

Attributed Hypergraph Generation with Realistic Interplay Between Structure and Attributes

Jaewan Chun*, Seokbum Yoon*, Minyoung Choe, Geon Lee, Kijung Shin

Kim Jaechul Graduate School of AI, KAIST, Seoul, Republic of Korea

{jjwpalace, jing9044, minyoung.choe, geonlee0325, kijungs}@kaist.ac.kr

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].

* Equal Contributions

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 (Node Attribute based Hypergraph 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.

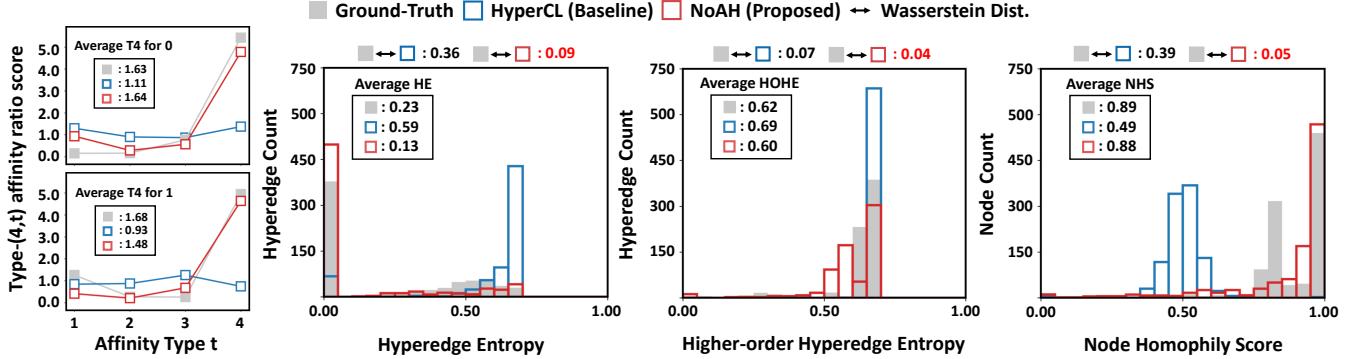


Fig. 1. Structure-attribute interplay with respect to the first node attribute in the Amazon Music dataset. Our proposed generative model, NoAH, fitted by its parameter learning algorithm NoAHFit, effectively captures the interplay between structure and node attributes, outperforming the baseline model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

- **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 <https://github.com/jaewan01/NoAH>.

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 are designed to generate structures resembling real-world hypergraphs [15], [16], often to uncover potential mechanisms behind such structures. To this end, for example, several models exploit node hierarchies [8], [17]–[19], which are commonly found in the real world [20]–[23]. In our model, NoAH, we capture such hierarchies through a simple yet effective core–fringe structure.

Recently, various deep learning based (hyper)graph generative models have been introduced, such as variational autoencoders [24], generative adversarial networks [25], [26], and diffusion or hierarchical models [27]. However, these models mostly require a large collection of graph instances for training, while the aforementioned models operate on a single instance. Moreover, deep learning models often struggle to provide insights into the generative mechanisms of real-world (hyper)graphs due to their intrinsic black-box nature.

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 where similar nodes tend to connect [13]. Accordingly, several models incorporate node attributes, including the exponential random graph model [28], the stochastic block model [29], and the latent space model [30]. Among them, the Multiplicative Attribute Graph (MAG) model [31] is distinctive in that node attributes directly govern edge formation 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

TABLE I
FREQUENTLY-USED SYMBOLS.

Notation	Definition
$\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$	hypergraph with nodes \mathcal{V} , hyperedges \mathcal{E}
$\mathbf{X} \in \{0, 1\}^{ \mathcal{V} \times k}$	node attribute matrix (binary)
\mathcal{C}	set of core nodes
\mathcal{F}	set of fringe nodes
$\Theta_C = \{\theta_{C_1}, \dots, \theta_{C_k}\}$	set of core group affinity matrices
$\Theta_F = \{\theta_{F_1}, \dots, \theta_{F_k}\}$	set of fringe attachment affinity matrices

the hypergraph structure given the latent community structure, meaning attributes do not directly drive hyperedge formation. In contrast, in our model, NoAH, hyperedges are formed explicitly based on attribute relationships among nodes.

III. PRELIMINARIES

In this section, we introduce (1) key notations, (2) six measures for evaluating the interplay between structure and attributes, (3) MAG [31] as a preliminary model, and (4) UMHS [32] as a core-node identification method for hypergraphs.

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 $\mathcal{V} = \{v_1, \dots, v_{|\mathcal{V}|}\}$, a set of hyperedges $\mathcal{E} = \{e_1, \dots, e_{|\mathcal{E}|}\}$, 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 binary node attributes, i.e., $\mathbf{X} \in \{0, 1\}^{|\mathcal{V}| \times k}$, which 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, categorized into: (1) **hyperedge-level measures** (hyperedge entropy and

affinity ratio score), which capture how node attributes are distributed within hyperedges, and a (2) **node-level measure** (node homophily score), which measures the tendency of nodes to form hyperedges with others of similar attributes.

Type- s Affinity Ratio Scores. Veldt et al. [33] 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 $t \in \{1, 2, \dots, s\}$, they defined the type- (s, t) *affinity score* for label Y as follows:

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

where \mathcal{V}_Y is the set of nodes with label Y , $d(v)$ the degree of node v , and $d_{s,t}(v)$ the number of size s hyperedges containing v and $t - 1$ additional nodes with v 's label (Y). It quantifies how frequently nodes with label Y appear in size- s hyperedges with exactly t such nodes. 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.,

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

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 values 0 and 1 of each attribute. In this work, we focus on $s \in \{2, 3, 4\}$, as small hyperedges are common in many real-world hypergraphs, and to allow reliable statistics. For simplicity, we refer to the type- $(s, 1)$ to type- (s, s) affinity ratio scores as the type- s affinity ratio scores.

(Higher-order) Hyperedge Entropy. Lee et al. [14] observed that real-world hyperedges exhibit label homogeneity, containing nodes with similar labels more often than their randomized counterparts. This effect persists even after multiple steps of propagation of labels to incident nodes and hyperedges. To measure the label homogeneity, they utilized *hyperedge entropy*, defined as the entropy of the labels of nodes incident to each hyperedge. They also introduced *higher-order hyperedge entropy*, measuring the entropy after label propagation.

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. A distribution skewed toward 0 reflects that hyperedges mostly comprise nodes with identical attribute values.

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:

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

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 containing v , and $l \in \{1, 2, \dots, k\}$. This measure indicates the ratio of incident

nodes that share the same l -th attribute value with node v , with a high value reflecting a greater tendency for v to form hyperedges with nodes having the same attribute value. We consider the distribution of node homophily score for each attribute to capture node-level homogeneity in a hypergraph.

Overall, the above measures offer complementary perspectives on structure-attribute interplay: (1) the **type- s affinity ratio 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 [31] proposed the Multiplicative Attribute Graph (MAG) model to capture how node attributes affect edge formation in pairwise graphs. Given a set of nodes \mathcal{V} and an associated node attribute matrix $\mathbf{X} \in \{0, 1\}^{|\mathcal{V}| \times k}$, MAG estimates the probability of edges based on node attributes.¹ Specifically, MAG defines a set of attribute affinity matrices $\Theta = \{\theta_1, \dots, \theta_k\}$, where each $\theta_l \in \mathbb{R}^{2 \times 2}$ captures the affinity between values of the l -th attribute, with $\theta_l[x_1, x_2] \in [0, 1]$ denoting the affinity between binary attribute values $x_1, x_2 \in \{0, 1\}$. For the probability $P(v_i, v_j)$ of an edge between nodes v_i and v_j , MAG multiplies the affinities across all k attributes as follows:

$$P(v_i, v_j) = \prod_{l=1}^k \theta_l[\mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)}]. \quad (4)$$

Inspired by MAG, 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.

D. UMHS: Union of Minimal Hitting Sets

Amburg et al. [32] defined a *hitting set* of a hypergraph as a set of nodes that intersects every hyperedge, i.e., a set S satisfying $\forall e \in \mathcal{E}, \exists v \in S$ s.t. $v \in e$. Building on this concept, they proposed the union of minimal hitting sets (UMHS) algorithm to identify core nodes in a hypergraph. The algorithm consists of three steps: (1) initialize $S = \{\}$, generate a random permutation of the hyperedges, and traverse them, adding all nodes in a hyperedge e to S whenever $S \cap e = \emptyset$, (2) generate a random permutation of S and traverse them, removing any node v if $S \setminus \{v\}$ remains a hitting set, and (3) repeat the above two steps with different permutations and take the union of the resulting hitting sets.

IV. PROPOSED GENERATION METHOD: NOAH

In this section, we introduce NOAH, a novel hypergraph generative model based on node attributes.

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

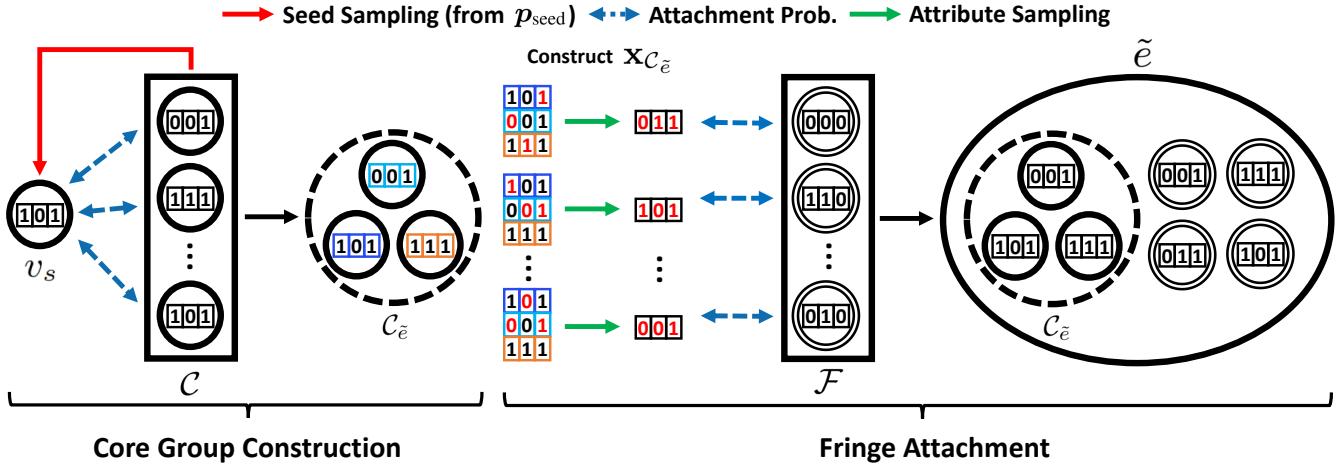


Fig. 2. Hyperedge generation process of NOAH consists of two steps: (1) Core group construction: sample a seed core node v_s according to \mathbf{p}_{seed} and attach additional core nodes to v_s to form a core group $\mathcal{C}_{\tilde{e}}$, and (2) Fringe attachment: attach fringe nodes to core group $\mathcal{C}_{\tilde{e}}$ based on mixed attribute vector $\mathbf{x}_{\mathcal{C}_{\tilde{e}}}$ 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.

A. Ideas behind NOAH.

In real-world hypergraphs, node attributes play a crucial role in hyperedge formation. 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 [34]. NOAH captures such attribute-structure interplay to generate more realistic hypergraphs.

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

Moreover, many real-world hypergraphs exhibit hierarchical structures [22], [23], 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], [18], [19]. 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. In line with Idea 1, nodes are sampled based on attribute relationships, using the node attributes \mathbf{X} . 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. $\mathcal{C} \cap \mathcal{F} = \emptyset$ and $\mathcal{C} \cup \mathcal{F} = \mathcal{V}$, and the two groups play distinct roles in hyperedge formation. Combining three ideas, NOAH generates each hyperedge in a two-step process. *First*, a subset of core nodes from \mathcal{C} form a core group based on attribute affinities. *Second*, fringe nodes from \mathcal{F} attach to the core group according to 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 $\mathbf{p}_{\text{seed}} \in [0, 1]^{|\mathcal{C}|}$, where $\|\mathbf{p}_{\text{seed}}\|_1 = 1$, and $\mathbf{p}_{\text{seed}}(v)$ is the probability of selecting node $v \in \mathcal{C}$ as a seed core. Then, NOAH calculates the inclusion probability of remaining core node $v_c \in \mathcal{C} \setminus \{v_s\}$ to the core group based on its attribute affinity with the seed node v_s , computed as:

$$P_C(v_c|v_s, \Theta_C) = \prod_{l=1}^k \theta_C^{(l)}[\mathbf{x}_s^{(l)}, \mathbf{x}_c^{(l)}], \quad (5)$$

where $\Theta_C = \{\theta_C^{(1)}, \dots, \theta_C^{(k)}\}$ is the set of attribute affinity matrices modeling interactions *among core nodes*, and each $\theta_C^{(l)} \in \mathbb{R}^{2 \times 2}$ 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:

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

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:

$$P_{\mathcal{F}}(v_f|\mathcal{C}_{\tilde{e}}, \Theta_{\mathcal{F}}) = \prod_{l=1}^k \theta_{\mathcal{F}}^{(l)}[\mathbf{x}_{\mathcal{C}_{\tilde{e}}}^{(l)}, \mathbf{x}_f^{(l)}], \quad (7)$$

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 \mathbf{p}_{seed}
(6) set of core group affinity matrices $\Theta_{\mathcal{C}}$
(7) set of fringe attachment affinity matrices $\Theta_{\mathcal{F}}$

Output: generated hypergraph $\tilde{\mathcal{H}} = (\mathcal{V}, \tilde{\mathcal{E}})$

```

1  $\tilde{\mathcal{H}} = (\mathcal{V}, \tilde{\mathcal{E}} = \emptyset)$ 
2 for each  $i = 1, \dots, m$  do
3   // 1. Core Group Construction
4   Sample  $v_s \sim \mathbf{p}_{\text{seed}}$ ,  $v_s \in \mathcal{C}$ 
5    $\mathcal{C}_{\tilde{e}} \leftarrow \{v_s\}$ 
6   for each  $v_c \in \mathcal{C} \setminus \{v_s\}$  do           ▶ Eq. (5)
7      $p \leftarrow P_{\mathcal{C}}(v_c|v_s, \Theta_{\mathcal{C}})$ 
8     with probability  $p$  do
9        $\mathcal{C}_{\tilde{e}} \leftarrow \mathcal{C}_{\tilde{e}} \cup \{v_c\}$ 
10  // 2. Fringe Node Attachment
11   $\mathcal{F}_{\tilde{e}} \leftarrow \emptyset$ 
12  for each  $v_f \in \mathcal{F}$  do           ▶ Eq. (6)
13    Construct  $\mathbf{x}_{\mathcal{C}_{\tilde{e}}}$  of  $\mathcal{C}_{\tilde{e}}$ 
14     $q \leftarrow P_{\mathcal{F}}(v_f|\mathcal{C}_{\tilde{e}}, \Theta_{\mathcal{F}})$            ▶ Eq. (7)
15    with probability  $q$  do
16       $\mathcal{F}_{\tilde{e}} \leftarrow \mathcal{F}_{\tilde{e}} \cup \{v_f\}$ 
17   $\tilde{e} \leftarrow \mathcal{C}_{\tilde{e}} \cup \mathcal{F}_{\tilde{e}}$ 
18   $\tilde{\mathcal{E}} \leftarrow \tilde{\mathcal{E}} \cup \{\tilde{e}\}$ 
19 return  $\tilde{\mathcal{H}} = (\mathcal{V}, \tilde{\mathcal{E}})$ 

```

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

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], [19].

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 [35].

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} . We utilize UMHS (see Section III-D) to identify the core node set \mathcal{C} . Then, 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., \mathbf{p}_{seed} , $\Theta_{\mathcal{C}}$, and $\Theta_{\mathcal{F}}$), we now derive the likelihood of each hyperedge $e \in \mathcal{E}$, denoted as $P(e|\mathcal{C}, \mathcal{F}, \mathbf{X}, \mathbf{p}_{\text{seed}}, \Theta_{\mathcal{C}}, \Theta_{\mathcal{F}})$, 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:

$$P_{\text{core}}(e) = \sum_{v_s \in \mathcal{C}_e} \mathbf{p}_{\text{seed}}(v_s) \cdot P_{\mathcal{C}}(\mathcal{C}_e \setminus \{v_s\}|v_s), \quad (8)$$

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:

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

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

²For 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., $P_{\text{core}}(e)/\sum_{v_c \in \mathcal{C}_e} \mathbf{p}_{\text{seed}}(v_c)$.

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

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

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., \mathbf{p}_{seed} , $\Theta_{\mathcal{C}}$, and $\Theta_{\mathcal{F}}$) as follows:

- **Negative Log-likelihood Loss ($\mathcal{L}_{\text{edge}}$):** 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:

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

- **Degree and Cardinality Losses (\mathcal{L}_{deg} and $\mathcal{L}_{\text{card}}$):** 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:

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

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

where $\mathbf{d}_c, \mathbf{d}_f$ denote the degrees of core and fringe nodes; and $\mathbf{c}_c, \mathbf{c}_f$ 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 [32] 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 hyperedges results in a total computation cost of $O(Tk|\mathcal{V}|\sum_{e \in \mathcal{E}} |\mathcal{C}_e|)$. 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, the total space complexity is $O(k + m + |\mathcal{V}|)$.

³Initialization method of parameters is explained in Appendix IX [35].

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 $w_{\text{deg}}, w_{\text{card}}$

Output: (1) set of core fringe nodes \mathcal{C}
(2) set of core fringe nodes \mathcal{F}
(3) seed core probabilities \mathbf{p}_{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
1 Split  $\mathcal{V}$  into  $\mathcal{C}, \mathcal{F}$ 
2 Initialize  $\mathbf{p}_{\text{seed}}, \Theta_{\mathcal{C}}$  and  $\Theta_{\mathcal{F}}$ 3
3 for each  $t = 1, \dots, T$  do
    // 2. Log-likelihood Loss
    4  $\mathcal{L}_{\text{edge}} \leftarrow 0$ 
    5 for each  $e \in \mathcal{E}$  do
        6  $\mathcal{C}_e \leftarrow e \cap \mathcal{C}, \mathcal{F}_e \leftarrow e \cap \mathcal{F}$ 
        7  $P(e) \leftarrow P_{\text{core}}(e) \cdot P_{\text{fringe}}(e)$  ▶ Eq. (8) & Eq. (9)
        8  $\mathcal{L}_{\text{edge}} \leftarrow \mathcal{L}_{\text{edge}} - \log P(e)$ 
    // 3. Degree, Cardinality Losses
    9 Compute  $\tilde{\mathbf{d}}_c, \tilde{\mathbf{d}}_f$  and  $\tilde{\mathbf{c}}_c, \tilde{\mathbf{c}}_f$  using  $\mathbf{p}_{\text{seed}}, \Theta_{\mathcal{C}}$  and  $\Theta_{\mathcal{F}}$ 
10  $\mathcal{L}_{\text{deg}} \leftarrow \text{MSE}(\mathbf{d}_c, \tilde{\mathbf{d}}_c) + \text{MSE}(\mathbf{d}_f, \tilde{\mathbf{d}}_f)$ 
11  $\mathcal{L}_{\text{card}} \leftarrow \text{MSE}(\mathbf{c}_c, \tilde{\mathbf{c}}_c) + \text{MSE}(\mathbf{c}_f, \tilde{\mathbf{c}}_f)$ 
12  $\mathcal{L} \leftarrow \mathcal{L}_{\text{edge}} + w_{\text{deg}} \cdot \mathcal{L}_{\text{deg}} + w_{\text{card}} \cdot \mathcal{L}_{\text{card}}$ 
    // 4. Parameter Update
13  $\mathbf{p}_{\text{seed}} \leftarrow \mathbf{p}_{\text{seed}} + \eta \nabla_{\mathbf{p}_{\text{seed}}} \mathcal{L}$ 
14  $\Theta_{\mathcal{C}} \leftarrow \Theta_{\mathcal{C}} + \eta \nabla_{\Theta_{\mathcal{C}}} \mathcal{L}$ 
15  $\Theta_{\mathcal{F}} \leftarrow \Theta_{\mathcal{F}} + \eta \nabla_{\Theta_{\mathcal{F}}} \mathcal{L}$ 
16 return  $\mathcal{C}, \mathcal{F}, \mathbf{p}_{\text{seed}}, \Theta_{\mathcal{C}}, \Theta_{\mathcal{F}}$ 

```

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 [36]):** 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, indicating whether a paper includes each keyword or not.
- **Contact domain (High School [37], Workspace [38]):** Each node is an individual, and each hyperedge is a group of individuals in contact 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 of each worker. Since all attributes are categorical, we apply one-hot encoding to convert them into binary attributes.
- **Review Domain (Amazon Music [39], Yelp Restaurant, Yelp Bar [40]):** Each node is a reviewer, and each hyperedge 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 Domain⁴(Devops, Patents):** Each node represents a user on Stack Exchange, and each hyperedge cor-

⁴<https://archive.org/download/stackexchange>

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

responds 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 [41], HYPERLAP [7], hyper dK-series [42], THERA [8], HYCOSBM [11], HYREC [43]. 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 [35].

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 between the ground-truth and generated hypergraphs. We denote type- s affinity ratio 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.

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

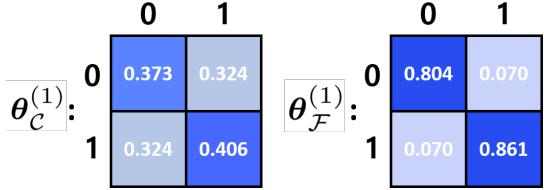


Fig. 3. $\theta_C^{(1)}$ and $\theta_F^{(1)}$ estimated by NOAHFIT on the Amazon Music dataset. NOAHFIT captures homophily by assigning higher affinities to same attribute value pairs ($0 \leftrightarrow 0$ and $1 \leftrightarrow 1$) than to different attribute value pairs ($0 \leftrightarrow 1$).

of NOAH in capturing the structure-attribute interplay of real-world hypergraphs.

However, there are cases where NOAH underperforms. First, its relatively low performance 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. Second, HYPERLAP and THERA tend to preserve higher-order hyperedge entropy (HOHE) more effectively than NOAH. This is because, due to the label-propagation steps, HOHE is strongly influenced by hyperedge overlaps, which HYPERLAP and THERA explicitly target.

C. Case Study

To gain deeper insight into the effectiveness of NOAH, we conduct a case study on the Amazon Music dataset [39]. 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 therefore without collective consideration of multiple core nodes). As reported in Table III, NOAH consistently outperforms NOAH-CF across all datasets, showing the importance of the core-fringe hierarchy in realistic hyperedge formation.

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.
HYPERCCL	6,816	10,702	10,672	19.81	122.02	19.94	6.7
HYPERPAA	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
HYCoSBBM	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.
HYPERCCL	2,099	4,206	4,481	8.02	63.64	7.99	7.0
HYPERPAA	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
HYCoSBBM	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.
HYPERCCL	20.4	51.4	106.4	1.180	1.369	1.730	7.2
HYPERPAA	20.6	52.6	107.8	1.175	1.421	1.654	8.0
HYPERFF	20.2	61.6	106.5	0.017	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
HYCoSBBM	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.
HYPERCCL	7.5	17.3	13.0	0.599	0.181	0.890	7.0
HYPERPAA	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
HYCoSBBM	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.
HYPERCCL	27.3	53.0	63.6	1.016	0.382	1.053	6.8
HYPERPAA	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
HYCoSBBM	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.
HYPERCCL	17.8	58.9	90.4	0.979	0.249	1.148	6.5
HYPERPAA	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
HYCoSBBM	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.
HYPERCCL	57.1	112.1	133.2	1.392	1.064	1.343	7.0
HYPERPAA	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	1.527	1.102	1.230	6.0
THERA	55.5	106.0	133.3	1.124	1.061	1.221	4.3
HYCoSBBM	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.
HYPERCCL	1,257	3,450	5,793	4.89	30.86	8.45	3.3
HYPERPAA	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
HYCoSBBM	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.
HYPERCCL	6,586	13,091	16,515	64.30	105.59	103.33	2.5
HYPERPAA	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
HYCoSBBM	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	

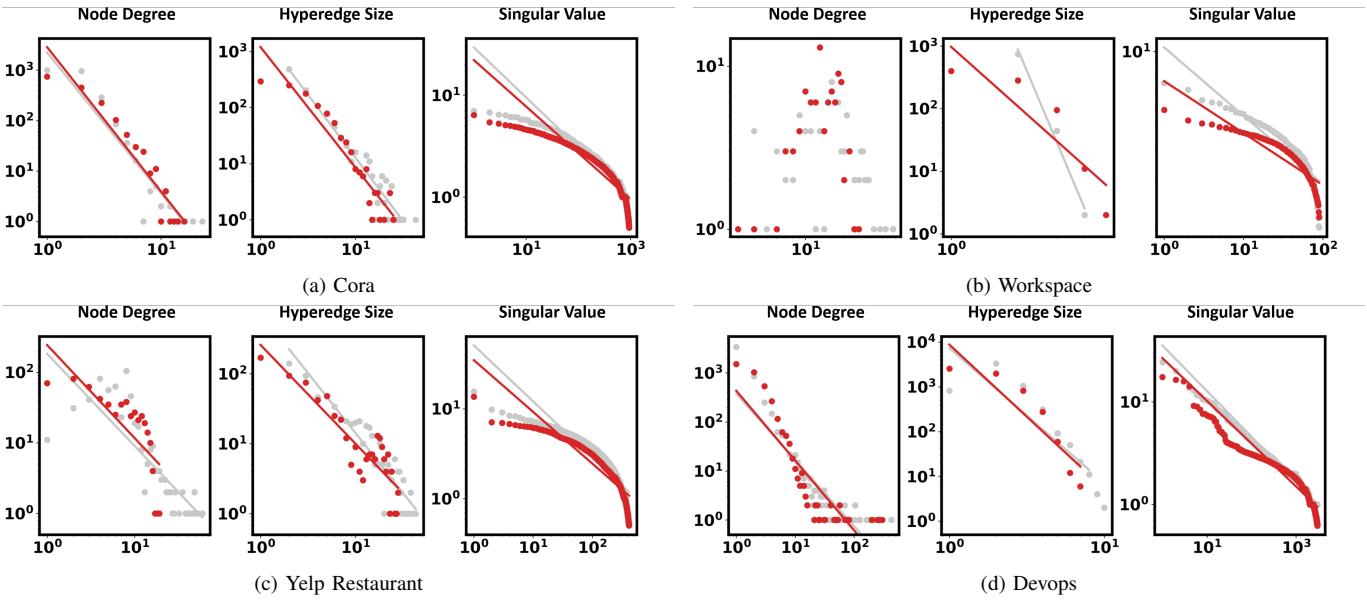


Fig. 4. Structural patterns of real-world hypergraphs (gray) and those generated by NOAH tuned by NOAHFIT (red). These results demonstrate that, despite its focus on structure–attribute interplay, NOAH also successfully reproduces purely structural patterns.

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 nearly 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 [19]. 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 [35], 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, when compared with 8 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

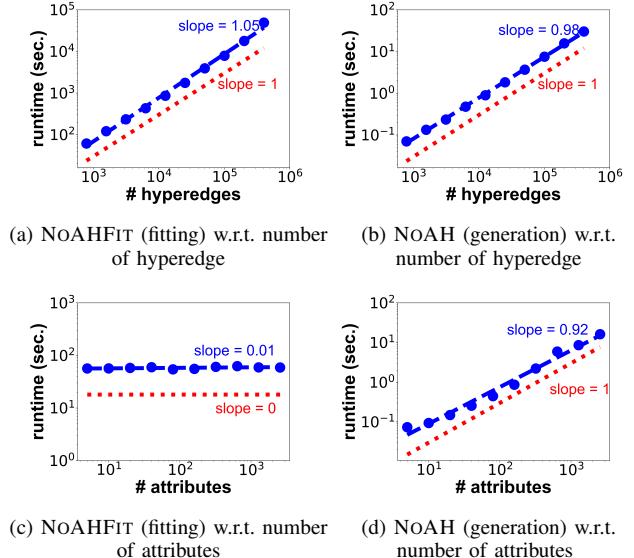


Fig. 5. NOAHFIT scales nearly linearly with the number of hyperedges and remains almost constant with respect to the number of attributes. NOAH scales nearly linearly with both the number of hyperedges and attributes.

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.

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.

Future Directions. We plan to extend our model and fitting algorithm to handle more complex attributes, including continuous ones. We also aim to further enhance their scalability to support even larger, web-scale hypergraphs.

ACKNOWLEDGEMENTS

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00871 / RS-2022-II220871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration, 50%) (No. RS-2024-00457882, AI Research Hub Project, 40%) (RS-2019-II190075, Artificial Intelligence Graduate School Program (KAIST), 10%).

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] J. Jia and A. R. Benson, “Random spatial network models for core-periphery structure,” in *WSDM*, 2019.
- [18] M. Papachristou and J. Kleinberg, “Core-periphery models for hypergraphs,” in *KDD*, 2022.
- [19] 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.
- [20] S. B. Seidman, “Network structure and minimum degree,” *Social Networks*, vol. 5, no. 3, pp. 269–287, 1983.
- [21] S. P. Borgatti and M. G. Everett, “Models of core/periphery structures,” *Social Networks*, vol. 21, no. 4, pp. 375–395, 2000.
- [22] 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.
- [23] 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.
- [24] M. Simonovsky and N. Komodakis, “Graphvae: Towards generation of small graphs using variational autoencoders,” in *ICANN*, 2018.
- [25] A. Bojchevski, O. Shchur, D. Zügner, and S. Günnemann, “Netgan: Generating graphs via random walks,” in *ICML*, 2018.
- [26] N. De Cao and T. Kipf, “Molgan: An implicit generative model for small molecular graphs,” *arXiv preprint arXiv:1805.11973*, 2018.
- [27] 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.
- [28] 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.
- [29] E. M. Airola, D. Blei, S. Fienberg, and E. Xing, “Mixed membership stochastic blockmodels,” *NeurIPS*, 2008.
- [30] 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.
- [31] M. Kim and J. Leskovec, “Multiplicative attribute graph model of real-world networks,” *Internet Mathematics*, vol. 8, no. 1–2, pp. 113–160, 2012.
- [32] 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.
- [33] 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.
- [34] 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.
- [35] J. Chun, S. Yoon, M. Choe, G. Lee, and K. Shin, “Attributed hypergraph generation with realistic interplay between structure and attributes: Code, datasets, and online appendix,” 2025. [Online]. Available: <https://github.com/jaewan01/NoAH>
- [36] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar, “Hypergen: A new method for training graph convolutional networks on hypergraphs,” *NeurIPS*, 2019.
- [37] P. S. Chodrow, N. Veldt, and A. R. Benson, “Generative hypergraph clustering: From blockmodels to modularity,” *Science Advances*, vol. 7, no. 28, p. eabhi1303, 2021.
- [38] 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.
- [39] J. Ni, J. Li, and J. McAuley, “Justifying recommendations using distantly-labeled reviews and fine-grained aspects,” in *EMNLP*, 2019.
- [40] I. Amburg, N. Veldt, and A. R. Benson, “Fair clustering for diverse and experienced groups,” *arXiv:2006.05645*, 2020.
- [41] Y. Kook, J. Ko, and K. Shin, “Evolution of real-world hypergraphs: Patterns and models without oracles,” in *ICDM*, 2020.
- [42] K. Nakajima, K. Shudo, and N. Masuda, “Randomizing hypergraphs preserving degree correlation and local clustering,” *IEEE TNSE*, vol. 9, no. 3, pp. 1139–1153, 2021.
- [43] 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.