Knowledge and Information Systems (2026) 68:245
https://doi.org/10.1007/s10115-026-02850-x

RESEARCH

From binary to general attributes: attributed hypergraph generation with realistic interplay between structure and attributes

Jaewan Chun[1] · Seokbum Yoon[1] · Minyoung Choe[1] · Geon Lee[1] · Kijung Shin[1]

Received: 3 January 2026 / Revised: 14 July 2026 / Accepted: 20 July 2026

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 fits NoAH to a given real-world hypergraph so that generated hypergraphs reproduce structural and attribute-related patterns. Through experiments on nine datasets across four different domains, we show that NoAH with NoAHFit achieves the best overall average rank among the nine evaluated hypergraph generative models when evaluated across six structure–attribute interplay metrics. Moreover, we discuss variants of NoAH for different types of node attributes, including binary, categorical, and continuous attributes. For cases without pre-existing node attributes, we extend NoAH and NoAHFit to jointly learn latent node attributes together with the parameters of NoAH and use the learned attributes for generation.

✉ Kijung Shin
kijungs@kaist.ac.kr

Jaewan Chun
jjwpalace@kaist.ac.kr

Seokbum Yoon
jing9044@kaist.ac.kr

Minyoung Choe
minyoung.choe@kaist.ac.kr

Geon Lee
geonlee0325@kaist.ac.kr

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

Published online: 21 August 2026

Keywords Hypergraph · Generator · Node attribute · Attributed hypergraph

1 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 [9]. 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 [10–12], and hyperedge prediction [13].

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 [14, 15], 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 Fig. 1, NoAH, fitted by NoAHFit, achieves the best overall average rank among the compared hypergraph generative models 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.

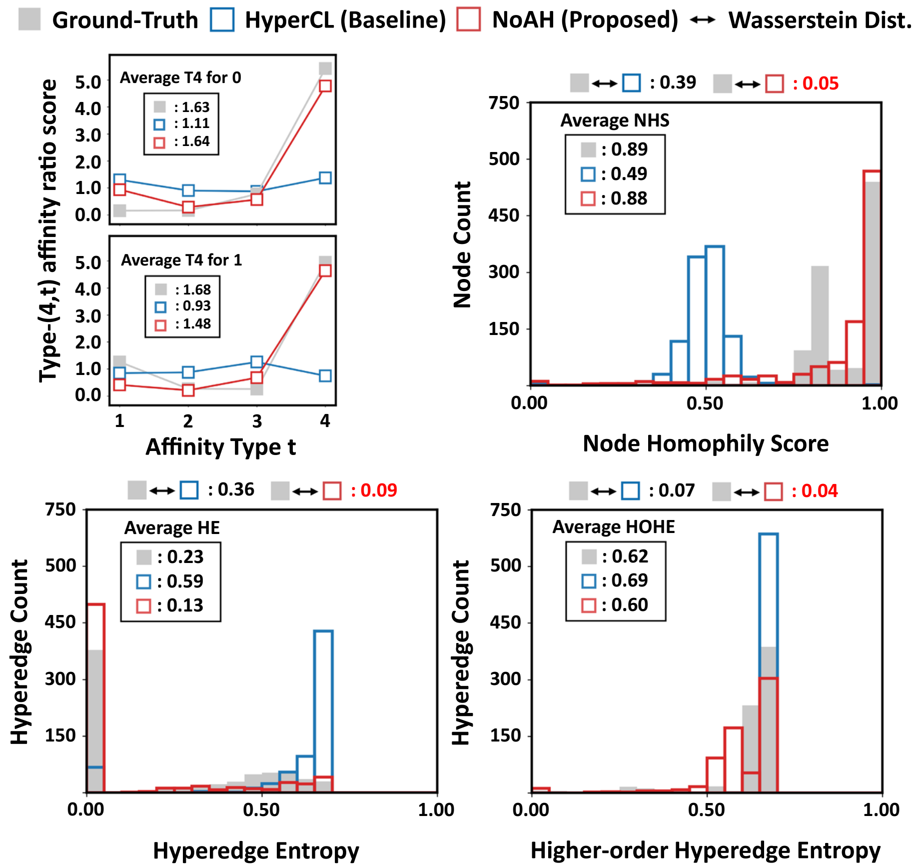


Fig. 1 Structure–attribute interplay with respect to the first node attribute in the Amazon Music dataset. Our proposed generative model, NoAH, fitted by its parameter learning algorithm NoAHFit, effectively captures the interplay between structure and node attributes, outperforming the baseline model (HyperCL). Refer to Sect. 3.2 for the definitions of the measures and to Sect. 7.3 for the interpretation of the results

- **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 the competitive overall performance of NoAH, which achieves the best average rank across the evaluated datasets and metrics.

For **reproducibility**, we make the source code and data publicly available at https://github.com/jaewan01/NoAH.

This work extends our previous study [16], which was presented at the IEEE International Conference on Data Mining (ICDM) 2025. In this extended version, first, we generalize NoAH beyond binary attributes to support categorical and continuous attributes, extending the attribute affinity formulation and likelihood accordingly (Sect. 6.1). Second, we broaden the applicability of NoAH and NoAHFit to settings without pre-existing node attributes (Sect. 6.2). To this end, we propose NoAHFit- X, which models node attributes as latent variables and jointly optimizes attribute assignment probabilities together with the parameters of NoAH; and we also extend NoAH in two alternative ways with distinct strengths, namely

NoAH- X and NoAH- X$^+$ (Sect. 6.2.3). Third, we present empirical results for the extension to non-binary attributes (Sect. 7.6) and for the settings without pre-existing attributes (Sect. 7.7). Lastly, we provide a detailed analysis (proof of Theorem 1) and extra experiments (Appendix D) on structural properties.

2 Related work

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

2.1 (Hyper)graph generative models

(Hyper)graph generative models are designed to generate structures resembling real-world hypergraphs [9, 17], often to uncover potential mechanisms behind such structures. To this end, for example, several models exploit node hierarchies [8, 18–20], which are commonly found in the real world [21–24]. 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 [25], generative adversarial networks [26, 27], and diffusion or hierarchical models [28]. 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.

2.2 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 [14]. Accordingly, several models incorporate node attributes, including the exponential random graph model [29], the stochastic block model [30], and the latent space model [31]. Among them, the Multiplicative Attribute Graph (MAG) model[32] 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 [12]. In this model, node attributes are conditionally independent of 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.

3 Preliminaries

In this section, we introduce (1) key notations, (2) six measures for evaluating the interplay between structure and attributes, (3) MAG [32] as a preliminary model, and (4) UMHS [33] as a core node identification method for hypergraphs.

Table 1 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
$\Theta_{\mathcal{F}} = \{\boldsymbol{\theta}_{\mathcal{F}_1}, \ldots, \boldsymbol{\theta}_{\mathcal{F}_k}\}$	Set of fringe attachment affinity matrices

$\Theta_{\mathcal{C}} = \{\boldsymbol{\theta}_{\mathcal{C}_1}, \ldots, \boldsymbol{\theta}_{\mathcal{C}_k}\}$ & set of core group affinity matrices

3.1 Notations

First, we discuss the notations used in this paper. Refer to Table 1 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 $\mathbf{x}^{(l)} \in \mathbb{R}^{|\mathcal{V}|}$ to denote the l-th attribute vector (i.e., the l-th column of $\mathbf{X}$), and we use $\mathbf{x}_i^{(l)} \in \mathbb{R}$ to denote the l-th attribute value of node v_i. In this work, unless otherwise specified, 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. However, our proposed model is not restricted to binary attributes, and extensions to non-binary attributes are discussed in Sect. 6.1. For categorical attributes, the l-th attribute vector $\mathbf{x}^{(l)}$ takes values in $\{0, \ldots, c_l - 1\}^{|\mathcal{V}|}$, where c_l denotes the number of categories of the l-th attribute. For continuous attributes, the l-th attribute vector $\mathbf{x}^{(l)}$ takes values in $\mathbb{R}^{|\mathcal{V}|}$, or in $[0, 1]^{|\mathcal{V}|}$ after normalization.

3.2 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. [34] 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 $\mathcal{V}_Y$ 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.,

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

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.

We consider all possible values of each attribute when computing the type-(s, t) affinity ratio score. Specifically, for categorical attributes, we consider attribute values $\{0, 1, \ldots, c-1\}$, where c denotes the number of categories, and for binary attributes, we consider the two values $\{0, 1\}$. 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. [15] 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_i at l-th attribute is defined as:

$$n_i^{(l)} := \frac{\sum_{e\in\mathcal{E}_{v_i}} |\{v_j \in e, v_i \neq v_j \mid \mathbf{x}_i^{(l)} = \mathbf{x}_j^{(l)}\}|}{\sum_{e\in\mathcal{E}_{v_i}} (|e|-1)}, \tag{3}$$

where $n_i^{(l)}$ is a node homophily score of node v_i for the l-th attribute, $\mathcal{E}_{v_i} \subseteq \mathcal{E}$ is a set of hyperedges containing v_i, 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, 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.

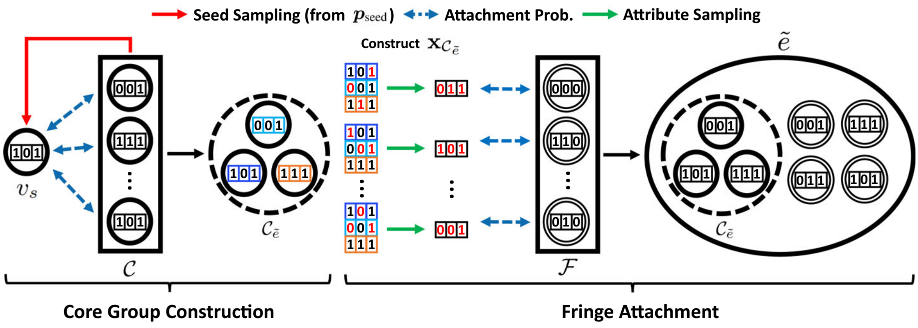


Fig. 2 Hyperedge generation process of NoAH consists of two steps: (1) Core group construction: sample a seed core node v_s according to $\boldsymbol{p}_{\text{seed}}$ and attach additional core nodes to v_s to form a core group $\mathcal{C}_{\tilde{e}}$, and (2) Fringe attachment: attach fringe nodes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

3.3 MAG: multiplicative attribute graph model

Kim and Leskovec [32] 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 $\Theta = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k\}$, where each $\boldsymbol{\theta}_l \in \mathbb{R}^{2\times 2}$ captures the affinity between values of the l-th attribute, with $\boldsymbol{\theta}_l[x_1, x_2] \in [0, 1]$ 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:

$$P(v_i, v_j) = \prod\nolimits_{l=1}^{k} \boldsymbol{\theta}_l[\mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)}]. \tag{4}$$

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.

3.4 UMHS: union of minimal hitting sets

Amburg et al. [33] 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 $S\backslash\{v\}$ remains a hitting set, and (3) repeat the above two steps with different permutations and take the union of the resulting hitting sets.

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

4 Proposed generation method: NoAH

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

4.1 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) [15], while in others, hyperedges among nodes with dissimilar attributes (heterophily) are common [35]. 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 [23, 24], 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, 19, 20]. 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).*

4.2 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 Fig. 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. $\mathcal{C} \cap \mathcal{F} = \varnothing$ 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 $\boldsymbol{p}_{\text{seed}} \in [0, 1]^{|\mathcal{C}|}$, where $\|\boldsymbol{p}_{\text{seed}}\|_1 = 1$, and $\boldsymbol{p}_{\text{seed}}(v)$

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:

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

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 $\mathcal{C}_{\tilde{e}}$. Each l-th attribute $\mathbf{x}_{\mathcal{C}_{\tilde{e}}}^{(l)}$ is independently sampled as:

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

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:

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

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.

Remark on relations among hyperedges.
Although NOAH constructs each hyperedge through a local attachment process without explicitly modeling relations between hyperedges, the generated hyperedges are coupled through shared global components, including the seed core distribution and the core affinity matrices. In particular, their core groups follow a common attribute-driven mechanism, and similar core group profiles tend to induce similar fringe attachment patterns. This shared parameter design avoids the additional parameters required to explicitly model hyperedge-level relations, allowing NOAH to remain scalable and interpretable.

4.3 Theoretical analysis of NOAH

We analyze the 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

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 $\tilde{\mathcal{H}} = (\mathcal{V}, \tilde{\mathcal{E}})$

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 [9, 20].

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

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

then the probability mass of an attribute vector becomes proportional to the $(-\delta)$-th power of the node's expected attachment probability (and hence its expected degree). When $|\mathcal{C}| \to \infty$, the resulting degree distribution can be viewed as a binomial distribution, and applying Stirling's approximation yields the probability of a node having degree d proportional to $d^{-\delta-\frac{1}{2}}$ as stated in Theorem 7.1 of [32].

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

$$P(\mathbf{x}_c^{(l)} = 1 | v_c \in g, \mathbf{x}_s^{(l)} = 1) = \frac{\mu_{\mathcal{C}}^{(l)} \times \boldsymbol{\theta}_{\mathcal{C}}^{(l)}[1,1]}{(1-\mu_{\mathcal{C}}^{(l)}) \times \boldsymbol{\theta}_{\mathcal{C}}^{(l)}[1,0] + \mu_{\mathcal{C}}^{(l)} \times \boldsymbol{\theta}_{\mathcal{C}}^{(l)}[1,1]}.$$

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

$$P(\mathbf{x}_c^{(l)} = 1 | v_c \in g, \mathbf{x}_s^{(l)} = 0) = \frac{\mu_{\mathcal{C}}^{(l)} \times \boldsymbol{\theta}_{\mathcal{C}}^{(l)}[0,1]}{(1-\mu_{\mathcal{C}}^{(l)}) \times \boldsymbol{\theta}_{\mathcal{C}}^{(l)}[0,0] + \mu_{\mathcal{C}}^{(l)} \times \boldsymbol{\theta}_{\mathcal{C}}^{(l)}[0,1]}.$$

If we set $\boldsymbol{\theta}_{\mathcal{C}}^{(l)}[0,0] \times \boldsymbol{\theta}_{\mathcal{C}}^{(l)}[1,1] = \boldsymbol{\theta}_{\mathcal{C}}^{(l)}[0,1] \times \boldsymbol{\theta}_{\mathcal{C}}^{(l)}[1,0]$, then the following holds:

$$P(\mathbf{x}_c^{(l)} = 1 | v_c \in g, \mathbf{x}_s^{(l)} = 1) = P(\mathbf{x}_c^{(l)} = 1 | v_c \in g, \mathbf{x}_s^{(l)} = 0).$$

Since the attribute distribution of attached core nodes is now independent of the seed core attribute, let's denote it as the l-th attribute following Bernoulli($\mu_a^{(l)}$). Moreover, the l-th attribute of the core group of a hyperedge 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 minus one. Then, if we adjust $\mu_{\mathcal{F}}^{(l)}$ so that

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

by the same argument as for the core degree distribution, using Theorem 7.1 of [32], as $|\mathcal{F}| \to \infty$, the probability that a fringe node has degree d is proportional to $d^{-\delta-\frac{1}{2}}$.

(iii) 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. □

5 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.

5.1 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 Sect. 3.4) 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 $\mathcal{F}_e = e \cap \mathcal{F}$.

5.2 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:

$$P_{\text{core}}(e) = \sum\nolimits_{v_s \in \mathcal{C}_e} \boldsymbol{p}_{\text{seed}}(v_s) \cdot P_{\mathcal{C}}(\mathcal{C}_e \setminus \{v_s\}|v_s),\ ^2 \tag{8}$$

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:

$$P_{\mathcal{C}}(\mathcal{C}_e \setminus \{v_s\}|v_s) = \prod_{v_c \in \mathcal{C}_e \setminus \{v_s\}} P_{\mathcal{C}}(v_c|v_s) \cdot \prod_{v_c \in \mathcal{C} \setminus \mathcal{C}_e} (1 - P_{\mathcal{C}}(v_c|v_s)). \tag{9}$$

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:

$$P_{\text{fringe}}(e) = \prod_{v_f \in \mathcal{F}_e} P_{\mathcal{F}}(v_f|\mathcal{C}_e) \cdot \prod_{v_f \in \mathcal{F} \setminus \mathcal{F}_e} (1 - P_{\mathcal{F}}(v_f|\mathcal{C}_e)), \tag{10}$$

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

$$\begin{aligned} P_{\mathcal{F}}(v_f|\mathcal{C}_e) &= \mathbb{E}_{\mathbf{x}_{\mathcal{C}_e}}\left[\prod\nolimits_{l=1}^{k} \boldsymbol{\theta}_{\mathcal{F}}^{(l)}[\mathbf{x}_{\mathcal{C}_e}^{(l)}, \mathbf{x}_f^{(l)}]\right] \\ &= \prod_{l=1}^{k} \left[(1 - p_e^{(l)}) \cdot \boldsymbol{\theta}_{\mathcal{F}}^{(l)}[0, \mathbf{x}_f^{(l)}] + p_e^{(l)} \cdot \boldsymbol{\theta}_{\mathcal{F}}^{(l)}[1, \mathbf{x}_f^{(l)}]\right], \end{aligned} \tag{11}$$

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

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

5.3 Parameter initialization

Below, we describe how the parameters of NoAHFit are initialized.

Seed Core Probability $\boldsymbol{p}_{\text{seed}}$.
We initialize the seed core probability of each node to be proportional to its degree, i.e.,

$$\boldsymbol{p}_{\text{seed}}(v_c) = \frac{|\{e \in \mathcal{E} | v_c \in e\}|}{\sum_{e \in \mathcal{E}} |\mathcal{C}_e|}. \tag{12}$$

Core and Fringe Affinity Matrices $\Theta_{\mathcal{C}}$ ***and*** $\Theta_{\mathcal{F}}$.

For each core and fringe affinity matrix, we initialize its entries to a constant value chosen so that the resulting mean expected cardinality of the core and fringe subsets matches those of the target hypergraph, that is,

$$\forall l \in \{1, \ldots, k\},\ \forall i \in \{0, 1\},\ \forall j \in \{0, 1\},$$

$$\theta_{\mathcal{C}}^{(l)}[i, j] = \left(\frac{\overline{c}_c - 1}{|\mathcal{C}|}\right)^{1/k}, \tag{13}$$

$$\theta_{\mathcal{F}}^{(l)}[i, j] = \left(\frac{\overline{c}_f}{|\mathcal{F}|}\right)^{1/k}, \tag{14}$$

where $\overline{c}_c$ and $\overline{c}_f$ denote the mean cardinalities of the core and fringe subsets in the target hypergraph.

5.4 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:

$$\mathcal{L}_{\text{edge}} = \sum_{e \in \mathcal{E}} - \log P(e). \tag{15}$$

- **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:

$$\mathcal{L}_{deg} = \text{MSE}(\boldsymbol{d}_c, \tilde{\boldsymbol{d}}_c) + \text{MSE}(\boldsymbol{d}_f, \tilde{\boldsymbol{d}}_f), \tag{16}$$

$$\mathcal{L}_{card} = \text{MSE}(\boldsymbol{c}_c, \tilde{\boldsymbol{c}}_c) + \text{MSE}(\boldsymbol{c}_f, \tilde{\boldsymbol{c}}_f), \tag{17}$$

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.

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

The final loss is a weighted sum of the above losses (see Line 11 of Algorithm 2), with the weights as hyperparameters.

5.5 Complexity of NoAHFit

We provide the time and space complexity analysis of NoAHFit. Regarding the time complexity, the core and fringe partitioning [33] 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 $O(k + m + |\mathcal{V}|)$.

6 Extensions to more general attribute settings

In this section, we demonstrate how NoAH can be applied beyond the setting considered in Sects. 4 and 5 to more realistic attribute scenarios. First, we generalize NoAH to support non-binary attributes, including categorical and continuous attributes. Then, we consider a more challenging setting in which node attributes are unavailable. These extensions broaden the applicability of NoAH to a wider range of real-world hypergraph data.

6.1 Extensions to non-binary attributes

While we have thus far focused on NoAH with binary node attributes, many real-world hypergraphs exhibit *non-binary* attributes. To address this, we generalize NoAH to support categorical and continuous attributes while preserving its core generative framework.

6.1.1 NoAH for categorical attributes

We first consider categorical attributes, where the l-th attribute value of a node v_i is $\mathbf{x}_i^{(l)} \in \{0, 1, \ldots, c_l - 1\}$, where c_l is the number of categories. We extend NoAH by generalizing the binary 2×2 affinity matrices with categorical counterparts, i.e., $\boldsymbol{\theta}_{\mathcal{C}}^{(l)} \in [0, 1]^{c_l \times c_l}$ and $\boldsymbol{\theta}_{\mathcal{F}}^{(l)} \in [0, 1]^{c_l \times c_l}$, for core group construction and fringe attachment, respectively.

Core Group Construction.
Given a seed core node v_s, the probability that another core node v_c is included in the core group is computed by multiplying the categorical affinities across attributes, i.e.,

$$P_{\mathcal{C}}(v_c | v_s, \Theta_{\mathcal{C}}) = \prod_{l=1}^{k} \boldsymbol{\theta}_{\mathcal{C}}^{(l)}[\mathbf{x}_s^{(l)}, \mathbf{x}_c^{(l)}], \tag{18}$$

which naturally generalizes Eq. (5).

Fringe Attachment.
For a hyperedge e with core group $\mathcal{C}_e$, we define a hyperedge-wise categorical group attribute for each l-th attribute based on the empirical category distribution within $\mathcal{C}_e$. Specifically, for

$$\pi_{\mathcal{C}_e, l}(a) = \frac{1}{|\mathcal{C}_e|} \sum_{v_c \in \mathcal{C}_e} \mathbb{I}\left(\mathbf{x}_c^{(l)} = a\right), \qquad a \in \{0, 1, \ldots, c_l - 1\}, \tag{19}$$

we sample the group attribute as

$$\mathbf{x}_{\mathcal{C}_e}^{(l)} \sim \text{Categorical}\left(\pi_{\mathcal{C}_e, l}\right). \tag{20}$$

Then, the probability that a node v_f attaches to the core group $\mathcal{C}_e$ is defined as

$$P_{\mathcal{F}}(v_f | \mathcal{C}_e, \Theta_{\mathcal{F}}) = \prod_{l=1}^{k} \boldsymbol{\theta}_{\mathcal{F}}^{(l)}[\mathbf{x}_{\mathcal{C}_e}^{(l)}, \mathbf{x}_f^{(l)}], \tag{21}$$

generalizing Eq. (7). Note that both core group construction and fringe attachment process reduce to the original binary formulation when $c_l = 2$ for each attribute dimension l.

6.1.2 NoAH for continuous attributes

We next consider continuous attributes, where the l-th attribute of node v_i is given as $\mathbf{x}_i^{(l)} \in \mathbb{R}$.

Attribute Projection Function.
To align with the binary formulation of NoAH, we introduce an *attribute projection function* for each attribute dimension that maps continuous values to the unit interval [0, 1]. For each attribute dimension l, we define an attribute projection function $f^{(l)}$, which is a non-decreasing function mapping from $\mathbb{R}$ to [0, 1]. Formally, $f^{(l)} \in \{g : \mathbb{R} \to [0,1] \mid g(x) \geq g(y), \forall x \geq y,\ x, y \in \mathbb{R}\}$. We denote the projected value of the l-th attribute of node v_i by $\tilde{\mathbf{x}}_i^{(l)} = f^{(l)}(\mathbf{x}_i^{(l)})$. In the experiments described in Sect. 7.6, we employ the following attribute projection function:

$$f^{(l)}(\mathbf{x}_i^{(l)}) = \sigma(\alpha^{(l)}\mathbf{x}_i^{(l)} + \beta^{(l)}), \tag{22}$$

where $\sigma(\cdot)$ denotes the sigmoid function and $\alpha^{(l)}, \beta^{(l)} \in \mathbb{R}$ are learnable parameters.

Continuous Affinity.
After projecting attributes, for each l-th attribute, we reuse the entries of the corresponding binary affinity matrix $\boldsymbol{\theta}_\star^{(l)} \in [0,1]^{2\times 2}$, where $\star \in \{\mathcal{C}, \mathcal{F}\}$. Given two projected attribute values $\tilde{\mathbf{x}}_i^{(l)}, \tilde{\mathbf{x}}_j^{(l)} \in [0, 1]$, we define the continuous affinity as a convex combination of the four binary affinities as follows:

$$\begin{aligned}\tilde{\boldsymbol{\theta}}_\star^{(l)}(\tilde{\mathbf{x}}_i^{(l)}, \tilde{\mathbf{x}}_j^{(l)}) = (1-\tilde{\mathbf{x}}_i^{(l)}) \times (1-\tilde{\mathbf{x}}_j^{(l)}) \times \boldsymbol{\theta}_\star^{(l)}[0,0] + (1-\tilde{\mathbf{x}}_i^{(l)}) \times \tilde{\mathbf{x}}_j^{(l)} \times \boldsymbol{\theta}_\star^{(l)}[0,1] \\ + \tilde{\mathbf{x}}_i^{(l)} \times (1-\tilde{\mathbf{x}}_j^{(l)}) \times \boldsymbol{\theta}_\star^{(l)}[1,0] + \tilde{\mathbf{x}}_i^{(l)} \times \tilde{\mathbf{x}}_j^{(l)} \times \boldsymbol{\theta}_\star^{(l)}[1,1],\end{aligned} \tag{23}$$

where $\star \in \{\mathcal{C}, \mathcal{F}\}$. Notably, Eq. (23) reduces to the original binary lookup when $\tilde{\mathbf{x}}_i^{(l)}, \tilde{\mathbf{x}}_j^{(l)} \in \{0, 1\}$.

Note that this continuous formulation above is intentionally simple and interpretable, serving as a continuous counterpart of the original binary formulation while preserving its attribute-wise affinity structure. Specifically, the bilinear interpolation in Eq. (23) keeps the 2×2 affinity matrix interpretable, where each entry corresponds to the affinity between low/high projected attribute values and, as mentioned above, reduces exactly to the original binary formulation when the projected values are restricted to {0, 1}. Similarly, the sigmoid projection in Eq. (22) is a simple monotone mapping to [0, 1], although other non-decreasing mappings can also be used. In Sect. 7.6, we empirically compare this continuous formulation with a more expressive neural affinity variant.

Core Group Construction.
While generalizing Eq. (5), given a seed core node v_s, we define the probability that another node v_c is included in the core group as

$$P_{\mathcal{C}}(v_c \mid v_s, \Theta_{\mathcal{C}}) = \prod_{l=1}^{k} \tilde{\boldsymbol{\theta}}_{\mathcal{C}}^{(l)}\left(\tilde{\mathbf{x}}_s^{(l)}, \tilde{\mathbf{x}}_c^{(l)}\right). \tag{24}$$

Fringe Attachment.
For a hyperedge e with core group $\mathcal{C}_e$, we define a hyperedge-wise continuous group attribute for each l-th attribute by averaging the normalized attribute values within $\mathcal{C}_e$:

$$\tilde{\mathbf{x}}_{\mathcal{C}_e}^{(l)} = \frac{1}{|\mathcal{C}_e|} \sum_{v_c \in \mathcal{C}_e} \tilde{\mathbf{x}}_c^{(l)}. \tag{25}$$

Then, the probability that a node v_f attaches to the core group $\mathcal{C}_e$ is defined as

$$P_{\mathcal{F}}(v_f \mid \mathcal{C}_e, \Theta_{\mathcal{F}}) = \prod_{l=1}^{k} \tilde{\boldsymbol{\theta}}_{\mathcal{F}}^{(l)} \left(\tilde{\mathbf{x}}_{\mathcal{C}_e}^{(l)}, \tilde{\mathbf{x}}_f^{(l)} \right), \tag{26}$$

which naturally extends Eq. (7).

Overall, this continuous attribute extension preserves the original generative steps of NoAH and reduces exactly to the binary formulation when attribute values are restricted to $\{0, 1\}$.

6.2 Extensions to settings without pre-existing attributes

While both NoAH and NoAHFit assume that node attributes are pre-existing, in many practical settings, such attributes are unavailable [15]. To broaden their applicability, this subsection presents extensions of NoAH and NoAHFit in which node attributes are learned to model attribute-driven interactions underlying hyperedge formation. Specifically, we first introduce NoAHFit- X, which represents latent node attributes via an attribute probability matrix and jointly optimizes these probabilities together with the parameters of NoAH under the same likelihood-based fitting framework. Then, we describe two extensions of NoAH with distinct strengths, namely NoAH- X and NoAH- X$^+$, which generate a hypergraph using the learned attribute probabilities through global attribute sampling and hyperedge-wise sampling, respectively.

6.2.1 Probabilistic representation of latent node attributes

In the setting without pre-existing attributes, the node attribute matrix $\mathbf{X} \in \{0, 1\}^{|\mathcal{V}| \times k}$ is not available. To enable learning without pre-existing attributes, while preserving the attribute-driven mechanisms of NoAH, we represent latent node attributes via an *attribute assignment probability matrix*, $\boldsymbol{\Phi} \in [0, 1]^{|\mathcal{V}| \times k}$, where k is the number of latent node attributes and is treated as a hyperparameter in practice. The i-th row of $\boldsymbol{\Phi}$, denoted as $\phi_i = \boldsymbol{\Phi}_{i,:} \in \mathbb{R}^k$, represents the attribute assignment vector of node $v_i \in \mathcal{V}$. We use $\phi_i^{(l)}$ to denote the probability that node $v_i \in \mathcal{V}$ has attribute value 1 for each l-th attribute that is binary. We interpret $\boldsymbol{\Phi}$ as parameterizing independent Bernoulli random variables for latent binary attributes, as follows:

$$P(\mathbf{X}|\boldsymbol{\Phi}) = \prod_{v_i \in \mathcal{V}} \prod_{l=1}^{k} (\mathbf{x}_i^{(l)} \times \phi_i^{(l)} + (1 - \mathbf{x}_i^{(l)}) \times (1 - \boldsymbol{\phi}_i^{(l)})). \tag{27}$$

Note that when binary attributes $\mathbf{X}$ are pre-existing, setting $\boldsymbol{\Phi} = \mathbf{X}$ reduces this formulation to the original setting.

The factorized Bernoulli formulation assumes independence across latent attribute dimensions. We adopt this formulation because it enables efficient learning and generation. However, this independence assumption may limit its ability to capture dependencies among latent attributes that could exist in real-world data. To examine this trade-off, in Appendix C, we further compare the factorized formulation with a dependency-aware latent attribute variant. The results show that the dependency-aware variant does not consistently improve performance, while requiring more parameters than NoAH, indicating that NoAH offers a favorable trade-off between expressiveness and parameter efficiency.

In addition, the learned latent attribute probability matrix Φ is not necessarily identifiable, since different latent representations may induce similar hyperedge likelihoods. Thus, Φ should not be interpreted as a uniquely recoverable estimate of true node attributes. Rather, it serves as a latent representation that captures node-level tendencies useful for reproducing realistic hyperedge formation patterns.

6.2.2 Fitting without pre-existing attributes

For parameter fitting, we extend NoAHFit to NoAHFit- X, which jointly tunes the attribute assignment probability matrix $\mathbf{\Phi}$ defined in Sect. 6.2.1 and the parameters of NoAH. The key idea is to replace the deterministic attribute lookup in NoAH with its expectation under $\mathbf{\Phi}$, yielding differentiable attachment probabilities. This allows NoAHFit- X to optimize both the NoAH parameters and $\mathbf{\Phi}$ within the same likelihood-based fitting framework as NoAHFit (Sect. 5).

Following Sect. 5.2, we define the likelihood of each hyperedge $e \in \mathcal{E}$ as $P^A(e|\mathcal{C}, \mathcal{F}, \mathbf{\Phi}, \boldsymbol{p}_{\text{seed}}, \Theta_{\mathcal{C}}, \Theta_{\mathcal{F}})$ and denote it by $P^A(e)$ for brevity. As before, $P^A(e)$ is decomposed into two components, $P^A_{\text{core}}(e)$ and $P^A_{\text{fringe}}(e)$, which correspond to the likelihoods of core group construction and fringe node attachment, respectively.

Likelihood of Sampling Core Nodes.
Recall that $P_{\text{core}}(e)$ in Eq. (8) depends on the core-core attachment probability $P_{\mathcal{C}}(v_c|v_s, \Theta_{\mathcal{C}})$. To incorporate uncertainty in attributes, we replace $P_{\mathcal{C}}$ with its expectation under $\mathbf{\Phi}$ and define $P^A_{\mathcal{C}}(v_c|v_s, \Theta_{\mathcal{C}}, \mathbf{\Phi})$, the probability that a seed core node v_s attaches to another core node $v_c \in \mathcal{C} \setminus \{v_s\}$, as follows:

$$
\begin{aligned}
P^A_{\mathcal{C}}(v_c|v_s) &= \mathbb{E}_{\mathbf{X}}\left[\prod_{l=1}^{k} \boldsymbol{\theta}^{(l)}_{\mathcal{C}}[\mathbf{x}^{(l)}_s, \mathbf{x}^{(l)}_c]\right] \\
&= \prod_{l=1}^{k} (\boldsymbol{\theta}^{(l)}_{\mathcal{C}}[0,0] \times (1-\phi^{(l)}_s) \times (1-\phi^{(l)}_c) + \boldsymbol{\theta}^{(l)}_{\mathcal{C}}[0,1] \times (1-\phi^{(l)}_s) \\
&\quad \times \phi^{(l)}_c + \boldsymbol{\theta}^{(l)}_{\mathcal{C}}[1,0] \times \phi^{(l)}_s \times (1-\phi^{(l)}_c) + \boldsymbol{\theta}^{(l)}_{\mathcal{C}}[1,1] \times \phi^{(l)}_s \times \phi^{(l)}_c),
\end{aligned}
\tag{28}
$$

where we omit $\Theta_{\mathcal{C}}$ and $\mathbf{\Phi}$ for brevity. By replacing $P_{\mathcal{C}}$ with $P^A_{\mathcal{C}}$ in Eq. (9), we obtain $P^A_{\text{core}}(e)$ for the unavailable attribute setting. Note that $P^A_{\mathcal{C}}$ reduces to the original $P_{\mathcal{C}}$ when $\mathbf{\Phi}$ is binary (i.e., $\mathbf{\Phi} = \mathbf{X}$).

Likelihood of Sampling Fringe Nodes.
Similarly, to incorporate $\mathbf{\Phi}$ in $P^A_{\text{fringe}}(e)$, we replace $P_{\mathcal{F}}$ with its expectation under $\mathbf{\Phi}$ and define $P^A_{\mathcal{F}}(v_f|\mathcal{C}_e, \Theta_{\mathcal{F}}, \mathbf{\Phi})$, the probability that a fringe node $v_f \in \mathcal{F}$ attaches to the core group $\mathcal{C}_e = \mathcal{C} \cap e$, as follows:

$$
\begin{aligned}
P^A_{\mathcal{F}}(v_f|\mathcal{C}_e) &= \mathbb{E}_{\mathbf{X}}\left[\prod_{l=1}^{k} \boldsymbol{\theta}^{(l)}_{\mathcal{F}}[\mathbf{x}^{(l)}_{\mathcal{C}_e}, \mathbf{x}^{(l)}_f]\right] \\
&= \prod_{l=1}^{k} (\boldsymbol{\theta}^{(l)}_{\mathcal{F}}[0,0] \times (1-\boldsymbol{\phi}^{(l)}_{\mathcal{C}_e}) \times (1-\boldsymbol{\phi}^{(l)}_f) + \boldsymbol{\theta}^{(l)}_{\mathcal{F}}[0,1] \times (1-\boldsymbol{\phi}^{(l)}_{\mathcal{C}_e}) \\
&\quad \times \boldsymbol{\phi}^{(l)}_f + \boldsymbol{\theta}^{(l)}_{\mathcal{F}}[1,0] \times \boldsymbol{\phi}^{(l)}_{\mathcal{C}_e} \times (1-\boldsymbol{\phi}^{(l)}_f) + \boldsymbol{\theta}^{(l)}_{\mathcal{F}}[1,1] \times \boldsymbol{\phi}^{(l)}_{\mathcal{C}_e} \times \boldsymbol{\phi}^{(l)}_f),
\end{aligned}
\tag{29}
$$

where $\boldsymbol{\phi}^{(l)}_{\mathcal{C}_e} = \frac{1}{|\mathcal{C}_e|}\sum_{v_c \in \mathcal{C}_e} \boldsymbol{\phi}^{(l)}_c$ denotes the expectation of the l-th attribute for core group $\mathcal{C}_e$. By replacing $P_{\mathcal{F}}$ with $P^A_{\mathcal{F}}$ in Eq. (11), we obtain $P^A_{\text{fringe}}(e)$ for the unavailable attribute setting. Note that $P^A_{\mathcal{F}}$ reduces to the original $P_{\mathcal{F}}$ when $\mathbf{\Phi}$ is binary (i.e., $\mathbf{\Phi} = \mathbf{X}$).

Update of Parameters.
Finally, NoAHFit- X jointly optimizes $\mathbf{\Phi}$ and the parameters of NoAH by minimizing the same loss function defined in Sect. 5.4, while treating $\mathbf{\Phi}$ as additional learnable variables.

6.2.3 Hypergraph generation without pre-existing attributes

As described above, by using NoAHFit- X, we obtain the estimated NoAH parameters $(\boldsymbol{p}_{\text{seed}}, \Theta_{\mathcal{C}}, \Theta_{\mathcal{F}})$ together with the attribute assignment probability matrix $\mathbf{\Phi}$. We now describe how to generate hyperedges using $\mathbf{\Phi}$ without pre-existing attributes. The key design choice is how to sample latent attributes from $\mathbf{\Phi}$ during generation, which leads to two generation schemes with distinct strengths, NoAH- X and NoAH- X$^+$. NoAH- X samples attributes once globally and reuses them across all hyperedges for efficiency, whereas NoAH- X$^+$ performs attribute resampling for each hyperedge, yielding greater diversity. Both schemes preserve the original generative steps of NoAH for sampling a seed node, constructing a core group, and attaching fringe nodes; they differ only in how latent attributes are sampled.
NoAH- X: ***Global Attribute Sampling.***
NoAH- X determines all node attributes at the beginning by sampling latent node attributes once for each node and then generates hyperedges using original NoAH using the sampled attributes. Specifically, we draw $\tilde{\mathbf{X}}$ as follows:

$$\tilde{\mathbf{X}} \sim P(\mathbf{X} \mid \mathbf{\Phi}), \tag{30}$$

where $P(\mathbf{X} \mid \mathbf{\Phi})$ is the factorized Bernoulli model in Eq. (27). That is, for each node $v_i \in \mathcal{V}$ and attribute $l \in \{1, \ldots, k\}$, we sample $\tilde{\mathbf{x}}_i^{(l)} \sim \text{Bernoulli}(\phi_i^{(l)})$ independently. Then, given the sampled attribute matrix $\tilde{\mathbf{X}}$, NoAH- X generates each hyperedge e by following the same procedure as NoAH, as described in Sect. 4.
NoAH- X$^+$: ***Hyperedge-wise Attribute Sampling.*** In contrast with NoAH- X, which samples node attributes once at the global level, NoAH- X$^+$ resamples latent attributes *on the fly* during the generation of each hyperedge to reflect the uncertainty encoded in $\mathbf{\Phi}$.

Specifically, for each hyperedge e, NoAH- X$^+$ samples hyperedge-specific attributes $\tilde{\mathbf{X}}_e$ from the same factorized Bernoulli model as in Eq. (30). That is, $\tilde{\mathbf{x}}_{e,i}^{(l)} \sim \text{Bernoulli}(\phi_i^{(l)})$ independently for all nodes $v_i \in \mathcal{V}$ and attribute dimensions $l \in \{1, \ldots, k\}$. Conditioned on $\tilde{\mathbf{X}}_e$, the hyperedge is then generated following the original procedure of NoAH, as described in Sect. 4. This sampling and generation process is repeated independently for each hyperedge, allowing different attributes to be used across hyperedges. This design reflects the observation that, in real-world hypergraphs, node attributes or states can be edge dependent [2, 5, 36, 37], varying according to the particular hyperedge in which the node participates.

Compared to NoAH- X, NoAH- X$^+$ increases diversity across generated hyperedges by resampling attributes for each hyperedge, especially when $\mathbf{\Phi}$ contains substantial uncertainty, that is, when it is far from a binary matrix. We empirically confirm this effect in Sect. 7.7 through hyperedge diversity analysis, and we show the diversity produced by NoAH- X$^+$ is closer to the ground truth than that produced by NoAH- X. Note that both NoAH- X and NoAH- X$^+$ reduce to original NoAH when $\mathbf{\Phi}$ is binary (i.e., $\mathbf{\Phi} = \mathbf{X}$), since the sampled attributes then coincide with the deterministic attribute matrix.

Complexity Analysis of NoAH- X ***and*** NoAH- X$^+$.

We provide the time and space complexity analysis of NoAH- X and NoAH- X$^+$. Both methods inherit the complexity of NoAH, with additional overhead arising from node attribute sampling. A single run of node attribute sampling takes $O(k|\mathcal{V}|)$ time. Since NoAH- X performs this step only once, its additional time complexity is $O(k|\mathcal{V}|)$. On the other hand, NOAH- X$^+$ performs node attribute sampling for each of the m hyperedges, leading to an additional time complexity of $O(mk|\mathcal{V}|)$. For space complexity, both methods require storing the sampled node attributes, which takes $O(k|\mathcal{V}|)$ space. Therefore, when combined with the complexity of NOAH in Sect. 4.3 (time complexity $O(mk|\mathcal{V}|)$, space complexity $O(k)$), both NOAH- X and NOAH- X$^+$ have overall time $O(mk|\mathcal{V}|)$ and space complexity $O(k|\mathcal{V}|)$.

7 Experiments

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

7.1 Experimental settings

Datasets. We use nine real-world hypergraphs from four distinct domains (see Table 2 for some statistics):

- **Academic Paper Domain (Citeseer, Cora** [38]**):** 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** [39], **Workspace** [40]**):** 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. For the main experiments, we apply one-hot encoding to convert categorical attributes into binary attributes.
- **Review Domain (Amazon Music** [41], **Yelp Restaurant, Yelp Bar** [42]**):** 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 Domain**[2] **(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). Some datasets are constructed from temporally ordered events. In the main experiments, we treat all datasets as static attributed hypergraphs, while we additionally examine the effect of incorporating temporal attributes on the review domain datasets and online Q&A-domain datasets in Appendix B.

Baselines.
We consider eight baseline generative models: HYPERCL [7], HYPERPA [6], HYPERFF [43], HYPERLAP [7], hyper dK [44], THERA [8], HYCOSBM [12], HYREC [45]. Among the baselines, HYCOSBM explicitly utilizes node attributes. For HYPERCL, HYPERLAP, and hyper dK-series (hyper dK), since node identities are preserved in the generated hypergraphs, we assign node attributes based on their correspondence to the original nodes. For HYPERPA,

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

Table 2 Summary statistics of 9 real-world hypergraphs from 4 domains

Dataset	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	k	$\|\mathcal{C}\|$	$\|\mathcal{F}\|$
Citeseer	1458	1079	3703	597	861
Cora	2388	1072	1433	841	1547
High School	327	7818	12	288	39
Workspace	92	788	5	71	21
Amazon Music	1106	686	7	379	727
Yelp Restaurant	565	594	9	273	292
Yelp Bar	1234	1188	15	625	609
Devops	5010	5684	429	2003	3007
Patents	4458	4669	2170	894	3564

$|\mathcal{V}|$: 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

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 A.

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 Sect. 3.2. 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.

7.2 Performance comparison

As shown in Table 3, 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 higher-order hyperedge entropy. These results demonstrate the overall superiority 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, perform well. Second, higher-order hyperedge entropy (HOHE) remains a relatively challenging metric for NoAH. While NoAH is competitive with THera on average, HyperLAP achieves a

Table 3 NoAH reproduces structure–attribute interplays best overall across 9 datasets

	T2	T3	T4	HE	HOHE	NHS	A.R.	T2	T3	T4	HE	HOHE	NHS	A.R.
HyperCL	6,816	10,702	10,672	19.81	122.02	19.94	6.5	2,099	4,206	*4,481*	8.02	63.64	7.99	6.8
HyperPA	6,871	10,739	10,559	25.90	103.76	22.46	7.7	2,081	4,315	4,615	14.47	62.62	12.78	8.2
HyperFF	6,472	***10,483***	10,677	*17.60*	66.97	**15.23**	4.0	2,040	4,094	4,518	8.06	***39.58***	***6.76***	5.0
HyperLAP	6,757	10,737	10,311	18.33	55.40	19.43	5.0	***2,031***	4,083	4,523	7.65	53.40	7.13	5.2
hyper dK	6,968	*10,234*	*10,154*	**13.38**	102.83	***16.09***	*3.7*	2,094	***4,048***	4,502	7.24	59.79	7.49	5.2
THera	***6,450***	10,498	10,476	***18.04***	***53.58***	18.59	***3.8***	*2,010*	**4,003**	***4,492***	***6.94***	*40.35*	7.20	**2.8**
HyCoSBM	*6,285*	10,613	10,278	19.89	135.30	38.60	5.8	**1,947**	4,049	4,502	7.24	69.77	15.02	5.2
HyRec	6,996	10,840	***10,271***	20.01	*51.37*	20.04	6.2	2,075	4,163	**4,465**	*6.51*	**19.56**	*6.54*	*3.2*
NoAH	**5,734**	**10,157**	**10,151**	24.31	**44.26**	*15.39*	**2.3**	2,058	*4,011*	4,517	**6.41**	41.64	**5.79**	*3.2*
		(a) Citeseer (NoAH ranks **first** overall)							(b) Cora (NoAH ranks **second** overall)					
HyperCL	20.4	***51.4***	106.4	1.180	1.369	1.730	6.3	7.5	17.3	***13.0***	0.599	0.181	0.890	6.5
HyperPA	20.6	52.6	107.8	1.175	1.421	1.654	7.2	7.1	19.5	**12.5**	0.562	0.217	0.835	5.5
HyperFF	20.2	61.6	106.5	1.017	1.290	1.785	6.7	6.3	14.9	13.9	0.445	0.399	***0.801***	***4.7***
HyperLAP	20.3	51.6	102.0	1.187	*0.977*	1.714	5.7	7.3	19.9	***12.6***	0.564	0.184	0.878	6.2
hyper dK	20.0	***51.4***	***97.8***	***0.931***	1.429	1.716	4.8	6.5	15.6	25.1	***0.408***	***0.178***	0.814	4.8
THera	19.9	*51.2*	99.8	1.166	**0.781**	*1.611*	*3.2*	***5.7***	***13.7***	16.7	0.542	*0.144*	0.820	*3.8*
HyCoSBM	**3.9**	52.1	97.9	1.819	1.414	1.797	6.0	**2.2**	*13.0*	24.0	0.705	0.233	0.885	5.8
HyRec	***19.8***	54.0	*92.8*	*0.740*	*0.960*	***1.637***	***3.3***	7.3	18.0	20.3	*0.371*	0.301	*0.753*	5.5
NoAH	*12.2*	**37.8**	**89.6**	**0.628**	1.273	**1.374**	**1.7**	*4.2*	**9.7**	20.0	**0.062**	**0.133**	**0.619**	**2.0**

Table 3 continued

	T2	T3	T4	HE	HOHE	NHS	A.R.	T2	T3	T4	HE	HOHE	NHS	A.R.
	(c) High School (NoAH ranks **first** overall)							(d) Workspace (NoAH ranks **first** overall)						
HyperCL	27.3	53.0	63.6	1.016	0.382	1.053	5.5	***17.8***	58.9	90.4	0.979	*0.249*	1.148	6.3
HyperPA	27.3	55.4	71.2	1.154	0.527	1.096	8.0	24.8	55.9	*76.0*	0.786	0.591	1.068	6.2
HyperFF	***24.1***	54.3	*60.5*	***0.449***	**0.299**	1.055	*4.0*	22.4	***54.1***	79.2	*0.459*	0.603	1.157	5.2
HyperLAP	27.3	52.4	68.3	1.026	*0.361*	1.042	5.2	22.4	54.6	85.9	1.012	**0.217**	1.129	5.8
hyper dK	31.4	52.4	***61.6***	1.249	0.450	1.026	6.3	24.6	54.5	82.9	0.726	0.511	0.978	5.3
THera	26.0	***50.6***	67.0	0.976	0.394	1.003	***4.3***	21.0	54.3	80.6	***0.704***	0.484	***0.964***	***3.8***
HyCoSBM	**11.8**	57.9	72.1	*0.306*	***0.371***	***0.900***	***4.3***	**5.0**	**52.2**	102.5	**0.281**	***0.403***	*0.932*	**2.8**
HyRec	25.3	***50.6***	61.8	1.138	0.402	***0.982***	4.5	24.7	55.7	***77.6***	0.797	0.523	0.968	5.7
NoAH	*21.0*	**47.8**	**55.1**	**0.275**	0.394	**0.229**	**1.8**	*9.3*	*53.0*	**66.2**	0.967	1.670	**0.524**	*3.7*
	(e) Amazon Music (NoAH ranks **first** overall)							(f) Yelp Restaurant (NoAH ranks **second** overall)						
HyperCL	57.1	112.1	***133.2***	1.392	1.064	1.343	6.3	1,257	3,450	***5,793***	***4.89***	30.86	*8.45*	***3.3***
HyperPA	54.7	114.4	139.9	1.090	1.391	1.283	6.2	2,428	6,670	9,427	26.22	66.76	20.30	8.3
HyperFF	54.6	115.2	*129.4*	***0.635***	**0.700**	1.430	4.5	2,382	6,605	9,204	25.71	66.74	20.89	7.3
HyperLAP	***54.1***	***107.2***	137.0	1.345	***0.978***	1.325	4.8	***1,246***	***3,363***	*5,780*	*4.82*	***30.63***	***8.99***	*2.7*
hyper dK	57.1	108.7	134.2	1.572	1.102	1.230	6.2	**1,136**	*2,765*	**4,787**	6.86	*28.56*	10.27	**2.3**
THera	55.5	*106.0*	133.3	1.124	1.061	***1.221***	*4.2*	2,387	6,567	9,136	25.90	65.53	20.52	7.0
HyCoSBM	**26.3**	128.9	161.3	**0.526**	1.053	*1.140*	***4.3***	*1,226*	**2,696**	7,027	**4.52**	53.83	25.43	3.7
HyRec	58.3	111.3	**123.8**	1.343	*0.953*	1.225	4.5	2,394	6,736	8,943	23.31	59.88	15.96	6.7

Table 3 continued

	T2	T3	T4	HE	HOHE	NHS	A.R.	T2	T3	T4	HE	HOHE	NHS	A.R.
NoAH	*44.7*	**94.0**	145.2	*0.589*	1.598	**0.468**	**3.8**	1,714	5,182	8,038	12.81	**19.75**	**6.59**	3.7
		(g) Yelp Bar (NoAH ranks **first** overall)							(h) Devops (NoAH ranks **fourth** overall)					
HyperCL	6,586	**13,091**	**16,515**	64.30	***105.59***	*103.33*	**2.5**	6.2	5.4	4.4	6.1	5.0	6.2	6.2
HyperPA	13,538	27,184	31,883	268.45	466.07	233.28	8.8	7.3	8.0	6.8	7.2	7.8	6.9	9.0
HyperFF	12,069	26,032	29,718	262.36	440.24	230.16	6.2	4.7	5.9	5.7	***4.3***	5.0	6.1	5.3
HyperLAP	***6,551***	*13,417*	*16,663*	***61.98***	*104.40*	***104.40***	**2.5**	4.7	4.9	5.0	5.8	**3.1**	5.2	4.5
hyper dK	*6,262*	15,041	23,320	*53.40*	**85.05**	**91.81**	**2.5**	5.8	*3.6*	***4.2***	***4.3***	5.2	***4.3***	*3.7*
THera	12,856	26,900	30,711	266.45	464.98	232.26	7.8	***4.6***	*3.6*	5.4	4.9	*4.2*	4.6	*3.7*
HyCoSBM	**5,828**	***13,984***	***18,112***	**48.52**	265.48	230.81	3.3	**1.2**	4.4	6.0	*3.8*	5.8	6.3	5.2
HyRec	14,769	26,521	29,705	263.42	442.54	228.55	6.8	7.0	6.8	**3.7**	5.0	4.6	*3.9*	4.8
NoAH	7,292	17,557	22,532	65.32	109.78	134.08	4.5	*2.9*	**2.1**	**3.7**	**3.4**	*4.2*	**1.4**	**1.3**
		(i) Patents (NoAH ranks **fifth** overall)							(j) Average Rank over Nine Datasets					

The top three results are indicated by **boldface** (first), *italics* (second), and ***bolditalic*** (third). A.R. denotes average rank

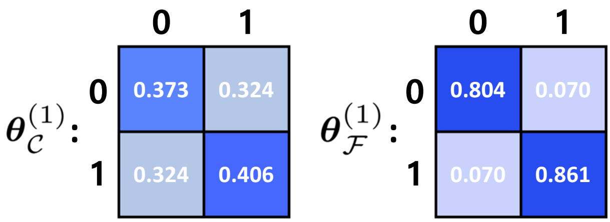


Fig. 3 $\boldsymbol{\theta}_{\mathcal{C}}^{(1)}$ and $\boldsymbol{\theta}_{\mathcal{F}}^{(1)}$ estimated by NoAHFit on the Amazon Music dataset. NoAHFit captures homophily by assigning higher affinities to same attribute value pairs ($0 \leftrightarrow 0$ and $1 \leftrightarrow 1$) than to different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

better average rank on HOHE. This is because, due to the label propagation steps, HOHE is strongly influenced by hyperedge overlaps, which HyperLAP and THera explicitly target.

7.3 Case study

To gain deeper insight into the effectiveness of NoAH, we conduct a case study on the Amazon Music dataset [41]. 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 Fig. 1 in Sect. 1, 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 Fig. 3.

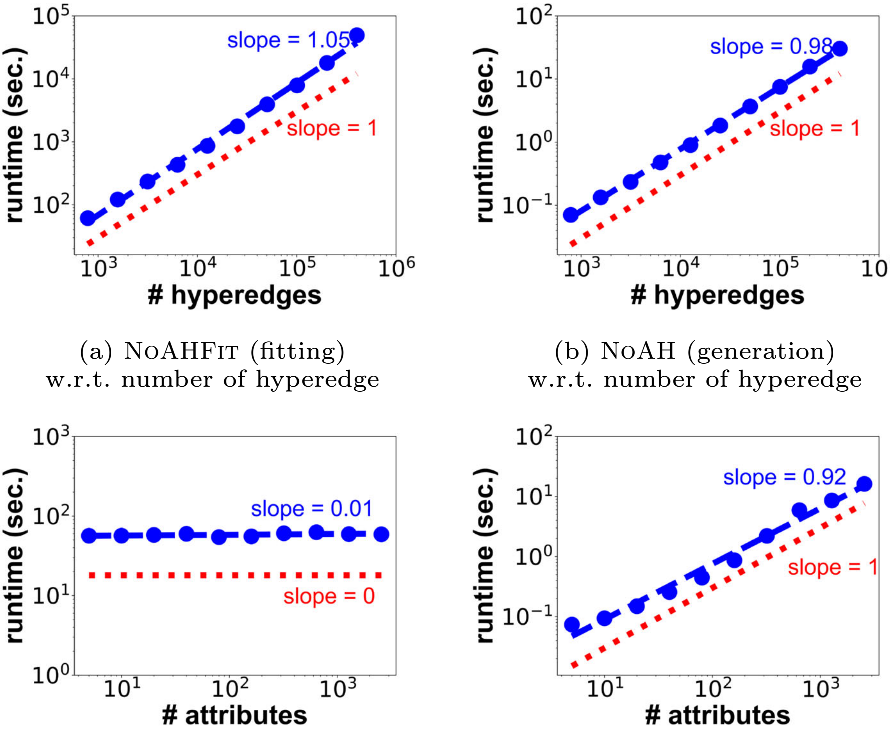
7.4 Ablation study

To assess the contribution of the core–fringe node hierarchy in NoAH, we compare its performance with two variants: NoAH-noCF, which omits this hierarchical structure, and NoAH-dCF, which selects the same number of cores as NoAH based on the highest degrees. In NoAH-noCF, 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), whereas NoAH-dCF follows the generation scheme of NoAH but differs only in the core selection algorithm. As reported in Table 4, NoAH-noCF consistently performs the worst among the three variants, which demonstrates the importance of introducing the core–fringe hierarchy in realistic hyperedge formation. Moreover, NoAH-dCF substantially improves upon NoAH-noCF across all datasets, showing that introducing a core–fringe split itself already provides large benefits. At the same time, NoAH achieves the best average rank overall and outperforms NoAH-dCF on most datasets, indicating that the effectiveness of NoAH comes not only from the existence of the split, but also from the quality of the UMHS-based core selection.

Table 4 Ablation study with two variants on core–fringe node hierarchy, NOAH-noCF and NOAH-dCF, described in Sect. 7.4

	T2	T3	T4	HE	HOHE	NHS	A.R.	T2	T3	T4	HE	HOHE	NHS	A.R.
NOAH-noCF	9,036	14,550	13,515	48.26	106.32	121.04	3.0	3,322	5,951	5,995	29.01	57.35	70.15	3.0
NOAH-dCF	*6,603*	*11,279*	*10,749*	*32.65*	*67.73*	*19.21*	*2.0*	*2,151*	*4,041*	**4,342**	*8.02*	**35.61**	*6.74*	*1.7*
NOAH	**5,734**	**10,157**	**10,151**	**24.31**	**44.26**	**15.39**	**1.0**	**2,058**	**4,011**	*4,517*	**6.41**	*41.64*	**5.79**	**1.3**
	(a) Citeseer							(b) Cora						
NOAH-noCF	19.2	50.3	103.9	0.682	**1.176**	1.635	2.7	7.3	18.4	**12.5**	0.186	0.283	0.905	2.7
NOAH-dCF	*14.9*	*42.6*	**89.5**	*0.643*	1.275	*1.456*	*2.0*	**4.0**	*17.0*	*20.0*	*0.071*	**0.128**	**0.584**	**1.5**
NOAH	**12.2**	**37.8**	*89.6*	**0.628**	*1.273*	**1.374**	**1.3**	*4.2*	**9.7**	*20.0textit*	**0.062**	*0.133*	*0.619*	*1.7*
	(c) High School							(d) Workspace						
NOAH-noCF	*21.8*	49.7	*58.0*	0.363	1.188	0.402	2.7	21.6	66.5	86.3	1.232	3.129	1.539	3.0
NOAH-dCF	29.3	**40.7**	62.4	**0.161**	*0.409*	*0.268*	*2.0*	*14.7*	**38.0**	**64.0**	**0.810**	**1.326**	**0.437**	**1.2**
NOAH	**21.0**	*47.8*	**55.1**	*0.275*	**0.394**	**0.229**	**1.3**	**9.3**	*53.0*	*66.2*	*0.967*	*1.670*	*0.524*	*1.8*
	(e) Amazon Music							(f) Yelp Restaurant						
NOAH-noCF	61.7	113.9	*127.9*	0.827	2.955	1.630	2.8	2,158	6,330	9,199	30.05	77.31	14.22	3.0
NOAH-dCF	**30.2**	**77.5**	**103.6**	**0.364**	**1.435**	**0.414**	**1.0**	**1,691**	**5,026**	**7,461**	*16.00*	*28.13*	*9.262*	**1.5**
NOAH	*44.7*	*94.0*	145.2	*0.589*	*1.598*	*0.468*	*2.2*	*1,714*	*5,182*	*8,038*	**12.81**	**19.75**	**6.59**	**1.5**
	(g) Yelp Bar							(h) Devops						
NOAH-noCF	14,298	27,920	31,497	268.81	468.97	236.31	3.0	2.9	3.0	2.6	3.0	2.8	3.0	3.0
NOAH-dCF	*7,533*	*18,512*	**22,387**	*131.37*	*159.81*	*140.65*	*1.8*	*1.8*	*1.6*	**1.4**	*1.7*	*1.7*	*1.7*	*1.8*
NOAH	**7,292**	**17,557**	*22,532*	**65.32**	**109.78**	**134.08**	**1.2**	**1.3**	**1.4**	*1.9*	**1.3**	**1.6**	**1.3**	**1.2**
	(i) Patents							(j) Average Rank over Nine Datasets						

The best results are shown in **bold**, and the second-best results are *italic*. A.R. denotes average rank



(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. 4 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

7.5 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 Workspace dataset by factors ranging from 2 to 512. Figure 4 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.

7.6 NoAH with non-binary attributes

We examine the behavior of NoAH with non-binary node attributes, specifically categorical and continuous attributes, using the extensions described in Sect. 6.1.

7.6.1 Experimental setup

Datasets.
For categorical attribute settings, we use the contact domain datasets (High School and Workspace) described in Sect. 7.1, in which each attribute takes values from a finite set of categories. For continuous attribute settings, we use the PubMed [38] dataset, a co-citation hypergraph where each node represents a publication and each hyperedge represents a set of publications co-cited by a publication. Node attributes are provided as TF-IDF vectors. The PubMed dataset contains 3,840 nodes, 7,963 hyperedges, and an attribute dimension of 500.

Baselines.
For categorical attribute settings, we compare the categorical extension of NoAH (denoted as NoAH-categorical) against the binary attribute version of NoAH (denoted as NoAH-binary) and the two strongest baselines in terms of structure–attribute interplay (i.e., hyper dK and THera). For continuous attribute settings, we compare two continuous versions of NoAH, NoAH-continuous and NoAH-neural (described in Sect. 7.6.3), against the two strongest baselines in terms of structure–attribute interplay (i.e., hyper dK and THera).

Evaluation.
For categorical attribute settings, we employ the structure–attribute interplay measures (type-s affinity ratio scores, hyperedge entropy, and node homophily score), described in Sect. 3.2, for evaluation. For continuous attribute settings, since we cannot directly apply the structure–attribute interplay metrics in Sect. 3.2, which are specific to discrete attributes, we examine continuous attribute coherence at the node and hyperedge levels using cosine similarity on TF-IDF feature vectors. Specifically, we use (1) *node cosine similarity* measures: For each node, we compute the average cosine similarity between its attribute vector and those of its neighbors, yielding a distribution over nodes; and (2) *hyperedge cosine similarity* measures: For each hyperedge, we compute the mean pairwise cosine similarity over all node pairs within the hyperedge, yielding a distribution over hyperedges. For each measure, we visualize the resulting distributions using histograms and quantify the discrepancy between the generated and ground truth distributions using the Wasserstein distance.

7.6.2 Results on categorical attributes

Table 5 reports the structure–attribute interplay evaluation in the categorical attribute setting. Overall, NoAH-categorical attains competitive average ranks across datasets and, on average, improves upon both the binary attribute variant (NoAH-binary) and the strongest baselines (hyper dK and THera) in terms of structure–attribute interplay. Compared with NoAH-binary, NoAH-categorical improves most type-s affinity ratio metrics, reducing the discrepancies for T2 and T3 on both datasets and for T4 on Workspace. Since type-s affinity ratio scores capture fine-grained hyperedge-level structure–attribute patterns, these results suggest that explicit categorical modeling helps reproduce such patterns more faithfully than naively reducing categorical attributes to binary representations. NoAH-categorical also improves NHS on both datasets, indicating better reproduction of node-level homophily patterns.

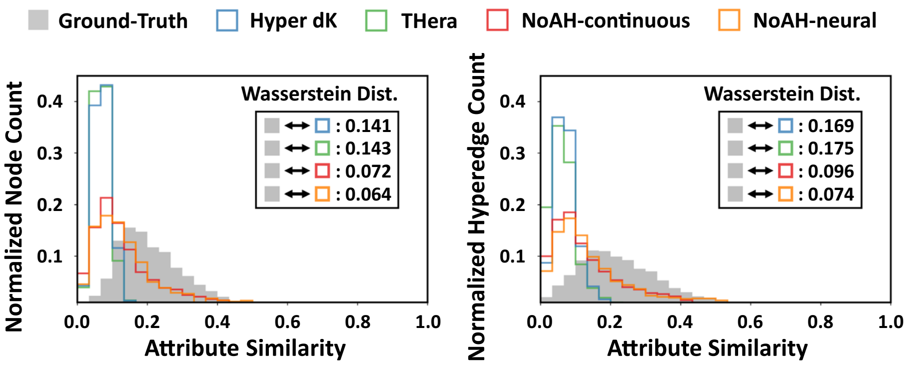
7.6.3 Results on continuous attributes

We present a case study on the PubMed dataset with continuous TF-IDF attributes. We first apply the continuous version of NoAH, denoted by NoAH-continuous, which uses the

Table 5 Structure–attribute interplay evaluation in the categorical attribute setting

	T2	T3	T4	HE	HOHE	NHS	A.R.	T2	T3	T4	HE	HOHE	NHS	A.R.
hyper dK	22.1	55.4	*105.4*	*0.573*	0.443	1.033	3.2	6.5	15.6	25.1	0.223	*0.039*	0.403	3.2
THERA	21.0	56.3	108.2	0.672	*0.282*	0.920	3.0	6.4	16.3	**10.9**	0.282	**0.033**	0.418	2.8
NOAH-binary	*12.9*	*42.9*	**98.5**	**0.568**	0.426	*0.814*	**1.8**	*4.2*	*9.7*	20.0	**0.032**	0.052	*0.309*	*2.2*
NOAH-categorical	**4.0**	**29.3**	119.2	0.940	**0.210**	**0.438**	*2.0*	**2.7**	**7.2**	*14.8*	*0.147*	0.114	**0.142**	**1.8**
	(a) High School							(b) Workspace						

Overall, NOAH-categorical achieves competitive average ranks and, on average, improves upon NOAH-binary and the strongest baselines (hyper dK and THERA), with particularly notable gains on T2, T3, and NHS. The best results are shown in **bold**, and the second-best are *italic*. A.R. denotes average rank



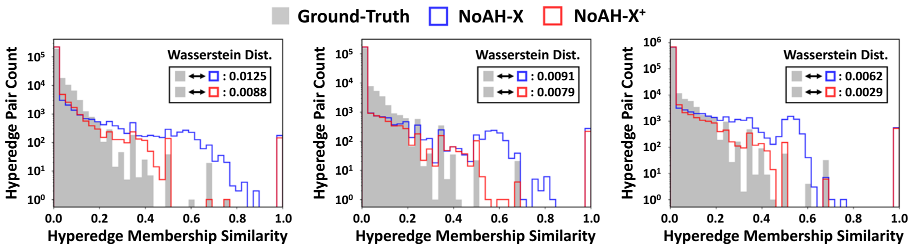
(a) Node cosine similarity distribution on PubMed

(b) Hyperedge cosine similarity distribution on PubMed

Fig. 5 A case study on the PubMed dataset with continuous attributes. We compare the distributions of (left) node cosine similarity (average attribute similarity between a node and its neighbors) and (right) hyperedge cosine similarity (mean pairwise attribute similarity over node pairs within a hyperedge). Both NOAH-continuous and NOAH-neural yield lower Wasserstein distances than hyper dK and THERA, indicating that NOAH-based continuous attribute modeling better captures attribute-based interactions at both node and hyperedge levels. Moreover, NOAH-neural achieves the lowest Wasserstein distances among the compared methods, showing the potential benefit of using a more expressive neural affinity function for continuous attributes

bilinear continuous affinity formulation described in Sect. 6.1.2. To examine whether a more expressive affinity function can better capture continuous attribute–structure relationships, we also consider a neural variant, NOAH-neural. NOAH-neural replaces the bilinear interpolation in Eq. (23) with a lightweight neural affinity function. For each attribute l, a shared one-layer MLP maps a pair of projected continuous values to a hidden representation, and an attribute-specific output head maps it to an affinity score in [0, 1]. The resulting affinity scores are multiplied across attributes, following the same multiplicative structure as NOAH-continuous. Although NOAH-neural is more expressive than NOAH-continuous, this comes at the cost of the simplicity and interpretability of the bilinear formulation.

Figure 5 shows the distributions of the node and hyperedge cosine similarity measures for the ground truth hypergraph and those generated by THERA, hyper dK, NOAH-continuous, and NOAH-neural, with the Wasserstein distance from each generated distribution to the ground truth distribution. Compared to the baselines, both NOAH-continuous and NOAH-neural produce distributions that more closely follow the trend of ground truth. In particular, they place greater mass on higher cosine similarity values, which is consistent with generating interactions among nodes with more similar attributes. Correspondingly, both NOAH-based continuous variants achieve smaller Wasserstein distances than THERA and hyper dK, suggesting that they better preserve attribute–coherence patterns at both node and hyperedge levels. Between the two variants, NOAH-neural further reduces the Wasserstein distances compared to NOAH-continuous, suggesting that its more expressive neural affinity function better captures continuous attribute–structure relationships on the PubMed dataset. At the same time, the gap between NOAH-neural and NOAH-continuous remains relatively small in this case study, indicating that the original bilinear formulation still serves as an effective continuous extension of NOAH while remaining simple and interpretable.



(a) Amazon Music (b) Yelp Restaurant (c) Yelp Bar

Fig. 6 Hyperedge diversity in the setting without pre-existing attributes. We report the distribution of pairwise similarity between (the node sets of) generated hyperedges in terms of Jaccard similarity. NOAH- X tends to yield more highly similar hyperedge pairs, whereas NOAH- X^{+} more closely matches the ground truth distribution by increasing diversity through hyperedge-wise attribute sampling

7.7 Setting without pre-existing attributes

In this subsection, we evaluate NOAH- X and NOAH- X^{+}, the extensions of NOAH for the setting without pre-existing attributes (Sect. 6.2), by examining whether they can reproduce realistic structure–attribute interplay in the absence of pre-existing attributes.

7.7.1 Experimental settings

We use the review domain datasets (Amazon Music, Yelp Restaurant, and Yelp Bar) with moderate attribute dimensions. Specifically, we retain the hypergraph structure $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ while withholding the ground truth attribute matrix $\mathbf{X}$ during fitting and generation; $\mathbf{X}$ is used only for evaluation.

We compare NOAH- X and NOAH- X^{+} with hyper dK and THERA, the two baseline methods that best reproduce structure–attribute interplay (Table 3). We also include NOAH (fitted by NOAHFIT), which relies on pre-existing attributes, for comparison. Both NOAH- X and NOAH- X^{+} are fitted by NOAHFIT- X, where the latent attribute dimension k is treated as a hyperparameter. As a reference setting, we first report the results obtained when k is set to the ground truth attribute dimension in Sect. 7.7.2, and we further evaluate a range of alternative values of k in the sensitivity analysis in Sect. 7.7.3.

As an additional comparison, we further examine the limitation of the factorized Bernoulli assumption by considering a categorical latent attribute variant that can represent dependencies among multiple binary attributes; details and results are provided in Appendix C.

All methods are evaluated using the structure–attribute interplay measures described in Sect. 7.1, computed with the ground truth attributes for evaluation.

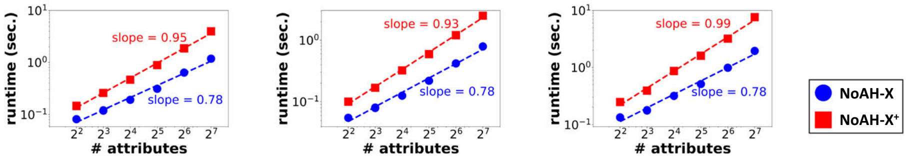
7.7.2 Performance comparison

As shown in Table 6, NOAH reproduces the structure–attribute interplay best overall, as expected, since it leverages pre-existing attributes during both fitting and generation. Among methods that do not rely on pre-existing attributes, NOAH- X^{+} performs best. It outperforms NOAH- X, hyper dK, and THERA in terms of average rank across interplay metrics. In particular, NOAH- X^{+} frequently ranks among the top methods across multiple metrics and is often competitive even with NOAH. These results indicate that NOAH- X^{+} faithfully reproduces attribute-dependent interaction patterns despite the absence of pre-existing attributes.

Table 6 Structure–attribute interplay evaluation in the setting without pre-existing attributes

	T2	T3	T4	HE	HOHE	NHS	A.R.	T2	T3	T4	HE	HOHE	NHS	A.R.
hyper dK	31.4	52.4	61.6	1.249	***0.450***	1.026	4.5	24.6	54.5	82.9	*0.726*	*0.511*	0.978	3.8
THERA	***26.0***	50.6	67.0	0.976	*0.394*	1.003	3.7	***21.0***	***54.3***	80.6	**0.704**	**0.484**	0.964	***2.7***
NOAH	**21.0**	**47.8**	***55.1***	**0.275**	**0.394**	**0.229**	**1.3**	**9.3**	*53.0*	**66.2**	0.967	***1.670***	**0.524**	**2.2**
NOAH-X	*24.9*	***48.7***	*54.8*	***0.305***	0.662	***0.307***	***3.0***	23.2	54.9	***71.9***	0.860	2.162	*0.601*	3.8
NoAH-X⁺	26.1	*48.6*	**53.1**	*0.292*	0.496	*0.306*	*2.5*	*16.3*	**52.5**	*70.2*	***0.832***	1.816	***0.629***	*2.5*
	(a) Amazon Music							(b) Yelp Restaurant						
hyper dK	57.1	108.7	134.2	1.095	*1.090*	1.220	4.0	5.0	4.7	4.3	3.7	*2.3*	4.7	4.5
THERA	***55.5***	106.0	***133.3***	1.124	**1.061**	1.221	3.5	***3.0***	3.7	4.0	***3.3***	**1.3**	4.3	***3.2***
NOAH	**44.7**	**94.0**	145.2	**0.589**	***1.598***	**0.468**	**2.0**	**1.0**	**1.3**	***3.0***	**2.3**	*2.3*	**1.0**	**1.5**
NOAH-X	55.5	*97.5*	**109.2**	***0.680***	2.208	***0.510***	***3.0***	3.3	***3.3***	*2.0*	***3.3***	5.0	***2.7***	3.3
NoAH-X⁺	*54.2*	***102.1***	*119.9*	*0.646*	1.990	*0.494*	*2.5*	*2.7*	*2.0*	**1.7**	**2.3**	4.0	*2.3*	*2.0*
	(c) Yelp Bar							(d) Average Rank over Review Domain						

As expected, NOAH, which leverages pre-existing node attributes, reproduces the interplay best overall. Among methods that do not access pre-existing attributes, NOAH- X⁺ performs best. The top three results are indicated by **boldface** (first), *italics* (second), and ***bolditalic*** (third). A.R. denotes average rank



(a) Generation time on Amazon Music

(b) Generation time on Yelp Restaurant

(c) Generation time on Yelp Bar

Fig. 7 Generation runtime of NoAH- X and NoAH- X$^+$ with respect to the latent attribute dimension on the review domain datasets. Both methods scale nearly linearly with the number of attributes, while NoAH- X$^+$ incurs higher runtime due to hyperedge-wise attribute sampling

As discussed above, NoAH- X$^+$ with hyperedge-wise attribute sampling generally outperforms NoAH- X with global attribute sampling. This is because hyperedge-wise attribute sampling increases the diversity of generated hyperedges, better reflecting the randomness encoded in $\mathbf{\Phi}$, whereas global attribute sampling tends to produce more highly similar hyperedges. This difference is illustrated in Fig. 6, where we compare hyperedge diversity profiles measured by the distribution of pairwise Jaccard similarity between (the node sets of) hyperedges. Note that NoAH- X$^+$ yields a hyperedge-diversity profile that more closely matches the ground truth, whereas NoAH- X produces a larger fraction of highly similar hyperedge pairs.

7.7.3 Sensitivity analysis

We examine the sensitivity of NoAH- X and NoAH- X$^+$ to the latent attribute dimension k. To this end, besides the setting where k is chosen to match the ground truth attribute dimension, we evaluate both methods with $k \in \{4, 8, 16, 32, 64, 128\}$ on the review domain datasets. The results are reported in Table 7. Overall, the choice of k affects the performance of both methods, and the best value depends on the dataset and generation scheme. While larger values of k tend to perform better than very small ones, the improvement is not consistent across all cases. This suggests that k can be treated as a hyperparameter and tuned based on the structure–attribute interplay measures. Notably, under most settings, NoAH- X and NoAH- X$^+$ outperform the baselines without pre-existing attributes, whose results are reported in Table 6.

7.7.4 Runtime analysis

We also compare the generation runtime of NoAH- X and NoAH- X$^+$ with respect to the latent attribute dimension k. Figure 7 reports the runtime on the review domain datasets as k varies over $\{4, 8, 16, 32, 64, 128\}$. Compared to NoAH- X, NoAH- X$^+$ incurs additional overhead because it resamples latent attributes for each generated hyperedge. Nevertheless, both methods scale near linearly with k, as indicated by regression line slopes smaller than one on a log-log scale.

7.8 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

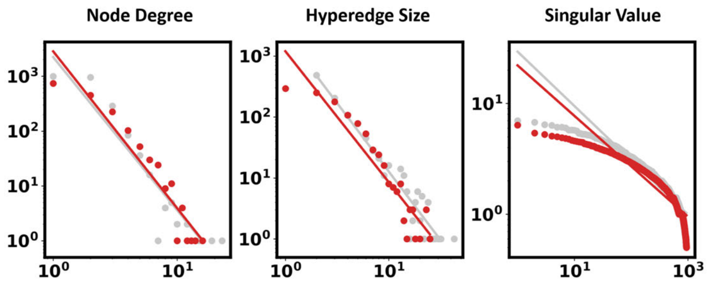
Table 7 Sensitivity analysis of the latent attribute dimension k in the setting without pre-existing attributes

	T2	T3	T4	HE	HOHE	NHS	A.R.	T2	T3	T4	HE	HOHE	NHS	A.R.
	(a) Amazon Music (GT dim.=7)							(b) Yelp Restaurant (GT dim.=9)						
NoAH-X (GT dim.)	24.9	48.3	55.5	0.300	***0.632***	0.311	4.8	21.8	51.8	69.0	***0.914***	2.183	0.591	5.0
NoAH-X (k=4)	25.4	48.4	53.6	0.314	0.666	0.397	5.8	22.9	*50.6*	73.0	**0.636**	**1.172**	0.769	***3.7***
NoAH-X (k=8)	21.3	48.1	55.5	0.301	0.633	***0.309***	4.5	19.0	62.1	78.3	*0.829*	*1.972*	0.639	4.8
NoAH-X (k=16)	***20.5***	47.1	55.3	0.389	1.156	**0.299**	4.5	10.5	**50.5**	73.0	0.926	2.285	***0.577***	4.0
NoAH-X (k=32)	**16.3**	***45.7***	*51.6*	**0.256**	0.652	0.320	*3.0*	**6.5**	51.8	***64.0***	0.944	2.115	**0.524**	*3.3*
NoAH-X (k=64)	*19.7*	*43.7*	**49.5**	*0.274*	*0.568*	**0.299**	**1.7**	***10.4***	***51.6***	*63.2*	0.927	***2.009***	*0.547*	**3.0**
NoAH-X (k=128)	22.9	**36.7**	53.8	***0.297***	**0.534**	0.376	***3.3***	7.4	54.0	**61.6**	0.975	2.039	0.591	4.0
NoAH-X^{+} (GT dim.)	26.1	48.6	53.1	0.292	*0.496*	0.306	4.7	16.3	52.5	70.2	*0.832*	*1.816*	0.629	4.0
NoAH-X^{+} (k=4)	*23.5*	48.0	59.4	*0.262*	0.523	0.392	4.3	19.2	55.9	78.7	**0.667**	**1.336**	0.731	5.0
NoAH-X^{+} (k=8)	25.1	56.9	52.6	***0.278***	***0.498***	**0.277**	4.2	16.5	53.9	70.4	***0.879***	1.837	0.602	4.8
NoAH-X^{+} (k=16)	24.6	52.5	**49.9**	0.333	0.662	***0.293***	4.8	14.1	***48.6***	*69.2*	0.934	1.876	0.555	4.0
NoAH-X^{+} (k=32)	24.3	**43.2**	51.8	0.322	0.581	*0.288*	*3.8*	***13.5***	49.5	**64.9**	0.924	1.932	**0.526**	*3.3*
NoAH-X^{+} (k=64)	***23.6***	***47.7***	*50.1*	0.313	0.561	0.318	*3.8*	*8.4*	*38.0*	***69.3***	0.977	2.013	***0.529***	***3.8***
NoAH-X^{+} (k=128)	**20.1**	*44.5*	***51.2***	**0.258**	**0.462**	0.377	**2.3**	*11.3*	**37.9**	76.8	0.919	***1.835***	**0.526**	**2.8**

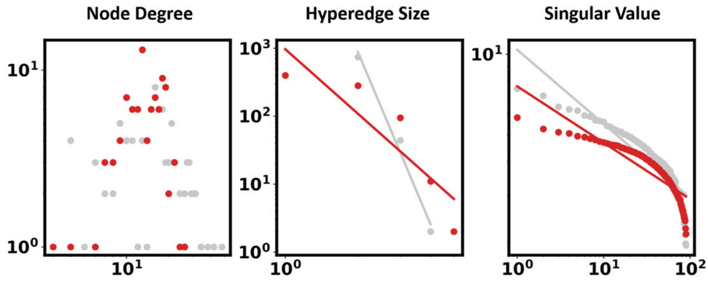
Table 7 continued

	T2	T3	T4	HE	HOHE	NHS	A.R.	T2	T3	T4	HE	HOHE	NHS	A.R.
	(a) Amazon Music (GT dim.=7)							(b) Yelp Restaurant (GT dim.=9)						
NoAH-X (GT dim.)	55.5	97.5	109.2	0.680	2.208	0.510	5.2	6.7	5.7	4.7	***3.7***	5.0	4.3	5.0
NoAH-X (k=4)	53.4	106.7	120.0	**0.460**	*1.809*	0.761	4.7	6.3	5.3	4.3	**2.7**	*2.7*	7.0	4.2
NoAH-X (k=8)	52.2	102.1	126.4	*0.564*	**1.798**	0.715	4.5	4.7	6.0	6.7	*3.0*	**2.3**	5.0	4.7
NoAH-X (k=16)	50.2	98.6	121.6	***0.673***	2.118	0.499	4.2	3.7	***3.0***	5.3	4.7	6.0	***2.7***	4.5
NoAH-X (k=32)	***48.6***	***83.9***	***106.2***	0.744	2.275	**0.424**	***4.0***	**1.7**	3.3	***2.7***	4.7	6.0	*2.3*	*3.5*
NoAH-X (k=64)	*46.5*	**77.7**	**86.2**	0.687	***2.047***	*0.437*	**2.3**	*2.3*	**2.0**	**1.3**	4.0	*2.7*	**1.7**	**1.8**
NoAH-X (k=128)	**42.8**	*81.5*	*90.8*	0.691	2.084	***0.461***	*3.2*	***2.7***	*2.7*	*2.3*	5.3	3.3	4.7	***3.8***
NoAH-X^{+} (GT dim.)	54.2	102.1	119.9	***0.646***	1.990	0.494	4.7	6.3	5.0	5.0	***3.0***	*2.7*	4.7	4.7
NoAH-X^{+} (k=4)	52.3	109.5	121.4	**0.510**	**1.769**	0.715	4.5	4.7	6.0	6.7	**1.3**	**2.0**	7.0	4.5
NoAH-X^{+} (k=8)	52.3	105.4	128.4	*0.598*	*1.927*	0.612	4.7	5.7	6.3	5.7	*2.7*	***3.0***	4.0	4.7
NoAH-X^{+} (k=16)	51.4	101.9	118.7	0.670	***1.980***	0.499	4.2	4.3	4.3	*2.3*	6.0	5.0	4.0	4.2
NoAH-X^{+} (k=32)	**39.8**	**86.0**	**106.9**	0.706	2.098	**0.436**	**3.0**	***2.7***	**2.0**	**2.0**	6.0	6.3	**1.3**	*3.2*
NoAH-X^{+} (k=64)	*43.4*	*87.9*	*108.7*	0.662	1.999	***0.451***	*3.2*	**2.0**	***2.3***	*2.3*	5.3	6.0	***3.7***	***3.3***
NoAH-X^{+} (k=128)	***45.0***	***89.3***	***109.4***	0.700	1.997	*0.441*	***3.7***	**2.0**	**2.0**	4.0	3.7	3.0	*3.0*	**2.5**
	(c) Yelp Bar (GT dim.=15)							(d) Average Rank over Review Domain						

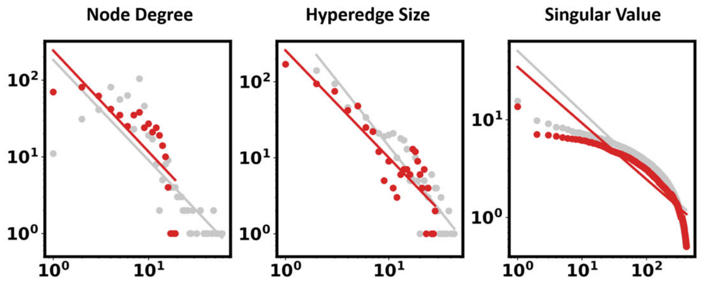
We report the performance of NoAH- X and NoAH- X^{+} when k is set to the ground truth attribute dimension (GT dim.) and when $k \in \{4, 8, 16, 32, 64, 128\}$. The top three results are indicated by **boldface** (first), *italics* (second), and ***bolditaic*** (third). A.R. denotes average rank



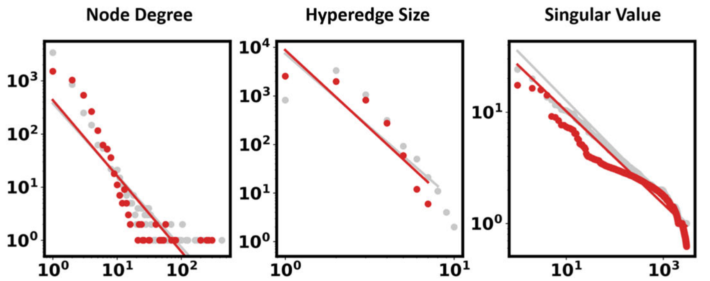
(a) Cora



(b) Workspace



(c) Yelp Restaurant



(d) Devops

Fig. 8 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 to follow heavy-tailed distributions in real-world hypergraphs [20]. In Fig. 8, we visually demonstrate that NoAH with NoAHFit reproduces such structural patterns, which closely resemble those of the input hypergraphs.[3] We also present a detailed evaluation on structural metrics in Appendix D, 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).

8 Conclusions and discussion

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. Across nine real-world hypergraphs and six structure–attribute interplay metrics, NoAH with NoAHFit achieves the best overall average rank among the nine evaluated methods. In addition, we broadened the applicability of NoAH in two directions. First, we extended NoAH and NoAHFit to support categorical and continuous node attributes through generalized affinity formulations. Second, for settings without pre-existing node attributes, we proposed NoAHFit- X, which jointly learns latent node attributes together with the parameters of NoAH, along with generation methods based on latent node attributes.

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 may support diverse applications in domains where hypergraphs naturally arise, enabling statistical analysis, simulation, and data anonymization.

Limitations and Future Directions
The design of NoAH is intentionally simple and interpretable, relying on a core–fringe node hierarchy and a multiplicative affinity formulation. This design choice supports scalability and interpretability but may limit the expressiveness of NoAH in capturing complex attribute–structure relationships in real-world hypergraphs. Our continuous attribute experiments partially illustrate this trade-off: The neural affinity variant, which is more expressive than the bilinear formulation, further improves the results at the cost of simplicity and interpretability. Moreover, NoAH captures dependencies among hyperedges only implicitly through shared parameters and assumes a fixed two-level node hierarchy. Future work may therefore explore multi-level or context-dependent node roles and explicitly model dependencies across hyperedges. Finally, NoAH is currently limited to static, homogeneous, undirected hypergraphs with unweighted hyperedges. Extending it to temporal and heterogeneous settings, as well as directed, weighted, and role-aware [37] group interactions, is a natural direction for future work.

[3] We note that the fit on the Workspace dataset appears relatively weaker than on the other datasets. A likely reason is that it has a distinctive structural pattern: it is the smallest dataset in our experiments, and its node degree and hyperedge size distributions are relatively discrete and narrow, with hyperedge sizes concentrated between 2 and 4. In contrast, NoAH is particularly effective at reproducing heavy-tailed structural patterns commonly observed in real-world hypergraphs. Thus, the narrower structural distributions of the Workspace dataset may make its structural patterns more challenging for NoAH to reproduce accurately.

Appendix A Detailed model configurations

The hyperparameter search space for each baseline and NoAH was set as follows:

- 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: 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: $\gamma \in \{0.0, 0.1, \ldots, 0.9\}$
- 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, $w_{deg} \in \{0, 0.1, 0.01\}$, $w_{card} \in \{0, 1, 2\}$.

Appendix B Experiments on temporal attributes

Although NoAH is designed as a static attributed hypergraph generator and does not explicitly model temporal dynamics, we additionally examine whether simple temporal information can be incorporated into NoAH as node attributes. To this end, we conduct additional experiments on the review domain datasets (Amazon Music, Yelp Restaurant, and Yelp Bar) and online Q&A-domain datasets (Devops and Patents), which are constructed from temporally ordered events.

For each node, we define its appearance timestamps based on the temporal events in which it participates. In the review domain datasets, where nodes represent reviewers and hyperedges represent groups of reviewers who reviewed the same product or business, the appearance timestamps of a node correspond to the review timestamps of the corresponding reviewer. In the online Q&A-domain datasets, where nodes represent users and hyperedges represent posts involving groups of users, the appearance timestamps of a node correspond to the timestamps of the posts in which the corresponding user participated.

Specifically, for each node, we construct temporal attributes using binary temporal bins. We divide the interval between the global minimum and maximum timestamps into equally spaced bins and assign each node a binary value for each bin indicating whether the node appeared in that bin. These binary temporal bin attributes are concatenated to the original binary node attributes. For Amazon Music, which has a relatively small original attribute dimension ($k = 7$), we use 3 temporal bins. For the other datasets, we use 10 temporal bins. Since both the original attributes and the temporal bin attributes are binary, all attributes are handled using the original binary affinity formulation of NoAH. The final attachment probability is obtained by multiplying the affinity scores over both the original binary attributes and the added temporal bin attributes, following the same multiplicative structure as NoAH.

We compare NoAH with additional temporal attributes (NoAH w/ time) against the original model with only binary attributes (NoAH w/o time), as well as hyper dK and THera, which are the two strongest baselines in terms of structure–attribute interplay. All methods are evaluated using the same structure–attribute interplay measures described in Sect. 7.1.

We report the results in Table 8. As shown in the table, adding temporal bin attributes generally improves the performance of NoAH. Compared with NoAH without temporal attributes, NoAH with temporal attributes achieves better average ranks on several datasets, including Yelp Restaurant, Yelp Bar, and Devops, and improves the overall average rank across the five temporal datasets. These results suggest that temporal information can provide useful additional signals even when simply incorporated as static binary node attributes.

Appendix C Experiments on modeling dependencies across latent Attributes

To further examine the effect of the independence assumption in the factorized Bernoulli formulation in Eq. (27), we conduct an experiment on modeling dependencies across latent attributes. The original formulation represents latent attributes as multiple independent binary dimensions and therefore does not explicitly model dependencies across latent attribute dimensions. As an alternative, we consider a dependency-aware categorical formulation that replaces multiple binary latent attributes with a single categorical latent attribute. This formulation can, in principle, represent dependencies among the corresponding binary states.

In this experiment, we use one categorical latent attribute with $2^5 = 32$ possible values, corresponding to the state space of five binary attributes. This design substantially increases the number of affinity parameters: the dependency-aware categorical formulation uses a $2^5 \times 2^5$ affinity matrix, containing 2^{10} parameters. For comparison, the largest binary setting in our sensitivity analysis uses $k = 128 = 2^7$, which corresponds to 2^7 binary 2×2 affinity matrices, i.e., 2^9 parameters in total.

Table 9 reports the results. Overall, the dependency-aware categorical formulation does not outperform the factorized binary formulation, even though it uses more affinity parameters than the largest binary setting. This suggests that, in our setting, the factorized Bernoulli formulation provides a favorable trade-off between expressiveness and parameter efficiency, despite its simplifying independence assumption.

Appendix D Performance comparison on structure

In this section, we evaluate NoAH along with 8 baseline generators in Sect. 7. We evaluate ten 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 [43].
- **S2. Pair Degree (Pair Deg.):** *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 nonzero 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 [43].
- **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., $|\{v \in \mathcal{V} : v \in e_1, v \in e_2\}|$.

Table 8 Effects of incorporating additional temporal attributes into NoAH on the review domain and online Q&A-domain datasets

	T2	T3	T4	HE	HOHE	NHS	A.R.	T2	T3	T4	HE	HOHE	NHS	A.R.
hyper dK	31.4	52.4	61.6	1.249	0.450	1.026	3.7	24.6	54.5	82.9	*0.726*	*0.511*	0.978	3.3
THERA	26.0	50.6	67.0	0.976	**0.394**	1.003	2.8	21.0	54.3	80.6	**0.704**	**0.484**	0.964	*2.3*
NoAH (w/o time)	**21.0**	*47.8*	*55.1*	**0.275**	**0.394**	*0.229*	**1.5**	*9.3*	*53.0*	**66.2**	0.967	1.670	*0.524*	2.5
NoAH (w/ time)	*25.5*	**45.6**	**54.0**	*0.368*	0.637	**0.202**	*1.8*	**8.6**	**50.8**	*74.5*	0.927	1.514	**0.521**	**1.8**
	(a) Amazon Music							(b) Yelp Restaurant						
hyper dK	57.1	108.7	134.2	1.095	*1.090*	1.220	3.2	**1,136**	**2,765**	**4,787**	**6.86**	28.56	10.27	**1.7**
THERA	55.5	106.0	*133.3*	1.124	**1.061**	1.221	2.8	2,387	6,567	9,136	25.90	65.53	20.52	4.0
NoAH (w/o time)	*44.7*	*94.0*	145.2	*0.589*	1.598	*0.468*	*2.5*	1,714	5,182	8,038	12.81	*19.75*	**6.59**	2.5
NoAH (w/ time)	**42.2**	**88.6**	**124.1**	**0.584**	1.663	**0.458**	**1.5**	*1,672*	*5,123*	*7,539*	*11.03*	**17.52**	*7.56*	*1.8*
	(c) Yelp Bar							(d) Devops						
hyper dK	**6,262**	**15,041**	23,320	**53.4**	**85.05**	**91.81**	**1.3**	2.8	2.8	2.8	**2.2**	**2.2**	3.0	*2.3*
THERA	12,856	26,900	30,711	266.45	464.98	232.26	4.0	3.4	3.4	3.4	3.2	**2.2**	3.6	3.5
NoAH (w/o time)	7,292	17,557	*22,532*	*65.32*	*109.78*	*134.08*	*2.3*	*2.2*	*2.4*	*2.4*	2.4	2.4	*1.8*	*2.3*
NoAH (w/ time)	*7,066*	*14,485*	**22,131**	69.90	115.42	134.65	*2.3*	**1.6**	**1.4**	**1.4**	**2.2**	3.0	**1.6**	**1.5**
	(e) Patents							(f) Average Rank over Five Datasets						

The best results are shown in **bold**, and the second-best results are *italic*. A.R. denotes average rank

Table 9 Effects of modeling dependencies across latent attributes in the setting without pre-existing attributes

	T2	T3	T4	HE	HOHE	NHS	T2	T3	T4	HE	HOHE	NHS
NoAH-X (Independent)	**22.9**	**36.7**	**53.8**	**0.297**	0.534	**0.376**	**7.4**	**54.0**	**61.6**	0.975	2.039	**0.591**
NoAH-X (Dependent)	26.8	67.5	69.2	0.666	**0.447**	1.001	19.1	63.1	79.9	**0.351**	**0.583**	1.100
NoAH-X$^+$ (Independent)	**20.1**	**44.5**	**51.2**	**0.258**	0.462	**0.377**	**11.3**	**37.9**	76.8	0.919	1.835	**0.526**
NoAH-X$^+$ (Dependent)	21.4	58.3	70.4	0.730	**0.420**	0.965	19.4	62.2	**75.3**	**0.319**	**0.665**	0.916
	(a) Amazon Music						(b) Yelp Restaurant					
NoAH-X (Independent)	**42.8**	**81.5**	**90.8**	0.691	2.084	**0.461**	**1.0**	**1.0**	**1.0**	1.7	2.0	**1.0**
NoAH-X (Dependent)	57.0	113.1	128.5	**0.438**	**0.829**	1.352	2.0	2.0	2.0	**1.3**	**1.0**	2.0
NoAH-X$^+$ (Independent)	**45.0**	**89.3**	**109.4**	0.700	1.997	**0.441**	**1.0**	**1.0**	**1.3**	1.7	2.0	**1.0**
NoAH-X$^+$ (Dependent)	56.2	115.4	136.2	**0.391**	**0.875**	1.044	2.0	2.0	1.7	**1.3**	**1.0**	2.0
	(c) Yelp Bar						(d) Average Rank over Review Domain					

We compare the factorized binary formulation (**Independent**) with the dependency-aware categorical formulation (**Dependent**). Better results are highlighted in **bold**

The distribution of nonzero intersection size tends to be heavy-tailed in real-world hypergraphs [43].

- **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 [46], and in many real-world hypergraphs, they are often highly skewed [43].
- **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:

$$C_v := 2 \times \frac{\text{the number of triangles involving } v}{\text{the number of connected triplets involving } v}, \tag{D1}$$

  We estimate the *global clustering coefficient* by averaging local clustering coefficients in the clique expansion (defined in S6) of a hypergraph using [47]. 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 the ratio of the number of nodes to the number of hyperedges, i.e., $\frac{|\mathcal{V}|}{|\mathcal{E}|}$ [48]. Hypergraphs within the same domain tend to exhibit similar levels of density significance [7].
- **S9. Overlapness (Overlap.):** 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 (Diam.):** 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 [49]. This statistic tends to be small in real-world hypergraphs [43].

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 10. Although NOAH primarily targets structure–attribute interplay, it also performs competitively on purely structural patterns, ranking an average of 4.6 compared with 8 structure-focused methods.

Table 10 Structural metric evaluation across 9 datasets. A.R. denotes average rank

	Degree	Pair Deg.	Size	Int. Size	SV	CC	GCC	Density	Overlap.	Diam.	A.R.
(a) Citeseer											
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	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
(b) Cora											
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	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

Table 10 continued

	Degree	Pair Deg.	Size	Int. Size	SV	CC	GCC	Density	Overlap.	Diam.	A.R.
(c) High School											
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	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
(d) Workspace											
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	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

Table 10 continued

	Degree	Pair Deg.	Size	Int. Size	SV	CC	GCC	Density	Overlap.	Diam.	A.R.
(e) Amazon Music											
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	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
(f) Yelp Restaurant											
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	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

Table 10 continued

	Degree	Pair Deg.	Size	Int. Size	SV	CC	GCC	Density	Overlap.	Diam.	A.R.
(g) Yelp Bar											
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	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
(h) Devops											
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	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

Table 10 continued

	Degree	Pair Deg.	Size	Int. Size	SV	CC	GCC	Density	Overlap.	Diam.	A.R.
(i) Patents											
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	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
(j) Average Rank over Nine Datasets (NoAH ranks **fifth** overall)											
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	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

Author Contributions J.C., S.Y., M.C., G.L., and K.S. contributed to the study conception and methodological design. J.C. and S.Y. conducted the theoretical analysis, developed the software, performed the experiments, and wrote the draft of the manuscript. M.C., G.L., and K.S. revised the manuscript. K.S. supervised the project and acquired funding. All authors reviewed the manuscript.

Funding Open Access funding enabled and organized by KAIST. This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-II220871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration) (No. RS-2024-00457882, AI Research Hub Project) (RS-2019-II190075, Artificial Intelligence Graduate School Program (KAIST)). This work was partly supported by the Collabo R&D between Industry, Academy, and Research Institute (No. RS-2025-02314596) funded by the Ministry of SMEs and Startups (MSS, Korea).

Data availability Our source code and data are available at https://github.com/jaewan01/NoAH.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Kumar T, Vaidyanathan S, Ananthapadmanabhan H, Parthasarathy S, Ravindran B (2020) Hypergraph clustering by iteratively reweighted modularity maximization. Appl Netw Sci 5(1):52. https://doi.org/10.1007/s41109-020-00300-3
2. Hayashi K, Aksoy SG, Park CH, Park H (2020) Hypergraph random walks, laplacians, and clustering. In: ACM international conference on information and knowledge management. https://doi.org/10.1145/3340531.3412034
3. Yu J, Tao D, Wang M (2012) Adaptive hypergraph learning and its application in image classification. IEEE Trans Image Process 21(7):3262–3272. https://doi.org/10.1109/TIP.2012.2190083
4. Silva J, Willett R (2008) Hypergraph-based anomaly detection of high-dimensional co-occurrences. IEEE Trans Pattern Anal Mach Intell 31(3):563–569. https://doi.org/10.1109/TPAMI.2008.232
5. Chun J, Lee G, Shin K, Jung J (2024) Random walk with restart on hypergraphs: fast computation and an application to anomaly detection. Data Min Knowl Disc 38(3):1222–1257. https://doi.org/10.1007/s10618-023-00995-9
6. Do MT, Yoon S, Hooi B, Shin K (2020) Structural patterns and generative models of real-world hypergraphs. In: ACM SIGKDD conference on knowledge discovery and data mining. https://doi.org/10.1145/3394486.3403060
7. Lee G, Choe M, Shin K (2021) How do hyperedges overlap in real-world hypergraphs?-patterns, measures, and generators. In: The web conference. https://doi.org/10.1145/3442381.3450010
8. Kim S, Bu F, Choe M, Yoo J, Shin K (2023) How transitive are real-world group interactions?-measurement and reproduction. In: ACM SIGKDD conference on knowledge discovery and data mining. https://doi.org/10.1145/3580305.3599382
9. Lee G, Bu F, Eliassi-Rad T, Shin K (2025) A survey on hypergraph mining: patterns, tools, and generators. ACM Comput Surv 57(8):1–36. https://doi.org/10.1145/3719002
10. Ghoshdastidar D, Dukkipati A (2017) Consistency of spectral hypergraph partitioning under planted partition model. Ann Stat 45(1):289–315
11. Ruggeri N, Contisciani M, Battiston F, De Bacco C (2023) Community detection in large hypergraphs. Sci Adv 9(28):9159. https://doi.org/10.1126/sciadv.adg9159

12. Badalyan A, Ruggeri N, De Bacco C (2024) Structure and inference in hypergraphs with node attributes. Nat Commun 15(1):7073. https://doi.org/10.1038/s41467-024-51388-5
13. Contisciani M, Battiston F, De Bacco C (2022) Inference of hyperedges and overlapping communities in hypergraphs. Nat Commun 13(1):7229. https://doi.org/10.1038/s41467-022-34714-7
14. McPherson M, Smith-Lovin L, Cook JM (2001) Birds of a feather: Homophily in social networks. Ann Rev Sociol 27(1):415–444. https://doi.org/10.1146/annurev.soc.27.1.415
15. Lee G, Lee SY, Shin K (2024) Villain: Self-supervised learning on homogeneous hypergraphs without features via virtual label propagation. In: The web conference. https://doi.org/10.1145/3589334.3645454
16. Chun J, Yoon S, Choe M, Lee G, Shin K (2025) Attributed hypergraph generation with realistic interplay between structure and attributes. In: IEEE international conference on data mining
17. Chakrabarti D, Faloutsos C (2006) Graph mining: laws, generators, and algorithms. ACM Comput Surv 38(1):2. https://doi.org/10.1145/1132952.1132954
18. Jia J, Benson AR (2019) Random spatial network models for core-periphery structure. In: ACM international conference on web search and data mining. https://doi.org/10.1145/3289600.3290976
19. Papachristou M, Kleinberg J (2022) Core-periphery models for hypergraphs. In: ACM SIGKDD conference on knowledge discovery and data mining. https://doi.org/10.1145/3534678.3539272
20. Ko J, Kook Y, Shin K (2022) Growth patterns and models of real-world hypergraphs. Knowl Inf Syst 64(11):2883–2920. https://doi.org/10.1007/s10115-022-01739-9
21. Seidman SB (1983) Network structure and minimum degree. Soc Netw 5(3):269–287. https://doi.org/10.1016/0378-8733(83)90028-X
22. Borgatti SP, Everett MG (2000) Models of core/periphery structures. Soc Netw 21(4):375–395. https://doi.org/10.1016/S0378-8733(99)00019-2
23. Bu F, Lee G, Shin K (2023) Hypercore decomposition for non-fragile hyperedges: concepts, algorithms, observations, and applications. Data Min Knowl Disc 37(6):2389–2437. https://doi.org/10.1007/s10618-023-00956-2
24. Tudisco F, Higham DJ (2023) Core-periphery detection in hypergraphs. SIAM J Math Data Sci 5(1):1–21. https://doi.org/10.1137/22M1480926
25. Simonovsky M, Komodakis N (2018) Graphvae: towards generation of small graphs using variational autoencoders. In: International conference on artificial neural networks. https://doi.org/10.1007/978-3-030-01418-6_41
26. Bojchevski A, Shchur O, Zügner D, Günnemann S (2018) Netgan: Generating graphs via random walks. In: International conference on machine learning
27. De Cao N, Kipf T (2018) MolGAN: an implicit generative model for small molecular graphs. In: ICML 2018 workshop on theoretical foundations and applications of deep generative models. https://doi.org/10.48550/arXiv.1805.11973
28. Liu C, Fan W, Liu Y, Li J, Li H, Liu H, Tang J, Li Q (2023) Generative diffusion models on graphs: methods and applications. In: International joint conferences on artificial intelligence. https://doi.org/10.24963/ijcai.2023/751
29. Robins G, Pattison P, Kalish Y, Lusher D (2007) An introduction to exponential random graph (p*) models for social networks. Soc Netw 29(2):173–191. https://doi.org/10.1016/j.socnet.2006.08.002
30. Airoldi EM, Blei D, Fienberg S, Xing E (2008) Mixed membership stochastic blockmodels. In: Annual conference on neural information processing systems
31. Wang S, Paul S, De Boeck P (2023) Joint latent space model for social networks with multivariate attributes. Psychometrika 88(4):1197–1227. https://doi.org/10.1007/s11336-023-09926-5
32. Kim M, Leskovec J (2012) Multiplicative attribute graph model of real-world networks. Internet Math 8(1–2):113–160. https://doi.org/10.1080/15427951.2012.625257
33. Amburg I, Kleinberg J, Benson AR (2021) Planted hitting set recovery in hypergraphs. J Phys Complex 2(3):035004. https://doi.org/10.1088/2632-072X/abdb7d
34. Veldt N, Benson AR, Kleinberg J (2023) Combinatorial characterizations and impossibilities for higher-order homophily. Sci Adv 9(1):3200. https://doi.org/10.1126/sciadv.abq3200
35. Li M, Gu Y, Wang Y, Fang Y, Bai L, Zhuang X, Lio P (2025) When hypergraph meets heterophily: New benchmark datasets and baseline. In: AAAI conference on artificial intelligence. https://doi.org/10.1609/aaai.v39i17.34022
36. Chitra U, Raphael B (2019) Random walks on hypergraphs with edge-dependent vertex weights. In: International conference on machine learning
37. Choe M, Kim S, Yoo J, Shin K (2023) Classification of edge-dependent labels of nodes in hypergraphs. In: ACM SIGKDD conference on knowledge discovery and data mining
38. Yadati N, Nimishakavi M, Yadav P, Nitin V, Louis A, Talukdar P (2019) Hypergcn: A new method for training graph convolutional networks on hypergraphs. In: Annual conference on neural information processing systems

39. Chodrow PS, Veldt N, Benson AR (2021) Generative hypergraph clustering: from blockmodels to modularity. Sci Adv 7(28):1303. https://doi.org/10.1126/sciadv.abh1303
40. Génois M, Barrat A (2018) Can co-location be used as a proxy for face-to-face contacts? EPJ Data Sci 7(1):1–18. https://doi.org/10.1140/epjds/s13688-018-0140-1
41. Ni J, Li J, McAuley J (2019) Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In: Conference on empirical methods in natural language processing. https://doi.org/10.18653/v1/D19-1018
42. Amburg I, Veldt N, Benson AR (2020) Hypergraph clustering for finding diverse and experienced groups. https://doi.org/10.48550/arXiv.2006.05645. arXiv:2006.05645
43. Kook Y, Ko J, Shin K (2020) Evolution of real-world hypergraphs: Patterns and models without oracles. In: IEEE international conference on data mining. https://doi.org/10.1109/ICDM50108.2020.00036
44. Nakajima K, Shudo K, Masuda N (2021) Randomizing hypergraphs preserving degree correlation and local clustering. IEEE Trans Netw Sci Eng 9(3):1139–1153. https://doi.org/10.1109/TNSE.2021.3133380
45. Choe M, Ko J, Kwon T, Shin K, Faloutsos C (2025) Kronecker generative models for power-law patterns in real-world hypergraphs. In: The web conference. https://doi.org/10.1145/3696410.3714893
46. Wall ME, Rechtsteiner A, Rocha LM (2003) Singular value decomposition and principal component analysis. In: A practical approach to microarray data analysis, 91–109. https://doi.org/10.1007/0-306-47815-3_5
47. Seshadhri C, Pinar A, Kolda TG (2014) Wedge sampling for computing clustering coefficients and triangle counts on large graphs. Stat Anal Data Mining ASA Data Sci J 7(4):294–307. https://doi.org/10.1002/sam.11224
48. Hu S, Wu X, Chan TH (2017) Maintaining densest subsets efficiently in evolving hypergraphs. In: ACM international conference on information and knowledge management. https://doi.org/10.1145/3132847.3132907
49. Leskovec J, Kleinberg J, Faloutsos C (2005) Graphs over time: densification laws, shrinking diameters and possible explanations. In: ACM SIGKDD conference on knowledge discovery and data mining. https://doi.org/10.1145/1081870.1081893

Jaewan Chun is a Ph.D. student at the Graduate School of AI at KAIST. He received his M.S. degree from the Graduate School of AI and his B.S. degree from the School of Computing, both at KAIST. His research interests include data mining, with a particular focus on hypergraph mining and spatio-temporal prediction. He received the Best Paper Award at ICDM 2025.

Seokbum Yoon is a machine learning engineer at RAPPORT LABS. He received his M.S. from the Kim Jaechul Graduate School of AI at KAIST in 2025 and his B.S. from the Department of Electrical Engineering at KAIST in 2019. His research interests include recommendation systems, graph/hypergraph data mining, and graph neural networks. He won the Best Paper Award at ICDM 2025.

Minyoung Choe is a Ph.D. student at KAIST. She received her B.S. in Computer Science and Engineering from Sogang University in 2021. Her research focuses on data mining and machine learning, particularly on graphs, hypergraphs, and knowledge graphs. Her work spans structural analysis, generative modeling, representation learning, and the construction and utilization of knowledge graphs for retrieval and multiagent systems.

Geon Lee is a postdoctoral researcher at KAIST. He received his Ph.D. from the Kim Jaechul Graduate School of AI at KAIST in 2026 and his B.S. from the Department of Computer Science and Engineering at Sungkyunkwan University in 2019. His research interests include graph and hypergraph mining, recommender systems, and large language models, with a focus on learning from relational data and developing reliable intelligent systems. He received the Ph.D. Dissertation Award from the KAIST College of Engineering in 2026.

Kijung Shin is an Associate Professor in the Kim Jaechul Graduate School of AI at KAIST, Republic of Korea. He received his Ph.D. in Computer Science from Carnegie Mellon University in 2019. His research focuses on data mining, particularly on (hyper)graphs, data streams, and spatio-temporal data. He has published over 100 referred articles in major data mining, database, and machine learning venus. He has received four ’best paper’ awards.