Attributed Hypergraph Generation with Realistic Interplay Between Structure and Attributes

***Abstract*—In many real-world scenarios, interactions happen in a group-wise manner with multiple entities, and therefore, hypergraphs are a suitable tool to accurately represent such interactions. Hyperedges in real-world hypergraphs are not composed of randomly selected nodes but are instead formed through structured processes. Consequently, various hypergraph generative models have been proposed to explore fundamental mechanisms underlying hyperedge formation. However, most existing hypergraph generative models do not account for node attributes, which can play a significant role in hyperedge formation. As a result, these models fail to reflect the interactions between structure and node attributes.**

To address the issue above, we propose NOAH, a stochastic hypergraph generative model for attributed hypergraphs. NOAH utilizes the core–fringe node hierarchy to model hyperedge formation as a series of node attachments and determines attachment probabilities based on node attributes. We further introduce NOAHFIT, a parameter learning procedure that allows NOAH to replicate a given real-world hypergraph. Through experiments on nine datasets across four different domains, we show that NOAH with NOAHFIT more accurately reproduces the structure–attribute interplay observed in the real-world hypergraphs than eight baseline hypergraph generative models, in terms of six metrics.

***Index Terms*—Hypergraph, Generator, Node Attribute**

I. INTRODUCTION

Many real-world interactions occur in groups, such as co-authorship among researchers, group discussions on online Q&A sites, and co-purchasing of items. Hypergraphs, which consist of hyperedges, naturally and effectively represent group interactions involving an arbitrary number of individuals or entities. Especially, hypergraph modeling has shown effectiveness in a variety of applications, including clustering [1], [2], classification [3], and anomaly detection [4], [5].

Hyperedges in real-world hypergraphs are not composed of random nodes but are generally formed in a more systematic manner. For instance, real-world hypergraphs often exhibit high-degree nodes [6], densely overlapping hyperedges [7], and high transitivity [8].

Building upon these findings, a number of hypergraph generative models have been proposed, incorporating hyperedge generation mechanisms that lead to realistic hypergraph structures. These hypergraph generation models allow a better understanding of real-world hypergraphs, and they are also employed in various data-mining applications, including community detection [9]–[11], and hyperedge prediction [12].

Despite the success of hypergraph generative models, they mostly overlook interplays between hypergraph structure and node attributes. Node attributes are commonly associated with real-world data. For example, in co-authorship hypergraphs, where nodes represent authors and hyperedges represent co-authored publications, node attributes, such as affiliation and field of study, offer valuable information. Especially, as exemplified by homophily [13], [14], such node attributes can influence the formation of collaborations (i.e., hyperedges).

Thus, in this paper, we propose NOAH (**No**de **A**ttribute based **H**ypergraph generator), a novel hypergraph generative model based on node attributes. Since a hyperedge can involve an arbitrary number of nodes, the number of hyperedge candidates increases exponentially with the number of nodes. As a result, generating a hypergraph by considering the formation probabilities of all candidates is computationally intractable. To address this challenge, NOAH models the formation of each hyperedge as a series of attachments of nodes to its seed node(s). The attachment probabilities are determined by node attributes. Specifically, we assign the degree of affinity based on the values of each node attribute, and we obtain the final attachment probability as the product of the affinity scores across all attributes. In addition, NOAH incorporates a core–fringe node hierarchy into the process to enhance realism.

We also introduce NOAHFIT, an algorithm designed to fit the parameters of NOAH to a given hypergraph. The hyperedge formation probabilities in NOAH are expressed through a parameterized formulation, and NOAHFIT updates the parameters to maximize the probabilities, capturing the structure-attribute interplay in the given hypergraph.

In our experiments, we extensively evaluate hypergraph generative models using six measures on their ability to capture structure–attribute interplay in nine real-world hypergraphs. As exemplified in Figure 1, NOAH, fitted by NOAHFIT, outperforms all eight existing hypergraph generative models in the overall assessment across the measures.

Our contributions are summarized as follows:

- **Model:** We propose NOAH, a stochastic generative model for attributed hypergraphs that produces a realistic interplay between structure and attributes.
- **Fitting Algorithm:** We develop NOAHFIT, a parameter fitting algorithm for NOAH that captures the relationship between structure and node attributes in a given hypergraph to maximize the formation probabilities of its hyperedges.
- **Experiments:** We empirically show that NOAH better reproduces the structure–attribute interplay in real-world hypergraphs than eight baseline hypergraph generative models.

For **reproducibility**, we make the source code and data publicly available at anonymous.4open.science/r/NoAH-246E.

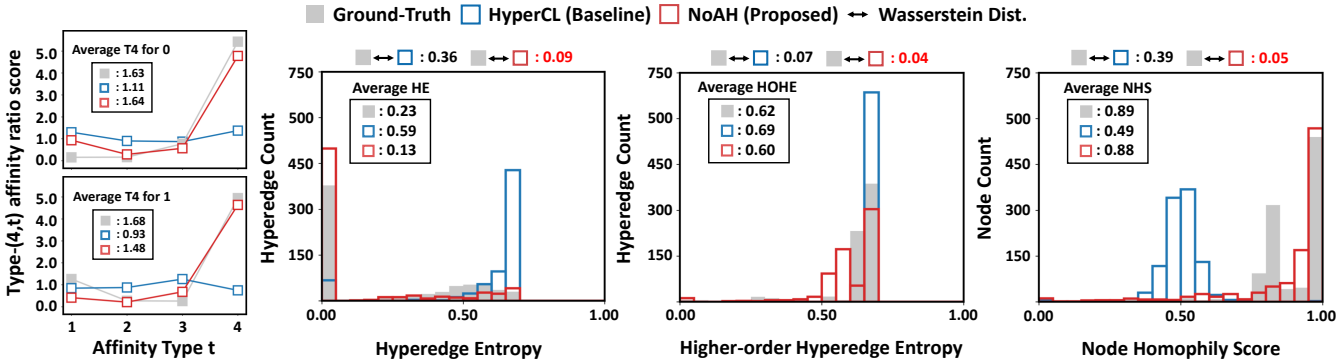


Fig. 1. Structure–attribute interplay with respect to the first node attribute in the Amazon Music dataset. Our proposed generative model, NOAH, fitted by its parameter learning algorithm NOAHFIT, effectively captures the interplay between structure and node attributes, outperforming the baseline model (HyperCL). Refer to Section III-B for the definitions of the measures and to Section VI-C for the interpretation of the results.

II. Related Work

In this section, we review prior work on the generation of graphs and hypergraphs.

A. (Hyper)graph Generative Models

(Hyper)graph generative models have been developed to generate (hyper)graphs structurally similar to those observed in the real world [15], [16]. One of their primary objectives is to uncover potential mechanisms underlying the formation of real-world (hyper)graphs. These models have also been used for various data-mining applications, including anonymization [17], link prediction [12], [18], and community detection [9]–[11], [18]. Notably, several models exploit node hierarchies [8], [19]–[21], commonly found in the real world [22]–[25]. In NOAH, we incorporate a core–fringe structure to capture hierarchies in a simple yet effective manner.

Recently, various deep learning based (hyper)graph generative models have been introduced, such as variational autoencoders [26], [27], generative adversarial networks (GANs) [28], [29], and diffusion or hierarchical models [30]–[32]. However, these models require a large collection of graph instances for training, while the aforementioned models require only a single graph. More critically, deep learning models often struggle to provide insights into the generative mechanisms of real-world (hyper)graphs due to their intrinsic black-box nature. For these reasons, we do not consider deep learning based approaches as direct competitors in this work.

B. Attributed-aware (Hyper)graph Generative Models

Node attributes can play a crucial role in graph generation, as demonstrated by homophily, a prevalent property of real-world (hyper)graphs that nodes with similar characteristics tend to be connected [13]. Accordingly, there have been a number of attempts to incorporate node attributes into generative models such as the exponential random graph model [33], the stochastic block model [18], and the latent space model [34]. Among them, the Multiplicative Attribute Graph (MAG) model [35] offers a distinctive generative framework in which node attributes directly govern the formation of edges via a multiplicative probability function.

However, very few hypergraph generative models incorporate node attributes. One exception is a stochastic block model variant specifically designed for community detection [11]. In this model, node attributes are conditionally independent of the hypergraph structure given the latent community structure, meaning node features do not directly influence hyperedge formation. To directly investigate how node attributes influence hyperedge formation, we propose NOAH, a generative hypergraph model that forms hyperedges based on attribute relationships among nodes.

TABLE I
FREQUENTLY-USED SYMBOLS.

Notation	Definition
$\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$	hypergraph with nodes $\mathcal{V}$, hyperedges $\mathcal{E}$
$\mathbf{X} \in \{0,1\}^{\lvert\mathcal{V}\rvert \times k}$	node attribute matrix (binary)
$\mathcal{C}$	set of core nodes
$\mathcal{F}$	set of fringe nodes
$\Theta_{\mathcal{C}} = \{\boldsymbol{\theta}_{\mathcal{C}_1}, \dots, \boldsymbol{\theta}_{\mathcal{C}_k}\}$	set of core group affinity matrices
$\Theta_{\mathcal{F}} = \{\boldsymbol{\theta}_{\mathcal{F}_1}, \dots, \boldsymbol{\theta}_{\mathcal{F}_k}\}$	set of fringe attachment affinity matrices

III. Preliminaries

In this section, we first define the notations used throughout this paper. Then, we discuss six measures to evaluate the interplay between structure properties and node attributes. Lastly, as a preliminary generative model, we review MAG [35].

A. Notations

First, we discuss the notations used in this paper. Refer to Table I for the frequently-used notations.

Attributed Hypergraphs. An *attributed hypergraph* $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ consists of a set of nodes $\mathcal{V} = \{v_1, \cdots, v_{|\mathcal{V}|}\}$, a set of hyperedges $\mathcal{E} = \{e_1, \cdots, e_{|\mathcal{E}|}\}$, and a node attribute matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times k}$. Each hyperedge $e \in \mathcal{E}$ is a non-empty subset of nodes, i.e., $e \subseteq \mathcal{V}, |e| \geq 1$. The i-th row of $\mathbf{X}$, denoted as $\mathbf{x}_i = \mathbf{X}_{i,:} \in \mathbb{R}^k$, represents the attribute vector of node $v_i \in \mathcal{V}$. We use $\mathbf{x}_i^{(l)} \in \mathbb{R}$ to denote the l-th attribute value of node v_i. In this work, we assume that node attributes are binary, i.e., $\mathbf{X} \in \{0,1\}^{|\mathcal{V}| \times k}$. This assumption simplifies both model design and implementation while remaining valid for many real-world datasets. Moreover, categorical and continuous attributes can also be converted into binary ones via one-hot encoding and thresholding, respectively, as in our experiments.

B. Measures for Structure-Attribute Interplay

We introduce several measures for evaluating the interplay between structural patterns and node attributes.

Type-s Affinity Ratio Scores. Veldt et al. [36] introduced a mathematical framework to quantify the significance of a label on group interactions of a fixed size s. First of all, for each affinity type $t \in \{1, 2, \ldots, s\}$, they defined the type-(s, t) *affinity score* for label Y as follows:

$$h_{s,t}(Y) := \frac{\sum_{v \in \mathcal{V}_Y} d_{s,t}(v)}{\sum_{v \in \mathcal{V}_Y} d(v)}, \tag{1}$$

where $\mathcal{V}_Y$ denotes the set of nodes whose label is Y, $d(v)$ denotes the degree of node v, and $d_{s,t}(v)$ denotes the number of size s hyperedges containing node v and $t-1$ additional nodes with v's label. It quantifies how frequently nodes with label Y participate in size-s hyperedges that contain exactly t nodes with label Y. The type-(s, t) *baseline score* $b_{s,t}(Y)$ for label Y is the probability that a node with label Y joins a size-s hyperedge containing exactly t nodes with label Y, if $s-1$ other nodes are selected uniformly at random, i.e.,

$$b_{s,t}(Y) := \frac{\binom{|\mathcal{V}_Y|-1}{t-1}\binom{|\mathcal{V}|-|\mathcal{V}_Y|}{s-t}}{\binom{|\mathcal{V}|-1}{s-1}}. \tag{2}$$

We examine the type-(s, t) *affinity ratio score* for label Y, which is the ratio of the affinity score to the baseline score, i.e., $\frac{h_{s,t}(Y)}{b_{s,t}(Y)}$, to evaluate the significance of the combination of t nodes labeled Y in a fixed hyperedge size s.

Since we assume binary attributes, we consider the type-(s, t) affinity ratio score for attribute values 0 and 1 of each node attribute. In this work, we focus on $s \in \{2, 3, 4\}$, as small-size hyperedges occur frequently in many real-world hypergraphs, allowing a valid statistical evaluation. For simplicity, from now, we refer to the type-(s, 1) through type-(s, s) affinity ratio scores as the type-s affinity ratio scores.

(Higher-order) Hyperedge Entropy. Lee et al. [14] observed that hyperedges exhibit label homogeneity in real-world hypergraphs. Compared to hyperedges in randomized hypergraphs, real-world hyperedges comprise nodes with similar labels. Moreover, even after multiple steps of propagation of labels to incident nodes and hyperedges, propagated labels exhibit a similar behavior. To measure the label homogeneity, they utilized *hyperedge entropy*. They compute the entropy of hyperedges based on their original labels as well as the entropy of hyperedges after label propagation (referred to as higher-order hyperedge entropy).

In this work, we consider the distribution of hyperedge entropy and higher-order hyperedge entropy of each node attribute to measure the dominance of attributes on hyperedge formation. If the distribution of entropy is skewed toward 0, it indicates that hyperedges tend to be formulated with nodes that have similar attributes.

Node Homophily Score. We propose a *node homophily score* to quantify the homogeneity of nodes in a hypergraph. Node homophily score of node v at l-th attribute is defined as:

$$n_v[l] := \frac{\sum_{e \in \mathcal{E}_v} |\{u \in e, u \neq v \mid \mathbf{X}[u,l] = \mathbf{X}[v,l]\}|}{\sum_{e \in \mathcal{E}_v} (|e|-1)}, \tag{3}$$

where $n_v[l]$ is a node homophily score of node v for the l-th attribute, $\mathcal{E}_v \subseteq \mathcal{E}$ is a set of hyperedges which contain node v, and $l \in \{1, 2, \ldots, k\}$. This measure indicates the ratio of incident nodes that share the same l-th attribute value with node v, suggesting that a high node homophily score reflects a greater tendency for v to form hyperedges with other nodes having similar attributes. Thus, we consider the distribution of node homophily score for each node attribute to measure the node homogeneity of a hypergraph.

Overall, the above measures offer complementary perspectives on structure-attribute interplay: (1) the **type-t affinity score** quantifies the fine-grained patterns in hyperedge-attribute distributions, (2) the **(higher-order) hyperedge entropy** quantifies the coarse-grained patterns in hyperedge-attribute distributions, and (3) the **node homophily score** quantifies the node-level patterns of attribute distributions.

C. MAG: Multiplicative Attribute Graph Model

Kim and Leskovec [35] proposed the Multiplicative Attribute Graph (MAG) model to capture how node attributes affect edge formation in real-world pairwise graphs. Given a set of nodes $\mathcal{V}$ and an associated node attribute matrix $\mathbf{X} \in \{0,1\}^{|\mathcal{V}| \times k}$ [1], MAG estimates the probability of edges based on node attributes. Specifically, MAG defines a set of attribute affinity matrices $\Theta = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k\}$, where each $\boldsymbol{\theta}_l \in \mathbb{R}^{2\times 2}$ [2] captures the affinity between values of the l-th attribute. Specifically, $\boldsymbol{\theta}_l[x_1, x_2] \in [0, 1]$ represents the affinity between attribute values x_1 and x_2. To compute the probability $P(u, v)$ of an edge between nodes u and v, MAG multiplies the affinities across all k attributes:

$$P(u,v) = \prod_{l=1}^{k} \boldsymbol{\theta}_l[\mathbf{x}_u^{(l)}, \mathbf{x}_v^{(l)}]. \tag{4}$$

Inspired by MAG, which effectively models pairwise interactions using node attributes, we develop a model for group interactions, which require more complex modeling of how node attributes collectively influence group formation. Moreover, compared to MAG, our model additionally captures commonly-observed structural node hierarchies by distinguishing between core and fringe node roles in group formation.

IV. Proposed Generation Method: NoAH

In this section, we introduce NoAH, a novel hypergraph generative model based on node attributes. We first present the intuition behind the hyperedge formation in NoAH, followed by its model details and theoretical analysis.

A. Ideas behind NoAH.

In real-world hypergraphs, node attributes play a crucial role in the formation of hyperedges. For example, in some hypergraphs, hyperedges among nodes with similar attributes are common (homophily) [14], while in others, hyperedges among nodes with dissimilar attributes (heterophily) are common [37]. In NoAH, we aim to reflect such attribute-structure interplay for generating more realistic hypergraphs.

[1] While MAG can consider a general categorical attribute, in most cases simplified MAG model with binary attribute is utilized.

[2] Binary attributes take values in $\{0, 1\}$, resulting in 2×2 affinity matrices.

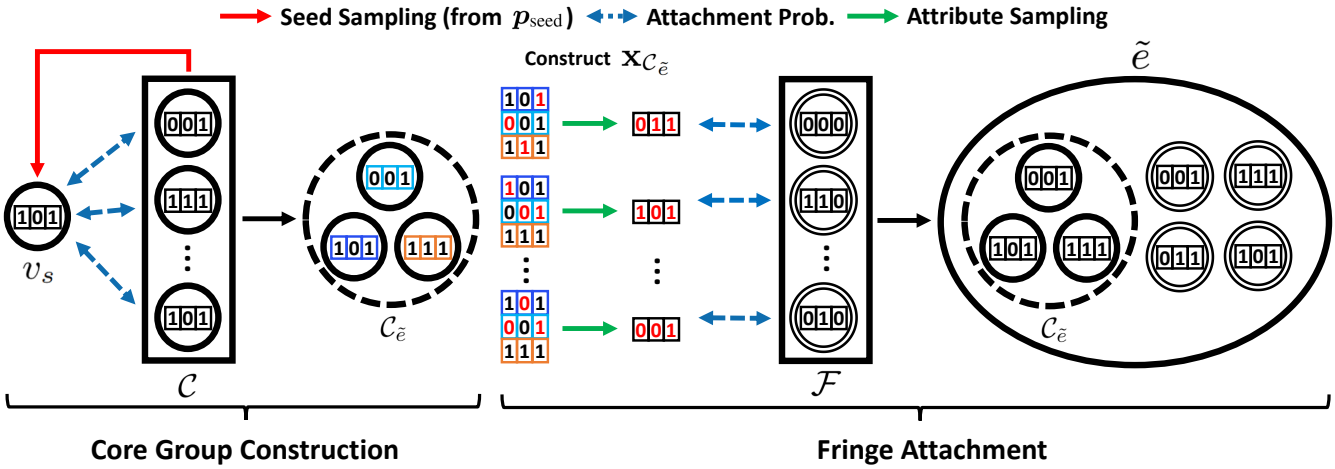


Fig. 2. Hyperedge generation process of NOAH consists of two steps: (1) Core group construction: sample a seed core node v_s according to $\boldsymbol{p}_{\text{seed}}$ and attach additional core nodes to v_s to form a core group $\mathcal{C}_{\tilde{e}}$, and (2) Fringe attachment: attach fringe nodes to core group $\mathcal{C}_{\tilde{e}}$ based on mixed attribute vector $\boldsymbol{x}_{C_{\tilde{e}}}$ obtained by attribute-wise sampling from $\mathbf{X}_{\mathcal{C}_{\tilde{e}}}$. The attached core and fringe nodes, together with the seed node, form a hyperedge.

Idea 1. *We model hyperedge formation using node attributes to reflect their interplay with structure.*

Moreover, many real-world hypergraphs exhibit hierarchical structures [24], [25], where certain nodes consistently play central roles in group interactions, while others interact more peripherally. This hierarchy has been incorporated into several generative models [8], [20], [21]. In NOAH, we introduce a surprisingly simple yet effective approach to reflect hierarchy by dividing nodes into two groups with distinct structural roles.

Idea 2. *We divide nodes into core and fringe nodes to capture the hierarchical structure commonly observed in real-world hypergraphs. Core nodes play a more central role in hyperedge formation, while fringe nodes participate more peripherally.*

Since the number of hyperedge candidates increases exponentially with the number of nodes, generating a hypergraph by considering the formation probabilities of all candidates is computationally intractable. To address this challenge, NOAH incrementally constructs each hyperedge through attachment.

Idea 3. *We model the formation of each hyperedge as a series of attachments of nodes to its seed node(s).*

B. Model Details of NOAH.

Based on Idea 3, NOAH generates a hypergraph by stochastically sampling nodes to attach to each hyperedge, as outlined in Algorithm 1 and illustrated in Figure 2. Moreover, in line with Idea 1, nodes are sampled based on attribute relationships, using the node attributes $\mathbf{X}$. In addition, to incorporate Idea 2, the node set $\mathcal{V}$ is partitioned into two disjoint subsets: the core node set $\mathcal{C} \subseteq \mathcal{V}$ and the fringe node set $\mathcal{F} \subseteq \mathcal{V}$, s.t. $\mathcal{C} \cap \mathcal{F} = \varnothing$ and $\mathcal{C} \cup \mathcal{F} = \mathcal{V}$, and the two groups play distinct roles in hyperedge formation. Specifically, combining all three ideas, NOAH generates each hyperedge through a two-step process. *First*, a core group is formed by sampling a subset of core nodes from $\mathcal{C}$ based on attribute affinities. *Second*, fringe nodes from $\mathcal{F}$ are attached to the core group according to their attribute affinity with the group, completing the hyperedge.

- **Step 1. Core Group Construction (lines 3 – 8):** To initiate hyperedge construction, NOAH begins by forming a subset of core nodes $\mathcal{C}_{\tilde{e}} \subseteq \mathcal{C}$, which plays a role as the *structural nucleus* of the hyperedge. Specifically, it first samples a *seed core node* $v_s \in \mathcal{C}$ according to a probability distribution $\boldsymbol{p}_{\text{seed}} \in [0,1]^{|\mathcal{C}|}$, where $\boldsymbol{p}_{\text{seed}}(v)$ is the probability of selecting node $v \in \mathcal{C}$ as a seed core node. Then, NOAH considers each remaining core node $v_c \in \mathcal{C} \setminus \{v_s\}$ for inclusion in the core group, sampling it with probability based on its attribute affinity with the seed node v_s, computed as:

$$P_{\mathcal{C}}(v_c | v_s, \Theta_{\mathcal{C}}) = \prod\nolimits_{l=1}^{k} \boldsymbol{\theta}_{\mathcal{C}}^{(l)}[\mathbf{x}_s^{(l)}, \mathbf{x}_c^{(l)}], \tag{5}$$

where $\Theta_{\mathcal{C}} = \{\boldsymbol{\theta}_{\mathcal{C}}^{(1)}, \ldots, \boldsymbol{\theta}_{\mathcal{C}}^{(k)}\}$ is the set of attribute affinity matrices modeling interactions *among core nodes*, and each $\boldsymbol{\theta}_{\mathcal{C}}^{(l)} \in \mathbb{R}^{2\times 2}$ captures the affinity between binary values of the l-th attribute.
- **Step 2. Fringe Attachment (lines 9 – 14):** Subsequently, NOAH samples fringe nodes from $\mathcal{F}$ to attach to the core group $\mathcal{C}_{\tilde{e}}$ and complete the hyperedge construction. For each fringe node $v_f \in \mathcal{F}$, NOAH first constructs a binary attribute vector $\mathbf{x}_{\mathcal{C}_{\tilde{e}}} \in \{0,1\}^k$ that summarizes the attributes of the nodes in the core group $\mathbf{x}_{\mathcal{C}_{\tilde{e}}}$. Each l-th attribute $\mathbf{x}_{\mathcal{C}_{\tilde{e}}}^{(l)}$ is independently sampled as:

$$\mathbf{x}_{\mathcal{C}_{\tilde{e}}}^{(l)} \sim \text{Bernoulli}\left(\frac{1}{|\mathcal{C}_{\tilde{e}}|}\sum\nolimits_{v_i \in \mathcal{C}_{\tilde{e}}} \mathbf{x}_i^{(l)}\right), \tag{6}$$

which intuitively, samples the attribute according to its average presence among the group members. Then, v_f is considered for attachment with the following probability:

$$P_{\mathcal{F}}(v_f | \mathcal{C}_{\tilde{e}}, \Theta_{\mathcal{F}}) = \prod\nolimits_{l=1}^{k} \boldsymbol{\theta}_{\mathcal{F}}^{(l)}[\mathbf{x}_{\mathcal{C}_{\tilde{e}}}^{(l)}, \mathbf{x}_f^{(l)}], \tag{7}$$

where $\Theta_{\mathcal{F}} = \{\boldsymbol{\theta}_{\mathcal{F}}^{(1)}, \ldots, \boldsymbol{\theta}_{\mathcal{F}}^{(k)}\}$ is the set of attribute affinity matrices that model interactions *between the core group and fringe nodes*, and each $\boldsymbol{\theta}_{\mathcal{F}}^{(l)} \in \mathbb{R}^{2\times 2}$ captures the affinity between binary values of the l-th attribute.

Algorithm 1: NOAH

Input: (1) number of hyperedges m
(2) node attribute matrix $\mathbf{X} \in \{0,1\}^{|\mathcal{V}| \times k}$
(3) set of core nodes $\mathcal{C}$
(4) set of fringe nodes $\mathcal{F}$
(5) seed core probabilities $\boldsymbol{p}_{\text{seed}}$
(6) set of core group affinity matrices $\Theta_{\mathcal{C}}$
(7) set of fringe attachment affinity matrices $\Theta_{\mathcal{F}}$
Output: generated hypergraph $\tilde{\mathcal{H}} = (\mathcal{V}, \tilde{\mathcal{E}})$

$\tilde{\mathcal{H}} = (\mathcal{V}, \tilde{\mathcal{E}} = \varnothing)$
for ***each*** $i = 1, \cdots, m$ **do**
  // 1. Core Group Construction
  Sample $v_s \sim \boldsymbol{p}_{\text{seed}}$, $v_s \in \mathcal{C}$
  $\mathcal{C}_{\tilde{e}} \leftarrow \{v_s\}$
  for ***each*** $v_c \in \mathcal{C} \setminus \{v_s\}$ **do**
    $p \leftarrow P_{\mathcal{C}}(v_c | v_s, \Theta_{\mathcal{C}})$ ▶ Eq. (5)
    with ***probability*** p **do**
      $\mathcal{C}_{\tilde{e}} \leftarrow \mathcal{C}_{\tilde{e}} \cup \{v_c\}$
  // 2. Fringe Node Attachment
  $\mathcal{F}_{\tilde{e}} \leftarrow \varnothing$
  for ***each*** $v_f \in \mathcal{F}$ **do**
    Construct $\mathbf{x}_{\mathcal{C}_{\tilde{e}}}$ of $\mathcal{C}_{\tilde{e}}$ ▶ Eq. (6)
    $q \leftarrow P_{\mathcal{F}}(v_f | \mathcal{C}_{\tilde{e}}, \Theta_{\mathcal{F}})$ ▶ Eq. (7)
    with ***probability*** q **do**
      $\mathcal{F}_{\tilde{e}} \leftarrow \mathcal{F}_{\tilde{e}} \cup \{v_f\}$
  $\tilde{e} \leftarrow \mathcal{C}_{\tilde{e}} \cup \mathcal{F}_{\tilde{e}}$
  $\tilde{\mathcal{E}} \leftarrow \tilde{\mathcal{E}} \cup \{\tilde{e}\}$
return $\tilde{\mathcal{H}} = (\mathcal{V}, \tilde{\mathcal{E}})$

C. Theoretical Analysis of NOAH.

We analyze complexity and structural properties of NOAH. **Complexity.** For core group construction, sampling the seed core node v_s takes $O(|\mathcal{C}|)$ time, while computing attachment probabilities $P_{\mathcal{C}}$ for the remaining core nodes $\mathcal{C} \setminus \{v_s\}$ requires $O(k|\mathcal{C}|)$ time. For fringe attachment, computing the attachment probabilities $P_{\mathcal{F}}$ for fringe nodes $\mathcal{F}$ takes $O(k|\mathcal{F}|)$ time. NOAH generates a hypergraph $\tilde{\mathcal{H}}$ consisting of m hyperedges by repeating these two steps m times. Thus, the overall time complexity of the hypergraph generation process of NOAH is $O(mk|\mathcal{V}|)$ where $|\mathcal{V}| = |\mathcal{C}| + |\mathcal{F}|$. Regarding space complexity, NOAH requires $O(1)$ space per hyperedge to store the seed core. The sampled attribute of a core group requires $O(k)$ space, which can be discarded after computing the attachment probabilities for the fringe nodes. Thus, the overall space complexity of the hypergraph generation by NOAH is $O(k)$. **Structural Properties.** NOAH is capable of generating hypergraphs with heavy-tailed node degree distributions, which is common in real-world hypergraphs [16], [21].

Theorem 1. *There exist configurations on node attributes and parameters on* NOAH *such that the generated hypergraph follows a power-law degree distribution.*

Proof of Theorem 1 is in Appendix VIII.

V. PROPOSED FITTING METHOD: NOAHFIT

In this section, we propose NOAHFIT, a method for tuning the parameters of NOAH to fit a given hypergraph, outlined in Algorithm 2. The goal of fitting is to enable NOAH to generate hypergraphs that closely resemble the input in both structure and attribute patterns. This is useful for various downstream applications, such as data anonymization and simulation, where generating realistic yet controllable hypergraphs is critical. We begin by describing how the nodes $\mathcal{V}$ is partitioned into core nodes $\mathcal{C}$ and fringe nodes $\mathcal{F}$. Next, we derive hyperedge likelihoods, which are a key component for parameter optimization. Then, we discuss parameter optimization for maximizing the likelihood of the given hypergraph. Finally, we analyze the complexity of NOAHFIT.

A. Core and Fringe Partition

Given a hypergraph $\mathcal{H}$, NOAHFIT partitions the node set $\mathcal{V}$ into a core node set $\mathcal{C}$ and a fringe node set $\mathcal{F}$. While various algorithms can be used for core–fringe splitting, we adopt the concept of a *hitting set*, following [38]. The hitting set of a hypergraph is a set of nodes such that every hyperedge contains at least one node from this set, i.e., a hitting set S satisfies $\forall\ e \in \mathcal{E},\ \exists\ v \in e, v \in S$. In particular, following [38], we use a union of minimal hitting sets that are obtained greedily as the core node set $\mathcal{C}$. Once the core set $\mathcal{C}$ is identified, the fringe set is defined as its complement, i.e., $\mathcal{F} = \mathcal{V} \setminus \mathcal{C}$. Each hyperedge $e \in \mathcal{E}$ consists of a subset of core nodes $\mathcal{C}_e = e \cap \mathcal{C}$ and a subset of fringe nodes $\mathcal{F}_e = e \cap \mathcal{F}$.

B. Derivation of the Likelihood of Each Hyperedge

Given the structural and attribute information of the hypergraph (i.e., $\mathcal{C}$, $\mathcal{F}$, and $\mathbf{X}$) and the parameters of NOAH (i.e., $\boldsymbol{p}_{\text{seed}}$, $\Theta_{\mathcal{C}}$, and $\Theta_{\mathcal{F}}$), we now derive the likelihood of each hyperedge $e \in \mathcal{E}$, denoted as $P(e|\mathcal{C}, \mathcal{F}, \mathbf{X}, \boldsymbol{p}_{\text{seed}}, \Theta_{\mathcal{C}}, \Theta_{\mathcal{F}})$, which we refer to as $P(e)$ for brevity. The likelihood is decomposed into two components: (1) $P_{\text{core}}(e)$, the likelihood of core group construction, and (2) $P_{\text{fringe}}(e)$, the likelihood of fringe node attachment. The total likelihood is then given by $P(e) = P_{\text{core}}(e) \cdot P_{\text{fringe}}(e)$.

Likelihood of Sampling Core Nodes. Given a hyperedge e, the likelihood $P_{\text{core}}(e)$ of its core group $\mathcal{C}_e$ is:

$$P_{\text{core}}(e) = \sum\nolimits_{v_s \in \mathcal{C}_e} \boldsymbol{p}_{\text{seed}}(v_s) \cdot P_{\mathcal{C}}(\mathcal{C}_e \setminus \{v_s\} | v_s), \quad [3] \qquad (8)$$

where $P_{\mathcal{C}}(\mathcal{C}_e \setminus \{v_s\} | v_s)$ is the likelihood of sampling the remaining core nodes $\mathcal{C}_e \setminus \{v_s\} \subset \mathcal{C}$ given the seed node v_s, which can be written as:

$$P_{\mathcal{C}}(\mathcal{C}_e \setminus \{v_s\} | v_s) = \prod_{v_c \in \mathcal{C}_e \setminus \{v_s\}} P_{\mathcal{C}}(v_c | v_s) \cdot \prod_{v_c \in \mathcal{C} \setminus \mathcal{C}_e} (1 - P_{\mathcal{C}}(v_c | v_s)).$$

For brevity, we omit $\Theta_{\mathcal{C}}$ in the above equations.

Likelihood of Sampling Fringe Nodes. Given the core group $\mathcal{C}_e$ of the hyperedge e, the likelihood of its fringe subset $\mathcal{F}_e$ is:

$$P_{\text{fringe}}(e) = \prod_{v_f \in \mathcal{F}_e} P_{\mathcal{F}}(v_f | \mathcal{C}_e) \cdot \prod_{v_f \in \mathcal{F} \setminus \mathcal{F}_e} (1 - P_{\mathcal{F}}(v_f | \mathcal{C}_e)), \qquad (9)$$

where $P_{\mathcal{F}}(v_f | \mathcal{C}_e)$ is the probability of attaching fringe node v_f to the core group $\mathcal{C}_e$. This probability is obtained by

[3] For training stability, in practice, we implement NOAHFIT by using a normalized core group likelihood, where $P_{\text{core}}(e)$ is divided by the total seed probability over the core group, i.e., $P_{\text{core}}(e) / \sum_{v_c \in \mathcal{C}_e} \boldsymbol{p}_{\text{seed}}(v_c)$.

marginalizing over the stochastic binary attribute vector $\mathbf{x}_{\mathcal{C}_e}$ of $\mathcal{C}_e$ as follows:

$$P_{\mathcal{F}}(v_f|\mathcal{C}_e) = \mathbb{E}_{\mathbf{x}_{\mathcal{C}_e}}\left[\prod_{l=1}^{k} \boldsymbol{\theta}_{\mathcal{F}}^{(l)}[\mathbf{x}_{\mathcal{C}_e}^{(l)}, \mathbf{x}_f^{(l)}]\right]$$
$$= \prod_{l=1}^{k}\left[(1-p_e^{(l)})\cdot \boldsymbol{\theta}_{\mathcal{F}}^{(l)}[0, \mathbf{x}_f^{(l)}] + p_e^{(l)}\cdot \boldsymbol{\theta}_{\mathcal{F}}^{(l)}[1, \mathbf{x}_f^{(l)}]\right],$$

where each l-th component of $\mathbf{x}_{\mathcal{C}_e}$ is sampled as a Bernoulli random variable with mean equal to $p_e^{(l)} = \frac{1}{|\mathcal{C}_e|}\sum_{v_i\in\mathcal{C}_e}\mathbf{x}_i^{(l)}$.

C. Update of the Parameters of NOAH

We present the loss functions used to update the parameters of NOAH (spec., $\boldsymbol{p}_{\text{seed}}$, $\Theta_{\mathcal{C}}$, and $\Theta_{\mathcal{F}}$) as follows:

- **Negative Log-likelihood Loss ($\mathcal{L}_{edge}$):** For each hyperedge e, we calculate the negative log-likelihood, i.e., $-\log P(e)$. These values are then summed over all hyperedges to compute the negative log-likelihood loss:

$$\mathcal{L}_{\text{edge}} = \sum\nolimits_{e\in\mathcal{E}} -\log P(e).$$

- **Degree and Cardinality Losses ($\mathcal{L}_{deg}$ and $\mathcal{L}_{card}$):** Unlike many existing hypergraph models that rely on degree or cardinality distributions as explicit model inputs, NOAH learns to reproduce realistic degree and cardinality patterns, without using them as model inputs (only using them for fitting), demonstrating its expressive modeling capability. To encourage this behavior, we use a mean squared error (MSE) loss that aligns the expected degrees and hyperedge sizes with the distributions in the original hypergraph:

$$\mathcal{L}_{deg} = \text{MSE}(\boldsymbol{d}_c, \tilde{\boldsymbol{d}}_c) + \text{MSE}(\boldsymbol{d}_f, \tilde{\boldsymbol{d}}_f)$$
$$\mathcal{L}_{card} = \text{MSE}(\boldsymbol{c}_c, \tilde{\boldsymbol{c}}_c) + \text{MSE}(\boldsymbol{c}_f, \tilde{\boldsymbol{c}}_f),$$

  where $\boldsymbol{d}_c, \boldsymbol{d}_f$ denote the degrees of core and fringe nodes; and $\boldsymbol{c}_c, \boldsymbol{c}_f$ denote the cardinalities of core and fringe subsets within hyperedges. The corresponding quantities with tildes indicate their expected values under NOAH. We compare distributions by sorting them and computing the MSE between corresponding values, focusing on overall distributional similarity rather than individual node identities.

The final loss is a weighted sum of the above losses (see Line 12 of Algorithm 2), with the weights as hyperparameters.

D. Complexity of NOAHFIT.

We provide the time and space complexity analysis of NOAHFIT. Regarding the time complexity, the core and fringe partitioning [38] requires $O(m|\mathcal{V}|)$ time. For each hyperedge $e \in \mathcal{E}$, computing its likelihood $P(e)$ takes $O(k|\mathcal{V}||\mathcal{C}_e|)$ time. With attachment probabilities calculated during the computation of $P(e)$, calculation of expected degree and cardinality from NOAH takes $O(|\mathcal{V}|)$ time. Repeating this for T training epochs over all m hyperedges, the total computation cost becomes $O(Tk|\mathcal{V}|\sum_{e\in\mathcal{E}}|\mathcal{C}_e|)$. Regarding the space complexity, core and fringe partition takes $O(|\mathcal{V}|)$ space. For each hyperedge $e \in \mathcal{E}$, computing its likelihood $P(e)$ takes $O(k + |\mathcal{V}|)$ space. Expected degree and cardinality require $O(m + |\mathcal{V}|)$ space. Thus, for the entire hypergraph, the total space complexity is $O(k + m + |\mathcal{V}|)$.

Algorithm 2: NOAHFIT

Input: (1) target hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$
(2) number of epochs T, learning rate η
(3) loss weights w_{deg}, w_{card}
Output: (1) set of core fringe nodes $\mathcal{C}$
(2) set of core fringe nodes $\mathcal{F}$
(3) seed core probabilities $\boldsymbol{p}_{\text{seed}}$
(4) set of core group affinity matrices $\Theta_{\mathcal{C}}$
(5) set of fringe attachment affinity matrices $\Theta_{\mathcal{F}}$

```
// 1. Core-fringe Split
Split V into C, F
Initialize p_seed, Θ_C and Θ_F [4]
for each t = 1, ..., T do
    // 2. Log-likelihood Loss
    L_edge ← 0
    for each e ∈ E do
        C_e ← e ∩ C, F_e ← e ∩ F
        P(e) ← P_core(e) · P_fringe(e)      ▶ Eq. (8) & Eq. (9)
        L_edge ← L_edge − log P(e)
    // 3. Degree, Cardinality Losses
    Compute d̃_c, d̃_f and c̃_c, c̃_f using p_seed, Θ_C and Θ_F
    L_deg ← MSE(d_c, d̃_c) + MSE(d_f, d̃_f)
    L_card ← MSE(c_c, c̃_c) + MSE(c_f, c̃_f)
    L ← L_edge + w_deg · L_deg + w_card · L_card
    // 4. Parameter Update
    p_seed ← p_seed + η∇_{p_seed} L
    Θ_C ← Θ_C + η∇_{Θ_C} L
    Θ_F ← Θ_F + η∇_{Θ_F} L
return C, F, p_seed, Θ_C, Θ_F
```

VI. EXPERIMENTS

In this section, we present experimental results demonstrating the effectiveness of NOAH and NOAHFIT.

A. Experimental Settings

Datasets. We use nine real-world hypergraphs from four distinct domains (see Table II for some statistics):

- **Academic Paper Domain (Citeseer, Cora [40]):** Each node is an academic paper. For the Citeseer dataset, each hyperedge is a set of papers co-cited by a paper, and for the Cora dataset, each hyperedge is a set of papers (co-)authored by the same author. Node attributes are binary bag-of-words attributes, indicating whether each paper includes each keyword or not.
- **Contact domain (High School [41], Workspace [42]):** Each node is an individual, and each hyperedge is a group of individuals who were in contact with one another during a time interval. For the High School dataset, node attributes include gender, class affiliation, and Facebook account ownership. For the Workspace dataset, node attributes represent the department to which the worker belongs. Since all attributes are categorical, we apply one-hot encoding to convert them into binary attributes.
- **Review Domain (Amazon Music [43], Yelp Restaurant, Yelp Bar [44]):** Each node is a reviewer, and each hyperedge

[4] Initialization method of parameters is explained in Appendix IX [39].

TABLE II
SUMMARY STATISTICS OF 9 REAL-WORLD HYPERGRAPHS FROM 4 DOMAINS. $|\mathcal{V}|$: THE NUMBER OF NODES. $|\mathcal{E}|$: THE NUMBER OF HYPEREDGES. k: THE NUMBER OF ATTRIBUTES. THE CORE-SET SIZE $|\mathcal{C}|$ AND THE FRINGE-SET SIZE $|\mathcal{F}|$ ARE OBTAINED BY NOAHFIT.

Dataset	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	k	$\|\mathcal{C}\|$	$\|\mathcal{F}\|$
Citeseer	1,458	1,079	3,703	597	861
Cora	2,388	1,072	1,433	841	1,547
High School	327	7,818	12	288	39
Workspace	92	788	5	71	21
Amazon Music	1,106	686	7	379	727
Yelp Restaurant	565	594	9	273	292
Yelp Bar	1,234	1,188	15	625	609
Devops	5,010	5,684	429	2,003	3,007
Patents	4,458	4,669	2,170	894	3,564

is a group of reviewers who reviewed a certain product or business. Node attributes indicate the types of products or businesses that each reviewer has reviewed at least once.

- **Online Q&A Domain**[5]**(Devops, Patents):** Each node represents a user on Stack Exchange, and each hyperedge corresponds to a post involving a set of users. Node attributes indicate the set of tags associated with the posts each user has participated in.

Note that our experiments are done with varying numbers of attributes, scaling from 5 (Workspace) to 3,703 (Citeseer).

Baselines. We consider eight baseline generative models: HYPERCL [7], HYPERPA [6], HYPERFF [45], HYPERLAP [7], hyper dK-series [46], THERA [8], HYCOSBM [11], HYREC [47]. As discussed in Section II, deep learning–based models are not direct competitors and are therefore excluded from comparison. Among the baselines, HYCOSBM explicitly utilizes node attributes. For HYPERCL, HYPERLAP, and hyper dK-series, since node identities are preserved in the generated hypergraphs, we assign node attributes based on their correspondence to the original nodes. For HYPERPA, HYPERFF, THERA, and HYREC, where node identities are not preserved during the generation process, we assign node attributes randomly. The hyperparameter search spaces for our method and the baselines are detailed in Appendix X [39].

Evaluation. We compare NOAH and the baselines in terms of their ability to reproduce the structure-attribute interplay observed in real-world hypergraphs based on the metrics described in Section III-B. For type-s affinity ratio scores, we evaluate hypergraph generators by comparing the sum of log scaled differences between the ground truth and the generated hypergraphs over all $t \in [1, s]$ and all attributes. For (higher-order) hyperedge entropy and node homophily score, since they are presented as distributions for each attribute, we calculated the sum of Wasserstein Distance (also known as Earth mover's Distance) between the ground truth and the generated hypergraphs. We denote type-s affinity scores as **Ts**, hyperedge entropy as **HE**, higher-order hyperedge entropy as **HOHE**, and node homophily score as **NHS**. For each metric and dataset, we compute both the raw values and the rankings of all compared models.

[5]https://archive.org/download/stackexchange

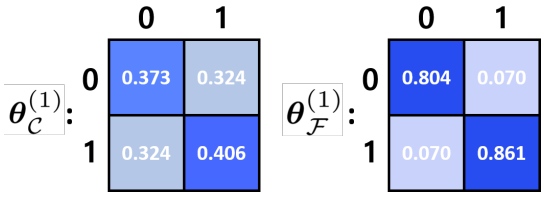


Fig. 3. $\theta_{\mathcal{C}}^{(1)}$ and $\theta_{\mathcal{F}}^{(1)}$ estimated by NOAHFIT on the Amazon Music dataset. NOAHFIT captures homophily by assigning higher affinities to same attribute value pairs ($0 \leftrightarrow 0$ and $1 \leftrightarrow 1$) than to different attribute value pairs ($0 \leftrightarrow 1$).

Machines. We conducted all experiments on a server with RTX A6000 GPUs.

B. Performance Comparison

As shown in Table III, NOAH achieves the best average rank in 5 out of 9 datasets. Averaged over nine datasets, NOAH ranks first in four metrics (type-3 affinity score, type-4 affinity score, hyperedge entropy, and node homophily score), second in type-2 affinity score, and third in higher-order hyperedge entropy. These results demonstrate the overall superiority of NOAH in capturing the structure–attribute interplay of real-world hypergraphs. The relatively low performance of NOAH in the online Q&A domain datasets (Devops and Patents) is likely due to the weak correlation between attribute and structure in these datasets. In the datasets, baselines that explicitly preserve degree and size distributions, including HYPERCL, HYPERLAP, and hyper dK-series, perform well.

C. Case Study

To gain deeper insight into the effectiveness of NOAH, we conduct a case study on the Amazon Music dataset [43]. In the Amazon Music dataset, a value of the first node attribute indicates whether a reviewer has reviewed music in the New York Blues genre (1) or not (0). For this attribute, we compare the structure-attribute interplay metrics between (1) the ground truth hypergraph, (2) the hypergraph generated by HYPERCL, and (3) the hypergraph generated by NOAH with NOAHFIT. As shown in Figure 1 in Section I, the distributions of hyperedge entropy and node homophily scores are highly skewed toward 0 and 1, respectively. Additionally, the type-(4, 4) affinity ratio score is high for both attribute values 0 and 1, indicating that many size 4 hyperedges consist of nodes sharing the same value for the first attribute. These results suggest that nodes in the Amazon Music dataset exhibit strong homophily with respect to the first attribute, and that the node attribute plays a crucial role in hypergraph formation. Whereas the baseline model, HYPERCL, fails to capture this structure–attribute interplay, NOAH successfully captures it through the use of affinity matrices shown in Figure 3.

D. Ablation Study

To assess the contribution of the core–fringe node hierarchy in NOAH, we compare its performance with a variant, NOAH-CF, which omits this hierarchical structure. In NOAH-CF, each hyperedge is generated by sampling a seed node from the entire node set and attaching additional nodes directly, without distinguishing between core and fringe roles (and

TABLE III

NOAH REPRODUCES STRUCTURE-ATTRIBUTE INTERPLAYS OVERALL BEST ACROSS 9 DATASETS. TOP THREE RESULTS ARE HIGHLIGHTED IN BLUE (FIRST), GREEN (SECOND), AND YELLOW (THIRD). REFER TO SECTION VI-D FOR NOAH-CF, A VARIANT OF NOAH. A.R. DENOTES AVERAGE RANK.

	T2	T3	T4	HE	HOHE	NHS	A.R.
HYPERCL	6,816	10,702	10,672	19.81	122.02	19.94	6.7
HYPERPA	6,871	10,739	10,559	25.90	103.76	22.46	7.7
HYPERFF	6,472	10,483	10,677	17.60	66.97	15.23	4.0
HYPERLAP	6,757	10,737	10,311	18.33	55.40	19.43	5.0
hyper dK-series	6,968	10,234	10,154	13.38	102.83	16.09	3.7
THERA	6,450	10,498	10,476	18.04	53.58	18.59	3.8
HYCOSBM	6,285	10,613	10,278	19.89	135.30	38.60	6.0
HYREC	6,996	10,840	10,271	20.01	51.37	20.04	6.2
NOAH	5,734	10,157	10,151	24.31	44.26	15.39	2.3
NOAH-CF	9,036	14,550	13,515	48.26	106.32	121.04	9.7

(a) Citeseer (NOAH ranks **first** overall)

	T2	T3	T4	HE	HOHE	NHS	A.R.
HYPERCL	2,099	4,206	4,481	8.02	63.64	7.99	7.0
HYPERPA	2,081	4,315	4,615	14.47	62.62	12.78	8.3
HYPERFF	2,040	4,094	4,518	8.06	39.58	6.76	5.0
HYPERLAP	2,031	4,083	4,523	7.65	53.40	7.13	5.2
hyper dK-series	2,094	4,048	4,502	7.24	59.79	7.49	5.5
THERA	2,010	4,003	4,492	6.94	40.35	7.20	2.8
HYCOSBM	1,947	4,049	4,502	7.24	69.77	15.02	5.5
HYREC	2,075	4,163	4,465	6.51	19.56	6.54	3.2
NOAH	2,058	4,011	4,517	6.41	41.64	5.79	3.2
NOAH-CF	3,322	5,951	5,995	29.01	57.35	70.15	9.3

(b) Cora (NOAH ranks **second** overall)

	T2	T3	T4	HE	HOHE	NHS	A.R.
HYPERCL	20.4	51.4	106.4	1.180	1.369	1.730	7.2
HYPERPA	20.6	52.6	107.8	1.175	1.421	1.654	8.0
HYPERFF	20.2	61.6	106.5	1.017	1.290	1.785	7.8
HYPERLAP	20.3	51.6	102.0	1.187	0.977	1.714	6.0
hyper dK-series	20.0	51.4	97.8	0.931	1.429	1.716	5.8
THERA	19.9	51.2	99.8	1.166	0.781	1.611	3.5
HYCOSBM	3.9	52.1	97.9	1.819	1.414	1.797	6.7
HYREC	19.8	54.0	92.8	0.740	0.960	1.637	4.3
NOAH	12.2	37.8	89.6	0.628	1.273	1.374	2.0
NOAH-CF	19.2	50.3	103.9	0.682	1.176	1.635	3.7

(c) High School (NOAH ranks **first** overall)

	T2	T3	T4	HE	HOHE	NHS	A.R.
HYPERCL	7.5	17.3	13.0	0.599	0.181	0.890	7.0
HYPERPA	7.1	19.5	12.5	0.562	0.217	0.835	6.0
HYPERFF	6.3	14.9	13.9	0.445	0.399	0.801	5.2
HYPERLAP	7.3	19.9	12.6	0.564	0.184	0.878	7.0
hyper dK-series	6.5	15.6	25.1	0.408	0.178	0.814	5.2
THERA	5.7	13.7	16.7	0.542	0.144	0.820	4.2
HYCOSBM	2.2	13.0	24.0	0.705	0.233	0.885	6.2
HYREC	7.3	18.0	20.3	0.371	0.301	0.753	6.2
NOAH	4.2	9.7	20.0	0.062	0.133	0.619	2.2
NOAH-CF	7.3	18.4	12.5	0.186	0.283	0.905	6.0

(d) Workspace (NOAH ranks **first** overall)

	T2	T3	T4	HE	HOHE	NHS	A.R.
HYPERCL	27.3	53.0	63.6	1.016	0.382	1.053	6.8
HYPERPA	27.3	55.4	71.2	1.154	0.527	1.096	8.8
HYPERFF	24.1	54.3	60.5	0.449	0.299	1.055	5.2
HYPERLAP	27.3	52.4	68.3	1.026	0.361	1.042	6.3
hyper dK-series	31.4	52.4	61.6	1.249	0.450	1.026	7.3
THERA	26.0	50.6	67.0	0.976	0.394	1.003	5.0
HYCOSBM	11.8	57.9	72.1	0.306	0.371	0.900	4.7
HYREC	25.3	50.6	61.8	1.138	0.402	0.982	5.3
NOAH	21.0	47.8	55.1	0.275	0.394	0.229	1.8
NOAH-CF	21.8	49.7	58.0	0.363	1.188	0.402	3.7

(e) Amazon Music (NOAH ranks **first** overall)

	T2	T3	T4	HE	HOHE	NHS	A.R.
HYPERCL	17.8	58.9	90.4	0.979	0.249	1.148	6.5
HYPERPA	24.8	55.9	76.0	0.786	0.591	1.068	6.3
HYPERFF	22.4	54.1	79.2	0.459	0.603	1.157	5.5
HYPERLAP	22.4	54.6	85.9	1.012	0.217	1.129	6.0
hyper dK-series	24.6	54.5	82.9	0.726	0.511	0.978	5.2
THERA	21.0	54.3	80.6	0.704	0.484	0.964	4.3
HYCOSBM	5.0	52.2	102.5	0.281	0.403	0.932	3.0
HYREC	24.7	55.7	77.6	0.797	0.523	0.968	5.7
NOAH	9.3	53.0	66.2	0.967	1.670	0.524	3.7
NOAH-CF	21.6	66.5	86.3	1.232	3.129	1.539	8.8

(f) Yelp Restaurant (NOAH ranks **second** overall)

	T2	T3	T4	HE	HOHE	NHS	A.R.
HYPERCL	57.1	112.1	133.2	1.392	1.064	1.343	7.0
HYPERPA	54.7	114.4	139.9	1.090	1.391	1.283	6.8
HYPERFF	54.6	115.2	129.4	0.635	0.700	1.430	4.8
HYPERLAP	54.1	107.2	137.0	1.345	0.978	1.325	5.2
hyper dK-series	57.1	108.7	134.2	1.572	1.102	1.230	6.0
THERA	55.5	106.0	133.3	1.124	1.061	1.221	4.3
HYCOSBM	26.3	128.9	161.3	0.526	1.053	1.140	4.7
HYREC	58.3	111.3	123.8	1.343	0.953	1.225	5.0
NOAH	44.7	94.0	145.2	0.589	1.598	0.468	4.0
NOAH-CF	61.7	113.9	127.9	0.827	2.955	1.630	7.2

(g) Yelp Bar (NOAH ranks **first** overall)

	T2	T3	T4	HE	HOHE	NHS	A.R.
HYPERCL	1,257	3,450	5,793	4.89	30.86	8.45	3.3
HYPERPA	2,428	6,670	9,427	26.22	66.76	20.3	9.0
HYPERFF	2,382	6,605	9,204	25.71	66.74	20.89	8.0
HYPERLAP	1,246	3,363	5,780	4.82	30.63	8.99	2.7
hyper dK-series	1,136	2,765	4,787	6.86	28.56	10.27	2.3
THERA	2,387	6,567	9,136	25.9	65.53	20.52	7.5
HYCOSBM	1,226	2,696	7,027	4.52	53.83	25.43	3.8
HYREC	2,394	6,736	8,943	23.31	59.88	15.96	7.2
NOAH	1,714	5,182	8,038	12.81	19.75	6.59	3.7
NOAH-CF	2,158	6,330	9,199	30.05	77.31	14.22	7.5

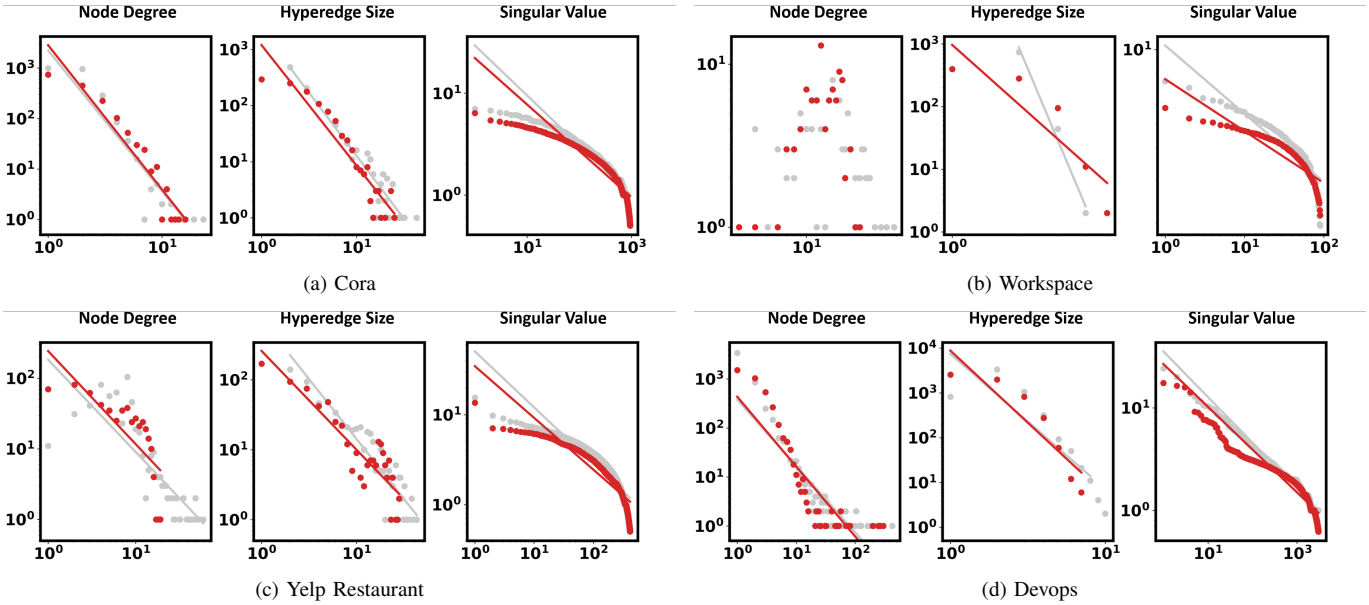
(h) Devops (NOAH ranks **fourth** overall)

	T2	T3	T4	HE	HOHE	NHS	A.R.
HYPERCL	6,586	13,091	16,515	64.30	105.59	103.33	2.5
HYPERPA	13,538	27,184	31,883	268.45	466.07	233.28	9.0
HYPERFF	12,069	26,032	29,718	262.36	440.24	230.16	6.2
HYPERLAP	6,551	13,417	16,663	61.98	104.40	104.40	2.5
hyper dK-series	6,262	15,041	23,320	53.40	85.05	91.81	2.5
THERA	12,856	26,900	30,711	266.45	464.98	232.26	7.8
HYCOSBM	5,828	13,984	18,112	48.52	265.48	230.81	3.3
HYREC	14,769	26,521	29,705	263.42	442.54	228.55	7.0
NOAH	7,292	17,557	22,532	65.32	109.78	134.08	4.5
NOAH-CF	14,298	27,920	31,497	268.81	468.97	236.31	9.7

(i) Patents (NOAH ranks **fifth** overall)

	T2	T3	T4	HE	HOHE	NHS	A.R.
HYPERCL	6.9	5.7	5.0	6.6	5.3	6.6	6.2
HYPERPA	7.8	8.6	7.4	7.7	8.0	7.2	9.7
HYPERFF	5.2	6.3	6.2	5.1	5.4	6.1	5.7
HYPERLAP	5.3	5.3	5.3	6.2	2.8	5.6	4.7
hyper dK-series	6.1	3.9	4.7	4.3	5.4	4.6	3.8
THERA	4.9	4.0	5.8	5.2	4.0	5.0	3.8
HYCOSBM	1.2	4.8	6.4	4.1	6.0	6.7	5.3
HYREC	7.7	7.1	3.9	5.6	5.1	4.0	5.0
NOAH	2.9	2.1	3.9	3.4	4.4	1.4	1.5
NOAH-CF	7.0	7.2	6.3	6.8	8.4	7.9	9.0

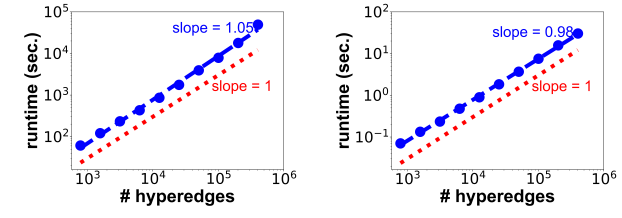
(j) Average Rank over Nine Datasets (NOAH ranks **first** overall)



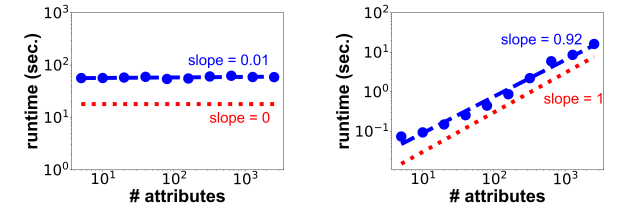
(a) Cora (b) Workspace

(c) Yelp Restaurant (d) Devops

Fig. 4. Structural patterns of real-world hypergraphs (gray) and those generated by NOAH tuned by NOAHFIT (red). These results demonstrate that, despite its focus on structure–attribute interplay, NOAH also successfully reproduces purely structural patterns.



(a) NOAHFIT (fitting) w.r.t. number of hyperedge (b) NOAH (generation) w.r.t. number of hyperedge



(c) NOAHFIT (fitting) w.r.t. number of attributes (d) NOAH (generation) w.r.t. number of attributes

Fig. 5. NOAHFIT scales nearly linearly with the number of hyperedges and remains almost constant with respect to the number of attributes. NOAH scales nearly linearly with both the number of hyperedges and attributes..

therefore without collective consideration of multiple core nodes). As presented in Table III, NOAH consistently outperforms NOAH-CF across all nine datasets, highlighting the importance of the core–fringe hierarchy in capturing realistic hyperedge formation.

E. Scalability Analysis

We evaluate the scalability of our fitting algorithm, NOAHFIT, and our generative model, NOAH, with respect to the number of hyperedges and attributes. To this end, we scale up the Contact-Workspace dataset by factors ranging from 2 to 512. Figure 5 shows the runtime of fitting and generation as functions of the number of hyperedges and attributes, plotted on a log–log scale. The proposed fitting algorithm, NOAHFIT, scales nearly linearly with the number of hyperedges and remains almost constant with respect to the number of attributes. The proposed generative model, NOAH, scales near-linearly with both the number of hyperedges and the number of attributes. These results demonstrate the scalability of both the fitting and generation components of our framework.

F. Further Analysis

Additionally, we examine the structural patterns of hypergraphs generated by NOAH in terms of (1) node degrees, (2) hyperedge sizes, and (3) singular values. These properties are known to follow heavy-tailed distributions in real-world hypergraphs [21]. In Figure 4, we visually demonstrate that NOAH with NOAHFIT reproduces such structural patterns, which closely resemble those of the input hypergraphs. We also present a detailed evaluation on structural metrics in Appendix XI [39], where we show that, despite its focus on structure–attribute interplay, NOAH achieves competitive results on purely structural patterns (with an average rank of 4.6 among 9 structure-focused methods).

VII. CONCLUSIONS

In this work, we proposed NOAH, a stochastic generative model for attributed hypergraphs that reproduces realistic interplay between structure and node attributes. By leveraging a two-level node hierarchy, core and fringe nodes, NOAH formulates hyperedge generation as sequential attachment of nodes (first cores, then fringes), where attachment probabilities are governed by node attributes. We also introduced NOAHFIT, a parameter estimation algorithm that fits NOAH to a given hypergraph by estimating affinity matrices and seed core probabilities. Through extensive experiments on nine real-world hypergraphs across four diverse domains, we demonstrated that NOAH with NOAHFIT more accurately reproduces the structure–attribute interplay than eight existing hypergraph generative models across six metrics. For repro-

ducibility, we make the source code and data publicly available at anonymous.4open.science/r/NoAH-246E.

Relevance to Data Mining and Broad Impact. The proposed framework aligns with the fundamental goal of data mining: finding models that explain complex, large-scale data. By generating realistic data, it supports diverse applications in domains where hypergraphs naturally arise, enabling statistical analysis, simulation, and data anonymization.

REFERENCES

[1] T. Kumar, S. Vaidyanathan, H. Ananthapadmanabhan, S. Parthasarathy, and B. Ravindran, "Hypergraph clustering by iteratively reweighted modularity maximization," *Applied Network Science*, vol. 5, no. 1, p. 52, 2020.

[2] K. Hayashi, S. G. Aksoy, C. H. Park, and H. Park, "Hypergraph random walks, laplacians, and clustering," in *CIKM*, 2020.

[3] J. Yu, D. Tao, and M. Wang, "Adaptive hypergraph learning and its application in image classification," *TIP*, vol. 21, no. 7, pp. 3262–3272, 2012.

[4] J. Silva and R. Willett, "Hypergraph-based anomaly detection of high-dimensional co-occurrences," *TPAMI*, vol. 31, no. 3, pp. 563–569, 2008.

[5] J. Chun, G. Lee, K. Shin, and J. Jung, "Random walk with restart on hypergraphs: fast computation and an application to anomaly detection," *Data Mining and Knowledge Discovery*, vol. 38, no. 3, pp. 1222–1257, 2024.

[6] M. T. Do, S. Yoon, B. Hooi, and K. Shin, "Structural patterns and generative models of real-world hypergraphs," in *KDD*, 2020.

[7] G. Lee, M. Choe, and K. Shin, "How do hyperedges overlap in real-world hypergraphs?-patterns, measures, and generators," in *WWW*, 2021.

[8] S. Kim, F. Bu, M. Choe, J. Yoo, and K. Shin, "How transitive are real-world group interactions?-measurement and reproduction," in *KDD*, 2023.

[9] D. Ghoshdastidar and A. Dukkipati, "Consistency of spectral hypergraph partitioning under planted partition model," *The Annals of Statistics*, vol. 45, no. 1, pp. 289–315, 2017.

[10] N. Ruggeri, M. Contisciani, F. Battiston, and C. De Bacco, "Community detection in large hypergraphs," *Science Advances*, vol. 9, no. 28, p. eadg9159, 2023.

[11] A. Badalyan, N. Ruggeri, and C. De Bacco, "Structure and inference in hypergraphs with node attributes," *Nature Communications*, vol. 15, no. 1, p. 7073, 2024.

[12] M. Contisciani, F. Battiston, and C. De Bacco, "Inference of hyperedges and overlapping communities in hypergraphs," *Nature Communications*, vol. 13, no. 1, p. 7229, 2022.

[13] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual Review of Sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[14] G. Lee, S. Y. Lee, and K. Shin, "Villain: Self-supervised learning on homogeneous hypergraphs without features via virtual label propagation," in *WWW*, 2024.

[15] D. Chakrabarti and C. Faloutsos, "Graph mining: Laws, generators, and algorithms," *ACM Computing Surveys*, vol. 38, no. 1, pp. 2–es, 2006.

[16] G. Lee, F. Bu, T. Eliassi-Rad, and K. Shin, "A survey on hypergraph mining: Patterns, tools, and generators," *ACM Computing Surveys*, vol. 57, no. 8, pp. 1–36, 2025.

[17] S.-H. Lim, S. Lee, S. S. Powers, M. Shankar, and N. Imam, "Survey of approaches to generate realistic synthetic graphs," Oak Ridge National Lab, Tech. Rep., 10 2016.

[18] E. M. Airoldi, D. Blei, S. Fienberg, and E. Xing, "Mixed membership stochastic blockmodels," *NeurIPS*, 2008.

[19] J. Jia and A. R. Benson, "Random spatial network models for core-periphery structure," in *WSDM*, 2019.

[20] M. Papachristou and J. Kleinberg, "Core-periphery models for hypergraphs," in *KDD*, 2022.

[21] J. Ko, Y. Kook, and K. Shin, "Growth patterns and models of real-world hypergraphs," *Knowledge and Information Systems*, vol. 64, no. 11, pp. 2883–2920, 2022.

[22] S. B. Seidman, "Network structure and minimum degree," *Social Networks*, vol. 5, no. 3, pp. 269–287, 1983.

[23] S. P. Borgatti and M. G. Everett, "Models of core/periphery structures," *Social Networks*, vol. 21, no. 4, pp. 375–395, 2000.

[24] F. Bu, G. Lee, and K. Shin, "Hypercore decomposition for non-fragile hyperedges: concepts, algorithms, observations, and applications," *Data Mining and Knowledge Discovery*, vol. 37, no. 6, p. 2389–2437, 2023.

[25] F. Tudisco and D. J. Higham, "Core-periphery detection in hypergraphs," *SIAM Journal on Mathematics of Data Science*, vol. 5, no. 1, pp. 1–21, 2023.

[26] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[27] M. Simonovsky and N. Komodakis, "Graphvae: Towards generation of small graphs using variational autoencoders," in *ICANN*, 2018.

[28] A. Bojchevski, O. Shchur, D. Zügner, and S. Günnemann, "Netgan: Generating graphs via random walks," in *ICML*, 2018.

[29] N. De Cao and T. Kipf, "Molgan: An implicit generative model for small molecular graphs," *ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

[30] C. Liu, W. Fan, Y. Liu, J. Li, H. Li, H. Liu, J. Tang, and Q. Li, "Generative diffusion models on graphs: Methods and applications," in *IJCAI*, 2023.

[31] D. Gailhard, E. Tartaglione, L. Naviner, and J. H. Giraldo, "Hygene: A diffusion-based hypergraph generation method," in *AAAI*, 2025.

[32] ——, "Feature-aware hypergraph generation via next-scale prediction," *arXiv preprint arXiv:2506.01467*, 2025.

[33] G. Robins, P. Pattison, Y. Kalish, and D. Lusher, "An introduction to exponential random graph (p*) models for social networks," *Social Networks*, vol. 29, no. 2, pp. 173–191, 2007.

[34] S. Wang, S. Paul, and P. De Boeck, "Joint latent space model for social networks with multivariate attributes," *Psychometrika*, vol. 88, no. 4, pp. 1197–1227, 2023.

[35] M. Kim and J. Leskovec, "Multiplicative attribute graph model of real-world networks," *Internet Mathematics*, vol. 8, no. 1-2, pp. 113–160, 2012.

[36] N. Veldt, A. R. Benson, and J. Kleinberg, "Combinatorial characterizations and impossibilities for higher-order homophily," *Science Advances*, vol. 9, no. 1, p. eabq3200, 2023.

[37] M. Li, Y. Gu, Y. Wang, Y. Fang, L. Bai, X. Zhuang, and P. Lio, "When hypergraph meets heterophily: New benchmark datasets and baseline," in *AAAI*, 2025.

[38] I. Amburg, J. Kleinberg, and A. R. Benson, "Planted hitting set recovery in hypergraphs," *Journal of Physics: Complexity*, vol. 2, no. 3, p. 035004, 2021.

[39] "Code, datasets, and online appendix," 2025. [Online]. Available: https://anonymous.4open.science/r/NoAH-246E

[40] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar, "Hypergcn: A new method for training graph convolutional networks on hypergraphs," *NeurIPS*, 2019.

[41] P. S. Chodrow, N. Veldt, and A. R. Benson, "Generative hypergraph clustering: From blockmodels to modularity," *Science Advances*, vol. 7, no. 28, p. eabh1303, 2021.

[42] M. Génois and A. Barrat, "Can co-location be used as a proxy for face-to-face contacts?" *EPJ Data Science*, vol. 7, no. 1, pp. 1–18, 2018.

[43] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," in *EMNLP*, 2019.

[44] I. Amburg, N. Veldt, and A. R. Benson, "Fair clustering for diverse and experienced groups," *arXiv:2006.05645*, 2020.

[45] Y. Kook, J. Ko, and K. Shin, "Evolution of real-world hypergraphs: Patterns and models without oracles," in *ICDM*, 2020.

[46] K. Nakajima, K. Shudo, and N. Masuda, "Randomizing hypergraphs preserving degree correlation and local clustering," *TNSE*, 2021.

[47] M. Choe, J. Ko, T. Kwon, K. Shin, and C. Faloutsos, "Kronecker generative models for power-law patterns in real-world hypergraphs," in *WWW*, 2025.

[48] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular value decomposition and principal component analysis," in *A practical approach to microarray data analysis*. Springer, 2003, pp. 91–109.

[49] C. Seshadhri, A. Pinar, and T. G. Kolda, "Wedge sampling for computing clustering coefficients and triangle counts on large graphs," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 7, no. 4, pp. 294–307, 2014.

[50] S. Hu, X. Wu, and T. H. Chan, "Maintaining densest subsets efficiently in evolving hypergraphs," in *CIKM*, 2017.

[51] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *KDD*, 2005.

APPENDIX

VIII. PROOF OF THEOREM 1

In this section, we provide proof of Theorem 1 which is given as:

Theorem 1. *There exist configurations on node attributes and parameters on* NOAH *such that the generated hypergraph follows a power-law degree distribution.*

Proof. Let l-th attribute of each core node be drawn from Bernoulli($\mu_{\mathcal{C}}^{(l)}$), and each fringe node from Bernoulli($\mu_{\mathcal{F}}^{(l)}$). Assume the seed probability is uniform: $p_{\text{seed}} = \frac{1}{|\mathcal{C}|}$

Core degree distribution. When $|\mathcal{C}| \to \infty$, the degree of core node v_c is proportional to the attachment probability of v_c to an arbitrary core node. If we adjust $\mu_{c,l}$ so that

$$\frac{\mu_{\mathcal{C}}^{(l)}}{1-\mu_{\mathcal{C}}^{(l)}} = \left(\frac{\mu_{\mathcal{C}}^{(l)} \times \theta_{\mathcal{C}}^{(l)}[1,1] + (1-\mu_{\mathcal{C}}^{(l)}) \times \theta_{\mathcal{C}}^{(l)}[0,1]}{\mu_{\mathcal{C}}^{(l)} \times \theta_{\mathcal{C}}^{(l)}[1,0] + (1-\mu_{\mathcal{C}}^{(l)}) \times \theta_{\mathcal{C}}^{(l)}[0,0]}\right)^{-\delta},$$

then following Theorem of [35], probability of core nodes having degree d is proportional to $d^{-\delta-\frac{1}{2}}$.

Fringe degree distribution. For core group g generated from seed core v_s, if $\mathbf{x}_s^{(l)} = 1$, the probability of l-th attribute of attached core node v_c having 1 is calculated as:

$$\begin{aligned} &p(\mathbf{x}_c^{(l)} = 1 | v_c \in g, \mathbf{x}_s^{(l)} = 1) \\ &= \frac{\mu_{\mathcal{C}}^{(l)} \times \theta_{\mathcal{C}}^{(l)}[1,1]}{(1-\mu_{\mathcal{C}}^{(l)}) \times \theta_{\mathcal{C}}^{(l)}[1,0] + \mu_{\mathcal{C}}^{(l)} \times \theta_{\mathcal{C}}^{(l)}[1,1]} \end{aligned}$$

If $\mathbf{x}_s^{(l)} = 0$, the probability of l-th attribute of attached core node v_c having 1 is calculated as:

$$\begin{aligned} &p(\mathbf{x}_c^{(l)} = 1 | v_c \in g, \mathbf{x}_s^{(l)} = 0) \\ &= \frac{\mu_{\mathcal{C}}^{(l)} \times \theta_{\mathcal{C}_l}[0,1]}{(1-\mu_{\mathcal{C}}^{(l)}) \times \theta_{\mathcal{C}_l}[0,0] + \mu_{\mathcal{C}}^{(l)} \times \theta_{\mathcal{C}_l}[0,1]} \end{aligned}$$

If we set $\theta_{\mathcal{C}_l}[0,1] = \theta_{\mathcal{C}_l}[1,0]$, and $\theta_{\mathcal{C}_l}[0,1]^2 = \theta_{\mathcal{C}_l}[0,0] \times \theta_{\mathcal{C}_l}[1,1]$, then

$$\begin{aligned} &p(\mathbf{x}_c^{(l)} = 1 | v_c \in g, \mathbf{x}_s^{(l)}) \\ &= p(\mathbf{x}_c^{(l)} = 1 | v_c \in g, \mathbf{x}_s^{(l)}) \end{aligned}$$

Since the attribute distribution of attached core nodes is now independent of the seed core attribute, let's denote it as l-th attribute following Bernoulli($\mu_a^{(l)}$). Moreover, for l-th attribute, the nodes in the core group of the hypergraph follows Bernoulli($\mu_g^{(l)}$), where

$$\mu_g^{(l)} = \frac{\mu_{\mathcal{C}}^{(l)} + \tilde{g} \times \mu_a^{(l)}}{1+\tilde{g}},$$

where $\tilde{g}$ denotes the average core group size. Then, if we adjust $\mu_{\mathcal{F}}^{(l)}$ so that

$$\frac{\mu_{\mathcal{F}}^{(l)}}{1-\mu_{\mathcal{F}}^{(l)}} = \left(\frac{\mu_g^{(l)} \times \theta_{\mathcal{F}}^{(l)}[1,1] + (1-\mu_g^{(l)}) \times \theta_{\mathcal{F}}^{(l)}[0,1]}{\mu_g^{(l)} \times \theta_{\mathcal{F}}^{(l)}[1,0] + (1-\mu_g^{(l)}) \times \theta_{\mathcal{F}}^{(l)}[0,0]}\right)^{-\delta},$$

following Theorem 7.1 of [35], probability of fringe nodes having degree d is proportional to $d^{-\delta-\frac{1}{2}}$ as $|\mathcal{F}| \to \infty$.

Conclusion. Thus, under the assumptions above, the probability of nodes having degree d is proportional to $d^{-\delta-\frac{1}{2}}$ for both core and fringe nodes, resulting in a power-law distribution for node degree. □

IX. PARAMETER INITIALIZATION OF NOAHFIT

In this section, we provide the parameter initialization method of NOAHFIT.

- **Seed Core Probability p_{seed}:** We initialize the seed core probability to a value proportional to the degree, i.e.,

$$\boldsymbol{p}_{\text{seed}}(v_c) = \frac{|\{e \in \mathcal{E} | v_c \in e\}|}{\sum_{e \in \mathcal{E}} |\mathcal{C}_e|}$$

- **Core and Fringe Affinity Matrices $\Theta_{\mathcal{C}}, \Theta_{\mathcal{F}}$:** For each core and fringe affinity matrices, we initialize to a constant value that produces the mean expected cardinality of (core, fringe) subset same as the target hypergraph, i.e.,

$$\begin{aligned} \forall l \in \{1,\dots,k\},\ \forall i \in \{0,1\},\ \forall j \in \{0,1\}, \\ \theta_{\mathcal{C}}^{(l)}[i,j] = (\overline{c_c} - 1)^{1/k}, \\ \theta_{\mathcal{F}}^{(l)}[i,j] = \overline{c_f}^{1/k}, \end{aligned}$$

  where $\overline{c_c}, \overline{c_f}$ each denotes the mean cardinality of the core and fringe subset in the target hypergraph.

X. DETAILED MODEL CONFIGURATIONS

- HYPERCL and HYPERLAP: Models utilize the degree and hyperedge size distribution of the input hypergraph.
- HYPERPA: Model utilizes the distribution of hyperedge size and number of new hyperedges for a new node of the input hypergraph.
- HYPERFF: $p \in \{0.42, 0.45, 0.48\}$ and $q \in \{0.1, 0.2, 0.3\}$
- hyper dK-series: Model utilizes the degree and hyperedge size distribution of the input hypergraph, and additional parameters $(d_v, d_e) \in \{(0,0),(0,1),(1,0)\}$
- THERA: Model utilizes the hyperedge size distribution of the input hypergraph. For parameters we tested $C \in \{8, 12, 15\}$, $p \in \{0.5, 0.7, 0.9\}$, $\alpha \in \{2, 6, 10\}$.
- HYCOSBM: $\gamma \in \{0.0, 0.1, \cdots, 0.9\}$
- HYREC: For all dataset, $L = 2, E = 100000$. For Stack domain, $\lambda_d \in \{10, 100, 1000\}, \lambda_s \in [10, 100, 1000]$. For other domains, $\lambda_d \in \{0, 0.1, 0.01\}, \lambda_s \in \{0, 1, 2\}$.
- NOAH: For all dataset, $T = 500, \eta = 0.01$. For Stack domain, $w_{deg} \in \{10, 100, 1000\}, w_{card} \in [10, 100, 1000]$. For other domains, $w_{deg} \in \{0, 0.1, 0.01\}, w_{card} \in \{0, 1, 2\}$.

XI. PERFORMANCE COMPARISON ON STRUCTURE

In this section, we evaluate NOAH along with 8 baseline generators in Section VI. We evaluate nine structural properties:

- **S1. Degree:** *Degree* of a node v, $d(v)$ is defined as the number of hyperedges containing v, i.e., $d(v) = |\{e \in \mathcal{E} : v \in e\}|$. The degree distribution tends to have a heavy tail in

the real world, which can't be reproduced in random models [45].

- **S2. Pair Degree:** *Pair degree* of two nodes u and v is defined as the number of hyperedges containing both u and v, i.e., $|\{e \in \mathcal{E} : u \in e, v \in e\}|$. The distribution of non-zero pair degree tends to have a heavier tail in the real world compared to randomized models [7].
- **S3. Size:** *Size* of a hyperedge e, $s(e)$ is defined as the number of nodes contained in e, i.e., $s(e) = |\{v \in e\}|$. The size distribution tends to have a heavy tail in the real world, which can't be reproduced in random models [45].
- **S4. Intersection Size (Int. Size):** *Intersection size* of two hyperedges e_1 and e_2 is defined as the number of nodes contained in both e_1 and e_2, i.e., $|\{v \in \mathcal{V} : v \in e_1, v \in e_2\}|$. The distribution of non-zero intersection size tends to be heavy-tailed in real-world hypergraphs [45].
- **S5. Singular Values (SV):** *Singular values* are derived from the incidence matrix of a hypergraph. Specifically, for each $i \in 1, \ldots, R$, we calculate the ratio $\frac{s_i^2}{\sum_{k=1}^{R} s_k^2}$, where s_i denotes the i-th largest singular value and R denotes the rank of the incidence matrix. These values reflect the variance captured by the associated singular vectors [48], and in many real-world hypergraphs, they are often highly skewed [45].
- **S6. Connected Component Size (CC):** We consider the proportion of nodes contained within each i-th largest connected component in the clique expansion of a hypergraph. The clique expansion of a hypergraph refers to the undirected graph formed by substituting each hyperedge $e \in \mathcal{E}$ with a clique with the nodes in e. In a real-world hypergraph, clique expansion has a giant connected component comprising a majority of nodes [6].
- **S7. Global Clustering Coefficient (GCC):** C_v, the *Local clustering coefficient* of a node v, is defined as follows:

$$C_v := 2 \times \frac{\text{the number of triangles involving } v}{\text{the number of connected triplets involving } v}, \tag{10}$$

  We estimate the *global clustering coefficient* by averaging local clustering coefficients in the clique expansion (defined in S6) of a hypergraph using [49]. This statistic tends to be larger in real-world hypergraphs than in uniform random hypergraphs [6].
- **S8. Density:** We consider the *density* which is defined as ratio of the number of hyperedges to the number of nodes, i.e., $\frac{|\mathcal{V}|}{|\mathcal{E}|}$ [50]. Hypergraphs within the same domain tend to exhibit similar levels of density significance [7].
- **S9. Overlapness:** We consider the *overlapness* which is defined as $\sum_{e \in \mathcal{E}} \frac{|e|}{|\mathcal{E}|}$ [7]. Hypergraphs within the same domain tend to exhibit similar levels of overlapness significance [7].
- **S10. Effective Diameter:** We consider the *effective diameter* which is defined as the smallest $d \in \mathbb{Z}$ such that the paths of length at most d in the clique expansion (defined in S6) connect 90% of reachable pairs of nodes [51]. This statistic tends to be small in real-world hypergraphs [45].

We used the Kolmogorov-Smirnov (D-statistic) for degree, size, pair degree, and intersection size, root mean square error (RMSE) for singular values, clustering coefficients, density, and overlapness, and relative difference for effective diameter. The results are summarized in Table IV. Although NOAH primarily targets structure–attribute interplay, it also performs competitively on purely structural patterns, ranking an average of 4.6 among 9 structure-focused methods.

TABLE IV
STRUCTURAL METRIC EVALUATION ACROSS 9 DATASETS. A.R. DENOTES AVERAGE RANK.

	Degree	Pair Degree	Size	Int. Size	SV	CC	GCC	Density	Overlapness	Diameter	A.R.
HYPERCL	0.195	0.183	0.000	0.220	0.038	0.286	0.120	0.289	0.289	0.555	4.8
HYPERPA	0.127	0.105	0.033	0.011	0.144	0.297	0.323	0.780	0.811	0.429	5.9
HYPERFF	0.366	0.056	0.396	0.209	0.104	0.301	0.164	2.247	1.153	0.031	6.4
HYPERLAP	0.244	0.030	0.000	0.024	0.054	0.260	0.196	0.227	0.227	0.021	3.2
hyper dK-series	0.001	0.185	0.148	0.222	0.067	0.252	0.118	0.039	0.003	0.533	4.5
THERA	0.456	0.079	0.013	0.006	0.057	0.129	0.307	0.350	0.389	0.370	4.7
HYCOSBM	0.484	0.176	0.114	0.177	0.085	0.298	0.050	0.110	0.843	0.704	6.1
HYREC	0.238	0.032	0.162	0.127	0.048	0.337	0.202	0.018	0.136	0.286	4.3
NOAH	0.111	0.172	0.513	0.171	0.118	0.075	0.304	0.517	0.097	0.204	5.0

(a) Citeseer (NOAH ranks **sixth** overall)

	Degree	Pair Degree	Size	Int. Size	SV	CC	GCC	Density	Overlapness	Diameter	A.R.
HYPERCL	0.189	0.115	0.000	0.412	0.034	0.286	0.228	0.276	0.276	0.464	4.7
HYPERPA	0.137	0.258	0.020	0.172	0.273	0.298	0.414	1.674	1.661	0.368	6.3
HYPERFF	0.441	0.199	0.251	0.351	0.041	0.298	0.258	4.410	2.328	0.117	6.6
HYPERLAP	0.177	0.034	0.000	0.334	0.024	0.279	0.162	0.236	0.236	0.361	2.6
hyper dK-series	0.171	0.119	0.001	0.417	0.057	0.288	0.225	0.176	0.174	0.439	4.7
THERA	0.601	0.053	0.015	0.236	0.104	0.257	0.091	1.227	1.162	0.376	4.7
HYCOSBM	0.349	0.112	0.039	0.387	0.091	0.297	0.209	0.291	0.617	0.599	5.9
HYREC	0.124	0.121	0.171	0.019	0.037	0.622	0.113	0.410	0.208	0.515	4.7
NOAH	0.096	0.110	0.272	0.409	0.046	0.191	0.310	0.447	0.140	0.398	4.6

(b) Cora

	Degree	Pair Degree	Size	Int. Size	SV	CC	GCC	Density	Overlapness	Diameter	A.R.
HYPERCL	0.031	0.272	0.000	0.034	0.029	0.000	0.409	0.000	0.000	0.284	2.7
HYPERPA	0.189	0.165	0.009	0.018	0.099	0.000	0.472	0.063	0.066	0.254	3.5
HYPERFF	0.899	0.100	0.115	0.061	0.260	0.000	0.126	0.907	0.896	0.935	5.8
HYPERLAP	0.034	0.067	0.000	0.020	0.016	0.000	0.264	0.000	0.000	0.312	1.9
hyper dK-series	0.327	0.251	0.245	0.030	0.122	0.000	0.335	0.093	0.003	0.295	4.6
THERA	0.266	0.373	0.008	0.093	0.056	0.000	0.066	0.000	0.006	0.090	3.4
HYCOSBM	1.000	1.000	0.458	0.239	0.194	0.000	0.986	0.924	6.376	0.658	7.9
HYREC	0.237	0.267	0.248	0.109	0.118	0.143	0.265	0.197	0.143	0.361	6.2
NOAH	0.495	0.349	0.530	0.035	0.120	0.000	0.731	0.000	0.304	0.270	5.4

(c) High School

	Degree	Pair Degree	Size	Int. Size	SV	CC	GCC	Density	Overlapness	Diameter	A.R.
HYPERCL	0.087	0.012	0.000	0.002	0.016	0.000	0.412	0.000	0.000	0.142	2.3
HYPERPA	0.086	0.087	0.007	0.006	0.048	0.000	0.411	0.028	0.024	0.040	3.9
HYPERFF	0.826	0.017	0.056	0.032	0.167	0.000	0.206	0.811	0.805	1.624	5.8
HYPERLAP	0.065	0.021	0.000	0.004	0.019	0.000	0.331	0.000	0.000	0.139	2.3
hyper dK-series	0.043	0.078	0.335	0.015	0.026	0.000	0.035	0.114	0.016	0.190	3.8
THERA	0.185	0.296	0.011	0.038	0.027	0.000	0.082	0.000	0.006	0.199	4.0
HYCOSBM	0.978	0.999	0.590	0.282	0.200	0.000	1.347	0.266	2.696	0.609	7.7
HYREC	0.512	0.429	0.286	0.188	0.284	0.342	0.241	0.516	0.446	0.655	7.4
NOAH	0.272	0.019	0.505	0.000	0.050	0.000	0.535	0.000	0.199	0.080	4.2

(d) Workspace

	Degree	Pair Degree	Size	Int. Size	SV	CC	GCC	Density	Overlapness	Diameter	A.R.
HYPERCL	0.097	0.051	0.000	0.102	0.027	0.002	0.027	0.063	0.063	0.088	2.2
HYPERPA	0.225	0.313	0.034	0.162	0.210	0.002	0.304	1.874	1.712	0.040	5.5
HYPERFF	0.139	0.040	0.740	0.442	0.048	0.002	0.150	6.974	0.210	1.218	5.5
HYPERLAP	0.126	0.033	0.000	0.015	0.031	0.002	0.108	0.067	0.067	0.227	2.6
hyper dK-series	0.497	0.261	0.420	0.357	0.254	0.002	0.618	0.000	0.008	0.093	5.5
THERA	0.456	0.123	0.025	0.181	0.224	0.002	0.299	0.611	0.517	0.087	5.3
HYCOSBM	0.431	0.305	0.501	0.380	0.089	0.005	0.018	0.119	0.684	0.343	6.6
HYREC	0.315	0.039	0.083	0.033	0.079	0.002	0.194	0.001	0.042	0.098	3.4
NOAH	0.233	0.156	0.280	0.242	0.119	0.018	0.228	0.063	0.458	0.603	6.1

(e) Amazon Music

	Degree	Pair Degree	Size	Int. Size	SV	CC	GCC	Density	Overlapness	Diameter	A.R.
HyperCL	0.093	0.103	0.000	0.035	0.053	0.000	0.025	0.024	0.024	0.124	2.2
HyperPA	0.549	0.368	0.023	0.187	0.253	0.000	0.294	0.326	0.320	0.308	7.0
HyperFF	0.352	0.069	0.479	0.126	0.085	0.000	0.068	1.158	0.271	1.479	6.1
HyperLAP	0.090	0.042	0.000	0.031	0.053	0.000	0.036	0.025	0.025	0.033	1.9
hyper dK-series	0.198	0.192	0.010	0.101	0.085	0.000	0.332	0.000	0.014	0.134	4.1
THera	0.177	0.149	0.020	0.169	0.148	0.000	0.200	0.000	0.042	0.311	4.7
HyCoSBM	0.347	0.193	0.265	0.083	0.063	0.004	0.100	0.287	0.448	0.083	6.1
HyRec	0.356	0.186	0.231	0.236	0.090	0.000	0.013	0.034	0.357	0.111	5.4
NoAH	0.252	0.027	0.285	0.082	0.053	0.066	0.032	0.058	0.267	0.494	4.9

(f) Yelp Restaurant

	Degree	Pair Degree	Size	Int. Size	SV	CC	GCC	Density	Overlapness	Diameter	A.R.
HyperCL	0.140	0.201	0.000	0.062	0.028	0.000	0.189	0.019	0.019	0.280	3.4
HyperPA	0.653	0.343	0.029	0.196	0.293	0.000	0.280	0.935	0.917	0.072	6.7
HyperFF	0.569	0.078	0.604	0.219	0.160	0.000	0.115	1.516	0.418	1.768	6.6
HyperLAP	0.125	0.109	0.000	0.031	0.026	0.000	0.024	0.016	0.016	0.046	1.7
hyper dK-series	0.177	0.259	0.008	0.101	0.057	0.000	0.369	0.000	0.012	0.282	4.0
THera	0.216	0.076	0.018	0.150	0.085	0.000	0.077	0.038	0.058	0.241	3.9
HyCoSBM	0.774	0.334	0.311	0.189	0.101	0.005	0.044	0.336	0.648	0.075	6.6
HyRec	0.332	0.061	0.259	0.236	0.116	0.000	0.024	0.265	0.078	0.042	4.3
NoAH	0.381	0.159	0.393	0.066	0.176	0.106	0.006	0.067	0.294	0.222	5.6

(g) Yelp Bar

	Degree	Pair Degree	Size	Int. Size	SV	CC	GCC	Density	Overlapness	Diameter	A.R.
HyperCL	0.328	0.024	0.129	0.011	0.041	0.225	0.068	0.369	0.495	0.127	3.9
HyperPA	0.350	0.103	0.127	0.126	0.102	0.194	0.383	0.272	0.389	0.335	4.8
HyperFF	0.537	0.273	0.273	0.061	0.165	0.293	0.682	1.165	1.726	0.366	7.7
HyperLAP	0.324	0.023	0.129	0.011	0.042	0.221	0.066	0.368	0.495	0.126	3.1
hyper dK-series	0.073	0.024	0.205	0.007	0.048	0.152	0.764	0.103	0.088	0.201	3.6
THera	0.433	0.021	0.134	0.001	0.232	0.215	0.124	0.000	0.084	0.481	4.1
HyCoSBM	0.815	0.019	0.277	0.035	0.119	0.293	0.243	0.098	2.662	0.615	6.4
HyRec	0.398	0.329	0.216	0.137	0.112	0.254	0.682	1.685	2.545	0.097	6.8
NoAH	0.339	0.048	0.175	0.015	0.139	0.176	0.111	0.353	0.194	0.187	4.4

(h) Devops

	Degree	Pair Degree	Size	Int. Size	SV	CC	GCC	Density	Overlapness	Diameter	A.R.
HyperCL	0.354	0.012	0.010	0.010	0.133	0.064	0.077	0.446	0.477	0.070	3.1
HyperPA	0.383	0.116	0.016	0.141	0.181	0.041	0.495	0.398	0.396	0.586	5.5
HyperFF	0.630	0.099	0.208	0.012	0.237	0.114	0.010	4.588	3.914	0.412	6.3
HyperLAP	0.345	0.013	0.010	0.008	0.133	0.063	0.096	0.435	0.465	0.084	2.9
hyper dK-series	0.009	0.009	0.167	0.014	0.133	0.050	0.378	0.087	0.005	0.095	3.1
THera	0.590	0.140	0.016	0.105	0.348	0.413	0.071	0.000	0.003	0.783	5.9
HyCoSBM	0.767	0.017	0.293	0.065	0.165	0.113	0.005	0.218	1.434	0.505	5.8
HyRec	0.463	0.304	0.213	0.059	0.263	0.280	0.314	4.086	4.262	0.272	7.4
NoAH	0.437	0.034	0.126	0.014	0.135	0.112	0.308	0.369	0.403	0.254	4.6

(i) Patents

	Degree	Pair Degree	Size	Int. Size	SV	CC	GCC	Density	Overlapness	Diameter	A.R.
HyperCL	2.9	4.6	2.1	3.9	1.8	6.2	4.3	4.4	3.8	4.6	2.8
HyperPA	4.7	7.1	3.8	5.0	6.9	6.1	7.8	6.6	6.3	4.0	6.3
HyperFF	7.1	4.2	7.6	6.3	6.4	7.7	4.6	8.8	7.4	6.8	7.3
HyperLAP	2.6	2.2	2.1	2.1	2.0	5.8	3.4	4.1	3.4	2.8	1.5
hyper dK-series	3.3	5.6	5.2	5.4	4.9	5.3	6.3	2.7	1.9	5.3	4.1
THera	6.2	4.8	3.7	4.9	6.3	6.0	4.2	4.0	4.1	5.3	4.4
HyCoSBM	8.2	6.4	7.6	6.8	6.0	7.6	4.4	5.3	8.1	6.9	7.6
HyRec	5.4	5.4	6.6	6.2	5.4	8.0	4.3	5.7	5.7	4.7	5.8
NoAH	4.6	4.7	7.4	4.3	5.9	5.8	5.6	5.0	4.4	4.7	4.6

(j) Average Rank over Nine Datasets (NoAH ranks **fifth** overall)