intersect.	Singular Value	Clustering Coef.	Density	Overlapness	Effective Diam.	Average
HyperCL	11,028	2,852,295	**1.455**	**1.000**	3.455	4.909	*2.364*	4.545	3.636	4.364	*3.455*	3.242
HyperFF	**2**	**2**	4.909	4.818	5.000	4.273	5.182	4.182	4.273	3.909	3.818	4.485
HyperPA	11,028	2,852,295	3.000	3.727	*3.000*	4.273	4.545	4.636	4.000	4.273	3.545	3.889
HyperLAP	11,028	2,852,295	*2.636*	**1.000**	**2.636**	**1.636**	3.182	**1.364**	*3.273*	*2.727*	4.727	**2.576**
THera	10,889	1,591,170	4.909	4.091	3.091	*2.364*	4.545	*2.636*	*3.273*	3.182	3.909	3.556
HyRec	*15*	*882*	4.091	4.818	3.818	3.545	**1.182**	3.636	**2.545**	**2.545**	**1.545**	*3.081*



(a) Fitting (b) Extrapolating

Figure 3: HyRec Demonstrates Superior Performance with Fewer Input Parameters. HyRec performs well in fitting and extrapolating real-world hypergraphs with a relatively small number of input parameters (in terms of the number of scalars), proving its efficiency and effectiveness. The rankings are averaged over all 9 patterns and 11 real-world hypergraphs, with lower values indicating better performance.

7.2 Fitting to Real-World Hypergraphs

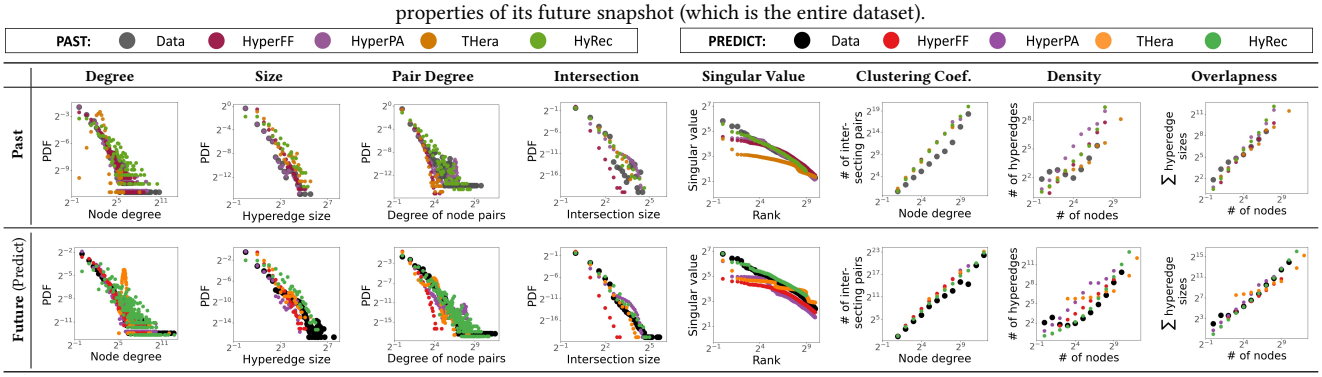
In this subsection, we present visual and statistical analyses demonstrating that HyRec excels in accurately fitting the distribution patterns of real-world hypergraphs while requiring minimal input parameters. Table 3(a) visually confirms that HyRec produces distributions closely resembling real-world hypergraphs. Table 3(b) reports the average rankings of generators in matching each of the nine properties across eleven real-world datasets. HyRec ranks within the top three for six properties, demonstrating strong alignment with most properties. Although HyRec underperforms some baselines in terms of node degrees and hyperedge sizes, it is important to note that these baselines (except for HyperFF) directly rely on detailed node-degree and hyperedge-size distributions as inputs, making their strong performance on these properties unsurprising. This adds complexity, as they require detailed statistics for every generation. In contrast, HyRec uses only an initiator matrix fitted by SingFit and a few scalars while offering both tractability and extrapolation ability, as shown in Table 1. HyperPA is inapplicable when the node count is too small (email-Enron) and runs out of memory when the hypergraph size is too large (coauth-geology), resulting in it ranking last in these cases. Notably, as illustrated in Figure 3, HyRec provides the best balance between performance and input parameter size (in terms of the number of scalars). We also analyze how HyRec's performance is affected by hyperparameters (e.g., the initiator matrix size and the unit number) in Appendix E.

7.3 Extrapolating Real-World Hypergraphs

In this subsection, we demonstrate both visually and statistically that HyRec exhibits strong extrapolation ability in forecasting the evolution of hypergraph properties. We fit HyRec using the hyperedges up until the first 50% of the nodes appear, and we test it against the full original hypergraph. The same protocol is applied to HyperPA, HyperFF, and THera, but by their design, HyperLAP and HyperCL cannot predict beyond the input data. Table 4(a) visually confirms that HyRec is able to closely mirror a given hypergraph (past) and accurately predict its future. Table 4(b) and Figure 3 present that HyRec ranks first on average while requiring two orders of magnitude fewer input parameters (in terms of the number of scalars) than the second-best one. HyRec also achieves the best

Table 4: HyRec Extrapolates Real-World Hypergraphs.

(a) **HyRec predicts hypergraph growth accurately.** After fitting to a past snapshot of the NDC-substances dataset, HyRec (green) accurately predicts the properties of its future snapshot (which is the entire dataset).



(b) **HyRec is accurate in extrapolation, even with a small number of input parameters.** HyRec performs **overall best** in extrapolation, and notably it requires two orders of magnitude fewer parameters than the second-best model (THera). Note that HyperCL and HyperLAP are inapplicable to extrapolation. The best, second-best, and third-best performance are highlighted in blue, green, and yellow, respectively.

	# Input Parameters		Average Ranking (Across 11 Hypergraph Datasets)									
	Min	Max	Degree	Size	Pair Deg.	Intersect.	Singular Value	Clustering Coef.	Density	Overlapness	Effective Diam.	Average
HyperFF	2	2	2.818	2.909	3.273	2.818	2.909	**1.818**	2.909	2.909	3.091	2.828
HyperPA	466	1,167,675	*2.636*	**1.818**	3.273	3.182	3.636	2.818	3.000	3.091	*2.273*	2.859
THera	349	537,114	3.000	*2.273*	**1.636**	**1.909**	**1.636**	2.909	**2.000**	*2.182*	**2.091**	*2.182*
HyRec	*12*	*396*	**1.545**	2.545	*1.818*	*2.091*	*1.818*	*2.455*	*2.091*	**1.818**	2.545	**2.081**

Table 5: HyRec's Efficiency Gains with Increased Units.

Number of Units (L)	Avg. Unit Matrix Size ($\lvert\Theta^{[S_i]}\rvert$)	Fitting Time (ms)	Generation Time (ms)
L = 1	85,766,121	378.983	157,656.327
L = 2	9,261	23.697	19,619.597
L = 3	441	28.730	5,369.581
L = 4	231	42.813	10,151.488
L = 5	105	48.959	7,815.624

performance for extrapolation based on a snapshot with the first 25% of nodes, as shown in Online Appendix [8].

7.4 Efficiency in Fitting Large Hypergraphs

In this subsection, we show that unit sampling (described in Section 6.2) significantly improves the efficiency of SingFit, achieving near-linear scalability in fitting and linear scalability in generation. Table 5 shows that increasing the number of units (i.e., L) in the contact-primary dataset reduces the total size of unit matrices and the runtime for both fitting and generation.[5] In this experiment, the initiator matrix is fixed at 3×7, with a Kronecker power of order 6. These results align with the complexity analysis provided in Appendix C (Tables 7 and 8). Moreover, in Appendix F, we highlight the speed of our generation method in comparison to competitors, demonstrating its efficiency even for large hypergraphs.

[5]Since units are processed serially (while operations within each unit are parallelized), the total time may increase slightly with more units.

8 Conclusions

In this study, we uncover eight power-law-related patterns in real-world hypergraphs and introduce HyRec, a generative model leveraging the Kronecker product to replicate these patterns. We mathematically demonstrate that HyRec captures both structural and evolutionary patterns. Additionally, we develop SingFit, a fast and space-efficient algorithm for fitting HyRec to given hypergraphs. Our experiments on eleven real-world hypergraphs confirm the model's efficacy in fitting and forecasting hypergraph properties. Our contributions are summarized as follows:

- **Discoveries**: Identification of eight power-law-related patterns in real-world hypergraphs (Figures 1-2 and Table 2).
- **Model**: Design of HyRec, a tractable and realistic (Figure 3 and Tables 3-4) generative model supported by SingFit.
- **Proofs**: Mathematical validation that HyRec adheres to these identified patterns (Theorems 1 and 2).

For **reproducibility**, our code and data are available at **https://github.com/young917/HyRec**.

Acknowledgements. This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00438638, EntireDB2AI: Foundations and Software for Comprehensive Deep Representation Learning and Prediction on Entire Relational Databases, 50%) (No. RS-2019-II190075, Artificial Intelligence Graduate School Program (KAIST), 10%). This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00406985, 40%).

References

[1] Sinan G Aksoy, Tamara G Kolda, and Ali Pinar. 2017. Measuring and modeling bipartite graphs with community structure. *Journal of Complex Networks* 5, 4 (2017), 581–603.

[2] Austin R Benson, Rediet Abebe, Michael T Schaub, Ali Jadbabaie, and Jon Kleinberg. 2018. Simplicial closure and higher-order link prediction. *PNAS* 115, 48 (2018), E11221–E11230.

[3] Austin R Benson, Ravi Kumar, and Andrew Tomkins. 2018. Sequences of sets. In *KDD*.

[4] Zhiqiang Bi, Christos Faloutsos, and Flip Korn. 2001. The "DGX" distribution for mining massive, skewed data. In *KDD*.

[5] Giulia Cencetti, Federico Battiston, Bruno Lepri, and Márton Karsai. 2021. Temporal properties of higher-order interactions in social networks. *Scientific reports* 11, 1 (2021), 1–10.

[6] David G Champernowne. 1952. The graduation of income distributions. *Econometrica: Journal of the Econometric Society* (1952), 591–615.

[7] Philip S Chodrow, Nate Veldt, and Austin R Benson. 2021. Generative hypergraph clustering: From blockmodels to modularity. *Science Advances* 7, 28 (2021), eabh1303.

[8] Minyoung Choe, Jihoon Ko, Taehyung Kwon, Kijung Shin, and Christos Faloutsos. 2025. Kronecker Generative Models for Power-Law Patterns in Real-World Hypergraphs: Online Appendix and Source Code. https://github.com/young917/HyRec

[9] Hyunjin Choo and Kijung Shin. 2022. On the persistence of higher-order interactions in real-world hypergraphs. In *SDM*.

[10] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. 2009. Power-law distributions in empirical data. *SIAM review* 51, 4 (2009), 661–703.

[11] Cazamere Comrie and Jon Kleinberg. 2021. Hypergraph Ego-networks and Their Temporal Evolution. In *ICDM*.

[12] Pravallika Devineni, Danai Koutra, Michalis Faloutsos, and Christos Faloutsos. 2015. If walls could talk: Patterns and anomalies in facebook wallposts. In *ASONAM*.

[13] Manh Tuan Do, Se-eun Yoon, Bryan Hooi, and Kijung Shin. 2020. Structural patterns and generative models of real-world hypergraphs. In *KDD*.

[14] Nicole Eikmeier, Arjun S Ramani, and David Gleich. 2018. The hyperkron graph model for higher-order features. In *ICDM*.

[15] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. 1999. On power-law relationships of the internet topology. *SIGCOMM CCR* 29, 4 (1999), 251–262.

[16] Frédéric Giroire, Nicolas Nisse, Thibaud Trolliet, and Małgorzata Sulkowska. 2022. Preferential attachment hypergraph with high modularity. *Network Science* 10, 4 (2022), 400–429.

[17] David F Gleich and Art B Owen. 2012. Moment-based estimation of stochastic Kronecker graph parameters. *Internet Mathematics* 8, 3 (2012), 232–256.

[18] Shuguang Hu, Xiaowei Wu, and TH Hubert Chan. 2017. Maintaining densest subsets efficiently in evolving hypergraphs. In *CIKM*.

[19] Bernardo A Huberman. 2001. *The laws of the Web: Patterns in the ecology of information*. MIT Press.

[20] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).

[21] Sunwoo Kim, Fanchen Bu, Minyoung Choe, Jaemin Yoo, and Kijung Shin. 2023. How Transitive Are Real-World Group Interactions?–Measurement and Reproduction. In *KDD*.

[22] Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *ECML*. Springer.

[23] Jihoon Ko, Yunbum Kook, and Kijung Shin. 2022. Growth patterns and models of real-world hypergraphs. *KAIS* 64, 11 (2022), 2883–2920.

[24] Yunbum Kook, Jihoon Ko, and Kijung Shin. 2020. Evolution of real-world hypergraphs: Patterns and models without oracles. In *ICDM*.

[25] Amy N Langville and William J Stewart. 2004. The Kronecker product and stochastic automata networks. *Journal of computational and applied mathematics* 167, 2 (2004), 429–447.

[26] Geon Lee, Fanchen Bu, Tina Eliassi-Rad, and Kijung Shin. 2024. A Survey on Hypergraph Mining: Patterns, Tools, and Generators. *arXiv preprint arXiv:2401.08878* (2024).

[27] Geon Lee, Minyoung Choe, and Kijung Shin. 2021. How do hyperedges overlap in real-world hypergraphs?-patterns, measures, and generators. In *WWW*.

[28] Geon Lee and Kijung Shin. 2021. Thyme+: Temporal hypergraph motifs and fast algorithms for exact counting. In *ICDM*.

[29] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. 2010. Kronecker graphs: an approach to modeling networks. *JMLR* 11, 2 (2010).

[30] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD*.

[31] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2007. Graph evolution: Densification and shrinking diameters. *TKDD* 1, 1 (2007), 2–es.

[32] Fredrik Liljeros, Christofer R Edling, Luis A Nunes Amaral, H Eugene Stanley, and Yvonne Åberg. 2001. The web of human sexual contacts. *Nature* 411, 6840 (2001), 907–908.

[33] Mohammad Mahdian and Ying Xu. 2007. Stochastic kronecker graphs. In *WAW*.

[34] Rossana Mastrandrea, Julie Fournet, and Alain Barrat. 2015. Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PloS one* 10, 9 (2015), e0136497.

[35] Sebastian Moreno, Jennifer Neville, and Sergey Kirshner. 2018. Tied Kronecker product graph models to capture variance in network populations. *TKDD* 12, 3 (2018), 1–40.

[36] Richard C Murphy, Kyle B Wheeler, Brian W Barrett, and James A Ang. 2010. Introducing the graph 500. *CUG* 19 (2010), 45–74.

[37] Ahmed Mehedi Nizam, Md Nasim Adnan, Md Rashedul Islam, and Mohammad Akbar Kabir. 2012. Properties of stochastic Kronecker graph. *arXiv preprint arXiv:1210.1300* (2012).

[38] Garry Robins and Malcolm Alexander. 2004. Small worlds among interlocking directors: Network structure and distance in bipartite graphs. *Computational & Mathematical Organization Theory* 10 (2004), 69–94.

[39] Comandur Seshadhri, Ali Pinar, and Tamara G Kolda. 2013. An in-depth analysis of stochastic Kronecker graphs. *JACM* 60, 2 (2013), 1–32.

[40] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *WWW*.

[41] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Régis, Bruno Lina, et al. 2011. High-resolution measurements of face-to-face contact patterns in a primary school. *PloS one* 6, 8 (2011), e23176.

[42] Trevor Steil, Scott McMillan, Geoffrey Sanders, Roger Pearce, and Benjamin Priest. 2020. Kronecker graph generation with ground truth for 4-cycles and dense structure in bipartite graphs. In *IPDPSW*.

[43] Hao Yin, Austin R Benson, Jure Leskovec, and David F Gleich. 2017. Local higher-order graph clustering. In *KDD*.

[44] Peng Zhang, Jinliang Wang, Xiaojia Li, Menghui Li, Zengru Di, and Ying Fan. 2008. Clustering coefficient and community structure of bipartite networks. *Physica A: Statistical Mechanics and its Applications* 387, 27 (2008), 6869–6875.

Appendix

A Datasets

We consider eleven real-world hypergraphs from six different domains [2]: (a) **Emails** [22, 30, 43] with nodes as email accounts and hyperedges as emails (i.e., the sender and receivers); (b) **Contacts** [34, 41] with nodes as people and hyperedges as group interactions; (c) **Drugs (NDC)** with nodes as drug substances and classes and each hyperedge as the group contained in a drug; (d) **Tags** with nodes as tags and hyperedges as questions attached with relevant tags; (e) **Threads** with nodes as users and each hyperedge as the group discussing in a thread; (f) **Co-authorship** [40] with nodes as authors and each hyperedge as the coauthors of a publication. We use all hyperedges in each dataset, without filtering out duplicates or large-scale hyperedges. Their statistics are given in Table 6.

Table 6: Summary of Real-world Hypergraphs.

Dataset	# Nodes	# Hyperedges	Max. Degree	Max. Size
email-Enron	143	10,885	1,327	37
email-Eu	1,005	25,148	8,664	40
contact-primary	242	106,879	2,234	5
contact-high	327	172,035	4,495	5
NDC-classes	1,161	49,726	5,358	39
NDC-substances	5,556	112,919	6,693	187
tags-ubuntu	3,029	271,233	21,004	5
tags-math	1,629	822,059	71,046	5
threads-ubuntu	125,602	192,947	2,332	14
threads-math	176,445	719,792	12,511	21
coauth-geology	1,261,129	1,591,166	1,153	284

B Theoretical Proofs of HyRec

In this section, we provide the mathematical proofs for the theorems introduced in Section 5, establishing the theoretical characteristics of HyRec.

B.1 Notations

We begin with introducing several notations.

- v_i: node in $\mathcal{G}$ for the i-th row of its incidence matrix $I(\mathcal{G})$.
- e_i: hyperedge for the i-th column of $I(\mathcal{G})$.
- $v_{i,j}$: node in $\mathcal{G} \otimes \mathcal{H}$ for the $((i-1)N_2 + j)$-th row of its incidence matrix $I(\mathcal{G} \otimes \mathcal{H})$ where $I(\mathcal{G}) \in \{0,1\}^{N_1 \times M_1}$ and $I(\mathcal{H}) \in \{0,1\}^{N_2 \times M_2}$.
- $e_{i,j}$: hyperedge for the $((i-1)M_2 + j)$-th column of $I(\mathcal{G} \otimes \mathcal{H})$.
- $v_{i_1,\cdots,i_K}$: node in HyRec$(\mathcal{G}, K)$ for the $\left(\sum_{k=1}^{K}\left((i_k-1)\cdot N^{K-k}\right)+1\right)$-th row of its incidence matrix $I(\mathcal{G})^{[K]}$ where $I(\mathcal{G}) \in \{0,1\}^{N \times M}$.
- $e_{i_1,\cdots,i_K}$: hyperedge for the $\left(\sum_{k=1}^{K}\left((i_k-1)\cdot M^{K-k}\right)+1\right)$-th column of $I(\mathcal{G})^{[K]}$.

B.2 Preliminary: Multinomial Distributions

Multinomial distributions are a generalization of binomial distributions. The parameters of a multinomial distribution are (1) k for the number of event types, (2) n for the number of (independent) trials, and (3) p_i for the probability for the i-th event occurring at each trial, for each $i \in \{1, \cdots, k\}$, where $\sum_{i=1}^{k} p_i = 1$. After n independent trials, the probability for the i-th event occurring exactly c_i times for every $i \in \{1, \cdots, k\}$, where $c_1, \cdots, c_k \geq 0$ and $\sum_{i=1}^{k} c_i = n$, is $\frac{n!}{c_1! \cdots c_k!} p_1^{c_1} \cdots p_k^{c_k}$. It is a well-known fact that, with a careful choice of the parameters, multinomial distributions behave similarly to log-logistic and power-law distributions [4, 10, 29].

B.3 Proof of Theorem 1

Theorem. *HyRec$(\mathcal{G}, K)$ has multinomial distributions of (1) degrees, (2) hyperedge sizes, (3) pair degrees, and (4) intersection sizes.*

Proof. By Theorem 5 in [29], we can easily show that (1) degrees and (2) hyperedge sizes follow multinomial distributions. Regarding (3) pair degrees, let S_1 denote the multiset[6] $\{\omega_{i,i'} : i, i' \in \{1, \cdots, N_1\} \wedge i \neq i'\}$ of node pair degrees in $\mathcal{G}_1$ and S_2 denote the multiset $\{\bar{\omega}_{j,j'} : j, j' \in \{1, \cdots, N_2\} \wedge j \neq j'\}$ of node pair degrees in $\mathcal{G}_2$, where $I(\mathcal{G}_1) \in \{0,1\}^{N_1 \times M_1}$ and $I(\mathcal{G}_2) \in \{0,1\}^{N_2 \times M_2}$. If a pair of nodes v_i and v_i' in $\mathcal{G}_1$ has pair degree $\omega_{i,i'}$, and a pair of nodes v_j and v_j' in $\mathcal{G}_2$ has pair degree $\bar{\omega}_{j,j'}$, then a pair of nodes $v_{i,j}$ and $v_{i',j'}$ in $\mathcal{G}_1 \otimes \mathcal{G}_2$ has pair degree $\omega_{i,i'}\bar{\omega}_{j,j'}$. Thus, after the $K-1$ kronecker product operations, HyperK$(\mathcal{G}_1, K)$ has the multiset of node pair degrees $\{\omega_{i_1,j_1} \cdots \omega_{i_K,j_K} : i_1, \cdots, i_K, j_1, \cdots, j_K \in \{1, \cdots, N_1\} \wedge (i_1, \cdots, i_K) \neq (j_1, \cdots, j_K)\}$. Let $s_1, \cdots, s_l$ be the distinct elements in S_1, and o_k be the number of occurrences of s_k in S_1. Then, the multiset of node pair degrees in HyperK$(\mathcal{G}_1, K)$ follows multinomial distribution where each node pair degree $s_1^{c_1} \cdots s_l^{c_l}$ (where $c_1, \cdots, c_l$ are non-negative integers and $\sum_{i=1}^{l} c_i = K$) occurs with a probability proportional to $\frac{K!}{c_1! \cdots c_l!} o_1^{c_1} \cdots o_l^{c_l}$. Since (4) intersection sizes of a pair of hyperedge j and j' of $\mathcal{G}_1$ is the number of common entries between j-th column and j'-th column of $\mathcal{G}_1$, the above proof is applied similarly. ∎

[6] A *multiset* generalizes a *set* by allowing duplicate elements.

B.4 Proof of Theorem 2

Theorem. *In HyRec $(\mathcal{G}, K)$, both singular values and singular vectors of its incidence matrix follow multinomial distributions*

Proof. Let the singular value decomposition (SVD) of the incidence matrix $I(\mathcal{G}) \in \{0,1\}^{N_1 \times M_1}$ of the initiator hypergraph $\mathcal{G}$ be $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{N_1 \times R}, \mathbf{\Sigma} \in \mathbb{R}^{R \times R}$, and $\mathbf{V} \in \mathbb{R}^{M_1 \times R}$. By the properties of Kronecker product [25], the SVD of $I(\mathcal{G})^{[K]}$ is $\mathbf{U}^{[K]}\mathbf{\Sigma}^{[K]}(\mathbf{V}^{[K]})^\top$. Thus, HyRec$(\mathcal{G}, K)$ has multinomial distributions of both singular values and singular vectors. ∎

B.5 Proof of Theorem 3

Theorem. *In HyRec$(\mathcal{G}, K)$ where $I(\mathcal{G}) \in \{0,1\}^{N_1 \times M_1}$ exhibits a $1 : \frac{\log M_1}{\log N_1}$ power-law relationship between the number of nodes and the number of hyperedges as K increases.*

Proof. The node count $N(K)$ and the hyperedge count $E(K)$ in HyRec$(\mathcal{G}, K)$ become N_1^K and M_1^K, respectively. Consequently, the relationship between $E(K)$ and $N(K)$ satisfies $E(K) = M_1^K = (N_1^K)^a = N(K)^a$ where $a = \frac{logM_1}{logN_1}$. ∎

B.6 Proof of Theorem 4

We begin with presenting Lemma 1 and Lemma 2, which are used for proofs.

Lemma 1. *The following claims hold:*

- *There exists a hyperedge containing $v_{i,j}$ and $v_{k,l}$ in $\mathcal{G} \otimes \mathcal{G}'$ if and only if (a) there is a hyperedge containing both v_i and v_k in $\mathcal{G}$, and (b) there is a hyperedge containing both v_j and v_l in $\mathcal{G}'$.*
- *The hyperedge $e_{p,q}$ in $\mathcal{G} \otimes \mathcal{G}'$ contains $v_{i,j}$ if and only if (a) e_p in $\mathcal{G}$ contains v_i, and (b) e'_q in $\mathcal{G}'$ contains v'_j.*

Proof. The claims are straightforwardly deduced from the definition of the Kronecker product. ∎

Lemma 2. *If two hypergraphs $\mathcal{G}$ and $\mathcal{G}'$ each have a diameter at most D, then $\mathcal{G} \otimes \mathcal{G}'$ also has a diameter at most D.*

Proof. Consider two arbitrary nodes v_i, v_k in $\mathcal{G}$ and two arbitrary nodes v'_j, v'_l in $\mathcal{G}'$. Let a be the distance between node v_i and v_k in $\mathcal{G}$, and b be the distance between node v'_j and v'_l in $\mathcal{G}'$. Then, there is a path $(e_{p_1}, \cdots, e_{p_a})$ between v_i and v_k in $\mathcal{G}$, and there is a path $(e'_{q_1}, \cdots, e'_{q_b})$ between v'_j and v'_l in $\mathcal{G}'$. By Lemma 1, in $\mathcal{G} \otimes \mathcal{G}'$, the node $v_{i,k}$ is contained in the hyperedge e_{p_1,q_1}, and the node $v_{j,l}$ is contained in the hyperedge e_{p_a,q_b}. Additionally, for every $k \leq \max(a,b) - 1$, $e_{p_{\min(k,a)},q_{\min(k,b)}} \cap e_{p_{\min(k+1,a)},q_{\min(k+1,b)}} \neq \emptyset$ holds. Therefore, a path $(e_{p_{\min(1,a)},q_{\min(1,b)}}, \ldots, e_{p_{\min(\max(a,b),a)},q_{\min(\max(a,b),b)}})$ exists between $v_{i,k}$ and $v_{j,l}$ in $\mathcal{G} \otimes \mathcal{G}'$. Since $\max(a,b) \leq D$, the distance between any two nodes in $\mathcal{G} \otimes \mathcal{G}'$ is at most D. ∎

Theorem. *If the initiator hypergraph $\mathcal{G}$ has a diameter D, the diameter of HyRec$(\mathcal{G}, K)$ is exactly D.*

Proof. From Lemma 2, we can easily show the diameter of HyRec$(\mathcal{G}, K)$ is at most D by employing induction on K. Since the diameter of $\mathcal{G}$ is D, there exists a pair of nodes (v_i, v_j) such

Table 7: Generation Time and Memory Complexity Comparisons between HyRec and its Competitors.

Generator	Time Complexity	Memory Complexity
HyperCL	$O(log_2\|\mathcal{V}\| \cdot \sum_{e\in\mathcal{E}} \|e\|)$	$O(\|\mathcal{V}\| + \sum_{e\in\mathcal{E}} \|e\|)$
HyperFF	$O(\|\mathcal{V}\| \cdot \sum_{e\in\mathcal{E}} \|e\|)$	$O(\|\mathcal{V}\| + \sum_{e\in\mathcal{E}} \|e\|)$
HyperPA	$O(\sum_{e\in\mathcal{E}} log_2(\binom{\|\mathcal{V}\|}{\|e\|}))$	$O(\sum_{e\in\mathcal{E}} 2^{\|e\|})$
HyperLAP	$O(log_2\|\mathcal{V}\| \cdot \sum_{e\in\mathcal{E}} \|e\|)$	$O(\|\mathcal{V}\| + \sum_{e\in\mathcal{E}} \|e\|)$
THera	$O(log_2\|\mathcal{V}\| \cdot \sum_{e\in\mathcal{E}} \|e\|)$	$O(\|\mathcal{V}\| + \sum_{e\in\mathcal{E}} \|e\|)$
HyRec	$O(L(\|\mathcal{V}\|\|\mathcal{E}\|)^{\frac{1}{L}} + \sum_{e\in\mathcal{E}} \|e\|)$	$O((\|\mathcal{V}\|\|\mathcal{E}\|)^{\frac{1}{L}} + \sum_{e\in\mathcal{E}} \|e\|)$

Table 8: Time Complexity of SingFit (Algorithm 1).

SingFit (Algorithm 1)	Time Complexity
Generation (Line 11)	$O(L(\|\mathcal{V}\|\|\mathcal{E}\|)^{\frac{1}{L}})$
Computing Singular Values (Line 12)	$O(L \cdot min(\|\mathcal{E}\|^2\|\mathcal{V}\|, \|\mathcal{E}\|\|\mathcal{V}\|^2)^{\frac{1}{L}})$
Computing Degree Distributions (Line 13)	$O(L \cdot (\sum_{e\in\mathcal{E}} \|e\|)^{\frac{1}{L}} + \|\mathcal{V}\|)$
Computing Size Distributions (Line 14)	$O(L \cdot (\sum_{e\in\mathcal{E}} \|e\|)^{\frac{1}{L}} + \|\mathcal{E}\|)$

that the distance between v_i and v_j is exactly D. Suppose the diameter of HyRec$(\mathcal{G}, K)$ is at most $D-1$. Then, there is a path $(e_{p_{11},\cdots,p_{1K}}, e_{p_{21},\cdots,p_{2K}}, \cdots, e_{p_{l1},\cdots,p_{lK}})$ between $v_{i,\cdots,i}$ and $v_{j,\cdots,j}$ in HyRec$(\mathcal{G}, K)$, whose length l is at most $D-1$. Since HyRec$(\mathcal{G}, K)$ is equivalent to HyRec$(\mathcal{G}, K-1) \otimes \mathcal{G}$, $(e_{p_{1K}}, e_{p_{2K}}, \cdots, e_{p_{lK}})$ is a valid path between v_i and v_j in $\mathcal{G}$, and its length is at most $D-1$, which is a contradiction. ∎

B.7 Proof of Theorem 5

Theorem. *If the initiator hypergraph $\mathcal{G}$ has a diameter D, then the effective diameter of HyRec$(\mathcal{G}, K)$ converges to D as K increases.*

Proof. Suppose we randomly select a node $a = (v_{a_1,\cdots,a_K})$ and a node $b = (v_{b_1,\cdots,b_K})$ from HyRec$(\mathcal{G}, K)$, where $I(\mathcal{G}) \in \{0,1\}^{N_1 \times M_1}$. Since the diameter of $\mathcal{G}$ is D, there exists a pair of nodes (v_i, v_j) such that the distance between v_i and v_j is exactly D, then with probability $1-(1-\frac{1}{N_1^2})^K$, there is some index $k \in \{1,\cdots,K\}$ s.t $(a_k, b_k) = (i, j)$. Thus, the probability that the distance between two nodes in HyRec$(\mathcal{G}, K)$ is exactly D is at least $1-(1-\frac{1}{N_1^2})^K$. Since there exists $K_0 = 3N_1^2 \in \mathbb{N}_+$ such that $1-(1-\frac{1}{N_1^2})^K > 1-\frac{1}{e^3} > 0.9$ for all $K > K_0$, the effective diameter of HyRec$(\mathcal{G}, K)$ converges to D as K increases. ∎

C Analysis of Time and Memory Complexity

Table 7 summarizes the time and memory complexities of all models, including HyRec. In all experiments, we use at least two units (i.e., $L \geq 2$). The complexities of competing methods are discussed in detail in [21], while the formal proofs of HyRec's time and memory complexities are provided in Online Appendix [8]. Additionally, Table 8 outlines the time complexity of SingFit, with further details available in Online Appendix [8].

D Experimental Settings

D.1 Competitors

We consider 5 baseline generators: **HyperCL** [27], **HyperFF** [23], **HyperPA** [13], **HyperLap** [27], and **THera** [21], with their features summarized in Table 1. HyperCL and HyperLAP rely on node degree and hyperedge size distributions as inputs. HyperPA is limited to hyperedges of size under 20 and requires distributions for both hyperedge sizes and the number of new hyperedges for nodes. THera also demands hyperedge size distributions. HyperFF is based on the forest fire model controlled by two parameters. While more generators [1, 7] could be considered for comparison, we focus on competitive open-source generators that have been shown to reproduce realistic structural properties.

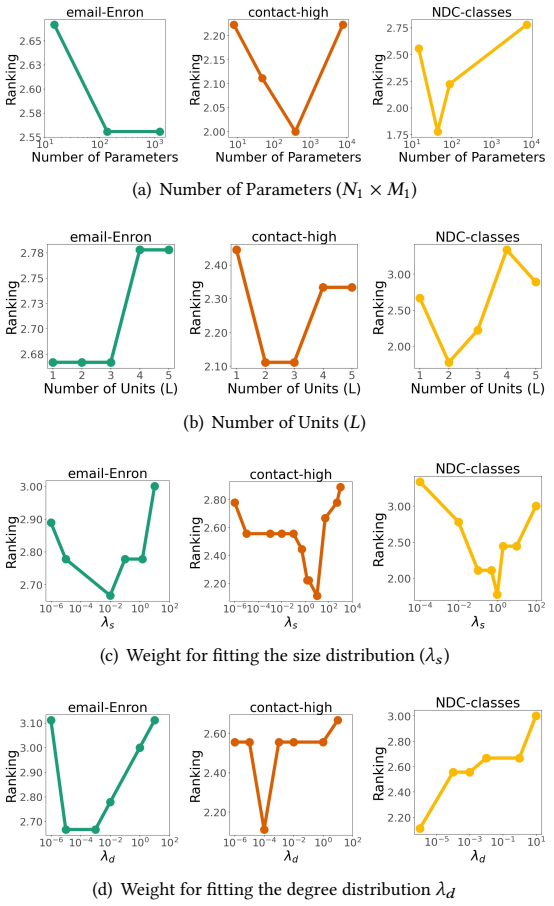
D.2 Parameter Settings

- **HyperCL** and **HyperLAP**: Both models require the distribution of node degrees and hyperedge sizes.
- **HyperFF**: $p \in [0.45, 0.48, 0.51]$ and $q \in [0.2, 0.3]$.
- **HyperPA**: It requires the distribution of hyperedge sizes and the number of new hyperedges per new node.
- **THera**: Its parameters are $C \in [8, 12, 15]$, $p \in [0.5, 0.7, 0.9]$, $\alpha \in [2, 6, 10]$, and hyperedge size distributions.
- **HyRec**: The entries of the initiator matrix are parameters. The size of the initiator incidence matrix, $N_1 \times M_1$, is determined by $N_1 = \left\lceil |\mathcal{V}|^{1/K} \right\rceil$ and $M_1 = \left\lceil |\mathcal{E}|^{1/K} \right\rceil$. Here K is chosen from $k \in [2, 50]$ to minimize $\left| \left\lceil |\mathcal{V}|^{1/k} \right\rceil^k \left\lceil |\mathcal{E}|^{1/k} \right\rceil^k - |\mathcal{V}||\mathcal{E}| \right|$, subject to $1 < \left\lceil |\mathcal{V}|^{1/k} \right\rceil \left\lceil |\mathcal{E}|^{1/k} \right\rceil \leq S$, where $S \in [50, 100, 1000]$. The other parameters are (a) the learning rate $\alpha \in [0.001, 0.003, 0.005, 0.008, 0.01]$, (b) the Gumbel-Softmax temperature $\tau = 0.0005$, (c) the number of units $L \in [2, 3, 4]$; and (d) $\lambda_d \in [0.0, 0.0001, 0.001, 0.01, 0.1]$ and $\lambda_s \in [0.0, 2.0]$, which are the weights for losses.

E Parameter Sensitivity of HyRec

We analyze the effect of hyperparameters on the performance of HyRec as shown in Figure 4. The hyperparameters include (see Algorithm 1): (1) the number of parameters, i.e., the size of the initiator matrix ($N_1 \times M_1$), (2) the number of units (L), (3) λ_s, and (4) λ_d. We conduct experiments on three small real-world hypergraphs, email-Enron, contact-high, and NDC-classes, in terms of the number of nodes. For each experiment, performance is evaluated based on the average ranking derived from reproducing nine properties of the target hypergraph.

Number of Parameters ($N_1 \times M_1$). We observe improved performance (i.e., lower ranking) as the number of parameters increases. For the email-Enron dataset, the ranking converges as the number increases. For the contact-high and NDC-classes dataset, performance improves with fewer than 1,000 and 100 parameters, respectively, but declines as the parameter count approaches 10,000. We hypothesize that too many parameters lead to an excess of zero entries. The difference between the space $N_1 \times M_1$ and the actual size of the hypergraph $|\mathcal{V}| \times |\mathcal{E}|$ increases as more parameters are added, introducing noise. Therefore, while increasing the number of parameters enhances performance initially, an overabundance



(a) Number of Parameters ($N_1 \times M_1$)

(b) Number of Units (L)

(c) Weight for fitting the size distribution (λ_s)

(d) Weight for fitting the degree distribution λ_d

Figure 4: The Effect of Hyperparameters on the Performance of HyRec.

can negatively impact the results. For this specific experiment, we fix the number of units at 2 and we explore best values for λ_d and λ_s within the search space detailed in Appendix D.2 to find the best ranking for each parameter count.

Number of Units (L). Using a single unit corresponds to handling the full Kronecker power, which requires managing all possible connections between nodes and hyperedges. However, utilizing multiple units maintains the performance on the email-Enron dataset and even improves performance on the contact-high and NDC-classes dataset, while reducing the computational complexity of HyRec, as demonstrated in the complexity analysis (Appendix C). Increasing the number of units beyond a certain point, however, does not yield further performance gains. We hypothesize this is because smaller unit sizes result in noisier singular values, reducing their usefulness. For example, when the number of units exceeds 3 in email-Enron, the smallest unit size becomes 3×5, which may generate less informative singular values compared to larger units. For this experiment, we fix the number of parameters at 3×5 and 4×12, respectively, and search for best values of λ_d and λ_s within the search space outlined in Appendix D.2, respectively, to determine the best ranking performance for each unit number.

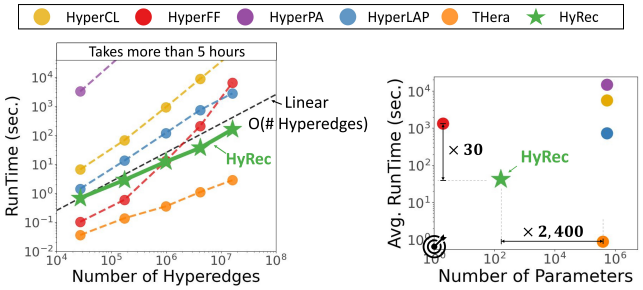


Figure 5: Generation Time Efficiency of HyRec Compared to Other Hypergraph Generators Across Synthetic Datasets.

λ_s and λ_d. We observe that performance improves up to a certain point, beyond which it declines as the values of λ_s or λ_d exceed that point. Specifically, HyRec performs best on the email-Enron dataset when $\lambda_s = 0.01$ but performs best at $\lambda_s = 10.0$ on the contact-high dataset and $\lambda_s = 1.0$ on the NDC-classes dataset. Since these hyperparameters control the weights for the loss functions that match the size and degree distributions, respectively, we hypothesize that matching size distributions is more challenging for the contact-high dataset, which has a maximum hyperedge size limited to 5, thus necessitating a higher weight for λ_s. For λ_d, although the optimal values vary slightly across datasets, they remain relatively consistent compared to the variations in λ_s. This suggests that degree distribution alignment is less sensitive to hyperparameter tuning than size distribution alignment. For the specific experimental setup, we set the parameters to $(N_1, M_1, L, \lambda_s, \lambda_d) = (3, 5, 2, 0.01, 0.001)$ and $(4, 12, 2, 1.5, 0.0)$ for each dataset, which are the best hyperparameters found within the search space described in Appendix D.2. We then vary only the target hyperparameter λ_s or λ_d to evaluate the impact of each on the results.

F Comparison of Generation Time Across Hypergraph Generators

In addition to the time complexity analysis of generation provided in Appendix C, we compare the empirical runtime of all methods, as shown in Figure 5. We use synthetic hypergraphs generated by HyRec, with an initiator matrix fitted to the email-Eu hypergraph, and vary the Kronecker power from 5 to 9 (up to 10^7 hyperedges). Specifically, HyperFF is set with parameters $p = 0.51$ and $q = 0.3$, THera with $C = 8$, $p = 0.7$, and $\alpha = 10$, and HyRec with two units ($L = 2$). We set a runtime limit of 5 hours, so generators without markers in some cases in Figure 5 indicate that they exceeded this limit. Specifically, HyperFF and HyperPA surpassed the 5-hour threshold. The results show that HyRec consistently requires less time than HyperLAP, HyperCL, and HyperPA, and even for the largest hypergraph with over 10^6 hyperedges, HyRec achieves the second-fastest runtime. While THera is the fastest method, it requires 2,400 times more input parameters (as shown on the right side of Figure 5), and HyRec outperforms it in the extrapolation task (refer to Tables 4).